

Trabalho Prático - Processamento e Análise de Imagens

Felipe Costa Amaral, Henrique Mendonça Castelar Campos, Larissa Kaweski Siqueira

1

Abstract. *This article proposes and implements an interface capable of loading images and classifying them as containing or not cancer cells. Two artificial intelligences will be implemented capable of classifying between five types of cancers and the healthy cell. Besides this classifier, the interface will be capable of processing and analyzing the image.*

Resumo. *Este artigo propõe e implementa uma interface capaz de ler imagens e classificá-las como contendo ou não células cancerígenas. Serão implementadas 2 inteligências artificiais capazes de classificar entre 5 tipos de câncer e a célula saudável. Além deste classificador, a interface será capaz de fazer certos processamentos e análises na imagem.*

1. Introdução

O Câncer Cervical, também conhecido como Câncer de Colo de Útero, é uma doença comum entre as mulheres ao redor do mundo. Um dos mecanismos utilizados para a prevenção desse tipo de doença é o uso de imagens médicas para diagnóstico. Um dos métodos de imagem médica utilizados é a citologia, por meio do exame Papanicolau, que detecta anomalias celulares. Ele consiste na extração de amostras de células do paciente, e a aplicação de corantes nelas. Dependendo da forma como as células reagem aos corantes, isso indica a presença ou não de câncer.

Este artigo aborda o desenvolvimento de técnicas de aprendizado de máquina para a detecção de células cancerosas no exame Papanicolau, por meio de dois classificadores: o *Support Vector Machines* (SVM) e o *Residual Neural Network* de 50 camadas (ResNet50).

2. Implementação

A implementação da solução de software foi feita na linguagem de programação Python. Para isso foram criados três pacotes (packages): modelo, visão e controlador, seguindo o padrão Model View Controller.

No pacote modelo foram criadas as classes:

- *ImagemRGB*, que implementa os métodos e atributos utilizados para a manipulação de uma imagem RGB.
- *ImagemHSV*, que implementa os métodos e atributos utilizados para a manipulação de uma imagem HSV.
- *ImagemTonsCinza*, que implementa os métodos e atributos utilizados para a manipulação de uma imagem em tons de cinza.
- *Imagem*, classe abstrata que é herdada pelas demais classes de Imagem, que implementa os métodos e atributos comuns utilizados na manipulação de imagens.

No pacote visão (visao) foram criadas as classes:

- *FrameImagem*, que é o frame que é exibido quando uma imagem é mostrada.
- *FramePrincipal*, que é o primeiro frame a ser mostrado ao abrir o programa.
- *FrameClassificador*, que mostra as opções para classificar a imagem.
- *FrameDescritoresHaralick*, que mostra os Descritores de Haralick.
- *FrameMomentosInvariantesHu*, que mostra os Momentos Invariantes de Hu.
- *JanelaPrincipal*, que é responsável por mostrar a janela principal do programa.
- *JanelaZoom*, que mostra opções avançadas de zoom na imagem.

No pacote controlador foi criada a classe Controlador, que é responsável por relacionar a interface gráfica com os dados do modelo.

A interface gráfica foi construída com o uso da biblioteca tkinter, nativa do Python. Para isso foram criadas duas janelas e cinco frames.

As janelas desenvolvidas foram:

- A janela principal, que é a primeira janela aberta pelo programa.
- A janela de zoom, que permite a especificação da nova área da imagem a ser exibida.

E os frames desenvolvidos foram:

- Frame principal, que é o primeiro frame a ser exibido pelo programa. Nele contém os nomes dos desenvolvedores.
- Frame imagem, que é o frame utilizado na exibição de imagens.
- Frame classificador, que é o frame utilizado para classificar uma imagem.
- Frame descritores de Haralick, que mostra os Descritores de Haralick.
- Frame momentos invariantes de Hu, que mostra os Momentos invariantes de Hu.

Na janela principal é mostrado a barra de menu com os menus: 'Arquivo', 'Visualizar', 'Converter', 'Histogramas', 'Caracterizar' e 'Classificar'.

O menu 'Arquivo' possui o botão 'Abrir imagem', que permite abrir um arquivo de imagem.

O menu 'Visualizar' possui os botões:

- 'Aumentar zoom (Simples)', que permite aumentar o zoom da imagem no seu centro.
- 'Aumentar zoom (Avançado)', que permite aumentar o zoom por meio da inserção da nova fronteira.
- 'Remover zoom', que permite retirar o zoom.

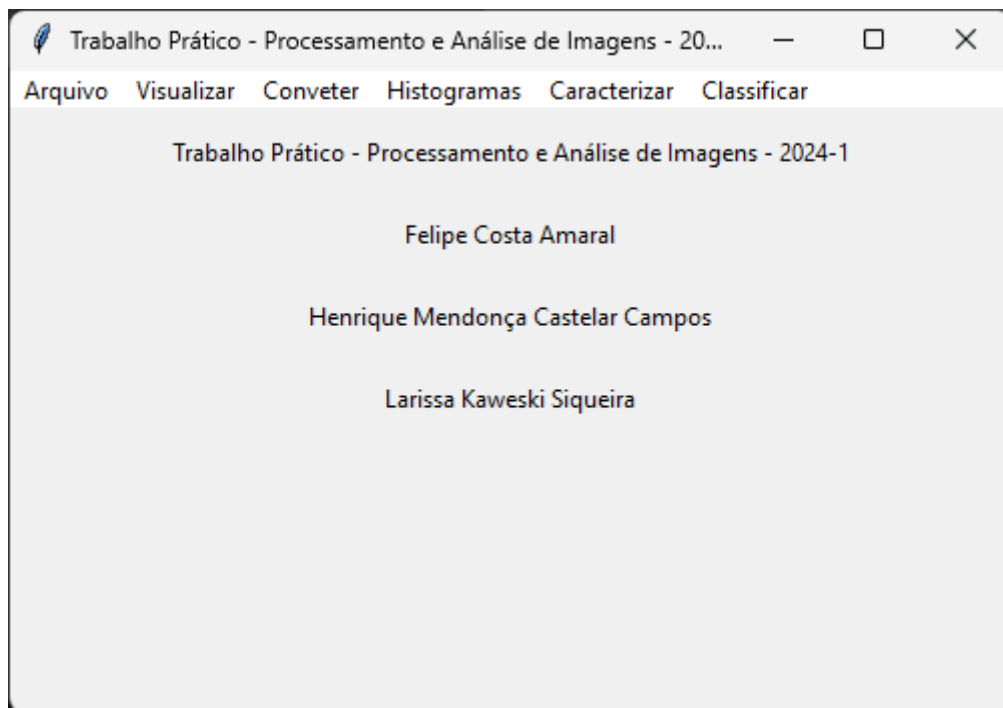


Figura 1. Janela principal, quando o programa é iniciado.

