

BLM 5113 - Sayısal Video İşleme

CIFAR-10 Veri Seti için CNN Mimarisi Kullanılarak
Sınıflandırıcı Tasarımı

Metin Uslu

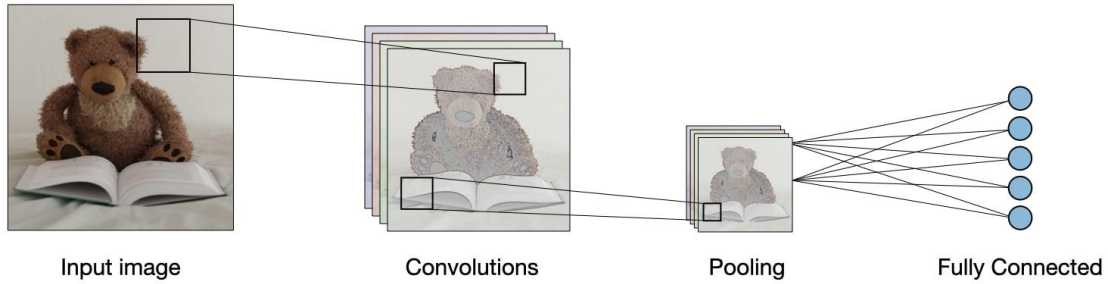
235B7014

1. Özet

Bu çalışmada CIFAR-10 veri seti üzerinde Konvolüsyonel(Evrişimli) Sinir Ağları (Convolutional Neural Network (CNN)) mimarisi kullanılarak çok sınıflı (multi class) sınıflandırma yapılması amaçlanmıştır. CNN mimarisi, derin öğrenme alanında yaygın olarak kullanılan ve görüntü işleme görevlerinde etkili sonuçlar veren bir modeldir. CIFAR-10 veri seti, 10 sınıflı, 32x32 en x boy oranına sahip ve 60K örnek görsellerden oluşan popüler bir veri setidir. Bu çalışmada, CIFAR-10 veri kümesi üzerinde CNN kullanılarak görüntü sınıflandırma performansının incelenmesi amaçlanmıştır. Derin Öğrenmenin son yıllarda popüleritesinin artmasında CNN mimarileri oldukça önemli bir yer tutmakta ve klasik Makine Öğrenmesi algoritmalarından öte performansları ile adından söz ettirmektedirler.

Çalışmada ilk olarak; veri seti hakkında temel açıklayıcı bilgilerin paylaşılması, ardından Konvolüsyonel Sinir Ağı mimarisi ve mimariyi etkileyen Evrişim Katmanı Sayısı, Hiper Parametre Seçimi (Evrişim Sayısı, Filtre Size, Filter Count), Modelin Kurulmasına değinilecektir. Son olarak; Train ve Validation veri seti ile model parametrelerinin optimize edilmesi, Test veri seti ile performans ölçülmesi ve çok sınıflı sınıflandırma modeli için karışıklık matrisinin çizilmesi ve değerlendirilmesi yapılacaktır.

2. Giriş



Kaynak: <https://stanford.edu/~shervine/teaching/cs-230/illustrations/architecture-cnn-en.jpeg?3b7fccd728e29dc619e1bd8022bf71cf>

Derin Öğrenmenin son yıllarda popüleritesinin artmasında (AlexNet Mimarisi ve ImageNet Yarışması 2012)CNN mimarileri oldukça önemli bir yer tutmakta ve klasik Makine Öğrenmesi algoritmalarından öte performansları ile adından söz ettirmektedirler. Görüntü sınıflandırma, bilgisayarlı görü(görüş) alanının en önemli görevlerinden bir tanesi olup, birçok bilinen ve kullandığımız uygulama için temel bir bileşenidir. CIFAR-10, genellikle görüntü sınıflandırma algoritmalarının performansını değerlendirmek için kullanılan popüler bir veri kümelerinden bir tanesidir. CIFAR-10, 10 farklı (airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks) sınıfa ait renkli görüntüler (RGB) içeren bir veri kümesidir ve toplamda 60K görüntü içerir. Bu çalışmada istendiği üzere, Eğitim, Doğrulama ve Test veri setleri 40K, 10K ve 10K olmak üzere rastgele olacak şekilde bölümlenme (split) edilmiştir. Ardından 32x32'lik RGB görseller (0 ile 255 arasında değer aldığı bilinen) normalize edilerek piksel değerleri 0-1 aralığına çekilmiştir.

