

AYGAZ AI BOOTCAMP

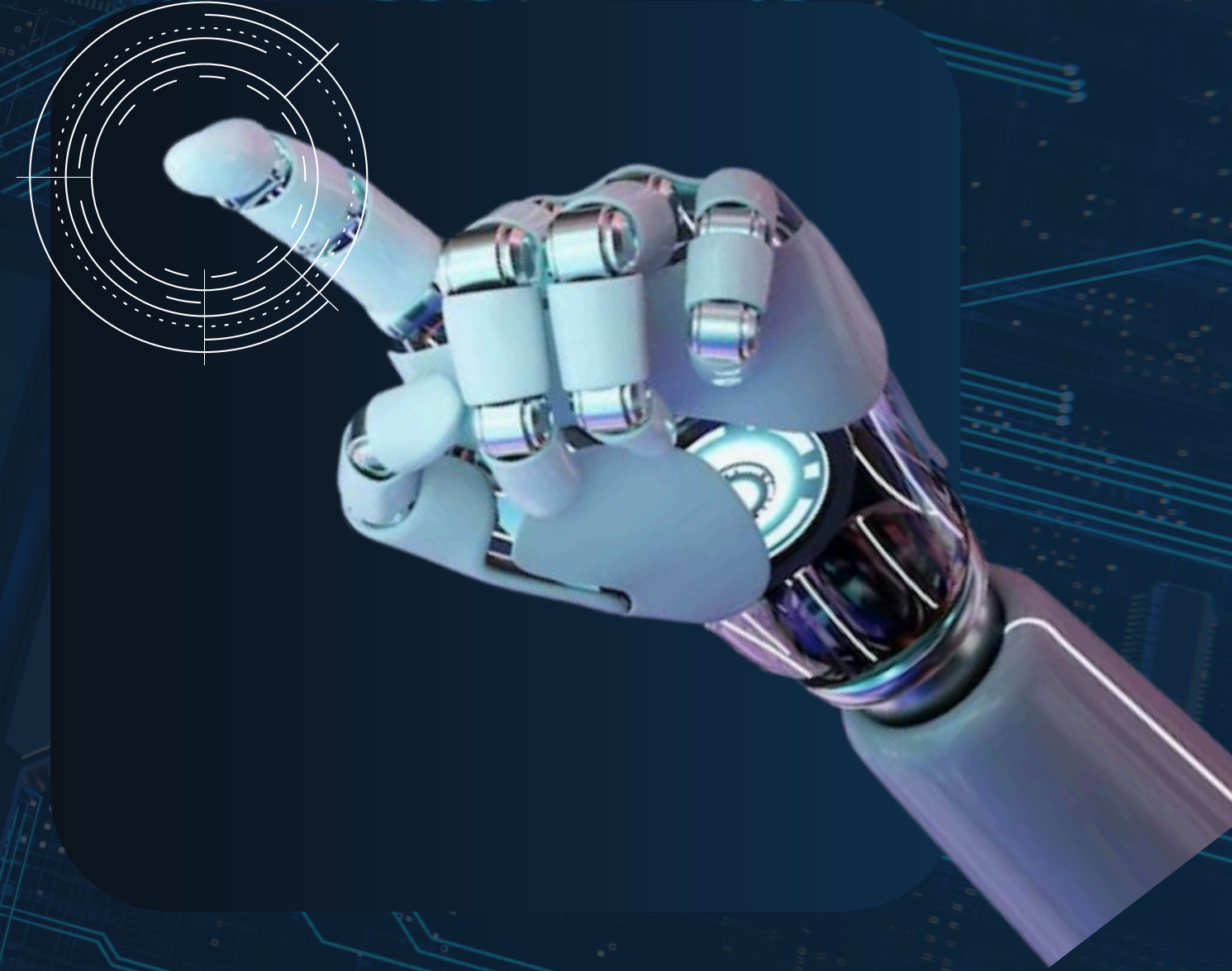
# Cifar10 Veri Seti ile Görüntü Sınıflandırma Projesi

BULUT ÇAĞRI ÇİFTÇİ



# Proje Amacı

- Görüntü verileri üzerinde makine öğrenmesi ve derin öğrenme tekniklerini kullanarak sınıflandırma yapmak.
- Cifar10 veri seti üzerinde özel bir CNN modeli geliştirilerek sınıflandırma başarısını değerlendirmek.