O menu 'Converter' possui os botões:

- 'RGB', que converte a imagem para RGB.
- 'Tons de cinza', que converte a imagem para tons de cinza.
- 'HSV', que converte a imagem para HSV.

O menu 'Histogramas' possui os botões:

- 'Tons de cinza', que gera e mostra o histograma da imagem em tons de cinza.
- 'HSV_2D', que gera e mostra o histograma HSV 2D, relacionando os valores de Hue com Value.
- Submenu 'HSV', que gera e mostra os histogramas para 'Hue', 'Saturation' e 'Value'.

O menu 'Caracterizar' possui os botões:

- 'Descritores de Haralick', que mostra os Descritores de Haralick.
- 'Momentos invariantes de Hu', que mostra os Momentos invariantes de Hu.

E o botão 'Classificar', que alterna para o frame de classificação, que mostra as opções para classificar a imagem.

As bibliotecas utilizadas junto ao programa foram:

- Numpy, que é utilizado para a manipulação de vetores.

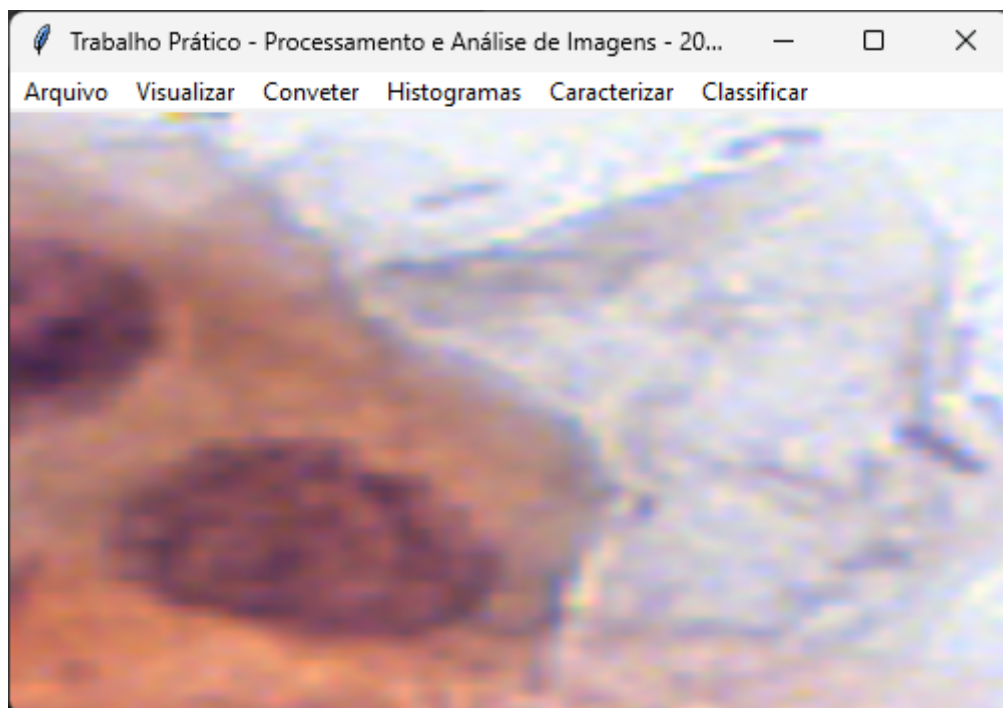


Figura 2. Janela principal, quando uma imagem é aberta.



Figura 3. Janela principal, exibindo uma imagem em tons de cinza.

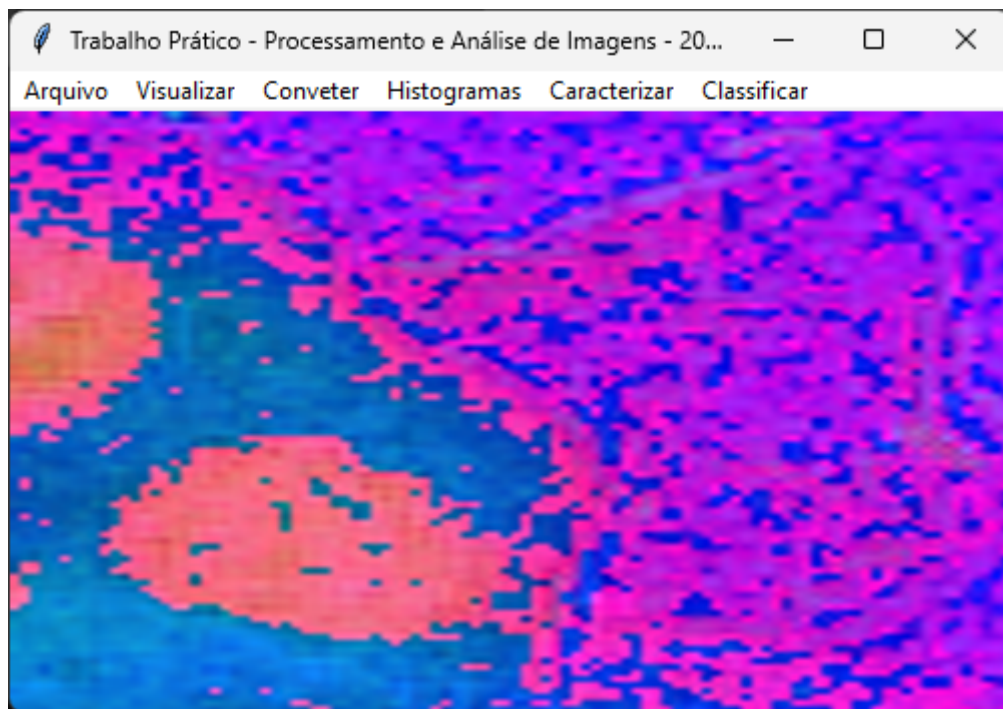


Figura 4. Janela principal, exibindo uma imagem HSV.