Derin Öğrenme algoritmaları, yarı-yapısal (semi-structure) ve yapısal(unstructure) olmayan veri setleri üzerinde oldukça başarılı sonuçlar alınmaktadır. Buradaki başarıyı sağlayan en temel şeyler, veri büyüklüğü, veri içerisinden özelliklerin algoritma tarafından çıkarılması olarak bilinmektedir. CNN algoritmaları özellikle görsel veri setleri üzerinde Görüntü Sınıflandırma, Nesne Tanıma, Görüntü Segmentasyon, Yüz Tanıma, Yüz Tanımlama ve Stil Transferi gibi amaçlar için yaygın olarak kullanılmaktadır. En popüler CNN algoritmaları, LeNet-5, AlexNet, VGGNet, GoogLeNet(Inception), ResNet olarak belirtilebilir.

Bu çalışma da CIFAR-10 veri seti (RGB ve 32X32) üzerinde CNN mimarisi ile çoklu sınıflandırma yapılmaktadır. CNN mimarisi, Konvolüsyon Katmanları(Convolution layer (CONV)), Pooling Katmanları(Pooling (POOL)), Full Bağlantılı Katmanlar(Fully Connected (FC)), Aktivasyon Fonksiyonları ve Hiperparametrelerden (Filtre Boyutu ve Sayısı, Adım Sayısı(Stride), Doldurma(Padding)) genel olarak oluşmaktadır.

3. Yöntem

Bu çalışmada CNN mimarisi 3 ve 5 olmak üzere 2 farklı konvolüsyon ve 3x3 ve 5x5 olmak üzere 2 farklı filtre boyutu mimariler oluşturulmuştur. Toplam 2x2 olmak üzere 4 tane farklı mimari CIFAR-10 veri seti üzerinden sınıflandırma için değerlendirilmiştir. Bu 4 farklı mimari için de sistem kaynakları (cpu and ram) göz önüne alınarak; **batch_size** (mini_batch_size) değeri 64 ve **epoch_size** 10 olarak belirlenmiştir.

Mimari-1: 3 Konvolüsyon Katmanlı ve 64 Adet 3x3 Filtre Boyutlu Mimari

Mimari-2: 3 Konvolüsyon Katmanlı ve 64 Adet 5x5 Filtre Boyutlu Mimari

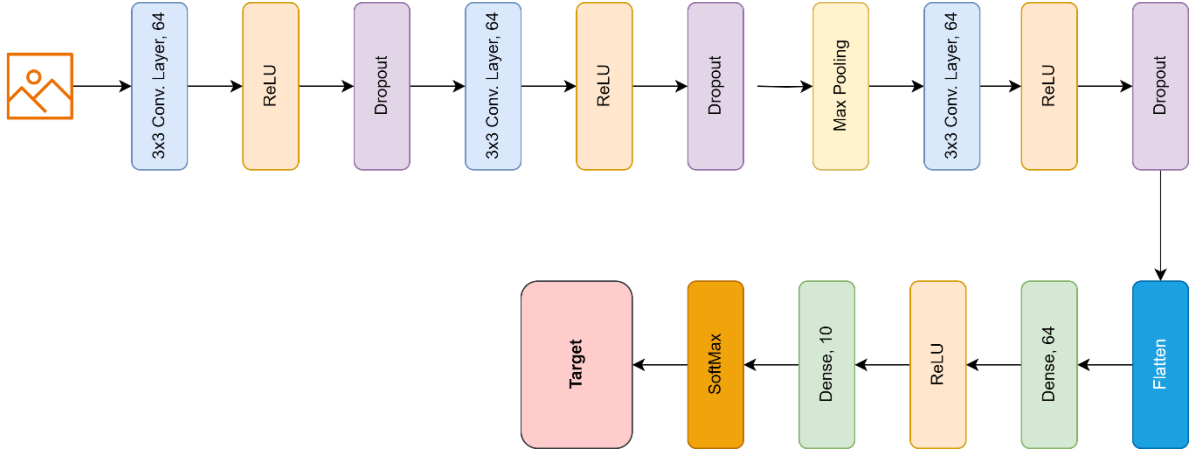
Mimari-3: 5 Konvolüsyon Katmanlı ve 64 Adet 3x3 Filtre Boyutlu Mimari