# Veri Seti ve Veri Ön İşleme

## Veri Seti

- Cifar10 Veri Seti:
  - İçeriği: 10 farklı sınıfta 60,000 renkli görüntü
  - Görüntü boyutları: 32x32 piksel
  - Eğitim seti: 50,000 görüntü
  - Test seti: 10,000 görüntü

## Veri Görselleştirme

- Örnek Görsellerin Görselleştirilmesi:
  - Veri setindeki 10 farklı sınıftan örnek görsellerin gösterimi

## Veri Ön İşleme Adımları

- Veri Bölme:
  - $X_{train}$ ,  $X_{test}$ ,  $y_{train}$ ,  $y_{test}$  olarak veri setini ayırma
- Veri Normalize Etme:
  - Görüntüleri  $[0,1]$  aralığına normalize etme
- Etiketlerin One-Hot Encoding ile Dönüştürülmesi:
  - $y_{train\_encoded}$  ve  $y_{test\_encoded}$  oluşturma



# Model Oluřturma ve Eđitim

## Kullanılan Yöntemler

Özel Convolutional Neural Network (CNN) Modeli:

- Model Mimarisi: 3 Convolutional Katman, 3 MaxPooling Katman, Flatten Katmanı, 2 Dense Katman, Dropout Katmanı

## Model Derleme

- Optimizer: Adam
- Loss Function: CategoricalCrossentropy
- Deđerlendirme Metrikleri: Accuracy, Precision, Recall, F1 Score

## Model Eđitimi

- Epoch Sayısı: 15
- Validation Data: X\_test ve y\_test kullanılarak modelin dođruluđunu deđerlendirme



# Model Özeti

Katman (Layer)	Çıkış Şekli (Output Shape)	Parametre Sayısı (Param #)
Conv2D	(32, 32, 32)	896
MaxPooling2D	(16, 16, 32)	0
Conv2D	(16, 16, 64)	18,496
MaxPooling2D	(8, 8, 64)	0
Conv2D	(8, 8, 128)	73,856
MaxPooling2D	(4, 4, 128)	0
Flatten	(2048)	0
Dense	(128)	262,272
Dropout	(128)	0
Dense	(10)	1,290
Toplam Parametre Sayısı	357,810	



# Katman Detayları

## 1.Girdi Katmanı

- Input Shape: (32, 32, 3) - CIFAR-10 veri setindeki görüntülerin boyutu 32x32 piksel ve 3 renk kanalına sahiptir.

## 2.Convolutional Katmanı

- Conv2D: 32 filtre, 3x3 kernel boyutu, 'relu' aktivasyon fonksiyonu, 'same' padding.
- Açıklama: Görüntülerin uzaysal özelliklerini öğrenir ve 32 filtre uygular.

## 3. MaxPooling Katmanı

- MaxPooling2D: 2x2 havuzlama boyutu.
- Açıklama: Özellik haritalarının boyutunu yarıya indirerek bilgi kaybını minimize ederken hesaplama maliyetini azaltır.

## 4. Convolutional Katmanı

- Conv2D: 64 filtre, 3x3 kernel boyutu, 'relu' aktivasyon fonksiyonu, 'same' padding.
- Açıklama: Daha derin özellikleri öğrenir ve 64 filtre uygular.

## 5. MaxPooling Katmanı

- MaxPooling2D: 2x2 havuzlama boyutu.
- Açıklama: Özellik haritalarının boyutunu yeniden yarıya indirir.

## 6. Convolutional Katmanı

- Conv2D: 128 filtre, 3x3 kernel boyutu, 'relu' aktivasyon fonksiyonu, 'same' padding.
- Açıklama: Daha karmaşık özellikleri öğrenir ve 128 filtre uygular.

## 7.MaxPooling Katmanı

- MaxPooling2D: 2x2 havuzlama boyutu.
- Açıklama: Özellik haritalarının boyutunu tekrar yarıya indirir.

## 8.Flatten Katmanı

- Flatten: Çok boyutlu özellik haritalarını tek boyutlu vektöre dönüştürür.
- Açıklama: Yoğun bağlantılı katmanlara geçiş için veri hazırlığı yapar.

## 9.Dense Katmanı

- Dense: 128 nöron, 'relu' aktivasyon fonksiyonu.
- Açıklama: Yüksek seviyeli özellikleri öğrenir ve yoğun bağlantılıdır.

