

# Pré-Processamento de Imagens

## Base Papanicolau

Arthur Henrique Neves Dias<sup>1</sup>, Bárbara Luciano Araújo<sup>1</sup>, Paulo Antônio Eufrázio Araujo<sup>1</sup>

<sup>1</sup> Instituto de Informática – Pontifícia Universidade Católica de Minas Gerais (PUC-MG)  
Caixa Postal 1.686 – 30.535-610 – Belo Horizonte – MG – Brazil

paearaujo@sga.pucminas.br

barbara.araujo.1299141@sga.pucminas.br

ahndias@sga.pucminas.br

**Abstract.** *This article describe and exemplifies the implementation of a graphical menu able to view images in 'PNG' and 'JPG' format of any resolution and convert them to grayscale. Furthermore, the interface has the capacity to: generate grayscale histograms, characterize the images viewed using Haralick descriptors and classify a sub-image.*

**Resumo.** *Este artigo descreve e exemplifica a implementação de um menu gráfico capaz de visualizar imagens nos formatos 'PNG' e 'JPG' de qualquer resolução e convertê-las em tons de cinzas. Além disso, tal interface possui a capacidade de: gerar histogramas de tons de cinza, caracterizar as imagens visualizadas por meio de descritores de Haralick e classificar uma sub-imagem.*

## 1. Introdução

O câncer cervical é uma das principais causas de mortalidade entre mulheres no mundo. A detecção antecipada e o tratamento correto são fundamentais para diminuir esses valores de mortalidade geradas por essa doença. O exame de Papanicolau é um procedimento bastante utilizado na identificação de alterações celulares no colo do útero que possam indicar a presença de câncer.

Apesar da sua grande eficiência, as análises de tais exames são feitas em processos humanos sujeitos a falhas e com isso diagnósticos inconscientes que podem gerar complicações no tratamento do câncer. Por esse fator, há um grande aumento na demanda por formas mais automatizadas que possam servir de complemento para o trabalho que já é feito atualmente, visando melhorar a precisão e eficiência dos diagnósticos.

Diante desse contexto, o desenvolvimento de uma interface gráfica capaz de ler imagens de exames de Papanicolau e realizar o reconhecimento automático de células cancerosas representa um avanço significativo. Utilizando certas técnicas de processamento de imagens e aprendizado de máquina, é possível criar sistemas que analisam as amostras de maneira rápida e precisa, auxiliando os profissionais de saúde na identificação de alterações celulares malignas.

Este estudo tem como objetivo, o desenvolvimento de uma interface gráfica que utilize algoritmos avançados para a análise de imagens de Papanicolau, juntamente com uma rede neural que facilite a detecção precoce do câncer cervical. A implementação

deste sistema visa ter uma acurácia alta com muita precisão dos diagnósticos e além disso, contribui para a melhoria dos cuidados de saúde para mulheres em todo o mundo.

## 2. Descrição do Problema

O exame de Papanicolaou é uma ferramenta histológica importante para detectar alterações nas células cervicais e desempenha um papel vital na identificação precoce de lesões precursoras do câncer cervical. Embora importante, o processo de análise manual das imagens geradas por este exame está sujeito a erros humanos e pode consumir muito tempo. À medida que a tecnologia avança, surgem oportunidades para automatizar este processo, aumentando assim a precisão e a eficiência do diagnóstico.

Atualmente, a análise de imagens dos exames de Papanicolaou é em grande parte manual, com os citologistas examinando as lâminas em busca de anormalidades celulares. Este método está sujeito a algumas limitações, como:

- **Subjetividade e variabilidade humana:** A interpretação dos resultados pode variar amplamente entre os profissionais, aumentando o risco de inconsistência diagnóstica.
- **Atrasos no diagnóstico:** A análise manual pode ser um processo lento, especialmente quando a demanda por exames é alta, levando a atrasos no diagnóstico e no início do tratamento.
- **Fadiga do examinador:** Os citologistas podem sentir fadiga visual e mental ao examinar um grande número de lâminas, o que pode afetar a precisão da análise.

