



**Mühendislik Fakültesi  
Yazılım Mühendisliği Bölümü**

**FET312 - Derin Öğrenme**

Final Proje Raporu

**Trafik İşaretlerinin Evrişimli Sinir Ağları (CNN) ile  
Sınıflandırılması**

**Ekip Bilgileri**

Adı Soyadı	Öğrenci No	E-posta
Ramazan Bozkurt	22040301027	ramazanbozkurt@stu.topkapi.edu.tr
Ömer Utlu Aktemur	22040301043	omerutkuaktemur@stu.topkapi.edu.tr

**GitHub Repo Bağlantısı:**

[https://github.com/RamazanBozkurrtt/DevTeam177\\_GTSRB-Traffic-Sign-Classification-CNN-Final](https://github.com/RamazanBozkurrtt/DevTeam177_GTSRB-Traffic-Sign-Classification-CNN-Final)

## İçindekiler

<b>1 Problem Tanımı ve Motivasyon.....</b>	<b>3</b>
1.1 İş/Bilimsel Soru.....	3
1.2 Görev Türü.....	3
1.3 Hedef Değişkenler.....	3
1.4 Başarı Kriterleri.....	3
<b>2 Proje Yönetimi .....</b>	<b>3</b>
2.1 Zaman Çizelgesi (Milestones).....	3
2.2 Roller ve Sorumluluklar .....	4
<b>3 İlgili Çalışmalar (Literatür Taraması).....</b>	<b>4</b>
<b>4 Veri Açıklaması ve Yönetimi.....</b>	<b>4</b>
4.1 Veri Kümesi (Dataset Description).....	4
4.2 Boyut ve Şema.....	4
<b>5 Yöntemler ve Mimari.....</b>	<b>4</b>
5.1 Model A Tasarımı (Ramazan Bozkurt).....	5
5.2 Model B Tasarımı (Ömer Utku Aktemur).....	5
<b>6 Deney Tasarımı ve Sonuçlar.....</b>	<b>5</b>
6.1 Deney Kurulumu.....	5
6.2 Performans Karşılaştırması .....	6
<b>7 Kullanılan Araçlar.....</b>	<b>6</b>
<b>8 Kaynaklar.....</b>	<b>6</b>

## 1 Problem Tanımı ve Motivasyon

### 1.1 İş/Bilimsel Soru

Otonom sürüş sistemlerinin ve gelişmiş sürücü destek sistemlerinin (ADAS) en kritik bileşenlerinden biri, çevresel farkındalıktır. Bu proje kapsamında, araçların seyir halindeyken karşılaşabileceği trafik işaretlerinin (hız sınırları, dur, girilmez vb.) kameraların görüntüleri üzerinden otomatik olarak algılanması ve sınıflandırılması problemi ele alınmıştır. Temel bilimsel soru şudur: "*Sınırlı işlem gücüne sahip araç içi sistemlerde çalışabilecek, yüksek doğruluk oranına sahip, farklı derinliklerdeki özgün CNN mimarileri nasıl tasarlanabilir?*"

### 1.2 Görev Türü

Bu proje, bilgisayarlı görü (computer vision) alanında çok sınıfı bir **Sınıflandırma (Classification)** problemidir. Girdi olarak alınan trafik işaretleri görüntüler, 43 farklı kategoriden (sınıftan) birine atanmaktadır.

### 1.3 Hedef Değişkenler

- **Girdi:** 64x64 piksel boyutuna indirgenmiş, 3 kanallı (RGB) trafik işaretleri görüntüler.
- **Çıktı (Hedef):** 43 farklı trafik işaretinin sınıfından biri (Örn: Class 0: 20 km/s hız sınırı, Class 14: Dur işaret). Hedef değişken kategorik ve nominaldir.

### 1.4 Başarı Kriterleri

Projenin base model aşaması için belirlenen başarı kriterleri şunlardır:

- **Doğruluk (Accuracy):** Test setinde  $\geq 90\%$  doğruluk oranı (Gelişmiş modeller için hedef  $\geq 96\%$ ).
- **Loss (Kayıp):** Eğitim ve doğrulama kaybı arasında aşırı uyum (overfitting) olmaması.
- **F1-Score:** Dengesiz veri dağılımına karşı ağırlıklı F1 skorunun tatmin edici düzeyde olması.

## 2 Proje Yönetimi

### 2.1 Zaman Çizelgesi (Milestones)

- **1. Hafta :** GTSRB veri setinin incelenmesi ve görev dağılımı.
- **2. Hafta :** Veri ön işleme ve DataLoader yapılarının kurulması.
- **3-4. Hafta :** Özgün Base Model (SimpleCNN ve SimpleBatchNormCNN) tasarımları ve eğitimi.
- **5-6. Hafta (Final Aşaması):** ResNet-18 ve DenseNet-121 mimarilerinin transfer learning ile entegrasyonu, hiperparametre optimizasyonu ve final raporunun hazırlanması.

## 2.2 Roller ve Sorumluluklar

- **Ramazan Bozkurt:** Veri setinin analizi ve 'SimpleBatchNormCNN' modelinin geliştirilmesi. Bu modelde Batch Normalization katmanlarının eğitim stabilitesine etkisi üzerine odaklanılmıştır. Ayrıca karmaşıklık matrisi görselleştirmeleri yapılmıştır. Daha gelişmiş modeller ile basit modellerin karşılaştırması yapılması hedeflenmiştir. Dolayısıyla DenseNet-121 ve ResNet-18 advanced modelleri ile veri seti eğitilip ilk aşamada yapılan modellerle karşılaştırılmasına odaklanılmıştır.
- **Ömer Utku Aktemur:** 'SimpleCNN' modelinin geliştirilmesi. Daha az parametreye sahip, hesaplama maliyeti düşük ancak etkili bir model tasarımları (2 Konvolüsyon katmanı) üzerine çalışılmıştır. Veri düzenleme ve yükleme (DataLoader) modüllerinin optimizasyonu sağlanmıştır.

## 3 İlgili Çalışmalar (Literatür Taraması)

Trafik işareteti tanıma konusunda literatürde "German Traffic Sign Recognition Benchmark (GTSRB)" veri seti üzerinde yapılan çalışmalar temel alınmıştır.

1. **Cireşan et al. (2012):** "Multi-column Deep Neural Networks for Image Classification" çalışmasında, birden fazla DNN'in birleştirilmesiyle insan üstü performans ( )
2. **Sermanet ve LeCun (2011):** "Traffic Sign Recognition with Multi-Scale Convolutional Networks" makalesinde çok ölçekli özelliklerin beslenmesi yöntemi kullanılmıştır.
3. **He et al. (2016):** ResNet mimarisi ile derin ağlardaki kaybolan gradyan problemi çözülmüştür.
4. **Huang et al. (2017):** DenseNet ile özellik iletimi ve parametre verimliliği artırılmıştır.

**Farklılaşma Noktası:** Grup üyeleri olarak, hazır mimariler (ResNet, VGG) kullanmak yerine, problemin doğasına uygun, işlem gücü kısıtlı cihazlarda çalışabilecek iki farklı "Custom CNN" mimarisi tasarlayarak performanslarını kıyasladık.

## 4 Veri Açıklaması ve Yönetimi

### 4.1 Veri Kümesi (Dataset Description)

Projede **GTSRB (German Traffic Sign Recognition Benchmark)** veri seti kullanılmıştır. Veriler, Almanya'daki gerçek trafik koşullarında çekilmiş görüntülerden oluşmaktadır.

