

Universidade Federal de Santa Catarina (UFSC)

Centro Tecnológico – CTC

Departamento de Engenharia Elétrica e Eletrônica – DEEL



Projeto Final

## **Classificação de Leucemia**

EEL7513 – Tópicos Avançados em Controle processos de  
sinais IV (Aprendizado de Máquina)

Semestre 23.1

Bruno Mingoti

Matrícula: 20204165

# Sumário

<b>1. Introdução.....</b>	<b>3</b>
<b>2. Trabalhos relacionados .....</b>	<b>3</b>
<b>3. Dataset.....</b>	<b>5</b>
<b>4. Metodologia .....</b>	<b>6</b>
4.1 Preparação dos Dados .....	6
4.2 Análise Exploratória dos Dados (EDA).....	6
4.3 Separação dos conjuntos de dados.....	7
4.4 Criação dos <i>datasets</i> e pré-processamento .....	7
4.5 Métricas.....	8
4.6 Escolha da Arquitetura .....	8
4.7 Class weight .....	9
4.8 Data Augmentation.....	9
4.9 Treinamento com o conjunto de dados de treino completo.....	10
4.10 Aplicação da arquitetura em imagem HSV .....	11
4.11 Ensemble Averaging .....	11
4.12 Retreinamento .....	12
<b>5. Avaliação e Resultados .....</b>	<b>12</b>
5.1 Matriz de Confusão .....	12
5.2 Imagens Erradas .....	13
5.3 Comparação com a performance de outros modelos da literatura.....	13
<b>6. Conclusão.....</b>	<b>14</b>
<b>7. Referências.....</b>	<b>14</b>

## 1. Introdução

A Leucemia é uma doença maligna dos glóbulos brancos, geralmente de origem desconhecida, que tem como principal característica o acúmulo de células defeituosas na medula óssea, que substituem as células sanguíneas normais. Essa doença pode ser classificada em quatro grupos: *acute lymphoblastic leukemia* (ALL), *acute myeloid leukemia* (AML), *chronic lymphocytic leukemia* (CLL) e *chronic myeloid leukemia* (CML). Globalmente em 2015 a doença teve uma incidência alta de 876.000 pessoas diagnosticadas e uma taxa de mortalidade de 110.500 pessoas, principalmente crianças.

Seu diagnóstico ocorre por meio de exames de sangue ou por biópsia da medula óssea, porém grande parte dos diagnósticos ocorrem em estágios mais avançados, quando já existe uma grande quantidade de células malignas presentes no sangue. Contudo, uma vez a doença detectada nas fases iniciais, sua taxa de cura é de 80%, o que demonstra a importância do diagnóstico precoce.

A classificação por processamento de imagens de células com leucemia ganhou bastante interesse nos últimos anos, sendo inclusive tema de desafio em classificação de células cancerosas por meio de imagens. Há um interesse ainda maior pelo tema devido à dificuldade em diferenciar células malignas e saudáveis, uma vez que ambas possuem alta similaridade morfológica.

Esse artigo visa elaborar um modelo de aprendizado de máquina capaz de classificar células saudáveis e células com leucemia do tipo ALL e está dividido da seguinte forma: Seção 2 discute os trabalhos relacionados já realizados na área. Seção 3 apresenta informações sobre o dataset. Seção 4 a metodologia aplicada. Seção 5 fornece os resultados obtidos no dataset de teste. Seção 6 contém a conclusão contendo os principais pontos observados durante a execução. E Seção 7 consta todas as referências.

## 2. Trabalhos relacionados

Por se tratar de um problema advindo de um desafio, há vários artigos na literatura com diferentes propostas de resolução ao problema proposto de classificação de células com Leucemia.

Nayak [7] realizou modificações na arquitetura VGG16 com *batch normalization* no dataset pré-treinado da ImageNet. Resultados experimentais mostraram 80% de weighted F1-Score.

Marzahl et al. [8] propôs a utilização da ResNet18 com vários métodos de data augmentation e incorporou um mecanismo básico de atenção baseado em uma região da sub-rede proposta para impulsionar o resultado, obtendo um weighted F1-Score de 82,84% no dataset de teste final.

Kulhalli et al. [3] Propôs um método de segmentação de glóbulos brancos imaturos de imagens microscópicas e, utilizando um *stain deconvolutional layer*, obteve uma acurácia de 93,2% na tarefa de diferenciar entre células saudáveis e malignas.

