



Detecção de Borramento em Imagens Digitais

P.E.L.R. Meireles

H. Pedrini

Relatório Técnico - IC-PFG-19-05

Projeto Final de Graduação

2019 - Julho

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.
O conteúdo deste relatório é de única responsabilidade dos autores.

Detecção de Borramento em Imagens Digitais

Pedro Elias Lucas Ramos Meireles*

Hélio Pedrini[†]

Resumo

Este relatório descreve um projeto no qual foi realizada uma implementação de indicadores descritos na literatura de processamento de imagens digitais, os quais quantificam a resolução e o grau de borramento de imagens digitais. Estes indicadores foram utilizados em um banco de imagens para validar se eles podem ser utilizados para classificar se uma imagem está ou não borrada. Com base nestes indicadores e aplicando técnicas de aprendizado de máquina, conclui-se que é possível criar um classificador de imagens borradas e não borradas, utilizando tanto o algoritmo de Support Vector Classification (SVC) quanto o AdaBoost. Além disso, avaliou-se a necessidade de redimensionalidade das imagens originais para ganho de precisão do sistema ou de velocidade de processamento.

*Instituto de Computação, Universidade Estadual de Campinas, 13083-852 Campinas, SP.

[†]Instituto de Computação, Universidade Estadual de Campinas, 13083-852 Campinas, SP.

Acrônimos

ACMO Absolute Central Moment

BREN Brenner's Focus Measure

CONT Image Contrast

GDER Gaussian Derivative

GLLV Graylevel Local Variance

GLVA Graylevel Variance

GRAE Energy of Gradient

GRAS Squared Gradient

GRAT Threshold Gradient

HELM Helmi's Mean Method

HISE Histogram Entropy

HISR Histogram Range

LAPD Diagonal Laplacian

LAPE Energy of Laplacian

LAPM Modified Laplacian

LAPV Variance of Laplacian

SFF Spectral Feature Fitting

SFIL Steerable Filters

SFRQ Spatial Frequency

SVC Support Vector Classification

TENG Tenengrad

TENV Tenengrad Variance

Dedicatória

Este trabalho é dedicado aos meus amigos que me auxiliaram ao longo da minha graduação, prestando suporte e conselhos, tanto didáticos quanto emocionais; aos meus professores que me guiaram e me ensinaram os meios da academia, em especial, meu orientador de Trabalho de Conclusão de Curso de Graduação, Hédio Pedrini, e meu orientador de Iniciação Científica, Heiko Hornung; e aos meus familiares, que nunca se recusaram a me prestar apoio ao longo da minha vida, em destaque, meus pais, Maria Amélia, e José Carlos.

Um agradecimento profundo à Beatriz Malengo Andrade, por me acompanhar ao longo da conclusão do meu curso.

E a todos que chegaram até aqui.

Sumário

1	Introdução	5
2	Objetivos	5
3	Metodologia	5
3.1	Fluxograma do Modelo	6
3.2	Arquivos de Implementação	6
3.3	Indicadores de Foco	7
3.3.1	Absolute Central Moment (ACMO)	7
3.3.2	Brenner's Focus Measure (BREN)	7
3.3.3	Image Contrast (CONT)	8
3.3.4	Gaussian Derivative (GDER)	8
3.3.5	Graylevel Variance (GLVA)	8
3.3.6	Graylevel Local Variance (GLLV)	8
3.3.7	Energy of Gradient (GRAE)	9
3.3.8	Threshold Gradient (GRAT)	9
3.3.9	Squared Gradient (GRAS)	9
3.3.10	Helmli's Mean Method (HELM)	9
3.3.11	Histogram Entropy (HISE)	10
3.3.12	Histogram Range (HISR)	10
3.3.13	Energy of Laplacian (LAPE)	10
3.3.14	Modified Laplacian (LAPM)	10
3.3.15	Variance of Laplacian (LAPV)	11
3.3.16	Diagonal Laplacian (LAPD)	11
3.3.17	Steerable Filters (SFIL)	11
3.3.18	Spatial Frequency (SFRQ)	12
3.3.19	Tenengrad (TENG)	12
3.3.20	Tenengrad Variance (TENV)	12
3.4	Imagens	12
3.5	Aprendizado de Máquina	13
4	Resultados	13
4.1	Melhores Algoritmos	13
4.2	Matrizes de Confusão	14
4.3	Conclusões	15

1 Introdução

A área de processamento digital de imagens é um vasto campo na área da computação, cujo principal objetivo é ser uma ferramenta que auxilia diversos campos acadêmicos e industriais, de maneira a melhorar a qualidade ou compreensão de imagens para a interpretação humana, seja esta através de quantificação de atributos qualitativos de uma imagem, ou então na alteração de imagens para empregar uma melhoria na interpretação automática ou manual das imagens analisadas.