- **Kaynak:** Institut für Neuroinformatik, Ruhr-Universität Bochum.
- **Lisans:** Akademik kullanım için açiktır.

### 4.2 Boyut ve Şema

- **Sınıf Sayısı:** 43 farklı sınıf.
- **Veri Miktarı:** 39.000 Eğitim, 12.000 Test görüntüsü.
- **Ön İşleme:** Orijinalde farklı boyutlarda olan görüntüler 64x64 piksel boyutuna sabitlenmiş ve kanal bazlı normalizasyon (Mean: 0.5, Std: 0.5) uygulanmıştır. Gelişmiş modellerin eğitiminde daha iyi sonuçlara ulaşmak için veri çöklama, rastgele döndürme ( $15^\circ$ ) ve renk titremesi uygulanmıştır.

## 5 Yöntemler ve Mimari

Her grup üyesi problemi çözmek için farklı bir Base Model yaklaşımı benimsemiştir. Aşağıda her bir modelin Evrişim (Convolution) ve Tam Bağlantılı (Fully Connected) katman yapıları detaylandırılmıştır.

### 5.1 Model A Tasarımı (Ramazan Bozkurt)

Tarafimdan geliştirilen **SimpleBatchNormCNN** modeli, kararlı bir eğitim süreci sağlamak amacıyla 3 konvolüsyon katmanına ve Batch Normalization yapısına sahiptir.

- **Evrişim Katmanları (Feature Extraction):**

1. Conv2d (32 filtre, 3x3) → BatchNorm → ReLU → MaxPool(2x2)
2. Conv2d (64 filtre, 3x3) → BatchNorm → ReLU → MaxPool(2x2)
3. Conv2d (128 filtre, 3x3) → BatchNorm → ReLU → MaxPool(2x2)

- **Tam Bağlantılı Katmanlar (Fully Connected Layers):**

1. Flatten (Düzleştirme)
2. Linear (128 nöron) → BatchNorm1d → ReLU
3. Dropout (0.5)
4. Linear (43 sınıf - Çıktı Katmanı)

- **Özellik:** Model, hem konvolüsyon hem de tam bağlantılı katmanlarda Batch Normalization kullanarak "Internal Covariate Shift" problemini azaltmayı hedefler.

- **ResNet-18**

1. **Mimari:** ImageNet üzerinde önceden eğitilmiş (pre-trained) ağırlıklara sahip ResNet-18 modeli tercih edilmiştir.
2. **Özellik:** Derin ağlarda gradyan akışını optimize eden "Residual Blocks" yapısı sayesinde daha stabil ve yüksek performanslı bir öğrenme süreci hedeflenmiştir.
3. **Optimizasyon:** Son katman 43 sınıfa güncellenmiş ve aşırı öğrenmeyi engellemek adına Dropout (0.4) eklenmiştir.

- **DenseNet-121**

1. **Mimari:** Katmanlar arası yoğun bağlantı mimarisine sahip olan DenseNet-121 modeli kullanılmıştır.
2. **Özellik:** Özellik haritalarının (feature maps) katmanlar arasında yeniden kullanılmasını sağlayan yapısı sayesinde parametre verimliliği maksimize edilmiştir.
3. **Optimizasyon:** AdamW optimizasyon algoritması ve **1e-4** öğrenme oranı (learning rate) kullanılarak hiperparametre ince ayarı yapılmıştır.

## 5.2 SimpleCNN Modeli Tasarımı (Ömer Utku Aktemur)

Grup üyesi Ömer Utku tarafından geliştirilen **SimpleCNN** modeli, daha sade ve hızlı eğitilen bir yapı hedeflemiştir.

- **Evrişim Katmanları (Feature Extraction):**

1. Conv2d (16 filtre, 3x3) → ReLU → MaxPool(2x2)
2. Conv2d (32 filtre, 3x3) → ReLU → MaxPool(2x2) • **Tam Bağlantılı Katmanlar**

- **(Fully Connected Layers):**

1. Flatten (Düzleştirme)
2. Linear (128 nöron) → ReLU
3. Linear (43 sınıf - Çıktı Katmanı)

- **Özellik:** Batch Normalization veya Dropout kullanılmamış, modelin ham özellik çıkarma yeteneğine ve hızına odaklanılmıştır.

## 5.3 Efficient-B0 Modeli Tasarımı (Ömer Utku Aktemur)

Grup üyesi Ömer Utku tarafından geliştirilen EfficientNet-B0 modeli, Google tarafından geliştirilen modern "Compound Scaling" (Bileşik Ölçeklendirme) teknolojisini kullanarak maksimum parametre verimliliği ve yüksek doğruluk hedeflemiştir.

- **Evrişim Katmanları (Feature Extraction):**

1. Stem (Giriş): Conv2d (3x3) → BatchNormalization → Swish Aktivasyonu
2. MBConv Blokları: 7 Aşamalı "Mobile Inverted Bottleneck" blokları.
  - Her blokta: Derinlemesine Ayrılabilir Evrişim (Depthwise Separable Conv) + Squeeze-and-Excitation (SE) modülleri kullanılarak özellikler sıkıştırılıp genişletilir.
3. Feature Mix: Conv2d (1x1) → BatchNormalization → Swish

- **Tam Bağlantılı Katmanlar (Fully Connected Layers):**

1. Global Average Pooling: (Klasik düzleştirme yerine uzamsal veriyi ortalamaya indirger, parametre sayısını düşürür).
2. Dropout: (Oran: 0.2) → Aşırı öğrenmeyi (Overfitting) engellemek için rastgele nöron kapatma.
3. Linear: (43 sınıf - Çıktı Katmanı).

- **Özellik:** Geleneksel ReLU yerine daha modern olan Swish aktivasyon fonksiyonu kullanılmış ve Transfer Learning (ImageNet ağırlıkları) ile modelin genel görme yeteneğinden faydalانılmıştır.

## 5.4 GoogLeNet Modeli Tasarımı (Ömer Utku Aktemur)

Grup üyesi Ömer Utku tarafından geliştirilen GoogLeNet (Inception v1) modeli, klasik ardışık mimarilerin aksine "Multi-Branch" (Çok Dallı) yapıyı kullanarak derinlik ve genişlik dengesini hedeflemiştir.

- **Evrişim Katmanları (Feature Extraction):**

1. Giriş Bloğu: Conv2d (7x7) → MaxPool → Conv2d (3x3) → MaxPool.
  2. Inception Modülleri: (9 adet ardışık modül).
    - Her modülün içinde Paralel İşleme: Aynı girdiye aynı anda 1x1, 3x3 ve 5x5 filtreler uygulanır ve sonuçlar birleştirilir (Concatenation).
  3. Yardımcı Çıkışlar (Auxiliary Classifiers): Modelin orta katmanlarında eğitim kaybını (loss) dengelemek için iki adet ek çıkış bloğu bulunur.
- **Tam Bağlantılı Katmanlar (Fully Connected Layers):**
    1. Global Average Pooling: (7x7 boyutundaki öznitelikleri 1x1'e indirger).
    2. Dropout: (Oran: 0.2 veya 0.4) → Ezberlemeyi önler.
    3. Linear: (43 sınıf - Çıktı Katmanı).
  - **Özellik:** Model, "Inception" mimarisi sayesinde görüntüdeki hem çok küçük (1x1) hem de büyük (5x5) detayları aynı anda yakalar. Full Fine-Tuning yapılarak tüm katmanlar trafik levhalarına göre özelleştirilmiştir.