## 3. Técnicas Implementadas

### 3.1. Pré-Processamento

Pré-processamento é a etapa inicial do tratamento de dados brutos antes da análise ou modelagem. No nosso caso, essa fase incluiu tarefas específicas para preparar as imagens das células, conforme descrito abaixo:

- **Limpeza dos Dados:** A limpeza dos dados envolveu a extração de sub-imagens de células individuais a partir das imagens originais. Utilizando um arquivo CSV com coordenadas específicas (nucleus\_x, nucleus\_y) e classificações (Bethesda System), cada célula foi recortada para formar uma nova sub-imagem de 100x100 pixels.
- **Transformação:** Após a extração, as sub-imagens foram padronizadas para garantir consistência na análise. Isso incluiu a normalização dos dados para uma escala padrão e a modificação para ter média zero e desvio padrão um, facilitando a comparação entre diferentes células.
- **Organização:** As sub-imagens foram organizadas em diretórios correspondentes às suas classificações no Sistema Bethesda. Além disso, limitamos o número de sub-imagens por categoria a 1000 para manter um conjunto de dados equilibrado e manejável.

Essas etapas garantiram que os dados estivessem prontos e na resolução adequada (100x100 pixels) para análises e treinamentos futuros, proporcionando uma base sólida para modelos de aprendizado de máquina.

### 3.2. Interface

A interface do aplicativo foi desenvolvida na linguagem Python utilizando a biblioteca Tkinter, proporcionando uma interação intuitiva e funcional para os usuários. A interface esta no arquivo: "gui.py". As funcionalidades implementadas na interface incluem:

- **Upload de Imagem:** Permite ao usuário carregar imagens de exames de Papanicolau para análise.
- **Conversão para Tons de Cinza:** Transforma a imagem colorida em uma versão em tons de cinza, facilitando o processamento e a análise de características.
- **Geração de Histogramas:** Histogramas HSV que exibem a distribuição dos valores de matiz, saturação e valor da imagem e Histograma em Tons de Cinza que mostram a distribuição dos níveis de cinza na imagem.
- **Descritores de Haralick:** Calcula e apresenta os descritores de textura de Haralick, que são úteis para a identificação de padrões nas células.
- **Zoom:** Oferece a funcionalidade de aumentar e diminuir o zoom da imagem, permitindo uma análise mais detalhada das células.
- **Restaurar Coloração Original:** Permite ao usuário reverter a imagem para sua coloração original após a aplicação de filtros.

Tais funcionalidades combinadas geram uma ferramenta sólida para a análise de exames de Papanicolau, integrando técnicas de processamento de imagem em uma interface bastante intuitiva.

### 3.3. Rede Neural

A rede neural é fundamental para o reconhecimento do câncer, por sua capacidade de aprender características complexas das imagens, permitindo uma classificação precisa e automatizada das lesões e, os modelos robustos podem auxiliar no diagnóstico precoce e no tratamento eficaz das doenças cancerígenas.

Para o trabalho proposto, o código foi implementado na linguagem Python e foram utilizados dois classificadores rasos e dois profundos. Dessa forma, os conjuntos de treino e teste seguiram uma proporção de 4:1 das classes e 80% dos dados era o conjunto de teste e 20% o conjunto de treino.

#### 3.3.1. Classificadores Rasos:

O classificador raso escolhido para o modelo foi o SVM (Support Vector Machine). O SVM é uma forma de aprendizado supervisionado, utilizado em problemas de classificação e regressão. Ele é considerado um classificador raso devido à sua simplicidade conceitual e implementação direta.

O SVM foi trabalhado de duas maneiras: binária e multiclasse. O formato binário é o modelo mais simples de implementação, sendo treinado para distinguir entre duas classes: negativa e demais. A classe negativa representa uma condição que não indica a presença de uma doença ou anomalia. Já a classe das demais, representa a presença da doença ou anomalia diagnosticada.

Para a implementação do SVM binário, os dados foram preparados e treinados da seguinte maneira:

- **Extração de Características:** As características das imagens foram extraídas utilizando histogramas de cor HSV (Matiz, Saturação, Valor).
- **Codificação e Padronização das Classes:** As classes foram codificadas numericamente usando LabelEncoder e padronizadas usando StandardScaler, para terem média zero e variância unitária.
- **Inicialização do SVM:** Um SVM com kernel linear foi instanciado para o treinamento, implicando que o SVM usou um hiperplano linear para separar as classes no espaço de características.
- **Treinamento:** O modelo SVM foi treinado usando as características padronizadas e as labels binárias (binary\_labels\_train).