Uma das dificuldades desta área ocorre quando uma imagem é obtida por meio de um meio instável ou um equipamento com qualidade duvidosa, acarretando em imagens de péssima qualidade para seus observadores, tais como imagens borradas.

Borramento é o fenômeno descrito quando a qualidade de resolução de uma imagem é inferior à esperada pelos observadores e/ou para quem fotografou a imagem no respectivo momento. Este fenômeno pode ocorrer devido a vários motivos, tais como os descritos previamente.

A resolução de uma imagem pode ser mensurada por meio de várias métricas, cujos indicadores já foram discutidos ocasionalmente na literatura acadêmica. Entretanto, há ainda poucas abordagens que utilizam um conjunto destes indicadores para classificar, de maneira automática, se uma imagem está ou não borrada.

2 Objetivos

O principal objetivo deste trabalho é, com base em heurísticas e metodologias válidas em diferentes campos de estudo, determinar de maneira automática se uma imagem está borrada ou não, utilizando diferentes indicadores baseados na resolução de uma imagem, e empregando análises qualitativas, com processamentos de aprendizado de máquina na definição do classificador, tanto com um conjunto de imagens que são utilizadas originalmente (sem modificações) quanto com uma nova imagem obtida por meio de filtros que forcem o borramento das imagens originais.

Este estudo será feito tanto com um conjunto de imagens originais (o qual será descrito adiante) quanto com versões destas imagens sofrendo modificação de tamanho, para avaliar se essa mudança altera significativamente a qualidade do classificador.

Acredita-se que é possível categorizar imagens que estão borradas sem depender da avaliação supervisionada humana, desde que haja um conhecimento prévio se o conjunto de imagens do conjunto estão ou não borradas. Caso o conjunto de amostras só possua imagens não borradas, uma amostra de imagens borradas pode ser gerada de maneira procedural, aplicando filtros de borramento, tal como foi empregado neste projeto.

O sucesso deste experimento não deve ser inferido no conjunto de imagens utilizado, mas sim na metodologia aplicada e descrita ao longo do relatório.

3 Metodologia

Esta seção descreve as principais etapas que compõem a metodologia proposta.

3.1 Fluxograma do Modelo

O sistema de classificação de imagens borradas possui alguns fluxos principais para que sua funcionalidade ocorra. Para isso, os seguintes passos devem ser realizados:

- As imagens devem ser processadas, para que todos os indicadores sejam computados e registrados.
- Com os dados gerados no item anterior, estes dados devem ser processados em uma estratégia de validação cruzada, bem como na seleção do melhor algoritmo de aprendizado de máquina possível. O procedimento para realização do aprendizado será descrito mais adiante no relatório.
- Para os melhores algoritmos, será gerada a matriz de confusão do sistema, para avaliar a eficácia dos algoritmos empregados.

Para o experimento, alguns aspectos devem ser destacados:

- Analisou-se se a modificação do tamanho das imagens influencia os resultados. Então, o fluxo descrito foi feito com um conjunto de imagens nos seus tamanhos originais e, as mesmas imagens após sofrerem modificação de tamanho para 256×256 e 512×512 . O impacto destas modificações será discutido na seção de resultados.
- Em um conjunto real de dados, recomenda-se que seja selecionada uma amostra de imagens borradas e imagens não borradas. Como na realização dos experimentos não havia um conjunto de imagens com garantia de que estivessem borradas, o banco de imagens empregado supõe-se como imagens não borradas, e as imagens borradas foram geradas artificialmente, aplicando-se filtros de borramento da biblioteca OpenCV [1].

3.2 Arquivos de Implementação

Para realizar os objetivos previamente listados, foi implementado um programa em linguagem de programação Python com várias funcionalidades e módulos. A seguir, serão listados os nomes dos arquivos da aplicação e suas principais funcionalidades no projeto:

- `indicators.py`: implementa a iteração entre todas as imagens do diretório das imagens, com todos os indicadores implementados.
- `fmeasure.py`: implementa todos os indicadores de borramento, que serão descritos posteriormente.
- `plotter.py`: implementa os algoritmos de aprendizado de máquina no fluxograma e de processamento dos dados computados no `indicators.py`.
- `opener.py`: implementa uma classe responsável pela abertura das imagens utilizando o OpenCV.

- `fileManager.py`: implementa algumas funções de gerenciamento de arquivos, como localizar todas as imagens recursivamente em um diretório, ou mover todos os arquivos de um caminho para outro.
- `converter.py`: classe que converte as entradas de imagens coloridas em um modelo diferente. Neste trabalho, todas as imagens utilizadas foram convertidas para a escala de cinza.
- `main.py`: arquivo que faz as chamadas de função dos principais módulos (ou seja, o `indicators` e o `plotter`).