## 6 Deney Tasarımı ve Sonuçlar

### 6.1 Deney Kurulumu

Her iki basit model de aynı veri seti üzerinde, benzer hiper parametrelerle (Epoch: 15, Batch: 32/64, Opt: Adam) eğitilmiştir.

Gelişmiş modeller olan DenseNet-121 ve ResNet-18 modellerde **3e-4 (ResNet)** ve **1e-4 (DenseNet)** öğrenme oranları kullanılmıştır.

### 6.2 Performans Karşılaştırması

Aşağıdaki tablo, grup üyelerinin geliştirdiği base modellerin test seti üzerindeki performanslarını özetlemektedir:

**Yorum:** Ramazan'ın geliştirdiği model (Model A), Batch Normalization ve ekstra derinliği sayesinde %98 gibi çok yüksek bir genel doğruluğa ulaşmıştır. Ömer Utku'nun modeli (Model B) ise daha basit yapısına rağmen %94 gibi oldukça başarılı bir sonuç elde etmiştir. Bu, veri setinin kalitesi ve "shallow" mimarilerin bile bu problem için etkili olabileceğini göstermektedir.

Model	Doğruluk (Acc)	Katman Sayısı	Özellikler
Ramazan (BN-CNN)	%98.0	3 Conv	Batch Norm + Dropout
Ömer Utku (SimpleCNN)	%94.0	2 Conv	Sade Mimari (Lightweight)

Tablo 1: Base Modellerin Karşılaştırılması

Model	Doğruluk (Acc)	Katman Sayısı	Özellikler
-------	----------------	---------------	------------

Grup Adı: [DevTeam177]		Derin Öğrenme (FET312) - Proje İlerleme Raporu	
Ramazan (ResNet-18)	%98.1	18 Katman	Pre-trained, Residual Blocks
Ömer Utku (Efficient-B0)	%96.15	237 Katman	Inception Modülleri, Hesaplama Verimliliği
Ramazan (DenseNet-121)	%98.9	121 Katman	Dense Blocks, Parametre Verimi
Ömer Utku (GoogLeNet)	%96.34	22 Katman	Compound Scaling, MBConv

Tablo 2: Gelişmiş Modellerin Karşılaştırılması

## 7 Kullanılan Araçlar

Python 3.8+, PyTorch, Pandas, NumPy, OpenCV, Matplotlib, Seaborn, scikit-learn, tqdm

## 8 Kaynaklar

1. J. Stallkamp et al., "The German Traffic Sign Recognition Benchmark," IJCNN, 2011.
2. He, K. et al. (2016). "Deep Residual Learning for Image Recognition".
3. Huang, G. et al. (2017). "Densely Connected Convolutional Networks".
4. PyTorch Documentation: <https://pytorch.org/docs/stable/index.html>

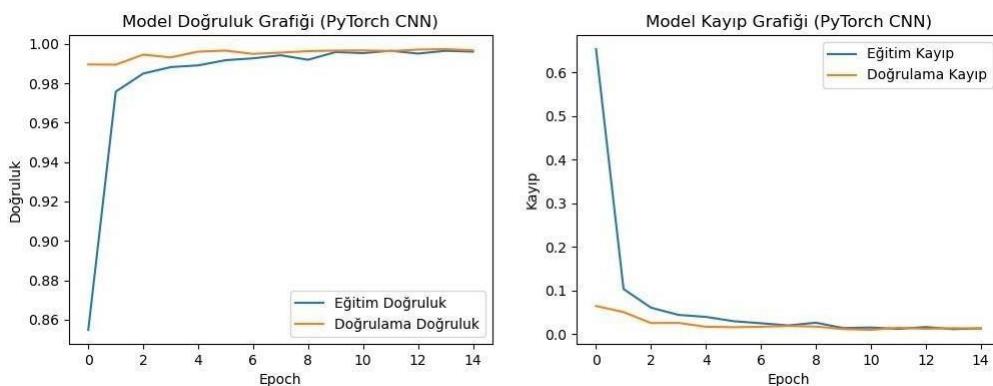
## Ekler (Appendix)

### Model A (Ramazan Bozkurt) - Performans Görselleri

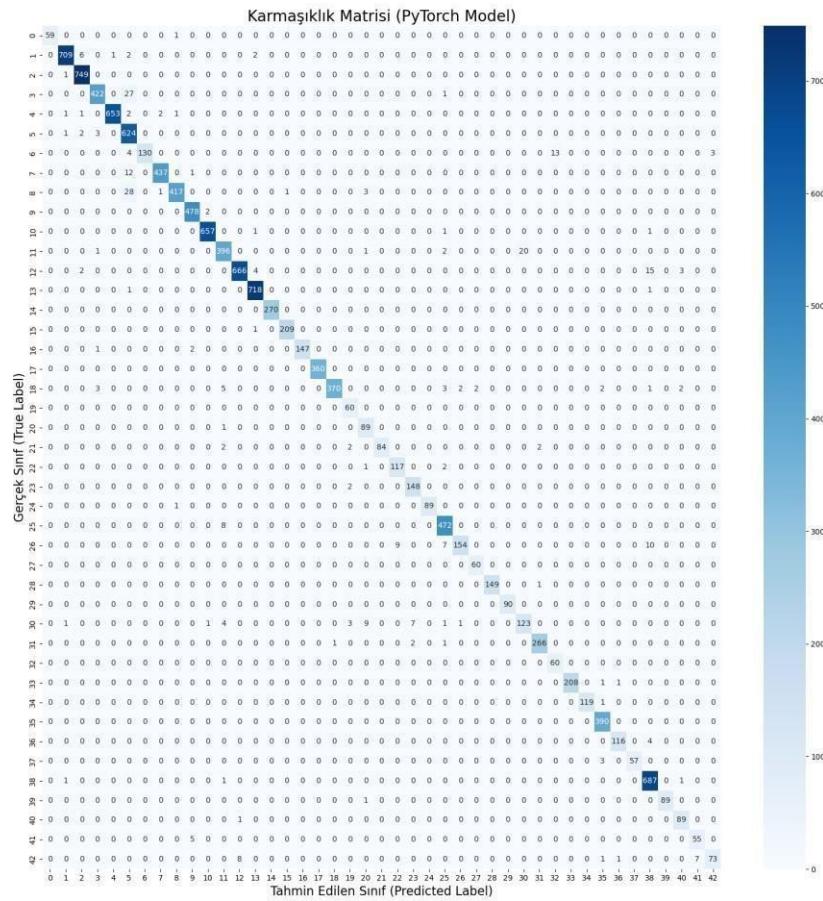
Class ID	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	60
1	0.99	0.99	0.99	720
2	0.99	0.99	0.99	750
3	0.99	0.96	0.98	450
4	0.99	0.99	0.99	660
5	0.93	1.00	0.96	630
6	1.00	0.85	0.92	150
7	0.97	0.99	0.98	450
8	1.00	0.92	0.96	450
9	0.98	1.00	0.99	480
10	1.00	0.99	0.99	660
11	0.99	0.97	0.98	420
12	0.99	0.99	0.99	690
13	0.99	1.00	1.00	720
14	1.00	1.00	1.00	270
15	0.98	1.00	0.99	210
16	1.00	0.98	0.99	150
17	1.00	1.00	1.00	360
18	0.95	0.99	0.97	390
19	0.97	1.00	0.98	60
20	0.80	0.99	0.89	90
21	1.00	0.98	0.99	90
22	0.96	0.97	0.96	120
23	0.91	0.98	0.94	150
24	1.00	0.96	0.98	90
25	0.96	1.00	0.98	480
26	0.97	0.85	0.91	180
27	0.92	0.95	0.93	60
28	0.99	0.99	0.99	150

