İSTANBUL MEDENİYET ÜNİVERSİTESİ

DERİN ÖĞRENME DERSİ FİNAL PROJESİ

BLOODCELL-AI: KAN HÜCRESİ SINIFLANDIRMA PROJE RAPORU

Ad Soyad: Süeda Nur Sarıcan

Öğrenci No: 23120205031

Ders: Derin Öğrenme

GitHub Depo Adresi: https://github.com/suedasarican/BloodCell_CNN

1. Proje Konusu ve Uygulama Alanı

1.1. Seçilme Gerekçesi ve Önemi

Hematolojik analizler, tıbbi teşhis süreçlerinin temel taşlarından biridir. Periferik yayma testlerinde lökosit, eritrosit ve trombosit gibi hücrelerin doğru sınıflandırılması; lösemi, anemi ve çeşitli enfeksiyonların erken teşhisinde kritik rol oynar. Özellikle "Immature" (olgunlaşmamış) hücrelerin tespiti, akut kan hastalıklarının habercisi olabilir.

Geleneksel yöntemlerde uzmanlar tarafından mikroskop altında yapılan manuel sayımlar zaman alıcıdır ve insan yorgunluğuna bağlı hata risklerini barındırır. Bu projenin temel amacı, bu süreci yapay zeka ile otomatize ederek teşhis hızını ve doğruluğunu artırmaktır.

1.2. Literatürdeki Uygulamalar

Literatür incelendiğinde, geçmiş çalışmaların genellikle şekil ve renk histogramı gibi manuel öznitelik çıkarımına (feature extraction) dayandığı görülmüştür. Ancak kan hücrelerinin (özellikle Lenfosit ve Immature hücreler) görsel benzerliği, bu sığ (shallow) yöntemlerin başarısını kısıtlamıştır. Günümüzde ise Evrişimli Sinir Ağları (CNN), görsel öznitelikleri hiyerarşik olarak öğrenebildiği için medikal görüntü analizinde standart haline gelmiştir.

2. Veri Setinin Belirlenmesi

Projede, tıbbi görüntüleme araştırmalarında standart bir kıyaslama noktası olan BloodMNIST veri seti kullanılmıştır. Veri setinin istatistiksel dağılımı şu şekildedir:

- **Toplam Görüntü Sayısı:** 17,092 adet.
- **Görüntü Boyutu:** 28x28 piksel, RGB (3 kanal).
- **Sınıf Sayısı:** 8 farklı hücre türü (Nötrofil, Eozinofil, Basofil, Lenfosit, Monosit, Immature, Eritroblast, Trombosit).

Veri seti, modelin genelleme yeteneğini doğru ölçmek adına literatür standartlarına uygun olarak Eğitim (%70), Doğrulama (%10) ve Test (%20) kümelerine ayrılmıştır.

2.2. Veri Ön İşleme (Preprocessing)

Kod içerisinde (data_loader.py ve train.py) model performansını artırmak için şu işlemler uygulanmıştır:

1. **Normalizasyon**: [0, 255] aralığındaki piksel değerleri 255'e bölünerek [0, 1] aralığına çekilmiştir.

2. **Kanal Dönüşümü (Transpose):** Görüntüler standart (Yükseklik, Genişlik, Kanal) formatından, matris çarpımlarının optimize edilebilmesi için (Kanal, Yükseklik, Genişlik) formatına dönüştürülmüştür.
3. **Veri Artırma (Data Augmentation):** Eğitim sırasında aşırı öğrenmeyi (overfitting) engellemek için, her bir eğitim yığınına (batch) %50 olasılıkla rastgele yatay ve dikey çevirme (random flip) işlemleri uygulanmıştır.

3. Uygulanan Yöntem ve Seçim Gerekçesi

3.1. Literatürdeki Yöntemlerin Karşılaştırmalı Analizi

Kan hücresi sınıflandırma problemi için literatürde ve bu proje geliştirme sürecinde iki ana yaklaşım test edilmiştir: **Geleneksel Öznitelik Mühendisliği (MLP)** ve **Derin Öğrenme (CNN)**.

MLP: Bu yaklaşımda görüntülerden HOG (Histogram of Oriented Gradients) ve renk histogramları manuel olarak çıkarılıp Çok Katmanlı Algılayıcıya (MLP) verilmiştir.

- **Dezavantajı:** Hücrelerin karmaşık dokularını yakalamakta yetersiz kalmış ve 400 epoch gibi uzun bir eğitim süresi gerektirmiştir.
- **Sonuç:** %87.78 Test Başarısı.

CNN - Convolutional Neural Networks: Projede seçilen CNN mimarisi, öznitelikleri manuel işlem gerektirmeden ham piksel verisinden otomatik ve hiyerarşik olarak öğrenir.