3.3 Indicadores de Foco

A seguir, serão explicadas quais foram as funções matemáticas dos operadores de foco [2] que servirão como indicadores para descrever se uma imagem encontra-se borrada ou não, pelo classificador implementado no projeto. Os nomes das funções serão escritos em inglês, mas os operadores serão descritos em português.

3.3.1 ACMO

Shirvaikar et al. [3] propuseram uma medida de foco, denominada ACMO, baseada em medidas estatísticas.

A definição da ACMO é dada pela operação realizada no histograma H da imagem:

$$ACMO = \sum_{k=1}^L |k - \mu| P_k, \quad (1)$$

em que μ é o valor de intensidade média de H , L são os níveis de escalas de cinza da imagem e P_k é a frequência relativa da imagem no k -ésimo nível da escala de cinza.

Espera-se que imagens borradas possuam menos resolução do que imagens não borradas. Isto implica que o histograma será mais agrupado para valores similares, o que faz com que o resultado geral da ACMO seja menor nas imagens borradas do que nas imagens não borradas.

3.3.2 BREN

Uma medida de foco baseada na segunda diferença de uma imagem I em nível de cinza é definida por [4–6]:

$$\phi = \sum_{(i,j)} \max(|I_{(i,j)} - I_{(i+2,j)}|, |I_{(i,j)} - I_{(i,j+2)}|)^2. \quad (2)$$

Essa medida considera tanto a segunda diferença no eixo X , quanto no eixo Y [5]. Espera-se que o resultado deste operador seja menor para imagens borradas do que para imagens não borradas.

3.3.3 CONT

Nanda et al. [7] utilizaram o contraste de imagem como medida de foco para auto-foco:

$$C_{(x,y)} = \sum_{i=x-1}^{x+1} \sum_{j=y-1}^{y+1} |I_{(x,y)} - I_{(i,j)}|. \quad (3)$$

Essa medida considera tanto a segunda diferença no eixo X quanto no eixo Y . Espera-se que o resultado deste operador seja menor para imagens borradas do que para imagens não borradas.

3.3.4 GDER

Baseado no modelo de defoco, Geusebroek et al. [8] propuseram uma medida de foco para auto-foco em microscópios baseada na primeira ordem da derivada da Gaussiana [8, 9]:

$$\phi = \sum_{(x,y)} ((I * \Gamma_x)^2 + (I * \Gamma_y)^2), \quad (4)$$

em que Γ_x e Γ_y são as derivadas parciais da função Gaussiana $\Gamma_{(x,y,\gamma)}$, a qual é definida por:

$$\Gamma_{(x,y,\gamma)} = \frac{1}{2\pi\gamma} e^{-\frac{x^2+y^2}{2\gamma^2}}. \quad (5)$$

Como operador de defoco, é esperado que, para imagens borradas, este valor seja menor do que para imagens sem borramento.

3.3.5 GLVA

A medida GLVA de uma imagem em escala de cinza é um dos métodos mais populares para computar o foco de uma imagem. Ela é aplicada tanto para auto-foco [4–6, 10–20] quanto para Spectral Feature Fitting (SFF) [21–24]:

$$\phi_{x,y} = \sum_{(i,j) \in \Omega(x,y)} (I(i,j) - \mu)^2, \quad (6)$$

em que μ é a média dos pixels pertencentes a $\Omega(x,y)$. É esperado que, para imagens borradas, este valor seja menor do que para imagens sem ou com poucos borramentos.

3.3.6 GLLV

Pech et al. [25] propuseram o GLLV como uma medida de foco para auto-foco de diatomáceas em microscopia de campo claro. Para essa aplicação da medida SFF, este operador é reformulado como:

$$\phi_{x,y} = \sum_{(i,j) \in \Omega(x,y)} (L_v(i,j) - \bar{L}_v)^2, \quad (7)$$

em que $L_v(i,j)$ é computado como a variância dos níveis de escala de cinza com uma vizinhança de tamanho $Wx \times Wy$, centralizada em (i,j) . \bar{L}_v é a média de L_v . É esperado que o valor da medida em imagens borradas seja menor do que para imagens não borradas.

3.3.7 GRAE

A soma dos quadrados da primeira derivada nas direções dos eixos X e Y de uma imagem também já foram propostos como medida de foco [10, 11, 23]:

$$\phi_{x,y} = \sum_{(i,j) \in \Omega(x,y)} (I_x(x,y)^2 + I_y(x,y)^2). \quad (8)$$

Espera-se que o resultado da medida em imagens borradas seja menor do que em imagens não borradas.

