**T.C**

**SİİRT ÜNİVERSİTESİ**

**MÜHENDİŞLİK FAKÜLTESİ**

**BİLGİSİYAR MÜHENDİSLİĞİ**

**Keras ve TensorFlow Kullanarak Malaria (Sıtma) Varlığını Tespit Eden CNN Modal**

*MUHAMMAD SHEHARYAR BUTT*

*180304118*

**DATASET:** (Kaggle) **-** National Library of Medicine (Malariya Results 27558 İmages)

**SINIFLANDIRMA:** Keras, TensorFlow CNN Modeli, Adam Classifiers ve Dense, Flatten, Conv2D Filters

**SONUÇLAR:** Train Accuracy - %96.19

Validation Accuracy - %95.75

**YAPAY SINIR AĞLARI FINAL PROJESİ**

Dr. Öğr. Üyesi MAHMUT KAYA

**Keras ve TensorFlow Kullanarak Malaria (Sıtma) Varlığını Tespit Eden CNN Modal**

**ÖZET:** Bu projede, Keras ve TensorFlow kullanılarak Derin Sinir Ağları (CNN) ile bir veri kümesinde Malaria varlığını tespit etmeyi amaçladık. Veri setimizde sağlıklı ve enfekte olan hücre görüntülerini içeren bir dizi bulunmaktadır. CNN modeli, bu görüntüler üzerinde eğitilerek sağlıklı ve enfekte hücreleri doğru bir şekilde sınıflandırmayı öğrenmiştir. Veri ön işleme adımları, modelin doğruluğunu artırmak için yapılmıştır. Sonuçlarımız, CNN'in Malaria varlığını tespit etmede başarılı olduğunu göstermiş ve bu teknolojinin tıbbi teşhis süreçlerinde potansiyel bir yardımcı olarak kullanılabileceğini göstermiştir.

## Giriş

Tıbbi görüntü analizi, sağlık sektöründe önemli bir yer tutmakta ve hastalıkların teşhisinde değerli bir araç oluşturmaktadır. Bu alanda, özellikle Malaria gibi yaygın ve tehlikeli hastalıkların tespiti önem arz etmektedir. Malaria, özellikle gelişmekte olan ülkelerde ciddi bir sağlık sorunudur ve erken teşhisin önemi büyüktür.

Bu bağlamda, veri madenciliği ve makine öğrenimi teknikleri, tıbbi görüntü analiziyle Malaria gibi hastalıkların teşhisinde önemli bir rol oynamaktadır. Derin öğrenme yöntemleri, özellikle Convolutional Neural Networks (CNN) gibi, tıbbi görüntülerde hastalık tespiti için başarılı bir şekilde kullanılmaktadır. Bu yöntemler, büyük miktarda görüntü verisi üzerinde eğitilerek hastalık teşhisinde önemli başarılar elde etmiştir.

Bu çalışmada, Keras ve TensorFlow gibi derin öğrenme kütüphanelerini kullanarak bir CNN modeli geliştirilerek, Malaria varlığını tespit etme amacı güdülmüştür. Geliştirilen modelin, sağlıklı ve enfekte hücre görüntülerini doğru bir şekilde sınıflandırarak Malaria teşhisine yardımcı olması hedeflenmiştir. Bu çalışma, tıbbi görüntü analizi alanında veri madenciliği tekniklerinin Malaria teşhisi üzerindeki etkisini incelemeyi ve bu alanda yapılan çalışmalara katkıda bulunmayı amaçlamaktadır. Kaynaklar belirtilerek, konunun önemi ve projenin amaçları detaylı bir şekilde açıklanacaktır.

## Materyal ve Yöntemi

Kullandığı tüm materyal ve nasıl işlemler devam ettiği burada bilgilendirdim.

### Veri Kümesi (Dataset)

Bu çalışmada kullanılan veri kümesi, Malaria varlığını tespit etmek amacıyla toplanmıştır. Toplamda **13,780 parazit içeren** ve **13,780 sağlıklı (enfekte olmayan**) hücre görüntüsünden oluşmaktadır. Her bir görüntü, benzersiz özelliklere sahip olup, boyutları ve çözünürlükleri aynıdır.