Para a implementação do SVM multiclasse, os dados foram preparados e treinados da seguinte maneira:

- **Codificação e Padronização das Classes:** As classes foram codificadas numericamente usando LabelEncoder e padronizadas usando StandardScaler, para terem média zero e variância unitária.
- **Inicialização do SVM:** Um SVM com kernel linear foi instanciado para o treinamento, implicando que o SVM usou um hiperplano linear para separar as classes no espaço de características.
- **Treinamento:** O modelo SVM foi treinado usando as características padronizadas e as labels multiclasse (train\_labels\_encoded).

### 3.3.2. Classificadores Profundos:

O classificador profundo escolhido para o modelo foi o ResNet50. O ResNet50 é uma arquitetura de CNN (rede neural convolucional) profunda que introduz os conceitos de "residual learning" e "transfer learning", sendo muito eficaz para tarefas complexas de visão computacional, como classificação de imagens.

Assim como o classificador raso SVM, o ResNet50 também foi trabalhado de duas maneiras: binária e multiclasse. O formato binário foi preparado de acordo com as duas classes de interesse: negativa e demais. A classe negativa representa uma condição que não indica a presença de uma doença ou anomalia. Já a classe das demais, representa a presença da doença ou anomalia diagnosticada.

Para a implementação do ResNet50 binário, os dados foram preparados e treinados da seguinte maneira:

- **Preparação dos Dados:** Os dados são preparados de maneira semelhante, mas agora usando class\_mode='binary' no ImageDataGenerator para lidar com a classificação binária entre lesões negativas e outras lesões.
- **Construção do Modelo Binário:** A ResNet50 é carregada com pesos pré-treinados no conjunto de dados ImageNet, excluindo a camada densa superior e mantendo as camadas convolucionais. Após o carregamento da base ResNet50, foram adicionadas camadas globais de média pooling, camadas densas, normalização em lote e dropout para regularização. A camada de saída utiliza ativação sigmoid para a classificação binária. Além disso, para utilizar os pesos pré-treinados da ResNet50 e evitar o ajuste excessivo nos primeiros estágios de treinamento, as camadas foram congeladas.

- **Treinamento:** O modelo foi treinado utilizando o gerador de dados de treinamento e validação, definindo o número de épocas e passos por época.

Para a implementação do ResNet50 multiclases, os dados foram preparados e treinados da seguinte maneira:

- **Preparação dos Dados:** Os dados foram preparados usando ImageDataGenerator do TensorFlow para realizar o aumento de dados nas imagens de treino e para normalização nos conjuntos de validação e teste. As classes são definidas explicitamente com `num_classes = 6`, correspondendo aos diferentes tipos de lesões.
- **Construção do Modelo:** O modelo ResNet50 é carregado com os pesos pré-treinados do ImageNet, excluindo a camada final (`include_top=False`). Em seguida, camadas personalizadas são adicionadas no topo do ResNet50 para adaptá-lo ao problema específico de classificação multiclasse.
- **Treinamento e Avaliação:** O modelo é treinado em duas fases: inicialmente com camadas base congeladas e depois com ajuste fino das camadas finais.

## 4. Análise dos Resultados

### 4.1. Rede Neural

#### 4.1.1. Resultados gerais

Os resultados evidenciam que a rede neural profunda ResNet50 supera consideravelmente o classificador raso SVM em tarefas de classificação binária e multiclasse. A ResNet50 alcança uma acurácia de 75% em ambas as configurações, enquanto o SVM apresenta apenas 83% na configuração binária e 37% na multiclasse. As matrizes de confusão revelam que o SVM falha na identificação precisa das classes, especialmente na configuração binária, ao passo que a ResNet50 demonstra maior robustez e taxas de acerto mais altas.