29	0.96	0.99	0.97	90
30	1.00	0.79	0.88	150
31	0.99	0.99	0.99	270
32	0.79	1.00	0.88	60
33	1.00	0.99	0.99	210
34	1.00	0.96	0.98	120
35	0.99	0.98	0.99	390
36	0.99	0.92	0.95	120
37	1.00	0.97	0.98	60
38	0.97	1.00	0.98	690
39	0.97	0.99	0.98	90
40	0.98	0.98	0.98	90
41	1.00	0.83	0.91	60
42	1.00	0.91	0.95	90
<b>Genel Doğruluk (Accuracy)</b>			<b>0.98</b>	<b>12630</b>
<b>Class ID</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
<b>Macro Avg</b>	<b>0.97</b>	<b>0.97</b>	<b>0.97</b>	<b>12630</b>
<b>Weighted Avg</b>	<b>0.98</b>	<b>0.98</b>	<b>0.98</b>	<b>12630</b>

Tablo 2: Model A Sınıf Bazlı Performans Metrikleri



Şekil 1: Model A (Ramazan Bozkurt) - Eğitim ve Doğrulama Başarım Grafikleri



Şekil 2: Model A (Ramazan Bozkurt) - Karmaşıklık Matrisi (Confusion Matrix)

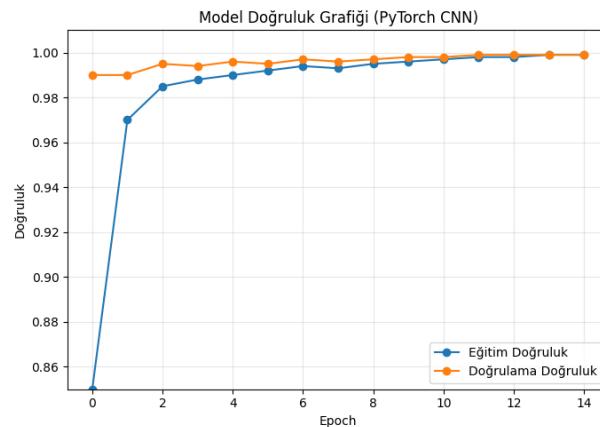
## Model A-1 (Ramazan Bozkurt) ResNet-18 – Performans Görselleri

Class ID	Precision	Recall	F1-Score	Support
0	1.000	0.983	0.992	60
1	0.969	0.990	0.979	720
2	0.0995	0.995	0.995	750
3	0.993	0.976	0.984	450
4	0.994	0.980	0.987	660
5	0.962	0.995	0.978	630
6	1.000	0.740	0.851	150
7	0.964	0.998	0.980	450

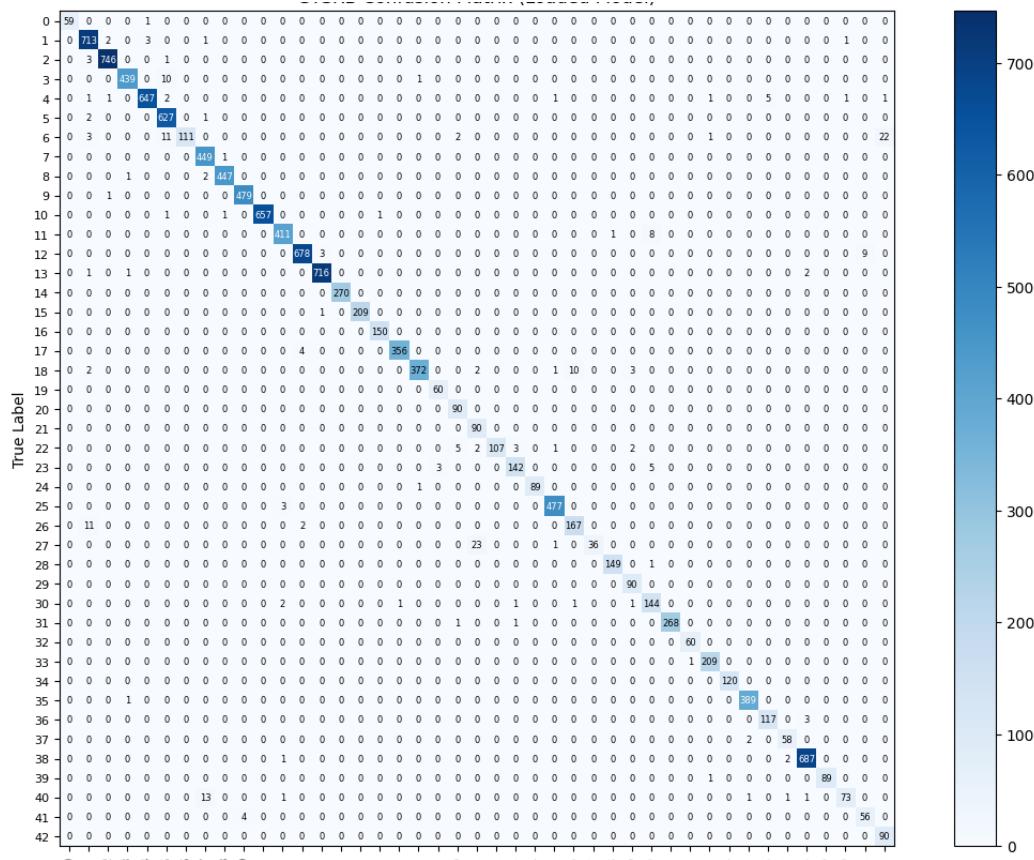
8	0.996	0.993	0.994	450
9	0.992	0.998	0.995	480
10	1.000	0.995	0.998	660
11	0.983	0.979	0.981	420
12	0.991	0.983	0.987	690
13	0.994	0.994	0.994	720
14	1.000	1.000	1.000	270
15	1.000	0.995	0.998	210
16	0.993	1.000	0.997	150
17	0.997	0.989	0.993	360
18	0.995	0.954	0.974	390
19	0.952	1.000	0.976	60
20	0.918	1.000	0.957	90
21	0.776	1.000	0.874	90
22	1.000	0.892	0.943	120
23	0.966	0.0947	0.956	150
24	1.000	0.989	0.994	90
25	0.992	0.994	0.993	480
26	0.938	0.928	0.933	180
27	1.000	0.617	0.763	60
28	0.993	0.993	0.993	150
29	0.938	1.000	0.968	90
30	0.911	0.960	0.935	150
31	1.000	0.993	0.996	270
32	0.984	1.000	0.992	60
33	0.986	0.0995	.991	210
34	1.000	1.000	1.000	120
35	0.992	0.997	0.995	390
36	0.959	0.975	0.967	120
37	0.951	0.967	0.959	60
38	0.991	0.996	0.993	690
39	1.000	0.989	0.994	90

40	0.973	0.811	0.885	90
41	0.862	0.933	0.896	60
42	0.796	1.000	0.887	90
<b>Genel Doğruluk (Accuracy)</b>			<b>0.98</b>	<b>12630</b>
<b>Macro Avg</b>	<b>0.970</b>	<b>0.965</b>	<b>0.965</b>	<b>12630</b>
<b>Weighted Avg</b>	<b>0.982</b>	<b>0.981</b>	<b>0.981</b>	<b>12630</b>