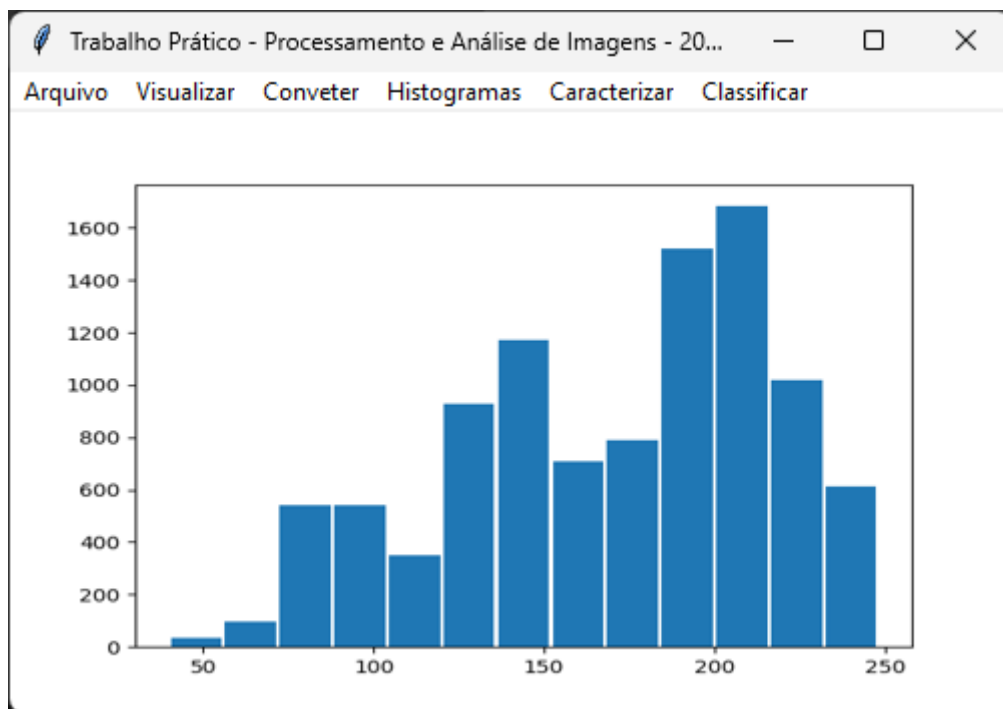


Figura 5. Janela principal, exibindo o histograma de uma imagem em tons de cinza.

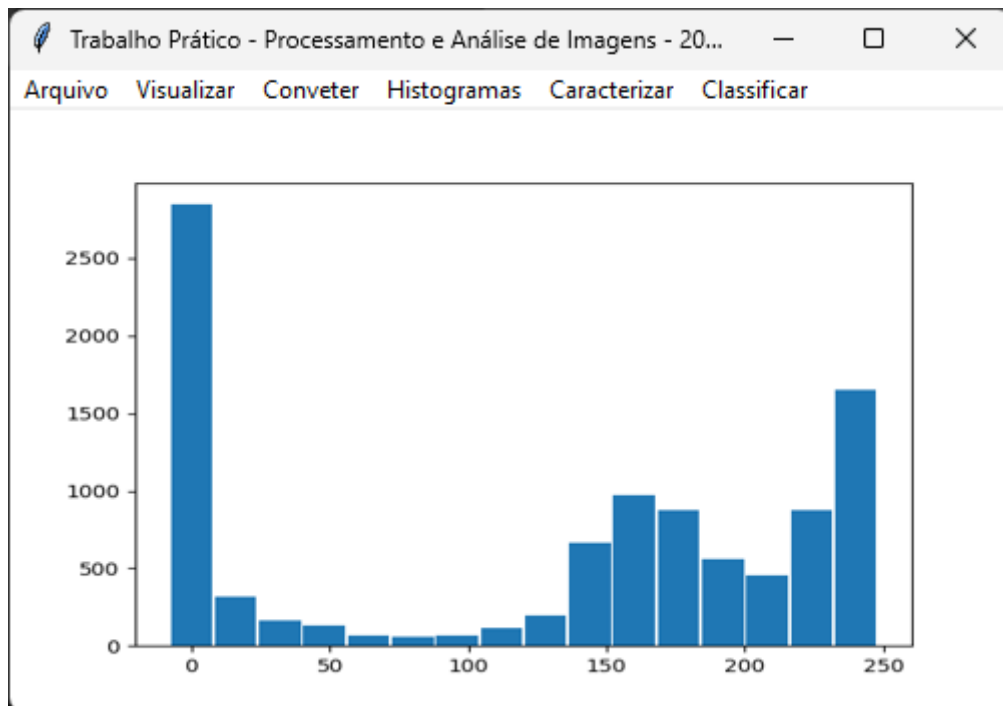


Figura 6. Janela principal, exibindo o histograma Hue da imagem HSV.

