



## Çeşitli görüntülerin sınıflandırılması için yeni bir Evrişimsel Sinir Ağı önerisi A new Convolutional Neural Network proposal for classifying various images

Yeşim Tiraki<sup>1</sup> , Hasan Temurtaş<sup>2</sup> , Soydan Serttaş<sup>3</sup> , Çiğdem Bakır<sup>4,\*</sup> 

<sup>1</sup> Kütahya Dumlupınar Üniversitesi, Lisansüstü Eğitim Enstitüsü, Bilgisayar Mühendisliği Bölümü, 43000, Kütahya Türkiye

<sup>2,3</sup> Kütahya Dumlupınar Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 43000, Kütahya Türkiye

<sup>4</sup> Kütahya Dumlupınar Üniversitesi, Mühendislik Fakültesi, Yazılım Mühendisliği Bölümü, 43000, Kütahya Türkiye

### Öz

Günümüzde derin öğrenme metotları robotik, ses işleme, tıp ve görüntü gibi birçok alanda yaygın kullanılmaktadır. Bu çalışmada literatürde yapılan görüntü sınıflandırma ve analizine yönelik çalışmalar detaylı bir şekilde incelenmiştir. Ayrıca yapılan çalışmada CNN kullanılarak derin öğrenme metotlarının MNIST, Fashion MNIST, CIFAR-10 ve CIFAR-100 isimindeki 4 farklı veri seti üzerinde performans analizleri yapılmıştır. Oluşturulan derin öğrenme modellerinin yapıları, eğitim için kullanılan parametre değerleri, kullanılan katmanlar, doğrulama verileri için elde edilen karmaşıklık matrisleri, doğruluk ve kayıp grafikleri ayrıntılı olarak gösterilmiştir. Çalışmamız 3 adet konvolüsyon katmanı, 3 adet batch normalizasyon katmanı, 2 adet maxpooling katmanı, 1 flatten, 2 dropout, 2 dense katmanından oluşan farklı bir ağ yapısı ile gerçekleştirilmiştir. Ayrıca önerdiğimiz modelle görüntü sınıflandırılmasının farklı veriler üzerindeki performansı artırılmıştır. Test sonunda çalışmamız çeşitli değerlendirme metriklerine göre doğruluk sonuçları karşılaştırılmıştır. Kullanılan tüm veri setleri için en iyi ve en kötü bulunan görüntüler tespit edilmiştir. Önerilen CNN modeli ile MNIST ve Fashion MNIST veri setleri için yüksek doğruluk oranları gözlemlenmiş olup bu değerler sırasıyla %99.22 ve %99.21'dir.

**Anahtar kelimeler:** Derin öğrenme, Evrişimli Sinir Ağı, CIFAR-10, CIFAR-100, MNIST, Fashion MNIST

### 1 Giriş

Görüntü işleme çeşitli görüntülerin iyileştirilmesi, görüntü kalitesini artırma, görüntü kodlama, görüntüyü dönüştürme, görüntüdeki kenarları tespit etme, bölütleme, nesne tanıma, görüntüyü dönüştürme, görüntü tanıma ve bilgisayarla görü gibi birçok bilgisayar biliminde yaygın kullanılan bir alandır [1]. Nesne takibi, trafik kontrol, plaka tanıma, ultrason, tomografi, biyomedikal uygulamalar, nesne takibi, yüz tanıma, uzaktan algılama, tarımsal, endüstriyel uygulamalar, coğrafi sistemler, astronomi, askeri ve güvenlik uygulamalarında sıklıkla kullanılır. Özellikle de görüntü işleme teknikleri kullanılarak gerçekleştirilen iris ve parmak izi tanıma güvenlik alanında oldukça önemli bir yere sahiptir.

### Abstract

Nowadays, deep learning methods are widely used in many fields such as robotics, sound processing, medicine and imagery. In this study, studies on image classification and analysis in the literature were examined in detail. In addition, performance analyzes of deep learning methods were carried out using CNN on 4 different data sets named MNIST, Fashion MNIST, CIFAR-10 and CIFAR-100. The structures of the created deep learning models, the parameter values used for training, the layers used, the complexity matrices obtained for the validation data, accuracy and loss graphs are shown in detail. Our study was carried out with a different network structure consisting of 3 convolution layers, 3 batch normalization layers, 2 maxpooling layers, 1 flatten, 2 dropout, 2 dense layers. In addition, the performance of image classification on different data has been increased with our proposed model. At the end of the test, the accuracy results of our study were compared according to various evaluation metrics. For all the data sets used, the best and worst images were determined. With the proposed CNN model, high accuracy rates were observed for MNIST and Fashion MNIST datasets, with these values being 99.22% and 99.21%, respectively.

**Keywords:** Deep learning, Convolutional Neural Networks, CIFAR-10, CIFAR-100, MNIST, Fashion MNIST

Kamera gibi farklı yollarla elde edilen görüntülerden anlamlı bilgiler elde edilerek yorumlanması için renk dönüştürme, kenar algılama, filtreleme, morfolojik işlemler gibi birtakım görüntü işleme teknikleri kullanılır [2]. Elde edilen görüntüler ham olarak değerlendirilmez. Görüntünün netleşmesi, görüntünün bulanıklığının ortadan kaldırılması ve görüntünün kalitesini artırmak için filtreleme gibi birtakım ön işleme adımları uygulanır. Ön işlemeden geçirilen görüntüler üzerinden nesnelerin doğru tespit edilebilmesi amacıyla önemli özelliklerin belirlenmesi için doğrusal ya da doğrusal olmayan özellik çıkarımı gerçekleştirilir. Bu adımdan sonra görüntüden anlamlı bilgileri çıkarmak amacıyla görüntü analizi gerçekleştirilir. Görüntü analizi ile görüntünün hesaplamalı algoritmalar kullanılarak işlenmesi sonucunda elde edilen bulgular ve

\* Sorumlu yazar / Corresponding author, e-posta / e-mail: cigdem.bakir@dpu.edu.tr (Ç. Bakır)

Geliş / Received: 05.09.2023 Kabul / Accepted: 04.12.2023 Yayınlanma / Published: 15.01.2024

doi: 10.28948/ngumuh.1355726

değerlendirmeler görselleştirilir. Böylelikle elde edilen sonuçlar kişilere daha kolay ve hızlı bir şekilde sunulur.

Hızla gelişen teknolojinin önemli konularından biri de yapay zekâ ve alt dalı olan makine öğrenmesine bağlı derin öğrenmedir. Derin öğrenme, yapay sinir ağları ve makine öğrenme algoritmalarını kapsayarak verilen veri seti ile sonuçlar tahmin edilebilmektedir [3].

Derin öğrenme, görüntü, ses, video ve biyomedikal görüntü işleme alanında sıklıkla kullanılan katmanlı mimariye sahip bir yöntemdir. Derin öğrenme yapısı girdiler, gizli katmanlar ve çıktılardan oluşur. Bir katmanın girdisinin kendisinden önce gelen katmanın çıktısı olması prensibine dayanır. Tekrarlayan Sinir Ağları (RNN), Sınırlı Boltzman Makineleri (RBM), Derin Oto-Kodlayıcıları (AE), Uzun-Kısa vadeli Hafıza Ağları (LSTM) ve Evrişimli Sinir Ağları (CNN- Convolutional Neural Networks) başlıca derin öğrenme metotlarıdır [4].

Çalışmamızda çeşitli görüntülerin sınıflandırılması ve analiz edilmesi amacıyla derin öğrenmeye dayalı bir model önerilmiştir. Önerdiğimiz modelde literatürde kullanılan modellerin tersine farklı yeni bir ağ yapısı ile daha derin katmanlar kullanılarak daha önemli özellikler çıkarılmıştır. Böylelikle daha doğru bir sınıflandırma yöntemi geliştirilmiştir. Derin öğrenme metotları görüntü sınıflandırma üzerinde başarılı olarak gerçekleştirilmiştir. Çalışmamızda önerdiğimiz CNN ağı farklı veri setleri üzerinde denenmiş ve sonuçlar karşılaştırılmalı olarak sunulmuştur.

## 2 İlgili çalışmalar

Literatürde CIFAR-10 veri setini kullanan çalışmalara baktığımızda; Abouelnaga yapmış olduğu çalışmasında [5] Lojistik Regresyon, K-En Yakın Komşular (KNN), Destek Vektör Makinesi (SVM) gibi çeşitli geleneksel makine öğrenimi algoritmaları ve bu algoritmaların Temel Bileşen Analizi (PCA) ile bir kombinasyonu tarafından elde edilen doğruluğu incelemiştir. Aynı zamanda CNN'leri incelemeyi amaçlamış ve dört CNN sınıflandırıcısı ve bir KNN sınıflandırıcısı kullanarak %94.03'lük bir doğruluk elde etmiştir. Goyal çalışmasında [6] özellikle CIFAR-10 veri seti için sıfırdan tasarlanmış, son derece popüler son teknoloji CNN'lerden büyük ölçüde ilham alan bazı CNN'leri incelemeyi ve aralarında adil bir karşılaştırma sunmayı amaçlamıştır. Çeşitli CNN'lerin artırılmamış CIFAR-10 veri kümesi üzerindeki performansını değerlendirmiş ve daha basit ağların karmaşık ağlardan daha iyi performans gösterebileceğini bulmuştur. Düzenleştirmeyi kullandıktan sonra bile daha derine inmenin her zaman işe yaramayabileceği sonucuna varılmıştır. Ho-Phuoc çalışmasında [7], CIFAR-10 veri setinde derin sinir ağları ile insanların karşılaştırmasını tartışmıştır. Bazı ağların %93.91'lik insan doğruluğunu aşabileceğine dair sonuçlar elde etmiştir. Yang Ju ve arkadaşları, Raspberry Pi mobil cihazında kolayca çalışabilen hafif evrişimli sinir ağı TripleNet ile CIFAR-10, SVHN veri setleri üzerinde görüntü sınıflandırma gerçekleştirmişlerdir [8]. Blok bağlantılarından uyarlanan TripleNet yeni önerilen ağ modelinin doğruluk ve çıkarım açısından performans sonuçları hesaplanmıştır. Ayrıca zamansal olarak

MobileNet, ThreshNet, EfcientNet ve HardNet ile karşılaştırıldığında TripleNet'in görüntü başına çıkarım süresinin sırasıyla %16, %17, %24 ve %30 oranında kısaldığını göstermişlerdir. ImageNet'in büyük veri kümesinden farklı olarak, CIFAR-10 ve SVHN gibi küçük veri kümeleri, hesaplama açısından maliyetli sinir ağları gerektirmez. Bu sebeple performans açısından önerilen model kötü sonuçlar doğurabilir.

Literatürde CIFAR-10 ve CIFAR-100 veri setlerini kullanan çalışmalara baktığımızda; Liu ve arkadaşları, CIFAR-10 ve CIFAR-100 verilerini aktivasyon fonksiyonu ve farklı ağa dayalı Inception modeli ile sınıflandırmışlardır [9]. Çok büyük katmanlarda ResNet yönteminin derin katmanları yeni aktivasyon fonksiyonuyla değiştirilmiştir. Önerdikleri modelde başlangıç bloklarında uygun ayarlamaları yaptıktan sonra, ResNet'in daha derin katmanlarını başlangıçta kullandıkları bloklarla ve Relu katmanını da farklılaştırılmış aktivasyon fonksiyonuyla değiştirmişlerdir. Parametre sayısını azaltmak için 1x1 evrişim katmanı kullanılmıştır. Bu iki yöntemle parametre sayısı azaltılarak çalışma süresinin kısaltılması sağlanmıştır. Gürültü içermeyen veriler için Relu katmanında küçük negatif değerler verilerek doğruluk artırılmıştır. Ancak gürültülü verilerde başarı oranı düşüktür. Gürültülü veriler için Gauss gürültüsü eklenebilir. Zhou ve arkadaşları [10], CIFAR-10 ve CIFAR-100 veri setlerinde görüntüyü tanımak amacıyla Discrete Wavelet Transform (DWT) ve Wavelet-Attention (WA) bloğu tasarlamışlardır. Önerdikleri modelin başarısını karşılaştırmalı olarak göstermişlerdir. Özellikle alanlarını ayırtmak için DWT'yi kullanan WA bloğu, düşük frekans alanındaki gereksiz bilgileri filtrelemek için yönlendirebilir. Ayrıca bu yöntem yüksek frekans alanındaki özelliklerin ayrıntılı bilgilerini entegre ederek eğitim aşamasında modelin öğrenme yeteneğini artırabilir. Ancak önerilen bu yöntemin büyük görüntü verilerinde aynı başarıyı gösterip göstermeyeceği belirsizdir. Zhang ve arkadaşları, görüntüdeki özellikleri belirlemek amacıyla MFR-DenseNet yöntemi önermişlerdir [11]. MFR-DenseNet, farklı evrişimsel özelliklerin karşılıklı bağımlılıkları açık bir şekilde modelleyen DenseNet'in geliştirilmiş halidir. Çalışmanın ilk aşamasında katmanlar arası özelliği ağırlıklandırmak için DenseNet oluşturmuş, sonrasında CFR-DenseNet yönteminin topluluk öğrenme yaklaşımı birleştirerek MFR-DenseNet elde edilmiştir. Oluşturulan 100 katmanlı MFR-DenseNet yöntemi CIFAR-10 ve CIFAR-100 veri setleri üzerinde eğitilmiş ve önerilen modelin başarısı ölçülmüştür.

