

UNIVERSIDADE FEDERAL DO PARANÁ

GÜNTER HEINRICH HERWEG FILHO

DIAGNÓSTICO DE IMUNOENSAIOS DE FLUXO LATERAL POR MEIO  
DE RECONHECIMENTO DE PADRÕES

CURITIBA PR

2018

GÜNTER HEINRICH HERWEG FILHO

DIAGNÓSTICO DE IMUNOENSAIOS DE FLUXO LATERAL POR MEIO  
DE RECONHECIMENTO DE PADRÕES

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Informática no Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Luiz Eduardo S. Oliveira.

CURITIBA PR

2018

Catálogo na Fonte: Sistema de Bibliotecas, UFPR  
Biblioteca de Ciência e Tecnologia

---

H581d Herweg Filho, Günter Heinrich  
Diagnóstico de Imunoensaios de fluxo lateral por meio de reconhecimento de padrões [Recurso eletrônico] / Günter Heinrich Herweg Filho – Curitiba, 2018.

Dissertação (mestrado) - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-graduação em Informática, 2018.

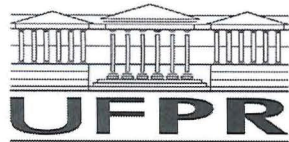
Orientador: Luiz Eduardo Soares Oliveira

1. Reconhecimento de padrões. 2. Processamento de imagens. 3. Testes rápidos (Computação). I. Universidade Federal do Paraná. II. Oliveira, Luiz Eduardo Soares. III. Título.

CDD: 006.42

---

Bibliotecária: Roseny Rivelini Morciani CRB-9/1585



MINISTÉRIO DA EDUCAÇÃO  
SETOR SETOR DE CIÊNCIAS EXATAS  
UNIVERSIDADE FEDERAL DO PARANÁ  
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO INFORMÁTICA

## TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **GÜNTER HEINRICH HERWEG FILHO** intitulada: **Diagnóstico de Imunoensaios de fluxo lateral por meio de reconhecimento de padrões**, após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 19 de Novembro de 2018.

LUIZ EDUARDO SOARES DE OLIVEIRA  
Presidente da Banca Examinadora (UFPR)

LEONARDO FOTI  
Avaliador Externo (IBMP)

LUCAS FERRARI DE OLIVEIRA  
Avaliador Interno (UFPR)



*Dedico a Deus e a minha família.*

## **AGRADECIMENTOS**

Ao meu orientador, professor Luiz Eduardo S. Oliveira, cujo conhecimento e experiência me permitiu realizar este trabalho

Ao meu primo e amigo, professor Daniel Weingaertner e minha prima e amiga Lísia Weingaertner, pelo incentivo e apoio incondicional.

Ao meu co-orientador Leonardo Foti e Bruno Dias Ramos por suas valiosas explicações, sempre muito presentes para tirar as dúvidas e pelo trabalho com a base de dados usada neste trabalho.

A minha tia Simone Westphal pelas revisões, correções no texto e pela torcida.

Minha família, pela paciência ao longo desta caminhada.

Também agradeço a todos meus amigos e familiares que de alguma maneira contribuíram para que eu pudesse terminar meu trabalho.

## RESUMO

Os testes imunológicos rápidos, ou *lateral flow immunoassays* (LFIA), são testes simples, que geram uma resposta rápida e sem que, para isso, tenham que passar por processos laboratoriais. Os testes para detecção de gravidez provavelmente sejam o exemplo mais comum de sua aplicação. Todavia, os imunoensaios de fluxo lateral são agora usados em diversas áreas, como para detecção de contaminantes na água, nos alimentos, nas substâncias ilícitas no organismo, mas principalmente na medicina, para detecção de doenças. Seu resultado é visível por meio da mudança de cor das linhas presentes na tira. Para testes quantitativos, a intensidade da cor é proporcional à concentração do elemento procurado na amostra e, por isso, necessitam do auxílio de leitoras para revelar o seu resultado. Esses dispositivos geralmente são equipados com uma câmera, que adquire a imagem da tira para a análise. Em outros casos, usam alguma tecnologia de escaneamento, excitando os pontos nas linhas de teste e controle, por meio do auxílio da luz, para detectá-las e classificá-las. Valores de corte são usados para eliminar ou validar os testes.

Os resultados deste trabalho mostram um ganho significativo no desempenho da classificação dos testes rápidos por meio de reconhecimento de padrões e algoritmos de aprendizado de máquina se comparados aos tradicionais métodos de classificação. Os problemas causados na aquisição de imagens como variações na luminosidade ou ruídos foram minimizados, indicando boa generalização do modelo.

**Palavras-chave:** Testes rápidos. Classificação. Processamento de imagens. Reconhecimento de padrões. Aprendizado de máquina.

## ABSTRACT

Lateral flow immunoassays (LFIA) are simple tests that allow a rapid response without the need of laboratories processes. Pregnancy test kits may be the most common example of your application. However, lateral flow immunoassays are now used in a number of areas, including detection of contaminants in water, food or illicit substances in the body, but especially in medicine, for diseases detection. For quantitative tests the color intensity is proportional to the amount of target element in the sample. Generally is combined with a reader technology for measurement of the intensity of colors produced at test and control lines of strip. These devices are usually equipped with a camera to acquire an image of the strip for analysis or use some scanning technology, exciting the points on the test and control lines to detect and classify them. Cut values are used to eliminate or validate the tests.

The results obtained show a significant improvement in the performance of the Lateral flow assays classification through pattern recognition and machine learning algorithms compared to traditional classification methods. Problems caused at acquisition time as variations in luminosity or noise were minimized in this model.

**Keywords:** Lateral flow assay. Classification. Image processing. Pattern recognition. Machine learning.

## LISTA DE FIGURAS

2.1	Antígenos com epítomos se ligam ao paratopo dos anticorpos. Fonte: [Copstead e Banasik, 2000].	17
2.2	Estrutura do teste. A solução segue pela membrana da amostra, passando pelas linhas de teste e de controle até ser absorvida pela membrana de absorção. . . . .	18
2.3	Funcionamento do teste sanduíche usando ouro coloidal como molécula reporter. Neste formato a intensidade do sinal é diretamente proporcional a presença de analito na amostra. . . . .	20
2.4	Etapas do processo de classificação de imagens. . . . .	20
2.5	[scikit image, 2018]. Aplicação da limiarização de <i>Otsu</i> em uma imagem. . . . .	21
2.6	a) Canal V da imagem de uma tira teste na representação de cores HSV. b) Projeção de histograma lateral da imagem (linha laranja). A curva da projeção suavizada pelo método ALS (linha azul). Na figura estão identificados dois picos de intensidade, a linha amarela representa a largura da curva na região do pico. . . . .	23
2.7	Exemplo de cálculo do histograma LBP com raio=1 e vizinhança= 8. O valor do elemento central funciona como limiar para valorar os vizinhos. É gerado um número binário de 8 bits, que então é convertido para decimal. . . . .	24
2.8	Os ângulos definem a direção dos <i>pixels</i> a serem considerados em um raio de tamanho <i>D</i> . . . . .	24
2.9	Aplicação das propriedades do GLCM. Para cada tipo de propriedade um padrão diferente é ressaltado. . . . .	25
2.10	Camadas típicas de uma rede neural convolucional. Divisão das etapas de extração de características e classificação. . . . .	26
2.11	Fluxograma da solução <i>end-to-end</i> . Etapas tradicionais de pré-processamento e segmentação não estão presentes nesta metodologia. . . . .	27
2.12	Exemplo de um gráfico que mostra a curva ROC, em vermelho. . . . .	28
3.1	Processo de correspondência entre descritores FAST e BRIEF (ORB) na imagem de referência (à esquerda) e na imagem original (à direita). Linhas brancas denotam os pontos correspondentes. Em verde, é representada a região de interesse. . . . .	30
3.2	Comparativo entre diferentes repositórios de imagens com os dados das fitas com concentração de álcool na saliva usando o padrão de cores RGB. . . . .	31
4.1	Imagens das leitoras e um exemplo de imagem gerada. . . . .	32
4.2	Imagens de bandas segmentadas. (a) Exemplo de banda reagente. (b) Exemplo de banda não reagente. (c) Exemplo de banda inválida. . . . .	33
4.3	Exemplos de imagens aumentadas de cada uma das duas bases. . . . .	34
5.1	As etapas da metodologia 1. Pré-processamento, segmentação, extração de características e classificação. . . . .	35
5.2	O fluxograma mostra as etapas do modelo. A extração acontece nos segmentos e não na imagem inteira. . . . .	36
5.3	Etapas da metodologia <i>end-to-end</i> . . . . .	37
5.4	a) Imagem original. b) A imagem com contraste aumentado. c) A imagem b suavizada com filtro bilateral. . . . .	37
5.5	O resultado do primeiro passo da segmentação gera dois segmentos. . . . .	38
5.6	Processo de segmentação com uma imagem com 2 linhas. A partir da imagem original, são obtidos 2 segmentos correspondentes as regiões de controle e teste. Os contornos (em amarelo) mostram se o segmento contém uma linha ou não contém. . . . .	39

5.7	Processo de segmentação com uma imagem com 1 linha. Essa imagem mostra que através dos contornos obtidos é possível saber se tal contorno corresponde a uma linha ou não. A última imagem da figura mostra um contorno que não corresponde a uma linha. . . . .	39
5.8	a) A imagem binarizada. b) Projeção de histograma lateral. c) Projeção de histograma vertical. . . . .	43
5.9	Camadas da CNN . . . . .	44
6.1	Curvas R.O.C. do modelo com RF e conjunto original. . . . .	46
6.2	Curvas R.O.C. do modelo com SVM e conjunto original. . . . .	46
6.3	Gráfico de rejeição e erro do modelo com RF e conjunto original. . . . .	47
6.4	Gráfico de rejeição e erro do modelo com SVM e conjunto original. . . . .	47
6.5	Curvas R.O.C. do modelo com RF e conjunto 1. . . . .	48
6.6	Curvas R.O.C. do modelo com SVM e conjunto 1. . . . .	48
6.7	Gráfico de rejeição e erro do modelo com RF e conjunto 1. . . . .	48
6.8	Gráfico de rejeição e erro do modelo com SVM e conjunto 1. . . . .	48
6.9	Curvas R.O.C. do modelo com RF e conjunto 2. . . . .	49
6.10	Curvas R.O.C. do modelo com SVM e conjunto 2. . . . .	49
6.11	Gráfico de rejeição e erro do modelo com RF e conjunto 2. . . . .	49
6.12	Gráfico de rejeição e erro do modelo com SVM e conjunto 2. . . . .	49
6.13	Curvas R.O.C. do modelo com RF e conjunto 3. . . . .	50
6.14	Curvas R.O.C. do modelo com SVM e conjunto 3. . . . .	50
6.15	Gráfico de rejeição e erro do modelo com RF e conjunto 3. . . . .	50
6.16	Gráfico de rejeição e erro do modelo com SVM e conjunto 3. . . . .	50
6.17	Gráfico de rejeição e erro do modelo combinado e conjunto 1. . . . .	54
6.18	Gráfico de rejeição e erro do modelo combinado e conjunto 2. . . . .	54
6.19	Gráfico de rejeição e erro do modelo combinado e conjunto 3. . . . .	54
6.20	Curvas R.O.C. do modelo utilizando base de imagens originais. . . . .	57
6.21	Curvas R.O.C. do modelo utilizando base de imagens Aumentadas. . . . .	57
6.22	Gráfico de rejeição e erro do modelo utilizando a metodologia 2 com as imagens originais. . . . .	57
7.1	Exemplos de tiras indefinidas classificadas como negativas. . . . .	59
7.2	Gráfico de rejeição e erro do modelo com SVM e conjunto 3. . . . .	59
7.3	Gráfico de rejeição e erro do modelo com classificadores combinados e conjunto 3. . . . .	59

