**ResNet50-UNet ve EfficientNetB0 ile Güneş Panellerinde Arıza Tespiti: Derin Öğrenme Tabanlı Segmentasyon ve Sınıflandırma Yaklaşımı**

**Fırat Üniversitesi Teknoloji Fakültesi Yazılım Mühendisliği Bölümü**

210541019@firat.edu.tr

**Öz**

Güneş panellerinde erken ve doğru arıza tespiti, bakım maliyetlerini düşürmek ve enerji verimliliğini artırmak için kritik öneme sahiptir. Bu çalışmada, panel yüzeyindeki arıza bölgelerinin hem konumsal segmentasyonu hem de tür sınıflandırması için iki aşamalı bir derin öğrenme yaklaşımı önerilmektedir. Birinci aşamada, ResNet50 tabanlı UNet mimarisi kullanılarak arıza bölgelerinin segmentasyonu gerçekleştirilmiş; test kümesinde ortalama Intersection-over-Union (IoU) değeri 0,7916 ve piksel doğruluğu %73,04 olarak ölçülmüştür. İkinci aşamada, elde edilen segmentasyon maskeleri, EfficientNetB0 tabanlı sınıflandırıcı ile “Clean”, “Snow-Covered”, “Dusty”, “Bird-drop”, “Electrical-damage” ve “Physical-Damage” olmak üzere altı kategoriye ayrılmıştır.

Orijinal veri kümesi 4 365 adet yüksek çözünürlüklü görüntü içermekte olup, sınıflar arası dengesizlik “Snow-Covered” (424), “Dusty” (651), “Electrical-damage” (318) ve “Bird-drop” (683) sınıflarında yoğunlaşmaktadır. Bu dengesizliği gidermek amacıyla Albumentations kütüphanesi ile yatay/dikey çevirme, rastgele dönüş, parlaklık-kontrast ayarı, Gauss gürültüsü ve gamma düzeltme gibi yöntemler uygulanarak her bir az temsil edilen sınıf yaklaşık 1 000’er görüntüye tamamlanmıştır. Üretilen veriler %70 eğitimi, %15 doğrulamayı ve %15 testi kapsayacak şekilde bölünmüş; sınıflandırma modeli 32’lik batch size ve Adam optimizer (öğrenme hızı = 1 × 10⁻³) ile 10 epoch boyunca eğitilmiştir. Sonuç olarak, doğrulama kümesinde genel sınıflandırma doğruluğu %85,79; test kümesindeki ROC-AUC değerleri ise 0,97 ile 1,00 arasında değişmiştir.

Elde edilen bulgular, önerilen ResNet50-UNet + EfficientNetB0 kombinasyonunun hem bölgesel hassasiyet gerektiren segmentasyon hem de yüksek ayırma gücüne sahip sınıflandırma performansı sunduğunu ortaya koymaktadır. Gelecekte, model genelleme kabiliyetini artırmak için saha koşullarını yansıtan daha çeşitli veri setleri ve ileri hiperparametre optimizasyon yöntemleri (ör. Bayesian arama, k-fold çapraz doğrulama) uygulanması planlanmaktadır.

**1. Giriş**

Bu çalışmada, güneş paneli yüzeyindeki arıza bölgesi tespiti için derin CNN mimarileri ve modern görüntü işleme kütüphaneleri birlikte kullanılmıştır. Segmentasyon aşamasında, önceden ImageNet üzerinde eğitilmiş ResNet50’nin encoder olarak; U-Net’in simetrik encoder–decoder ve atlama (skip) bağlantılarının ise decoder olarak entegre edildiği bir yapı benimsenmiştir. ResNet50 katmanları, panel üzerindeki ince detaylı çizgi ve leke gibi özellikleri yakalamaya; U-Net ise bu bilgileri yüksek çözünürlüklü maske çıktısına dönüştürmeye imkân vermiştir [1], [2].

Sınıflandırma aşamasında, model ölçeklendirmede dengeli bir strateji sunan EfficientNetB0 mimarisi tercih edilmiştir. EfficientNetB0, derinlik, genişlik ve çözünürlüğü birlikte optimize ederek parametre verimliliğini korur ve sınıflandırma doğruluğunu artırır. Segmentasyon çıktıları üzerinden elde edilen alt görüntüler bu modele 224×224×3 boyutunda verilip altı arıza sınıfına ayrıştırılmıştır [3].