Mimari-4: 5 Konvolüsyon Katmanlı ve 64 Adet 5x5 Filtre Boyutlu Mimari

3.1. 3 Konvolüsyon Katmanlı ve 64 Adet [3x3 ve 5x5] Filtre Boyutlu Mimari Hakkında

Bu CNN mimarisinde 3 adet konvolüsyon katmanı bulunup her bir konvolüsyon katmanında 64 adet 3x3 ve 5x5'lik olmak üzere 2 farklı filtre bulunmaktadır. Konvolüsyon katmanlarında ReLU aktivasyon fonksiyonu kullanılarak doğrusal olmayan (non-linear) özelliklerin elde edilmesi sağlanmış ve son katmanında ise çoklu sınıflandırma (multi class) için SoftMax aktivasyon fonksiyonu kullanılmıştır. Her Konvolüsyon katmanından sonra MaxPooling kullanılarak boyutlar ve parametreler azaltılmış öğrenmenin hızlanması amaçlanmıştır. Regülerizasyon tekniği olarak dropout katmanı eklenmiş ve belirli bir yüzde ile (0.2) nöronların bağlantılarını sıfırlayarak aşırı öğrenmeyi önlenmesi ve genelleştirilebilir bir model elde edilmesi amaçlanmıştır. Konvolüsyon katmanından hemen sonra özellikleri düzleştirmek için Flatt edilmiş ve ardından 2 adet Full Connected layer 64 ve 10 adet olmak üzere eklenerek çoklu sınıflandırma mimarisi tasarlanmıştır. Model eğitimi aşamında Adam optimizasyon algoritması kullanılmış, model metriği olarak accuracy, loss için SparseCategoricalCrossentropy kayıp fonksiyonu kullanılmıştır.

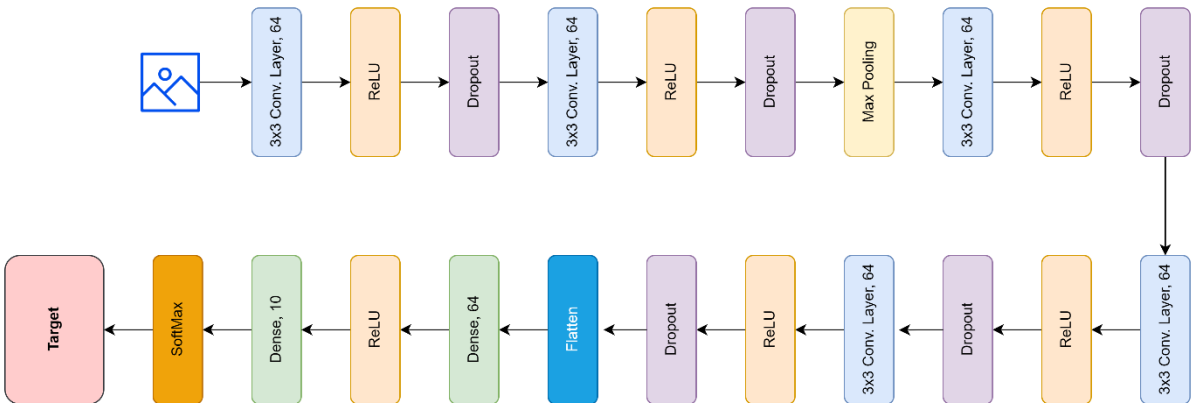
- ❖ 3 Konvolüsyon Katmanlı ve 64 Adet 3x3 Filtre Boyutlu Mimari için Parametre Sayısı: 666,186
- ❖ 3 Konvolüsyon Katmanlı ve 64 Adet 5x5 Filtre Boyutlu Mimari için Parametre Sayısı: 472,650



3.2. 5 Konvolüsyon Katmanlı ve 64 Adet [3x3 ve 5x5] Filtre Boyutlu Mimari Hakkında