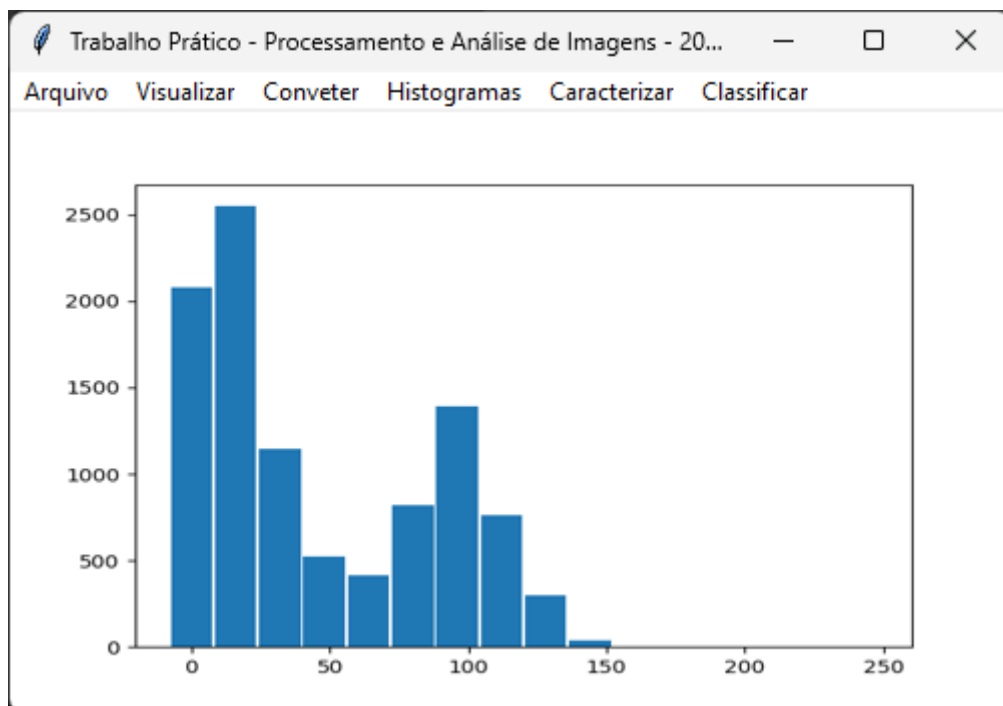


Figura 7. Janela principal, exibindo o histograma Saturation da imagem HSV.

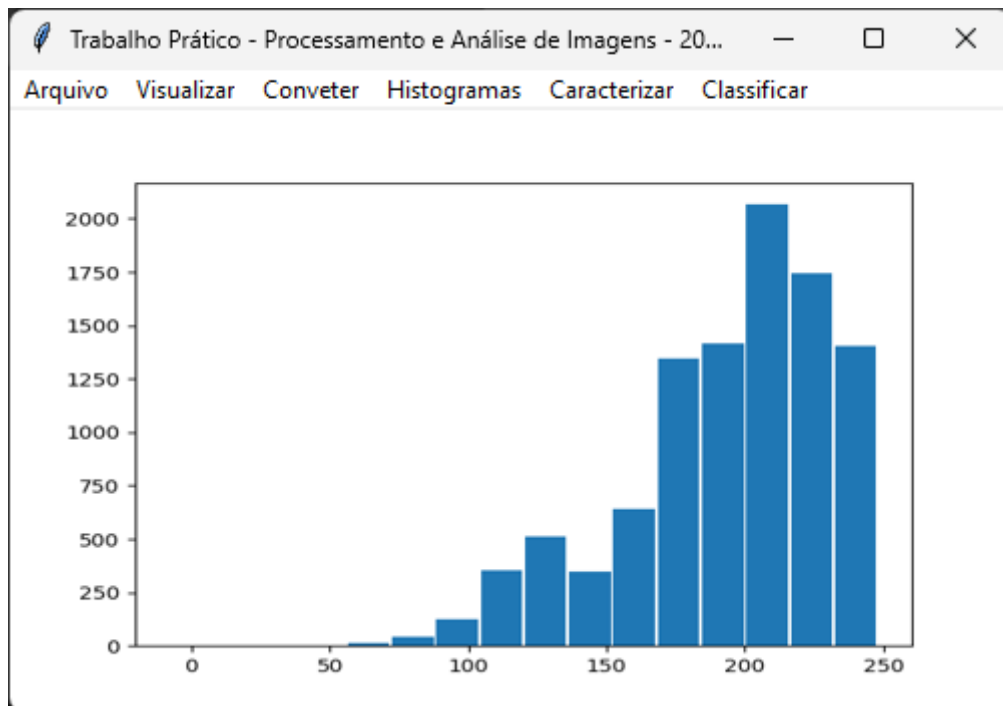


Figura 8. Janela principal, exibindo o histograma Value da imagem HSV.

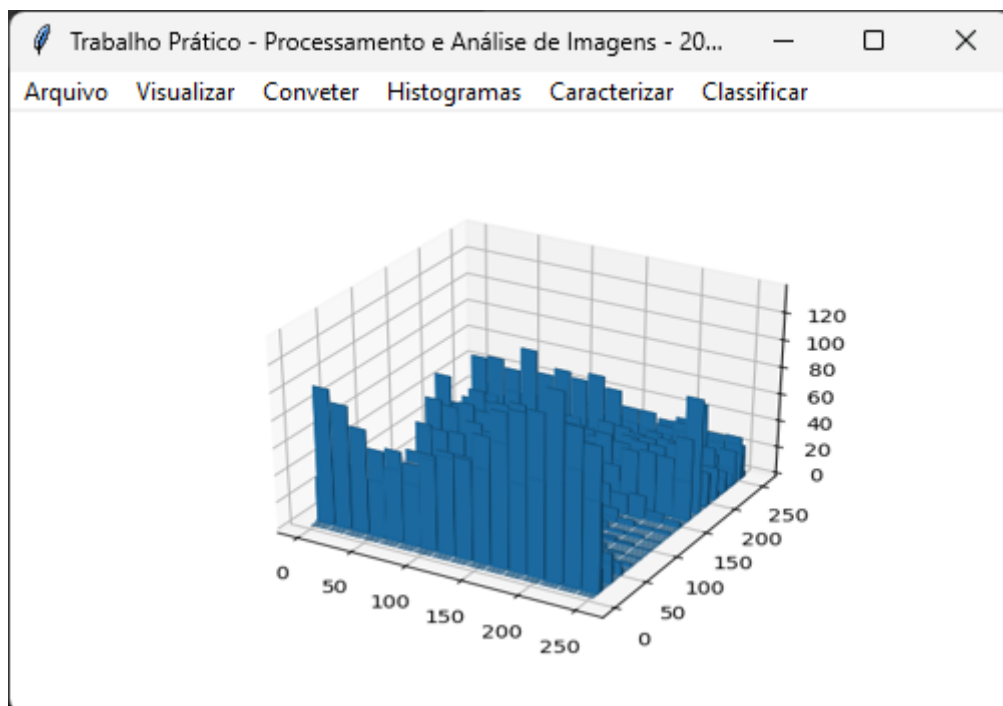


Figura 9. Janela principal, exibindo o histograma 2D da imagem HSV.