Literatürde MNIST veri setini kullanan çalışmalara baktığımızda; Shou ve arkadaşları, kişilerin el yazısındaki farklılıklar ve el yazılarının özelliklerini karakteristik analizinin zorluğundan dolayı MNIST veri seti üzerinde derin öğrenme yöntemleriyle görüntü tanıma çalışması yapmışlardır [12]. Çok sayıda parametre değerleri kullanarak eğitim verisinde hızlı bir yakınsama oranı ile iyileştirme sağlamışlardır. Yaptıkları çalışmayla MNIST veri seti kullanarak el yazısı dijital tanıma ile sınıflandırma için DNN ağını geliştirmeyi amaçlamışlardır. Diğer bir çalışmada Plesovskaya ve Ivanov SVD ve Kernel Yoğunluk Tahmini kullanarak MNIST veri seti üzerinde görüntü sınıflandırma

yapmışlardır [13]. Bu yöntemlerin görüntü işlemdeki doğruluğunu artırmak için hiyerarşik sınıflandırma yöntemi geliştirmişlerdir. Önerdikleri yöntem birçok alanda sıklıkla kullanılan makine öğrenimi algoritması SVD'den daha başarılı sonuç vermiştir. Bu çalışmada boyut azaltma ve kernel yoğunluk tahminine dayalı yöntem kullanılmıştır. Hiyerarşik sınıflandırma yönteminin başarısı ve analizi SVM ile karşılaştırmalı olarak verilmiştir. Pei ve Ye MNIST veri setini eğitmek için kümeleme yöntemlerini kullanmışlardır [14]. Eğitim veri setindeki görüntüleri 60000×785 satırlık iki boyutlu matris verilerine dönüştürmüşlerdir. Daha sonrasında ise test verilerini K-means kümeleme yöntemiyle 10 kümeye bölerek MNIST veri setini sınıflandırmışlardır. T-SEN ile veri seti boyutunu azaltmışlardır. K-means kümeleme dışında farklı kümelerin bu veri seti üzerindeki performansını araştırmak amacıyla MiniBatchKMeans algoritması da kullanmışlardır. MiniBatchKMeans algoritması K-means kümeleme yöntemine göre daha başarılıdır.

Literatürde Fashion MNIST veri setlerini kullanan çalışmalar da mevcuttur. Çalışmalarında Dropout değeri açısından farklı CNN mimarilerini karşılaştıran Henrique vd. [15] düşük Dropout değeri içeren CNN-Dropout-3 modelinin %99 doğruluk değeri ile en başarılı model olduğu belirtmişlerdir. Birim [16] moda görsellerinin başarıyla sınıflandırılması konusunu içeren çalışmada Fashion MNIST veri setini kullanarak farklı mimarilerde CNN algoritmalarını incelemiştir. Bu CNN mimarileri, temel CNN, Dropout düzenleyici içeren, L2 düzenleyici içeren ve son olarak her iki düzenleyiciyi de içeren mimarilerdir. Çalışma sonucunda Dropout içeren CNN algoritmasının %94.3 doğruluk oranı ile en iyi performansı verdiğini belirtmiştir. Farklı CNN mimarileri inceleyen bir diğer çalışmada Bhatnagar vd. [17], ürün sınıflandırması yapmıştır. Yine üç farklı CNN mimarisi incelenmiştir. Bu mimariler temel CNN mimarisi, batch normalizasyon mimarisi ve hem batch normalizasyon hem de residual skip connections içeren mimaridir. Sonuçlara göre en iyi performansı doğruluk değeri %92.54 olan batch normalizasyon ve residual skip connections birlikte birlikte kullanıldığı mimarinin verdiği belirtilmiştir. Meshkini vd. [18], CNN modellerinden VGG, GoogleNet, DenseNet, AlexNet, SqueezeNet ve ResNet algoritmalarının sınıflandırma başarılarını incelemiştir. Batch normalizasyon kullanan SqueezeNet algoritmasının %93.43 ile en başarılı doğruluk oranını verdiği belirtilmiştir. Greeshma çalışmasında [19] veri büyütme (Data Augmentation) kullanarak Dropout düzenleyicinin ve hiperparametre optimizasyonunun CNN sınıflandırma başarısındaki rolünü incelemiştir. Çalışmada bu modelin %93.99 doğruluk oranı gösterdiği bulunmuştur. Kayed ve arkadaşları [20] CNN tabanlı LeNet-5 yapısını incelemiş ve %98 doğruluk değeri ile sınıflandırma yaptığı belirtmiştir. Yine Hur benzer çalışmasında [21] Kuantum CNN yapısını incelemiş ve %93.6 doğruluk değeri ile sınıflandırma ortaya çıkarmıştır. Seo ve Shin giysi sınıflandırması için hiyerarşik yapıyı kullanan çalışmalarında [22] VGGNet CNN mimarisinin hiyerarşik olmayan modele göre daha doğru sonuç verdiğini ve kaybın küçüldüğünü belirtmişlerdir. Çalışmada %93.52

doğruluk değeri bulunmuştur. Khanday ve arkadaşları [23] ise CIFAR-10 ve Fashion MNIST veri setlerini kullanarak filtre boyutu açısından CNN mimarilerini incelemiştir. Mimari başarısının filtre boyutu ile ters orantılı olduğu sonucuna varılmıştır. Her iki veri setinde de 3x3 filtrenin en iyi başarıyı gösterdiği belirtilmiş ve Fashion-MNIST için %93.68 değerinin en iyi doğruluk değeri olduğu söylenmiştir. Nocentini ve arkadaşları Fashion MNIST veri seti üzerinde 15 evrişimli katman içeren bir ağ kullanarak görüntü sınıflandırması gerçekleştirmişlerdir [24]. Bu veri setinde gerçekleştirdikleri doğruluk sonuçlarını çalışmalarında kullandıkları iki farklı moda-ürün veri seti ile birlikte karşılaştırmışlardır. Bu iki farklı veri setinde %60 ve %40 oranında doğruluk elde etmişlerdir. Ayrıca, farklı fully connected layers, batch size ve stride kullanarak önerdikleri modelin performansını değerlendirmişlerdir. Derin öğrenme algoritmaları görüntü işlemede kullanılan yaygın olarak kullanılmaktadır. Vijayaraj ve arkadaşları, farklı model görüntülerinin stillerini tahmin edebilmek amacıyla Fashion MNIST CNN ağlarıyla verilerini sınıflandırmışlardır [25]. Farklı aktivasyon fonksiyonları, öğrenme oranları ve parametreler kullanarak önerdikleri modeli test etmişlerdir. Giysi tanıma ve sınıflandırma amacıyla farklı veri setleri kullanmışlardır.

Literatürde CIFAR-10, CIFAR-100 ve MNIST veri setlerini birlikte kullanan çalışmalar da mevcuttur. Da Silva ve arkadaşları çalışmalarında [26] düşük karmaşıklıkta evrişimli sinir ağlarını geliştirmek ve optimize etmek için çok amaçlı bir gramer evrimi çerçevesi önermişlerdir. Önerilen model beş görüntü veri setinde değerlendirilmiştir. Bunlar CIFAR-10, CIFAR-100, MNIST, KMNIST ve EuroSAT'tır. Önerilen model CIFAR-10 ve EuroSAT veri setlerinde görüntü sıralamasında birinciliği elde etmiştir. Ayrıca sonuçlar, oluşturulan modellerin CIFAR-100, MNIST ve KMNIST veri kümeleri için en iyi CNN'lere istatistiksel olarak benzer sonuçlar elde ettiğini ve son teknoloji ağlara kıyasla ortalama %95 daha az eğitilebilir parametreye sahip olduğunu göstermiştir. Rashno çalışmasında [27] görüntü sınıflandırması için Neutrosophic Convolutional Neural Networks (NCNN) olarak adlandırılan, nötrozofik (NS) alanda veri belirsizliği işleyen yeni bir evrişimli sinir ağı yapısı önermiştir. Bu yöntem, 20 seviyeli Gauss gürültüsü ile kirlenmiş MNIST, CIFAR-10 ve CIFAR-100 veri setleri üzerinde değerlendirilmiştir. Sonuçlar, iki yollu NCNN'nin, MNIST ve CIFAR-10 veri setlerinde farklı gürültü seviyelerine sahip 5 çiftte (eğitim, test) sırasıyla %5.11 ve %2.21 oranında CNN'den daha iyi performans elde ettiğini göstermektedir. Son olarak, NVGG-Net, CIFAR-10 ve CIFAR-100 veri setlerinde doğruluğu VGG-Net'e kıyasla sırasıyla %3.09 ve %2.57 oranında artırmıştır. Güler'in çalışmasında [28] nesne sınıflandırması için CNN ile bir kapsül ağ tekniği kullanılarak hibrit bir model tasarlanmıştır. 14 farklı el hareketini içeren HG14 isimli bir veri seti kullanılmıştır. Modelin nesne tanıma performansına bakmak için Fashion MNIST, HG14 ve CIFAR-10 veri setleri üzerinde eğitim gerçekleştirilmiştir. Ayrıca HG14, Fashion MNIST ve CIFAR-10 veri kümelerindeki görüntüleri sınıflandırmak için VGG16, ResNet50, DenseNet ve CapsNet modelleri kullanılmıştır.



Eğitim sonuçları karşılaştırılmış ve önerilen model, HG14, Fashion MNIST ve CIFAR-10 verisetlerinde sırasıyla %90, %93.88 ve %81.42 değerleri ile en yüksek doğruluğa ulaşmıştır. Ryu çalışmasında [29] derin sinir ağlarında temiz görüntülere küçük bozulmalar ekleyerek düşmanca örnekler oluşturan düşmanca saldırılara (adversarial attacks) karşı gürültü giderici bir ağ önermiştir. MNIST, CIFAR-10, CIFAR-ve 100 veri kümeleri üzerinde yürütülen deneylerin sonuçları yüksek bir sınıflandırma doğruluğu ile sonuçlandığını göstermektedir. Önerilen yöntemin eğitimin son teknoloji rakip eğitim yöntemiyle karşılaştırıldığında, düşman saldırılarına karşı sağlanmakta ortalama %0.84, %27.33, %28.99 ve %17.61'lik iyileştirmeler sağladığını belirtmiştir. Siebert çalışmasında [30] bir çeşit evrişimli sinir ağı olan DenseNet için yapılmış yeni bir optimizasyon yöntemi önermiştir. Yöntemin amacı, ağın içinde bulunduğu ana göre karakteristik haritaların üretimini kontrol etmek ve doğrulukta minimum kayıpla ağın boyutunu azaltmaktır. Önerilen modelin davranışını doğrulamak için MNIST, Fashion MNIST, CIFAR-10, CIFAR-100, CALTECH-101, Cats vs Dogs ve TinyImageNet veri setleri ile deneyler yapılmıştır. MNIST ve Fashion MNIST veri seti için %43 parametre azalması ve CIFAR-10 ağ parametrelerinde %44 azalma elde edilirken, CIFAR-100 bazında parametre düşüşü %43 olmuştur. Assunção çalışmasındaki [31] ağ önerisini CIFAR-10 veri kümesi ile test etmiş ve %95.22 doğruluğa ulaşmıştır. Ayrıca, bu ağı MNIST, Fashion-MNIST ve CIFAR-100'ü sınıflandırmak için de uygulamış ve en etkileyici sonucun CIFAR-100 veri kümesindeki %78.75'lik sınıflandırma doğruluğu olduğunu belirtmiştir. Hafız ve arkadaşları, ImageNet, Natural Images, CIFAR-10, CIFAR-100, MNIST ve Fashion-MNIST veri setlerini ResNet-50, DenseNet, WRN-28-10 ve NSGANetV1 ile adlandırılan ensemble yöntemlerle eğitmişlerdir [32]. Önerilen ensemble yöntemler karar ağaçlarından esinlenerek gerçekleştirilmiştir. Ensemble yöntemler büyük ve karmaşık problemlerde ortaya çıkan düşük performans sorununu çözmek amacıyla kullanılmıştır. Yapılan deneysel çalışmalarla kullanılan ensemble yöntemlerin başarısı ve performansı karşılaştırılmıştır.

