

Igor Parreira de Paula

# **Classificação de Células Sanguíneas para Diagnóstico de Malária utilizando MLP**

Goiânia

2025

Igor Parreira de Paula

# **Classificação de Células Sanguíneas para Diagnóstico de Malária utilizando MLP**

Relatório técnico apresentado à disciplina de Inteligência Artificial (2025.2-CMP1110/A01), ministrada pelo Prof. Clarimar Jose Coelho, como requisito parcial para avaliação da N2.

Pontifícia Universidade Católica de Goiás – PUC Goiás  
Escola Politécnica  
Curso de Engenharia de Computação

Orientador: Prof. Clarimar Jose Coelho

Goiânia  
2025

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>3</b>
<b>1.1</b>	<b>PROBLEMA DE PESQUISA</b>	<b>3</b>
<b>1.2</b>	<b>OBJETIVOS</b>	<b>3</b>
1.2.1	Objetivo Geral	3
1.2.2	Objetivos Específicos	3
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>4</b>
<b>3</b>	<b>METODOLOGIA</b>	<b>5</b>
<b>4</b>	<b>RESULTADOS E DISCUSSÃO</b>	<b>6</b>
<b>4.1</b>	<b>CURVAS DE APRENDIZADO</b>	<b>6</b>
<b>4.2</b>	<b>AVALIAÇÃO DO DESEMPENHO</b>	<b>6</b>
4.2.1	Matriz de Confusão	6
4.2.2	Métricas Detalhadas	7
<b>4.3</b>	<b>CURVA ROC E AUC</b>	<b>7</b>
<b>5</b>	<b>CONCLUSÃO</b>	<b>9</b>

# 1 INTRODUÇÃO

A malária é uma doença infecciosa grave causada por parasitas do gênero *Plasmodium*. O diagnóstico tradicional baseia-se na análise manual de imagens microscópicas de esfregaços sanguíneos, um processo que exige especialistas treinados e é propenso a erros humanos.

Soluções baseadas em Inteligência Artificial, especificamente Redes Neurais Artificiais (RNAs), apresentam grande potencial para automatizar esse processo, oferecendo diagnósticos rápidos e com boa acurácia. Este trabalho propõe o desenvolvimento de um pipeline de classificação utilizando uma rede neural do tipo *Multilayer Perceptron* (MLP) para distinguir entre células infectadas (*Parasitized*) e não infectadas (*Uninfected*).

## 1.1 PROBLEMA DE PESQUISA

Como desenvolver e validar um modelo de rede neural MLP capaz de classificar corretamente imagens microscópicas de células sanguíneas, diferenciando casos de malária de células saudáveis, utilizando apenas os valores de intensidade dos pixels como entrada?

## 1.2 OBJETIVOS

### 1.2.1 Objetivo Geral

Desenvolver um pipeline completo de classificação de imagens microscópicas para distinguir células sanguíneas infectadas e não infectadas por malária utilizando uma rede neural MLP.

### 1.2.2 Objetivos Específicos

- Pré-processar o conjunto de dados, incluindo redimensionamento para 64x64 pixels e conversão para escala de cinza;
- Implementar uma arquitetura MLP com camadas densas e função de ativação ReLU;
- Treinar o modelo utilizando o otimizador Adam e a função de perda *Binary Cross-entropy*;
- Avaliar o desempenho do modelo através de métricas como Acurácia, Matriz de Confusão e Curva ROC.

## 2 REFERENCIAL TEÓRICO

O conjunto de dados utilizado contém imagens de células sanguíneas organizadas em duas classes: *Parasitized* (infectadas) e *Uninfected* (não infectadas). As imagens estão em formato PNG e requerem pré-processamento para serem utilizadas em redes neurais densas.

A *Multilayer Perceptron* (MLP) é uma classe de rede neural *feedforward* que consiste em pelo menos três camadas de nós: uma camada de entrada, uma camada oculta e uma camada de saída. Exceto pelos nós de entrada, cada nó é um neurônio que utiliza uma função de ativação não linear. O aprendizado ocorre através do algoritmo de *backpropagation*.