## LISTA DE TABELAS

4.1	Resultado do teste conforme combinação das linhas. . . . .	32
4.2	Tabela que mostra a quantidade de imagens de cada base, e sua divisão por tipo. . . . .	34
5.1	Quantidade de características por tipo. . . . .	43
6.1	Resultados da acurácia de cada modelo da metodologia 1 com respectivos valores de desvio padrão.	45
6.2	Matriz de confusão do modelo com RF e conjunto original. . . . .	46
6.3	Matriz de confusão do modelo com SVM e conjunto original. . . . .	46
6.4	Métricas de desempenho do modelo com RF e conjunto original.. . . .	46
6.5	Métricas de desempenho do modelo com SVM e conjunto original.. . . .	46
6.6	Matriz de confusão do modelo com RF e conjunto 1. . . . .	47
6.7	Matriz de confusão do modelo com SVM e conjunto 1. . . . .	47
6.8	Métricas de desempenho do modelo com RF e conjunto 1.. . . . .	47
6.9	Métricas de desempenho do modelo com SVM e conjunto 1.. . . . .	47
6.10	Matriz de confusão do modelo com RF e conjunto 2. . . . .	48
6.11	Matriz de confusão do modelo com SVM e conjunto 2. . . . .	48
6.12	Métricas de desempenho do modelo com RF e conjunto 2.. . . . .	49
6.13	Métricas de desempenho do modelo com SVM e conjunto 2.. . . . .	49
6.14	Matriz de confusão do modelo com RF e conjunto 3. . . . .	50
6.15	Matriz de confusão do modelo com SVM e conjunto 3. . . . .	50
6.16	Métricas de desempenho do modelo com RF e conjunto 3.. . . . .	50
6.17	Métricas de desempenho do modelo com SVM e conjunto 3.. . . . .	50
6.18	Acurácias do modelo utilizando as duas bases de dados aumentados. Ambas as bases são compostas por dados originais e aumentados. . . . .	51
6.19	Matriz de confusão do modelo com SVM com base aumentada 1. . . . .	51
6.20	Matriz de confusão do modelo com SVM com base aumentada 2. . . . .	51
6.21	Acurácias do modelo usando bases formadas com imagens de cada leitora. . . . .	51
6.22	Matriz de confusão do modelo com imagens da leitora preta.. . . . .	51
6.23	Métricas de desempenho do modelo com imagens da leitora preta. . . . .	52
6.24	Matriz de confusão do modelo com imagens da leitora cinza. . . . .	52
6.25	Métricas de desempenho do modelo com imagens da leitora cinza. . . . .	52
6.26	Matriz de confusão do modelo com imagens da leitora verde. . . . .	52
6.27	Métricas de desempenho do modelo com SVM e conjunto 3 da metodologia 2.. . . . .	52
6.28	Acurácias dos conjuntos de características usando uma combinação de classificadores por meio de voto majoritário da média de probabilidades.. . . . .	53
6.29	Matriz de confusão do primeiro nível do modelo em cascata, com o algoritmo SVM. . . . .	55
6.30	Métricas de desempenho do primeiro nível do modelo em cascata. . . . .	55
6.31	Matriz de confusão do segundo nível do modelo em cascata, utilizando CNN. . . . .	55
6.32	Métricas de desempenho do segundo nível do modelo em cascata. . . . .	55
6.33	Taxas de precisão e sensibilidade finais do modelo em cascata, cuja acurácia final ficou em 95,3 pontos percentuais. . . . .	55

6.34	Resumo dos resultados da metodologia 2 com respectivos valores de desvio padrão. . . . .	56
6.35	Matriz de confusão do modelo utilizando a base de imagens originais. . . . .	56
6.36	Matriz de confusão do modelo utilizando a base de imagens aumentadas. . . . .	56
6.37	Métricas de desempenho do modelo utilizando a base de imagens originais. . . . .	56
6.38	Métricas de desempenho do modelo utilizando a base de imagens aumentadas.. . . .	56
7.1	Comparativo entre o conjunto de características originais e as implementadas para simular as mesmas.. . . . .	58
7.2	Comparativo de acurácia dos algoritmos de classificação entre os modelos. . . . .	58

## LISTA DE ACRÔNIMOS

AHE	<i>Adaptive Histogram Equalization</i>
ALS	<i>Asymmetric Least Squares</i>
AUC	<i>Area Under the Curve</i>
CCD	<i>Charge-Coupled Device</i>
CER	Curvas de Erro X Rejeição
CLAHE	<i>Contrast Limited Adaptive Histogram Equalization</i>
CLBP	<i>Compound Local binary patterns</i>
CNN	<i>Convolutional Neural Network</i>
DINF	Departamento de Informática
FN	<i>False Negative</i>
FP	<i>False Positive</i>
GLCM	<i>Gray-Level Co-Occurrence Matrix</i>
HSV	Padrão de formato de cores ( <i>Hue, Saturation, Value</i> )
IBMP	Instituto de Biologia Molecular do Paraná
LAB	Padrão de formato de cores, também conhecido por CIEL*a*b*
LBP	<i>Local binary patterns</i>
LFIA	<i>Lateral Flow Immunoassays</i>
PPM	Pontos por milhão
PPGINF	Programa de Pós-Graduação em Informática
POC	<i>Point of Care</i>
PPV	<i>Positive Predictive Value</i>
RF	<i>Random Forest</i>
RGB	Padrão de formato de cores ( <i>Red, Green, Blue</i> )
ROC	<i>Receiver Operating Characteristic</i>
SVM	<i>Support Vector Machine</i>
TN	<i>True Negative</i>
TP	<i>True Positive</i>
UFPR	Universidade Federal do Paraná
YCrCb	Padrão de formato de cores
YUV	Padrão de formato de cores, corresponde ao YCrCb

## LISTA DE SÍMBOLOS

- $\alpha$  alfa, primeira letra do alfabeto grego
- $\beta$  beta, segunda letra do alfabeto grego
- $\gamma$  gama, terceira letra do alfabeto grego

# SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>15</b>
1.1	Desafios	15
1.2	Motivação	16
1.3	Objetivo	16
1.4	Organização do documento	16
<b>2</b>	<b>Fundamentação</b>	<b>17</b>
2.1	Imunoensaios de fluxo lateral	17
2.1.1	Estrutura do teste	18
2.1.2	Moléculas Reporter	18
2.1.3	Formatos de Imunoensaios	19
2.2	Reconhecimento de padrões	20
2.2.1	Pré-processamento	20
2.2.2	Segmentação	21
2.2.3	Extração de características	21
2.2.4	Classificação	25
2.3	Rede Neural Convolutacional (CNN)	26
2.4	Métricas de avaliação dos classificadores	27
<b>3</b>	<b>Estado da arte</b>	<b>29</b>
3.1	Dispositivos de aquisição	29
3.2	Metodologias de análise de imagens	30
<b>4</b>	<b>Base de Dados</b>	<b>32</b>
4.1	Aumento de dados	33
<b>5</b>	<b>Metodologia</b>	<b>35</b>
5.1	Etapa de Pré-processamento	37
5.2	Etapa de Segmentação	37
5.3	Extração de características	40
5.3.1	Conjunto de características (originais) extraídas pela leitora	40
5.3.2	Conjunto 1: Características de cor	41
5.3.3	Conjunto 2: Características de cor e textura	42
5.3.4	Conjunto 3: <i>Transfer Learning CNN</i>	43
5.4	Classificação	44
<b>6</b>	<b>Resultados Experimentais</b>	<b>45</b>
6.1	Resultados da metodologia 1	45
6.1.1	Conjunto de características extraídos pelas leitoras	46
6.1.2	Conjunto de características 1	47
6.1.3	Conjunto de características 2	48
6.1.4	Conjunto de características 3	49

6.2	Base com dados aumentados . . . . .	50
6.3	Base separada por leitora. . . . .	51
6.3.1	Leitora preta . . . . .	51
6.3.2	Leitora cinza . . . . .	52
6.3.3	Leitora verde. . . . .	52
6.4	Combinação de classificadores . . . . .	52
6.5	Combinação de classificadores em dois níveis . . . . .	54
6.6	Resultados da metodologia 2. . . . .	56
<b>7</b>	<b>Conclusão.</b> . . . . .	<b>58</b>
	<b>REFERÊNCIAS.</b> . . . . .	<b>60</b>

# 1 Introdução

Os testes imunológicos rápidos se tornaram populares em meados dos anos 80 por apresentarem uma solução de baixo custo e resultado rápido. Logo foram utilizados em diversas áreas como a medicina veterinária, para detectar doenças em animais, no setor alimentício, para detecção de agentes contaminantes em alimentos, e principalmente na área médica, na detecção de doenças por meio da urina, sangue ou outro fluido corporal. Sua aplicação mais conhecida são os testes para confirmação de gravidez. Seu funcionamento é simples e consiste na ativação de reagentes fixados na membrana do teste, que indicam a presença ou ausência do analito em questão, geralmente por meio de mudança de cor. Em testes do tipo sanduíche, a intensidade da coloração é diretamente proporcional a presença do analito na amostra.

Os testes que reportam seu resultado de forma qualitativa, são de fácil interpretação, com uma simples inspeção visual é possível determinar seu resultado. Porém novas tecnologias de agentes marcadores como o ouro coloidal, *quantum-dots* e marcadores paramagnéticos [Koczula e Gallotta, 2016], permitem que os testes sejam desenvolvidos para reportar seu resultado de forma quantitativa. Isso possibilita um aumento na sensibilidade e na precisão dos testes, mas também os tornam cada vez mais dependentes de sistemas automatizados de leitura e interpretação dos resultados [O'Farrell, 2009].

Os sistemas de leitura mais comuns fazem o escaneamento da fita com ajuda de um emissor de luz. Depois que seu sinal é digitalizado, a análise do resultado é feita de acordo com os picos de intensidade causados pela absorção da luz na fita. No entanto, esses sistemas são suscetíveis a erros de deslocamento e de posicionamento da tira no momento da aquisição. Outros tipos de sistemas usam câmeras de *smartphones* para adquirir a imagem da tira, que é processada no próprio aparelho ou enviada para processamento remoto.

A correta interpretação do resultado dos testes rápidos é uma premissa fundamental na sua aplicação mas também representa um desafio. Resultados quantitativos tornam a leitura do resultado subjetiva, uma vez que são baseados na intensidade da coloração das linhas e, assim, propensa ao erro humano. Recentemente, sistemas baseados em reconhecimento de padrões estão sendo usados para classificar as fitas de testes rápidos utilizando algoritmos de aprendizado de máquina. Tais sistemas são treinados para categorizar uma imagem de acordo com suas propriedades, por meio de uma base de treinamento (exemplos de imagens rotuladas). Padrões de textura, forma e coloração na imagem são alguns exemplos de padrões que são utilizados para medir similaridades entre as imagens e que permitem classificá-las corretamente. Além disso, o uso de redes neurais permite extrair as características mais relevantes das imagens sem a necessidade de programá-las manualmente, pois são identificadas durante o processo de aprendizagem [Erickson et al., 2017].

## 1.1 Desafios

Os diferentes dispositivos utilizados na aquisição da base fazem com que as imagens tenham variação de iluminação, rotação ou deslocamento da fita na leitora. De maneira geral, tais problemas comprometem o sistema de análise de imagens, pois os *softwares* são calibrados para operar dentro de um intervalo de valores predeterminado. Um sistema capaz de classificar, com alta taxa de acurácia, as imagens de diferentes dispositivos e de maneira eficiente em termos de tempo, é um desafio a ser vencido.

E finalmente, o treinamento de uma rede neural convolucional (CNN) requer uma grande base de imagens. Para bases pequenas, técnicas de aumento de dados são utilizadas para melhorar a eficácia dos classificadores. O problema com bases pequenas é que os modelos treinados com ela não generalizam bem dados oriundos das bases de validação e de teste, assim, esses modelos sofrem com problemas de *overfitting*. Nesses casos, técnicas de *Transfer Learning* podem ser exploradas [Wang e Perez, 2017].

## 1.2 Motivação

O desenvolvimento de testes imunológicos rápidos com sistemas de leitores integrados é uma característica importante de muitos sistemas de análise de POC (*Point of Care*) de próxima geração e representam um desafio técnico, particularmente para plataformas de diagnóstico destinadas ao uso em campo. As inúmeras tecnologias de leitores existentes são ideais para aplicação em ambientes do primeiro mundo, sendo unidades robustas, de baixo custo de calibração e alto desempenho, com serviço ao cliente e suporte disponíveis. No entanto, para o contexto mundial, esses leitores geralmente são muito complexos e caros, e a infraestrutura de suporte para calibração, manutenção e operação não é acessível.

Neste contexto, é fundamental para as próximas gerações de sistema de leitores que a análise se torne mais flexível e adaptável a variações na forma de aquisição. Em sistemas baseados em *smartphones*, por exemplo, a variação nas condições de captura da imagem deve ser minimizada pelo *software*, que deve identificar e processar tais variações, sem depender somente de soluções físicas.

## 1.3 Objetivo

O principal objetivo deste trabalho é propor um método eficiente de leitura de bandas reagentes em testes imunológicos rápidos baseado em modelos de reconhecimento de padrões. Para isso, podemos definir alguns objetivos secundários como:

1. Experimentar o estado da arte em algoritmos de aprendizagem de máquina como Random Forest [Breiman, 2001] e o SVM (Support Vector Machine) [Hearst et al., 1998].
2. Definir quais características são importantes para se ter uma boa classificação explorando diversas representações como textura, cor e formato.
3. Também será explorado o uso de CNNs, utilizando técnicas de *transfer learning*.
4. Comparar os resultados obtidos com as técnicas tradicionais de análise de testes rápidos.

## 1.4 Organização do documento

O capítulo 2 apresenta a fundamentação teórica sobre o contexto dos testes rápidos, como funcionam e do que são compostos. Explica quais características são importantes para análise do resultado e quais fatores interferem nele. O capítulo 3 busca compilar o estado da arte em testes rápidos, que métodos e novos materiais estão sendo utilizados e como estão resolvendo o problema de classificação. Também mostra propostas de novos sistemas de leitura e reconhecimento automático destes testes entre outras pesquisas relacionadas ao tema. O capítulo 4 mostra como foi adquirida a base de dados usada nos experimentos preliminares. O capítulo 5 detalha a metodologia usada para o modelo e as propostas de trabalho. e finalmente o capítulo 6 trás os resultados dos experimentos.