### 3 Materyal ve metod

Derin öğrenme, görüntü ile videodan nesne tanıma, nesne takibi ile nesne algılama, genomik, konuşma tanıma tavsiye sistemleri gibi birçok alanda kullanılmaktadır [33]. Makine öğrenimi yöntemleri çok fazla zaman ve emek harcamaya sebep olan ön işlem ve özellik çıkarımı metodlarıyla uğraşırken; derin öğrenme yöntemleri veriyi ham bir şekilde kullanır. Ön işleme ve özellik çıkarımı ağın içerisinde otomatik olarak gerçekleştirilir. Genellikle görüntü analizinde kullanılan derin öğrenme yöntemleri farklı filtreler ile görüntünün özneliklerini belirler. Derin öğrenme metodlarında bir görüntünün tespit edilmesi için gizli katmanlar kullanılır.

Derin öğrenme, çok katmanlı sinirsel ağlardan oluşan makine öğreniminin bir alt dalıdır. Bir veya birden fazla gizli katman bulunduran makine öğrenimi yöntemidir [34]. Girilen veri setlerinin gizli katmanlarda işlenerek hangi çıkış katmanına çıkacağını tahmin etmesi mantığı ile çalışır. Her

bitişik katman, kendinden önceki çıktıyı girdi olarak alır. Çıktı olarak olasılığı en yüksek olan çıkış seçilir. Derin öğrenme yöntemleri gözetimli ve gözetimsiz olarak iki şekilde gerçekleştirilir. Gözetimli; girilen veriler ve çıkış verisi makineye tanımlanır ve sonuç buna göre beklenir. Hatalı bir sonuç çıkarsa tekrar algoritma ya da hesaplamalara göz atıp hata düzeltilebilir. Gözetimsiz; sonucun ne olduğunu makine kendisi karar verir. Girilen karmaşık verileri makine kendisi çözmeye çalışarak olasılıklar dahilinde çözdüğü verileri bir çıkış verisi haline getirir. Üç derin öğrenme modeli vardır [35]:

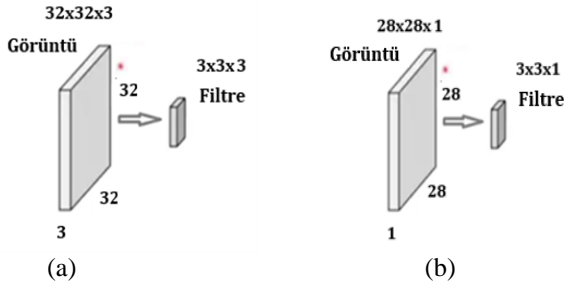
- Çok katmanlı Perceptron (Multilayer Perceptrons-MLP)
- Evrişimsel Sinir Ağı (Convolutional Neural Networks-CNN)
- Tekrarlamalı Sinir Ağı (Recurrent Neural Networks-RNN)

MLP, delta öğrenme metodu kullanan çok katmanlı bir derin öğrenme metodudur [36]. Tek nöron yerine çok sayıda nöron kullanılmaktadır ve bu nöronlar katmanlar şeklinde ifade edilir. Giriş, gizli katmanlar ve çıktı katmanından oluşur. En az bir gizli katman mevcuttur ve gizli katmanlar problemin karmaşıklığına göre artırılabilir. Her katmanın çıktısı kendisinden önceki katmanların girdileri işlenerek belirlenir. MLP özellikle genelleme ve sınıflandırma problemlerinde etkin bir şekilde çalışır.

CNN yüksek doğruluk oranı ile görüntüleri algılayabilen, sınıflandırabilen ve yeniden yapılandırabilen derin öğrenmeli bir evrişimsel sinir ağıdır. Bu modelin amacı öğrenme işlemlerinin gerçekleştirilmesi ve öğrenilen yapıların sistem tarafından doğru bir şekilde yorumlanmasını amaçlar. Farklı görselleri işleyen ve bu görsellere sınıflandırma işlemleri yapan birçok katmandan oluşan algoritmadır. Görüntü ve video üzerinde gösterdiği başarı performans sebebiyle yaygın bir şekilde tercih edilmektedir.

RNN sisteme yeni verilen giriş bilgilerinin yanı sıra daha önceden verilen ve sonradan verilecek giriş bilgilerini de kullanan, zamana dayalı bir derin öğrenme metodudur [37]. Eski verilerle yeni verileri karşılaştırarak bir tahmin üretir ve doğal dil işleme gibi alanlarda sıklıkla kullanılır. Her zaman aralığında yapılan işlemler tespit edilerek ağın yapısı oluşturulur. RNN ağ yapısı gizli durumları, verinin şimdiki zaman adımına kadarki tüm geçmiş bilgilerini içerir. Ancak her zaman artırımında kullanılan parametre sayıları değişmez.

Evrişimsel sinir ağları çalışma prensibi gereği girdi olarak matris formatında görüntü ya da videolar almaktadır. Bu çalışmada Şekil 1-a'da CIFAR-10 ve CIFAR-100 veri setleri için 32x32 ve 3x3'lük matrisler kullanılmıştır. MNIST ve Fashion MNIST veri setleri için ise Şekil 1-b'de gösterilen 28x28 ve 3x3'lük matrisler kullanılmıştır. Böylelikle matrise uyguladığımız filtre ile görüntü üzerinden belli özellikler elde edilmiştir.



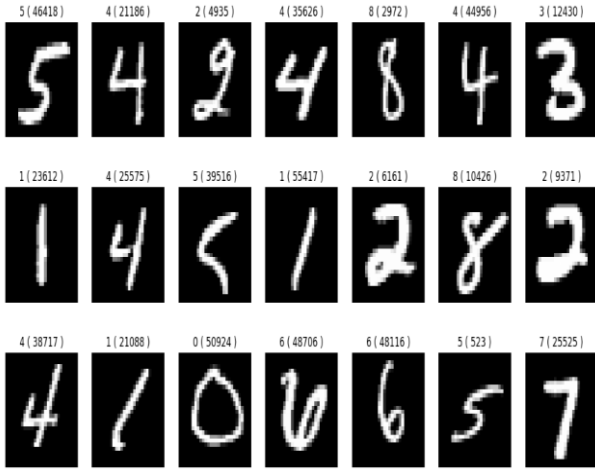
Şekil 1. a) CIFAR-10, CIFAR-100 veri setleri için matrise uygulanan filtre. b) MNIST ve MNIST Fashion veri setleri için matrise uygulanan filtre.

### 3.1 Veri setleri

Çalışmada kullanılan veri setlerinin özellikleri aşağıda detaylı olarak verilmiştir.

#### 3.1.1 MNIST veri seti

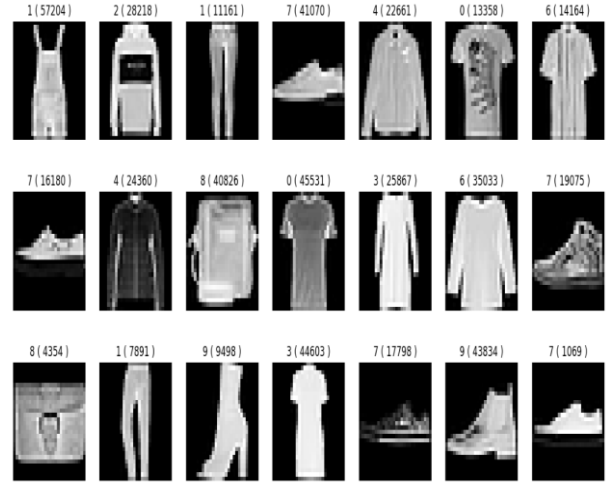
Derin öğrenmede temel veri setlerinden biri olarak kabul edilen bu veri seti el yazısı rakamlarından oluşmaktadır [38]. Başlangıç seviyesindeki birçok uygulama bu veri setini kullanmaktadır. Veri seti 60000 eğitim ve 10000 test görüntüsü olmak üzere toplamda 70000 görüntü içermektedir. Bu veri setindeki rakamlar  $28 \times 28$  çözünürlükte siyah beyaz görüntüler olarak kaydedilmiştir. Veri setindeki her bir görüntü Şekil 2’te gösterildiği gibi sayısal değer karşılığıyla etiketlenmiştir.



Şekil 2. MNIST veri seti.

#### 3.1.2 Fashion veri seti

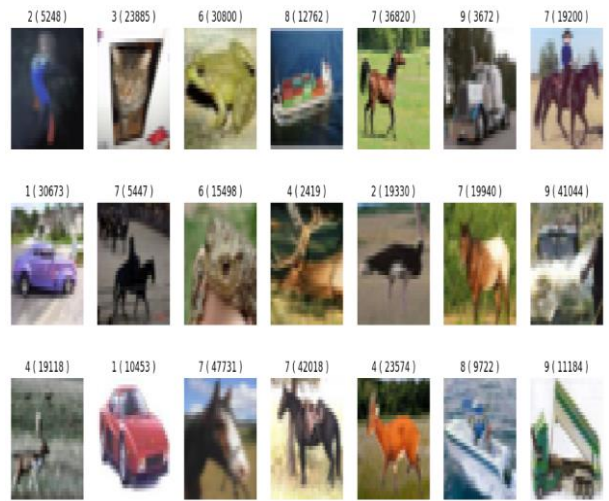
Fashion MNIST veri seti, yine çeşitli makine öğrenimi sistemlerini eğitmek ve test etmek için yaygın ve ücretsiz olarak kullanılan büyük bir moda görüntüleri veritabanıdır. Bu veri seti de 10 kategoriden moda ürünlerinin  $28 \times 28$  çözünürlükte gri tonlamalı görüntülerini içermektedir. Kategori başına 7,000 olmak üzere toplamda 70000 görüntü içermektedir [39]. Eğitim seti 60000 görüntüden, test seti ise 10000 görüntüden oluşmaktadır. Şekil 3’te Fashion MNIST veri seti görülmektedir.



Şekil 3. Fashion MNIST veri seti.

#### 3.1.3 CIFAR-10 veri seti

Yine temel veri setlerinden biri olan CIFAR-10 veri setinde [40] toplam 10 farklı sınıf bulunmaktadır. Bu sınıfların her biri  $32 \times 32$  çözünürlükte ve renkli 6000 adet görüntüden oluşmakta ve veri setinde toplamda 60,000 görüntü bulunmaktadır. Bu görüntülerin 50000 tanesi eğitim ve 10000 tanesi de test görüntülerinden oluşmaktadır. Şekil 4’te her bir sınıfta bulunan rastgele veri kümeleri gösterilmiştir.



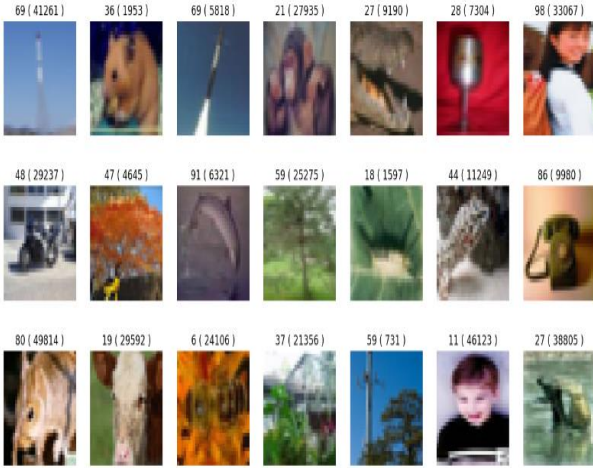
Şekil 4. CIFAR-10 veri seti.

#### 3.1.4 CIFAR-100 veri seti

CIFAR-10 veri setinin geliştirilmiş hali olan CIFAR-100 veri setinde [40] 100 sınıf mevcuttur (Şekil 5). Her sınıf 600 görüntü içermekte olup sınıf başına 500 eğitim görüntüsü ve 100 test görüntüsü bulunmaktadır. Bu veri setindeki 100 sınıf 20 üst sınıfa ayrılmıştır. Her görüntü ait olduğu sınıf ve ait olduğu üst sınıf etiketi ile birlikte gelir. Veri setindeki alt ve üst sınıfların isimleri Tablo 1’de verilmiştir.