### 3 METODOLOGIA

A metodologia adotada consistiu nas seguintes etapas, implementadas em linguagem Python com a biblioteca TensorFlow/Keras:

1. **Aquisição e Pré-processamento:** As imagens foram carregadas das pastas *Parasitized* e *Uninfected*. Cada imagem foi convertida para escala de cinza e redimensionada para  $64 \times 64$  pixels. Os valores dos pixels foram normalizados para o intervalo  $[0, 1]$ .
2. **Vetorização:** As imagens 2D ( $64 \times 64$ ) foram transformadas em vetores unidimensionais de tamanho 4096 para alimentar a camada de entrada da MLP.
3. **Divisão dos Dados:** O dataset foi dividido em conjuntos de treino (80%) e teste (20%), de forma estratificada.
4. **Arquitetura da Rede:** A Tabela 1 detalha a estrutura da rede neural implementada, totalizando mais de 2,2 milhões de parâmetros treináveis.

Tabela 1 – Resumo da Arquitetura do Modelo MLP

Camada (Tipo)	Formato de Saída	Parâmetros #
dense (Dense)	(None, 512)	2.097.664
dense_1 (Dense)	(None, 256)	131.328
dense_2 (Dense)	(None, 128)	32.896
dense_3 (Dense)	(None, 1)	129
<b>Total de Parâmetros</b>		<b>2.262.017</b>

Fonte: Código Python.

5. **Treinamento:** O modelo foi treinado por 20 épocas, com batch size de 32, utilizando Otimizador Adam e Loss Binary Crossentropy.

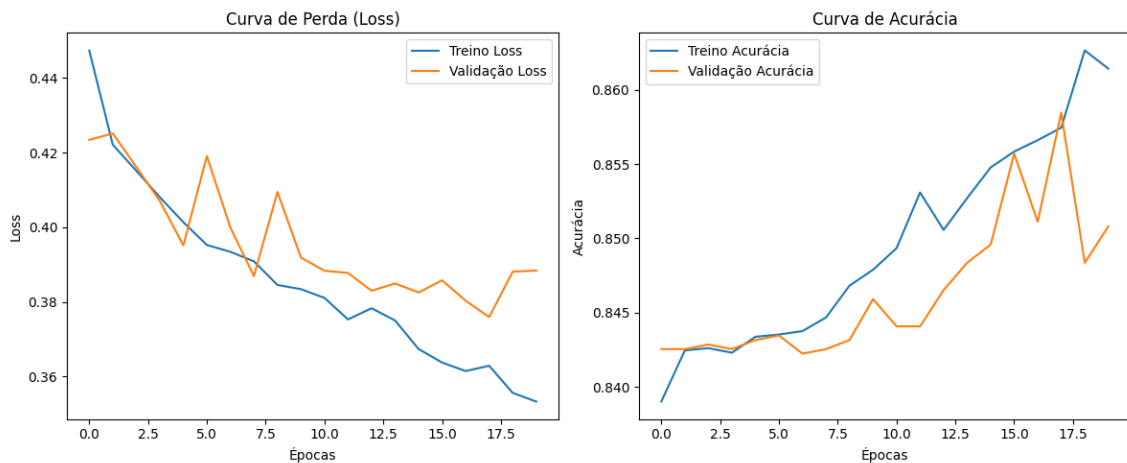
## 4 RESULTADOS E DISCUSSÃO

Nesta seção, são apresentados os resultados obtidos após o treinamento da rede neural.

### 4.1 CURVAS DE APRENDIZADO

A Figura 1 apresenta a evolução da perda (loss) e da acurácia durante as 20 épocas de treinamento.

Figura 1 – Curvas de Perda e Acurácia durante o Treinamento



Fonte: Código Python.

Observa-se que a acurácia de validação (linha laranja) estabiliza em torno de 85%, enquanto a perda de validação começa a oscilar e subir levemente após a época 15, indicando um início de *overfitting*, fenômeno comum dada a simplicidade da MLP para dados complexos de imagem.