## 10.Dropout Katmanı

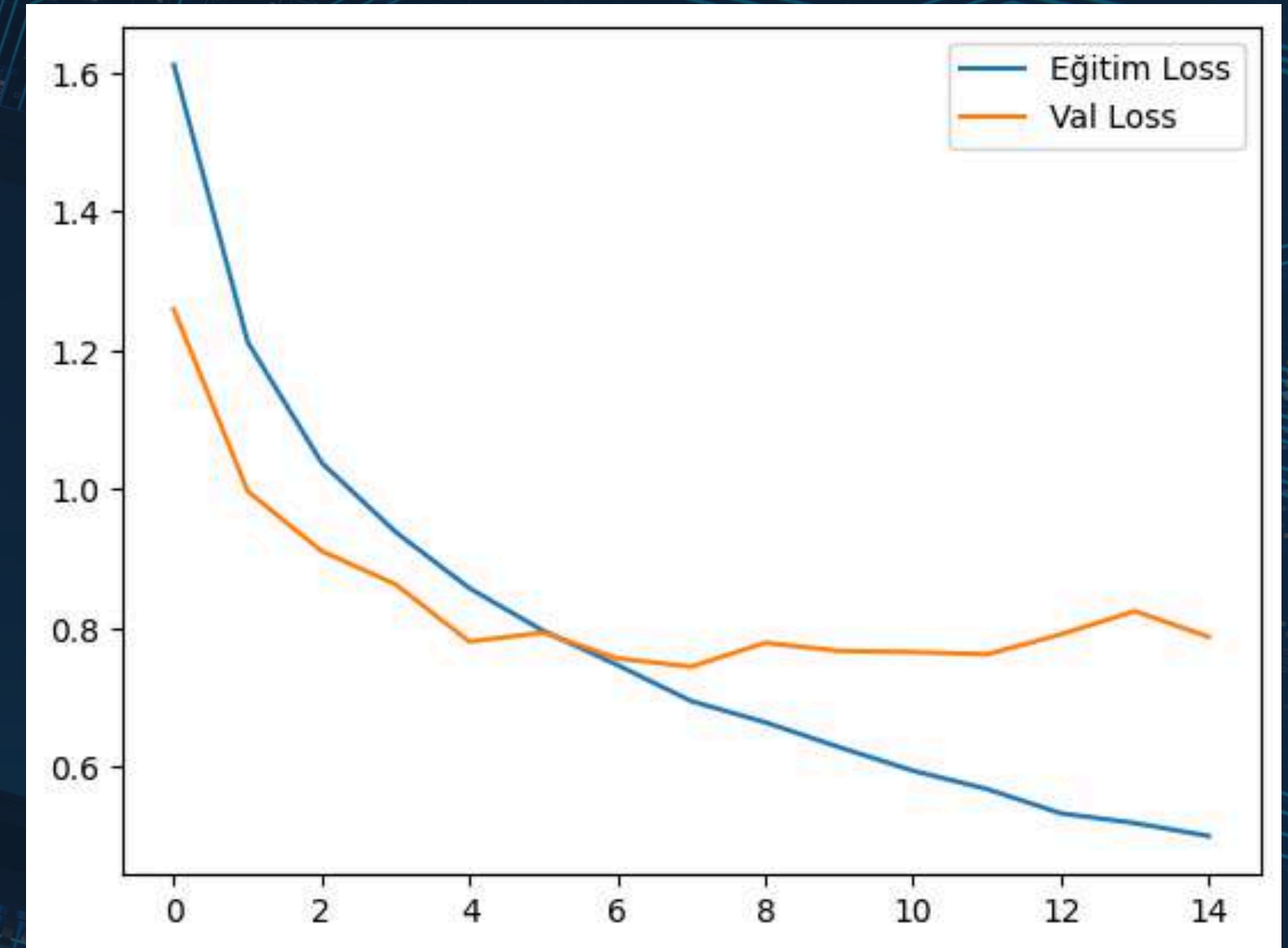
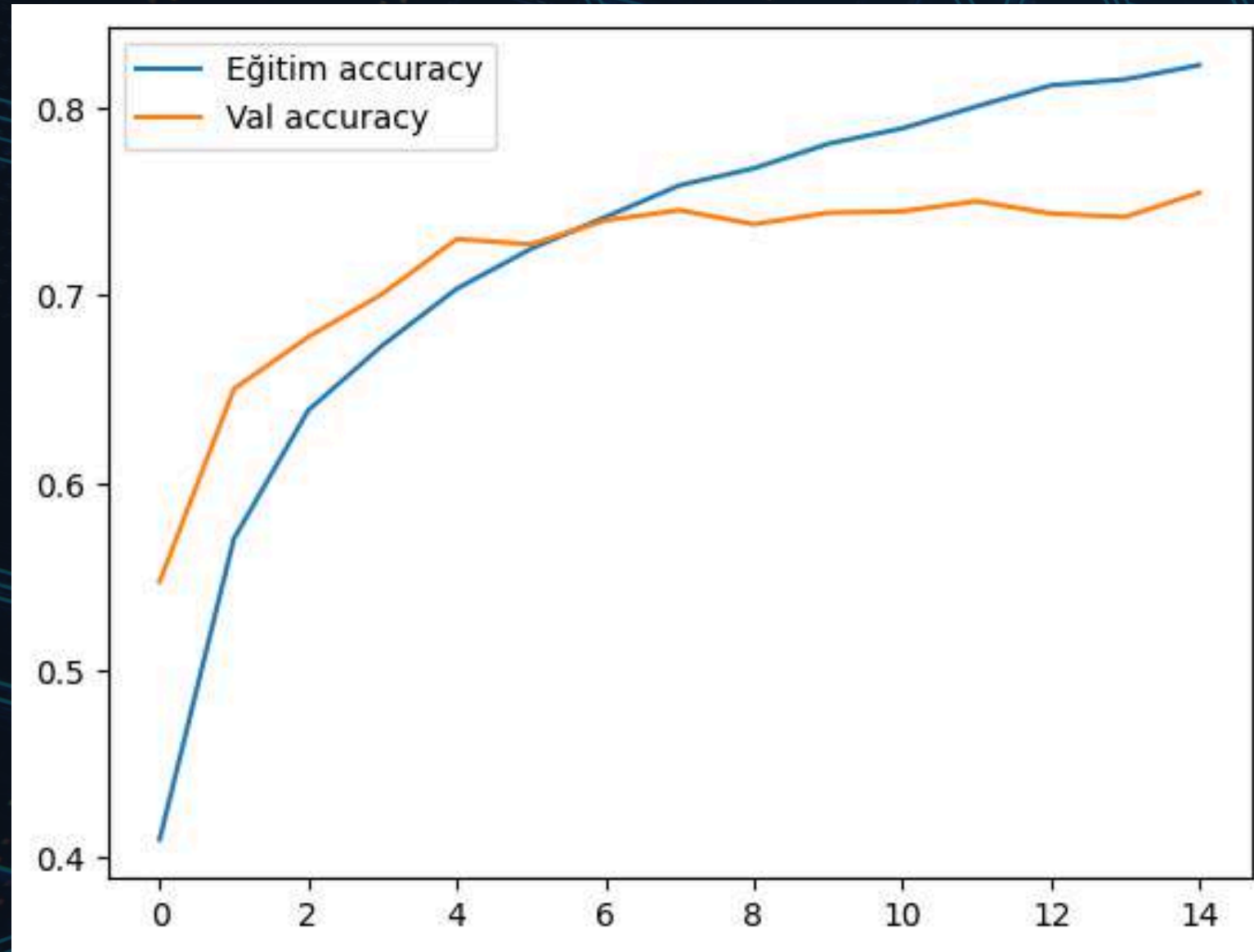
- Dropout: %50 rastgele nöronları devre dışı bırakır.
- Açıklama: Aşırı uyumlamayı (overfitting) önlemeye yardımcı olur.