**Tablo 1.** CIFAR-100 veri setindeki alt ve üst sınıflar.

Üst Sınıf	Alt Sınıf
suda yaşayan memeliler	kunduz, yunus, su samuru, fok, balina
balık	akvaryum balığı, yassı balık, ray, köpekbalığı, alabalık
çiçekler	orkideler, gelincikler, güller, ayçiçekleri, laleler
gıda kapları	şişeler, kaseler, kutular, bardaklar, tabaklar
meyve ve sebzeler	elma, mantar, portakal, armut, tatlı biber
elektrikli ev aletleri	saat, bilgisayar klavyesi, lamba, telefon, televizyon
ev eşyası	yatak, sandalye, kanepeler, masa, gardırop
böcekler	arı, böcek, kelebek, tırtıl, hamamböceği
büyük etoburlar	ayı, leopar, aslan, kaplan, kurt
insan yapımı büyük dış mekan yapıları	köprü, kale, ev, yol, gökdelen
büyük doğal dış mekan sahneleri	bulut, orman, dağ, ova, deniz
büyük omnivorlar ve otçullar	deve, sığır, şempanze, fil, kanguru
orta boy memeliler	tilki, kirpi, opossum, rakun, kokarca
böcek olmayan omurgasızlar	yengeç, istakoz, salyangoz, örümcek, solucan
insanlar	bebek, oğlan, kız, erkek, kadın
sürüngenler	timsah, dinazor, kertenkele, yılan, kaplumbağa
Küçük memeliler	hamster, fare, tavşan, kır faresi, sincap
ağaçlar	akçaağaç, meşe, palmye, çam, söğüt
araçlar 1	bisiklet, otobüs, motosiklet, kamyonet, tren
araçlar 2	çim biçme makinesi, roket, tramvay, tank, traktör



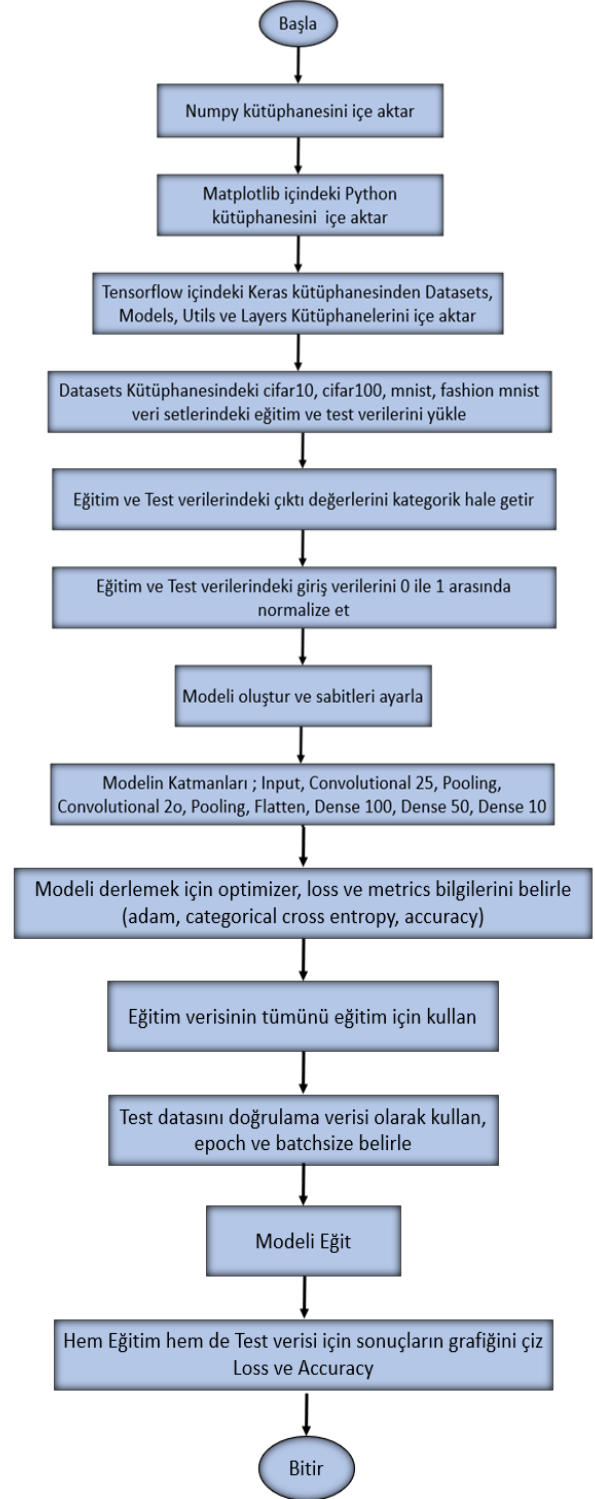
**Şekil 5.** CIFAR-100 veri seti

### 3.2 Önerilen model

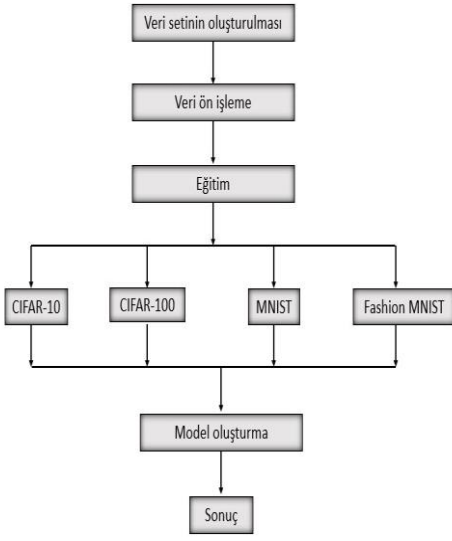
Önerdiğimiz modelin işlem basamakları Şekil 6'da verilmiştir. Kullanılan dört veri setinin eğitim ve test verileri kategorik hale getirilmiş ve normalize edilmiştir. Önerdiğimiz ağ yapısına göre oluşturulan modelin başlangıç değerleri ile modelin katmanları ve parametreleri belirlenmiştir. Farklı iterasyon ve farklı epoch değerleri ile oluşturulan model eğitilmiş ve test verilerinin sınıflandırma sonuçları hesaplanmıştır.

Önerilen modelin iş akış şeması Şekil 7'de gösterilmiştir. Sinir ağı modellerinde her ne kadar çok katmanlı bir yapı kullanılabilse de bazı durumlarda doğruluk ve performans

kayı gözlemlenebilmektedir. Bu sorunları çözmek için çalışmamızda CNN ağları önerilmiştir. CNN ağları yapay sinir ağlarından farklı olarak evrişim ve havuzlama katmanları içermektedir. Böylelikle fazla katman eklenerek modelin performansı artırılabilir.



**Şekil 6.** Önerilen modelin işlem basamakları

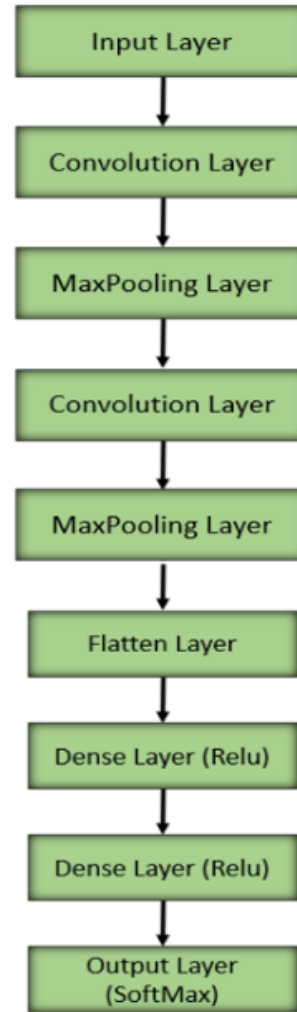


Şekil 7. İş akış şeması

Şekil 8’de CNN katmanları gösterilmektedir.

- **Giriş Katmanı (Input Layer):** CNN’de giriş katmanı görüntü, ses ve sinyal verilerini gösterir. Giriş katmanında görüntü verisi 3 boyutlu matris ile ifade edilir. Girdi katmanının aldığı veri tipine göre, sinir ağında yapılacak işlemler de değişir. Sinir ağının yapısında bulunan katmanlar arası nöronların bağlantılarının ağırlıkları hesaplanılarak çıktı değeri bulunur. Her bir nöronun girdi değeri ağırlıkları ile çarpılır.
- **Evrişim Katmanı (Convolution Layer):** CNN’in temel katmanı olarak adlandırılabilir. Bu katmanda çeşitli filtreler kullanılarak verilen giriş görüntüsü üzerinden özellik çıkarımı gerçekleştirilir. Bu özellik çıkarımı matrisler biçiminde gösterilir ve görüntüye ait bilgiler matristeki değerlerle ifade edilir. Bu katmanda özelliklerin belirlenmesi için orijinal görüntü üzerinden bir filtre dolaşır. Bu katman görüntünün özelliklerini belirlemekten sorumludur. Evrişim katmanında orijinal görüntü üzerinde küçük bir filtre dolaşır ve görüntüdeki önemli özellikleri belirlemeye çalışır.
- **Maksimum Havuzlama Katmanı (MaxPooling Layer):** Havuzlama katmanının amacı boyutu indirmektir. Böylelikle işlem gücü azalır ve görüntüdeki daha önemli özellikler belirlenmiş olunur. CNN modellerinde çoğunlukla Max (Maksimum) ve Average (Ortalama) pooling olmak üzere iki tür havuzlama katmanı bulunmaktadır. Konvolüsyon yerine havuzlama tekniği uygulanır. Eğer maksimum havuzlama uygulanıyorsa filtrenin en büyük değerini; ortalama havuzlama uygulanırsa filtredeki değerlerin ortalamasını alır.
- **Tam Bağlantılı Katman (Fully Connected Layer):** Bu katmanda havuzlama ve evrişimli katmanda oluşan görseller düz bir vektör halinde gösterilir. Bu katmanın kullanımından önce düzleştirme yapılır. Böylelikle bu katmanın girdileri oluşur.

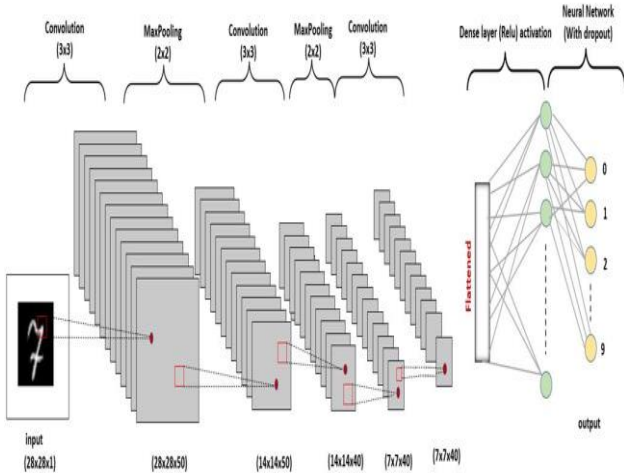
- **Düzleştirici Katman (Flatten Layer):** En son katman için (Fully Connected Layer) verileri hazırlar. Evrişim ve Pooling katmanından gelen değerlerin tek boyutlu halini gösterir. Bundan önceki katmanlarda gerçekleştirilen işlemler matrisler şeklinde gerçekleştirilirken, bu katmanda ise işlemler tek boyutlu yapıda gerçekleştirilir.
- **Yoğun Katman (Dense Layer):** Relu, Sigmoid ve Tanh gibi aktivasyon fonksiyonlarıyla çıkışı belirler. Girdinin her bir düğümü çıkıştaki düğüme bağlıdır. Her bir katmanın girişi kendisinden önceki katmanın çıkışı ile belirlenir.
- **ReLU (Rectified Linear Unit):** Relu, doğrusal olmayan  $f(x) = \max(0, x)$  şeklinde bir fonksiyondur. Örneğin -50 değerini alan bir ReLU fonksiyonu çıktı olarak 0’ı, 50 değerini alan bir fonksiyon ise 50’i verir. Temel fikir negatif değerlerden kurtulmak olan ReLU CNN’lerde oldukça önemli bir aktivasyon fonksiyonudur.
- **Çıkış Katmanı (Output Layer):** CNN sonucu elde edilen çıkış görüntüsünü gösterir.