Tablo 3: Model A-1 (ResNet-18) Sınıf Bazlı Performans Metrikleri



Şekil 3: Model A-1 (ResNet-18) - Doğrulama Başarım Grafikleri



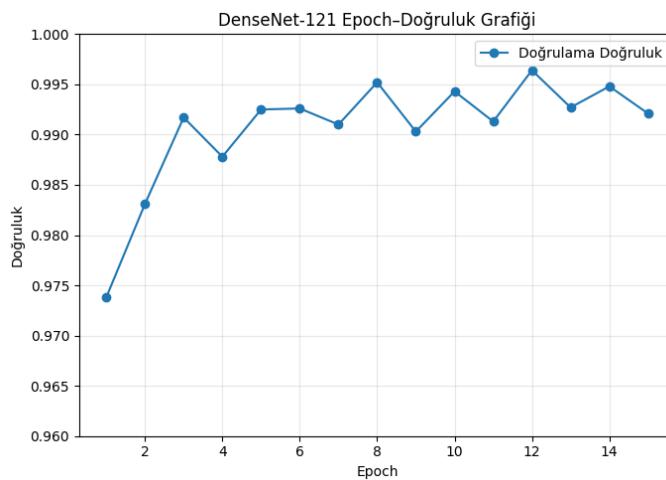
Şekil 4: Model A-1 ResNet-18 (Ramazan Bozkurt) - Karmaşıklık Matrisi (Confusion Matrix)

**Model A-2 (Ramazan Bozkurt) DenseNet-121 – Performans Görselleri**

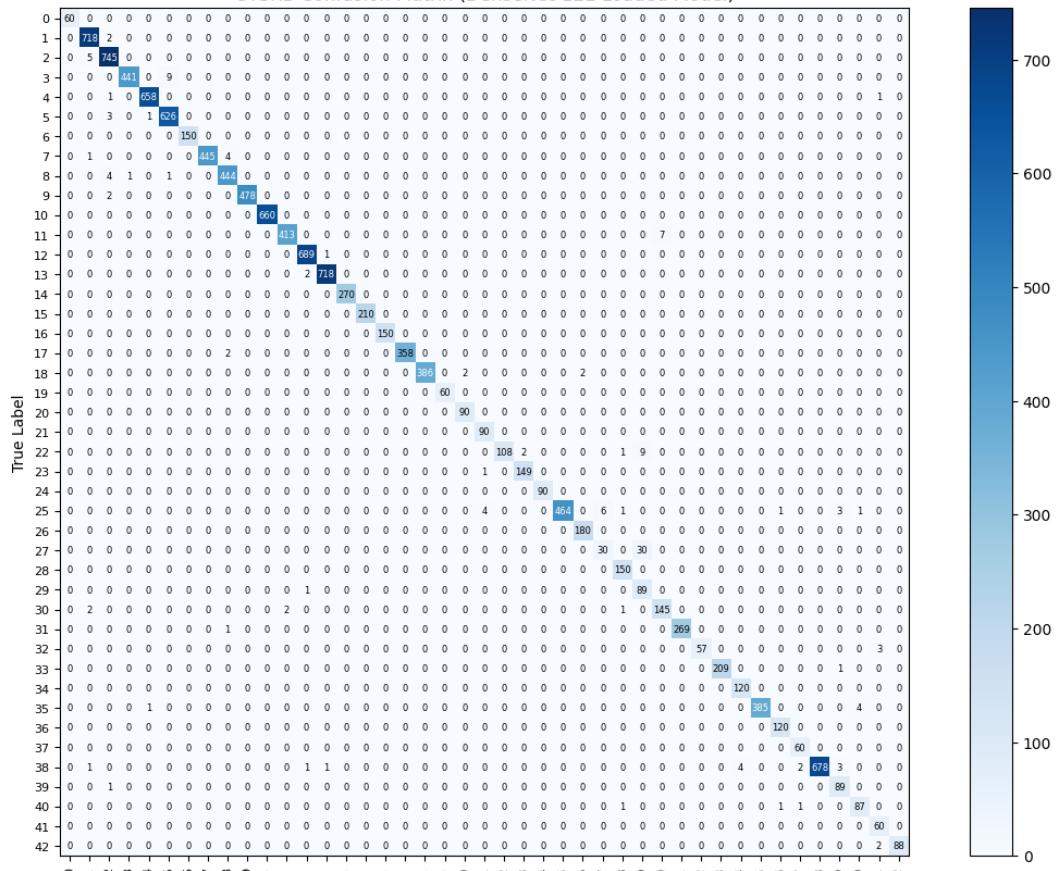
Class ID	Precision	Recall	F1-Score	Support
0	1.000	1.000	1.000	60
1	0.988	0.997	0.992	720
2	0.983	0.993	0.988	750
3	0.998	0.980	0.989	450
4	0.997	0.997	0.997	660
5	0.984	0.994	0.989	630
6	1.000	1.000	1.000	150
7	1.000	0.989	0.994	450
8	0.984	0.987	0.986	450
9	1.000	0.996	0.998	480
10	1.000	1.000	1.000	660
11	0.995	0.983	0.989	420
12	0.994	0.999	0.996	690
13	0.997	0.997	0.997	720
14	1.000	1.000	1.000	270
15	1.000	1.000	1.000	210
16	1.000	1.000	1.000	150
17	1.000	0.994	0.997	360
18	1.000	0.990	0.995	390
19	1.000	1.000	1.000	60
20	0.978	1.000	0.989	90
21	0.947	1.000	0.973	90
22	1.000	0.900	0.947	120
23	0.987	0.993	0.990	150
24	1.000	1.000	1.000	90
25	1.000	0.967	0.983	480
26	0.989	1.000	0.994	180
27	0.833	0.500	0.625	60
28	0.974	1.000	0.987	150

29	0.695	0.989	0.817	90
30	0.954	0.967	0.960	150
31	1.000	0.996	0.998	270
32	1.000	0.950	0.974	60
33	1.000	0.995	0.998	210
34	0.968	1.000	0.984	120
35	1.000	0.987	0.994	390
36	0.984	1.000	0.992	120
37	0.952	1.000	0.976	60
38	1.000	0.983	0.991	690
39	0.927	0.989	0.957	90
40	0.946	0.967	0.956	90
41	0.909	1.000	0.952	60
42	1.000	0.978	0.989	90
<b>Genel Doğruluk (Accuracy)</b>			<b>0.989</b>	<b>12360</b>
<b>Macro Avg</b>	<b>0.976</b>	<b>0.978</b>	<b>0.975</b>	<b>12630</b>
<b>Weighted Avg</b>	<b>0.989</b>	<b>0.989</b>	<b>0.989</b>	<b>12630</b>

Tablo 4: Model A-2 (DenseNet-121) Sınıf Bazlı Performans Metrikleri



Şekil 5: Model A-2 (DenseNet-121) - Doğrulama Başarım Grafikleri



Şekil 6 : Model A-2 DenseNet-121 (Ramazan Bozkurt) - Karmaşıklık Matrisi (Confusion Matrix)