## 11.Çıktı Katmanı

- Dense: 10 nöron, 'softmax' aktivasyon fonksiyonu.
- Açıklama: CIFAR-10 veri setindeki 10 sınıfı tahmin eder.

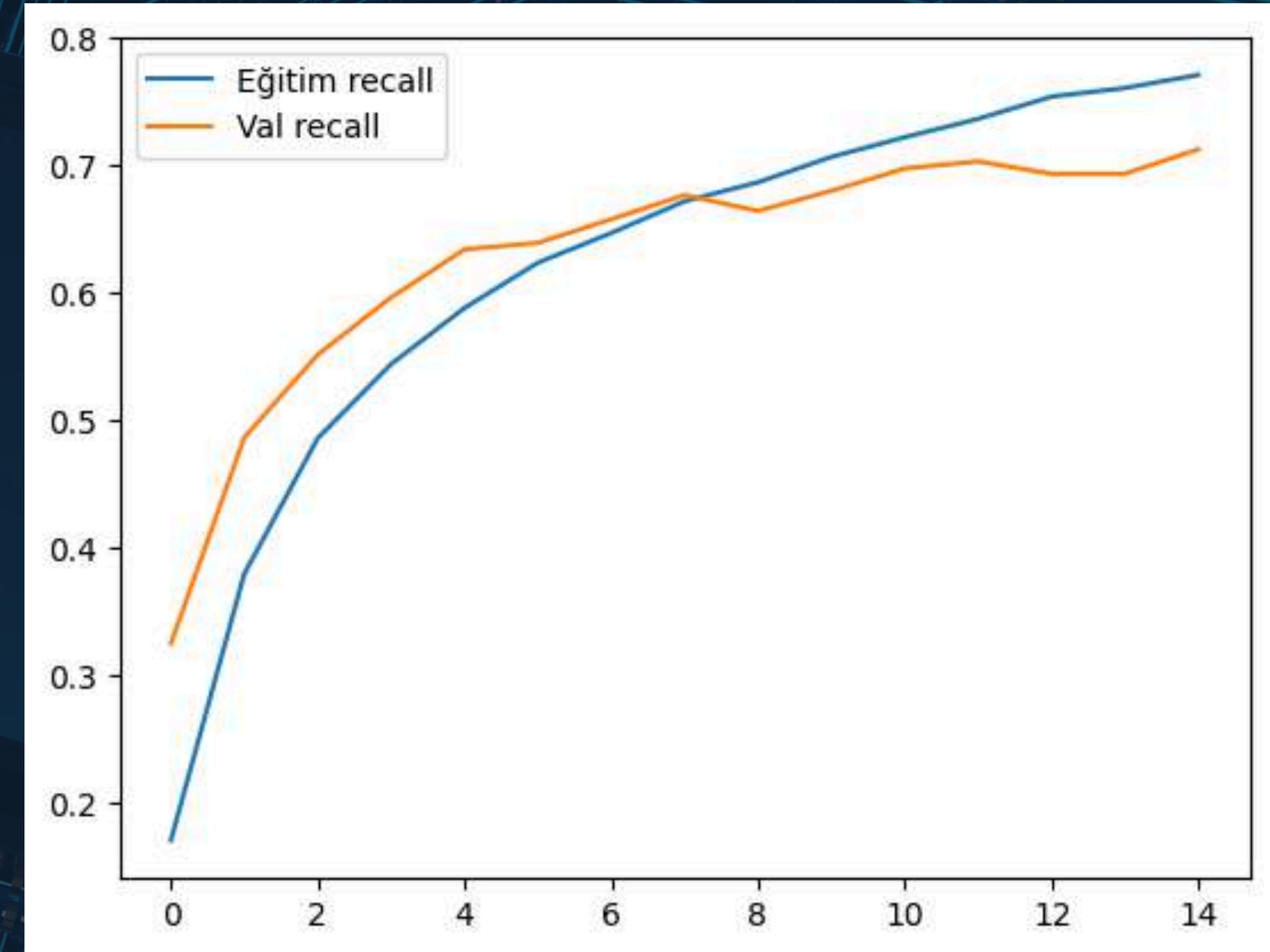
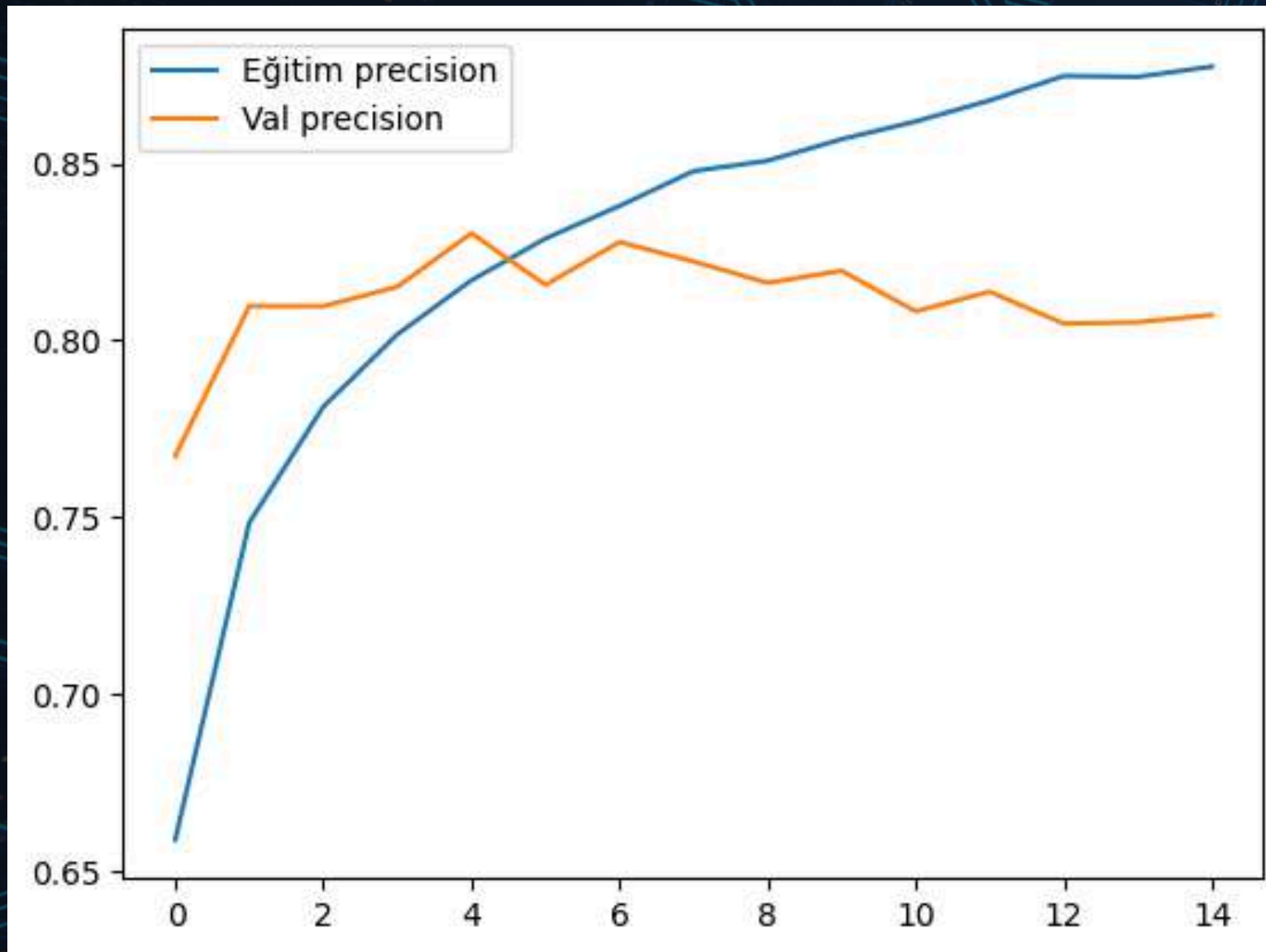