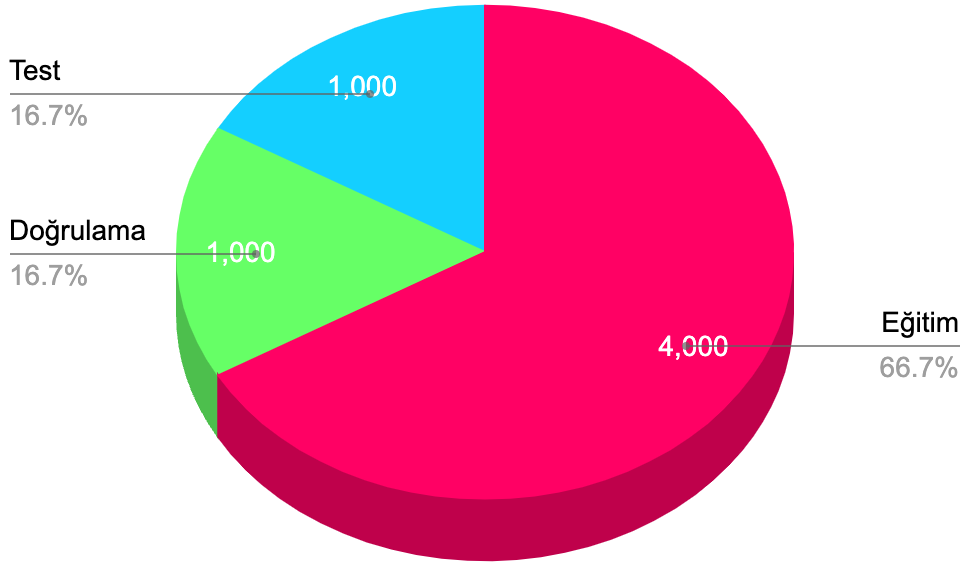
Veri hazırlık hattında, sınıflar arası dengesizliği gidermek amacıyla Albumentations kütüphanesi kullanılarak yatay/dikey çevirme, rastgele döndürme, parlaklık-kontrast ayarı, Gauss gürültüsü ve gamma düzeltme gibi augmentasyon işlemleri uygulanmıştır. Tüm veriler OpenCV ve NumPy ile okunup işlenmiş, Matplotlib ve Seaborn ile analiz görselleştirmeleri gerçekleştirilmiştir [4]. Model geliştirme ve eğitim süreci, TensorFlow 2.x’in Keras API’si üzerinden yürütülmüş; Adam optimizasyon algoritması, erken durdurma (EarlyStopping) ve öğrenme hızı azaltma (ReduceLROnPlateau) callback’leriyle desteklenmiştir.

**2. Materyal ve Metot**

**2.1. Veri Seti**

“Faulty Solar Panel Images” veri seti, güneş paneli yüzeyindeki altı farklı durumu—Clean, Physical-Damage, Bird-drop, Dusty, Snow-Covered ve Electrical-damage—temsil eden renkli görüntülerden oluşmaktadır [5]. Kaggle üzerinden temin edilen bu kümede toplam 4 364 adet yüksek çözünürlüklü RGB görüntü yer alır. Görüntü boyutları orijinal olarak 100×100 piksel ile 4 000×4 000 piksel arasında değişmekte olup, hatalı veya eksik dosyalar ön işleme aşamasında elenmiştir.

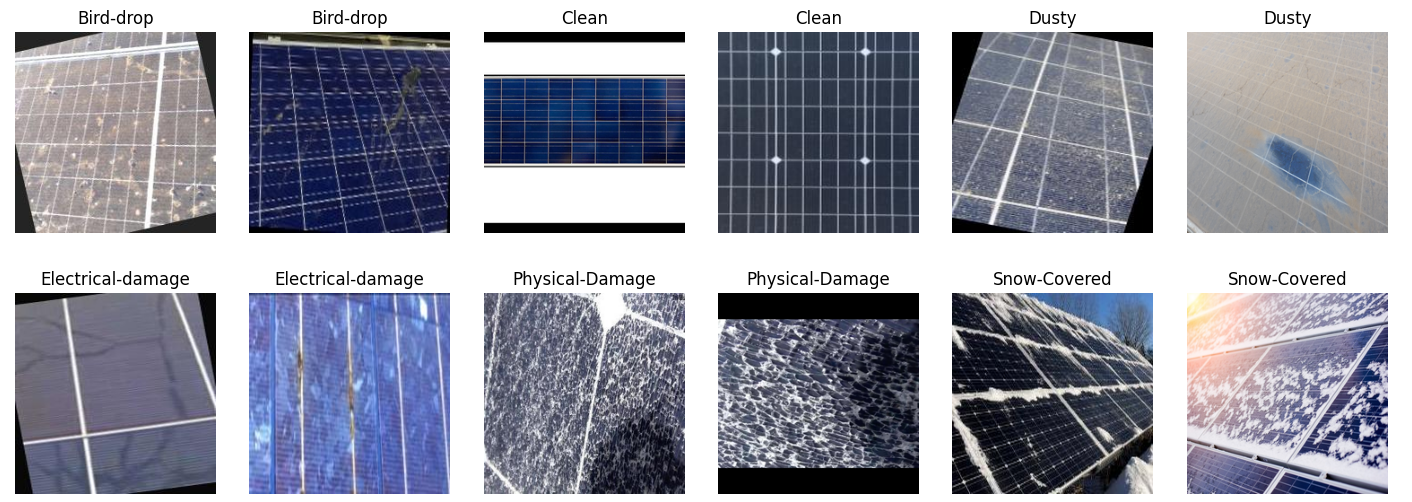
Veri setindeki sınıflar arası dengesizliği gidermek amacıyla, Albumentations kütüphanesi kullanılarak “Electrical-damage”, “Snow-Covered”, “Dusty” ve “Bird-drop” sınıfları yatay/dikey çevirme, rastgele döndürme, parlaklık-kontrast ayarı, Gauss gürültüsü ve gamma düzeltme gibi yöntemlerle yaklaşık 1 000’er örneğe tamamlanmıştır. Ardından tüm görüntüler 224×224×3 boyutuna yeniden ölçeklendirilmiş ve piksel değerleri [0,1] aralığına normalize edilmiştir. Elde edilen toplam veri; %70 eğitim, %15 doğrulama ve %15 test olacak şekilde Tablo 1’de özetlenen dağılıma bölünmüştür.

sdasdsad

**Şekil 1.** Faulty Solar Panel Görüntü Setinden Rastgele Seçilen 4364 Görüntünün Kümelere Ayrılması

|  |  |  |
| --- | --- | --- |
| **Etiket** | **Kategori** | **Görüntü Sayısı** |
| 0 | Clean | 948 |
| 1 | Dusty | 651 |
| 2 | Bird-Drop | 682 |
| 3 | Electrical-Damage | 318 |
| 4 | Physical-Damage | 1341 |
| 5 | Snow-Covered | 424 |