Veri kümesi, toplamda **27,560 görüntü** örneği içermektedir. Görüntüler, **64x64** piksel boyutunda ve renkli (RGB) formatında olup, toplamda üç nitelik içermektedir: genişlik, yükseklik ve renk kanalları (Height, Width, Channels)

Veri kümesi, eğitim ve test olarak iki ayrı bölüme ayrılmıştır. Eğitim seti, toplam veri kümesinin **%90'ini** oluşturacak şekilde seçilmiştir. Geriye kalan **%10'luk** kısım ise test seti olarak kullanılmıştır. Bu şekilde, model eğitilirken kullanılan veri miktarı ile modelin genelleme yeteneği test edilebilmiştir. Veri kümesinin bu şekilde bölünmesi, eğitim ve test aşamalarında güvenilir sonuçlar elde etmek adına yapılmıştır.

### Normalizasyon

Görüntü verisi genellikle 0 ile 255 arasında değerlerle temsil edilir; burada her bir değer bir pikselin ışık yoğunluğunu gösterir. Bu değerlerin 0 ile 1 arasında ölçeklenmesi, derin öğrenme modelleri gibi bazı makine öğrenimi algoritmaları için daha uygun bir giriş oluşturur.

Bu normalizasyon işlemi, piksel değerlerinin 0 ile 1 arasında ölçeklenmiş bir forma getirilmesini sağlayarak, modelin daha hızlı ve daha etkili bir şekilde eğitilmesine olanak tanırken, aynı zamanda modelin genelleme yeteneğini de artırabilir.

### Sınıflandırma

**CNN Modeli:** Veri setindeki görüntüleri sınıflandırmak için derin öğrenme yaklaşımı olan CNN kullanılmıştır. Model, evrişimli katmanlar, tam bağlantılı katmanlar ve aktivasyon fonksiyonları içerir. Eğitim ve test doğruluklarına dayalı olarak performansı değerlendirilmiştir.

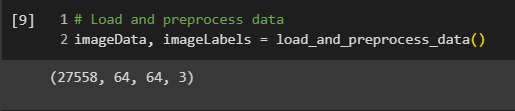
## Deneyesel Sonuçlar

Yaptığı çalışmaları nasıl nelerdir veya model öğrenmeye ile ne kadar iyi bir sonuç aldığı detaylı burada bilgilendirdim.

### Veri Yükleme ve Ön İşleme

İlk adımda, veri yükleme ve ön işleme işlemleri gerçekleştirildi. **'load\_and\_preprocess\_data()'** fonksiyonu kullanılarak, Malaria tespiti için hücre görüntüleri içeren bir veri seti yüklenmiş ve işlenmiştir. Veri seti iki farklı sınıfa (Enfekte Edilmiş ve Enfekte Edilmemiş) ayrılmıştır. Görüntüler, boyutlarını (64, 64) olarak 3 Kanal RGB yeniden şekillendirilmiş ve normalleştirilmiştir.

Normalizasyon için Görüntü verisi genellikle 0 ile 255 arasında değerlerle temsil edilir. Burada her bir pikseli 255 ile bölme yaptığından bizim elemizde 0-1 arasında normalize ettiği verileri alındı. Fonksyonu çıktısı Resim.1(a) gösterildi.



Resim.1(a)

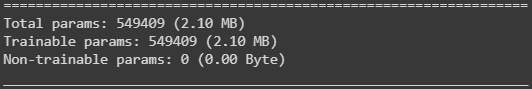
Çıktığıdan aldığı sonuçta toplam görüntü sayısı 27558, piksel boyutu 64x64 ve RGB 3 Kanal olmalıdır. Her bir görüntüdeki piksel 255 ile bölme yaparken yeni normalize ettiği 0-1 arasında data array oluşturmuş.