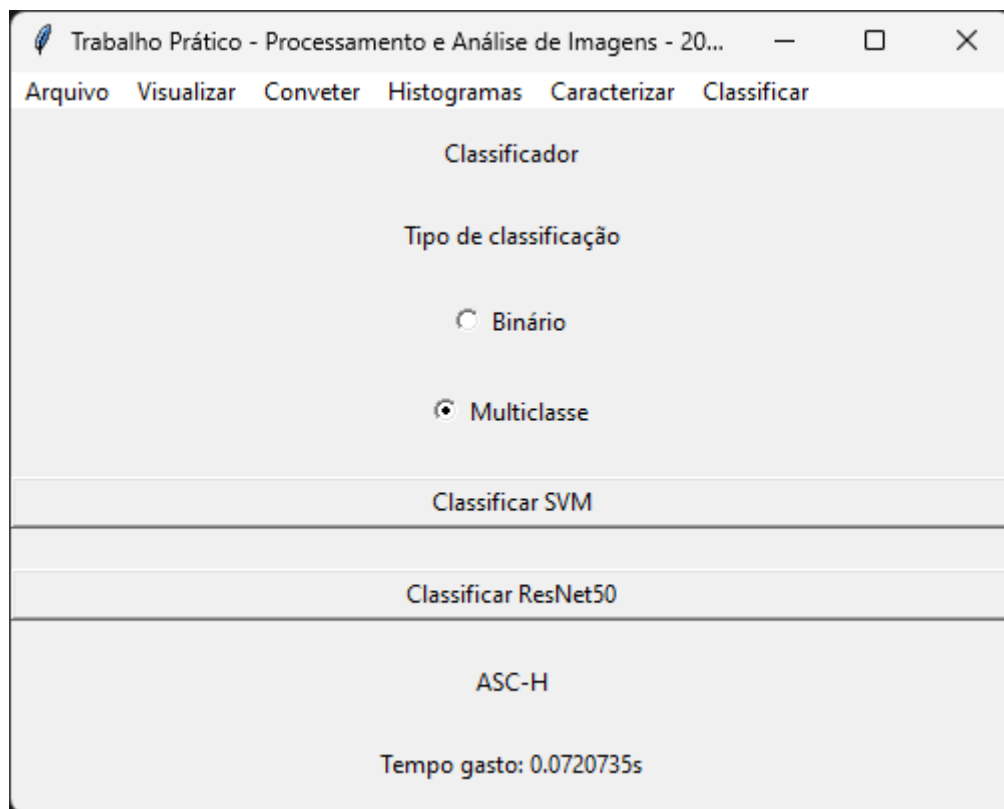


Figura 10. Janela principal, exibindo as opções para classificar a imagem.



Figura 11. Janela principal, mostrando os Momentos Invariantes de Hu.

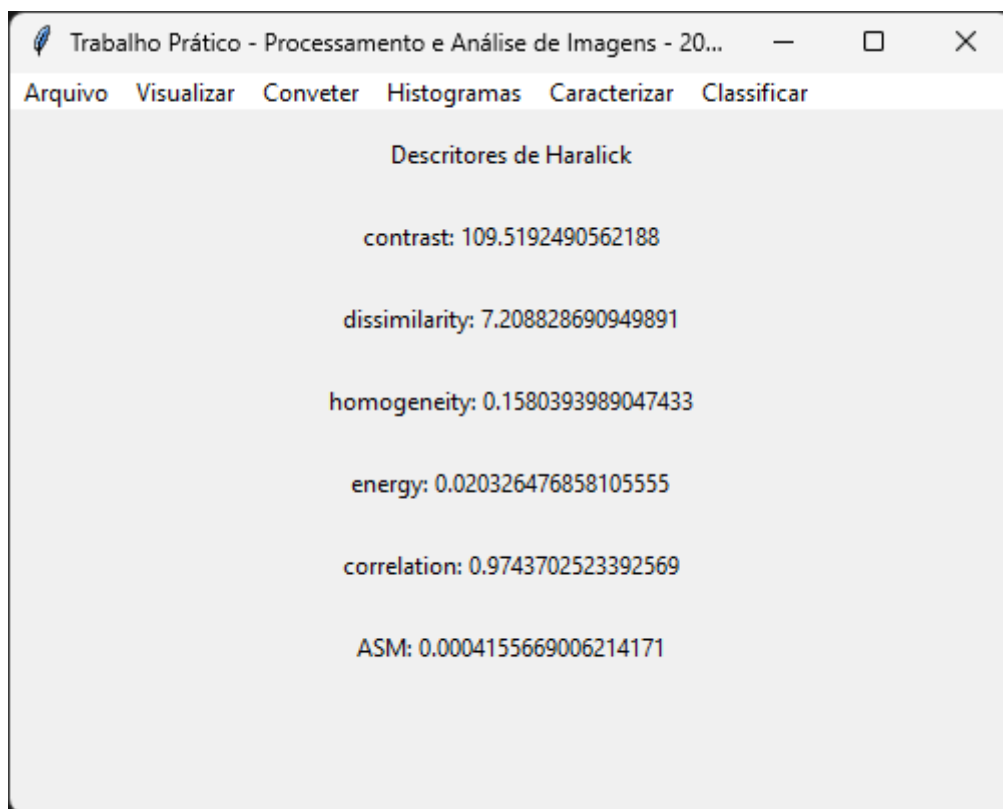


Figura 12. Janela principal, mostrando os Descritores de Haralick.

- Pillow, que é utilizado para o carregamento e manipulação de imagens.
- TensorFlow, que é utilizado para a criação dos modelos de aprendizado de máquina (SVM e ResNet50).
- Matplotlib, que é utilizado para a geração de gráficos (nos histogramas).

3. Resultados

3.1. Qualidade dos resultados

O ResNet50 mostrou um resultado melhor que o SVM, porém as métricas relevantes não foram alcançadas com a precisão desejada. Esse desempenho era esperado

Métrica	Valor
Taxa de Verdadeiro Positivo	0,46149403734347655
Taxa de Verdadeiro Negativo	0,865422535324905
Taxa de Falso Positivo	0,13457746467509502
Taxa de Falso Negativo	0,5385059626565234
Precisão	0,2630684113588874
Sensibilidade	0,46149403734347655
Acurácia	0,7842680523203728
F-Measure	0,2559065269629602

Tabela 1. Métricas do SVM.