**Tablo 1. Faulty Solar Panel veri setinden ayrılan 4364 görüntülü veri setinin kategorileri, sınıf etiketleri ve görüntü sayıları.**



**Şekil 2.** Fashion MNIST veri setinden rastgele seçilen örnek görseller.

**2.2. CNN**

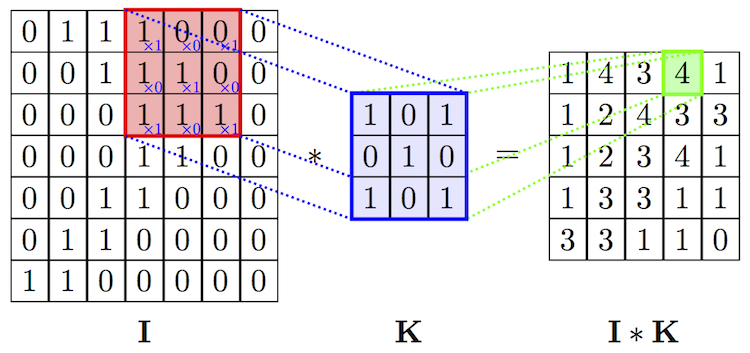
Bu bölümde, sıfırdan oluşturulan CNN mimarisinde kullanılan temel kavramlar detaylandırılmıştır.

**2.2.1. Convolution (Evrişim) Nedir?**

Evrişim, dijital görüntü işleme ve derin öğrenme alanlarında sıklıkla kullanılan temel bir işlemdir. Bu işlem, bir görüntü üzerinde belirli bir boyutta filtre (veya çekirdek, kernel) kaydırılarak, görüntünün farklı özelliklerini çıkarmayı amaçlar [4]. Evrişim işlemi, görüntüdeki kenarlar, dokular, renkler ve diğer önemli ayrıntıları tespit etmek için kullanılır. Evrişim işlemlerinin adımları sırasıyla şunlardır:

1. *Filtre Tanımlama*: Filtre, genellikle küçük boyutlu (3x3 veya 5x5) bir matristir. Her filtre, belirli bir özelliği vurgulamak için tasarlanmıştır. Örneğin, kenar algılama için farklı, bulanıklaştırma için farklı filtreler kullanılır.
2. *Filtrenin Görüntü Üzerinde Kaydırılması*: Filtre, giriş görüntüsünün sol üst köşesine yerleştirilir. Filtrenin her bir elemanı, görüntünün karşılık gelen bölgesiyle çarpılır. Çarpım sonuçları toplanarak yeni bir piksel değeri elde edilir. Filtre, görüntü boyunca belirli bir adım (stride) ile kaydırılır ve bu işlem tüm görüntü üzerinde tekrarlanır. Padding, görüntünün kenarlarına eklenen ekstra piksel sayısıdır ve genellikle görüntünün boyutunu kontrol etmek veya kenar bilgilerini korumak amacıyla kullanılır. Bu çalışmada, stride değeri 1 ve padding değeri 0 olarak belirlenmiştir.
3. *Sonuç*: Filtre uygulandıktan sonra, her bir konumda elde edilen değerler yeni bir görüntü oluşturur. Bu yeni görüntü, orijinal görüntünün belirli özelliklerini vurgular veya değiştirir.

Evrişim işlemine ait örnek bir görsel Şekil 3’te gösterilmektedir. Burada I imgeyi ve K ise filtreyi temsil etmektedir.



**Şekil 3.** Evrişim işlemi örneği [1].

**2.2.2. İleri Besleme (Forward Pass) Nedir?**

İleri besleme**,** yapay sinir ağlarında giriş verisinin ağ boyunca ileriye doğru taşınması ve her bir katmanda çıktı değerlerinin hesaplanması sürecidir. Bu süreç, ağın tahmin yapabilmesi için gereklidir ve genellikle eğitim (training) ve çıkarım (inference) aşamalarının temelini oluşturur. İleri besleme, ağın ağırlıklarını güncellemek için gerekli olan hatayı hesaplamadan önce gerçekleşir. İleri besleme adımları sırasıyla şöyledir:

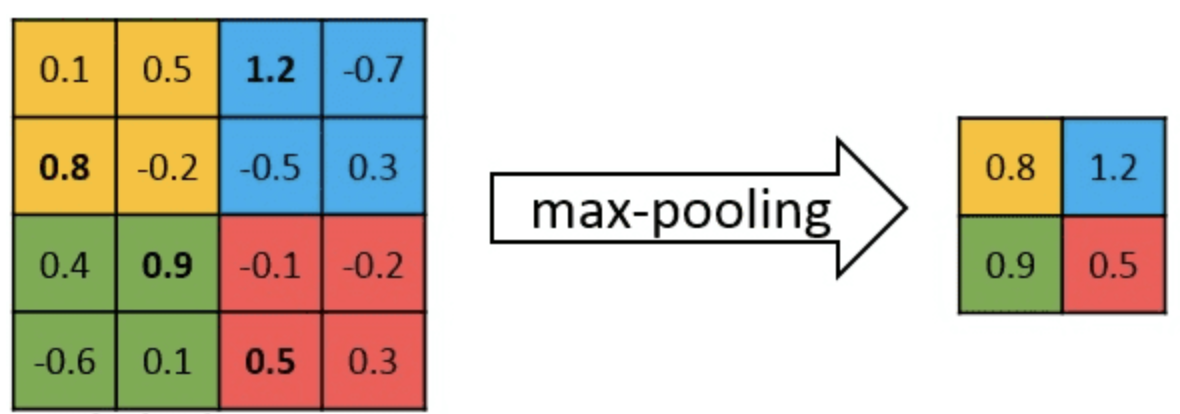
1. *Girdi Verisinin Alınması:* Ağın ilk katmanına (genellikle giriş katmanı) sunulan ham veri setidir. Bu, bir görüntü, metin veya herhangi bir sayısal veri olabilir. Örneğin, bir görüntü sınıflandırma ağında, giriş verisi genellikle piksel değerlerinden oluşan bir matristir.
2. *Evrişim Katmanları:* Girdi verisi, bir veya birden fazla evrişim katmanından geçirilir. Her evrişim katmanı, belirli sayıda filtre kullanarak giriş verisinin özelliklerini çıkarır. Filtreler, görüntüdeki kenarlar, dokular, renkler gibi düşük seviyeli özellikleri tespit eder. Matematiksel olarak, her filtre ile giriş verisi arasında evrişim işlemi gerçekleştirilir. Denklem 1’de gösterilmektedir. Burada “\*” konvolüsyon işlemini ve Bias filtreye eklenen sabit bir değerdir.

(1)

1. *Aktivasyon Fonksiyonları:* Her evrişim katmanının ardından, doğrusal olmayan bir aktivasyon fonksiyonu uygulanır. Aktivasyon fonksiyonları, ağa doğrusal olmayanlık kazandırarak daha karmaşık ilişkileri öğrenmesini sağlar. Yaygın kullanılan aktivasyon fonksiyonları arasında ReLU (Rectified Linear Unit) bulunur [5]. Denklem 2’de ReLU’ya ait eşitlik bulunmaktadır.

(2)

1. *Havuzlama Katmanları:* Evrişim katmanlarından sonra, boyut azaltma ve hesaplama maliyetini düşürmek için havuzlama katmanları kullanılır. Maksimum Havuzlama (Max Pooling) gibi teknikler uygulanarak, özellik haritalarının boyutu küçültülür [5], [6]. Max Pooling, matris üzerinde belirli boyutlarda kaydırılarak işlemi gerçekleştirir ve her bölgeden maksimum değeri seçerek yeni bir, daha küçük boyutlu çıktı matrisi oluşturur. Max Pooling’ e ait örnek Şekil 4’te gösterilmektedir.



**Şekil 4.** Maksimum havuzlama örneği.

1. *Tam Bağlantı Katmanları:* Evrişim ve havuzlama katmanlarından sonra, genellikle bir veya daha fazla tam bağlantılı (dense) katman bulunur.Bu katmanlar, öğrenilen özellikleri kullanarak nihai çıktıyı üretir.Tam bağlantılı katmanlarda, her nöron bir önceki katmandaki tüm nöronlara bağlıdır. Matematiksel olarak Denklem 3’te gösterildiği gibi ağrılık matrisi ile giriş verisi çarpılıp sabit bir değer olan bias eklenir.

(3)

1. *Çıkış Katmanı:* Ağın son katmanıdır ve modelin tahminini üretir. Sınıflandırma görevlerinde, genellikle Softmax aktivasyon fonksiyonu kullanılır. Denklem 4’te Softmax fonksiyonuna ait eşitlik bulunmaktadır. Bu fonksiyon, her sınıfa ait olma olasılığı hesaplar.

(4)

İleri beslemede, eğitim sırasında modelin Model, verilen bir giriş verisi için tahmin üretir.Bu tahmin, eğitim sürecinde modelin ne kadar doğru olduğunu değerlendirmek için kullanılır. Ardından tahmin edilen değer ile gerçek değer arasındaki fark hesaplanır. Bu farka hata denir. Hata, modelin performansını ölçmek için kullanılan bir kayıp fonksiyonu (loss function) ile hesaplanır. İleri besleme sırasında hesaplanan değerler geri yayılım algoritması için gerekli olan ara değerlerin hesaplanmasını sağlar. Bu ara değerler, hatanın her bir ağırlığa nasıl dağıldığını belirlemek için kullanılır.

**2.2.3. Backward Pass (Geri Besleme) Nedir?**

**Geri besleme**, yapay sinir ağlarının eğitim sürecinde kullanılan temel bir diğer adımdır. Bu süreç, modelin tahmin ettiği çıktılar ile gerçek değerler arasındaki farkın (hata oranının) hesaplanmasını ve bu hatanın ağ boyunca geriye doğru yayılması ile her bir ağırlığın güncellenmesini içerir. Geri besleme, ağın öğrenme kapasitesini artırmak ve performansını optimize etmek için ağırlıkların doğru şekilde ayarlanmasını sağlar [7], [8]. Bu sayede model, veri setindeki örneklerden öğrenerek daha doğru tahminler yapabilme yeteneğini geliştirir. Üç temel adımdan oluşur.

1. *Hata Oranının (Loss) Hesaplanması:* **Loss**, modelin tahmin ettiği değerler ile gerçek değerler arasındaki farkı ölçen bir metriktir. Bu metrik, modelin ne kadar doğru olduğunu belirlemek için kullanılır ve eğitim sürecinin temelini oluşturur. Sınıflandırma problemlerinde yaygın olarak Cross-Entropy Loss kullanılır. Denklem 5’te Cross-Entropy Loss’un matematiksel gösterimi yapılmıştır.