Ding et al. [6] utilizou uma técnica de ensemble denominada Stacking que combina múltiplos modelos classificadores por meio de um *meta-classifier*, obtendo um weighted F1-Score de 85,52%.

Pan et al. [10] Propôs um *neighborhood-correction algorithm* (NCA), que consistia em ajustar (fine-tuning) uma rede residual pré-treinada usando dados de treinamento e produzindo rótulos iniciais e *feature maps* para os dados de teste; construção de um Vetor de Fisher para cada imagem com base nesses *feature maps*; e correção do rótulo inicial de cada imagem de teste por meio do *Classifier Voting* ponderada com base em seus vizinhos semelhantes. O modelo atingiu um weighted F1-Score de 91,04% no teste final, classificando em primeiro lugar no desafio do C-NMC na época.

Singhal et al. [13] Utilizou um SVM para a detecção automática das células malignas baseada na geometria de suas *features* e no *local binary pattern* (LBP) de suas texturas.

Mourya et al. [12] Elaborou um modelo denominado LeukoNet, que combina *features* de domínio *Discrete Cosine Transform* (DCT) extraído via CNN com recursos de espaço de *Optical Density* (OD) para a construção de um modelo mais robusto.

Kramer [3] aplicou a arquitetura ResNeXt com módulos de *Squeeze-and-Extraction*, atingindo um weighted F1-Score de 88,91%.

Kassani et al. [1] Elaborou um modelo que faz uso de um híbrido de duas CNNs e, diferentemente de outras abordagens da literatura, ao invés de utilizar características profundas extraídas de todas as arquiteturas pré-treinadas, decidiu fundir os recursos de camadas de abstração específicas como forma de recurso auxiliar que leva a uma melhoria adicional de precisão na classificação. Nessa abordagem, os recursos extraídos dos níveis inferiores são combinados para ajudar a melhorar a capacidade discriminativa dos recursos intermediários e superar eventuais problemas com o desaparecimento ou explosão do gradiente de rede. Ao final, obteve uma acurácia de 96,17%.

Ramaneswaran et al. [5] Explorou diversos *Classification Heads*, em vez do softmax comumente utilizado no problema em questão. Foram testadas várias técnicas de data augmentation e aplicado um híbrido entre a arquitetura InceptionV3 e o XGBoost como *classification head*. Obteve-se 98,6 de F1-Score.

Xie et al. [11] Desenvolveu uma arquitetura *multi-stream* e *multi-feature* com a adoção da Inception ResNetV2 como rede básica e empregou uma arquitetura híbrida que utiliza o conhecimento de três redes básicas: (i) imagens redimensionadas para (299, 299); (ii) imagens recortadas para apenas a região de interesse e redimensionadas para (299, 299); e (iii) imagens convertidas de RGB para HSV e redimensionadas para (299, 299). Por fim, as *features* são extraídas de cada rede básica, são concatenadas e, então, o resultado é usado como input de uma *multi-layer perception* (MLP) para obter o resultado final da classificação. Com esse método, obteve-se 83,35% de F1-Score.

### 3. Dataset

O presente artigo se baseia no conjunto de dados do ISBI C-NMC 2019, dataset preparado no Laboratório de Oncologia, AIIMS, New Delhi durante a competição para *Classification of Normal versus Malignant Cells in B-All White Blood Cancer Microscopic Images*, organizado pela CodaLab e disponibiliza um total de 10661 imagens de glóbulos brancos coletados de 60 pacientes diagnosticados com o câncer do tipo ALL e de 41 pessoas saudáveis.

Portanto, para a construção do modelo deste relatório, o número de classes utilizado será dois: ALL (*Acute Lymphoblastic Leukemia*) e HEM (*Healthy Cell*). Os dados foram obtidos a partir de três *folds* – *fold 0*, *fold 1* e *fold 2* – todos, contendo duas classes, ALL e HEM e com uma distribuição aproximadamente consistente entre os três *folds*, o que nos traz o primeiro desafio, o desbalanceamento dos dados. Tal desbalanceamento foi constatado na análise exploratória dos dados e será melhor abordada na Seção 4 referente à Metodologia aplicada.

Além disso, cada *fold* possui um conjunto de pacientes, denotados por IDs, e cada paciente está associado a um conjunto de imagens de células.