Comparando as métricas de classificação, a ResNet50 também se destaca, especialmente no recall (75%) e no F1-score (65%). Estes resultados reforçam a eficácia da ResNet50 na extração de características complexas dos dados, justificando sua preferência em aplicações de classificação onde a precisão é crucial. Em resumo, a ResNet50 é uma escolha superior e robusta em comparação com o SVM, alinhando-se com a tendência de utilizar redes neurais profundas para tarefas de classificação complexas. Esses resultados podem ser visualizados nas tabelas 2 3 4 5 6.

#### 4.1.2. Acurácia

A acurácia desempenha um papel fundamental na avaliação de modelos de redes neurais, pois representa a medida de precisão na classificação de dados. Em contextos onde a exatidão é crucial, como diagnósticos médicos baseados em imagens, a precisão dos resultados obtidos pode ter um impacto significativo na tomada de decisões clínicas. Sendo assim, visando aprimorar a acurácia da rede neural trabalhada, foram conduzidos diversos testes e refinamentos no modelo, como ajustes nos hiperparâmetros e balanceamento dos dados. Dessa forma, abaixo segue os resultados das tentativas de melhoria da acurácia em uma amostra de 1.000 imagens de cada classe:

Os resultados de acurácia dos modelos variam significativamente com diferentes abordagens de balanceamento de dados. Sem balanceamento, os modelos inicialmente

Balanceamento de Dados			
Classificador	Sem Balanceamento	Undersampling (17 treino e 65 teste)	Oversampling SMOTE
SVM Binário	71%	75%	53%
SVM Multiclasse	55%	47%	43%
Resnet50 Binário	18%	12%	40%
Resnet50 Multi-classe	30%	16%	35%

apresentam desempenhos variados. O uso de undersampling melhora a acurácia do SVM binário, enquanto o SVM multiclasse e a ResNet50 demonstram uma piora. Por outro lado, a técnica de oversampling SMOTE resulta em desempenhos mais padronizados entre os modelos. Com isso, podemos destacar a importância do balanceamento para melhorar a capacidade de generalização e precisão dos modelos em cenários de desbalanceamento de classes.

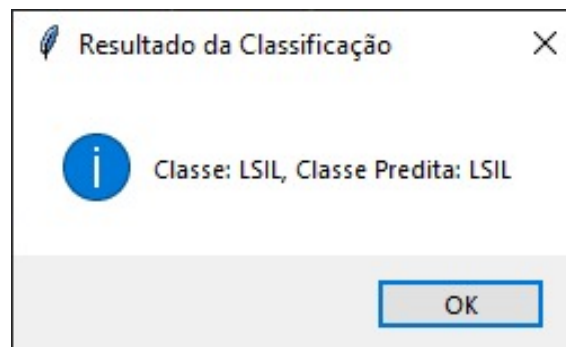
Ajuste nos Hiperparâmetros		
Classificador	1	2
SVM Binário	53%	73%
SVM Multiclasse	43%	18%
Resnet50 Binário	40%	42%
Resnet50 Multiclasse	35%	42%

**Tabela 1. Legenda:** O número 1 representa os hiperparâmetros `rotation_range=50`, `width_shift_range`, `height_shift_range`, `shear_range`, `zoom_range=0.3`, 1024 unidades, Dropout(0.5)(x).  
O número 2 representa `rotation_range=60`, `width_shift_range`, `height_shift_range`, `shear_range`, `zoom_range=0.4`, 512 unidades, Dropout(0.4)(x).

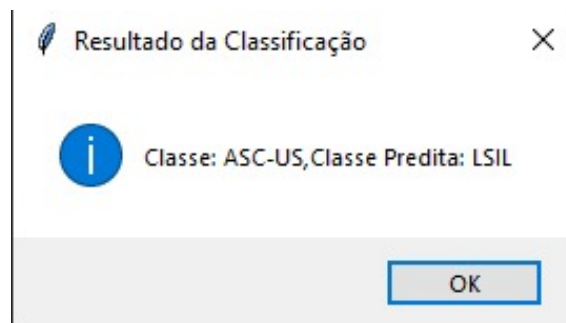
Os resultados variados de acurácia refletem a influência significativa dos hiperparâmetros na performance dos modelos. O aumento do valor de `rotation_range` para 60, o ajuste dos intervalos de `width_shift_range`, `height_shift_range`, `shear_range` e `zoom_range` para 0.4 e a redução das unidades na camada densa para 512, combinada com um dropout de 0.4 na ResNet50 provocaram melhorias notáveis na acurácia do SVM binário, alcançando 73%, indicando que esses parâmetros de aumento de dados aumentaram a capacidade do modelo de generalizar a partir dos dados de treinamento. Além disso, os modelos de ResNet50 binário e multiclasse obtiveram um aumento dessa métrica. No entanto, o SVM multiclasse teve desempenho inferior com esses ajustes. Portanto, a otimização dos hiperparâmetros demonstra ser crucial para maximizar a acurácia dos modelos, evidenciando a sensibilidade desses parâmetros na capacidade de generalização e desempenho das redes neurais.