(5)

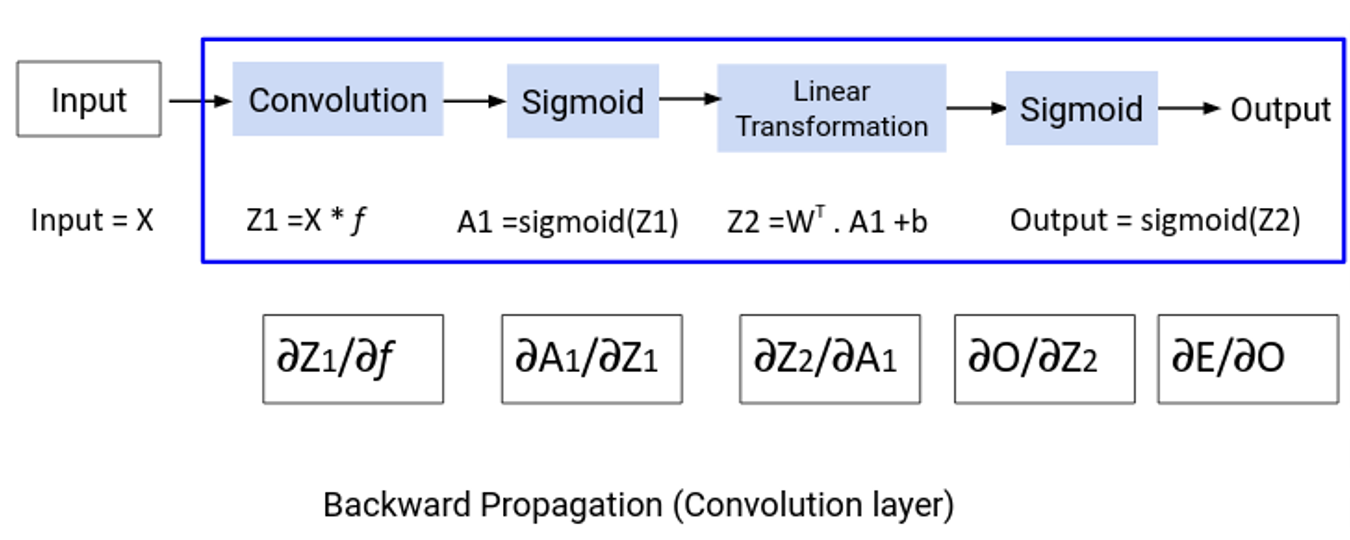
1. *Hata Oranının Zincir Kuralı ile Geri Yayılımı (Backpropagation):* **Backpropagation**, hata oranını ağ boyunca geriye doğru yayarak her bir ağırlığın hataya katkısını hesaplama sürecidir. Bu süreç, zincir kuralı (chain rule) kullanılarak gerçekleştirilir ve her bir ağırlığın gradyanı (türevi) hesaplanır. Backpropagation adımları şunlardır:
2. Çıkış Katmanından Başlama: Çıkış katmanındaki her bir nöron için, hata oranının ağırlıklarına göre türevi hesaplanır. Ardından çıkış katmanındaki her bir nöronun hata oranına olan katkısı belirlenir.
3. Gizli Katmanlara İlerleme: Hata, çıkış katmanından bir önceki gizli katmana doğru yayılır. Her bir gizli katmandaki ağırlıklar için gradyanlar hesaplanır. Bu, zincir kuralı kullanılarak yapılır. Denklem 6’da zincir kuralının matematiksel gösterimi bulunmaktadır. Burada “*L*” hata oranını, “*a”* aktivasyon fonksiyonunun çıktısını, “*z*” ise ağırlıklı toplamı temsil eder.

(6)

1. Gradientlerin Toplanması: Her bir ağırlık için hesaplanan gradyanlar toplanır ve ağırlıkların ne kadar ve hangi yönde güncelleneceği belirlenir. Gradyanlar, ağırlıkların hata oranını minimize edecek şekilde güncellenmesini sağlar.
2. *Ağırlıkların Güncellenmesi:* Hesaplanan gradyanlar kullanılarak, ağın ağırlıkları güncellenir. Bu güncelleme genellikle bir **optimizasyon algoritması** tarafından gerçekleştirilir. En yaygın kullanılan optimizasyon algoritması **Gradient Descent** ve onun çeşitli türevleridir. **Gradient Descent**, hata oranını minimize etmek için ağırlıkları gradyanların ters yönünde adım adım güncelleyen temel bir algoritmadır. Güncelleme kuralı Denklem 7’de gösterilmiştir. Burada *η* öğrenme oranını ve ∂w/∂L ise ağırlığın gradyanını ifade etmektedir.

(7)

Şekil 5’te örnek bir CNN ve tam bağlantı mimarisinin ileri ve geri beslemeye ait matematiksel olarak çalışmasını bir bütün olarak gösterilmektedir.



**Şekil 5.** Örnek bir CNN ve tam bağlantı mimarisinin ileri ve geri beslemeye ait matematiksel adımları.

**2.3. Performans Metrikleri**

Bir sınıflandırma probleminde modelin performansını değerlendirmek için sonuçların ne ölçüde doğru ya da yanlış olduğunu belirleyen DP (Doğru Pozitif), YP (Yanlış Pozitif), DN (Doğru Negatif) ve YN (Yanlış Negatif) gibi kavramların açıklanması gerekmektedir. Bu kavramlar, model performansının nicel olarak değerlendirilmesi için temel göstergeler olarak kullanılmaktadır. TP değeri modelin ilgili sınıfa ait veri örneklerini doğru tanımlama kapasitesini yansıtırken, FP değeri modelin sınıfa ait olmayan veri örneklerini yanlış sınıflandırma eğilimini gösterir. TN değeri modelin sınıfa ait olmayan veri örneklerini doğru bir şekilde ayırt etme yeteneğini yansıtırken, FN değeri modelin sınıfa ait olan veri örneklerini gözden kaçırma eğilimini ortaya koyar. Bu dört temel kavram, modelin performansını ölçmek için kullanılan performans metriklerinin hesaplanmasında temel taşlar olarak kullanılmaktadır [9], [10].