# Accuracy ve Loss



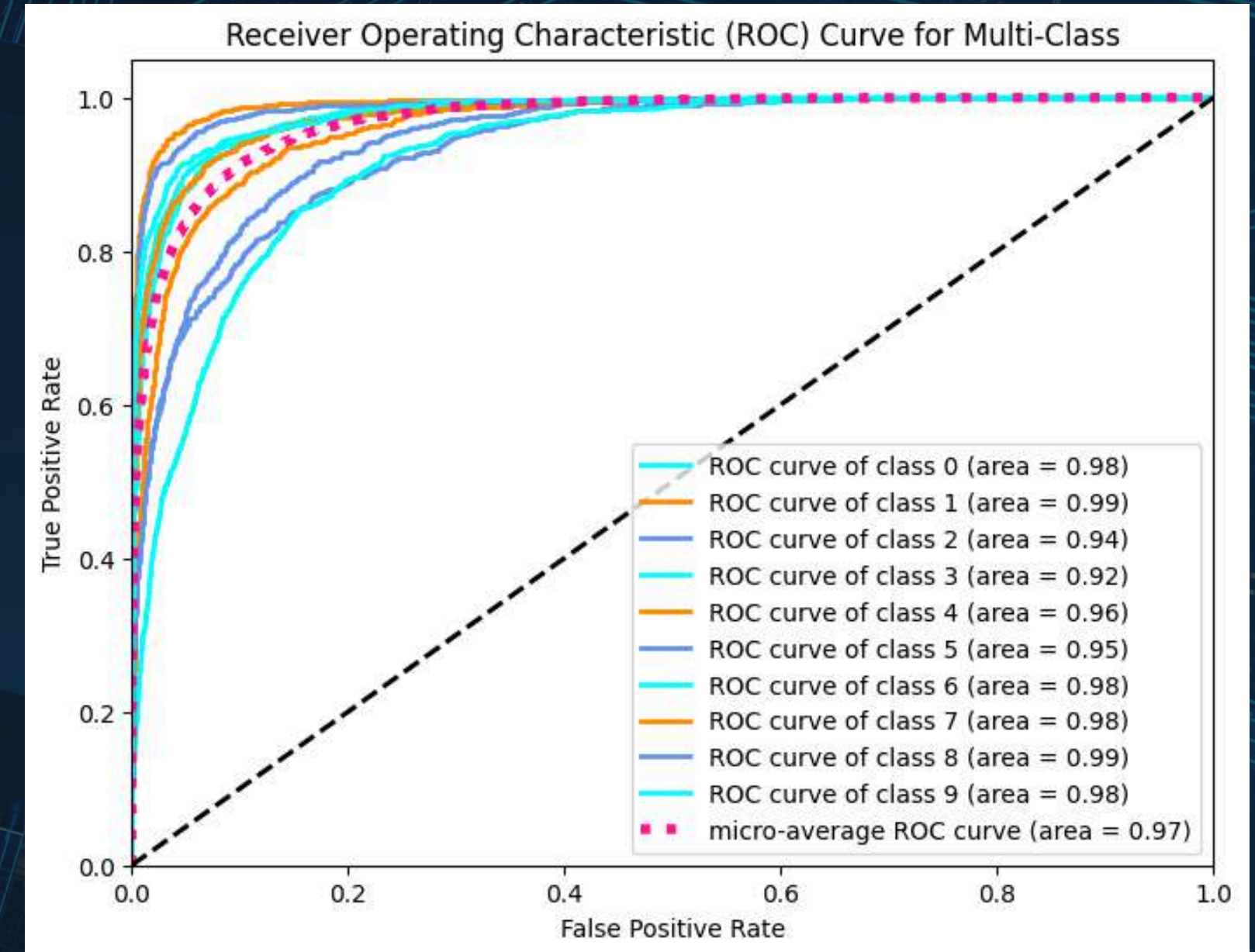
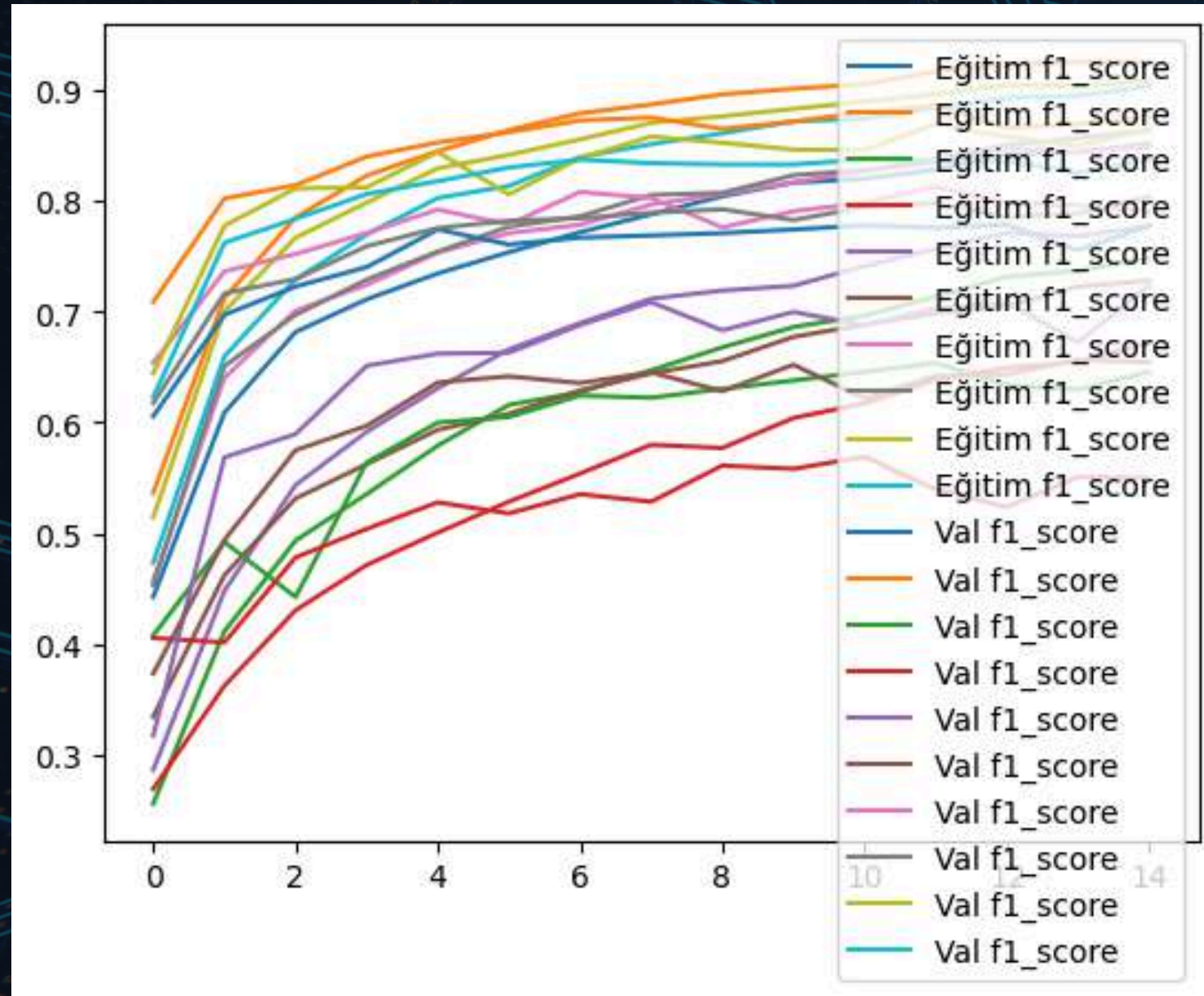