## 4.2. Interface

Essas são capturas de tela da interface gráfica do usuário (gui.py) em funcionamento. Esta interface foi desenvolvida para facilitar o pré-processamento e a análise de imagens de Papanicolau, e desempenhou um papel crucial no suporte ao sistema de reconhecimento automático de células cancerosas.



**Figura 1. Resultado ResNet50**



**Figura 2. Resultado SVM**

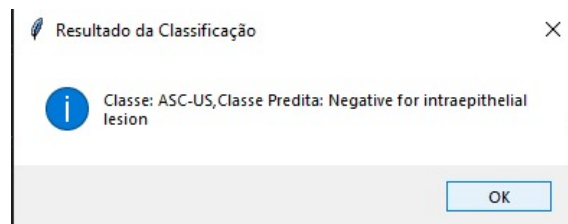
Os testes realizados, envolveram técnicas como ajustes de hiperparâmetros, demonstraram que o pré-processamento adequado e a manipulação de imagens são essenciais para otimizar o desempenho dos modelos. A interface implementada permite aplicar essas técnicas de forma eficiente e intuitiva, garantindo que as imagens estejam em uma condição ideal para o processamento pelos modelos de machine learning.

Essas capacidades da interface tornam-na uma ferramenta poderosa e alinhada com as necessidades dos modelos de aprendizado profundo discutidos no artigo, permitindo a obtenção de resultados com alta precisão e confiabilidade. Ao fornecer uma plataforma fácil de usar para o pré-processamento de imagens, a interface suporta diretamente os objetivos de melhorar a acurácia e a capacidade de generalização dos modelos de rede neural.

## **5. Conclusão**

Com base na abordagem desenvolvida neste estudo, é possível concluir que a criação de uma interface gráfica para análise de imagens de exames de Papanicolau, combinada com técnicas avançadas de processamento de imagens e aprendizado de máquina, representa um avanço significativo para a detecção precoce do câncer cervical. A implementação da ResNet50 como parte essencial desse processo demonstrou ser não apenas eficiente, mas também promissora na automação da classificação de lesões cervicais, superando métodos tradicionais como o SVM em termos de precisão e robustez.

Além de converter e analisar imagens para tons de cinza, gerar histogramas ou calcular descritores de textura, a interface desenvolvida oferece funcionalidades avançadas, como zoom e restauração da coloração original. Essas características são cruciais para



**Figura 3. Resultado ResNet Binário**

auxiliar os citologistas e profissionais de saúde na identificação precisa de anomalias celulares, proporcionando diagnósticos mais rápidos e confiáveis.

Este estudo valida não apenas a viabilidade técnica da implementação de uma interface gráfica robusta para análise de imagens médicas, mas também destaca sua importância na melhoria dos cuidados de saúde para mulheres em todo o mundo. Ao facilitar diagnósticos mais precisos e possibilitar tratamentos mais eficazes para o câncer cervical, essa abordagem não apenas salva vidas, mas também eleva a qualidade de vida das pacientes, oferecendo esperança e confiança na luta contra esta doença desafiadora.



## **6. Referências das Bibliotecas**

### **6.1. Numpy - 1.23.0**

Biblioteca científica que suporta grandes matrizes e arrays multidimensionais, junto com uma coleção de funções matemáticas de alto nível para operar nessas estruturas. No projeto, NumPy foi utilizado para operações de processamento de dados numéricos.