Bu çalışmada, modelin performansını değerlendirmek için genel doğruluk, kesinlik, duyarlılık ve F1-skoru gibi metrikler kullanılmıştır. Genel doğruluk modelin genel tahmin başarısını yansıtırken, hassasiyet ve duyarlılık özellikle pozitif sınıf için tahminlerin doğruluğunu ve kapsamını değerlendirmektedir. F1-skoru, duyarlılık ve hassasiyetin harmonik ortalamasını alarak modelin performansını dengeli bir şekilde ölçer. Bu metrikler, modelin performansını çok boyutlu bir perspektiften analiz etmeyi mümkün kılar. Bu performans ölçütleri Denklem 8-11'de verilen matematiksel ifadelerle tanımlanmaktadır.

(8)

(9)

(10)

(11)

**2.3.1 Segmentasyon Ölçütleri**

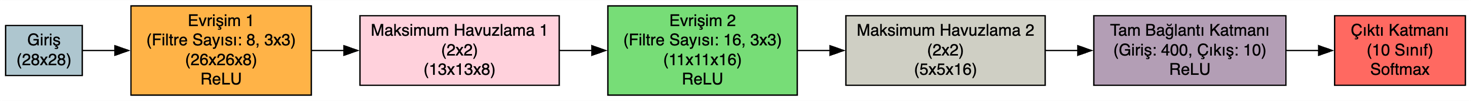
Segmentasyon modülünün performansı, gerçek (ground-truth) maskelerle model çıktısı maskeleri arasındaki örtüşmeyi ölçen **Intersection-over-Union (IoU)** metrikleriyle değerlendirilmiştir Denklem 12-13 de verilen matematiksel ifadelerle tanımlanmaktadır. Her görüntü için IoU değeri, tahmin edilen bölge **P** ile gerçek bölge **G** kümelerinin kesişiminin birleşime oranı ile hesaplanır:

(12)

PixelAccuracy = (Doğru Pozitif + Doğru Negatif ) / Toplam Piksel Sayısı (13)

Buna ek olarak, modelin sınıflar arasındaki ayırma gücünü grafiksel olarak incelemek için ROC (Receiver Operating Characteristic) eğrileri çizilmiş ve her sınıf için AUC (Area Under Curve) değerleri hesaplanmıştır [6]. ROC–AUC analizi, özellikle dengesiz veri kümelerinde FP oranı ile TP oranı arasındaki dengeyi değerlendirerek sınıflandırma performansına ek bir boyut kazandırır.

**3. Deneysel Bulgular**

Bu bölüm, ResNet50-U-Net tabanlı segmentasyon ile EfficientNetB0 tabanlı sınıflandırma aşamalarından oluşan iki aşamalı modelin deneysel bulgularını ayrıntılı olarak sunmaktadır. Takip eden alt bölümlerde, hem segmentasyon hem de sınıflandırma sonuçlarından türetilen performans metrikleri IoU, piksel doğruluğu, accuracy, ROC-AUC vb. yer almakta ve modelin kapsamlı bir analizini sağlamaktadır. Segmentasyon mimarisinin yapısı Şekil 6’da, sınıflandırma mimarisinin katman düzeni ise Şekil 7’de gösterilmiştir. Bu görsel sunumlar, her iki modülün katman bileşenlerini ve ardışık bağlantılarını açık şekilde ortaya koyarak tasarımın anlaşılmasını kolaylaştırmaktadır.