Şekil 8. CNN katmanları

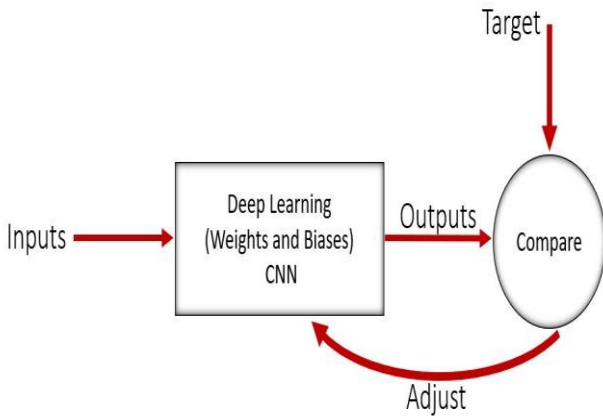


Şekil 9’da önerdiğimiz modelin ağ yapısı gösterilmektedir. 3x3’lük bir filtre giriş görüntüsü üzerinde gezdirilerek öznelikler tespit edilmeye çalışılır. Ağırlık ve bias değerleri sürekli güncellenerek öğrenme süreci oluşturulur. Çok sayıda evrişim katmanı arasında havuzlama katmanı bulunur. Bu katmanın amacı görüntünün boyutunu küçültmektir. 2x2’lik maksimum havuzlama katmanı ile görüntü boyutu dörtte bir indirgenir. Sonuç olarak görüntü sınıflandırılır.



Şekil 9. CNN ağı.

CNN ağlarındaki öğrenme süreci Şekil 10’da görüldüğü üzere ağırlık ve bias değerlerinin sürekli güncellenmesiyle gerçekleşir. Nöronlar verilen girdi değerlerini güncellenen ağırlıklar ile çarpıp bias değeriyle toplayarak üretilen sonucu aktivasyon fonksiyonundan geçirip bir sonraki nörona iletilerek çıktı görüntüsü elde edilir. Bulunan çıktı görüntüsü eğitim setindeki görüntüyle karşılaştırılarak test verisinin sınıfı tespit edilir.



Şekil 20. Genel ağırlıkların değişiminin gösterilmesi.

#### 4 Bulgular ve tartışma

Tablo 2’de gösterilen Karışıklık Matrisine göre True Positive ve True Negative sınıfın doğru olarak bulunduğuunu, False Positive ve False Negative ise sınıfın yanlış bulunduğuunu gösterir. Karışıklık Matrisi verilen gerçek sınıflar ile sınıflandırma işlemi sonucu doğru ya da yanlış olarak tahmin edilen sınıfları gösterir.

Tablo 2. Karışıklık matrisi.

		Tahmin edilen durumlar	
		Pozitif Durumlar	Negatif Durumlar
Gerçek durumlar	Pozitif	TP	FP
	Negatif	FN	TN

- **Doğruluk:** Tüm doğru tahminlerin tüm tahminlere oranı ile bulunur.

$$\text{Doğruluk} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

- **Kesinlik:** Positive doğru tahmin edilen bir durumdaki değerlerin başarısını gösterir.

$$\text{Kesinlik} = \frac{TP}{TP + FP} \quad (2)$$

- **Duyarlılık:** Positive durumların ne kadar başarılı tahmin edildiğini gösterir.

$$\text{Duyarlılık} = \frac{TP}{TP + FN} \quad (3)$$

- **F<sub>1</sub> Score:** Doğruluk gibi değerlendirme ölçütü olarak kullanılabilir. Kesinlik ve duyarlılık değerlerinin harmonik ortalaması alınarak bulunur.

$$F_1 = 2 * \frac{\text{Kesinlik} * \text{Duyarlılık}}{\text{Kesinlik} + \text{Duyarlılık}} \quad (4)$$

MNIST, Fashion MNIST, CIFAR-10 ve CIFAR-100 veri setleri için önerilen sistemin kesinlik (precision), duyarlılık (recall), doğruluk (accuracy) ve F1-skor değerlendirme kriterlerine göre sonuçları aşağıda sunulmuştur.

##### 4.1 MNIST sonuçları

MNIST veri seti için kullandığımız katmanlar ve kullanılan parametre sayıları Tablo 3’te verilmiştir. Çalışmamızda öğrenme katsayısı 0.001, batch size 100, optimizasyon için adam optimizer algoritması ve değerlendirme metriği olarak da doğruluk kriteri belirlenmiştir. Tablo 4’te eğitim verisi için %99.88 başarı elde edildiği görülürken, test verisi için de %99.22 başarı elde edildiği Tablo 5’de görülmüştür.



**Tablo 3.** Kullanılan katmanlar ve parametreler.

Layer (type)	Output shape	Param #
Conv1 (Conv2D)	(None, 28, 28, 100)	1000
Batchnorm1 (Batchnormalization)	(None, 28, 28, 100)	400
Maxpool1 (Maxpooling2D)	(None, 14, 14, 100)	0
Conv2 (Conv2D)	(None, 14, 14, 100)	90100
Batchnorm2 (Batchnormalization)	(None, 14, 14, 100)	400
Maxpool2 (Maxpooling2D)	(None, 7, 7, 100)	0
Conv3 (Conv2D)	(None, 7, 7, 100)	90100
Batchnorm3 (Batchnormalization)	(None, 7, 7, 100)	400
Flatten1 (Flatten)	(None, 4900)	0
Dense1 (Dense)	(None, 100)	490100
Dropout1 (Dropout)	(None, 100)	0
Dense2 (Dense)	(None, 100)	10100
Dropout2 (Dropout)	(None, 100)	0
Dense3 (Dense)	(None, 10)	1010
Total params: 683,610		
Trainable params: 683,010		
Non-trainable params: 600		

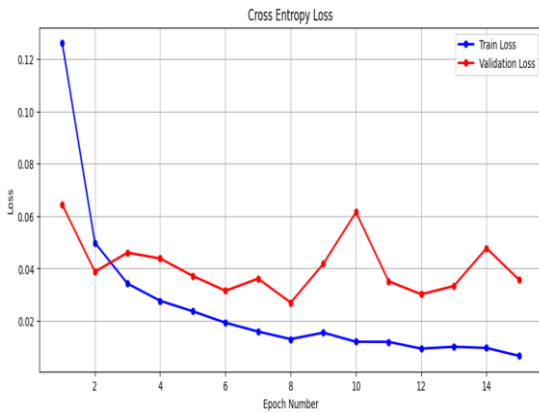
**Tablo 4.** Eğitim verisi başarı oranı.

Train Loss	Train Accuracy
0.0039570387452840805	0.9988833069801331

**Tablo 5.** Test verisi başarı oranı.

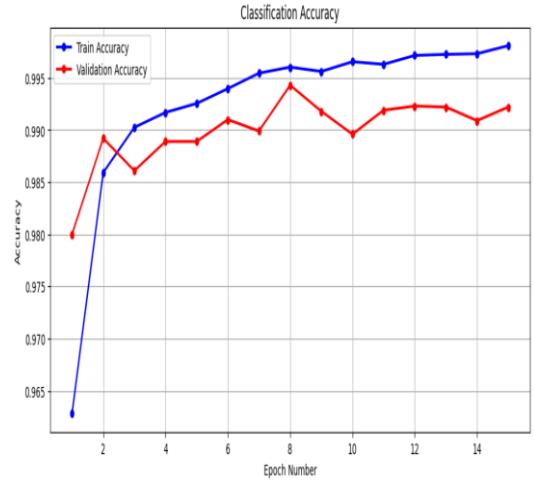
Test Loss	Test Accuracy
0.035637471824884415	0.9922000169754028

Kullandığımız veri seti için çapraz entropi (cross entropy) hata değerlerinin değişimi Şekil 11'de epoch değerlerine göre gösterilmiştir. Grafikte anlaşıldığı üzere 10. epoch da hata en yüksek ,15. epoch da hata en düşük olmaktadır.

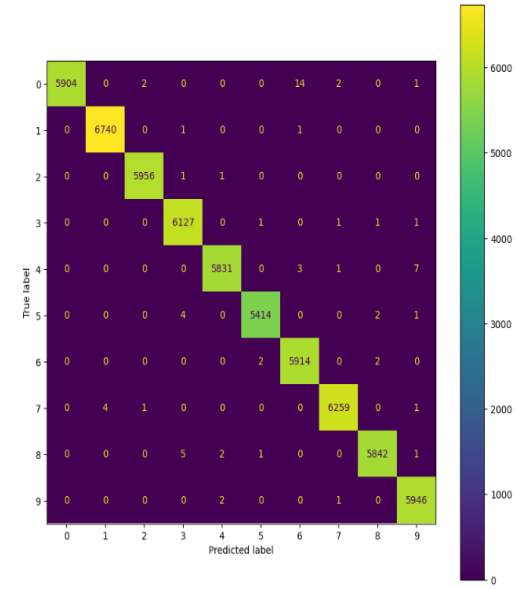


**Şekil 31.** Çapraz entropi hata grafiği.

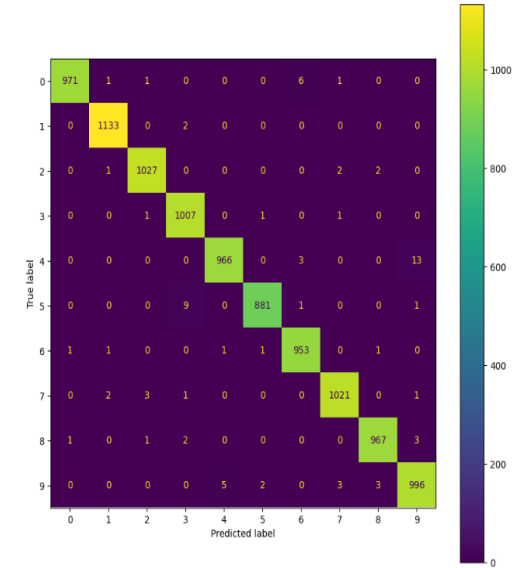
Ağırlıklar sürekli güncellenmekte olup, en iyi öğrenmenin 8. Epoch da gerçekleştiği Şekil 12'de gösterilmiştir.



**Şekil 42.** Sınıflandırma doğruluğu grafiği.



**Şekil 53.** Eğitim verisi (Karışıklık matrisi).



**Şekil 64.** Test verisi (Karışıklık matrisi).

Tablo 6 ve Tablo 7’de sırasıyla eğitim ve test verileri için (0’dan 9’a kadar karakterler) görüntülerin belirtilen değerlendirme metriklerine göre sınıflandırma sonuçları verilmiştir.

**Tablo 6.** Eğitim için sınıflandırma raporu.

For train data				
	Precision	Recall	F1-score	Support
0	1.0000	0.9968	0.9984	5923
1	0.9994	0.9997	0.9996	6742
2	0.9995	0.9997	0.9996	5958
3	0.9982	0.9993	0.9988	6131
4	0.9991	0.9981	0.9986	5842
5	0.9993	0.9987	0.9990	5421
6	0.9970	0.9993	0.9981	5918
7	0.9992	0.9990	0.9991	6265
8	0.9991	0.9985	0.9988	5851
9	0.9980	0.9995	0.9987	5949
Accuracy			0.9989	60000
Macro avg	0.9989	0.9989	0.9989	60000
Weighted avg	0.9989	0.9989	0.9989	60000

**Tablo 7.** Test için sınıflandırma raporu.

For test data				
	Precision	Recall	F1-score	Support
0	0.9979	0.9908	0.9944	980
1	0.9956	0.9982	0.9969	1135
2	0.9942	0.9952	0.9947	1032
3	0.9863	0.9970	0.9916	1010
4	0.9938	0.9837	0.9887	982
5	0.9955	0.9877	0.9916	892
6	0.9896	0.9948	0.9922	958
7	0.9932	0.9932	0.9932	1028
8	0.9938	0.9928	0.9933	974
9	0.9822	0.9871	0.9871	1009
Accuracy			0.9922	10000
Macro avg	0.9922	0.9921	0.9921	10000
Weighted avg	0.9922	0.9922	0.9922	10000

#### 4.2 MNIST fashion sonuçları

MNIST fashion veri seti için kullandığımız katmanlar ve kullanılan parametre sayıları Tablo 8’de verilmiştir. Çalışmamızda öğrenme katsayısı 0.001, batch size 100, optimizasyon için adam optimizer algoritması ve değerlendirme metriği olarak da doğruluk kriteri belirlenmiştir. Tablo 9’da eğitim verisi için %98.49 başarı elde edildiği görülürken, test verisi için de %92.21 başarı elde edildiği Tablo 10’da görülmüştür.