3.3.8 GRAT

Uma variação da medida GRAE é o GRAT, que considera um limiar T para definir a soma da média da operação [5]:

$$\phi = \sum_{(i,j) \in \Omega(x,y)} \max(|I_x(i,j)|, |I_y(i,j)|), \quad \text{para } \max(|I_x(i,j)|, |I_y(i,j)|) \geq T. \quad (9)$$

A eficácia dessa medida é afetada pela seleção de T . Por questão de generalidade e para se basear na referência bibliográfica que define a medida, o limiar T foi definido como 0. É esperado que imagens borradas possuam valores menores para esta medida do que as imagens não borradas.

3.3.9 GRAS

Ao invés de aplicar GRAT, as primeiras derivada são elevadas ao quadrado, para aumentar a influência de gradientes maiores [5, 6, 11, 26]. Além disso, ao invés de considerar os dois eixos X e Y , como foi considerado na medida GRAT, o valor de GRAS somente foi computado com o eixo X .

Tal como em GRAT, espera-se que o valor computado em imagens borradas seja menor do que em imagens não borradas.

3.3.10 HELM

Helmli e Scherer [21] propuseram uma medida de contraste local, computando a razão $R(x,y)$, entre os níveis de intensidade entre todos os pixels $I(x,y)$ e a média da escala de cinza na vizinhança $\mu(x,y)$:

$$R(x,y) = \begin{cases} \frac{\mu(x,y)}{I(x,y)}, & \text{se } \mu(x,y) \geq I(x,y) \\ \frac{I(x,y)}{\mu(x,y)}, & \text{caso contrário} \end{cases} \quad (10)$$

Essa razão será igual a 1 quando a região tiver valor constante ou de baixo contraste. Uma vizinhança $M \times N$, centralizada em (x,y) é usada para computar $\mu(x,y)$. O valor do foco para $I(x,y)$ é computado somando-se todos os valores de $R(x,y)$ em uma região $\Omega(x,y)$. É esperado que o valor desse indicador seja menor em imagens borradas do que em imagens não borradas.

3.3.11 HISE

Como imagens focadas costumam possuir um nível maior de conteúdo de informação, a entropia de uma imagem e a área de alcance de seu histograma podem ser utilizadas como medida de foco [4–6, 12, 13, 19].

A medida HISE é definida por:

$$\phi = - \sum_{k=1}^L (P_k \log_2(P_k)), \quad (11)$$

em que P_k é a frequência relativa do k -ésimo nível de cinza da imagem. Vale a ressalva de que valores menores são esperados em imagens borradas do que em relação a imagens não borradas.

3.3.12 HISR

O indicador HISR tem sido utilizado como medida de foco e auto foco como [4–6]:

$$\phi = \max(k|H > 0) - \min(k|H > 0). \quad (12)$$

É esperado que imagens borradas tenham um valor menor desta medida do que em imagens não borradas, por possuir uma carga de informação menor do que as imagens não borradas.

3.3.13 LAPE

A energia da segunda derivativa de uma imagem tem sido utilizado como medida tanto para auto foco [6, 9–12, 14–16, 18, 19, 27] quanto para SFF [28]:

$$\phi_{x,y} = \sum_{(i,j) \in \Omega(x,y)} \Delta I(i,j)^2, \quad (13)$$

em que ΔI é a imagem Laplaciana obtida ao se aplicar uma convolução entre a imagem I e o filtro Laplaciano.

É esperado que o valor da medida em imagens borradas seja menor do que em imagens não borradas.

3.3.14 LAPM

Nayar [29] propôs uma medida de foco baseada em uma definição alternativa do Laplaciano:

$$\phi_{x,y} = \sum_{(i,j) \in \Omega(x,y)} \Delta mI(i,j), \quad (14)$$

em que ΔmI é uma modificação do Laplaciano de I , computada como:

$$\Delta mI(i,j) = |I * \aleph_x| + |I * \aleph_y|. \quad (15)$$

As máscaras de convolução utilizadas para computar o Laplaciano modificado são:

$$\aleph_x = \begin{bmatrix} -1 & 2 & -1 \end{bmatrix} \quad (16)$$

e

$$\aleph_y = \aleph_x^T. \quad (17)$$

Esperam-se as mesmas conclusões de LAPE e LAPM.

3.3.15 LAPV

Essa medida utiliza a variância de uma imagem laplaciana como uma medida de foco para auto foco [25]. Em SFF, essa medida pode ser definida como:

$$\phi_{x,y} = \sum_{(i,j) \in \Omega(x,y)} (\Delta I(i,j) - \bar{\Delta I})^2, \quad (18)$$

em que $\bar{\Delta I}$ é a média do valor da imagem laplaciana em $\Omega(x,y)$.

Esperam-se as mesmas conclusões de LAPE se apliquem em LAPV.

