

# SVC ve CNN Model Eğitimi

---

# Proje Hakkında

Bu projenin amacı, CIFAR-10 veri seti üzerinde Convolutional Neural Network (CNN) ve Support Vector Classifier (SVC) modellerini kullanarak görüntü sınıflandırması gerçekleştirmektir. Proje sonunda eğitilen iki modelin başarı oranları karşılaştırılacaktır.

---

# İçindekiler

- Materyal ve Metod
- Veri Seti (CIFAR-10) ile İlgili Detaylı Bilgi
- Model ve Mimariler
- SVC Model Değerlendirmesi
- CNN Model Değerlendirmesi
- Genel Değerlendirme
- Sonuç
- Kaynaklar



# Materyal ve Metod

## Metod

Veri Ön İşleme

Model Eğitimi

Model Değerlendirmesi

## Materyal

Python Programlama dili

Colab

CIFAR-10 veri seti

Numpy: Sayısal hesaplamalar ve büyük, çok boyutlu dizilerle çalışmak için kullanılır.

Pandas: Veri manipülasyonu ve analizi için kullanılır

Matplotlib.pyplot: Veri görselleştirme için temel grafik kütüphanesi.

Seaborn: İstatistiksel verilerin görselleştirilmesi için kullanılan üst seviye grafik kütüphanesi.

Scikit-learn: Çeşitli makine öğrenmesi algoritmaları ve araçları için kullanılan kapsamlı bir kütüphane.

tensorflow: Derin öğrenme modelleri oluşturmak ve eğitmek için kullanılan popüler bir kütüphane.

warnings: Python kodundaki uyarıları kontrol etmek ve yönetmek için kullanılır.