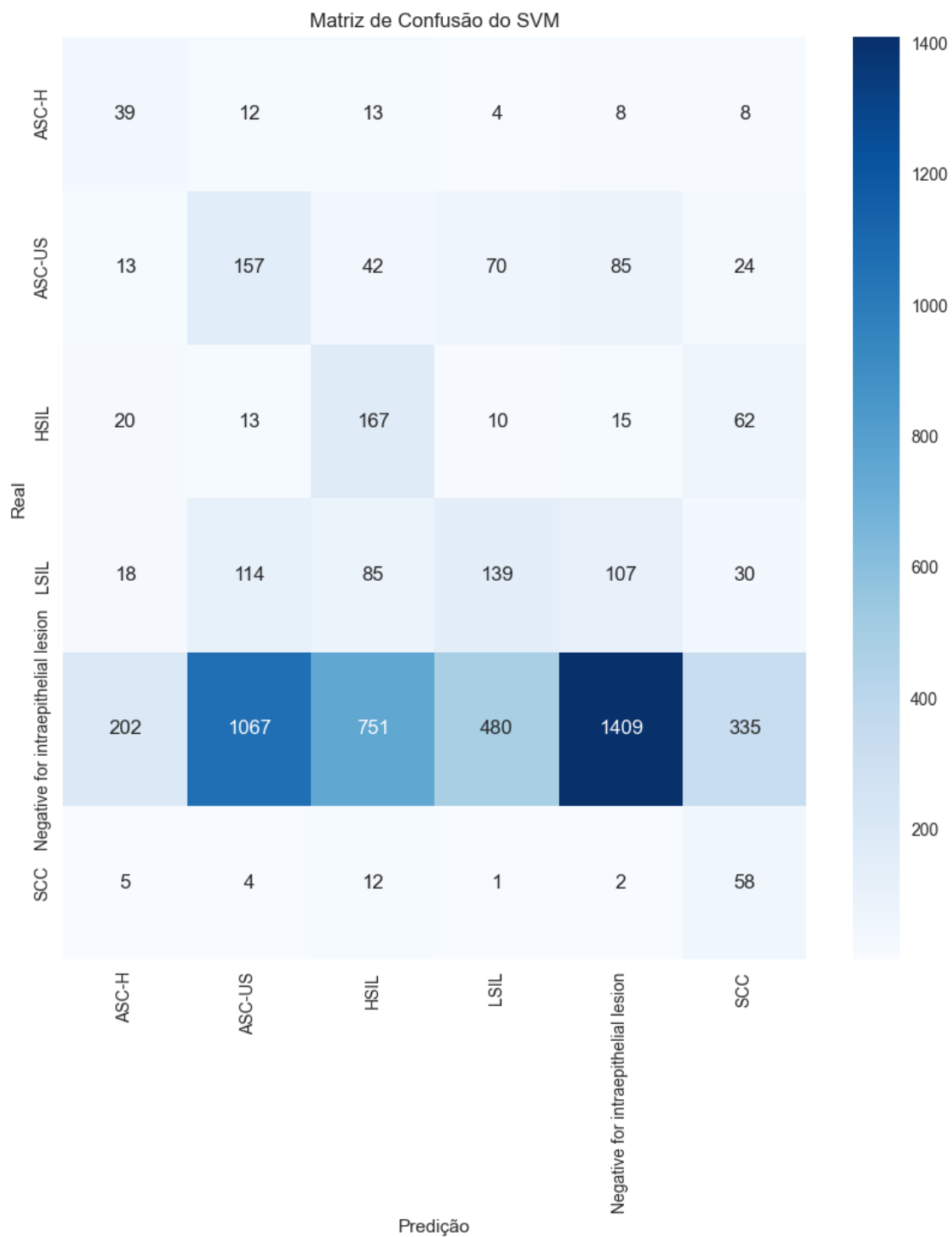


Figura 13. Matriz de confusão do SVM

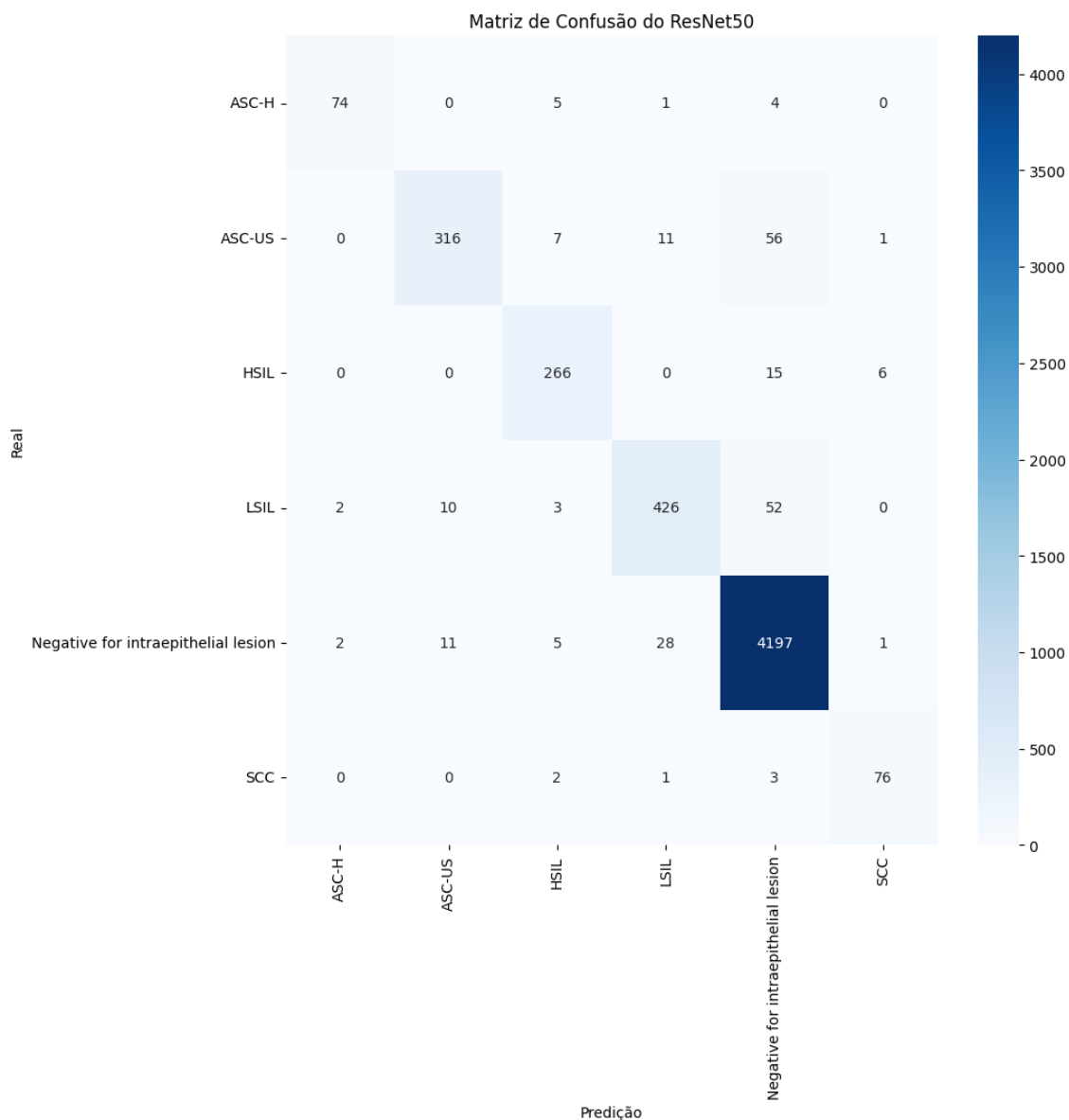


Figura 14. Matriz de confusão do ResNet50

Métrica	Valor
Taxa de Verdadeiro Positivo	0,8993029942974969
Taxa de Verdadeiro Negativo	0,9807208072160782
Taxa de Falso Positivo	0,019279192783921716
Taxa de Falso Negativo	0,10069700570250299
Precisão	0,9328230136088257
Sensibilidade	0,8993029942974969
Acurácia	0,9865018216568119
F-Measure	0,9149068540667632

Tabela 2. Métricas do ResNet50.

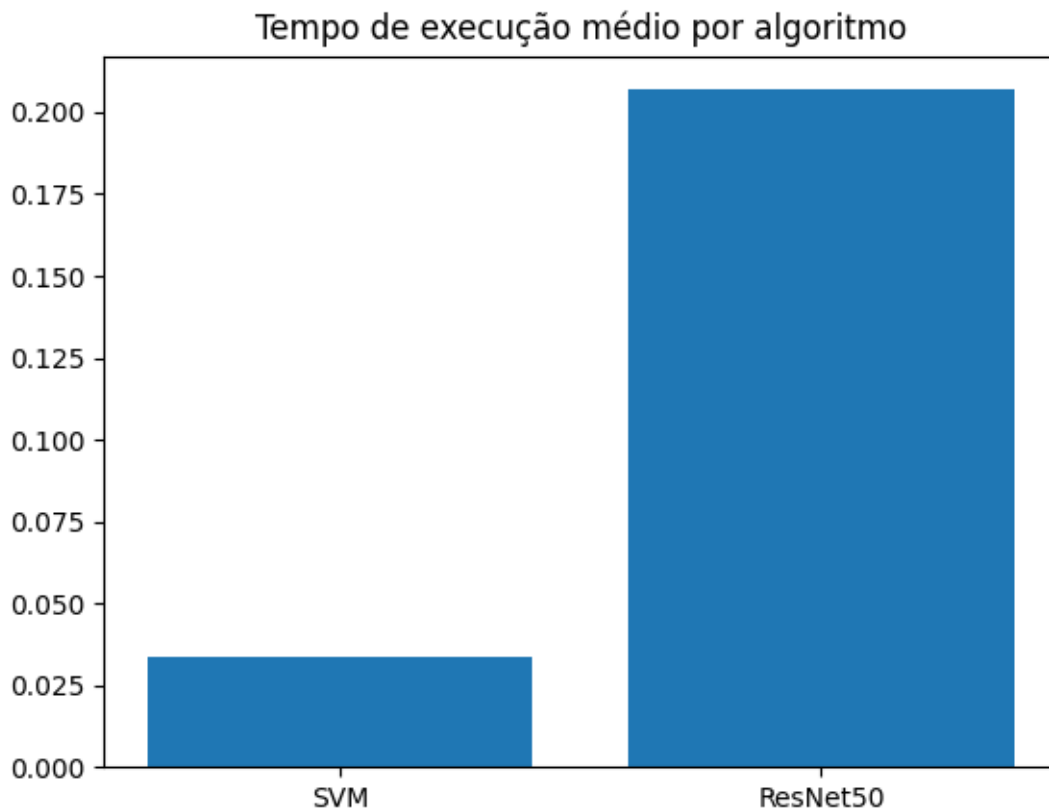


Figura 15. Gráfico comparando o tempo gasto médio de classificação no SVM e no ResNet50.

pois as máquinas utilizadas para treinar o modelo não tinham as configurações ideais para esse tipo de computação.

Através da matriz de confusão do modelo SVM, foi possível concluir que o Supporting Vector Machine não é adequado para a identificação de células cancerígenas.

O SVM foi treinado de diversas formas diferentes, sempre maximizando o score em um valor fixo, o que significa que houve overfit durante o treinamento.

Novamente houve a tentativa de treinamento balanceando a base de treinamento porém a solução não foi eficiente o suficiente para ser satisfatória. Entretanto foi possível evitar o overfit e garantir que o resultado era melhor do que aleatório.

3.2. Tempo

Através do tempo médio de processamento dos modelos, foi possível concluir que o Supporting Vector Machines é mais rápido para classificar uma única imagem que o Residual Neural Network de 50 camadas.