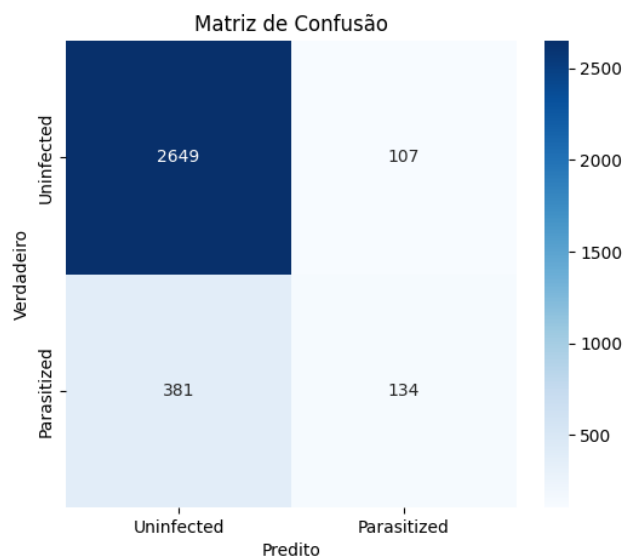
### 4.2 AVALIAÇÃO DO DESEMPENHO

O modelo atingiu uma acurácia global de 85% no conjunto de teste. No entanto, a análise detalhada revela um desbalanceamento significativo.

#### 4.2.1 Matriz de Confusão

A Figura 2 mostra a matriz de confusão obtida no conjunto de teste.

Figura 2 – Matriz de Confusão



Fonte: Código Python.

A matriz revela que o modelo é excelente em identificar células não infectadas (2649 acertos), mas falha significativamente em identificar células infectadas (apenas 134 acertos contra 381 falsos negativos). Isso ocorre devido ao desbalanceamento do dataset (13.779 *Uninfected* vs 2.575 *Parasitized*).

#### 4.2.2 Métricas Detalhadas

Conforme os logs de execução, as métricas por classe foram:

- **Uninfected:** Precision: 0.87, Recall: 0.96.
- **Parasitized:** Precision: 0.56, Recall: 0.26.

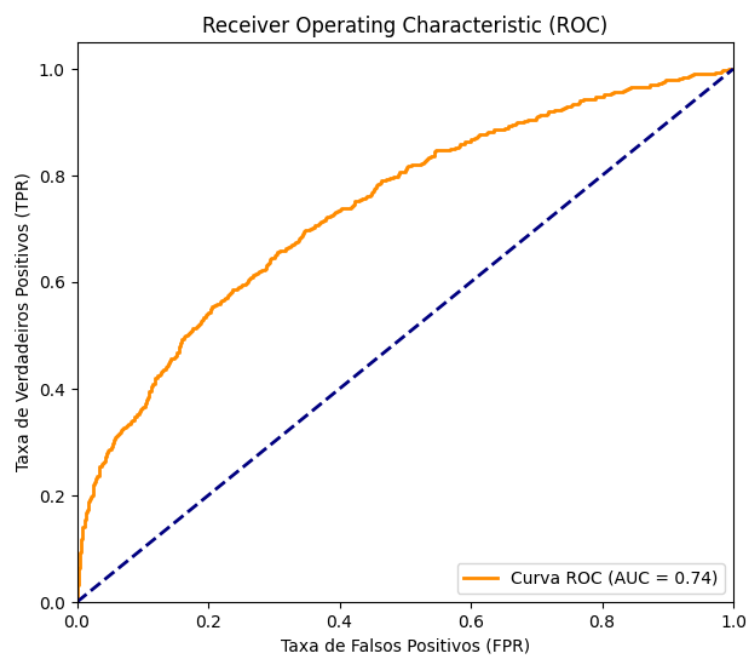
O baixo *recall* (0.26) para a classe *Parasitized* torna o modelo inadequado para uso clínico imediato, pois muitos doentes não seriam diagnosticados.

### 4.3 CURVA ROC E AUC

A Curva ROC (Receiver Operating Characteristic) é apresentada na Figura 3.



Figura 3 – Curva ROC



Fonte: Código Python.

A Área Sob a Curva (AUC) obtida foi de 0.74, o que indica uma capacidade de discriminação razoável, mas que pode ser melhorada com técnicas de balanceamento de dados (como *oversampling* ou *class weights*) ou uso de redes convolucionais (CNNs).

## 5 CONCLUSÃO

O trabalho atingiu o objetivo de desenvolver um pipeline de classificação MLP. Embora a acurácia global de 85% pareça promissora, a análise aprofundada mostrou que o modelo é enviesado para a classe majoritária.