**Tablo 8.** Kullanılan katmanlar ve parametreler.

Layer (type)	Output shape	Param #
Conv1 (Conv2D)	(None, 28, 28, 100)	1000
Batchnorm1 (Batchnormalization)	(None, 28, 28, 100)	400
Maxpool1 (Maxpooling2D)	(None, 14, 14, 100)	0
Conv2 (Conv2D)	(None, 14, 14, 100)	90100
Batchnorm2 (Batchnormalization)	(None, 14, 14, 100)	400
Maxpool2 (Maxpooling2D)	(None, 7, 7, 100)	0
Conv3 (Conv2D)	(None, 7, 7, 100)	90100
Batchnorm3 (Batchnormalization)	(None, 7, 7, 100)	400
Flatten1 (Flatten)	(None, 4900)	0
Dense1 (Dense)	(None, 100)	490100
Dropout1 (Dropout)	(None, 100)	0
Dense2 (Dense)	(None, 100)	10100
Dropout2 (Dropout)	(None, 100)	0
Dense3 (Dense)	(None, 10)	1010
Total params: 683.610		
Trainable params: 683.010		
Non-trainable params: 600		

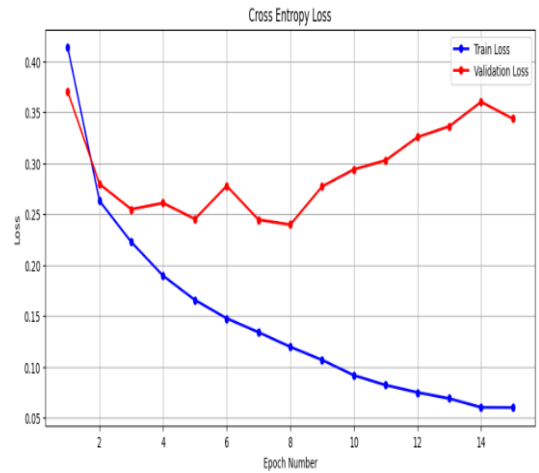
**Tablo 9.** Eğitim verisi başarı oranı.

Train Loss	Train Accuracy
0.04257245734333992	0.9849333167076111

**Tablo 10.** Test verisi başarı oranı.

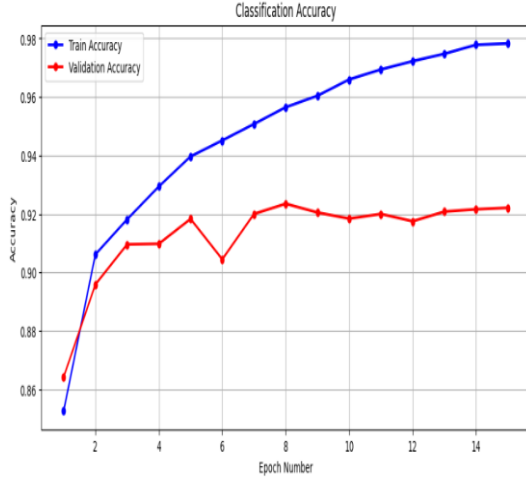
Test Loss	Test Accuracy
0.34361010789871216	0.9221000075340271

Kullandığımız veri seti için çapraz entropi (cross entropy) hata değerlerinin değişimi Şekil 15’te epoch değerlerine göre gösterilmiştir. Grafikte anlaşıldığı üzere 14. epoch da hata en yüksek, 15. epoch da hata en düşük olmaktadır.



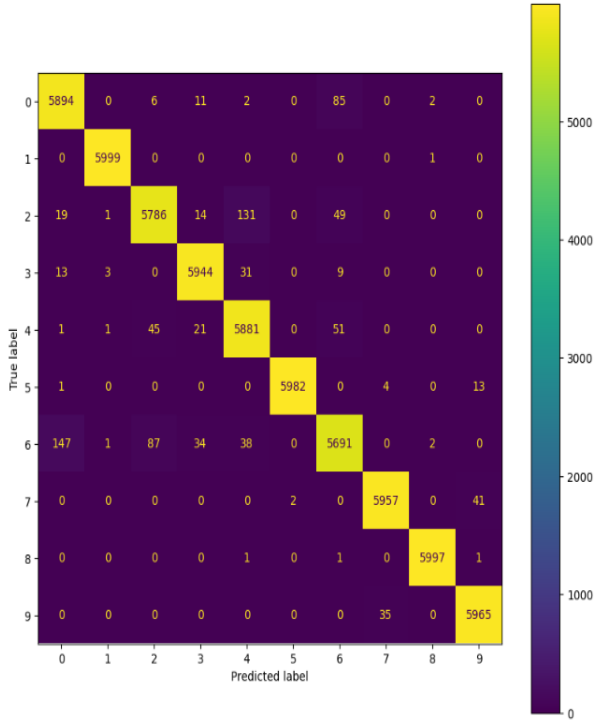
**Şekil 75.** Çapraz entropi hata grafiği.

Ağırlıklar sürekli güncellenmekte olup, en iyi öğrenmenin 8. Epoch da gerçekleştiği Şekil 16'da gösterilmiştir.



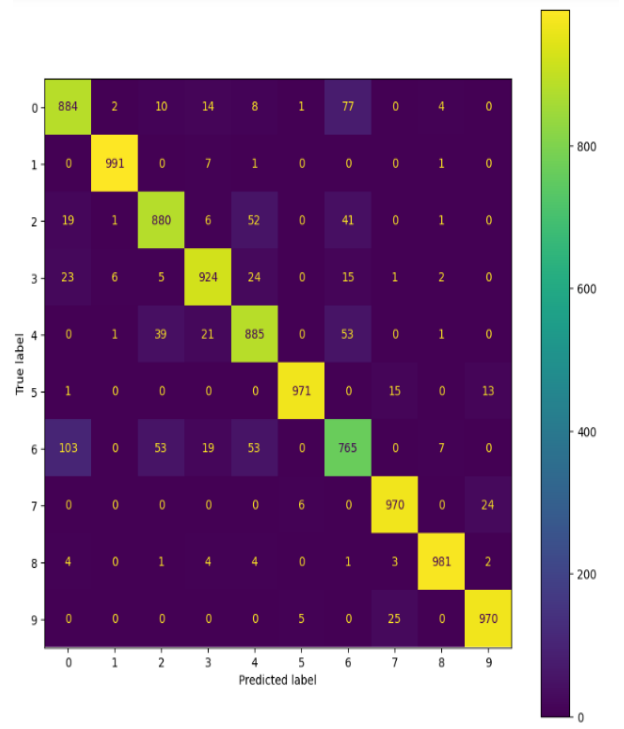
Şekil 86. Sınıflandırma doğruluğu grafiği.

Önerilen modelin eğitim ve test karışıklık matrisleri Şekil 17 ve Şekil 18'de gösterilmiştir. Buna göre eğitim ve test için en başarılı görüntü 2. görüntü iken en başarısız görüntü ise 7. görüntüdür.



Şekil 97. Eğitim verisi (Karışıklık matrisi).

Tablo 11 ve Tablo 12'de sırasıyla eğitim ve test verileri için (0'dan 9'a kadar karakterler) görüntülerin belirtilen değerlendirme metriklerine göre sınıflandırma sonuçları verilmiştir.



Şekil 108. Test verisi (Karışıklık matrisi).

Tablo 11. Eğitim için sınıflandırma raporu.

For train data				
	Precision	Recall	F1-score	Support
0	0.9702	0.9823	0.9762	6000
1	0.9990	0.9998	0.9994	6000
2	0.9767	0.9643	0.9705	6000
3	0.9867	0.9907	0.9887	6000
4	0.9666	0.9802	0.9734	6000
5	0.9997	0.9970	0.9983	6000
6	0.9669	0.9485	0.9576	6000
7	0.9935	0.9928	0.9932	6000
8	0.9992	0.9995	0.9993	6000
9	0.9909	0.9942	0.9925	6000
Accuracy			0.9849	60000
Macro avg	0.9849	0.9849	0.9849	60000
Weighted avg	0.9849	0.9849	0.9849	60000

**Tablo 12.** Test için sınıflandırma raporu.

For test data				
	Precision	Recall	F1-score	Support
0	0.8549	0.8840	0.8692	1000
1	0.9900	0.9910	0.9905	1000
2	0.8907	0.8800	0.8853	1000
3	0.9286	0.9240	0.9263	1000
4	0.8617	0.8850	0.8732	1000
5	0.9878	0.9710	0.9793	1000
6	0.8036	0.7650	0.7838	1000
7	0.9566	0.9700	0.9633	1000
8	0.9840	0.9810	0.9825	1000
9	0.9613	0.9700	0.9657	1000
Accuracy			0.9221	10000
Macro avg	0.9219	0.9221	0.9219	10000
Weighted avg	0.9219	0.9221	0.9219	10000

#### 4.3 CIFAR-10 sonuçları

CIFAR-10 veri seti için kullandığımız katmanlar ve kullanılan parametre sayıları Tablo 13'te verilmiştir. Çalışmamızda öğrenme katsayısı 0.001, batch size 100, optimizasyon için adam optimizer algoritması ve değerlendirme metriği olarak da doğruluk kriteri belirlenmiştir. Tablo 14'te eğitim verisi için %95.64 başarı elde edildiği görülürken, test verisi için de %74.63 başarı elde edildiği Tablo 15'te görülmüştür.

**Tablo 13.** Kullanılan katmanlar ve parametreler.

Layer (type)	Output shape	Param #
Conv1 (Conv2D)	(None, 32, 32, 100)	2800
Batchnorm1 (Batchnormalization)	(None, 32, 32, 100)	400
Maxpool1 (Maxpooling2D)	(None, 16, 16, 100)	0
Conv2 (Conv2D)	(None, 16, 16, 100)	90100
Batchnorm2 (Batchnormalization)	(None, 16, 16, 100)	400
Maxpool2 (Maxpooling2D)	(None, 8, 8, 100)	0
Conv3 (Conv2D)	(None, 8, 8, 100)	90100
Batchnorm3 (Batchnormalization)	(None, 8, 8, 100)	400
Flatten1 (Flatten)	(None, 6400)	0
Dense1 (Dense)	(None, 100)	640100
Dropout1 (Dropout)	(None, 100)	0
Dense2 (Dense)	(None, 100)	10100
Dropout2 (Dropout)	(None, 100)	0
Dense3 (Dense)	(None, 10)	1010
Total params: 835.410		
Trainable params: 834.810		
Non-trainable params: 600		

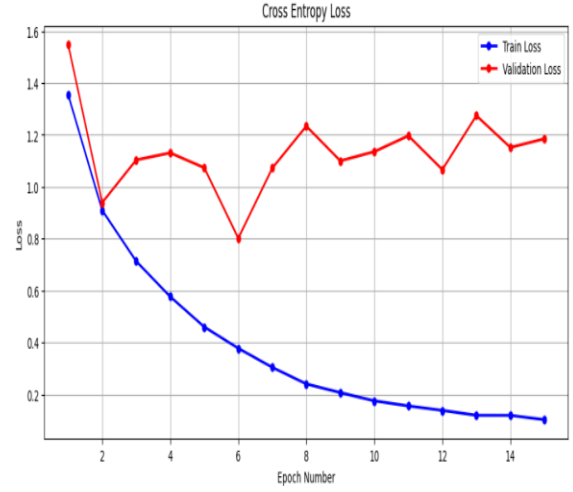
**Tablo 14.** Eğitim verisi başarı oranı.

Train Loss	Train Accuracy
0.1318219006061554	0.9564200043678284

**Tablo 15.** Test verisi başarı oranı.

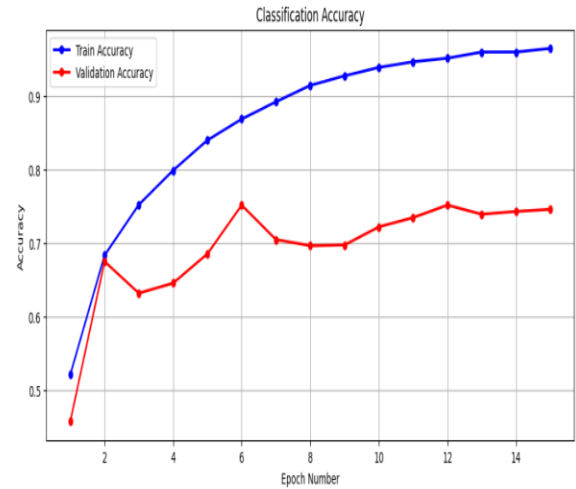
Test Loss	Test Accuracy
1.185105323791504	0.746399998664856

Kullandığımız veri seti için çapraz entropi (cross entropy) hata değerlerinin değişimi Şekil 19'da epoch değerlerine göre gösterilmiştir. Grafikte anlaşıldığı üzere 14. epoch da hata en yüksek, 15. epoch da hata en düşük olmaktadır.