## 2 Fundamentação

Este capítulo mostra os principais conceitos abordados neste trabalho. Primeiro, um resumo sobre os testes de diagnóstico rápido, sua composição, aplicação e os principais problemas envolvendo sua classificação. Em seguida, são mostrados quais os principais métodos de aquisição de imagem e como funcionam. E finalmente, a conceituação sobre a metodologia que será usada, desde a extração de características até as métricas de avaliação dos modelos.

### 2.1 Imunoensaios de fluxo lateral

Os testes de diagnóstico rápido, como são popularmente conhecidos, ou *Lateral Flow Immunoassays* (LFIA), rapidamente se consolidaram na medicina, por sua praticidade e rapidez em testar várias patologias, sem a necessidade do uso de complexos laboratoriais. Também, por ser portátil (geralmente é um dispositivo pequeno, que cabe em um bolso), permite que seja transportado para regiões geograficamente remotas e de difícil acesso, onde há falta de infraestrutura médica básica.

Além de sua óbvia utilização na detecção de doenças, os testes são usados também para controle de qualidade, medindo a presença de patógenos [Nielsen et al., 2009], na medicina veterinária, para detecção de doenças animais [Mei et al., 2013], segurança na produção de alimentos, medindo a presença de toxinas [Ching et al., 2015], produtos químicos e poluentes na água [Kuang et al., 2013] e [Ngom et al., 2010].

Sua estrutura é baseada em tecnologias para detectar e/ou quantificar a presença ou ausência de determinada molécula - o analito - em uma solução - a amostra - e funcionam com base no movimento do líquido através de uma membrana de material polimérico, contendo reagentes desidratados, que são ativados, ou não, pela passagem da amostra líquida através da tira.

Para este trabalho, está sendo considerado o método de detecção baseado em anticorpo-antígeno, no qual o analito é formado exclusivamente por anticorpos que são proteínas geradas pelo corpo em resposta a invasão de alguma molécula (antígeno) no corpo. Portanto, o teste se baseia na propriedade dos anticorpos de se ligar a uma região específica dos antígenos, chamada epítopo, conforme figura 2.1.

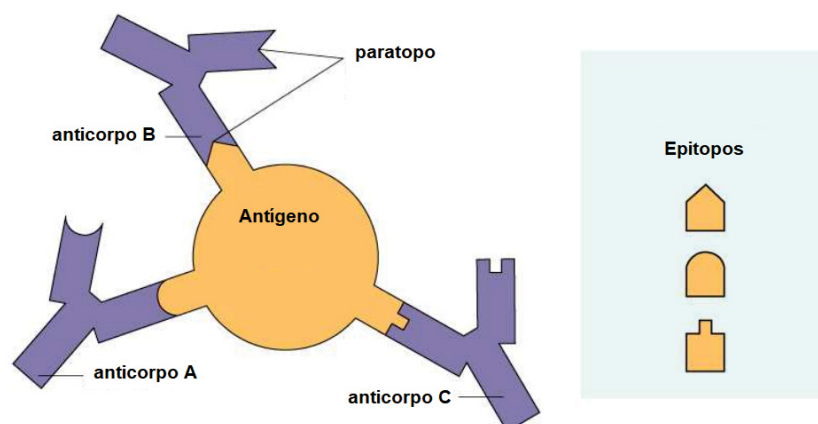


Figura 2.1: Antígenos com epítopos se ligam ao paratopo dos anticorpos. Fonte: [Copstead e Banasik, 2000].

Seu resultado é reportado de forma quantitativa. A análise qualitativa dos testes pode ser feita por meio de uma inspeção visual na coloração das linhas da tira, constatando a presença ou ausência do analito. Esse tipo de análise tem a vantagem da rapidez em se obter a resposta, pois, normalmente seus resultados são relatados de forma binária como positivos/negativos. Testes que exigem uma análise quantitativa são sensíveis a quantidade e a

concentração da amostra e, embora sejam testes mais precisos, são mais suscetíveis a gerar resultados falsos positivos. Também necessitam do auxílio de leitoras e de *softwares* específicos para analisar a intensidade da coloração na tira.

### 2.1.1 Estrutura do teste

A estrutura do teste de diagnóstico rápido é formada por camadas sobrepostas, em que a amostra flui, interagindo com as moléculas fixadas em cada região. Essa estrutura e suas camadas são detalhadas na figura 2.2, que mostra também as diversas regiões que constituem o teste.

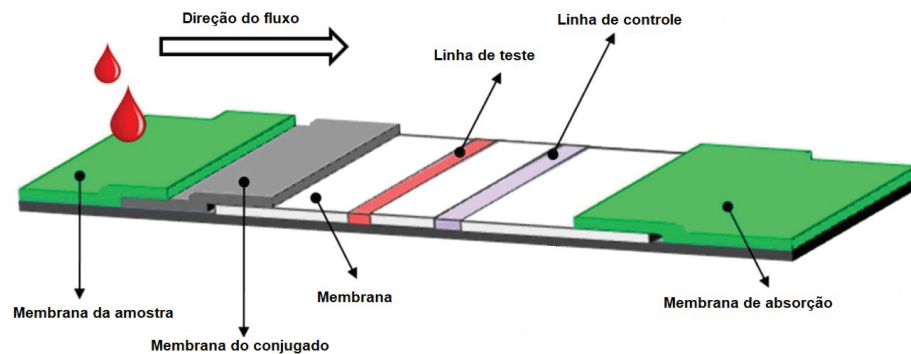


Figura 2.2: Estrutura do teste. A solução segue pela membrana da amostra, passando pelas linhas de teste e de controle até ser absorvida pela membrana de absorção.

- **Membrana de amostra**

A membrana de amostra geralmente é feita de fibras de celulose ou de sílica treliçada [Posthuma-Trumpie et al., 2009] e é onde a amostra é inserida. Sua composição geralmente é feita para filtrar elementos interferentes, como os glóbulos vermelhos do sangue. Suas principais funções são permitir uma absorção uniforme da amostra e direcioná-la para a membrana do conjugado. Uma variedade de tipos de amostras pode ser usada no teste, como urina, saliva, soro, plasma e sangue.

- **Membrana do conjugado**

A membrana do conjugado, assim como a membrana de amostra, é feita de celulose ou de sílica treliçada. Nessa região, está fixado o assim chamado conjugado, que é um composto formado por moléculas reporter e pelos anticorpos. As moléculas reporter são formadas por nanopartículas coloridas ou fluorescentes. Para este trabalho, trata-se de ouro coloidal, como será visto na subseção 2.1.2. Assim que a amostra chega nessa membrana, ela interage com o conjugado, e os antígenos presentes na amostra se fixam com os anticorpos marcados do conjugado. Essa solução, então, segue adiante na membrana.

- **Membrana de nitrocelulose e linhas de controle e teste**

Nessa membrana estão presentes as linhas de reação (ou bandas), que também são formadas pelo elemento proteico presente no conjugado, os anticorpos. Assim que a amostra chega às linhas, a solução marcada se fixa nos anticorpos e a tornam visível. No caso das nanopartículas de ouro, a linha se torna vermelha.

Ao menos duas linhas estão presentes na membrana, a linha de teste e a de controle. A linha de teste reporta o resultado do teste, e a linha de controle confirma que o fluxo está fluindo normalmente através da membrana em direção à membrana de absorção. A posição da linha de teste é um importante fator na sensibilidade do resultado. Idealmente ela deve estar o mais distante possível da membrana de amostra, pois a amostra terá mais tempo de reação. A concentração das moléculas reporter é outro fator importante.

- **Membrana de absorção**

O fluxo contínuo da substância através da membrana é garantido pela capilaridade do material da membrana de nitrocelulose, mas também pela membrana de absorção, que puxa a substância e, assim, garante o fluxo contínuo do material. Ela também retém o excedente da solução no final do teste.

### 2.1.2 Moléculas Reporter

As moléculas reporter (ou marcadores) são os elementos que tornam possível a visualização do resultado do teste. Elas são formadas por nanopartículas coloridas ou fluorescentes, com tamanhos que variam entre 15-800

nm. Esses elementos são usualmente feitos de ouro coloidal, látex, selênio, carbono, lipossomos, e os mais recentes, *quantum dots* e conversão de fósforo, sendo ouro coloidal e látex os mais usados; aproximadamente 94% das tiras de testes são feitas usando nanopartículas de ouro, segundo [Chun, 2009] e [Karakus e Salih, 2013].

A escolha do marcador a ser utilizado na fabricação de um teste depende de uma série de fatores, como o tipo do teste e qual o grau de sensibilidade necessário. Se o teste requer uma tecnologia de leitor, por exemplo, as opções são o ouro coloidal, látex colorido ou fluorescente e partículas de látex paramagnéticas. Para testes nos quais é possível ler o resultado a olho nu, sem a necessidade de leitor, o ouro coloidal é capaz de produzir maior sensibilidade coloral em comparação ao látex, devido ao menor tamanho de suas nanopartículas.

O ouro coloidal é estável sob exposição ao calor e à luz, e a degradação é limitada, principalmente pela estabilidade da proteína. Possui algumas desvantagens, como uma faixa dinâmica quantitativa muito limitada e um limite de detecção baixo, fator que pode inviabilizar o teste, mesmo com sistemas de leitura. É relativamente fácil e barato de se produzir em laboratório, sua cor é intensa e depende do tamanho da molécula; a cor vermelha se apresenta em partículas menores que 100 nm. O tamanho da partícula também é importante para a estabilidade e sensibilidade na formação do conjugado. Em [Aveyard et al., 2007], o tamanho ótimo foi encontrado com 80 nm.

Basicamente todos os métodos para produzi-lo usam um agente redutor para converter ouro iônico em ouro metálico de maneira controlada. Faraday, em 1857, descobriu a sensibilidade do ouro coloidal com eletrólitos, e, em 1971, Faulk e Taylor produziram ouro coloidal usando redução de fósforo branco para microscopia. Porém, o primeiro uso em testes imunológicos como conjugado com anticorpos foi em 1981, por Leuvering [Leuvering et al., 1981]. Danscher em [Danscher, 1984] e Holgate [Holgate et al., 1983] descrevem um método de aprimoramento do ouro coloidal, resultando na criação de camadas de prata, envolvendo as partículas de ouro, permitindo melhoria da visualização do coloidal com ordem de magnitude duas a três vezes maior.

Além da conjugação de nanopartículas com anticorpos, o ouro pode ser conjugado com antígenos [Karakus e Salih, 2013] e peptídeos antimicrobianos, porém, a maioria dos estudos utilizam o imunoconjugado de ouro coloidal com anticorpos.

### 2.1.3 Formatos de Imunoensaios

Os imunoensaios podem ser executados em vários formatos diferentes sendo os mais comuns o teste sanduíche, teste competitivo e o teste multiplex.

- **Teste sanduíche**

Nesse formato, o anticorpo é fixado na linha de teste, e uma mistura de analito de amostra e anticorpo marcado é aplicada na membrana do conjugado. Após a aplicação da amostra na tira, o analito migra para a membrana do conjugado, onde o analito alvo é capturado pelo anticorpo marcado, o que resulta na formação de uma solução conjugada marcada. Na linha de teste, o analito dessa solução é capturado pelos anticorpos primários, que estão fixados na linha. O excesso de solução migra pela membrana de nitrocelulose até a linha de controle, onde é capturada pelos anticorpos secundários fixados nessa linha. Para esse formato, a intensidade do sinal é diretamente proporcional à presença de analito na amostra, e a ausência de coloração na linha de teste indica ausência do analito procurado. A figura 2.3 ilustra o funcionamento do teste.

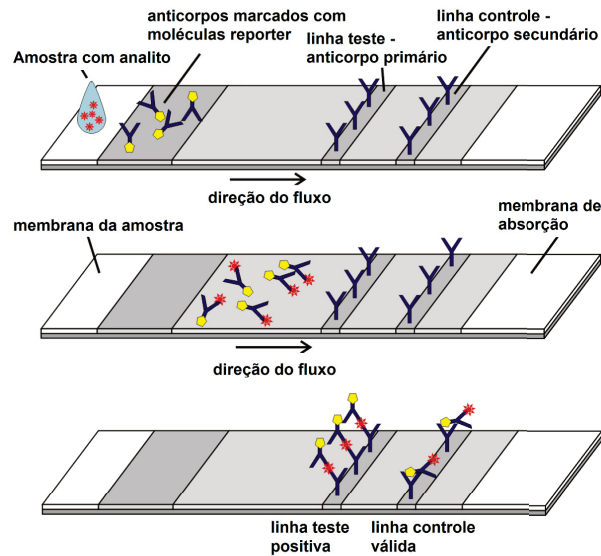


Figura 2.3: Funcionamento do teste sanduíche usando ouro coloidal como molécula reporter. Neste formato a intensidade do sinal é diretamente proporcional a presença de analito na amostra.