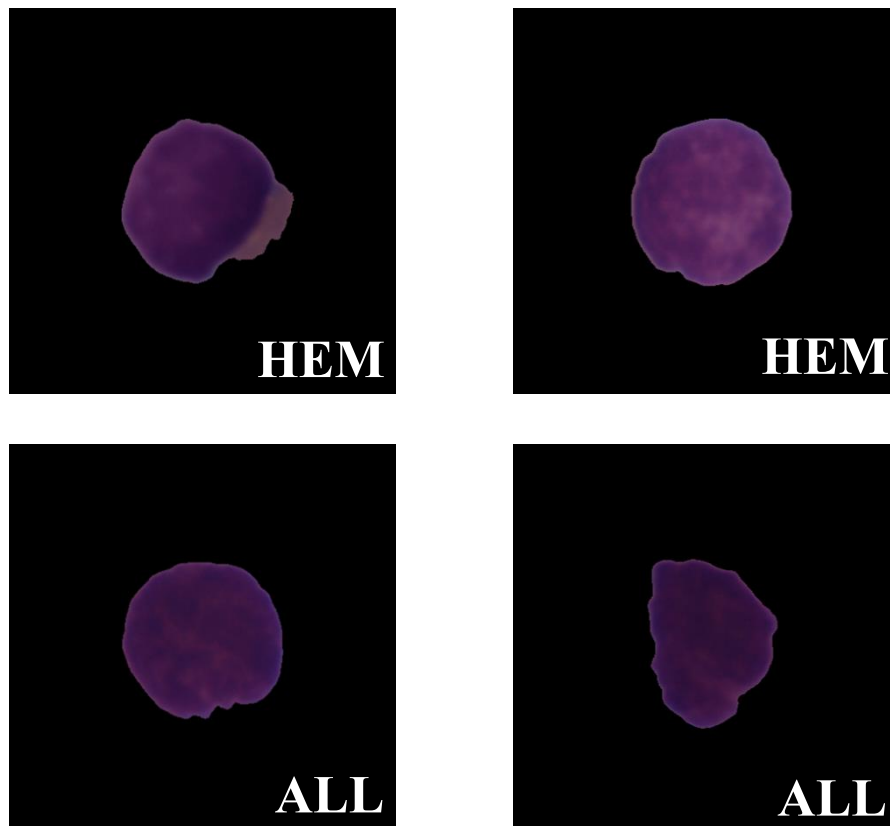


Figura 1: Exemplos de imagens do *dataset* ISBI C-NMC 2019

## 4. Metodologia

### 4.1 Preparação dos Dados

Inicialmente, todos os dados que antes estavam separados em *folds* conforme imagem abaixo, serviram para alimentar um *dataframe* de duas colunas nomeadas como *full\_path* – indicando o caminho até a imagem – e *label* – referente a classe pela qual a respectiva imagem pertence. Com o objetivo de executar todos os treinamentos, foi realizado um mapeamento em que as classes antes denominadas ALL passaram a se chamar 1 e as classes HEM passaram a se chamar 0. Esses valores binários foram colocados na coluna *label*.

<i>full_path</i>	<i>label</i>
/kaggle/input/cnmc-leukemia/fold_0/fold_0/all/...	1
/kaggle/input/cnmc-leukemia/fold_0/fold_0/all/...	1
/kaggle/input/cnmc-leukemia/fold_0/fold_0/all/...	1
/kaggle/input/cnmc-leukemia/fold_0/fold_0/all/...	1
/kaggle/input/cnmc-leukemia/fold_0/fold_0/all/...	1
...	...
/kaggle/input/cnmc-leukemia/fold_2/fold_2/hem/...	0
/kaggle/input/cnmc-leukemia/fold_2/fold_2/hem/...	0
/kaggle/input/cnmc-leukemia/fold_2/fold_2/hem/...	0
/kaggle/input/cnmc-leukemia/fold_2/fold_2/hem/...	0
/kaggle/input/cnmc-leukemia/fold_2/fold_2/hem/...	0

Figura 2 – *Dataframe* inicial que relaciona o caminho de cada imagem e sua respectiva *label* / classe

### 4.2 Análise Exploratória dos Dados (EDA)

Durante a análise exploratória dos dados, o único ponto de atenção observado, também já constatado pelos artigos disponíveis que utilizam esse *dataset* como base, foi o desbalanceamento entre as classes entre ALL e HEM.