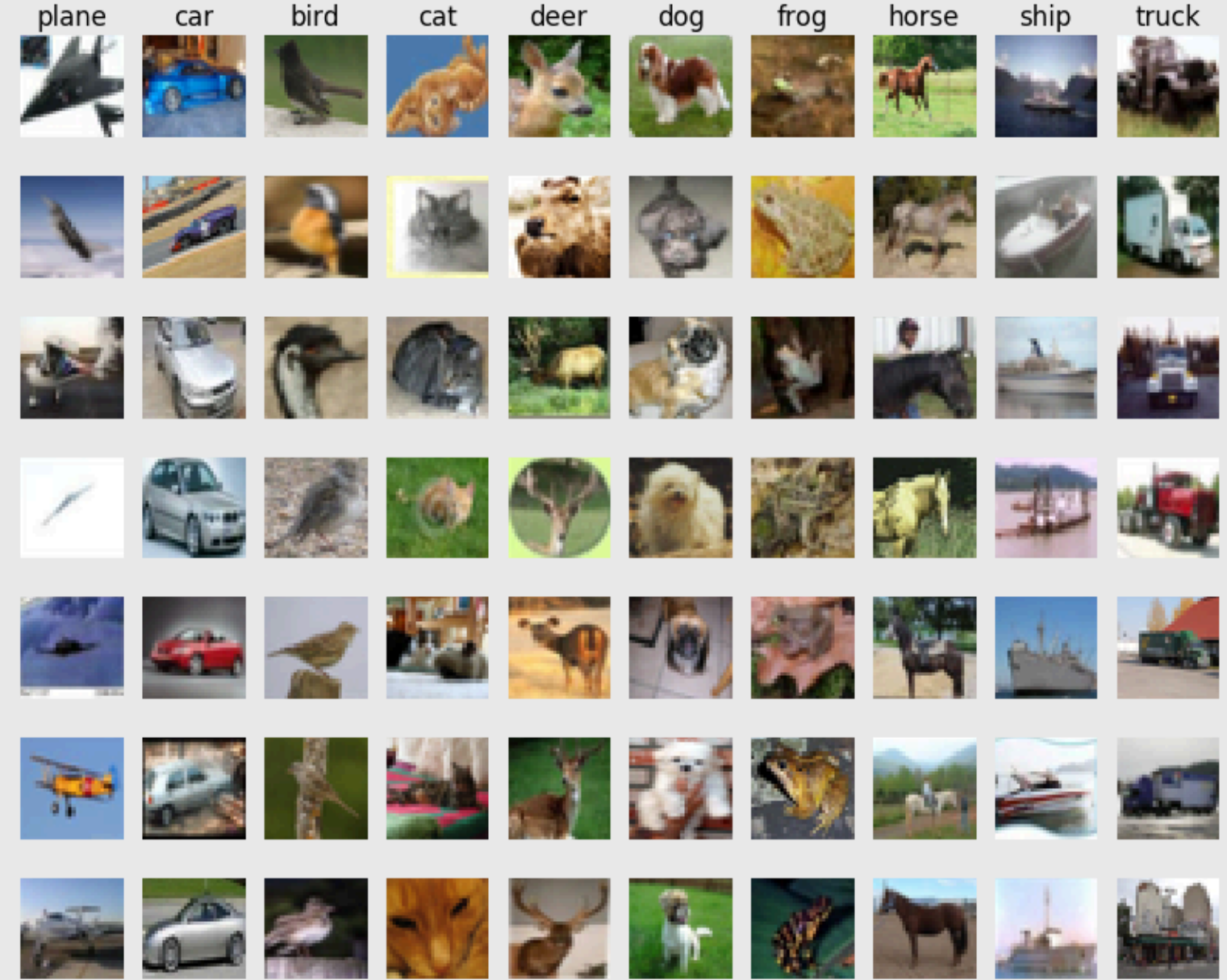
# Veri Seti

CiFAR-10 veri seti toplamda 60,000 renkli görüntü içerir.

Her bir görüntü 32x32 piksel boyutunda ve RGB formatındadır.

Veri seti 10 sınıfa ayrılmıştır ve her bir sınıfta 6000 görüntü bulunmaktadır.

Veri seti eğitim için 50,000, test için 10,000 görüntü olarak ikiye ayrılmıştır.