- **Teste de inibição competitivo**

Nesse formato, um conjugado analito-proteína é fixado na linha de teste, e uma mistura de anticorpo marcado e analito de amostra é aplicada na membrana do conjugado. Quanto maior a presença de analito, menor é o sinal gerado. A ausência de analito na amostra gera o sinal máximo, enquanto a presença de sinal em ambas as linhas indica um resultado negativo.

- **Teste multiplex**

Esse formato é usado para a detecção de mais de um tipo de analito. O número de linhas de teste é igual ao número de moléculas alvo a serem analisadas. É aplicado quando se deseja analisar múltiplos analitos simultaneamente, sob o mesmo conjunto de condições [Panhotra et al., 2005].

## 2.2 Reconhecimento de padrões

O estudo do reconhecimento de padrões está concentrado na descoberta automática de regularidades em dados, por meio do uso de algoritmos de computador e em como usar essas regularidades para realizar ações como classificar os dados em diferentes categorias [Bishop, 2006]. Um sistema tradicional para classificação de padrões envolve as etapas de pré-processamento, segmentação, extração de características e classificação, que são ilustradas no fluxograma abaixo.

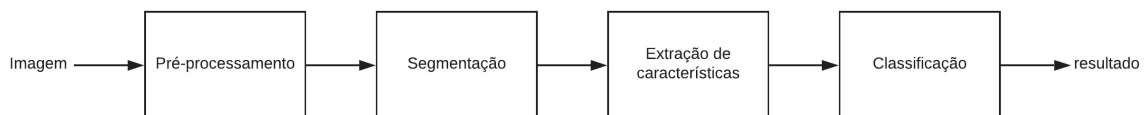


Figura 2.4: Etapas do processo de classificação de imagens.

Primeiro a imagem é pré-processada para facilitar as operações subsequentes. Depois, a operação de segmentação isola a porção de interesse da imagem. A área segmentada é submetida ao processo de extração de características, que extrai da imagem as propriedades mais relevantes para sua representação. Essas características, então, são enviadas ao classificador, que faz a análise e toma a decisão final.

### 2.2.1 Pré-processamento

Esta é uma etapa do processo que objetiva preparar a imagem para otimizar o resultado das próximas etapas. A otimização envolve remover ou reduzir ruídos estampados na aquisição da imagem. Sombras causadas

pelo equipamento de aquisição, falhas na imagem ou até a variação de luz ambiente que incide sobre o objeto são alguns exemplos de ruídos. Tudo isso causa distorções nas etapas de segmentação e extração, culminando em uma redução de eficiência do sistema como um todo. As soluções para reduzir os efeitos dos ruídos nas imagens são muitas e geralmente passam pela aplicação de filtros que suavizam imperfeições como o filtro Gaussiano ou filtro da média. Técnicas para salientar detalhes da imagem como bordas, cores ou contraste também são muito utilizadas nesta etapa, como a equalização adaptativa de histograma (AHE, do inglês, *Adaptive Histogram Equalization*), que é uma técnica usada para melhorar o contraste em imagens. Porém, ela tende a sobre-amplificar o ruído por toda a imagem quando o contraste é muito concentrado em apenas algumas regiões. O CLAHE (*Contrast Limited Adaptive Histogram Equalization*) é uma variante da equalização adaptativa de histograma em que a amplificação do contraste é limitada e localizada, de modo que reduzir o problema de amplificação de ruído.

## 2.2.2 Segmentação

A segmentação de uma imagem consiste em extrair dela objetos de interesse, chamados segmentos, de maneira que todos os *pixels* que compõe um segmento sigam um padrão de similaridade.

As abordagens para segmentação de imagens podem ser categorizadas de acordo com suas propriedades. A abordagem baseada na descontinuidade busca mudanças abruptas nos níveis de intensidade dos *pixels* da imagem, como pontos isolados, linhas ou bordas. Nesta abordagem, a detecção de bordas é a técnica mais utilizada. Para isso são aplicados filtros detectores de bordas que utilizam operadores como os de *Sobel*, *Robert*, *Prewitt*, *Laplacian* e *Canny* [Maini e Aggarwal, 2009], onde as bordas representam descontinuidades na imagem. Outro tipo de abordagem se baseia na similaridade. As técnicas de limiarização (do original em inglês *thresholding*) e de *clustering* são as mais utilizadas [Jain, 1989]. A limiarização é utilizada para reduzir a quantização dos *pixels* pois, a partir de uma imagem em escala de cinza, o limiar pode ser usado para criar imagens binárias. A maneira mais comum de converter uma imagem de nível de cinza em uma imagem binária é selecionar um único valor de limiar ( $T$ ) e aplicar a seguinte regra em cada *pixel*  $f(x,y)$  da figura: Se  $f(x,y) > T$  então  $f(x,y) = 0$  senão  $f(x,y) = 255$ . A vantagem de obter uma imagem binária é que ela reduz a complexidade dos dados. O método de limiarização de *Otsu* [Otsu, 1979], por sua vez, visa encontrar o valor ideal para o limiar global. Este método é utilizado para superar o inconveniente do limiar iterativo, isto é, calcular a média após cada passo. A figura 2.5 mostra a aplicação do método de Otsu.

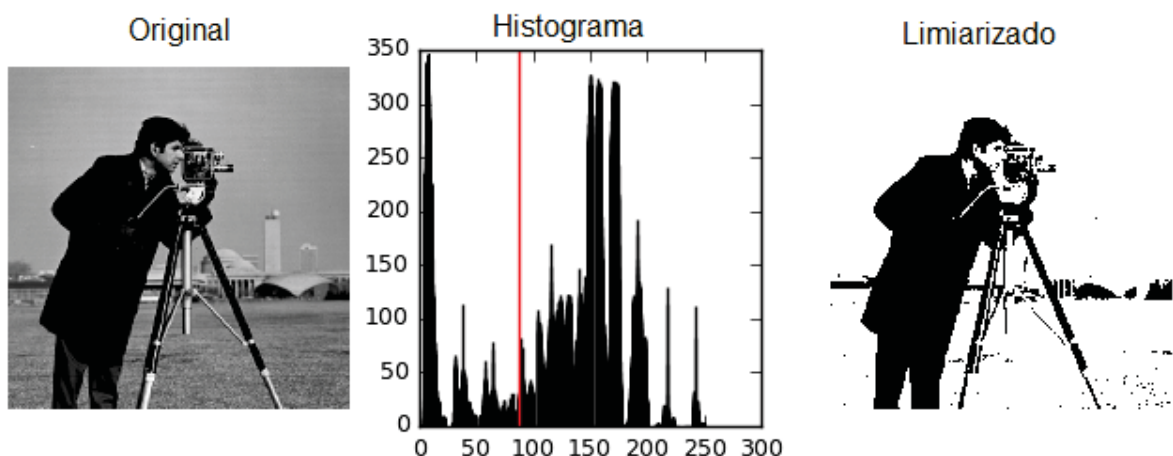


Figura 2.5: [scikit image, 2018]. Aplicação da limiarização de *Otsu* em uma imagem.

## 2.2.3 Extração de características

A redução da quantidade de dados necessários para representar uma imagem de maneira eficiente e não redundante é denominado extração de características. Uma vez que as características foram extraídas e estão representadas de forma conveniente (geralmente como um vetor multidimensional), elas podem ser usadas para medir similaridades entre imagens, representar imagens em buscas, mapeamentos e navegação em bases de dados, entre outros.

A cor é uma característica predominante usada na análise dos resultados dos testes de diagnóstico rápido. Possui propriedades interessantes como invariância em relação à escala, translação e rotação da imagem. O desempenho destes descritores depende da distribuição de cores das imagens. O histograma de cores tem um desempenho destacado em comparação com outros descritores quando as imagens apresentam uma distribuição de

cores bastante uniforme. Ele representa a distribuição do número de *pixels* da imagem e o número de elementos em um histograma depende do número de *bits* em cada *pixel*. Para uma imagem com 3 canais de cor RGB de 8 bits, com a faixa de valores de intensidade variando de 0 a 255 é formado um vetor com 512 valores. Os momentos podem ser calculados com base no histograma de cores da seguinte forma:

$$m\u00e9dia = \frac{\sum_{i=0}^{255} i * h(i)}{\sum_{i=0}^{255} h(i)} \quad (2.1)$$

sendo  $h(i)$  a frequ\u00eancia de ocorr\u00eancia do valor  $i$ .

Na maioria das categorias de imagens, os momentos de cor tamb\u00e9m mostram bom desempenho. Por\u00e9m, a combina\u00e7\u00e3o de descritores de cores produz melhor taxa de recupera\u00e7\u00e3o em compara\u00e7\u00e3o com descritores de cores individuais [Kodituwakku e Selvarajah, 2004]. Normalmente, os tr\u00eas momentos de mais baixa ordem s\u00e3o usados como caracter\u00edsticas de imagens, pois a maioria das informa\u00e7\u00f5es de distribui\u00e7\u00e3o de cores est\u00e1 contida neles. Tr\u00eas momentos de cor s\u00e3o calculadas por canal. A **m\u00e9dia**, **vari\u00e2ncia** e **desvio padr\u00e3o** de uma imagem s\u00e3o conhecidos como momentos de cor. As equa\u00e7\u00f5es seguintes definem a m\u00e9dia, a vari\u00e2ncia e o desvio padr\u00e3o de uma imagem de tamanho  $N \times M$ .

$$m\u00e9dia = \frac{\sum_{i=1}^N \sum_{j=1}^M x_{ij}}{MN} \quad (2.2)$$

$$vari\u00e2ncia = \frac{1}{MN} \sum_{i=1}^N \sum_{j=1}^M (x_{ij} - m\u00e9dia)^2 \quad (2.3)$$

$$desvio\ padr\u00e3o = \sqrt{vari\u00e2ncia} \quad (2.4)$$

Sendo que  $x_{ij}$  \u00e9 o valor do *pixel* na posi\u00e7\u00e3o  $(i, j)$  da imagem. Assim, o tamanho do vetor de caracter\u00edstica \u00e9 9, sendo 3 valores para cada canal da imagem.

Assim como a cor, caracter\u00edsticas relacionadas a intensidade luminositade da imagem s\u00e3o muito importantes. Em alguns casos, \u00e9 poss\u00edvel determinar o resultado de um teste por meio da an\u00e1lise da curva de intensidade das regi\u00e3o de pico. A curva de intensidade \u00e9 gerada pela proje\u00e7\u00e3o lateral de histograma da imagem. A figura 2.6(b) \u00e9 um exemplo da proje\u00e7\u00e3o de histograma de uma tira de teste. O m\u00e9todo ALS (*Asymmetric Least Squares*) [Eilers e Boelens, 2005] \u00e9 aplicado para suaviza\u00e7\u00e3o da curva.

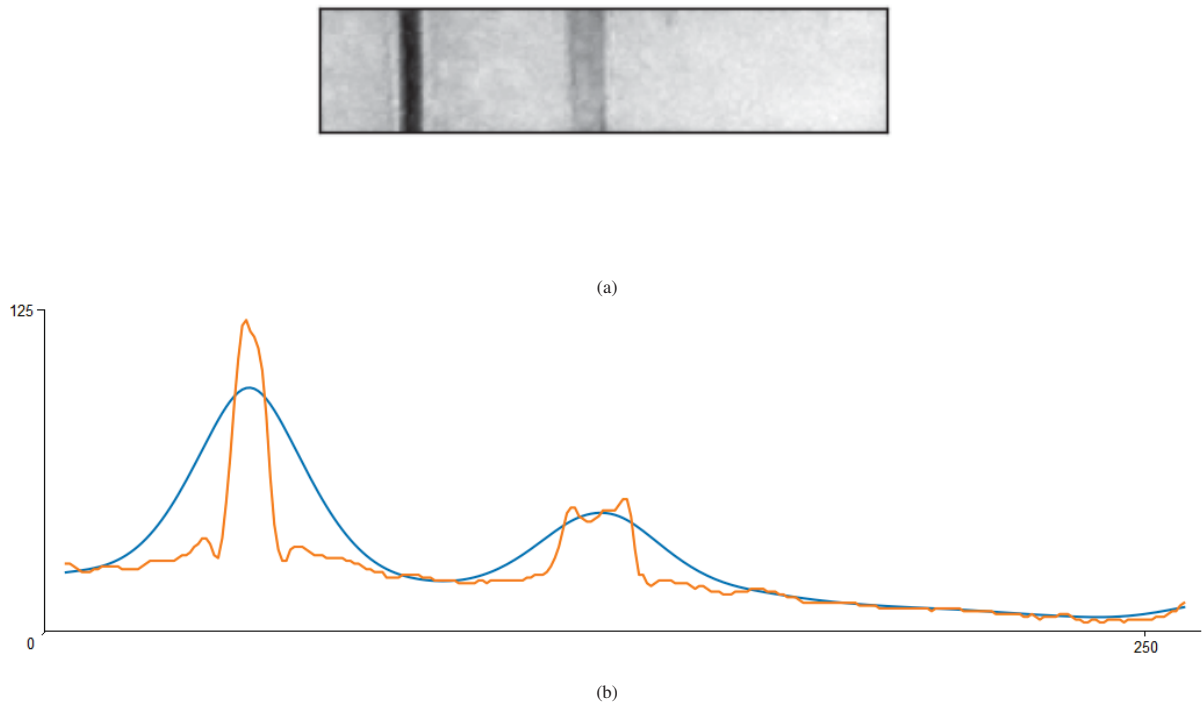


Figura 2.6: a) Canal V da imagem de uma tira teste na representação de cores HSV. b) Projeção de histograma lateral da imagem (linha laranja). A curva da projeção suavizada pelo método ALS (linha azul). Na figura estão identificados dois picos de intensidade, a linha amarela representa a largura da curva na região do pico.