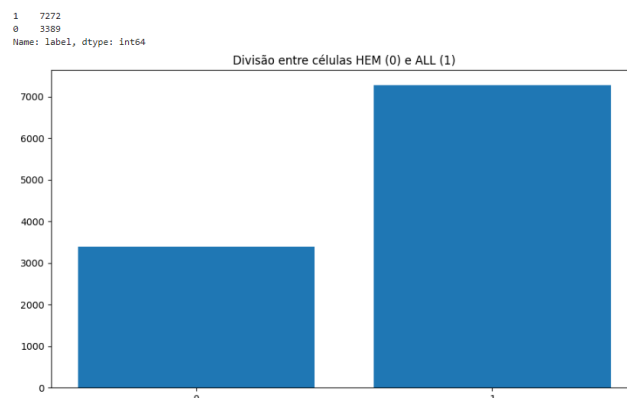


Figura 3: Distribuição entre as classes ALL e HEM

### 4.3 Separação dos conjuntos de dados

O *dataset* com um total de 10661 imagens foi dividido em 3 conjuntos de maneira estratificada com 75% das imagens para treino e os demais 25% divididos igualmente entre os conjuntos de validação e de teste. A divisão deve ser feita de maneira aleatória, buscando manter a distribuição de classe em cada subconjunto. Pode-se obter essa divisão aplicando a amostragem estratificada, tendo a classe como variável controle do processo de estratificação, garantido, assim, que a classe minoritária tenha oportunidade de estar nos conjuntos de treino, de validação e de teste. A adoção da separação estratificada foi o primeiro método aplicado para lidar com o desbalanceamento das classes.

Ainda, é importante destacar que, por motivos de otimização e redução do tempo de treinamento, todas as execuções intermediárias para a escolha do melhor tipo de pré-processamento, aplicação do data augmentation, aplicação de diferentes pesos entre as classes e escolha da arquitetura foi utilizado um *dataframe* reduzido de 500 imagens.

### 4.4 Criação dos *datasets* e pré-processamento

A elaboração dos *datasets* e etapas de pré-processamento, bem como todo o restante do código foi realizada utilizando o Keras / TensorFlow. Para a criação dos *datasets* foi adotada a seguinte abordagem:

As imagens originais possuem o tamanho de (450, 450), composta por um glóbulo branco centralizado em um background preto, obtidas por meio de segmentação. Como para o problema de classificação proposto o fundo preto não possui importância, foi realizado um *CenterCrop* para o valor de (448, 488). Ademais, após isso foi aplicado um *resize* para o valor (299, 299) para questões de otimização de memória e redução no tempo de treinamento dos modelos intermediários.

Para a criação dos modelos foram utilizadas algumas das arquiteturas do Keras Applications com os pesos da ImageNet e com as camadas descongeladas. Por fim, foi adicionada uma camada de ativação Softmax para as duas classes.

Os seguintes valores foram utilizados para o treinamento:

Hiperparâmetros	
Epochs	10
Learning rate	0.0001
Img size	(299, 299)
Batch size	16
Hyperparameter Tuning com Callback	ReduceLROnPlateau(factor=0.2, patience=3, verbose=1)

Tabela 1: hiperparâmetros utilizados nos testes intermediários do notebook.

## 4.5 Métricas

A métrica de avaliação convencionalmente utilizada e adotada pelo desafio da CodaLab é o F1 Score. Tal métrica é definida como a média harmônica entre a Precision e o Recall. Precision é a razão entre os verdadeiros positivos e a soma dos verdadeiros e falsos positivos. Enquanto o Recall é a razão entre os verdadeiros positivos e a soma dos verdadeiros positivos com os falsos negativos.

Ainda, conforme adotado pelo desafio, foi utilizado o argumento average com o valor de 'weighted' ao aplicar o F1-Score.

Como foi observado que existem artigos na literatura que utilizam o AUC, optou-se por calculá-lo, mas será apenas utilizado como métrica auxiliar.

$$Precision = \frac{VP}{VP + FP}$$
$$Recall = \frac{VP}{VP + FN}$$
$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

## 4.6 Escolha da Arquitetura

Para a escolha da arquitetura foi aplicado o modelo explicitado na Seção 4.4 em algumas das arquiteturas do Keras Applications utilizando uma GPU T4 x2. Foi obtido os seguintes resultados:

	Train F1 Score	Val F1 Score	Train AUC	Val AUC	Training Time
Arquitetura					
<b>AlexNet</b>	100.00%	80.07%	100.00%	85.13%	277 s
<b>InceptionV3</b>	100.00%	86.03%	100.00%	90.44%	224 s
<b>InceptionResNetV2</b>	100.00%	84.44%	100.00%	89.83%	479 s
<b>VGG16</b>	87.85%	76.79%	94.98%	85.25%	319 s
<b>ResNet18</b>	100.00%	79.17%	100.00%	83.23%	96 s

Figura 4: Comparação entre as performances das arquiteturas do Keras Applications.