Bu CNN mimarisinde 5 adet konvolüsyon katmanı bulunup her bir konvolüsyon katmanında 64 adet 3x3 ve 5x5'lik olmak üzere 2 farklı filtre bulunmaktadır. Konvolüsyon katmanlarında ReLU aktivasyon fonksiyonu kullanılarak doğrusal olmayan (non-linear) özelliklerin elde edilmesi sağlanmış ve son katmanında ise çoklu sınıflandırma (multi class) için SoftMax aktivasyon fonksiyonu kullanılmıştır. Her Konvolüsyon katmanından sonra MaxPooling kullanılarak boyutlar ve parametreler azaltılmış öğrenmenin hızlanması amaçlanmıştır. Regülerizasyon tekniği olarak dropout katmanı eklenmiş ve belirli bir yüzde ile (0.2) nöronların bağlantılarını sıfırlayarak aşırı öğrenmeyi önlenmesi ve genelleştirilebilir bir model elde edilmesi amaçlanmıştır. Konvolüsyon katmanından hemen sonra özellikleri düzleştirmek için Flatten edilmiş ve ardından 2 adet Full Connected layer 64 ve 10 adet olmak üzere eklenerek çoklu sınıflandırma mimarisi tasarlanmıştır. Model eğitimi aşamında Adam optimizasyon algoritması kullanılmış, model metriği olarak accuracy, loss için SparseCategoricalCrossentropy kayıp fonksiyonu kullanılmıştır.

- ❖ 5 Konvolüsyon Katmanlı ve 64 Adet 3x3 Filtre Boyutlu Mimari için Parametre Sayısı: 412,362
- ❖ 5 Konvolüsyon Katmanlı ve 64 Adet 5x5 Filtre Boyutlu Mimari için Parametre Sayısı: 415,434



4. Uygulama

Bu çalışmada Python dili ve en yaygın derin öğrenme kütüphanelerinden biri olan Tensorflow kullanılmıştır. Uygulama yerel geliştirme ortamı üzerinde CPU: Intel Core i7-7700HQ CPU 2.8 GHZ ve

Ram: 16 GB eğitilmiştir. Aşağıdaki tabloda 3 ve 5 katmanlı Konvolüsyonel Sinir Ağları üzerinde 3x3 ve 5x5 filtreler için Test veri seti üzerinde model performansları gösterilmiştir. 5 katmanlı ve 3x3 filtre boyutlu CNN modelinin 0.742 ile en iyi performans sahip olduğu görülmüştür. Yine 3 katmanlı ve 3x3 filtre boyutuna sahip diğer modelin en iyi performansa sahip 2. Model olduğu görülmektedir.