As texturas de uma imagem também podem revelar padrões relevantes. Foram propostas uma ampla variedade de técnicas para descrever a textura da imagem. [Tuceryan et al., 1993] dividiu os métodos de análise de textura em quatro categorias: estatística, geométrica, baseada em modelos e processamento de sinal. Os métodos estatísticos analisam a distribuição espacial dos valores de cinza, calculando as características locais em cada ponto da imagem e derivando um conjunto de estatísticas das distribuições das características locais.

O LBP (*Local Binary Pattern*) é um eficiente descritor de textura que se baseia no fato de que certos padrões binários locais à região de vizinhança de um *pixel* são propriedades de textura da imagem. É muito utilizado em aplicações de reconhecimento facial e possui a importante propriedade de ser tolerante a variações monotônicas de iluminação. O descritor delimita a vizinhança de cada *pixel* com base no seu valor e considera o resultado como um número binário, gerando uma representação numérica das características desta imagem. A equação geral para cada *pixel*  $(x_c, y_c)$  é a seguinte:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^c \quad (2.5)$$

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2.6)$$

Sendo que a notação  $P,R$  significa  $P$  pontos de amostragem em um círculo de raio de  $R$ . A imagem 2.7 mostra um exemplo de cálculo usando este descritor.

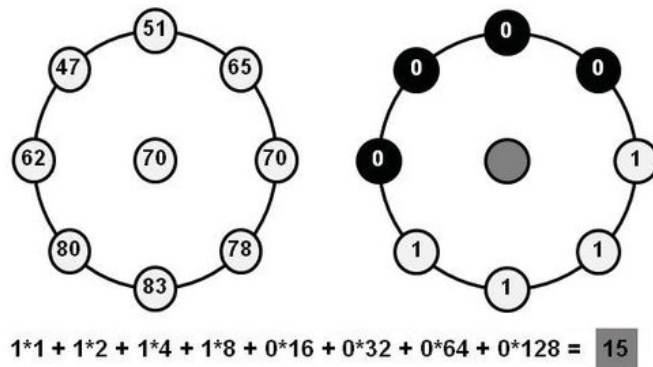


Figura 2.7: Exemplo de cálculo do histograma LBP com raio=1 e vizinhança= 8. O valor do elemento central funciona como limiar para valorar os vizinhos. É gerado um número binário de 8 bits, que então é convertido para decimal.

Existem algumas variantes deste operador de textura como o CLBP (*Compound Local Binary Pattern*) [Zhao et al., 2012], que torna o padrão invariante à rotação inserindo um *bit* extra que representa a magnitude da informação.

Outro descritor de textura, a matriz de co-ocorrência de níveis de cinza (GLCM), baseia-se na ocorrência repetida de alguns níveis de cinza na textura. Além da distância, este descritor utiliza a direção como parâmetro de cálculo. O GLCM de uma região de interesse é calculado usando o vetor de deslocamento, definindo raio e ângulos de rotação, conforme ilustrado na figura 2.8. Neste trabalho são considerados os ângulos de 0°, 45°, 90°, 135° e as propriedades de *energia*, que define a uniformidade do padrão; *contraste*, que mede a variação de intensidade local e; *homogeneidade*, que mede o quão suave é a distribuição das intensidades dos *pixels*. Suas fórmulas são descritas a seguir.

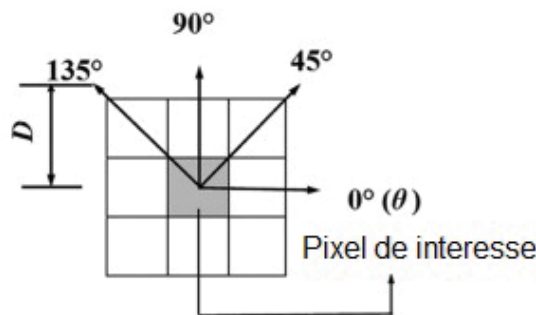


Figura 2.8: Os ângulos definem a direção dos *pixels* a serem considerados em um raio de tamanho *D*.

$$energia = \sum_{i,j=0}^{N-1} (P_{ij})^2 \tag{2.7}$$

$$contraste = \sum_{i,j=0}^{N-1} P_{ij}(i - j)^2 \tag{2.8}$$

$$homogeneidade = \sum_{i,j=0}^{N-1} \frac{P_{ij}}{1 + (i - j)^2} \tag{2.9}$$

Sendo que,  $P_{ij}$  = elemento  $(i,j)$  da matriz GLCM normalizada e  $N$  = número de níveis de cinza da matriz GLCM. A figura abaixo mostra o resultado da aplicação do GLCM em uma imagem. Para cada propriedade são gerados 12 valores, totalizando 36 características.

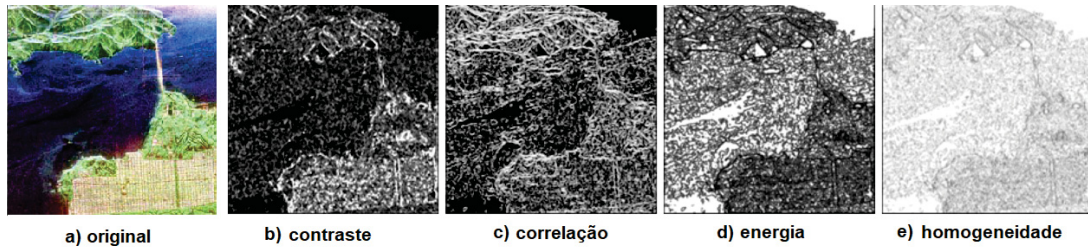


Figura 2.9: Aplicação das propriedades do GLCM. Para cada tipo de propriedade um padrão diferente é ressaltado.

## 2.2.4 Classificação

A tarefa de classificação é geralmente realizada por algoritmos de aprendizagem de máquina supervisionados. O aprendizado de máquina é um método de análise de dados que aprende padrões sem ser explicitamente programado para isso. O método deve ser capaz de se adaptar aos dados para conhecer o padrão de cada nova entrada e classificá-la corretamente. Variações de iluminação nas imagens, causadas na aquisição, podem ser "ensinadas" ao modelo para que seja corretamente reconhecido e, assim, aumentando a flexibilidade do modelo em categorizar novas instâncias. Esses algoritmos são geralmente aplicados na tarefa de categorizar os dados a partir desses padrões, uma vez que conseguem "aprender" (ou seja, melhorar progressivamente o desempenho em uma tarefa específica) com os dados. O resultado da aplicação do algoritmo de aprendizado de máquina supervisionado pode ser expresso pela função  $y(x)$  que recebe uma imagem  $x$  como entrada e gera um vetor  $y$  como saída. A função  $y(x)$  é determinada durante a fase de treinamento do algoritmo, também conhecida como aprendizado, com base nos dados de exemplos disponíveis.

### • Máquina de vetor de suporte (SVM)

A máquina de vetor de suporte (SVM)<sup>1</sup> é um algoritmo que, para classificação binária supervisionada, define um hiperplano ideal de separação dos dados de treinamento chamada de fronteira de decisão. Em um espaço bidimensional, um hiperplano é uma linha que divide o espaço dos dados. A distância entre o hiperplano e os pontos de dados mais próximos é chamada de margem. A maximização dessa margem (*MaxMargin*) gera uma fronteira ideal, que pode ser flexibilizada para que permita os, assim chamados, *outliers*, os quais são exemplos de dados que atravessam essa margem sem que a fronteira de decisão seja afetada por eles (*SoftMargin*). Em problemas multi-classe, o SVM implementa algumas estratégias como a "classificação de emparelhamento" e a estratégia do "um-contratodos". Para este último, se  $n$  é o número de classes do problema, então  $n * (n - 1) / 2$  classificadores binários são construídos e cada um deles é treinado para distinguir uma classe das demais classes do problema. [Pal, 2008].

Para problemas que não são separáveis linearmente, o SVM pode executar, com eficiência, classificação não linear por meio do uso de uma função de mapeamento chamada *kernel*, que mapeia implicitamente suas entradas em espaços dimensionais superiores. Para o *kernel* linear o ponto entre a entrada  $x$  e cada vetor de suporte  $x_i$  é calculada da seguinte forma:

$$f(x) = \sum a_i y_i (x \cdot x_i) + \beta \quad (2.10)$$

Esta é uma equação que envolve o cálculo dos produtos internos de um novo vetor de entrada  $x$  com todos os vetores de suporte nos dados de treinamento. Os coeficientes  $\beta$  e  $a_i$  são estimados a partir dos dados de treinamento pelo algoritmo de aprendizado. Além deste, outros *kernels* podem ser usados para transformar o espaço de entrada em dimensões mais altas, como o *kernel* polinomial e o *kernel* radial.

O *kernel* polinomial usa, em vez do produto de ponto, um núcleo polinomial:

$$f(x) = 1 + \sum (x * x_i)^d \quad (2.11)$$

Onde o grau  $d$  do polinômio deve ser especificado manualmente para o algoritmo de aprendizado. O  $d = 1$  equivale ao *kernel* linear. O *kernel* polinomial permite linhas curvas no espaço de entrada.

Para o *kernel* radial um valor de *gamma* deve ser especificado:

$$f(x) = \exp(-\gamma * \sum (x - x_i)^2) \quad (2.12)$$

### • Random Forest

<sup>1</sup>Do original em inglês "Support Vector Machine"

*Random forest* é um exemplo de *bagging ensemble* de árvores de decisão mas com algumas diferenças. *Ensemble* é um conjunto de classificadores cujas decisões são combinadas (através de voto) para classificar um novo exemplo, aumentando, assim, o poder de generalização e precisão em comparação a um classificador simples [Dietterich, 2000]. *Ensembles* são divididos em duas técnicas. O primeiro, *boosting*, gera classificadores de forma sequencial onde o conjunto de treinamento de cada classificador é definido de acordo com o desempenho do anterior. Por outro lado, a técnica de *bootstrap aggregation* ou *bagging* combina classificadores de forma independente agregando suas previsões para obter o resultado final, reduzindo a variância do modelo.

Em algoritmos como árvores de decisão e de regressão, ao selecionar um ponto de divisão, é permitido ao algoritmo examinar todas as variáveis e seus valores para selecionar o ponto de divisão ideal. O algoritmo de *random forest* altera esse procedimento para que ele esteja limitado a uma variável, tomada aleatoriamente na busca. Outra importante característica deste algoritmo é o fato dele medir a importância de cada característica no processo de treinamento. Com isso é possível descartar aquelas que não tem muito importância ao modelo.

O algoritmo de *random forest* evita o problema de *overfitting*, quando o modelo se torna especializado demais para os dados de treinamento. Em geral, *ensembles* tendem a diminuir a variância das previsões pela sua maior capacidade de generalização. No *random forest* isso é alcançado através do aumento do número de árvores do modelo, importante quando se tem um grande número de características.

#### • Múltiplos classificadores

Classificadores podem ser combinados para melhorar o desempenho de sistemas de reconhecimento de padrões, equilibrando suas fraquezas individuais. As técnicas de *ensemble* geram automaticamente múltiplos classificadores, que atuam em subconjuntos da base. Porém, algoritmos de aprendizado, com conhecimento em todo o espaço de características e com diferentes arquiteturas, podem ter seus resultados combinados por meio de alguma regra de fusão como, voto ou média de probabilidades [Kittler et al., 1998]. A escolha dos classificadores a serem combinados, geralmente é feita com base nas suas taxas de acurácia sobre a base de validação, embora existam *frameworks* que otimizam esta etapa do processo [Brun et al., 2018].

A regra do voto majoritário simplesmente conta os votos recebidos dos classificadores individuais. A classe que receber o maior número de votos (maioria simples) é selecionada como a decisão. Sistemas com número par de classificadores devem implementar uma forma de desempate. Já a regra da média, utiliza as probabilidades atribuídas, por cada classificador, para cada classe. Desta maneira, a classe que obtiver a maior média de probabilidades, será a classe selecionada. A equação da média é expressa pela fórmula:

$$\mu_j(x) = \frac{1}{T} \sum_{t=1}^T d_{t,j}(x) \quad (2.13)$$

A classe  $j$  que obtiver o maior valor de  $\mu_j(x)$  é a escolhida.  $T$  representa o número de classificadores e  $d_{t,j}(x)$  é a probabilidade aferida pelo classificador  $t$  para a classe  $j$ .