Ao final, foi escolhido a arquitetura com a melhor performance que seria utilizada para o restante do notebook, nesse caso, a InceptionV3.



## 4.7 Class weight

Além disso, foi realizado um experimento com a utilização do argumento `class_weight` no treinamento do melhor modelo obtido na etapa anterior. Tal função possibilita que o modelo “preste mais atenção” aos exemplos de uma classe sub-representada, nesse caso a classe referente às células HEM. A aplicação dessa técnica aprimorou a performance do modelo em 1,5%, conforme a figura abaixo.

	Train F1 Score	Val F1 Score	Train AUC	Val AUC
Arquitetura InceptionV3				
Sem class_weights	100.00%	84.53%	100.00%	89.55%
Com class_weights	100.00%	86.03%	100.00%	90.44%

Figura 5: Comparação entre as performances do modelo com e sem a aplicação do argumento `class_weight` no treinamento do modelo.

## 4.8 Data Augmentation

Após a aplicação de diferentes técnicas de Data Augmentation e tomando como referência trabalhos relacionados, os seguintes métodos foram escolhidos para aplicar no *dataset* em questão:

- *Flips* horizontais e verticais, cada um com 50% de probabilidade de ocorrência;
- Rotações aleatórias com fator 0.3, ou seja, as rotações ocorrem no intervalo  $[-30\% * 2\pi, 30\% * 2\pi]$  (função *RandomRotation* do TensorFlow);
- Translações aleatórias com fator de 20% na altura e largura da imagem (função *RandomTranslation* do TensorFlow);
- Recortes retangulares aleatórios (função *random\_cutout* do TensorFlow Addons);

Com a utilização do Data Augmentation, obteve-se uma performance no F1-Score de validação 1,75% melhor em comparação com a etapa anterior.

Modelo	Train F1-Score	Val F1-Score	Train AUC	Val AUC
Sem Data Augmentation	100%	86,03%	100%	90,44%
Com Data Augmentation	93,05%	87,78%	98,45%	93,29%

Tabela 2: Comparação entre o modelo com e sem o Data Augmentation

#### 4.9 Treinamento com o conjunto de dados de treino completo

Utilizou-se o modelo com Data Augmentation para prosseguir com o treinamento com todo o *dataset* de treino, com um total de 7995 imagens. Infelizmente, por limitações de memória, as imagens foram reduzidas a um tamanho de (200, 200) e *batch size* para 4. Além disso, mais épocas foram adicionadas.

Hiperparâmetros	
Epochs	15
Learning rate	0.0001
Img_size	(200, 200)
Batch_size	4
Hyperparameter Tuning com Callback	ReduceLROnPlateau(factor=0.2, patience=3, verbose=1)

Tabela 3: novos hiperparâmetros adotados para reduzir o uso da memória.

Foram obtidos os resultados abaixo. Com o aumento do conjunto de dados de treino, obteve-se um aumento em 5,01%.

Modelo	Train Weighted F1-Score	Val Weighted F1-Score	Train AUC	Val AUC
InceptionV3 + class_weight + data augmentation	93,04%	92,59%	97,25%	97,27%

Tabela 4: Desempenho do modelo principal com o conjunto de dados de treino completo.

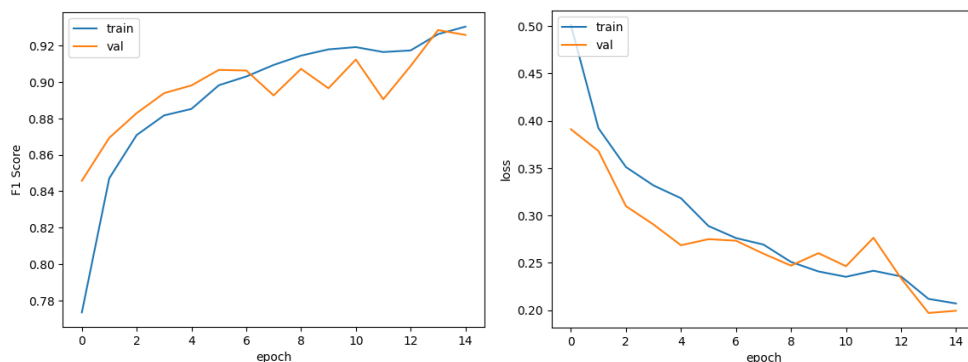


Figura 6: gráficos F1-Score x epochs e loss x epochs para os conjuntos de treino e validação do modelo principal.