# Precision ve Recall





# F1 Score ve ROC Eğrisi





# Model Performansı ve Karmaşıklık Matrisi

## Model Performansı

Modelin eğitimi sırasında elde edilen sonuçlar, modelin eğitim sürecinde belirgin bir iyileşme gösterdiğini ortaya koymaktadır. İlk epoch'ta modelin eğitim doğruluğu %40.97 iken, 15. epoch sonunda bu değer %82.28'e yükselmiştir. Doğrulama doğruluğu ise başlangıçta %54.73 iken, 15. epoch sonunda %75.47'ye ulaşmıştır. Bu ilerleme, modelin eğitim sürecinde veriyi öğrenme ve genelleme yeteneğinde ciddi bir artış olduğunu göstermektedir.

## Başarı ve Zayıf Noktalar

### Başarı Noktaları:

Modelin doğruluğu hem eğitim hem de doğrulama setlerinde sürekli olarak artmıştır. Precision ve recall değerlerinde de benzer bir iyileşme gözlemlenmiştir, bu da modelin hem doğru pozitifleri hem de negatifleri başarılı bir şekilde sınıflandırabildiğini göstermektedir.

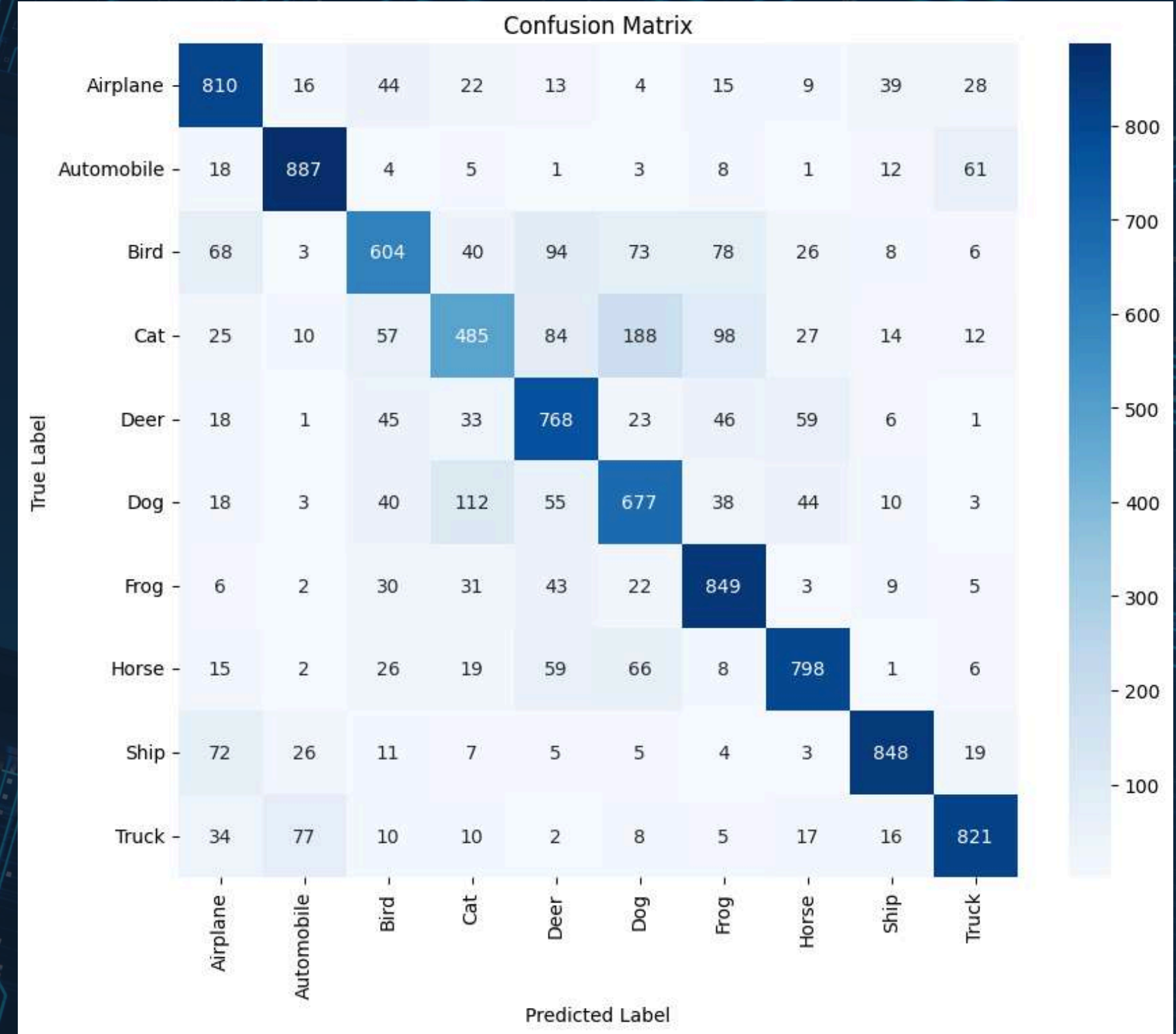
F1 skoru, dengeli bir performans metrik olduğu için, modelin eğitim sürecindeki gelişimini iyi bir şekilde yansıtmaktadır.

### Zayıf Noktalar:

Modelin doğrulama kaybı, eğitim kaybına göre daha yavaş bir düşüş göstermiştir ve bazı epoch'larda artış göstermiştir. Bu durum, modelin overfitting olabileceğine işaret etmektedir.

Son epoch'ta doğrulama kaybı hala belirgin bir seviyede, bu da modelin tam anlamıyla genelleme yapma yeteneğinde bazı eksiklikler olabileceğini göstermektedir.

Özellikle bazı sınıflarda modelin performansı diğer sınıflara göre daha düşük olabilir. Bu durum, özellikle karmaşık ya da birbirine benzeyen sınıfların ayrıştırılmasında zorluk yaşanabileceğini göstermektedir.