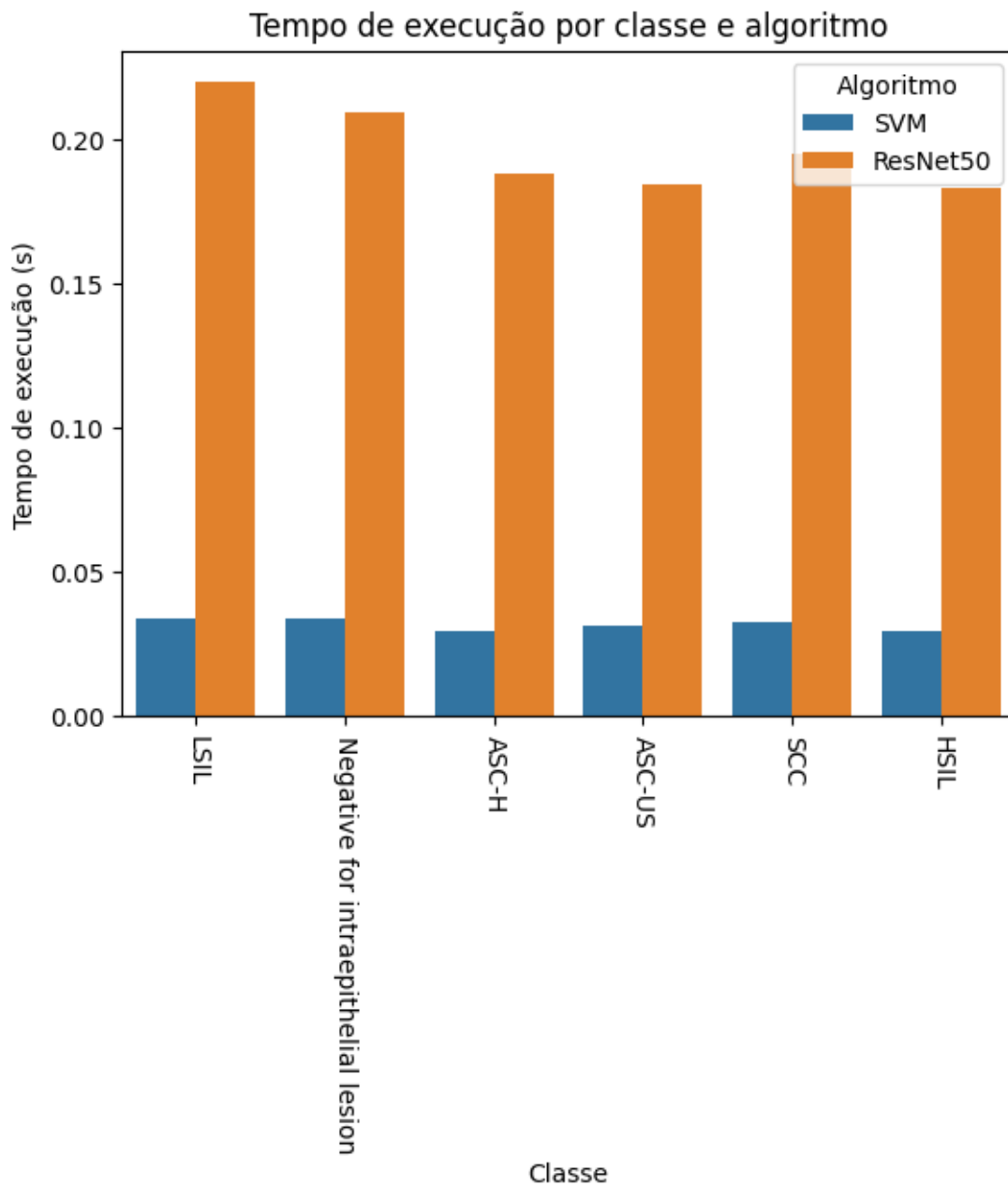


Figura 16. Gráfico comparando o tempo gasto de classificação no SVM e no ResNet50 por classe.

Não houve diferença de tempo significativa na classificação entre elementos de classe diferentes.

4. Conclusão

O presente trabalho apresentou o desenvolvimento de uma interface para classificação de imagens de células cancerígenas utilizando dois modelos de aprendizado de máquina: Support Vector Machines (SVM) e Residual Neural Network de 50 camadas (ResNet50). Através da implementação e análise dos resultados, foi possível verificar que

o modelo ResNet50 obteve desempenho significativamente superior em comparação ao SVM, com melhores métricas de precisão, sensibilidade e acurácia.

Os resultados mostraram que, enquanto o SVM apresentou problemas de overfitting e uma taxa de verdadeiro positivo inferior a 50%, o ResNet50 demonstrou uma alta taxa de verdadeiro positivo e um baixo índice de falsos negativos. Contudo, apesar da superioridade do ResNet50, ambos os modelos não alcançaram a precisão desejada, indicando a necessidade de ajustes e melhorias no processo de treinamento e otimização dos modelos.

Além das análises de desempenho, a implementação destacou a importância da escolha adequada dos modelos de aprendizado de máquina e da infraestrutura computacional necessária para o treinamento eficiente desses modelos. O tempo de processamento também foi um fator relevante, com o SVM se mostrando mais rápido na classificação de imagens individuais, enquanto o ResNet50, apesar de mais lento, apresentou resultados mais robustos.

Portanto, conclui-se que a utilização de redes neurais profundas como o ResNet50 é mais adequada para a tarefa de classificação de células cancerígenas, desde que sejam realizados aprimoramentos no treinamento e na otimização do modelo. Este trabalho contribui para o avanço das técnicas de diagnóstico automatizado, proporcionando uma base para futuras pesquisas e desenvolvimento de soluções mais eficazes na área de processamento e análise de imagens médicas.

Referências

- Machado, A. M. C. (2024a). Processamento e análise de imagens: Deep neural network architectures. Apresentação PDF distribuída em sala de aula.
- Machado, A. M. C. (2024b). Processamento e análise de imagens: Support vector machines. Apresentação PDF distribuída em sala de aula.
- Sarah Feldman, Annekathryn Goodman, J. F. P. (2024). Screening for cervical cancer in resource-rich settings.
- Wikipedia contributors (2023). Papanicolaou stain — Wikipedia, the free encyclopedia. [Online; accessed 16-June-2024].
- Wikipédia (2022). Coloração de papanicolau — wikipédia, a enciclopédia livre. [Online; accessed 8-maio-2022].