## 2.3 Rede Neural Convolutiva (CNN)

Uma rede neural convolutiva (CNN)<sup>2</sup> é um tipo de rede neural artificial que tem sido amplamente aplicada a uma variedade de problemas de reconhecimento de padrões, como visão computacional, classificação de imagens e para extração de características. É composta por diversas camadas de convolução e *pooling* conforme figura 2.10.

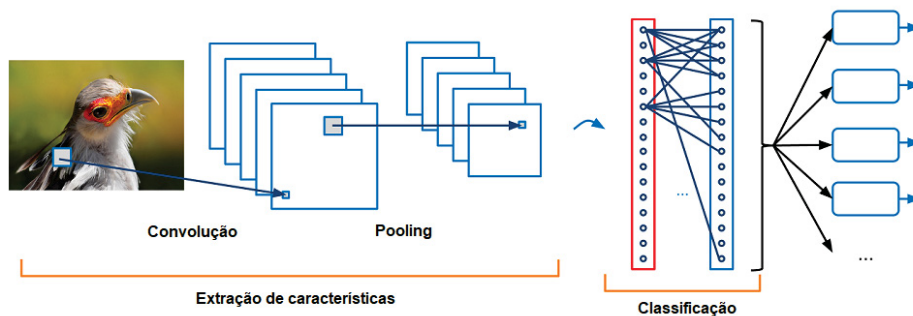


Figura 2.10: Camadas típicas de uma rede neural convolutiva. Divisão das etapas de extração de características e classificação.

<sup>2</sup>CNN - do inglês "Convolution Neural Network"

A camada convolucional implementa a operação de convolução, conceito que, em processamento de imagens, refere-se a uma operação linear aplicada em toda a imagem  $f(x, y)$  dado um filtro ou *kernel*  $w(x, y)$  conforme equação 2.14, e tem como principal objetivo a extração de características da imagem. Então, a convolução de diferentes filtros sobre uma imagem resulta na extração de diferentes tipos de características como, por exemplo, bordas e curvas. A camada de *pooling* geralmente é usada imediatamente após uma camada convolucional. Isso significa que as saídas da camada convolucional são as entradas para a camada de *pooling* da rede. A ideia dessa camada é reduzir o número de parâmetros reduzindo a dimensionalidade de cada mapa de características gerado pela camada convolucional mas mantendo as informações importantes.

$$w(x, y) \otimes f(x, y) = \sum_{i=1}^m \sum_{j=1}^n w(i, j) f(x - i, y - j) \quad (2.14)$$

sendo  $(x, y)$  uma posição na matriz da imagem de tamanho  $m \times n$ .

Na utilização da CNN para a tarefa de extração de características, o vetor resultante da última camada de *pooling* da rede é usado como representação da imagem em outro modelo. Esta é uma técnica conhecida como transferência de conhecimento (do original em inglês, *transfer learning*), ou seja, é possível aplicar o conhecimento aprendido anteriormente, em modelos pré-treinados, para resolver novos problemas mais rapidamente ou com melhores soluções. A motivação fundamental para a transferência de conhecimento no campo da aprendizagem de máquinas foi o conceito de "*learning to learn*", que se concentrou na necessidade de métodos de aprendizado de máquinas que reutilizam conhecimentos previamente aprendidos. As técnicas tradicionais de aprendizado de máquinas tentam aprender cada tarefa a partir do zero, enquanto as técnicas de transferência de conhecimento tentam transferir o conhecimento de algumas tarefas anteriores para uma tarefa nova. A CNN pode ser aplicada em sistemas de classificação de padrões conhecidos como *end-to-end*, onde as etapas de pré-processamento e segmentação não são aplicadas, conforme figura abaixo.

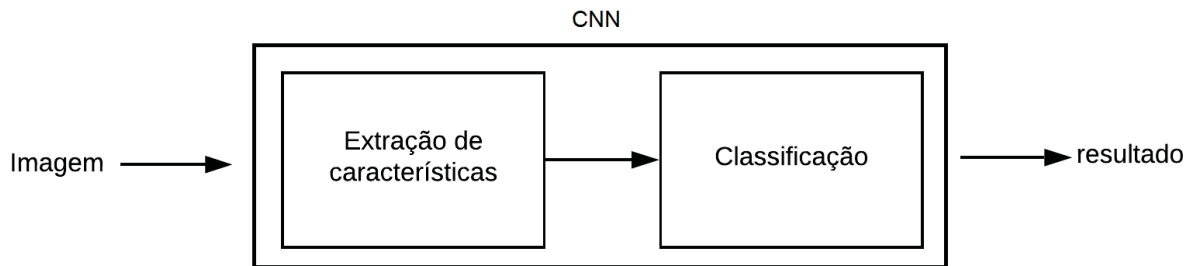


Figura 2.11: Fluxograma da solução *end-to-end*. Etapas tradicionais de pré-processamento e segmentação não estão presentes nesta metodologia.

## 2.4 Métricas de avaliação dos classificadores

A exatidão de um modelo pode ser mensurada por meio da contagem do número de observações reconhecidas corretamente (TP, do inglês *true positive*), pelo número de observações reconhecidas corretamente que não pertencem à classe (TN, do inglês *true negative*), e por observações que foram atribuídas incorretamente a classe (FP, do inglês *false positive*) ou não foram reconhecidas como exemplos dessa classe (FN, do inglês *false negative*). Esses valores formam a matriz de confusão e permitem calcular a acurácia, precisão e a sensibilidade/especificidade (*Recall*). Segundo [Sokolova e Lapalme, 2009] para problemas multi-classe as métricas são definidas pelas equações abaixo sendo, para cada classe  $C_i$ ,  $TP_i$  é TP para  $C_i$ ,  $TN_i$  é TN para  $C_i$ ,  $FP_i$  é FP para  $C_i$ ,  $FN_i$  é FN para  $C_i$  e  $l$  representa o número de classes.

Acurácia é uma média da relação entre observação corretamente previstas para aquela classe e o total de observações.

$$Acurácia = \frac{\sum_{i=1}^l \frac{TP_i + TN_i}{TP_i + FP_i + FN_i + TN_i}}{l} \quad (2.15)$$

Precisão é uma média da relação entre as observações verdadeiramente positivas e o total de observações positivas, sejam falsas ou verdadeiras.

$$Precisão = \frac{\sum_{i=1}^l \frac{TP_i}{TP_i + FP_i}}{l} \quad (2.16)$$

Sensibilidade é uma média da relação entre observações positivas verdadeiras e todas as observações na classe atual, também conhecido como *recall*.

$$Sensibilidade = \frac{\sum_{i=1}^l \frac{TP_i}{TP_i + FN_i}}{l} \quad (2.17)$$

*F-Score* é a média harmônica entre Precisão e *Recall*.

O desempenho do modelo também é avaliado pela sua curva ROC *Receiver operating characteristic curves* [Fawcett, 2006]. Por meio dela é possível medir a precisão através da área abaixo da curva ROC, AUC (Area Under the Curve). Os gráficos ROC são gráficos bidimensionais em que a taxa de TP é plotada no eixo Y e a taxa de FP é plotada no eixo X. Um gráfico ROC descreve a relação entre os verdadeiros positivos e falsos positivos (benefício X custo), conforme mostra a figura 2.12.

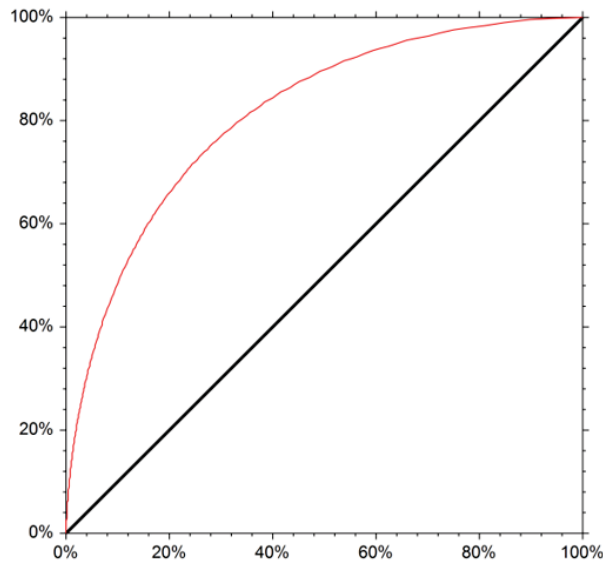


Figura 2.12: Exemplo de um gráfico que mostra a curva ROC, em vermelho.

Devido ao desbalanceamento da base, as curvas ROC permitem analisar os resultados não apenas pela sua precisão, mas também pela relação entre sensibilidade e especificidade, e assim, permite uma análise mais realista dos resultados. Curvas ROC são aplicadas em problemas binários mas podem ser estendidas para problemas multi-classe aplicando a regra de "um contra todos", onde cada classe é comparada as demais e a média das curvas representa o desempenho do modelo.

A curva de erro e rejeição (CER) é uma função que representa o erro de um classificador em função de sua taxa de rejeição [Oliveira et al., 2002]. É produzida pela plotagem da taxa de erro de um classificador contra a variação da taxa de rejeição, variando de 1 a 0 ou seja, inicia com 100% de rejeição. Todas as CERs têm um erro de 0 ponto percentual para um taxa de rejeição de 100 pontos percentuais e, portanto, convergem no ponto (0, 1). As equações abaixo representam as taxas de erro e rejeição:

$$Taxa\ de\ erro = \frac{N_{err}}{N_B} * 100 \quad (2.18)$$

$$Taxa\ de\ rejeição = \frac{N_{rej}}{N_B} * 100 \quad (2.19)$$

Sendo  $B$  um conjunto de testes com  $N_B$  imagens,  $N_{rej}$  o número de imagens rejeitadas,  $N_{rec}$  o número de imagens classificadas corretamente e  $N_{err}$  o número de imagens classificadas erroneamente.

Desta maneira é possível selecionar um modelo por meio da taxa de erro desejada, a taxa de rejeição aceitável ou ambas. Se a taxa de erro desejada é conhecida, move-se horizontalmente pelo gráfico e seleciona-se o classificador com a menor taxa de rejeição. Por outro lado, se a taxa de rejeição aceitável for conhecida, seleciona-se o classificador com a menor taxa de erro para essa taxa de rejeição.

### 3 Estado da arte

Não é de hoje que se busca otimizar o processo de análise dos resultados de testes rápidos, especialmente os quantitativos, cujas características são mais difíceis de analisar. Para se ter uma ideia, alguns estudos mostram que a cada inspeção visual há diferenças nos resultados, e que a discordância entre especialistas pode ser substancial, de 20 a 30 pontos percentuais. [Mudanyali et al., 2012].

A literatura destaca várias abordagens na tentativa de resolver o problema, uma delas é focada no aprimoramento do agente colorante, o que permite melhora no sinal das bandas da tira. Conforme [Omidfar et al., 2011], a utilização de ressonância da superfície das nanopartículas de ouro e espectroscopia de *Raman* em nanopartículas de ouro e prata, eleva a confiabilidade do teste [Luo et al., 2016]. Para [Zhang et al., 2010] existe uma relação direta entre a concentração do analito com os níveis de cinza na imagem. Para provar essa afirmação, inicialmente foi necessário a redução do ruído da imagem, e isso foi feito com a operação de adição de imagens. O filtro de mediana também foi aplicado para reduzir o ruído. De acordo com o espaço de cores YUV, Y corresponde a luminância da imagem e contém toda informação sobre sua escala de cinza. Pode ser convertido do padrão RGB com a seguinte equação:

$$Y = 0,299R + 0,587G + 0,114B \quad (3.1)$$

Com a imagem em escala de cinza é possível detectar as bordas de uma imagem com mais facilidade e, assim, segmentar as linhas de controle e teste. Com o valor de cada pixel da região segmentada pode-se, finalmente, fazer a correspondência com a concentração do analito na tira de teste original.

#### 3.1 Dispositivos de aquisição

Apesar do fato de que a melhora na intensidade do sinal facilita a análise do resultado do teste, na realidade o aumento da sensibilidade dos agentes colorantes cria a necessidade de aparatos mais sofisticados para realizar a leitura desse sinal. Como em [Gui et al., 2014], onde se propõem um leitor *CCD* de testes rápidos para fitas que usam *quantuns-dots* como elemento de marcação. Nesta solução a imagem capturada precisa se transferir e processada em um computador a parte. A captura de imagens com *CCD* tem a vantagem de se poder regular o tempo de exposição a luz, e com isso, alcançar maior sensibilidade na imagem [Yang et al., 2008]. Em [Zheng et al., 2016], por exemplo, são utilizadas câmeras *CMOS* (*Complementary metal-oxide-semiconductor*) para aquisição das imagens por questões de qualidade, pois, nesse trabalho, a segmentação das bandas das fitas é feita através do algoritmo de agrupamento (*clustering*) *K-Means*. Dispositivos *CCD* também são utilizadas em campo. Em [Mirasoli et al., 2012] se propõe um biossensor portátil e compacto para quantificação rápida e ultrasensível de fumonisinas (micotoxinas) (fumonisina B1 + fumonisina B2) em milho. O sistema é composto por uma caixa para inserção das tiras de imunoensaio de fluxo lateral e um dispositivo *CCD* portátil altamente sensível.