**Şekil 119.** Çapraz entropi hata grafiği.

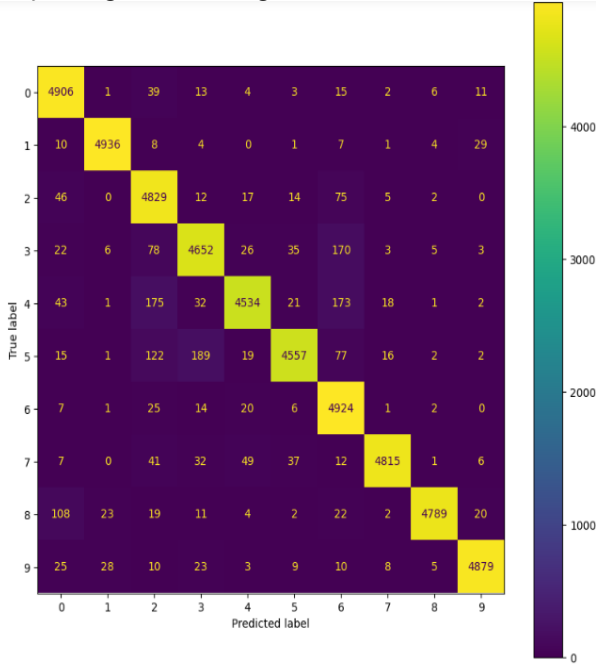
Ağırlıklar sürekli güncellenmekte olup, en iyi öğrenmenin 6. Epoch da gerçekleştiği Şekil 20'de gösterilmiştir.



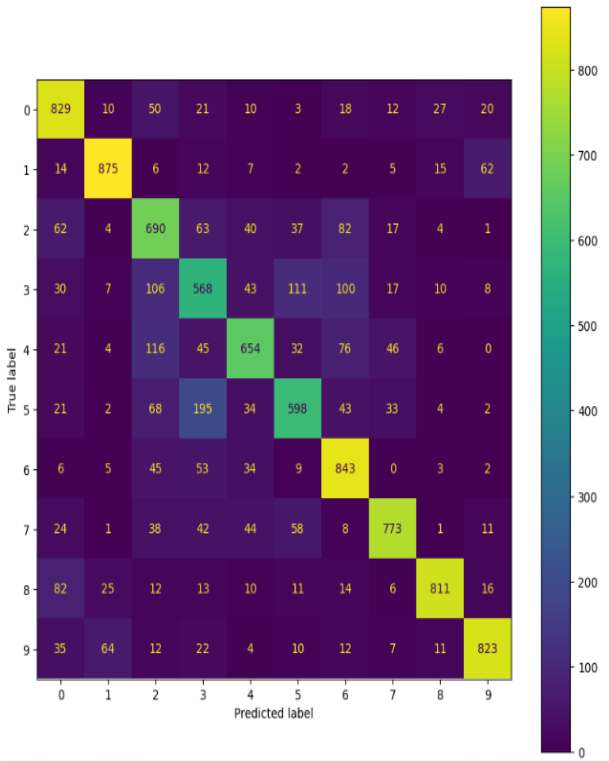
**Şekil 20.** Sınıflandırma doğruluğu grafiği.



Önerilen modelin eğitim ve test karışıklık matrisleri Şekil 21 ve Şekil 22’de gösterilmiştir. Buna göre eğitim için en başarılı görüntü 2.görüntü iken, en başarısız görüntü ise 5. görüntüdür. Test için ise en başarılı görüntü 2.görüntü iken, en başarısız görüntü ise 4. görüntüdür.



Şekil 21. Eğitim verisi (Karışıklık matrisi).



Şekil 22. Test verisi (Karışıklık matrisi).

değerlendirme metriklerine göre sınıflandırma sonuçları verilmiştir.

Tablo 16. Eğitim için sınıflandırma raporu.

For train data				
	Precision	Recall	F1-score	Support
0	0.9455	0.9812	0.9630	5000
1	0.9878	0.9872	0.9875	5000
2	0.9033	0.9658	0.9335	5000
3	0.9338	0.9304	0.9321	5000
4	0.9696	0.9068	0.9372	5000
5	0.9727	0.9114	0.9410	5000
6	0.8977	0.9848	0.9392	5000
7	0.9885	0.9630	0.9756	5000
8	0.9942	0.9578	0.9757	5000
9	0.9853	0.9758	0.9805	5000
Accuracy			0.9564	50000
Macro avg	0.9578	0.9564	0.9565	50000
Weighted avg	0.9578	0.9564	0.9565	50000

Tablo 17. Test için sınıflandırma raporu.

For test data				
	Precision	Recall	F1-score	Support
0	0.7375	0.8290	0.7806	1000
1	0.8776	0.8750	0.8763	1000
2	0.6037	0.6900	0.6440	1000
3	0.5493	0.5680	0.5585	1000
4	0.7432	0.6540	0.6957	1000
5	0.6866	0.5980	0.6392	1000
6	0.7037	0.8430	0.7671	1000
7	0.8439	0.7730	0.8069	1000
8	0.9092	0.8110	0.8573	1000
9	0.8709	0.8230	0.8463	1000
Accuracy			0.7464	10000
Macro avg	0.7526	0.7464	0.7472	10000
Weighted avg	0.7526	0.7464	0.7472	10000

Tablo 16 ve Tablo 17’de sırasıyla eğitim ve test verileri için (0’den 9’a kadar karakterler) görüntülerin belirtilen

#### 4.4 CIFAR-100 sonuçları

CIFAR-100 veri seti için kullandığımız katmanlar ve kullanılan parametre sayıları **Tablo 18**'de verilmiştir. Çalışmamızda öğrenme katsayısı 0.001, batch size 100, optimizasyon için adam optimizer algoritması ve değerlendirme metriği olarak da doğruluk kriteri belirlenmiştir. **Tablo 19**'da eğitim verisi için %87.73 başarı elde edildiği görülürken, test verisi için de %40.00 başarı elde edildiği **Tablo 20**'de görülmüştür.

**Tablo 18.** Kullanılan katmanlar ve parametreler.

Layer (type)	Output shape	Param #
Conv1 (Conv2D)	(None, 32, 32, 100)	2800
Batchnorm1 (Batchnormalization)	(None, 32, 32, 100)	400
Maxpool1 (Maxpooling2D)	(None, 16, 16, 100)	0
Conv2 (Conv2D)	(None, 16, 16, 100)	90100
Batchnorm2 (Batchnormalization)	(None, 16, 16, 100)	400
Maxpool2 (Maxpooling2D)	(None, 8, 8, 100)	0
Conv3 (Conv2D)	(None, 8, 8, 100)	90100
Batchnorm3 (Batchnormalization)	(None, 8, 8, 100)	400
Flatten1 (Flatten)	(None, 6400)	0
Dense1 (Dense)	(None, 100)	640100
Dropout1 (Dropout)	(None, 100)	0
Dense2 (Dense)	(None, 100)	10100
Dropout2 (Dropout)	(None, 100)	0
Dense3 (Dense)	(None, 10)	10100
Total params: 844.500		
Trainable params: 843.900		
Non-trainable params: 600		

**Tablo 19.** Eğitim verisi başarı oranı.

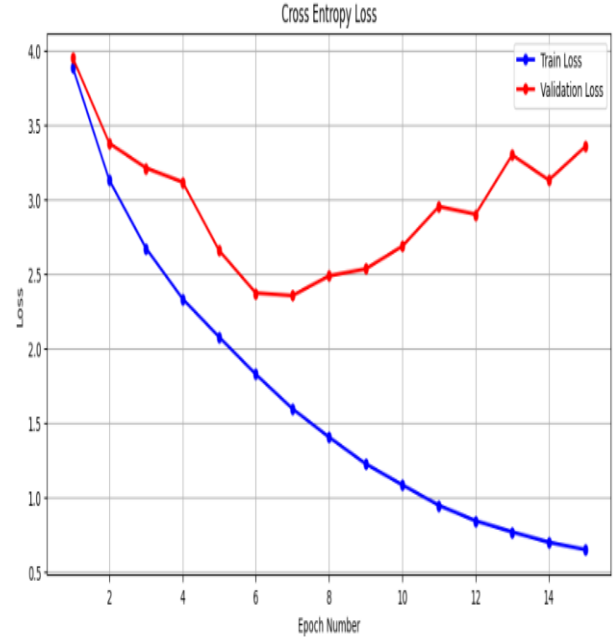
Train Loss	Train Accuracy
0.41445985436439514	0.8773599863052368

**Tablo 20.** Test verisi başarı oranı.

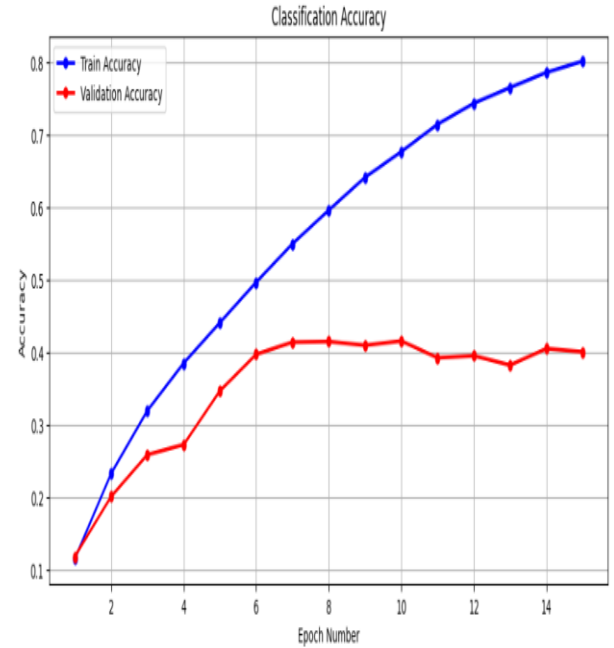
Test Loss	Test Accuracy
3.354889392852783	0.4000000059604645

Kullandığımız veri seti için çapraz entropi (cross entropy) hata değerlerinin değişimi **Şekil 23**'te epoch değerlerine göre gösterilmiştir. Grafikte anlaşıldığı üzere 14. epoch da hata en yüksek, 15. epoch da hata en düşük olmaktadır. Ağırlıklar sürekli güncellenmekte olup, en iyi

öğrenmenin 6. Epoch da gerçekleştiği **Şekil 24**'te gösterilmiştir.



**Şekil 23.** Çapraz entropi hata grafiği.



**Şekil 24.** Sınıflandırma doğruluğu grafiği.

Önerilen modelin eğitim ve test için karışıklık matrisi **Şekil 25**'te gösterilmiştir. Buna göre eğitim için en başarılı görüntü 24.görüntü iken, en başarısız görüntü ise 45. görüntüdür. Test için ise en başarılı görüntü 37.görüntü iken, en başarısız görüntü ise 14. görüntüdür.

```
For Train Data
[[462  0  1 ...  0  2  0]
 [  0 483  0 ...  0  0  0]
 [  2  0 435 ...  3  8  1]
 ...
 [  0  0  0 ... 475  0  0]
 [  0  0  2 ...  1 463  0]
 [  0  0  0 ...  0  2 433]]

For Test Data
[[69  3  0 ...  0  1  0]
 [  0 58  0 ...  1  0  1]
 [  2  2 27 ...  0  9  0]
 ...
 [  0  0  1 ... 42  0  1]
 [  0  1  7 ...  0 32  2]
 [  0  0  0 ...  0  1 41]]
```

Şekil 25. Eğitim ve test verisi için karışıklık matrisi.

```
For Train Data
[[462  0  1 ...  0  2  0]
 [  0 483  0 ...  0  0  0]
 [  2  0 435 ...  3  8  1]
 ...
 [  0  0  0 ... 475  0  0]
 [  0  0  2 ...  1 463  0]
 [  0  0  0 ...  0  2 433]]

For Test Data
[[69  3  0 ...  0  1  0]
 [  0 58  0 ...  1  0  1]
 [  2  2 27 ...  0  9  0]
 ...
 [  0  0  1 ... 42  0  1]
 [  0  1  7 ...  0 32  2]
 [  0  0  0 ...  0  1 41]]
```

Şekil 26. Eğitim ve test verisi için karışıklık matrisi.

Tablo 21 ve Tablo 22’de sırasıyla eğitim ve test verileri için (0’dan 99’a kadar karakterler) görüntülerin belirtilen değerlendirme metriklerine göre sınıflandırma sonuçları verilmiştir.

Tablo 21. Eğitim için sınıflandırma raporu.