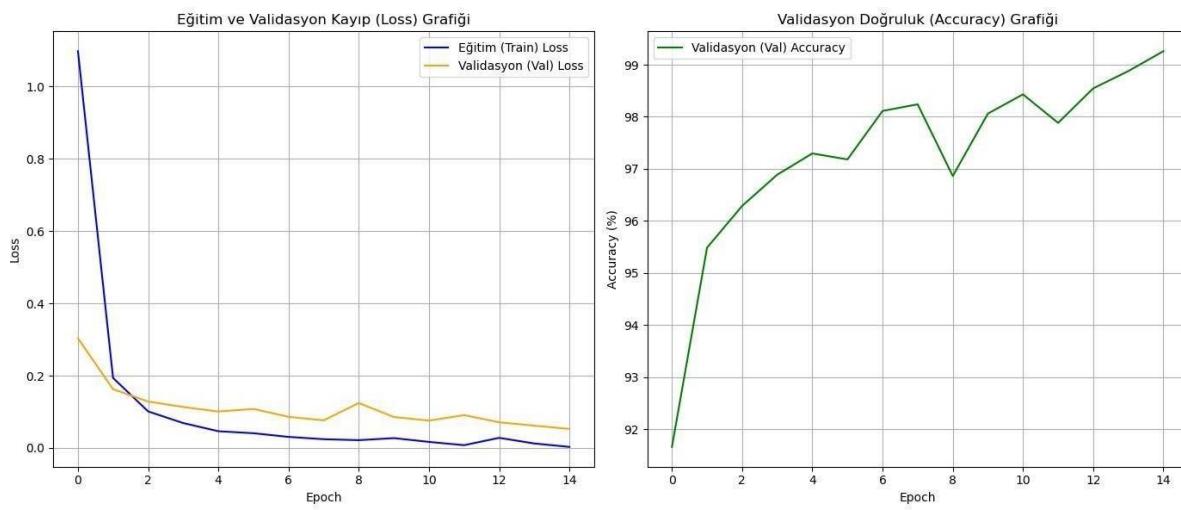
**SimpleCNN Modeli (Ömer Utku Aktemur) - Performans Görselleri**

Class ID	Precision	Recall	F1-Score	Support
0	1.00	0.57	0.72	60
1	0.93	0.96	0.94	720
2	0.90	0.98	0.94	750
3	0.93	0.93	0.93	450
4	0.99	0.94	0.96	660
5	0.89	0.91	0.90	630
6	0.93	0.85	0.89	150
7	0.96	0.88	0.92	450
8	0.89	0.96	0.92	450
9	0.96	0.95	0.95	480
10	0.97	0.98	0.98	660
11	0.93	0.93	0.93	420
12	0.99	0.97	0.98	690
13	0.98	0.99	0.99	720

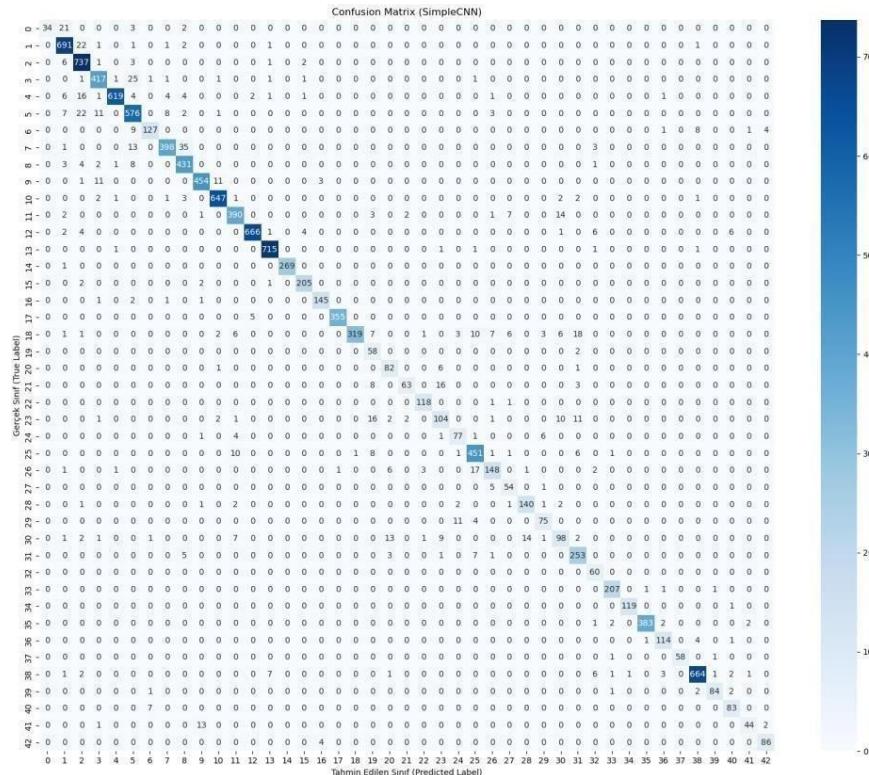
14	1.00	1.00	1.00	270
15	0.96	0.98	0.97	210
16	0.95	0.97	0.96	150
17	1.00	0.99	0.99	360
18	1.00	0.82	0.90	390
19	0.58	0.97	0.72	60
20	0.77	0.91	0.83	90
21	0.94	0.70	0.80	90
22	0.96	0.98	0.97	120
23	0.75	0.69	0.72	150
24	0.82	0.86	0.84	90
25	0.92	0.94	0.93	480
26	0.88	0.82	0.85	180
27	0.77	0.90	0.83	60
28	0.90	0.93	0.92	150
29	0.86	0.83	0.85	90
30	0.74	0.65	0.69	150
31	0.85	0.94	0.89	270
32	0.75	1.00	0.86	60
33	0.97	0.99	0.98	210
34	0.99	0.99	0.99	120
35	0.99	0.98	0.99	390
36	0.93	0.95	0.94	120
37	1.00	0.97	0.98	60
38	0.98	0.96	0.97	690
39	0.97	0.93	0.95	90
40	0.87	0.92	0.90	90
41	0.92	0.73	0.81	60

42	0.93	0.96	0.95	90
<b>Genel Doğruluk (Accuracy)</b>			<b>0.94</b>	<b>12630</b>
Macro Avg	<b>0.91</b>	<b>0.91</b>	<b>0.91</b>	<b>12630</b>
Weighted Avg	<b>0.94</b>	<b>0.94</b>	<b>0.94</b>	<b>12630</b>