### Modelin Oluşturulması ve Eğitimi

Eğitim sürecinde, **build\_train\_and\_evaluate\_model()** fonksiyonu kullanılarak, Convolutional Neural Network (CNN) modeli tanımlanıp oluşturuldu. Model, evrişimli katmanlar, dropout katmanları ve tam bağlantılı (dense) katmanlar içerir. Bu model, veri seti üzerinde 10 epoch boyunca ve 300 sayılı batch boyutu eğitildi ve Adam optimizer kullanılarak binary\_crossentropy loss fonksiyonu ile derlendi. Model paramtere sayısı çıktısı **Resim.2(a) ve Tablo.1(a)** gösterildi.

|  |  |  |
| --- | --- | --- |
| Layer (type) | Output Shape | Param # |
| conv2d (Conv2D) | (None, 62, 62, 32) | 896 |
| conv2d\_1 (Conv2D) | (None, 60, 60, 32) | 9248 |
| conv2d\_2 (Conv2D) | (None, 28, 28, 64) | 18496 |
| conv2d\_3(Conv2D) | (None, 26, 26, 64) | 36928 |
| conv2d\_4 (Conv2D) | (None, 11, 11, 128) | 73856 |
| conv2d\_5 (Conv2D) | (None, 9, 9, 128) | 147584 |
| dense (Dense) | (None, 128) | 262272 |
| dense\_1 (Dense) | (None, 1) | 129 |

**Tablo.1(a)**



**Resim.2(a)**

#### Epoch İşlemler

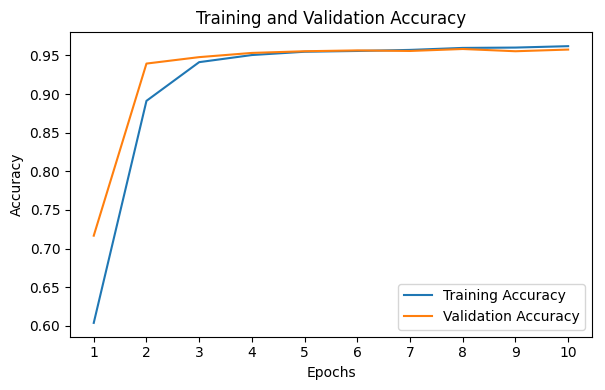
"Epoch" bir derin öğrenme modelinin eğitimi sırasında kullanılan bir terimdir. Bir epoch, eğitim veri setinin tamamının model tarafından bir kere geçirilmesini ifade eder. Bir eğitim döngüsü boyunca, veri seti belirli bir batch boyutuna ayrılır ve model, her bir batch'teki verileri kullanarak güncellenir. Tüm veri seti bu şekilde işlenir ve bir epoch tamamlanmış olur. **Tablo.2(b)** çalışmaları gösterildi.

|  |  |
| --- | --- |
| EPOCH # | RESULT |
| EPOCH 1 | 539s 6s/step - loss: 0.6572 - accuracy: 0.6037 - val\_loss: 0.5516 - val\_accuracy: 0.7166 |
| EPOCH 2 | 528s 6s/step - loss: 0.2687 - accuracy: 0.8911 - val\_loss: 0.2004 - val\_accuracy: 0.9394 |
| EPOCH 3 | 521s 6s/step - loss: 0.1753 - accuracy: 0.9413 - val\_loss: 0.1625 - val\_accuracy: 0.9478 |
| EPOCH 4 | 525s 6s/step - loss: 0.1572 - accuracy: 0.9504 - val\_loss: 0.1523 - val\_accuracy: 0.9532 |
| EPOCH 5 | 517s 6s/step - loss: 0.1415 - accuracy: 0.9547 - val\_loss: 0.1459 - val\_accuracy: 0.9554 |
| EPOCH 6 | 519s 6s/step - loss: 0.1368 - accuracy: 0.9557 - val\_loss: 0.1415 - val\_accuracy: 0.9565 |
| EPOCH 7 | 526s 6s/step - loss: 0.1309 - accuracy: 0.9571 - val\_loss: 0.1368 - val\_accuracy: 0.9557 |
| EPOCH 8 | 520s 6s/step - loss: 0.1246 - accuracy: 0.9597 - val\_loss: 0.1373 - val\_accuracy: 0.9583 |
| EPOCH 9 | 521s 6s/step - loss: 0.1228 - accuracy: 0.9601 - val\_loss: 0.1330 - val\_accuracy: 0.9554 |
| EPOCH 10 | 520s 6s/step - loss: 0.1149 - accuracy: 0.9619 - val\_loss: 0.1232 - val\_accuracy: 0.9575 |