3.3.16 LAPD

Thelen et al. [30] também incluíram variações na vertical em uma imagem para computar uma versão modificada da Laplaciana da imagem:

$$\Delta m I(i,j) = |I * \aleph_x| + |I * \aleph_y| + |I * \aleph_{x1}| + |I * \aleph_{x2}|, \quad (19)$$

em que \aleph_x e \aleph_y foram definidos anteriormente, e \aleph_{x1} e \aleph_{x2} são definidos por:

$$\aleph_{x1} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 0 & 1 \\ 0 & -2 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\aleph_{x2} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

3.3.17 SFIL

Minhas et al. [22] propuseram uma medida de foco baseada em uma versão filtrada da imagem I_f :

$$\phi_{x,y} = \sum_{(i,j) \in \Omega(x,y)} I_f(x,y), \quad (20)$$

em que I_f é definida como:

$$I_f(x,y) = \max_{(i,j)} R_{(i,j)}^{\Theta_n}, \quad (21)$$

em que R^{Θ_n} , $n = 1, 2, \dots, N$, é a imagem responsável pela n -ésima rotação do filtro definido por [31]:

$$R^{\Theta_n} = \cos(\Theta_n)(I * \Gamma_x) + \sin(\Theta_n)(I * \Gamma_y), \quad (22)$$

em que Γ_x e Γ_y são as derivadas da Gaussiana nos eixos X e Y , respectivamente.

3.3.18 SFRQ

Este operador [11] foi proposto para fusão de imagens multi-focais:

$$\phi_{x,y} = \sqrt{\sum_{(i,j) \in \Omega(x,y)} I_x(i,j)^2 + \sum_{(i,j) \in \Omega(x,y)} I_y(i,j)^2}, \quad (23)$$

em que I_x e I_y denotam a primeira derivada da imagem na direção dos eixos X e Y , respectivamente.

3.3.19 TENG

Este é um operador de medida de foco baseado na magnitude do gradiente da imagem, definido por [5, 6, 10–18, 21–23, 25, 27, 32]:

$$\phi_{x,y} = \sum_{(i,j) \in \Omega(x,y)} (G_x(x,y)^2 + G_y(x,y)^2) \quad (24)$$

em que G_x e G_y são os gradientes nos eixos X e Y computados pela convolução da respectiva imagem com o operador de Sobel.

3.3.20 TENV

Este operador usa a variância do gradiente da imagem como medida de foco. Ele originalmente foi utilizado como medida de auto-foco [25], mas também pode ser aplicado para SFF:

$$\phi_{x,y} = \sum_{(i,j) \in \Omega(x,y)} (G(x,y) - G(\bar{x},y))^2, \quad (25)$$

em que $G(\bar{x},y)$ é a média do valor em $\Omega(x,y)$ do gradiente da magnitude, o qual é calculado como:

$$G = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \quad (26)$$

3.4 Imagens

As imagens foram obtidas de acervos digitais públicos [33, 34] de estudos em processamento de imagens. No total, foram utilizadas cerca de 9.000 imagens ao longo do estudo. Todas as imagens utilizadas foram consideradas não borradas e foram catalogadas no processamento dos indicadores como não borradas.

Para gerar o acervo de imagens borradas, foram aplicados dois filtros da biblioteca OpenCV [1]: o filtro *blur* e o filtro Gaussiano. Além disso, para realizar a análise qualitativa do sistema de classificação, foi questionado se o tamanho da imagem influencia nos resultados. Portanto, foram geradas 3 amostras de dados para o mesmo conjunto de imagens no sistema como um todo: (a) as imagens no tamanho original; (b) as imagens com mudança de escala para 256×256 pixels; (c) as imagens com mudança de escala para 512×512 pixels.

Estes conjuntos foram analisados separadamente pelos algoritmos de aprendizado de máquina e serão discutidos brevemente na seção de resultados.

3.5 Aprendizado de Máquina

No módulo de aprendizado de máquina, utilizou-se a biblioteca em linguagem de programação Python chamada `scikit-learn` [35]. Ela possui uma documentação clara de como utilizar os algoritmos mais populares, assim como uma descrição matemática dos mesmos.

Para realizar o treinamento, a seguinte lista de algoritmos descritos na biblioteca foi utilizada:

- SVC [36].
- Classificador AdaBoost [37].
- Classificador de Perceptron multicamada [38].
- Um classificador de decisão em árvore [39].
- Um classificador de árvore aleatória [40].
- Algoritmo de classificação de Naïve Bayes Gaussiano [41].
- Um classificador implementando o voto do k -ésimo vizinho próximo [42].

Para a seleção do melhor algoritmo, foi utilizada a técnica de validação cruzada [43], separando o conjunto de teste com razão de 15% em relação ao conjunto de todos os dados, sendo o remanescente para o conjunto de treinamento. Dentro do conjunto de treinamento, foi realizado um particionamento de 5 partes utilizando a estratégia K -Fold [44], sendo que o melhor conjunto de dados obtido foi avaliado nos demais algoritmos.