CiFAR-10 Veri setine ait örnek görüntüler

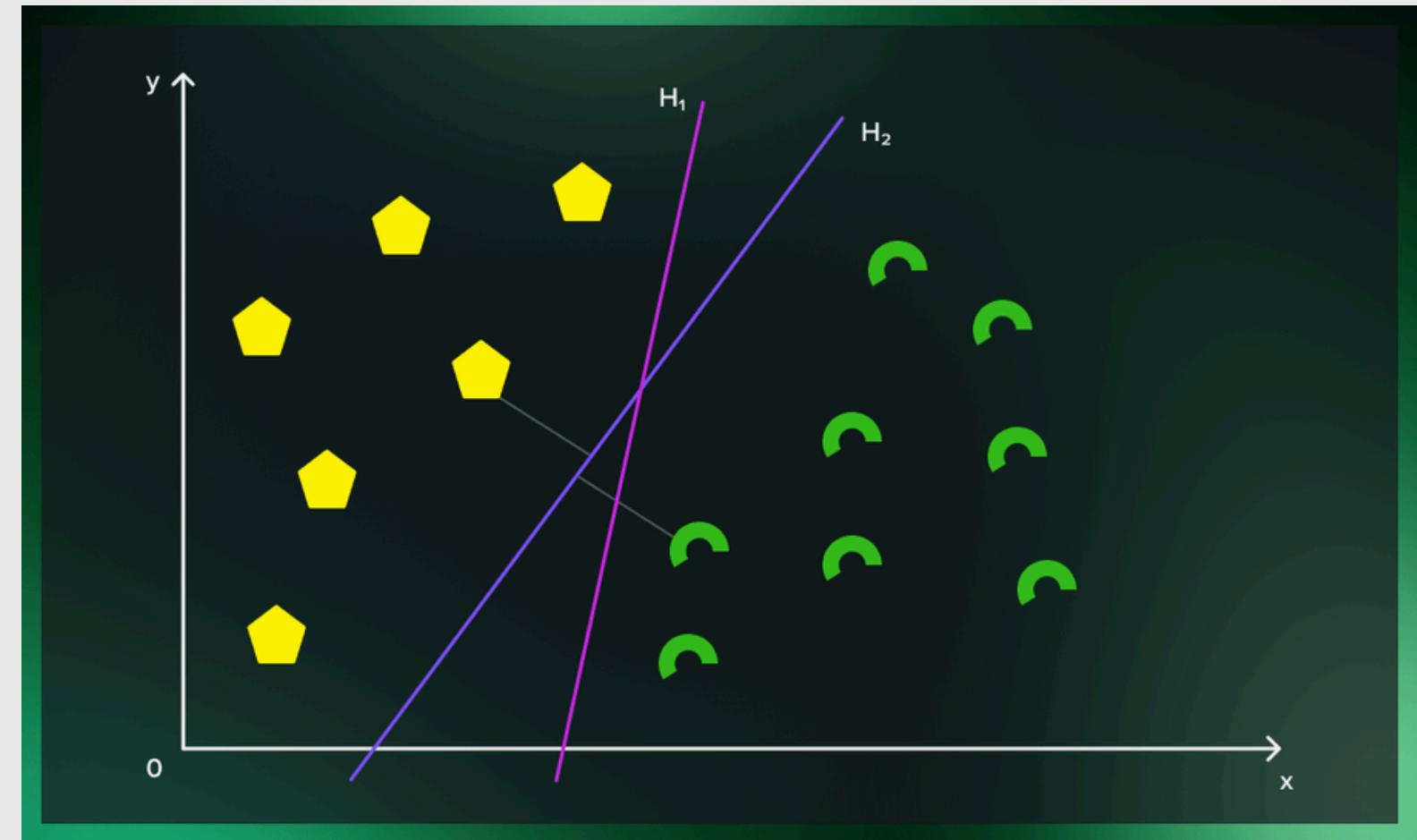
# Projede Kullanılan Modeller

## Support Vector Classifier (SVC)

Support Vector Classifier (SVC) sınıflandırma, regresyon ve aykırı değerleri bulmak için kullanılan denetimli (supervised) bir öğrenme tekniğidir. SVM algoritması classification kavramı adı altında gelişen ve diğer classification türlerinden farklı olan bir algoritmadır.

SVC algoritması bir düzlem üzerine yerleştirilmiş 2 veya daha fazla nokta kümelerini ayırmak için doğrular çizer. 2 veri kümesi düşünüldüğünde bu doğrunun, iki kümenin noktaları için de maksimum uzaklıkta olmasını amaçlar. Karmaşık ama küçük ve orta ölçekteki veri setleri için uygundur

<https://mfatihto.medium.com/support-vector-machine-algoritması-makine-öğrenmesi-8020176898d8>