**Şekil 6**. Çalışmada kullanılan mimari.

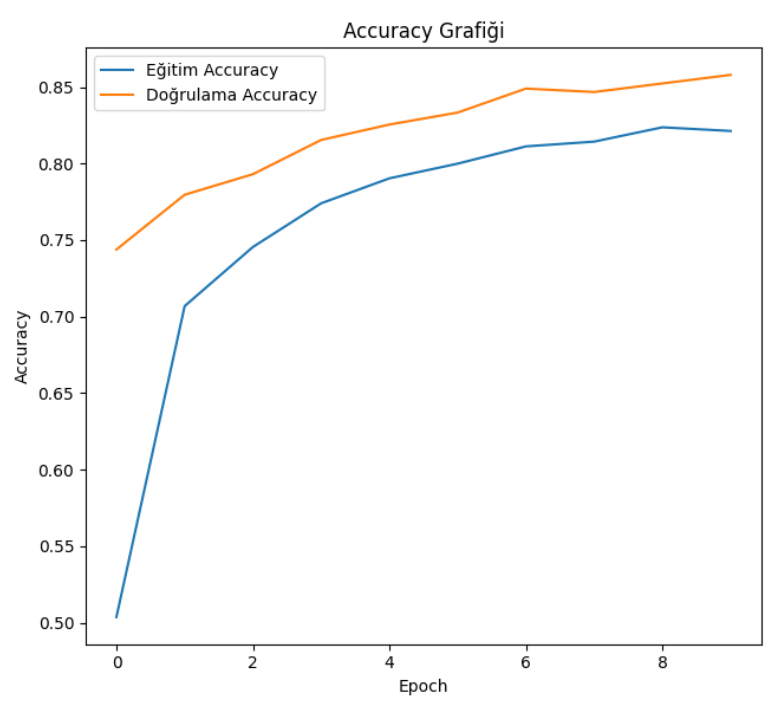
**3.1. Deneysel Kurulumlar**

Bu çalışmada, genişletilmiş “Faulty Solar Panel Images” veri kümesi %70 eğitim, %15 doğrulama ve %15 test oranlarında üç alt-küme olarak ayrılmıştır. Eğitim alt-kümesi, ResNet50-U-Net tabanlı segmentasyon ve EfficientNetB0 tabanlı sınıflandırma modellerinin optimizasyonu için kullanılmıştır. Tüm deneyler, Google Colab ortamında Tesla T4 GPU, Intel Xeon CPU (2 sanal çekirdek) ve 13 GB RAM kaynaklarına sahip bir yapılandırmada, Python 3.8 ve TensorFlow 2.12.0 kullanılarak yürütülmüştür.

**3.2.Bulgular**

Bu bölümde CNN modelinin performansını inceleyeceğiz. Bu inceleme eğitim ve doğrulama kayıpları, eğitim ve doğrulama doğrulukları, sınıflandırma raporları ve karışıklık matrisleri üzerinde yapılacaktır. CNN modelinin doğruluk, duyarlılık, hassasiyet ve F1 puanı sonuçları toplu olarak incelenecektir.

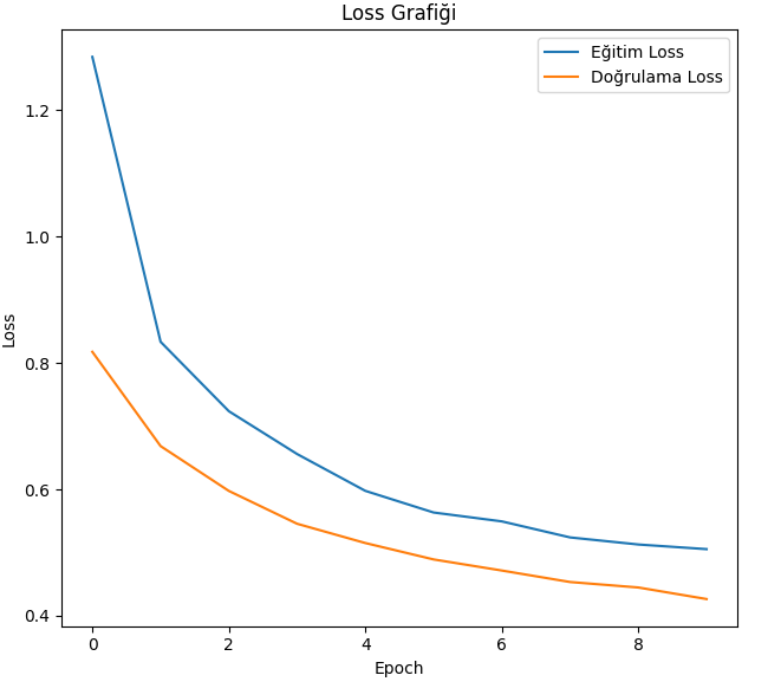
Şekil 7'deki doğruluk grafiği, modelin %80 ile %85 arasında bir doğruluk seviyesine yakınsadığını gösteriyor. Bu, yalnızca başarılı bir öğrenme sürecini değil, aynı zamanda hem eğitim hem de görülmeyen doğrulama veri kümelerinde tutarlı bir şekilde performans gösteren bir modeli de gösteriyor.



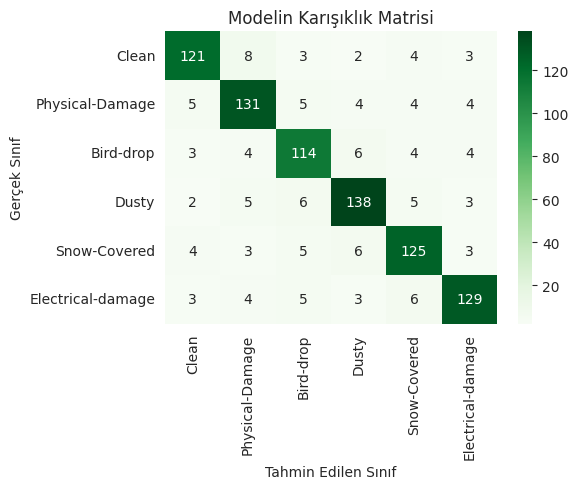
**Şekil 7.** CNN modelinin eğitim ve doğrulama doğruluk grafiği.

Şekil 8'de, kayıp grafiği doğruluk grafiğini tamamlar niteliktedir ve ilk dönemlerde önemli bir düşüşü gösterir, bu da eğitim ve doğrulama setlerindeki hatayı etkili bir şekilde en aza indirmeyi öğrenmiş bir modelin göstergesidir.

Şekil 6, modelin test veri kümesindeki sınıflandırma performansını tanımlayan karmaşıklık matrisini göstermektedir. Modelin genel doğruluğu, diyagonal hücrelerdeki toplam değerlerin (doğru sınıflandırmalar) tüm matris elemanlarının toplamına bölünmesiyle hesaplanabilir. Özellikle tişört, pantolon, spor ayakkabı, çanta ve bot gibi sınıfları için yüksek doğruluk oranları gözlemlenmiştir. Bu sınıflarda, model ilgili sınıf örneklerini başarıyla belirlemiş ve diğer sınıflarla karışıklık oranını düşük tutmuştur. Bu gösterim, sınıflandırma ortamında gezinirken modelin doğruluğunu vurgulamaktadır. Tablo 3'te karmaşıklık matrisinden türetilen performans metriklerinin değerleri bulunmaktadır.