Para os três melhores algoritmos, foi gerada a matriz confusão [45] para avaliar o desempenho dos algoritmos. Deve-se ressaltar que alguns dos classificadores necessita fazer um afinamento dos hiper-parâmetros junto com a validação cruzada [46]. Outra observação é que o conjunto de testes foi avaliado com a melhor partição detectada durante o período de treinamento, não utilizando todo o conjunto de treinamento no mesmo.

4 Resultados

Esta seção descreve os resultados experimentos obtidos com a metodologia proposta neste trabalho.

4.1 Melhores Algoritmos

Na Tabela 1, são apresentados os três melhores algoritmos para cada modelo do banco de dados registrado. Como podemos notar, não há modificação em relação aos 2 melhores algoritmos para todos os modelos, tendo soberania no modelo SVC. Vale a ressalva de que o modelo mais preciso de todos se encontrou nas imagens que não sofreram modificação de tamanho, enquanto que os menos precisos se encontraram na modificação de tamanho de 512x512.

Sem mudança de tamanho	256×256	512×512
1º SVC (0,9969)	SVC (0,9964)	SVC (0,9816)
2º AdaBoost (0,9968)	AdaBoost (0,9957)	AdaBoost (0,9799)
3º Multi Layer Perceptron (0,9935)	Multi Layer Perceptron (0,9955)	Árvore de Decisão (0,9795)

Tabela 1: Resultados de acurácia dos melhores algoritmos, para todos os conjuntos de dados extraídos (considerando se houve mudança de tamanho e quais são os novos tamanhos, se houve).

Logo, pode-se inferir que não é necessário modificar o tamanho da imagem para aumentar a acurácia do sistema de aprendizado de máquina. O que se ganha fazendo isso é velocidade de processamento da parte da geração dos indicadores, no cenário em que houve redução do tamanho da imagem original (caso isto não se aplique, não há ganho algum em reduzir o tamanho da imagem).

4.2 Matrizes de Confusão

Nesta seção, são apresentadas as matrizes de confusão da solução sem re-escala de tamanho, dos melhores algoritmos calculados.

1. SVC: Acurácia de 99,76% com conjunto de teste. Matriz mostrada na Tabela 2.

		Predição	
		Não Borrado	Borrado
Valor Real	Não Borrado	1361	9
	Borrado	1	2753

Tabela 2: Matriz de confusão do SVC.

2. Classificador AdaBoost: Acurácia de 99,52% no conjunto de teste. Matriz mostrada na Tabela 3.

		Predição	
		Não Borrado	Borrado
Valor Real	Não Borrado	1356	14
	Borrado	6	2748

Tabela 3: Matriz de confusão do classificador AdaBoost.

3. Classificador Multicamadas Perceptron: Acurácia de 99,37% no conjunto de teste. Matriz mostrada na Tabela 4.

		Predição	
		Não Borrado	Borrado
Valor Real	Não Borrado	1346	24
	Borrado	2	2752

Tabela 4: Matriz de confusão do classificador Multicamadas Perceptron.

Como é possível observar, no conjunto de teste, o resultado de alta acurácia e acertos ainda é válido para os mesmos algoritmos no conjunto de treinamento. Além disso, pode-se notar que há poucas incidências de falsos positivos em todos os cenários.

4.3 Conclusões

A partir dos resultados gerados neste trabalho, os indicadores de resolução foram aptos a descrever, de maneira satisfatória, se uma imagem está ou não borrada, dado um conjunto de imagens borradas e não borradas, com o uso de algoritmos de aprendizado de máquina. A lógica de gerar imagens borradas a partir de imagens não borradas foi realizada somente para obter um conjunto de imagens borradas e, em cenários menos controlados, é recomendado que o usuário possua tanto imagens borradas quanto não borradas.

O SVC foi o melhor algoritmo para todos os conjuntos de imagens, entretanto, seu treinamento, por exigir a utilização de hiper-parâmetros, é um dos mais custosos em tempo para gerar o aprendizado. O AdaBoost também é uma solução prática interessante e possui um treinamento mais rápido do que seu concorrente. Logo, se tempo para realização do aprendizado for um critério mais importante do que eficácia, o AdaBoost é uma opção interessante.

O redimensionamento das imagens não alterou significativamente a eficácia do sistema. No entanto, caso seja necessário que a computação dos indicadores seja mais rápida, sugere-se que a proporção entre os eixos x e y da imagem sejam preservados e que a redimensão seja em escalas pequenas em relação à original, caso contrário, pode haver perda de precisão no sistema de classificação.