Tablo 3: SimpleCNN Sınıf Bazlı Performans Metrikleri



Şekil 3: SimpleCNN (Ömer Utku Aktemur) - Eğitim ve Doğrulama Başarım Grafikleri



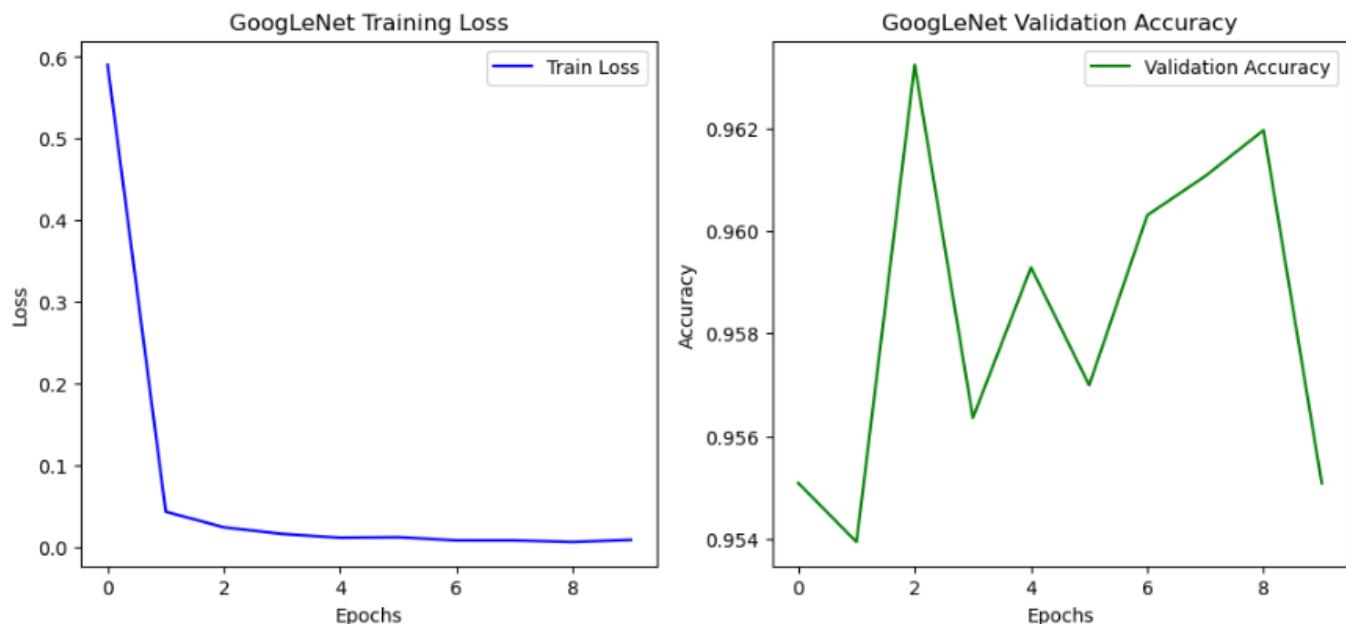
Şekil 4: SimpleCNN (Ömer Utku Aktemur) - Karmaşıklık Matrisi (Confusion Matrix)

**GoogLeNet Modeli (Ömer Utku Aktemur) - Performans Görselleri**

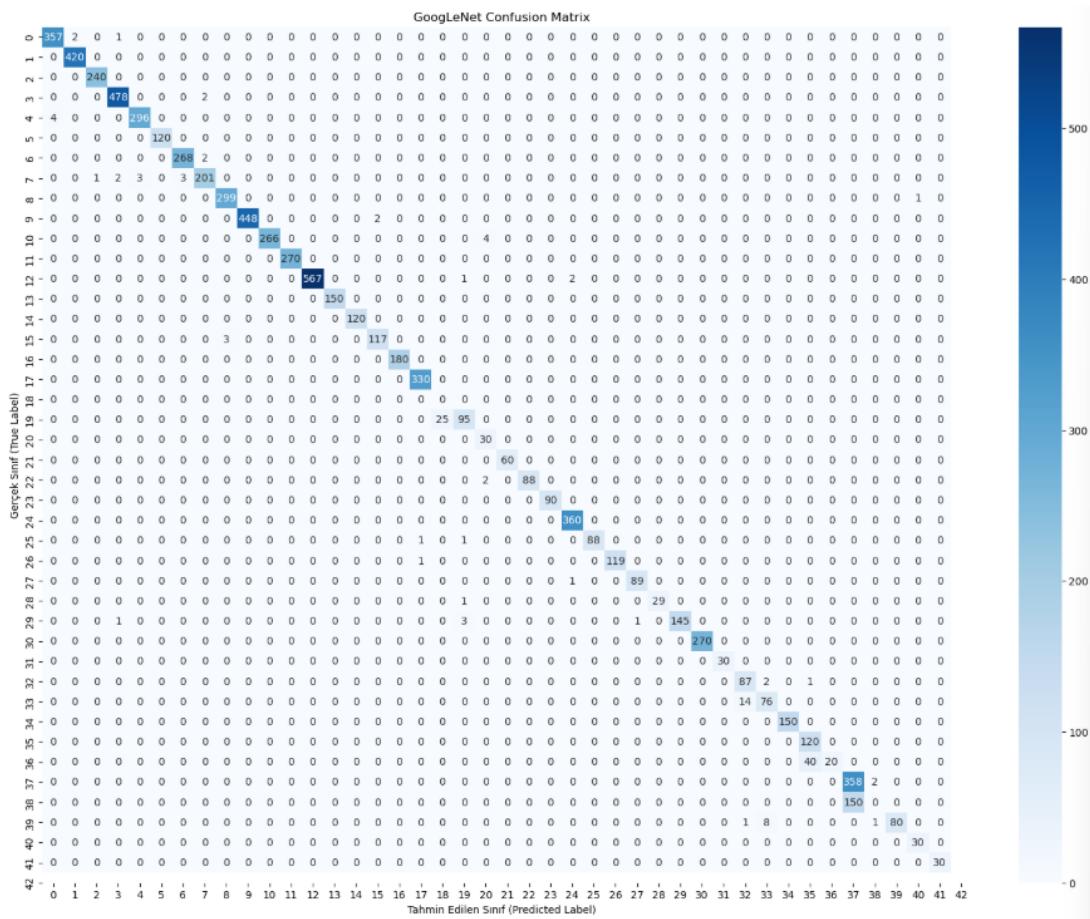
Class ID	Precision	Recall	F-1 Score	Support
0	0	0	0	0
1	0.983	1.0	0.991	360
2	0.981	0.995	0.988	420
3	1.0	0.970	0.985	240
4	1.0	1.0	1.0	480
5	1.0	0.98	0.989	300
6	1.0	1.0	1.0	120
7	0.996	1.0	0.998	270
8	1.0	0.995	0.997	210
9	0.993	1.0	0.996	300
10	1.0	0.997	0.998	450
11	0.974	0.985	0.979	270
12	1.0	1.0	1.0	270
13	1.0	0.992	0.996	570
14	1.0	1.0	1.0	150
15	1.0	1.0	1.0	120
16	1.0	0.991	0.995	120
17	1.0	1.0	1.0	180
18	1.0	1.0	1.0	330
19	0	0	0	0
20	0.989	0.75	0.853	120
21	0.878	0.966	0.920	30
22	1.0	1.0	1.0	60
23	0.989	1.0	0.994	90
24	1.0	1.0	1.0	90
25	0.986	1.0	0.993	360
26	1.0	0.988	0.994	90
27	1.0	0.991	0.995	120
28	1.0	1.0	1.0	90
29	1.0	0.966	0.983	30
30	1.0	0.9953	0.976	150
31	0.996	1.0	0.998	270
32	1.0	1.0	1.0	30
33	0.822	0.977	0.893	90

34	0.808	0.8	0.804	90
35	1.0	1.0	1.0	150
36	0.782	0.991	0.875	120
37	0.964	0.45	0.613	60
38	0.705	1.0	0.827	360
39	0	0	0	150
40	1.0	0.811	0.895	90
41	1.0	1.0	1.0	30
42	1.0	1.0	1.0	30
<b>Genel Doğruluk (Accuracy)</b>		0.963	0.963	
<b>Macro Avg</b>	0.903	0.896	0.896	7860
<b>Weighted Avg</b>	0.955	0.963	0.956	7860