Filter Size Num. of Conv. Layer	3x3	5x5
3	0.692	0.667
5	0.742	0.677

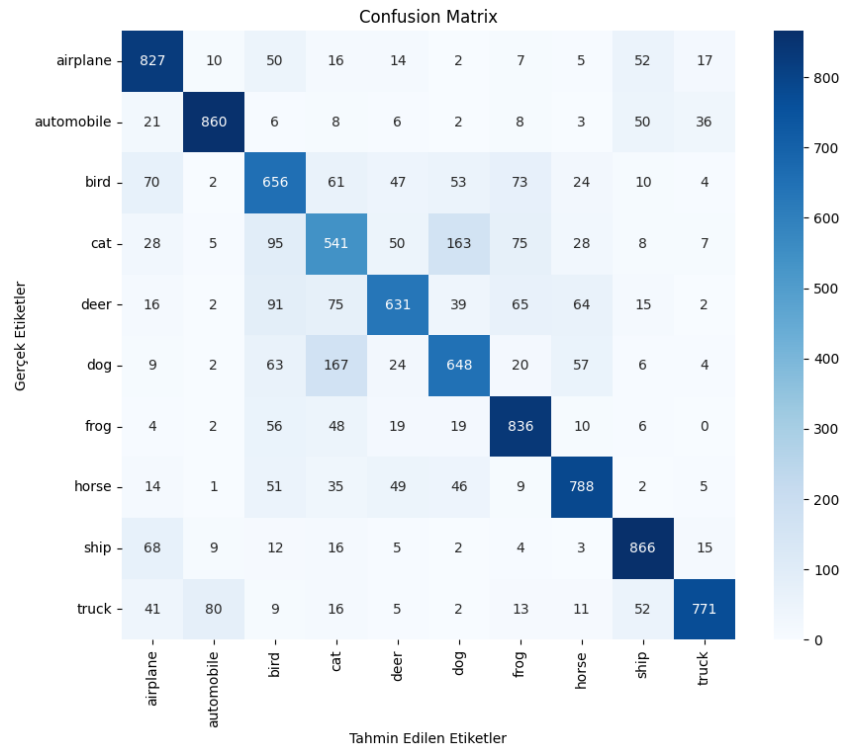
- i. Elinizdeki her bir sınıf için (toplamda 10 adet sınıf var) 2’şer tane rasgele resim seçiniz. Sorgu olarak kullandığınız resmi ve benzerlik sırasına göre en benzer bulduğu 5 sınıfı ve her sınıfın benzerlik olasılığını bir tabloda veriniz.

```

Class:airplane & ImageId:[5951, 237]
1/1 ----- 0s 28ms/step
1/1 ----- 0s 27ms/step
Sample-1 & ImageId:[5951 & Class&Proba[('airplane', 0.98439723), ('bird', 0.014498315), ('deer', 0.0004658232), ('cat', 0.0003598381), ('ship', 0.00013822354)]
Sample-2 & ImageId:[237 & Class&Proba[('bird', 0.30328687), ('airplane', 0.23042133), ('deer', 0.15728563), ('ship', 0.00028102), ('frog', 0.00314179)]]
-----
Class:automobile & ImageId:[4736, 7413]
1/1 ----- 0s 44ms/step
1/1 ----- 0s 39ms/step
Sample-1 & ImageId:[4736 & Class&Proba[('automobile', 0.99998677), ('truck', 7.572496e-06), ('airplane', 4.7792234e-06), ('ship', 6.7624484e-07), ('cat', 9.026776e-08)]
Sample-2 & ImageId:[7413 & Class&Proba[('automobile', 0.49624616), ('frog', 0.23280865), ('ship', 0.18780854), ('cat', 0.032500096), ('deer', 0.02495189)]]
-----
Class:bird & ImageId:[7670, 9917]
1/1 ----- 0s 26ms/step
1/1 ----- 0s 25ms/step
Sample-1 & ImageId:[7670 & Class&Proba[('bird', 0.55630976), ('deer', 0.39428917), ('frog', 0.024581948), ('horse', 0.01805142), ('dog', 0.0053649023)]
Sample-2 & ImageId:[9917 & Class&Proba[('bird', 0.81815654), ('frog', 0.14842747), ('airplane', 0.01758492), ('deer', 0.009201703), ('cat', 0.0030100807)]]
-----
Class:cat & ImageId:[1386, 5934]
1/1 ----- 0s 17ms/step
1/1 ----- 0s 38ms/step
Sample-1 & ImageId:[1386 & Class&Proba[('cat', 0.7719391), ('frog', 0.15986659), ('dog', 0.037223037), ('bird', 0.012012514), ('airplane', 0.007933517)]
Sample-2 & ImageId:[5934 & Class&Proba[('horse', 0.33293924), ('airplane', 0.18625374), ('truck', 0.17452993), ('ship', 0.1309104), ('cat', 0.10217558)]]
-----
Class:deer & ImageId:[1836, 1304]
1/1 ----- 0s 20ms/step
1/1 ----- 0s 33ms/step
Sample-1 & ImageId:[1836 & Class&Proba[('deer', 0.8765892), ('cat', 0.043499198), ('frog', 0.039558934), ('bird', 0.027838368), ('dog', 0.011636132)]