Referências

- [1] OpenCV, “OpenCV Modules,” <https://docs.opencv.org/3.4.0/index.html>, (Acesso em 22/06/2019).
- [2] S. Pertuz, D. Puig, and M. A. Garcia, “Analysis of Focus Measure Operators for Shape-from-Focus,” *Pattern Recognition*, vol. 46, no. 5, pp. 1415–1432, 2013.
- [3] M. V. Shirvaikar, “An Optimal Measure for Camera Focus and Exposure,” in *Thirty-Sixth Southeastern Symposium on System Theory*. IEEE, 2004, pp. 472–475.
- [4] L. Firestone, K. Cook, K. Culp, N. Talsania, and K. Preston Jr, “Comparison of Autofocus Methods for Automated Microscopy,” *Cytometry: The Journal of the International Society for Analytical Cytology*, vol. 12, no. 3, pp. 195–206, 1991.

- [5] A. Santos, C. Ortiz de Solórzano, J. J. Vaquero, J. Pena, N. Malpica, and F. Del Pozo, "Evaluation of Autofocus Functions in Molecular Cytogenetic Analysis," *Journal of microscopy*, vol. 188, no. 3, pp. 264–272, 1997.
- [6] Y. Sun, S. Duthaler, and B. J. Nelson, "Autofocusing in Computer Microscopy: Selecting the Optimal Focus Algorithm," *Microscopy Research and Technique*, vol. 65, no. 3, pp. 139–149, 2004.
- [7] H. Nanda and R. Cutler, "Practical Calibrations for a Real-Time Digital Omnidirectional Camera," *IEEE Conference on Computer Vision and Pattern Recognition - Technical Sketch*, vol. 20, p. 2, 2001.
- [8] J.-M. Geusebroek, F. Cornelissen, A. W. Smeulders, and H. Geerts, "Robust Autofocusing in Microscopy," *Cytometry: The Journal of the International Society for Analytical Cytology*, vol. 39, no. 1, pp. 1–9, 2000.
- [9] M. J. Russell and T. S. Douglas, "Evaluation of Autofocus Algorithms for Tuberculosis Microscopy," in *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2007, pp. 3489–3492.
- [10] M. Subbarao, T.-S. Choi, and A. Nikzad, "Focusing Techniques," *Optical Engineering*, vol. 32, no. 11, pp. 2824–2837, 1993.
- [11] W. Huang and Z. Jing, "Evaluation of Focus Measures in Multi-Focus Image Fusion," *Pattern Recognition Letters*, vol. 28, no. 4, pp. 493–500, 2007.
- [12] N. N. K. Chern, P. A. Neow, and M. H. Ang, "Practical Issues in Pixel-based Autofocusing for Machine Vision," in *IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, vol. 3. IEEE, 2001, pp. 2791–2796.
- [13] E. Krotkov and J.-P. Martin, "Range from Focus," in *IEEE International Conference on Robotics and Automation*, vol. 3. IEEE, 1986, pp. 1093–1098.
- [14] J.-H. Lee, K.-S. Kim, B.-D. Nam, J.-C. Lee, Y.-M. Kwon, and H.-G. Kim, "Implementation of a Passive Automatic Focusing Algorithm for Digital Still Camera," *IEEE Transactions on Consumer Electronics*, vol. 41, no. 3, pp. 449–454, 1995.
- [15] C.-H. Shen and H. H. Chen, "Robust Focus Measure for Low-Contrast Images," in *Digest of Technical Papers International Conference on Consumer Electronics*. IEEE, 2006, pp. 69–70.
- [16] C.-Y. Wee and R. Paramesran, "Measure of Image Sharpness using Eigenvalues," *Information Sciences*, vol. 177, no. 12, pp. 2533–2552, 2007.
- [17] G. Yang and B. J. Nelson, "Wavelet-based Autofocusing and Unsupervised Segmentation of Microscopic Images," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (Cat. No. 03CH37453)*, vol. 3. IEEE, 2003, pp. 2143–2148.