#### 4.10 Aplicação da arquitetura em imagem HSV

Complementando, também se aplicou a mesma arquitetura para imagens convertidas de RGB para HSV para observações mais claras das características das células, o que permite ter uma nova visualização em que é possível constatar uma textura predominante no canal S. Porém, apesar de ter sido treinado sob as mesmas condições do modelo principal, apresentou uma performance inferior ao modelo principal.

Modelo	Train weighted F1-Score	Val weighted F1-Score	Train AUC	Val AUC
InceptionV3 + class_weight + data augmentation + HSV	86,21%	86,03%	92,32%	91,83%

Tabela 5: Desempenho do modelo com as imagens convertidas para HSV com o conjunto de dados de treino completo.

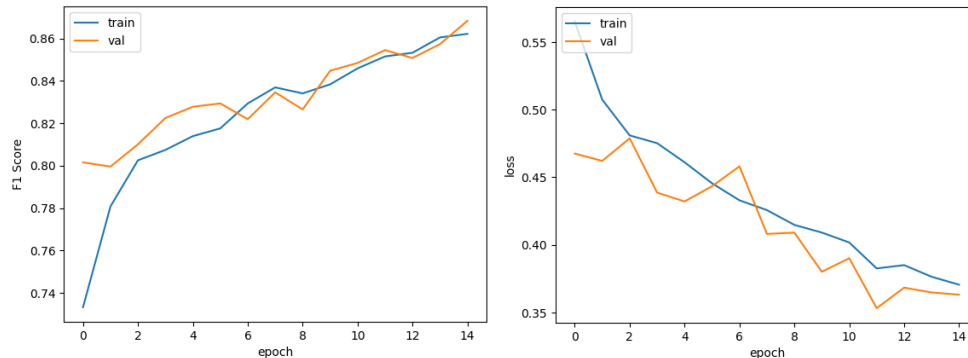


Figura 7: gráficos F1-Score x epochs e loss x epochs para os conjuntos de treino e validação do modelo em HSV.

#### 4.11 Ensemble Averaging

Prosseguindo com os testes para melhoria de performance do modelo, foi realizado um processo de Ensemble Averaging entre os dois modelos propostos (o melhor modelo obtido até o momento com a arquitetura InceptionV3 + Data Augmentation + class\_weight, intitulado “Main Model” com o respectivo modelo com a conversão RGB para HSV, intitulado “HSV Model”). Os resultados podem ser observados na figura abaixo e, portanto, podemos concluir que a performance em relação ao modelo principal foi aprimorada em 2,56%. Dessa forma, o modelo final passou a ser o obtido a partir do Ensemble.

	Test F1 Score	Test AUC
Modelo		
Main Model	0.913151	0.910476
HSV Model	0.860339	0.848142
Ensemble Model	0.938821	0.911180

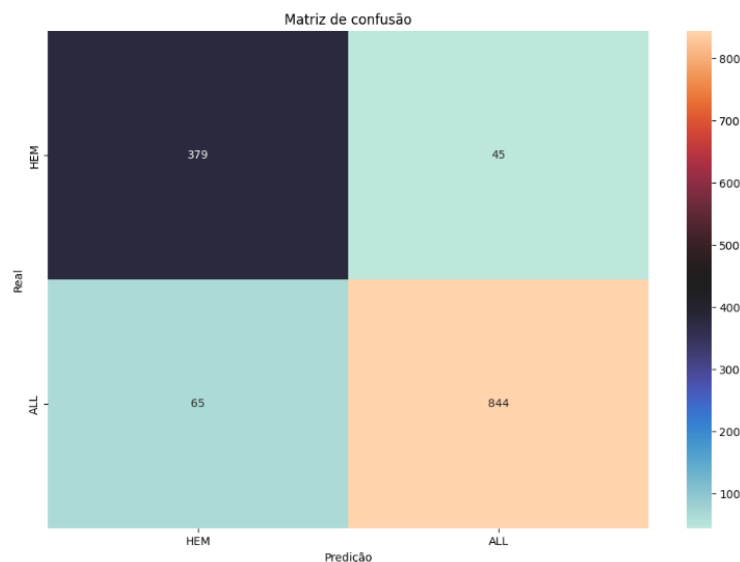
Figura 8: Comparação de desempenho entre os modelos propostos.