Instalação: `"pip install numpy"`

### **6.2. Pandas - 1.5.0**

Biblioteca de análise de dados que fornece estruturas de dados de alto desempenho e ferramentas de análise de dados fáceis de usar. Foi utilizada no projeto para manipulação e análise de dados tabulares.

Instalação: `"pip install pandas"`

### **6.3. Matplotlib - 3.6.2**

Biblioteca de plotagem 2D em Python que gera gráficos de alta qualidade. No projeto, foi utilizada para visualização de histogramas de imagens e outras representações gráficas.

Instalação: `"pip install matplotlib"`

### **6.4. OpenCV - 4.7.0**

Biblioteca para processamento de imagens e visão computacional. Foi usada no projeto para operações de pré-processamento de imagens, como conversão para tons de cinza e manipulação de histogramas.

Instalação: `"pip install opencv-python"`

### **6.5. Scikit-Image 0.21.0**

Biblioteca de ferramentas para processamento de imagens em Python. No projeto, foi usada para aplicar filtros e técnicas de análise de imagem.

Instalação: `"pip install scikit-image"`

### **6.6. TensorFlow - 2.14.0**

Plataforma de código aberto para machine learning e inteligência artificial. Foi utilizada para implementar modelos de reconhecimento de células cancerosas em imagens de Papanicolau.

Instalação: `"pip install tensorflow"`

### **6.7. Tkinter**

Biblioteca padrão do Python para criação de interfaces gráficas. No projeto, foi usada para desenvolver a interface gráfica do usuário (GUI) que permite o upload e visualização de imagens.

Instalação: Juntamente da instalação da linguagem Python.

6.8. Mahotas - 1.4.11

Biblioteca de processamento de imagem em Python que fornece ferramentas para a análise de texturas e outras operações. No projeto, foi usada para calcular os descritores de Haralick.

Instalação: "pip install mahotas"