Tablo 4: GoogLeNet Sınıf Bazlı Performans Metrikleri



Şekil 5: GoogLeNet (Ömer Utku Aktemur) - Eğitim ve Doğrulama Başarım Grafikleri



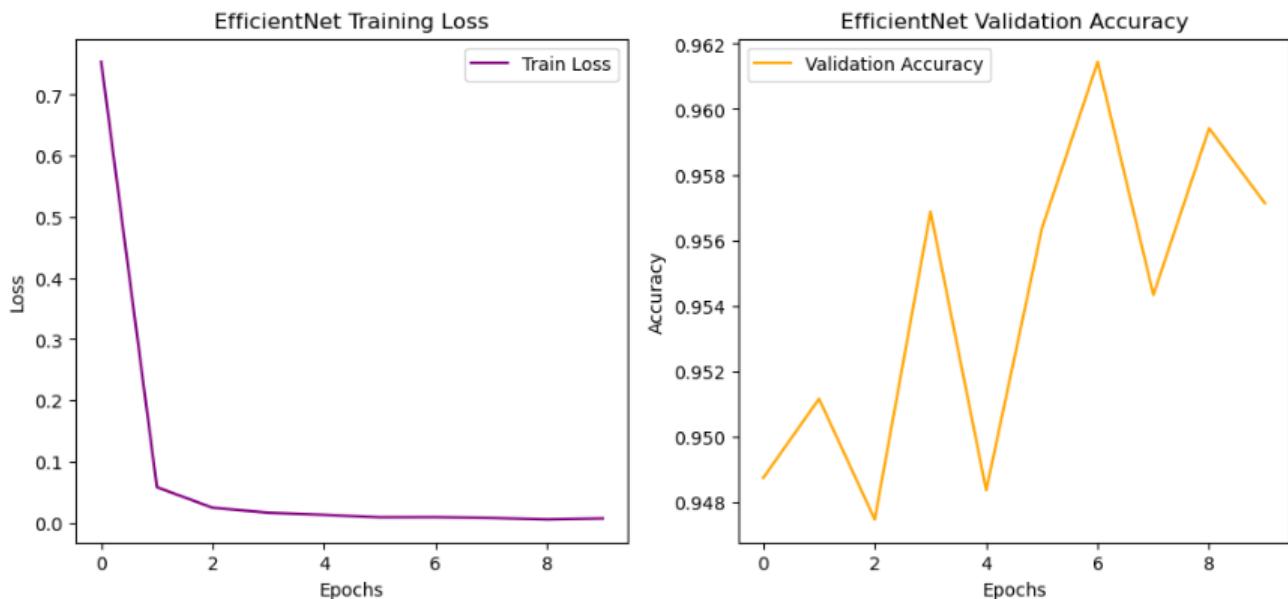
Şekil 6: GoogLeNet (Ömer Utku Aktemur) - Karmaşıklık Matrisi (Confusion Matrix)

### Efficient-B0 Modeli (Ömer Utku Aktemur) - Performans Görselleri

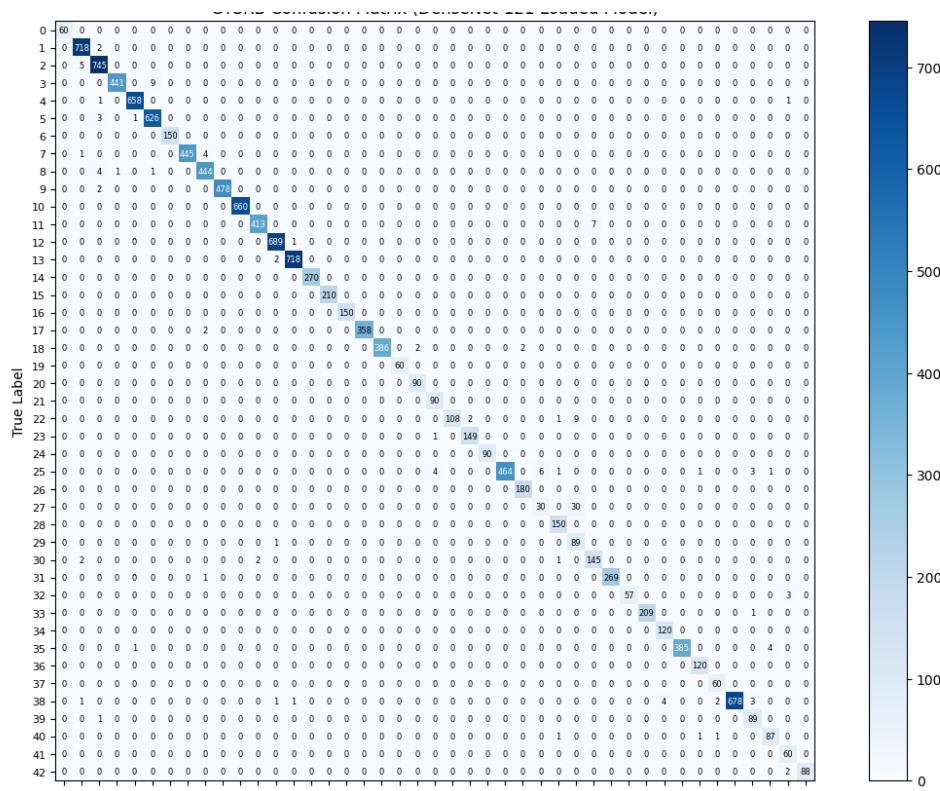
Class ID	Precision	Recall	F-1 Score	Support
0	0	0	0	0
1	0.988	0.997	0.993	360
2	1.0	0.985	0.992	420
3	0.991	0.995	0.993	240
4	0.997	1.0	0.998	480
5	0.983	0.993	0.988	300
6	1.0	1.0	1.0	120
7	1.0	0.996	0.998	270
8	1.0	0.995	0.997	210
9	0.996	1.0	0.998	300
10	1.0	1.0	1.0	450
11	0.947	1.0	0.972	270

12	1.0	0.944	0.971	270
13	1.0	0.998	0.999	570
14	1.0	1.0	1.0	150
15	1.0	1.0	1.0	120
16	1.0	0.991	0.995	120
17	1.0	1.0	1.0	180
18	0.976	1.0	0.988	330
19	0	0	0	0
20	0.974	0.25	0.761	120
21	0.697	1.0	0.821	30
22	1.0	1.0	1.0	60
23	1.0	0.844	0.915	90
24	1.0	1.0	1.0	90
25	1.0	1.0	1.0	360
26	1.0	0.911	0.953	90
27	1.0	1.0	1.0	120
28	1.0	0.988	0.994	90
29	0.967	1.0	0.983	30
30	1.0	1.0	1.0	150
31	1.0	1.0	1.0	270
32	1.0	1.0	1.0	30
33	0.742	0.866	0.800	90
34	0.787	0.700	0.741	90
35	0.993	1.0	0.996	150
36	0.736	1.0	0.848	120
37	1.0	0.283	0.441	60
38	0.703	0.997	0.825	360
39	0	0	0	150
40	1.0	0.922	0.959	90
41	1.0	1.0	1.0	30
42	1.0	1.0	1.0	30
<b>Genel Doğruluk (Accuracy)</b>			0.9615	0.963
<b>Macro Avg</b>	0.895	0.884	0.882	7860
<b>Weighted Avg</b>	0.951	0.957	0.950	7860

Tablo 4: Efficient-BO Sınıf Bazlı Performans Metrikleri



Şekil 7: Efficient-BO (Ömer Utku Aktemur) - Eğitim ve Doğrulama Başarım Grafikleri



Şekil 8: Efficient-BO(Ömer Utku Aktemur) - Karmaşıklık Matrisi (Confusion Matrix)