For Train Data				
	precision	recall	f1-score	support
0	0.9277	0.9240	0.9259	500
1	0.9200	0.9660	0.9424	500
2	0.8398	0.8700	0.8546	500
3	0.9117	0.8260	0.8667	500
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
97	0.8290	0.9500	0.8854	500
98	0.7578	0.9260	0.8335	500
99	0.7901	0.8660	0.8263	500
accuracy			0.8774	50000
macro avg	0.8838	0.8774	0.8772	50000
weighted avg	0.8838	0.8774	0.8772	50000

Tablo 22. Test için sınıflandırma raporu.

For Test Data				
	precision	recall	f1-score	support
0	0.6765	0.6900	0.6832	100
1	0.4496	0.5800	0.5066	100
2	0.2647	0.2700	0.2673	100
3	0.2500	0.2200	0.2340	100
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
97	0.3590	0.4200	0.3871	100
98	0.1916	0.3200	0.2397	100
99	0.3280	0.4100	0.3644	100
accuracy			0.4000	10000
macro avg	0.4211	0.4000	0.4015	10000
weighted avg	0.4211	0.4000	0.4015	10000

## 5 Sonuçlar

Çalışmamız farklı bir ağ yapısı ile modellenerek bu modelin dört veri seti için eğitim ve testlerdeki sınıflandırma başarısı doğruluk, duyarlılık, kesinlik ve F1 skoru metrikleri kullanılarak değerlendirilmiştir. Önerdiğimiz CNN modeli görüntü sınıflandırma ve tahmin edilmesinde başarılı sonuçlar vermiştir. MNIST ve Fashion MNIST veri setleri için doğruluk oranları sırasıyla %99.22 ve %99.21 olarak gözlemlenmiştir.

Ayrıca, hem eğitim hem de test verilerinin doğruluk sonuçlarının grafikleri çalışmamızda ayrı ayrı gösterilmiş

olup kullanılan tüm veri setleri için en iyi ve en kötü bulunan görüntüler de tespit edilmiştir.

CIFAR-100 sonuçları istenilen değerlerin altında çıktığı için ilerleyen çalışmalarda daha derin bir ağ ya da hibrit yöntemlerin tasarlanıp kullanılması hedeflenmektedir.

Diğer çalışmalardan farklı olarak önerilen yeni ağ modeli ile farklı görüntülerde hem performans hem de hız olarak başarılı sonuçlar elde edilmiştir. Daha fazla değerlendirme kriteri ile önerilen modelin daha fazla analiz edilmesi sağlanmıştır. Önerilen ağ modelinin amacı ileride biyomedikal görüntülerde, tıbbi verilerde hastalık tespiti ve sınıflandırılmasını daha doğru bir biçimde gerçekleştirmektir. İleride bu amaca yönelik çalışmalar yapılacaktır. Ayrıca gelecekte her hastalığa özgü farklı ağ modeline sahip yöntemler geliştirilecektir. Bu önerilen model gelecekte yapılacak çalışmaların temelini oluşturacaktır.

#### Çıkar çatışması

Yazarlar çıkar çatışması olmadığını beyan etmektedir. Bu çalışma Yeşim TİRAKİ'nin yüksek lisans tezinden üretilmiştir.

**Benzerlik oranı (iThenticate): %15**

#### Kaynaklar

- [1] E. Moen, D. Bannon, T. Kudo, W. Graf, M. Covert and D. Valen, Deep learning for cellular image analysis, Nature Methods, 16(12), 1233-1246, 2019.
- [2] F. Cevik, Z. Kilimci, Derin öğrenme yöntemleri ve kelime yerleştirme modelleri kullanılarak Parkinson hastalığının duygu analiziyle değerlendirilmesi, Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 27(2), 151-161, 2020. [https:// doi: 10.5505/pajes.2020.74429](https://doi.org/10.5505/pajes.2020.74429).
- [3] Z. Pekoz and T. İnkaya, Derin öğrenme ile talep tahmini: Bir üçüncü parti lojistik firması için covid-19 döneminde vaka analizi, Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 1000(1000), 1-8, 2022. [https:// doi: 10.5505/pajes.2022.73537](https://doi.org/10.5505/pajes.2022.73537).
- [4] T. Karahan and V. Nabiyev, Plant identification with convolutional neural networks and transfer learning, Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 27(5), 638-645, 2021. [https:// doi: 10.5505/pajes.2020.84042](https://doi.org/10.5505/pajes.2020.84042).
- [5] O. Abouelnaga and H. Ali, Cifar-10: knn-based ensemble of classifiers, 2016 International Conference on Computational Science and Computational Intelligence, Las Vegas, Nevada, USA, 15-17 December 2016.
- [6] G. Tushar, Comparative study of various convolutional neural networks on cifar-10, International Journal for Modern Trends in Science and Technology, 6(12), 402-406, 2020.
- [7] H. Tien, Cifar-10 to compare visual recognition performance between deep neural networks and humans. arXiv, abs/1811.07270, 2018.
- [8] R. Ju, Y. Lin and T. Jian, Efficient convolutional neural networks on raspberry pi for image classification, Journal of Real-Time Image Processing, 20(2), 1-9, 2023. <https://doi.org/10.1007/s11554-023-01271-1>
- [9] A. Yahya, K. Liu, A. Hawbani and Y. Wang, A novel image classification method based on residual network. Inception, and Proposed Activation Function. Sensors, 23(6), 2976, 2023. <https://doi.org/10.3390/s23062976>.
- [10] X. Zhao, P. Huang P and H. Shu, Wavelet-attention CNN for image classification, Multimedia Systems, 28(3), 915-924, 2022. [https://doi.org/ 0.1007/s00530-022-00889-8](https://doi.org/10.1007/s00530-022-00889-8).
- [11] K. Zhang, Y. X. Guo, J. Yuan and Q. Ding, Multiple feature reweight densenet for image classification, IEEE Access, 7, 9872-9880, 2019. [https://doi.org/ 10.1109/ACCESS.2018.2890127](https://doi.org/10.1109/ACCESS.2018.2890127).
- [12] H. Shao, E. Ma, M. Zhu, X. Deng, S. Zhai, Mnist handwritten digit classification based on convolutional neural network with hyperparameter optimization, Intelligent Automation & Soft Computing, 36(3), 2023. [https://doi.org/ 10.32604/iasec.2023.036323](https://doi.org/10.32604/iasec.2023.036323).
- [13] E. Plesovskaya and S. Ivanov, Hierarchical classification on the mnist dataset using truncated svd and kernel density estimation, Procedia Computer Science, 212, 368-377, 2022.
- [14] Y. Pei and L. Ye, Cluster analysis of mnist data set, In Journal of Physics: Conference Series, 2181(1), 12-12, 2022. <https://doi.org/10.1088/1742-6596/2181/1/012035>.
- [15] A. Henrique, Classifying garments from fashion-mnist dataset through CNNs, Journal Advances in Science Technology and Engineering Systems, 6(1), 989-994, 2021. [https://doi.org/ 10.25046/aj0601109](https://doi.org/10.25046/aj0601109).
- [16] Ş. Öztürk, Moda görseli sınıflandırma: düzenleyici teknikler ile evrimsel sinir ağları uygulaması, Bilgisayar Bilimleri ve Mühendisliği Dergisi, 15(1), 66-76, 2022. [https://doi.org/ 10.54525/tbbmd.1077432](https://doi.org/10.54525/tbbmd.1077432).
- [17] S. Bhatnagar, D. Ghosal and M.H. Kolekar, Classification of fashion article images using convolutional neural networks. Fourth International Conference on Image Information Processing, 357-362, India, 21-23 December 2017.
- [18] K. Meshkini, J. Platos and H. Ghassemian, An analysis of convolutional neural network for fashion images classification (fashion-mnist). International Conference on Intelligent Information Technologies for Industry, 85-95, Ostrava-Prague, Czech Republic, 2-7 December 2019.
- [19] K. Greeshma and K. Sree Kumar, Hyperparameter optimization and regularization on fashion-mnist classification, International Journal of Recent Technology and Engineering, 8(2), 3713-3719, 2019. [https://doi.org/ 10.35940/ijrte.B3092.078219](https://doi.org/10.35940/ijrte.B3092.078219).
- [20] M. Kayed, A. Anter and H. Mohamed, Classification of garments from fashion mnist dataset using cnn lenet-5 architecture. International Conference on Innovative Trends in Communication and Computer Engineering, 238-243, Aswan, Egypt, 8-9 February 2020.
- [21] T. Hur, L. Kim and D. Park, Quantum convolutional neural network for classical data classification,



- Quantum Machine Intelligence, 4(3), 1-18, (2022). <https://doi.org/10.1007/s42484-021-00061-x>.
- [22] Y. Seo and K. Shin, Hierarchical convolutional neural networks for fashion image classification, Expert Systems with Applications, 116, 328-339, 2019. <https://doi.org/10.1016/j.eswa.2018.09.022>.
- [23] OM. Khanday, S. Dadvandipour S and MA. Lone, Effect of filter sizes on image classification in cnn: a case study on cfar10 and fashion-mnist datasets, International Journal of Artificial Intelligence, 10(4), 872-878, 2021. <https://doi.org/10.11591/ijai.v10.i4.pp872-878>.
- [24] O. Nocentini, J. Kim and F. Cavallo, Image classification using multiple convolutional neural networks on the fashion-mnist dataset, Sensors, 22(23), 9544, 2022. <https://doi.org/10.3390/s22239544>.
- [25] A. Vijayaraj, V. Raj and R. Dhanagopal, Deep learning image classification for fashion design, Wireless Communications and Mobile Computing, 2022, 1-10, 2022. <https://doi.org/10.1155/2022/7549397>.
- [26] AC. Cleber, A novel multi-objective grammar-based framework for the generation of convolutional neural networks, Expert Systems With Applications, 212, 2023. <https://doi.org/10.1016/j.eswa.2022.118670>.
- [27] R. Elyas, A. Akbari and B. NaserSharif, Uncertainty handling in convolutional neural networks. Neural Computing and Applications, 34, 16753-16769, 2022. <https://doi.org/10.1007/s00521-022-07313-2>.
- [28] O. Güler and İ. Yücedağ, Hand gesture recognition from 2d images by using convolutional capsule neural networks, Arabian Journal for Science and Engineering, 47, 1211-1225, 2022. <https://doi.org/10.1007/s13369-021-05867-2>.
- [29] G. Ryu and D. Choi, A hybrid adversarial training for deep learning model and denoising network resistant to adversarial examples, Applied Intelligence, 53, 1-14, 2022.
- [30] AT. Silva and RS. Siebert, Densenet-dc: optimizing densenet parameters through feature map generation control, Revista de Informática Teórica e Aplicada, 27, 25-39, 2020.
- [31] F. Assunção, N. Lourenço and P. Machado, Denser: deep evolutionary network structured representation, Genetic Programming and Evolvable Machines, 20, 5-35, 2019.
- [32] AM. Hafız, RA. Bhat RA and M. Hassaballah, Image classification using convolutional neural network tree ensembles, Multimedia Tools and Applications, 82(5), 6867-6884, 2023. <https://doi.org/10.1007/s11042-022-13604-6>.
- [33] H. Gürkan ve A. Hanılçı, Evrimsel sinir ağı ve qrs imgeleri kullanarak ekg tabanlı biyometrik tanıma yöntemi, Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 26(2), 318-327, 2020.
- [34] P. Shinde ve S. Shah, A review of machine learning and deep learning applications, 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 16-18 August 2018.
- [35] G. Ciaburro and B. Venkateswaran, Neural Networks with R: Smart Models using CNN, RNN, Deep Learning, and Artificial Intelligence Principles, Packt Publishing Ltd, Birmingham, 2017.
- [36] M. Khishe and A. Safari, Classification of sonar targets using an mlp neural network trained by dragonfly algorithm, Wireless Personal Communications, 108(4), 2241-2260, 2019. <https://doi.org/10.1007/s11277-019-06520-w>.
- [37] P. Zhang, J. Xue, C. Lan, W. Zeng, Z. Gao and N. Zheng, Eleatt-rnn: adding attentiveness to neurons in recurrent neural networks, IEEE Transactions on Image Processing, 29, 1061-1073, 2019. <https://doi.org/10.1109/TIP.2019.2937724>.
- [38] CIFAR-10 and CIFAR-100, <https://yann.lecun.com/exdb/MNIST/> (10. 05.2023)
- [39] H. Xiao, R. Kashif and R. Vollgraf, Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747, 2017.
- [40] MNIST and Fashion MNIST, <https://www.cs.toronto.edu/~kriz/cifar.html> (10. 05.2023)