7. Figuras

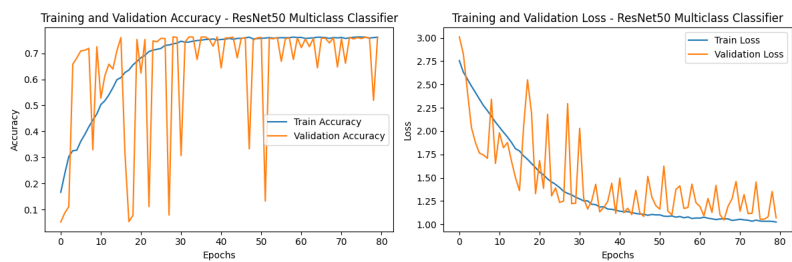


Tabela 2. Curva de aprendizado e perda do RestNet50 Multiclasses

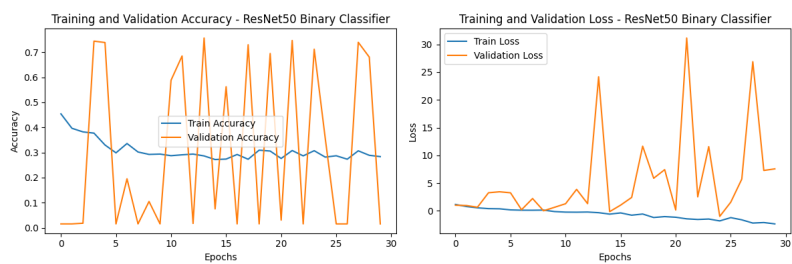


Tabela 3. Curva de aprendizado e perda do RestNet50 Binário

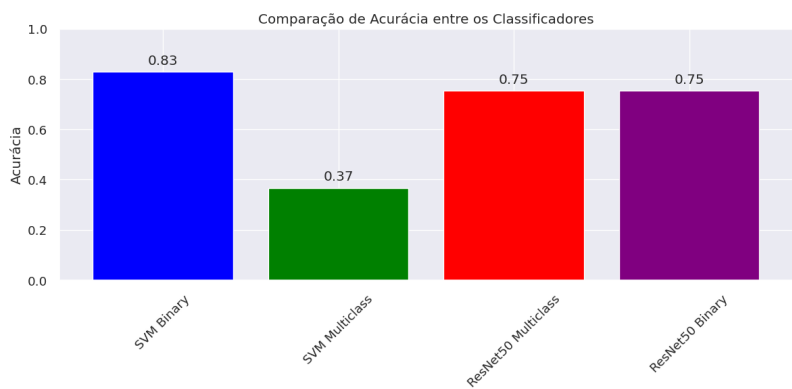
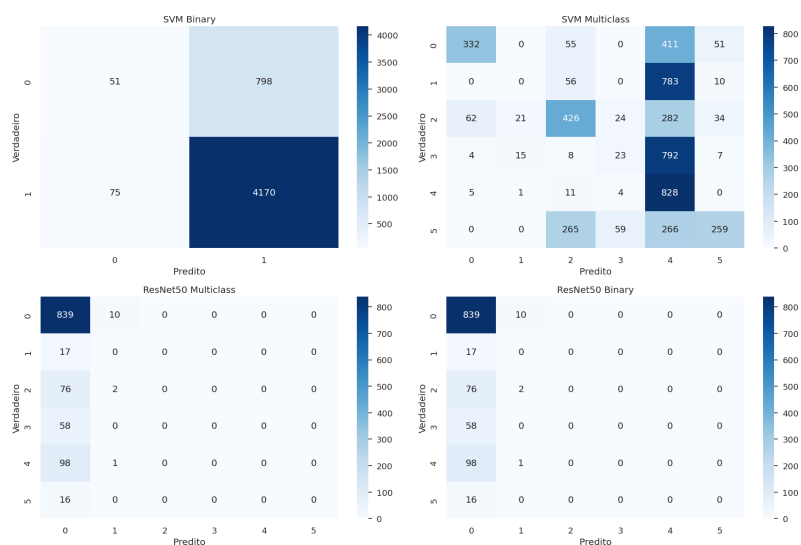
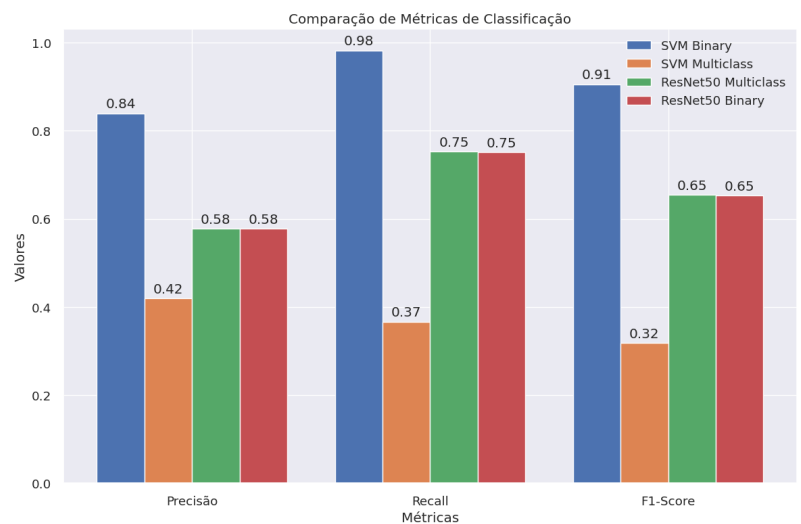


Tabela 4. Comparação da Acurácia dos 4 tipos de classificação (SVM binário, SVM multiclases, ResNet50 binário e ResNet50 multiclases)



**Tabela 5. Matriz de confusão dos 4 modelos**



**Tabela 6. Comparação das métricas de cada modelo**

## 8. Referências Bibliográficas

[Silva 2021], [Helder K. Shimo 2016], [Rezende 2021]

### Referências

Helder K. Shimo, Juliana P. Souza, R. F. B.-N. e. P. M. A.-M. (2016). Aplicação de descritores de haralick para caracterização de padrões radiológicos de doenças pulmonares difusas em tomografia computadorizada de alta resolução. Universidade de São Paulo (USP).

Rezende, M.T., S. R. B. F. (2021). Cric searchable image database as a public platform for conventional pap smear cytology data. Sci Data.

Silva, H. E. S. (2021). Classificação de lesões em imagens histológicas de mama usando wavelet e resnet-50. UNIVERSIDADE FEDERAL DE UBERLÂNDIA, 1º edição.