Sample-2 & ImageId:[1304 & Class&Proba[('ship', 0.5075991), ('frog', 0.1953047), ('airplane', 0.00529482), ('deer', 0.0090756), ('cat', 0.006040841)]
-----
Class:dog & ImageId:[8766, 2384]
1/1 ----- 0s 30ms/step
1/1 ----- 0s 33ms/step
Sample-1 & ImageId:[8766 & Class&Proba[('dog', 0.7412005), ('cat', 0.14787173), ('bird', 0.0442045), ('horse', 0.026501982), ('frog', 0.02389922)]
Sample-2 & ImageId:[2384 & Class&Proba[('deer', 0.4721211), ('cat', 0.2636016), ('dog', 0.11154625), ('bird', 0.00597031), ('horse', 0.054080616)]]
-----
Class:frog & ImageId:[663, 3880]
1/1 ----- 0s 27ms/step
1/1 ----- 0s 28ms/step
Sample-1 & ImageId:[663 & Class&Proba[('frog', 0.495008), ('bird', 0.478341), ('deer', 0.018040456), ('cat', 0.004440301), ('dog', 0.0026252791)]
Sample-2 & ImageId:[3880 & Class&Proba[('frog', 0.8001505), ('bird', 0.04718431), ('cat', 0.01287245), ('deer', 0.02686713), ('dog', 0.0097078895)]]
-----
Class:horse & ImageId:[2682, 5371]
1/1 ----- 0s 37ms/step
1/1 ----- 0s 26ms/step
Sample-1 & ImageId:[2682 & Class&Proba[('deer', 0.77672344), ('horse', 0.19041206), ('bird', 0.01299853), ('cat', 0.009848032), ('dog', 0.00844199)]
Sample-2 & ImageId:[5371 & Class&Proba[('horse', 0.9661773), ('dog', 0.0197823), ('bird', 0.0065346393), ('deer', 0.0036469631), ('cat', 0.0030883218)]]
-----
Class:ship & ImageId:[6829, 3562]
1/1 ----- 0s 25ms/step
1/1 ----- 0s 24ms/step
Sample-1 & ImageId:[6829 & Class&Proba[('ship', 0.78663695), ('frog', 0.15656415), ('airplane', 0.03779006), ('cat', 0.009795836), ('bird', 0.0040842648)]
Sample-2 & ImageId:[3562 & Class&Proba[('ship', 0.95280605), ('airplane', 0.0032743855), ('automobile', 0.002607033), ('truck', 0.0007526726), ('bird', 0.00013958303)]]
-----
Class:truck & ImageId:[3557, 5180]
1/1 ----- 0s 26ms/step
1/1 ----- 0s 26ms/step
Sample-1 & ImageId:[3557 & Class&Proba[('frog', 0.89444226), ('bird', 0.098408304), ('deer', 0.0034423508), ('cat', 0.0022284822), ('dog', 0.0000027745)]
Sample-2 & ImageId:[5180 & Class&Proba[('truck', 0.91957474), ('automobile', 0.040518088), ('airplane', 0.031086257), ('ship', 0.0036967755), ('cat', 0.0014496654)]]