Além disso, o uso de *smartphones* como dispositivo de aquisição vem sendo bastante explorado. Eles são semelhantes a computadores em miniatura, com sistemas operacionais, memória interna e câmeras com lentes de alta qualidade. Também são mais acessíveis e baratos do que os dispositivos laboratoriais analíticos portáteis. No entanto, em alguns países, possuem algumas restrições regulamentárias para aplicações de sensoriamento biomédico [Contreras-Naranjo et al., 2016]. Na literatura, o trabalho de [Carrio et al., 2015] apresenta um leitor baseado em *smartphones* para aquisição de imagens de tiras de testes de fluxo lateral. Para minimizar todos os problemas relacionados à aquisição da imagem, uma pequena caixa de luz foi projetada para manter constante a posição relativa entre o *smartphone* e a fita, ao mesmo tempo que bloqueia qualquer interferência luminosa externa. A mesma solução foi adotada em um dispositivo para detecção de cortisol na saliva através de fluorescência, descrito em [Zangheri et al., 2015]. O estudo de [Barbosa et al., 2015] utiliza tiras de teste *LFIA* convencionais em conjunto com anticorpos para detectar patógenos bacterianos de origem alimentar, a *Salmonella* e *Escherichia coli O157*. A fluorescência foi registrada inserindo a tira de nitrocelulose em um fluorímetro, que é composto por um módulo

óptico contendo uma fonte de luz de LED, um conjunto de filtros de fluorescência e uma lente acoplada à câmera do *smartphone*.

## 3.2 Metodologias de análise de imagens

O trabalho de [Carrio et al., 2015] utiliza a clássica metodologia de classificação de imagens baseada em reconhecimento de padrões, descrita no capítulo anterior. A base de treinamento é composta por 683 imagens, que contém 2.696 tiras de testes do tipo LFIA, usadas para medir a presença de drogas e álcool na saliva. Todas as imagens foram adquiridas com câmeras de *smartphones* adaptados. Neste trabalho, inicialmente, a imagem passa pela etapa de pré-processamento, para corrigir variações de rotação e posição. Depois, na segmentação, o primeiro passo é a localização da região das tiras na imagem por meio da correspondência de modelos (*template matching*). Para isso, são definidos os pontos ORB (*oriented FAST (Features from Accelerated Segment Test)*), e então é feito o cálculo do descritor BRIEF (*Binary Robust Independent Elementary Features*). Um exemplo dessa correspondência com a imagem de referência pode ser visualizado na figura 3.1. O próximo estágio da segmentação visa localizar as áreas coloridas em cada tira e, para isso, os canais R, G e B da imagem são binarizados. Uma operação morfológica "AND" é aplicada entre os canais G e B e entre os canais G e R. Para finalizar, uma operação morfológica de fechamento é aplicada na imagem resultante.



Figura 3.1: Processo de correspondência entre descritores FAST e BRIEF (ORB) na imagem de referência (à esquerda) e na imagem original (à direita). Linhas brancas denotam os pontos correspondentes. Em verde, é representada a região de interesse.

Na etapa de extração de características, um histograma lateral é extraído de cada área da imagem obtida na segmentação. E, finalizando o processo, para a classificação das imagens foram criadas CNNs especializadas para cada banda da tira, sendo um classificador para as bandas de teste e um para as bandas de controle. Os parâmetros dos classificadores foram definidos com a aplicação de um algoritmo de validação cruzada. As imagens foram divididas em bases de treinamento, validação e teste em uma proporção de 15%, 15% e 70%. A rede foi treinada com 6 tipos de classes, definidas de acordo com o grau de intensidade da coloração das linhas. Os resultados indicaram uma acurácia de 100% na classificação das linhas de controle e 89,76% nas linhas de teste. Por meio da matriz de confusão foi possível ver que a classe com intensidade de cor 'indeterminada' teve a maior taxa de confusão, sendo confundida com as classes de coloração baixa ( tiras negativas) em mais de 20% dos casos.

Em outro recente estudo, [Kim et al., 2017], a concentração de álcool na saliva é analisada com o auxílio de técnicas de aprendizado de máquina. Foram testados 4 padrões de cores (RGB, HSV, YUV, and LAB) na extração das características e 3 algoritmos de aprendizado de máquina (LDA (*Linear discriminant analysis*), SVM e CNN). As classes do problema representam as concentrações padrão de álcool detectado na saliva, sendo eles 0,0%, 0,02%, 0,04%, 0,08% e 0,30%. Segundo os autores, foram adquiridas 14 imagens de cada classe, totalizando 70 imagens para o treinamento. Para os testes, também foram adquiridas imagens de tiras com concentrações intermediárias (0,01%, 0,03%, 0,06%, e 0,15%). Cada classificador foi treinado com as diferentes representações de cores, e foi usado o PPV (*Positive Predictive Value*) como medida de desempenho. Todos modelos obtiveram valores acima dos 95 pontos percentuais. Além disso, os algoritmos SVM e CNN foram testados com outras bases de dados, [Asuncion e Newman, 2007], e sua acurácia comparada ao repositório deste trabalho, denominado 'RGB'. Os resultados podem ser visualizados na figura 3.2:

Dados	Tamanho	Classes	SVM		CNN	
			Tempo (s)	Acurácia (%)	Tempo (s)	Acurácia (%)
RGB	70	5	0.267	100	0.275	99
Íris	150	3	0.065	93.33	0.191	99
Coração	270	2	0.115	88.92	0.397	89.63
Segmentos	2310	7	1.551	92.64	21.488	96.8
Vinho	178	3	0.062	97.22	0.369	98.8
Vinho Tinto	1599	6	1.230	58.75	3.325	61.6
Vinho Branco	4898	7	7.977	50.71	20.228	54.98

Figura 3.2: Comparativo entre diferentes repositórios de imagens com os dados das fitas com concentração de álcool na saliva usando o padrão de cores RGB.

Em [Solmaz et al., 2018], é descrito um modelo que usa *smartphones* para aquisição de imagens, e algoritmos de aprendizado para classificação de imagens de tiras de teste colorimétricas. A base de imagens foi adquirida usando diferentes modelos de *smartphones* e simulando diferentes condições de iluminação, por meio do uso de lâmpadas incandescentes, fluorescentes e luz natural, formando um total de 385 imagens. O problema é composto por 6 classes de cores que representam concentrações de peróxido na tira (0, 1, 3, 10, 30 e 100 ) ppm. Os algoritmos SVM e Random Forest foram utilizados nos experimentos e a métrica utilizada para avaliar o desempenho foi a acurácia. A classificação usando os dados de treinamento com validação cruzada atingiu mais de 95% de acurácia.

Algoritmos de aprendizado de máquina também são utilizados na identificação de nanopartículas de ouro (NPs). No estudo de [SoRelle et al., 2016], os autores desenvolveram um algoritmo que identifica NPs acumuladas em secções do tecido. Para isso foram injetadas nanopartículas de ouro em tecidos de ratos, e posteriormente, capturadas imagens desses tecidos para serem analisadas. Assim, a segmentação visou separar as imagens em três componentes, sendo eles *i* fundo, *ii* tecido e *iii* potenciais NPs, baseado na média da intensidade de cada pixel. Essa segmentação permitiu um cálculo mais preciso da bio-distribuição, medindo o número de *pixels* que correspondem ao tecido dentro do campo de visão. Os limiares foram obtidos através da análise do histograma das intensidades dos *pixel* de cada camada. O histograma foi calculado com 510 *bins* e depois re-amostrado (usando interpolação) a cada 5 unidades de intensidade. *Pixels* com intensidades mais baixas foram segmentados como fundo. O limiar para segmentar o fundo foi calculado como sendo o primeiro mínimo do histograma, multiplicado por um parâmetro definido pelo usuário, que é ligeiramente maior do que 1, para permitir o ajuste fino. A base de treinamento foi formada apenas com áreas da imagem segmentadas como sendo potenciais NPs. Para o treinamento do algoritmo *k-means*, foram usados inicialmente 3, 4 e 5 *clusters*. Como resultado, quatro *clusters* foram encontrados pelo algoritmo e correspondem aos espectros esperados do tecido injetado e manchado. Vários outros algoritmos de aprendizado de máquinas também foram explorados para este propósito, incluindo o Support Vector Machine (SVM), mas o *k-means* teve os melhores resultados.

Em um estudo semelhante, redes neurais do tipo DBN (*Deep Belief Network*) são aplicadas na tarefa de quantificar a concentração de nanopartículas de ouro em imagens de tiras de testes imunocromatográficos (GICS), [Zeng et al., 2016]. O objetivo dessa análise é auxiliar na tarefa de segmentação, fazendo a distinção entre linhas teste e controle.

Em [Berg et al., 2015], testes rápidos do tipo ELISA *enzyme-linked immunosorbent assays* são submetidos a um processo de análise para estimar sua rotulação usando o algoritmo de aprendizado de máquina AdaBoost (*adaptive boosting machine learning algorithm*), alcançando uma acurácia de 99% na detecção do vírus HSV (*Herpes Simplex Virus*) tipo 1 e 99,4% para o HSV tipo 2.

## 4 Base de Dados

Para aquisição das imagens foram usados três leitores CCD desenvolvidos por um grupo de pesquisadores brasileiros liderados pelo pesquisador Leonardo Foti<sup>1</sup>, no Instituto de Biologia Molecular do Paraná em Curitiba-PR. As leitoras estão ilustradas na figura 4.1(a) e são identificadas com cores, sendo elas "cinza", "preta" e "verde". Foram fornecidas 7.236 imagens de tiras de teste com dimensões 60 x 280 *pixels*, padrão RGB e 24 *bits* de intensidade, sendo 8 *bits* por canal, conforme ilustrado na 4.1(b). As imagens apresentam algumas diferenças de luminosidade entre si devido aos diferentes leitores utilizados.



(a) Leitoras usadas para aquisição de imagem.



(b) Exemplo da imagem da tira teste.

Figura 4.1: Imagens das leitoras e um exemplo de imagem gerada.

A rotulação das tiras foi feita por dois especialistas treinados. Cada tira pode ser classificada em três categorias: '1' para Reagente, '0' para Não-Reagente e '-1' para Inválido. A combinação do resultado das bandas controle e teste de cada tira fornece o resultado final:

Rotulação (Controle/Teste)	Resultado
1/1	Reagente
1/0	Não-Reagente
0/-1	Inválido
0/1	Inválido
0/0	Inválido
1/-1	Inválido
-1/0	Inválido
-1/1	Inválido
-1/-1	Inválido

Tabela 4.1: Resultado do teste conforme combinação das linhas.

Com essa divisão a base é formada por 1.714 imagens de tiras positivas, 1.295 imagens de tiras negativas e 4.227 imagens de tiras inválidas. Considerando a divisão das tiras por banda, 6.088 são bandas rotuladas reagentes, 4.380 rotuladas não reagentes e 4.004 rotuladas como inválidas. As bandas de uma tira são mostradas na figura 4.2. A divisão da base por banda é importante pois, após a etapa de segmentação, a extração das características é feita a partir das bandas da tira.

<sup>1</sup><https://lattes.cnpq.br/3677115511155174>

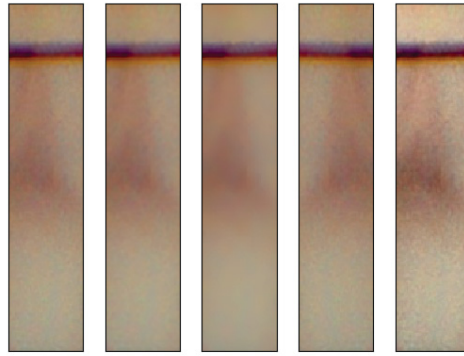


Figura 4.2: Imagens de bandas segmentadas. (a) Exemplo de banda reagente. (b) Exemplo de banda não reagente. (c) Exemplo de banda inválida.

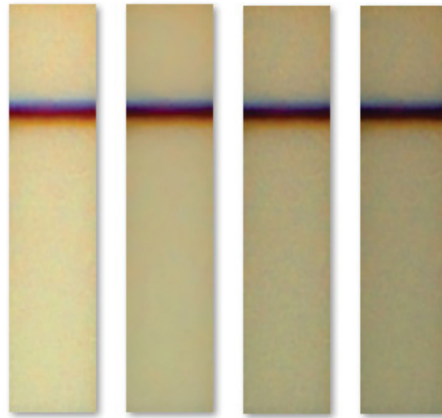
## 4.1 Aumento de dados

O aumento de dados (do inglês *Data Augmentation*) tem o objetivo de aumentar o desempenho do treinamento do algoritmo por meio do aumento do número de amostras da base, quando a base disponível tem uma quantidade limitada de amostras ou com pouca diversidade. É especialmente importante nas CNNs, que requerem um grande número de imagens do treinamento. Algumas técnicas utilizadas para gerar novas imagens a partir das imagens originais são: variação na translação, rotação ou escala, espelhamento, inserção de ruído na imagem entre outras.