# Sonuçlar

## Geliştirme Önerileri

### Epoch Sayısını Artırma:

- Daha fazla epoch ile eğitim, modelin daha iyi öğrenme ve genelleme yapmasına olanak tanıyabilir. Ancak, overfitting'i önlemek için erken durdurma (early stopping) tekniği kullanılabilir.

### Veri Artırma Teknikleri:

- Farklı veri artırma teknikleri (data augmentation) kullanarak eğitim verisinin çeşitliliğini artırmak, modelin genelleme yeteneğini geliştirebilir. Örneğin, rastgele döndürme, kaydırma, ölçekleme, yatay çevirme gibi yöntemler kullanılabilir.

### Daha Karmaşık Model Mimarileri:

- Daha derin ve karmaşık model mimarileri kullanarak model performansını artırmak mümkün olabilir. Örneğin, ResNet, VGG gibi önceden eğitilmiş modellerden transfer öğrenme (transfer learning) yapılabilir.
- Modelde kullanılan katman sayısını ve nöron sayısını artırmak, modelin daha fazla özelliği öğrenmesine yardımcı olabilir.

### Düzenleme Teknikleri:

- L2 regularization, dropout gibi düzenleme teknikleri kullanarak overfitting'i azaltmak ve modelin genelleme yeteneğini artırmak mümkündür.

## Sonuç

Projenin Genel Değerlendirmesi: CIFAR-10 veri seti üzerinde özel bir CNN modeli kullanarak yapılan bu görüntü sınıflandırma projesi, modelin eğitim sürecinde belirgin bir performans artışı göstermiştir. Eğitim ve doğrulama doğruluğu zamanla artmış ve model, başlangıçtaki performansına göre büyük bir iyileşme göstermiştir. Ancak, modelin overfitting'e eğilimli olduğu ve bazı sınıflarda zorluklar yaşadığı gözlemlenmiştir. Bu nedenle, daha karmaşık model mimarileri, veri artırma teknikleri ve düzenleme yöntemleri kullanarak modelin performansını daha da artırmak mümkündür.