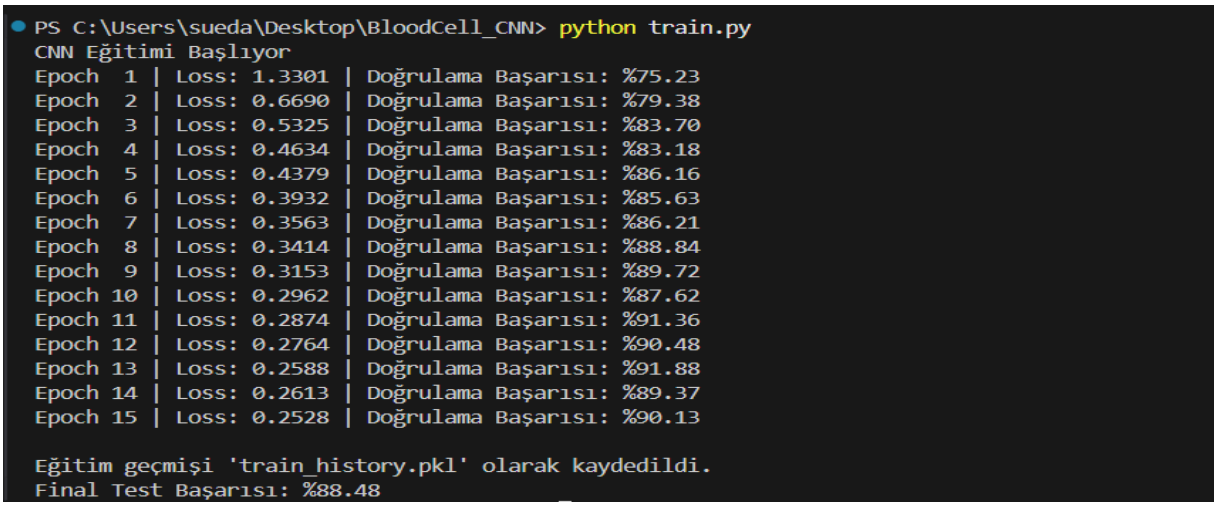
- **Avantajı:** "Uçtan uca öğrenme" (end-to-end learning) sayesinde, MLP modelinin 400 epoch'ta ulaştığı başarıyı sadece 15 epoch'ta geçmiştir.
- **Sonuç:** %88.48 Test Başarısı (En iyi doğrulama başarısı Epoch 13'te %91.88'e ulaşmıştır).

Bu karşılaştırma sonucunda; daha yüksek doğruluk, daha hızlı yakınsama ve manuel müdahaleyi ortadan kaldırması nedeniyle CNN yaklaşımı tercih edilmiştir.

```
PS C:\Users\sueda\Desktop\BloodCell_Project> python train.py
⏳ Veri setleri (Train, Val, Test) indiriliyor...
100%|████████████████████████████| 35.5M/35.5M [01:16<00:00, 466kB/s]
Veri Dağılımı -> Train: 11959, Val: 1712, Test: 3421
🚀 Veri Çoğaltma İşlemi (Giriş: 11959)...
 Özellikler hesaplanıyor.
⚙️47836 görüntü için HOG + Renk analizi yapılıyor...
⚙️1712 görüntü için HOG + Renk analizi yapılıyor...
⚙️3421 görüntü için HOG + Renk analizi yapılıyor...
Model Giriş Boyutu: 1326 (HOG+Renk+Histogram)
Eğitim Başlıyor (400 Epoch)...
Epoch 50/400 -> Loss: 0.6946 | Val Acc: %77.28
Epoch 100/400 -> Loss: 0.5101 | Val Acc: %81.25
Epoch 150/400 -> Loss: 0.4003 | Val Acc: %84.70
Epoch 200/400 -> Loss: 0.3195 | Val Acc: %86.21
Epoch 250/400 -> Loss: 0.2606 | Val Acc: %87.32
Epoch 300/400 -> Loss: 0.2165 | Val Acc: %88.61
Epoch 350/400 -> Loss: 0.1836 | Val Acc: %88.20
Epoch 400/400 -> Loss: 0.1589 | Val Acc: %88.55
Eğitim Tamamlandı!

========================================
🏆 FİNAL TEST BAŞARISI: %87.78
========================================
```

Şekil 1 : MLP Eğitim Süreci Terminal Çıktısı



Şekil 2: CNN Modelinin Eğitim Süreci

3.2. Matematiksel Altyapı ve Algoritmalar

Kod tarafında (utils.py ve model.py) aşağıdaki ileri teknikler uygulanmıştır:

- Modelin başlangıç ağırlıkları, gradyan kaybolması (vanishing gradient) problemini önlemek adına **He Initialization** yöntemiyle rastgele başlatılmıştır.
- **im2col (Image to Column):** Python döngülerinin yavaşlığını aşmak için, görüntü pikselleri im2col algoritması ile sütun vektörlerine dönüştürülmüş ve konvolüsyon işlemi matris çarpımına (Matrix Multiplication) indirgenerek vektörize edilmiştir.
- **Adam Optimizer:** Stokastik Gradyan İnişi (SGD) yerine, momentum ve RMSProp bileşenlerini içeren **Adam** algoritması $(\beta_1 = 0.9, \beta_2 = 0.999)$ elle kodlanarak modelin hızlı yakınsaması sağlanmıştır.

3.3. Model Mimarisi (BloodCell-AI)

Kodumuzda yer alan BloodCellAI sınıfı şu katmanlardan oluşmaktadır:

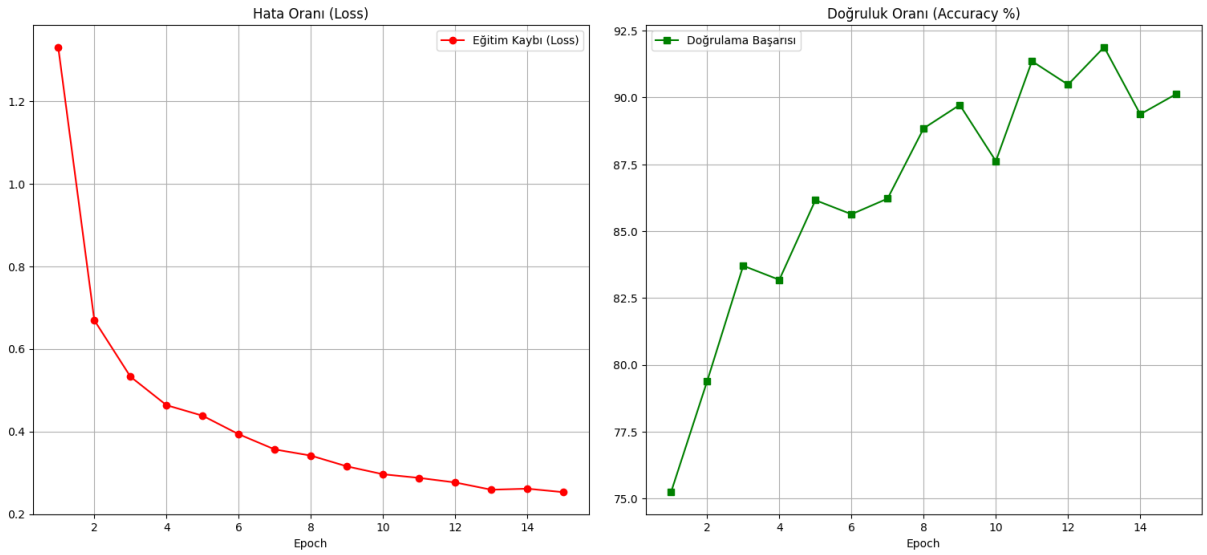
- **Giriş:** 3x28x28 Görüntü.
- **Conv Blok-1:** 32 Filtre (3x3), LeakyReLU Aktivasyonu, MaxPool (2x2).
- **Conv Blok-2:** 64 Filtre (3x3), LeakyReLU Aktivasyonu, MaxPool (2x2). *(Öznitelik haritası derinleştirilip boyutu 7x7'ye düşürülmüştür).*
- **Dense (Tam Bağlı) Katmanlar:** Flatten işleminden sonra 128 nöronlu gizli katman. Tam bağlı katman çıkışında lineerliği kırmak ve ölü nöron problemini engellemek için tekrar **LeakyReLU** aktivasyon fonksiyonu uygulanmıştır.
- **Çıkış:** 8 sınıf için logit değerleri üretilir ve **Softmax** ile olasılığa dönüştürülür (İşlem sırasında sayısal taşmaları önlemek için Numerical Stability tekniği uygulanmıştır).

4. Model Eğitimi ve Değerlendirme

4.1. Eğitim Süreci

Model, train.py modülü üzerinden **15 Epoch** boyunca, 128'lik batch boyutu ve 0.001 öğrenme oranı ile eğitilmiştir.

- **Kayıp Fonksiyonu:** Çoklu sınıflandırma problemi olduğu için **Categorical Cross-Entropy** formülü manuel olarak uygulanmıştır.
- **Sonuç:** Model, MLP (Çok Katmanlı Algılayıcı) denemelerine kıyasla çok daha hızlı öğrenerek, 15 epoch sonunda eğitim kaybını minimize etmiştir.