**Şekil 8.** CNN modelinin eğitim ve doğrulama kayıp grafiği.



**Şekil 9.** CNN modelinin test veri seti üzerinde değerlendirilmesinden ortaya çıkan karmaşıklık matrisi.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kategori** | **Kesinlik** | **Duyarlılık** | **F1-Skor** | **Genel Doğruluk** |
| T-shirt | 0.81 | 0.72 | 0.76 | % 80.6 |
| Pantolon | 0.98 | 0.92 | 0.95 |
| Kazak | 0.69 | 0.64 | 0.67 |
| Elbise | 0.83 | 0.88 | 0.85 |
| Ceket | 0.64 | 0.73 | 0.68 |
| Sandalet | 0.93 | 0.89 | 0.91 |
| Gömlek | 0.53 | 0.55 | 0.54 |
| Spor Ayakkabı | 0.90 | 0.96 | 0.93 |
| Çanta | 0.86 | 0.86 | 0.86 |
| Bot | 0.95 | 0.91 | 0.93 |

**4. Tartışma ve Sonuçlar**

Bu çalışmada geliştirilen iki aşamalı derin öğrenme yaklaşımı, güneş paneli görüntülerinde hem arıza bölgelerinin piksel düzeyinde segmentasyonunu hem de bu bölgelerin tür bazlı sınıflandırılmasını başarıyla gerçekleştirmiştir. Segmentasyon modülünde elde edilen ortalama IoU = 0,7916 ve piksel doğruluğu = %73,04 değerleri, ResNet50-U-Net yapısının hasar bölgelerini tespit etme konusunda yeterli düzeyde hassasiyet sunduğunu göstermektedir. Öte yandan, segmentasyon sonuçlarındaki küçük sızıntılar (bleeding) veya komşu sınıflar (özellikle tozlu ve kuş lekeli bölgeler) arasındaki örtüşme, zaman zaman sınırlı performans düşüşlerine yol açmıştır.

Sınıflandırma aşamasında EfficientNetB0 tabanlı model, doğrulama kümesinde %85,79’luk genel başarı ve test kümesinde sınıf başına hesaplanan AUC = 0,97–1,00 aralığıyla yüksek ayırma gücü sergilemiştir. Karışıklık matrisleri, “Dusty” ve “Bird-drop” sınıfları arasındaki görsel benzerlik nedeniyle sınırlı sayıda yanlış sınıflandırmanın söz konusu olduğunu ortaya koymuştur. Buna karşın “Clean” ve “Physical-Damage” kategorileri, %90’ın üzerinde F1-skorlarıyla net ayrım göstermiştir.

**Sınırlılıklar ve Gelecek Çalışmalar**

* **Veri Çeşitliliği**: Mevcut veri setindeki görüntüler, ağırlıklı olarak laboratuvar veya kontrollü saha koşullarından derlenmiştir. Farklı ışık ve çevresel koşulları yansıtacak ilave saha görüntülerinin toplanması, modelin gerçek uygulama ortamlarında genelleme yeteneğini artıracaktır.
* **Dosya Ölçeklendirme**: U-Net segmentasyon modülündeki up-sampling adımları, yüksek çözünürlüklü ince detayları bazen net çıkaramamıştır. Daha derin decoder katmanları veya multi-scale atrous konvolüsyonlar (ör. DeepLabV3+) ile incelenebilir.
* **Entegre Öğrenme**: İki aşamalı pipeline’ın tek bir uçtan uca (end-to-end) mimariye dönüştürülmesi; segmentasyon ve sınıflandırma görevlerinin ortak kayıp fonksiyonlarıyla eş zamanlı optimize edilmesi; hem hız hem de performans açısından avantaj sağlayabilir.
* **Model Topluluğu ve İnce Ayar**: Farklı ağların (EfficientNet ailesinin daha büyük ölçekleri, ResNeSt vb.) ensembling stratejileriyle birleştirilmesi ve Bayesian hiperparametre optimizasyonunun devreye alınması, mevcut doğruluk ve AUC değerlerini daha da yukarı çekebilir.

**Sonuç**  
Önerilen ResNet50-U-Net + EfficientNetB0 yaklaşımı, güneş paneli görüntülerinde arıza tespiti alanında hem bölgesel hassasiyet hem de yüksek sınıflandırma doğruluğu sunarak pratik uygulamalara yakın performans sergilemiştir. Elde edilen deneysel bulgular, bakım süreçlerinin otomasyonunu ve maliyet etkin izlenmesini mümkün kılacak bir temel oluşturmakta; gelecekteki çalışmalarla saha veri çeşitliliğinin ve model bütünlüğünün artırılması, yakından takip edilmesi gereken önemli iyileştirme alanları olarak öne çıkmaktadır.

**KAYNAKLAR**

[1] K. He, X. Zhang, S. Ren, J. Sun, “Deep Residual Learning for Image Recognition,” CVPR, 2016.

[2] O. Ronneberger, P. Fischer, T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” MICCAI, 2015.

[3] M. Tan, Q. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” ICML, 2019.

[4] A. Paszke et al., “Albumentations: Fast and Flexible Image Augmentations,” CVPR Workshops, 2020.

[5] “Faulty Solar Panel Images,” Kaggle,

[6] T. Fawcett, “An introduction to ROC analysis,” Pattern Recognition Letters, vol. 27, no. 8, pp. 861–874, 2006.