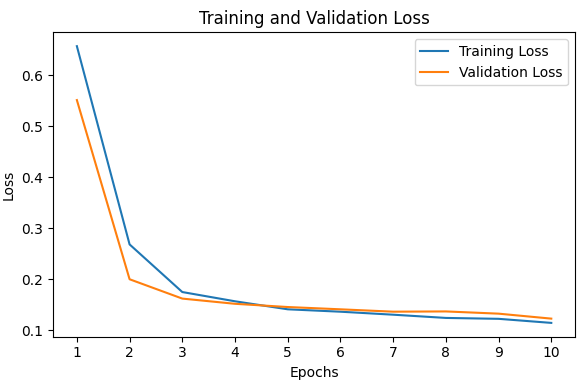
**Tablo.2(b)**

### Eğitim Sürecinin Sonuçları

Eğitim süreci boyunca, her epoch için eğitim ve doğrulama (validation) kayıpları ile doğruluk değerleri takip edilmiştir. Bu değerlerin grafiği, **plot\_accuracy\_loss()** fonksiyonu aracılığıyla çizilmiştir. Bu grafik **Graf.1(a) ve Graf.1(b)**, eğitim sürecinin başarısını ve overfitting durumunu göstermektedir.



**Graf.1(a)**

****

**Graf.1(b)**

Eğitim sürecinin sonuçlarını özetlemek adına, **print\_last\_epoch\_metrics()** fonksiyonu kullanıldı. Bu fonksiyon, son epoch'un eğitim ve doğrulama kayıplarını ile doğruluk değerlerini ekrana yazdırır. Bu metrikler, son eğitim aşamasındaki model performansını text olarak **Tablo.3(a) ve Tablo.3(b)** gösterir:

|  |  |
| --- | --- |
| TRAINING ACCURACY | VALIDATION ACCURACY |
| 0.9619 (%96.19) | 0.9575 (%95.75) |

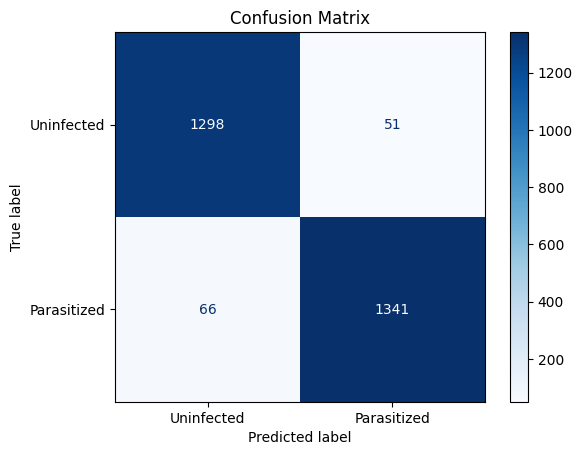
**Tablo.3(a) *Accuracy Tablo***

|  |  |
| --- | --- |
| TRAINING LOSS | VALIDATION LOSS |
| 0.1149 (%11.49) | 0.1232 (%12.32) |

**Tablo.3(b) *Loss Tablo***

### Confusion Matrix ve Performans Değerlendirmesi

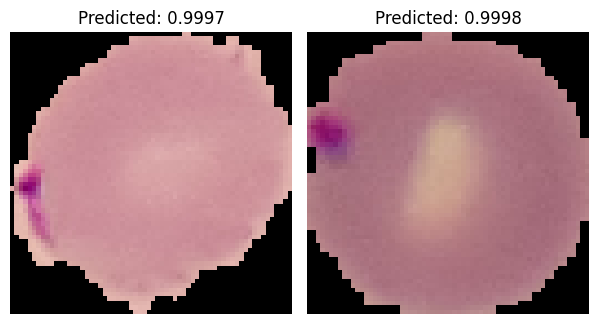
Eğitilen modelin performansı, **display\_confusion\_matrix()** fonksiyonu ile confusion matrix (karmaşıklık matrisi) olarak **Resim.3(a)** görselleştirilmiştir. Bu matris, modelin sınıflandırma başarısını (doğru ve yanlış sınıflandırmaları) göstermektedir. Ayrıca, accuracy, precision, recall gibi performans metrikleri hesaplanabilir.



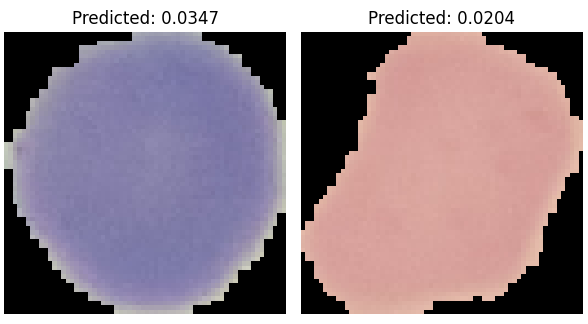
**Resim.3(a)**

### Örnek Görüntülerin Tahmin Edilmesi

Eğitim sürecinin ardından, modelin daha önce görmemiş olduğu, yalnızca test amaçlı ayrılmış olan görüntüler üzerinde değerlendirme işlemi gerçekleştirildi. Bu adım, **display\_sample\_images()** fonksiyonu aracılığıyla gerçekleştirildi. Fonksiyon, test veri setinde bulunan ancak modele daha önce sunulmamış olan görüntüleri kullanarak, modelin sınıflandırma yeteneğini test etti. Sonuçtan aldığı resimler ağşada **Resim.4(a) ve Resim.4(b)** gösterildi.



**Resim.4(a)**

****

**Resim.4(b)**

## Sonuç

Keras ve TensorFlow kullanılarak Derin Sinir Ağları (CNN) ile Malaria tespiti gerçekleştirildi. Sağlıklı ve enfekte hücre görüntülerinden oluşan veri setinde model eğitilerek, sağlıklı ve enfekte hücreleri doğru bir şekilde sınıflandırmayı öğrendi.

27.558 adet 64x64 boyutunda RGB görüntü içeren veri setinde veya toplam katman paramtere 549.409 eğitilen CNN modeli, yüksek bir doğruluk seviyesi elde etti. Modelin eğitim doğruluğu %96.19 ve doğrulama doğruluğu %95.75 olarak belirlendi. Bu sonuçlar, modelin sağlıklı ve enfekte hücreleri doğru bir şekilde ayırt edebildiğini gösteriyor.

Ayrıca, düşük eğitim ve doğrulama kayıpları (sırasıyla 0.1149 ve 0.1232), modelin performansını daha da güçlendiriyor. Bu sonuçlar, modelin Malaria varlığını tespit etme konusunda başarılı olduğunu ve tıbbi teşhis süreçlerinde potansiyel bir yardımcı olarak kullanılabileceğini gösteriyor.

## Kaynaklar

* <https://lhncbc.nlm.nih.gov/LHC-downloads/downloads.html#malaria-datasets>
* <https://www.kaggle.com/datasets/iarunava/cell-images-for-detecting-malaria>
* <https://github.com/amit-raj-repo/CNN-Malaria-Detection-Using-Keras/tree/master>
* <https://www.tensorflow.org/tutorials/images/cnn>
* <https://ocakhasan.github.io/evrisimsel-sinir-aglari-nedir/>