```

- ii. Denediğiniz bütün test resimleri için, resimlerin sınıflandırma başarısına göre karışıklık matrisini veriniz.



4.1. 3 Konvolüsyon Katmanlı ve 3x3 Filtre Boyutlu Mimari

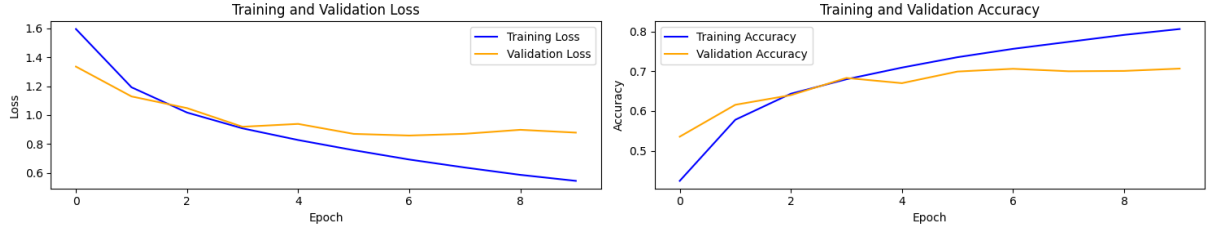


Fig. 1 : Model Performs for 3 Convolutional Layer and 3x3 Filter

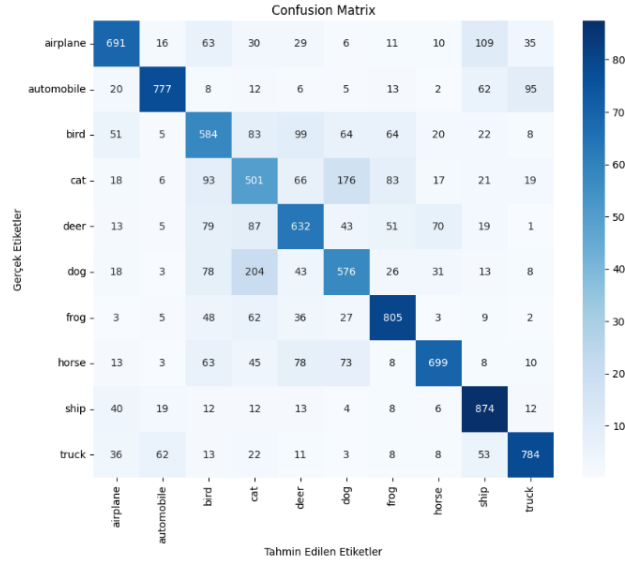


Fig. 2: Confusion Matrix for 3 Convolutional Layer and 3x3 Filter

4.2. 3 Konvolüsyon Katmanlı ve 5x5 Filtre Boyutlu Mimari

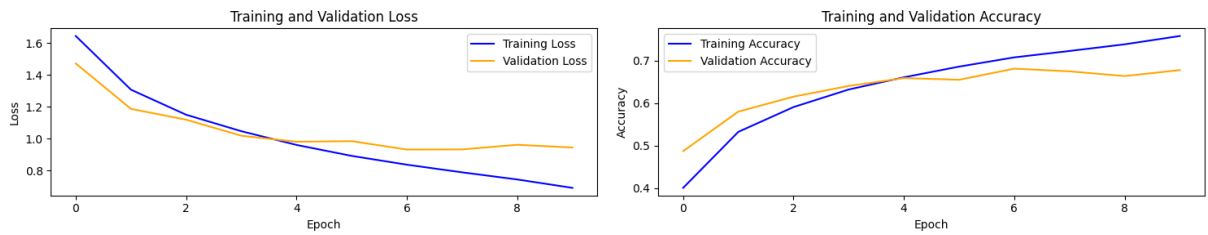


Fig. 3 : Model Performs for 3 Convolutional Layer and 5x5 Filter



Fig. 4 : Confusion Matrix for 3 Convolutional Layer and 5x5 Filter

4.3. 5 Konvolüsyon Katmanlı ve 3x3 Filtre Boyutlu Mimari

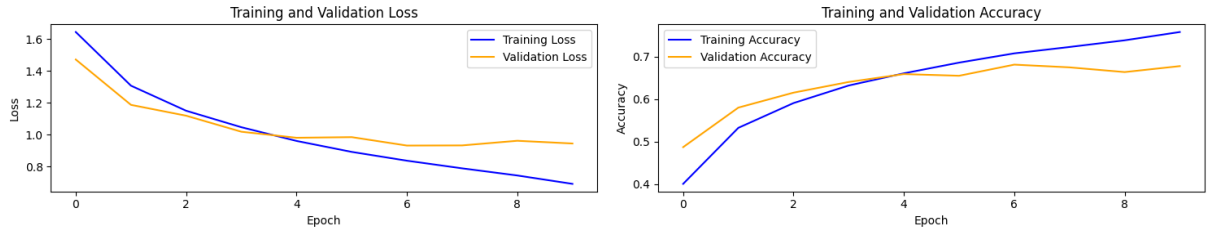


Fig. 5 : Model Performs for 5 Convolutional Layer and 5x5 Filter

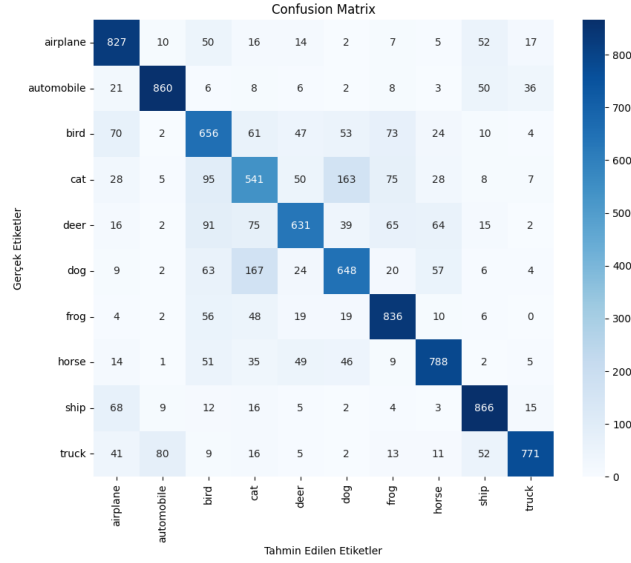


Fig. 6 : Confusion Matrix for 5 Convolutional Layer and 5x5 Filter

4.4. 5 Konvolüsyon Katmanlı ve 5x5 Filtre Boyutlu Mimari

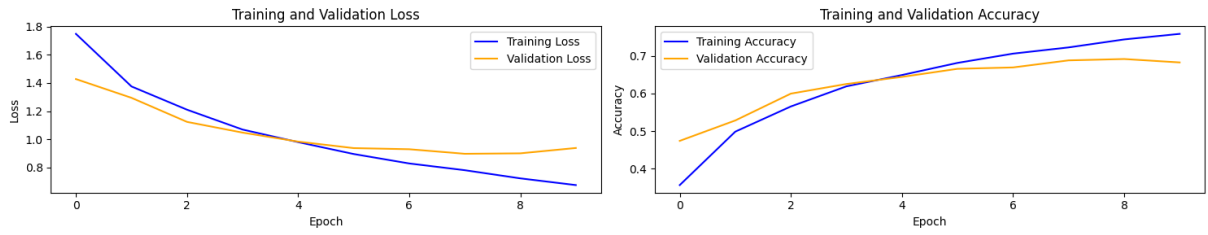