## 4.12 Retreinamento

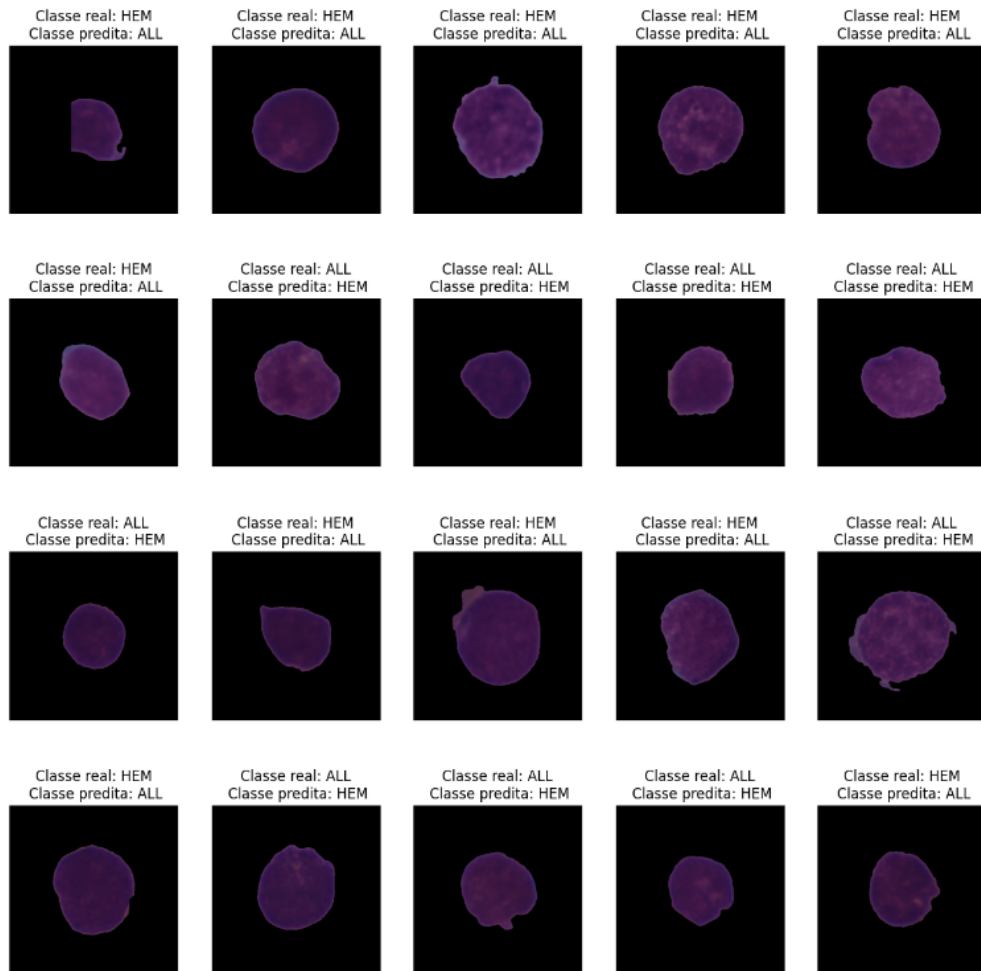
O código para essa etapa foi desenvolvido, porém optou-se por não o executar por limitações de memória. Nesse sentido o modelo utilizado a partir dessa etapa foi obtido a partir do treinamento com o dataset de treino completo com o Ensemble com o modelo com as imagens em HSV explicado anteriormente.

# 5. Avaliação e Resultados

## 5.1 Matriz de Confusão



## 5.2 Imagens Erradas



## 5.3 Comparação com a performance de outros modelos da literatura

Modelo	Weighted F1-Score
Nayak [7]	80%
Marzahl et al. [8]	82,84%
Kulhalli et al. [3]	93,2%
Ding et al. [6]	85,52%
Pan et al. [10]	91,04%
Kramer [3]	88,91%
Ramaneswaran [5]	98,6%
<b>Método proposto</b>	<b>93,88%</b>

## 6. Conclusão

Nesse manuscrito, foi proposto um simples e efetivo modelo de Ensemble Neural Network para a classificação binária de glóbulos brancos capaz de diferenciar entre células cancerígenas (ALL) e células saudáveis (HEM) do dataset ISBI C-NMC 2019. Nesse agrupamento foi utilizada a mesma arquitetura InceptionV3 e construção do modelo, sendo que a única diferença aplicada foi a alteração do espaço de cores RGB para HSV.