- [18] P. T. Yap and P. Raveendran, "Image Focus Measure based on Chebyshev Moments," *IEE Proceedings - Vision, Image and Signal Processing*, vol. 151, no. 2, pp. 128–136, 2004.
- [19] H. Xie, W. Rong, and L. Sun, "Wavelet-based Focus Measure and 3-D Surface Reconstruction Method for Microscopy Images," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 229–234.
- [20] J. Baina and J. Dublet, "Automatic Focus and Iris Control for Video Cameras," in *Fifth International Conference on Image Processing and its Applications*. IET, 1995.
- [21] F. S. Helmli and S. Scherer, "Adaptive Shape from Focus with an Error Estimation in Light Microscopy," in *2nd International Symposium on Image and Signal Processing and Analysis. In conjunction with 23rd International Conference on Information Technology Interfaces*. IEEE, 2001, pp. 188–193.
- [22] R. Minhas, A. A. Mohammed, Q. J. Wu, and M. A. Sid-Ahmed, "3D Shape from Focus and Depth Map Computation using Steerable Filters," in *International Conference Image Analysis and Recognition*. Springer, 2009, pp. 573–583.
- [23] A. S. Malik and T.-S. Choi, "A Novel Algorithm for Estimation of Depth Map using Image Focus for 3D Shape Recovery in the Presence of Noise," *Pattern Recognition*, vol. 41, no. 7, pp. 2200–2225, 2008.
- [24] Y. An, G. Kang, I.-J. Kim, H.-S. Chung, and J. Park, "Shape from Focus through Laplacian using 3D Window," in *Second International Conference on Future Generation Communication and Networking*, vol. 2. IEEE, 2008, pp. 46–50.
- [25] J. L. Pech-Pacheco, G. Cristóbal, J. Chamorro-Martinez, and J. Fernández-Valdivia, "Diatom Autofocusing in Brightfield Microscopy: A Comparative Study," in *15th International Conference on Pattern Recognition*, vol. 3. IEEE, 2000, pp. 314–317.
- [26] A. M. Eskicioglu, P. S. Fisher, and S.-Y. Chen, "Image Quality Measures and their Performance," *IEEE Transactions on Communications*, vol. 43, no. 12, pp. 2959–2965, 1995.
- [27] S.-Y. Lee, J.-T. Yoo, Y. Kumar, and S.-W. Kim, "Reduced Energy-Ratio Measure for Robust Autofocusing in Digital Camera," *IEEE Signal Processing Letters*, vol. 16, no. 2, pp. 133–136, 2009.
- [28] M. B. Ahmad and T. S. Choi, "Application of Three Dimensional Shape from Image Focus in LCD/TFT Displays Manufacturing," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 1, pp. 1–4, 2007.
- [29] S. K. Nayar, "Shape from Focus," Carnegie-Mellon University Pittsburgh, PA, Robotics Institute, Tech. Rep., 1989.

- [30] A. Thelen, S. Frey, S. Hirsch, and P. Hering, “Improvements in Shape-from-Focus for Holographic Reconstructions with Regard to Focus Operators, Neighborhood-Size, and Height Value Interpolation,” *IEEE Transactions on Image Processing*, vol. 18, no. 1, pp. 151–157, 2008.
- [31] W. T. Freeman and E. H. Adelson, “The Design and Use of Steerable Filters,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 9, pp. 891–906, 1991.
- [32] H. N. Nair and C. V. Stewart, “Robust Focus Ranging,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1992, pp. 309–314.
- [33] “Caltech101,” http://www.vision.caltech.edu/Image_Datasets/Caltech101/, (Acesso em 22/06/2019).
- [34] “Test images,” <https://homepages.cae.wisc.edu/~ece533/images/>, (Acesso em 22/06/2019).
- [35] “scikit-learn: machine learning in python — scikit-learn 0.21.2 documentation,” <https://scikit-learn.org/stable/>, (Acesso em 22/06/2019).
- [36] “sklearn.svm.svc — scikit-learn 0.21.2 documentation,” <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>, (Acesso em 22/06/2019).
- [37] “sklearn.ensemble.adaboostclassifier — scikit-learn 0.21.2 documentation,” <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html>, (Acesso em 22/06/2019).
- [38] “sklearn.neural_network.mlpclassifier — scikit-learn 0.21.2 documentation,” https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html, (Acesso em 22/06/2019).
- [39] “sklearn.tree.decisiontreeclassifier — scikit-learn 0.21.2 documentation,” <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>, (Acesso em 22/06/2019).
- [40] “sklearn.ensemble.randomforestclassifier — scikit-learn 0.21.2 documentation,” <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>, (Acesso em 22/06/2019).
- [41] “sklearn.naive_bayes.gaussiannb — scikit-learn 0.21.2 documentation,” https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html, (Acesso em 22/06/2019).
- [42] “sklearn.neighbors.kneighborsclassifier — scikit-learn 0.21.2 documentation,” <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>, (Acesso em 22/06/2019).

- [43] “Cross-validation: evaluating estimator performance — scikit-learn 0.21.2 documentation,” https://scikit-learn.org/stable/modules/cross_validation.html, (Acesso em 22/06/2019).
- [44] “sklearn.model_selection.kfold — scikit-learn 0.21.2 documentation,” https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html, (Acesso em 22/06/2019).
- [45] “sklearn.metrics.confusion_matrix — scikit-learn 0.21.2 documentation,” https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html, (Acesso em 22/06/2019).
- [46] “Parameter Estimation using Grid Search with Cross-Validation — scikit-learn 0.21.2 documentation,” https://scikit-learn.org/stable/auto_examples/model_selection/plot_grid_search_digits.html, (Acesso em 22/06/2019).