Fig. 7 : Model Performs for 5 Convolutional Layer and 5x5 Filter

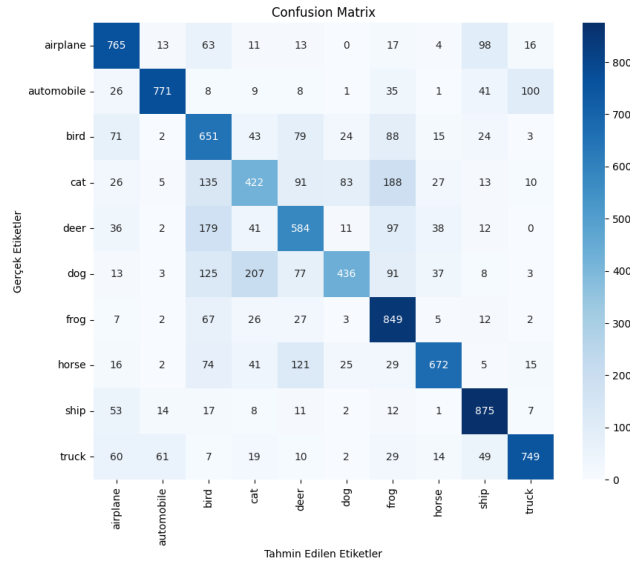


Fig. 8 : Confusion Matrix for 5 Convolutional Layer and 5x5 Filter

5. Sonuç

Konvolüsyonel Sinir Ağları (CNN) kullanarak resim nesneleri üzerinde çoklu sınıflandırma (multi class classification) çalışmalarında etkili bir şekilde gerçekleştirilebilmektedir. Bu çalışmada, CIFAR-10 veri seti üzerinde 3 ve 5 katmanlı Konvolüsyon mimarileri 3x3 ve 5x5 filtre boyutları ile denenmiştir. Filtre boyutları, CNN modelinde öğrenmeyi ve özelliklerin işlendiği/çıkarıldığı bir adımdır. Küçük filtreler daha fazla detayı yakalamak ve daha bölgesel özelliklerin çıkarılmasını sağlamaktadır. Büyük filtreler ise daha geniş özelliklerin algılanması olarak tanır ve ama model öğrenmesi için daha fazla parametre ve hesaplama ihtiyacı duyar. Bu çalışma da 32x32 en ve boy özelliklerine sahip, CIFAR-10 veri seti üzerinde 5 katmanlı ve 3x3 filtre boyutlu bir CNN mimarisi ile test veri seti üzerinde 0,74 ile en iyi performans gösterdiği görülmüştür.