Além disso, foram explorados diferentes métodos para lidar com o desbalanceamento das classes e variadas técnicas de data augmentation visando aprimorar o desempenho do modelo final, que apresentou um weighted F1-Score de 93,88%, um dos mais altos de acordo com a literatura e que ocuparia o 4º lugar no desafio da CodaLab.

## 7. Referências

- [1] S. Kassani, K. Schneider, P Kassani, M. Wesolowski, R. Deters, “A Hybrid Deep Learning Architecture for Leukemic B-lymphoblast Classification”. Disponível em: <https://arxiv.org/pdf/1909.11866>
- [2] R. Gupta, S. Gehlot, A. Gupta, “C-NMC: B-lineage acute lymphoblastic leukaemia: A blood câncer dataset”. Disponível em: ISBI 2019 C-NMC Challenge: Classification in Cancer Cell Imaging.
- [3] J. Prellberg, O. Kramer, “Acute Lymphoblastic Leukemia Classification from Microscopic Images using Convolutional Neural Networks”. Disponível em: <https://arxiv.org/abs/1906.09020>
- [4] Y. Liu, F. Long, “Acute lymphoblastic leukemia cells image analysis with deep bagging ensemble learning”. Disponível em: <https://www.biorxiv.org/content/10.1101/580852.full.pdf>
- [5] S. Ramaneswaran, K. Srinivasan, P.M Vincent, C. Chang, “Hybrid Inception v3 XGBoost Model for Acute Lymphoblastic Leukemia Classification”. Disponível em: <https://www.hindawi.com/journals/cmmm/2021/2577375/>
- [6] Y. Ding, Y. Cui, “Deep Learning for Classifying os White Blood Cancer”. Disponível em: ISBI 2019 C-NMC Challenge: Classification in Cancer Cell Imaging.
- [7] A. Honnalgere, G. Nayak, “Classification of Normal Versus Malignant Cells in B-ALL White Blood Cancer Microscopic Images”. Disponível em: ISBI 2019 C-NMC Challenge: Classification in Cancer Cell Imaging.
- [8] C. Marzahl, M. Aubreville, J. Maier, “Classification of Leukemic B-Lymphoblast Cells from Blood Smear Microscopic Images with na Attention-Based Deep Learning Method and Advanced Augmentation Techniques”. Disponível em: ISBI 2019 C-NMC Challenge: Classification in Cancer Cell Imaging.

- [9] R. Kulhalli, C. Savadikar, B. Garware, “Toward Automated Classification of B-Acute Lymphoblastic Leukemia”. Disponível em: ISBI 2019 C-NMC Challenge: Classification in Cancer Cell Imaging.
- [10] Y. Pan, M. Liu, X. Shen, D. Shen, “Neighborhood-Correction Algorithm for Classification of Normal and Malignant Cells”. Disponível em: [https://drive.google.com/file/d/1\\_KbpWDWNHbEZSoJuQ026huCkEgtxQxm0/view](https://drive.google.com/file/d/1_KbpWDWNHbEZSoJuQ026huCkEgtxQxm0/view)
- [11] X. Xie, Y. Li, M. Zhang, Y. Wu, L. Shen, “Multi-streams and Multi-features for Cell Classification”. Disponível em: ISBI 2019 C-NMC Challenge: Classification in Cancer Cell Imaging.
- [12] S. Mourya, S. Kant, P. Kumar, A. Gupta, R. Gupta, “Leukonet: DCT-Based CNN Architecture for the Classification os Normal Versus Leukemic Blasts in B-ALL Cancer”. Disponível em: <https://arxiv.org/pdf/1810.07961>
- [13] V. Singhal, P. Singh, “Local binary pattern for automatic detection of acute lymphoblastic leucemia”. Disponível em: [https://ieeexplore.ieee.org/iel7/6802611/6811235/06811261.pdf?casa\\_token=E0uPP-o9bosAAAAA:-SnKoN3TUMqCjBBFxIr6TZbO6c5CjG72YhnXHZ6dwzw20LF3M5G\\_kkEnl6sAY9u5i7CrkaHTzOE](https://ieeexplore.ieee.org/iel7/6802611/6811235/06811261.pdf?casa_token=E0uPP-o9bosAAAAA:-SnKoN3TUMqCjBBFxIr6TZbO6c5CjG72YhnXHZ6dwzw20LF3M5G_kkEnl6sAY9u5i7CrkaHTzOE)