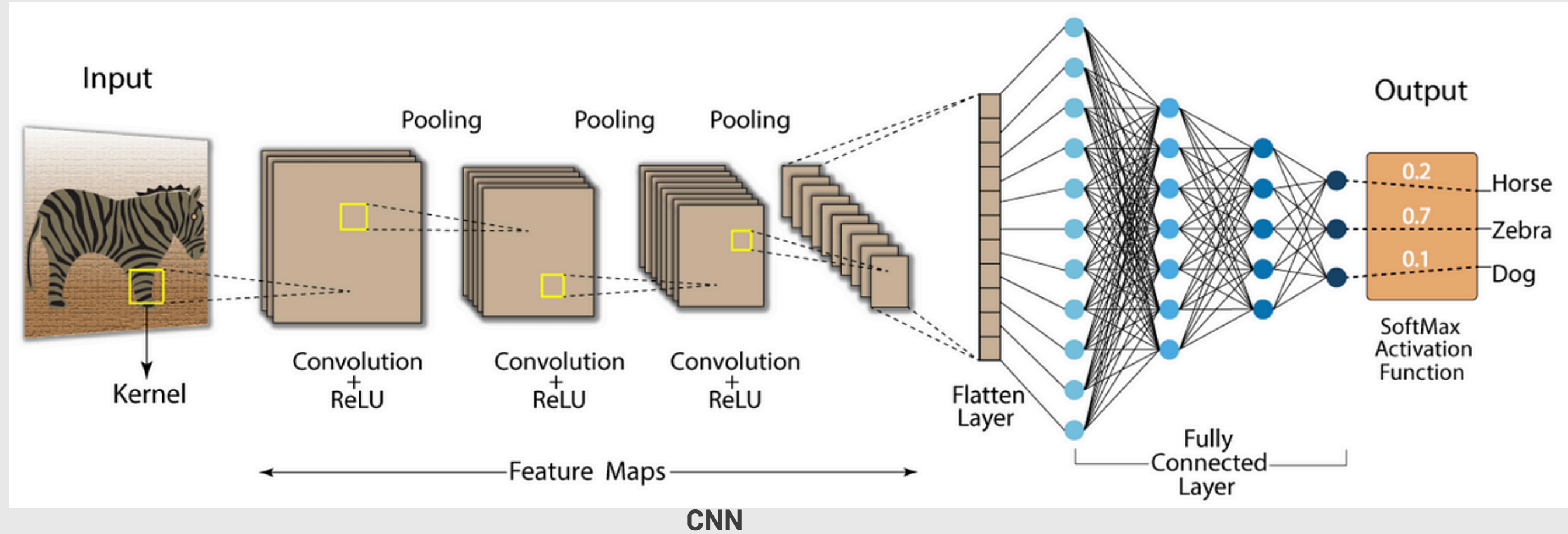


SVC

# Convolutional Neural Networks (CNN)

Convolutional Neural Networks, özellikle görüntü ve video işleme için tasarlanmış derin öğrenme modelleridir. CNN'ler, görüntülerdeki yerel özellikleri öğrenmek için konvolüsyon katmanlarını kullanır. Convolutional Layer, Pooling ve Fully Connected olan bu katmanlardan geçen görsel, farklı işlemlere tabii tutularak derin öğrenme modeline girecek kıvama gelir.

- Konvolüsyon Katmanı (Convolution Layer): Girdiye uygulanan filtreler (kernel) aracılığıyla özellik haritaları oluşturur. Bu katman yerel bağlantıları ve ağırlık paylaşımını kullanarak çalışır.