Şekil 3 : Loss ve Accuracy Grafikleri

4.2. Performans Sonuçları

Test seti (3,421 görüntü) üzerinde yapılan kapsamlı değerlendirme sonucunda **%88,48** oranında Genel Doğruluk (Accuracy) elde edilmiştir. Ancak modelin başarısını sadece genel doğruluk üzerinden değil, her bir hücre tipine özgü ayırt ediciliğini de göz önüne alarak değerlendirmek gerekmektedir. Bu bağlamda, veri setindeki sınıflar için ayrı ayrı hesaplanan **Kesinlik (Precision)**, **Duyarlılık (Recall)** ve **F1-Skoru** değerlerini içeren detaylı performans metrikleri Tablo 1'de sunulmuştur:

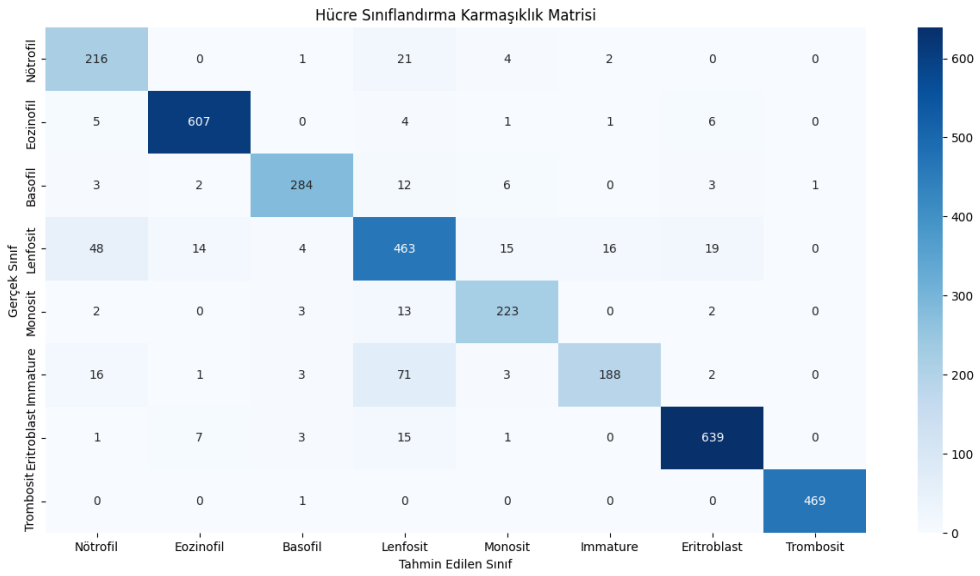
Tablo 1. BloodCell-AI Modelinin Sınıf Bazlı Performans Metrikleri

Hücre Türü	Precision (Kesinlik)	Recall (Duyarlılık)	F1-Skoru	Örnek Sayısı
Nötrofil	0.74	0.89	0.81	244
Eozinofil	0.96	0.97	0.97	624
Basofil	0.95	0.91	0.93	311
Lenfosit	0.77	0.80	0.79	579
Monosit	0.88	0.92	0.90	243
Immature	0.91	0.66	0.77	284
Eritroblast	0.95	0.96	0.96	666
Trombosit	1.00	1.00	1.00	470
GENEL (Accuracy)			0.88	3421

- **En Yüksek Başarı:** Trombosit (%100 Recall) ve Eozinofil (%97 Recall).
- **En Düşük Başarı:** Immature (%66 Recall).

4.3. Hata Analizi

Karmaşıklık matrisi incelendiğinde, modelin en çok Immature hücreleri Lenfosit ile karıştırdığı görülmüştür (**71 adet hata**). Modelin optimize edilmesiyle bu hata sayısı önceki denemelere (95 hata) göre düşürülmüş olsa da bu iki hücre tipinin morfolojik benzerliği nedeniyle bir miktar karışıklık devam etmektedir. Tıbbi literatürde bu iki hücre tipi morfolojik olarak (çekirdek/sitoplazma oranı) birbirine çok benzediği için, bu durum beklenen bir zorluktur ve modelin biyolojik gerçeklikle uyumlu davrandığını gösterir.



Şekil 4 : Karmaşıklık Matrisi

5.Sonuç

Bu çalışma, derin öğrenme modellerinin manuel öznitelik çıkarımı gerektiren geleneksel yöntemlere göre çok daha verimli ve başarılı olduğunu kanıtlamıştır. Sıfırdan NumPy ile geliştirilen model, test setinde **%88.48**, eğitim sırasında doğrulama setinde ise **%91.88** gibi yüksek bir başarı oranına ulaşarak akademik hedeflerini başarıyla tamamlamıştır.