- Havuzlama Katmanı (Pooling Layer): Özellik haritalarını aşağı örnekleyerek (downsample) boyutunu küçültür ve modelin hesaplama yükünü azaltır. Yaygın havuzlama yöntemleri max pooling ve average pooling'dir.
- Tam Bağlantılı Katman (Fully Connected Layer): Son katman olarak kullanılır ve sınıflandırma işlemini gerçekleştirir.



# SVC Eğitimi ve Değerlendirme

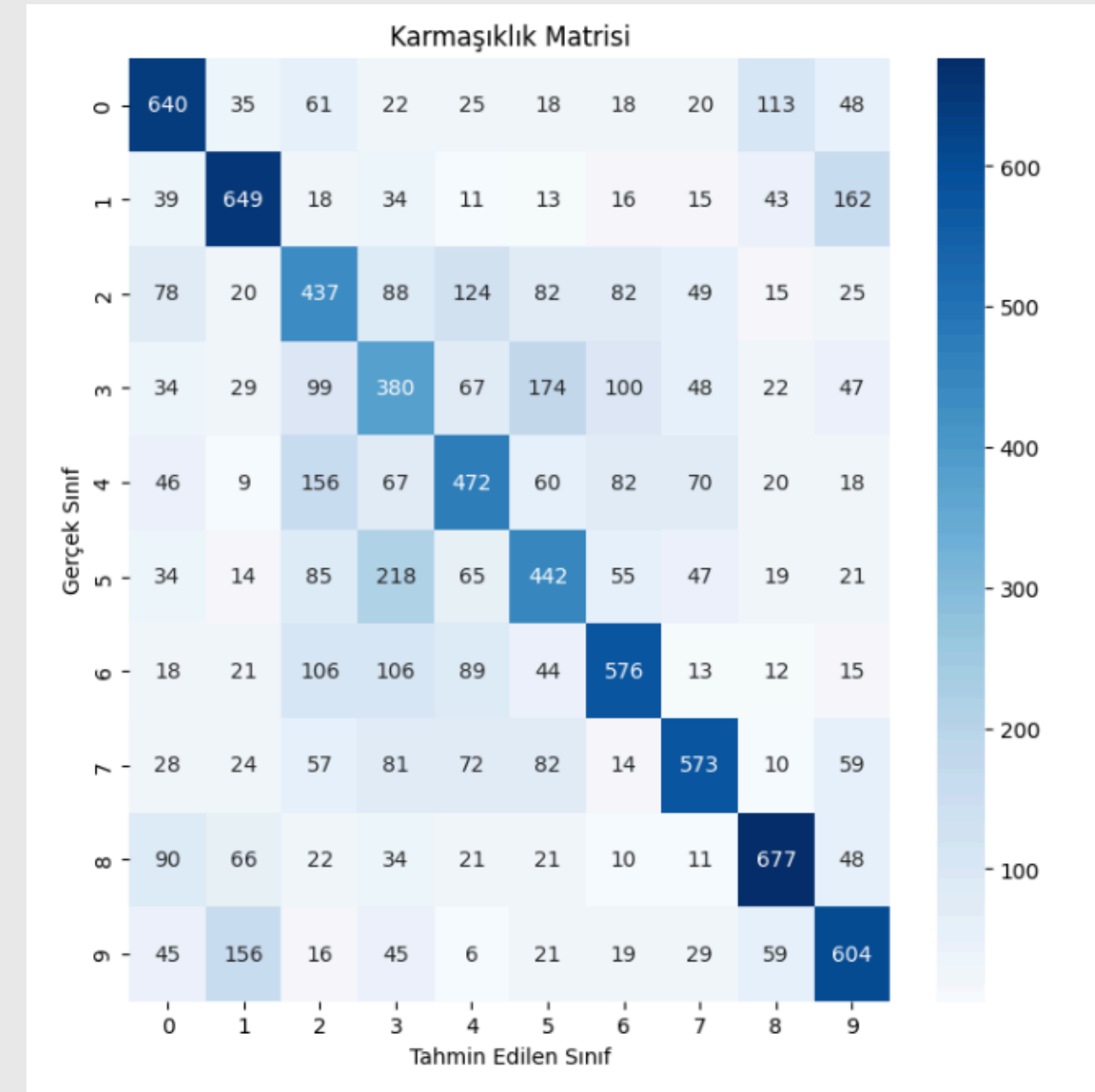
Veri seti yüklendi ve %70 oranında küçültüldü -eğitimin daha kısa sürmesi için-. Yapılan veri ön işlemlerinin ardından SVC modeli oluşturulup eğitildi. Elde edilen sonuçlar:

Accuracy: 0.545

Precision:0.5476

Recall: 0.545

F1 Score: 0.5458



Karmaşıklık Matrisi



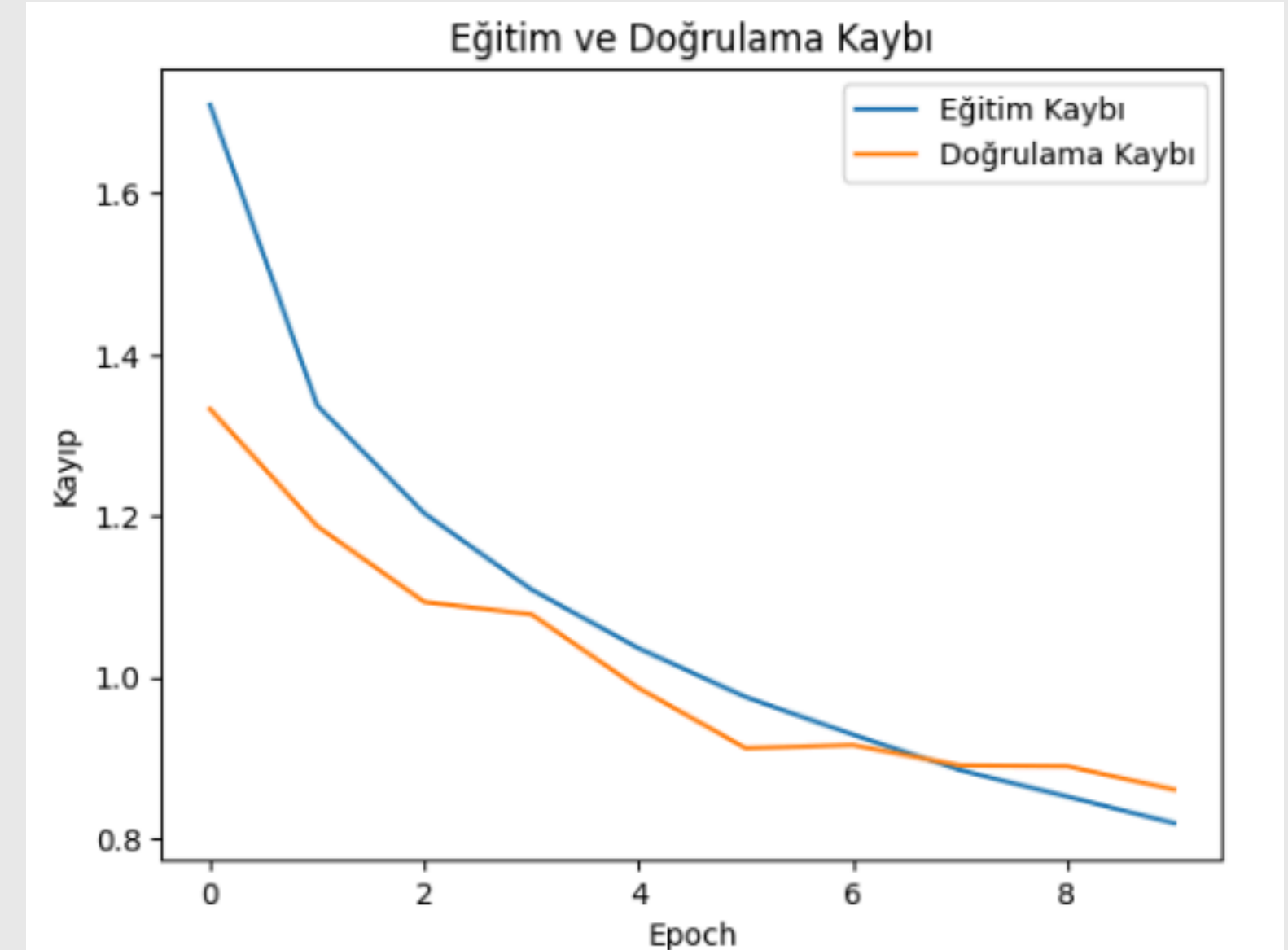
# CNN Eğitimi ve Değerlendirme

Veri seti yüklendi ve %70 oranında küçültüldü -eğitimin daha kısa sürmesi için-. Yapılan veri ön işlemlerinin ardından CNN modeli oluşturulup eğitildi. Elde edilen sonuçlar:

loss: 0.8605

accuracy: 0.7064

Test accuracy: 0.7064



Eğitim ve Doğrulama Kaybı Grafiği

# Genel Değerlendirme

## 1.) Performans Karşılaştırması:

- Hızlı Sonuç Verme: CNN modeli, SVC modeline göre daha hızlı sonuçlar üretti. Özellikle büyük veri kümelerinde, derin öğrenme modelleri genellikle daha hızlı eğitilir ve tahmin yapar.
- Başarı Oranı: CNN modeliniz, CIFAR-10 veri kümesinde daha yüksek bir doğruluk elde etti. Bu, derin öğrenme modelinin daha karmaşık ve belirgin özellikler öğrenme yeteneğini vurgulayabilir.

## 2.) Model Karmaşıklığı ve Esneklik:

- Model Karmaşıklığı: CNN modeli, çok katmanlı evrişim ve tam bağlantı katmanlarıyla daha karmaşık bir yapıya sahiptir. Bu, görüntü tanıma gibi karmaşık görevler için daha uygun olabilir.
- Esneklik: Derin öğrenme modelleri, veri kümesindeki desenleri daha doğru bir şekilde öğrenebilir ve genelleme yetenekleri daha yüksektir.

# Genel Değerlendirme

## 3.) Model Seçimi ve Uygulama Alanı:

- Uygulama Alanı: CIFAR-10 gibi görüntü sınıflandırma problemlerinde, derin öğrenme modelleri genellikle daha iyi performans gösterir. Ancak, veri setinizin boyutu, doğası ve spesifik gereksinimleri modele göre değişebilir.
- Model Seçimi: Performans ve hız arasında bir denge bulunması gerekebilir. SVC gibi geleneksel makine öğrenimi modelleri daha az hesaplama gücü gerektirebilir ancak derin öğrenme modelleri daha yüksek doğruluk sağlayabilir.

---

# Sonuç

Sonuç ve Öneriler:

- Öneriler: CIFAR-10 veya benzeri görevler için derin öğrenme modellerinin kullanılması, genellikle daha iyi sonuçlar verebilir. Ancak, modelinizi seçerken veri setinizin boyutu, kullanılabilir hesaplama gücü ve gereksinimlerini göz önünde bulundurmalısınız.
  - Gelecek Çalışmalar: Modelinizi daha da iyileştirmek için farklı hiperparametre ayarlamaları, model mimarisi değişiklikleri veya daha büyük veri kümeleri üzerinde denemeler yapılabilir.
-



---

# Kaynaklar

- <https://bartubozkurt35.medium.com/derin-sinir-ağları-için-aktivasyon-fonksiyonları-ve-türleri-787f07c233a8>
  - <https://www.tensorflow.org/tutorials/images/cnn?hl=tr>
  - <https://mfatihto.medium.com/support-vector-machine-algoritması-makine-öğrenmesi-8020176898d8>
  - <https://medium.com/deep-learning-turkiye/model-performansını-değerlendirmek-metrikler-cb6568705b1>
-

# Thank you!

---