A aplicação destas técnicas exige cuidados pois, a alteração de certas propriedades da imagem pode alterar seu significado. Especialmente na base deste trabalho, a adição de ruídos ou a rotação da imagem, por exemplo, comprometem a sua classificação, uma vez que o alinhamento e a textura das bandas, são características de decisão. Assim, duas novas bases de dados aumentados foram geradas, da seguinte forma: Para cada imagem da base original, foram geradas quatro novas imagens, aplicando: *i*) filtro gaussiano de *kernel*  $3 \times 3$ ; *ii*) filtro bilateral com diâmetro de vizinhança de tamanho 5; *iii*) espelhamento de imagem no eixo vertical (não prejudica suas propriedades) e; *iv*) equalização local do canal Y, do formato YCrCb, conforme ilustrado na figura 4.3(a). A segunda base de dados aumentados tem o objetivo único de reproduzir a variação de luminosidade que incide nas imagens no momento da aquisição. Para isso, cada imagem é convertida para o formato HSV, a fim de isolar o canal V, que detêm as características de iluminação da imagem. Por meio do aumento e da diminuição dos valores dos *pixels* deste canal, consegue-se o efeito de clareamento e escurecimento da imagem. Sendo assim, quatro imagens são criadas, em duas imagens são adicionados, respectivamente, 15 e 30 unidades de intensidade em cada *pixel* e, em outras duas imagens, são subtraídas, respectivamente, 15 e 30 unidades de intensidade de cada *pixel*. Para finalizar, cada imagem é convertida novamente ao padrão RGB. Como resultado, são criados duas imagens mais escuras e duas imagens mais claras que a original, conforme ilustrado na figura 4.3(b). Os valores subtraídos e acrescentados dos *pixels* foram definidos com base nos valores máximo e mínimo da média dos valores do canal V, dentre todas as imagens da base.



(a) Imagens geradas com *data augmentation* na base de tiras. A imagem original mais a esquerda e as imagens geradas a partir dela com a convolução de filtro gaussiano, convolução de filtro bilateral, espelhamento, equalização local.



(b) Imagens geradas com *data augmentation* aplicando variação de luminosidade do canal V das imagens com o aumento e diminuição da intensidade dos *pixels*.

Figura 4.3: Exemplos de imagens aumentadas de cada uma das duas bases.

Desta forma, foram criadas duas bases aumentadas contendo, cada uma, 28.944 imagens. Juntas, as bases de dados somam 65.124 imagens distintas, conforme tabela 4.2. Todos os experimentos da metodologia 6.1 utilizam apenas a base original, com exceção do experimento da seção 6.2. Já a metodologia 6.6 utilizada a base original e também dados aumentados, mostrados em dois experimentos distintos.

	Base original	Base aumentada 1	Base aumentada 2	<b>total</b>
positivas	1.714	6.856	6.856	15.426
negativas	1.295	5.180	5.180	11.655
inválidas	4.227	16.908	16.908	38.043
<b>total</b>	7.236	28.944	28.944	<b>65124</b>

Tabela 4.2: Tabela que mostra a quantidade de imagens de cada base, e sua divisão por tipo.

Em experimentos que fazem uso dos dados aumentados, a base de treinamento é composta tanto de imagens originais como de aumentadas. Porém, a base de teste é composta somente por imagens originais. A base de treinamento não contém imagens aumentadas, cujas originais estão no conjunto de teste.

## 5 Metodologia

Este capítulo descreve as duas metodologias de classificação utilizadas neste trabalho. O objetivo de se ter duas metodologias é a possibilidade de comparação entre elas com base nos resultados dos experimentos. A análise comparativa permite ressaltar os pontos fortes e fracos de cada modelo, assim como suas semelhanças e diferenças. Além disso, cada metodologia aborda de maneira distinta os algoritmos de aprendizagem utilizados, explorando suas particularidades. A primeira proposta (metodologia 1) segue o fluxo tradicional de classificação de imagens com reconhecimento de padrões, conforme fluxograma da figura 5.1.

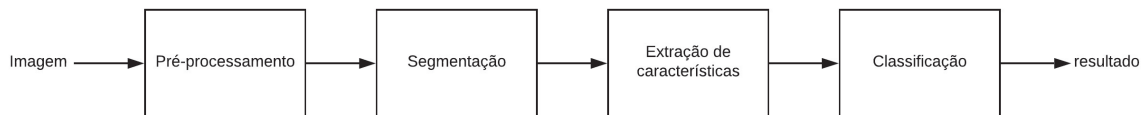


Figura 5.1: As etapas da metodologia 1. Pré-processamento, segmentação, extração de características e classificação.

Inicialmente a imagem é pré-processada para diminuir ruídos que podem atrapalhar a segmentação. Depois, a etapa de segmentação visa isolar as áreas da imagem que contêm as linhas da tira, resultado em duas imagens, que representam as linhas de teste e de controle. Ao término da segmentação, as características da imagem são extraídas, normalizadas e adicionadas em um vetor. Este vetor, então, é utilizado como entrada do algoritmo que fará a classificação da imagem. A figura 5.2 mostra o fluxograma detalhado da metodologia 1. A etapa de 'Concatenação' se refere a concatenação dos vetores de características extraídos de cada segmento, sendo que, o vetor referente a linha de controle é inserido no final do vetor referente a linha de teste.

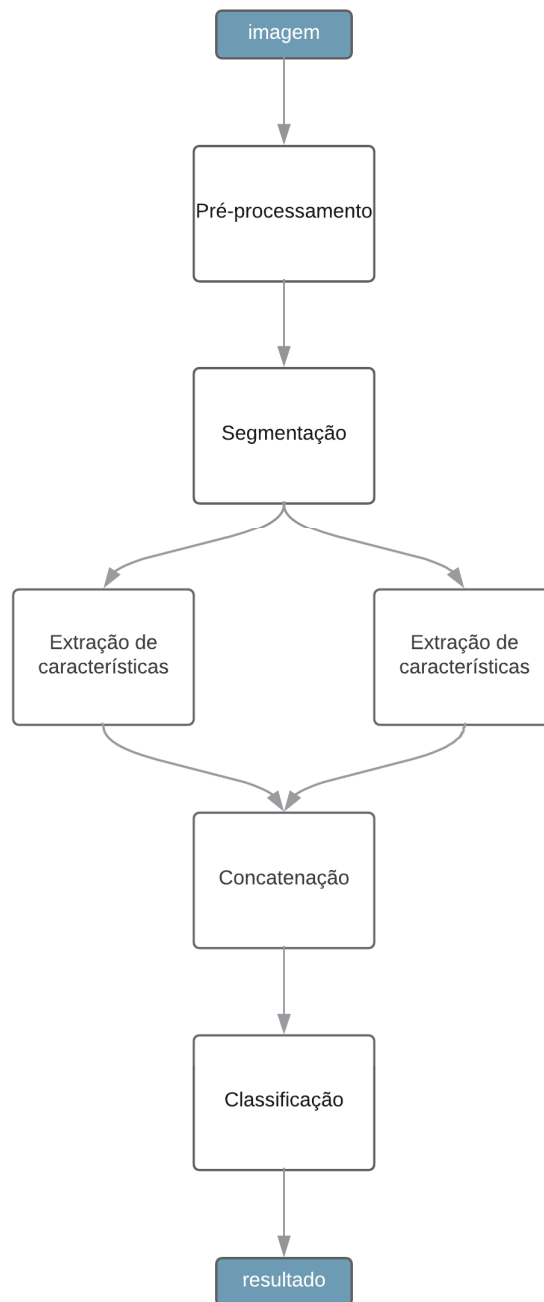


Figura 5.2: O fluxograma mostra as etapas do modelo. A extração acontece nos segmentos e não na imagem inteira.

Na segunda metodologia, as etapas de pré-processamento e de segmentação não são aplicadas. As características da imagem são adquiridas com a imagem inteira e não nos segmentos. A base de imagens é a mesma utilizada na metodologia anterior. Esta metodologia também é conhecida como solução *end-to-end* e segue o fluxo mostrado na figura abaixo.

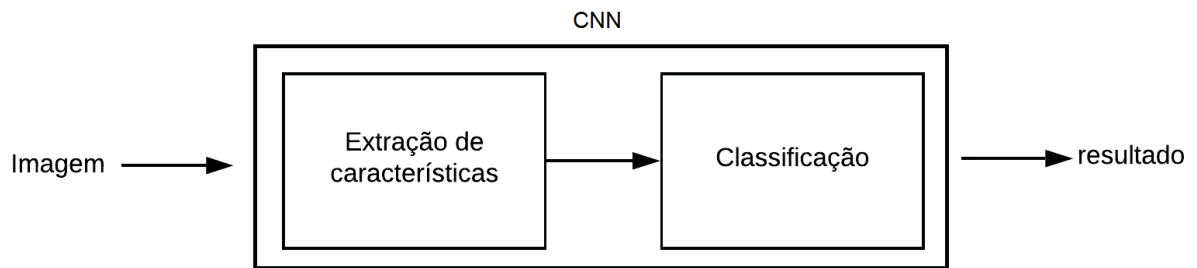


Figura 5.3: Etapas da metodologia *end-to-end*.

Para fins de nomenclatura, a metodologia que implementa a segmentação será chamada de metodologia 1, e a metodologia *end-to-end*, de metodologia 2.

## 5.1 Etapa de Pré-processamento

O pré-processamento tem o objetivo de salientar as linhas da tira em relação ao fundo da imagem e também remover ruídos. Para isso, cada imagem passa por uma equalização de histograma, o que aumenta seu contraste. Então, a imagem é convertida para o padrão YCrCb, e é aplicada uma equalização local de histograma no canal Y da imagem, usando o método CLAHE. O canal Y deste padrão contém as informações de luminância e, em uma imagem colorida, é a informação (canal) que precisa ser equalizada. Em seguida, a imagem é novamente convertida ao padrão RGB e passa por um processo de suavização por meio de um filtro bilateral. Este filtro suaviza a imagem removendo ruídos mas sem que as bordas das linhas sejam perdidas. Esse filtro funciona de forma localizada, preservando cantos e bordas como aqueles que estão presentes nas tiras dos testes imunológicos rápidos. O resultado desta etapa pode ser visto na figura 5.4.

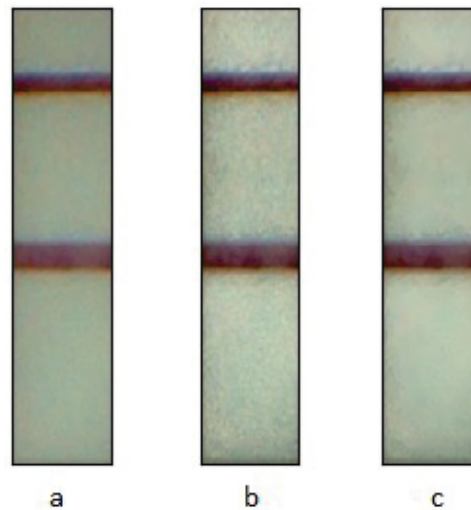


Figura 5.4: a) Imagem original. b) A imagem com contraste aumentado. c) A imagem b suavizada com filtro bilateral.

## 5.2 Etapa de Segmentação

A segmentação das tiras atende a metodologia 1 e visa isolar as áreas de interesse que, neste caso, são as linhas de controle e de teste. A segmentação consiste em dois passos. O primeiro passo divide a imagem em duas sub-imagens, ou segmentos, através de dois cortes horizontais, sendo um corte a  $3/8$  da altura da imagem e outro a  $3/4$  da altura, de acordo com valor de limiar estipulado. O restante da imagem é descartado por não conter informação relevante. Esse processo é descrito através do algoritmo 1.

---

**Procedure 1** Segmentação 1.
 

---

**Input:** *imagem*, *limiar*
**Output:** *segmentos*
 $segmentos = []$ 
 $imagemHSV = \text{converteBGRparaHSV}(imagem)$ 
 $h, s, v = \text{separaCanais}(imagemHSV)$ 
 $l = v.largura$ 
 $segControle = \text{divideImagem}(0, limiar)$ 
 $segTeste = \text{divideImagem}(limiar, limiar * 2)$ 
 $segmentos = \text{adiciona}(segTeste, segControle)$ 


---

Desta maneira, considerando o ponto 0 da imagem o canto superior esquerdo, a imagem gerada do *pixel* 0 até o primeiro limiar contém a região onde se localiza a linha de controle. A imagem gerada do primeiro limiar até o segundo, contém a região onde se localiza a linha de teste. Os valores de limiar foram obtidos através da análise da base de treinamento, que mostrou as posições média, mínima (posição mais distante do ponto zero) e máxima (posição mais próxima do ponto zero) das linhas na tira. Por meio destes valores é possível estimar a localização das linhas dentro da imagem. A figura 5.5 mostra a divisão da tira em segmentos.

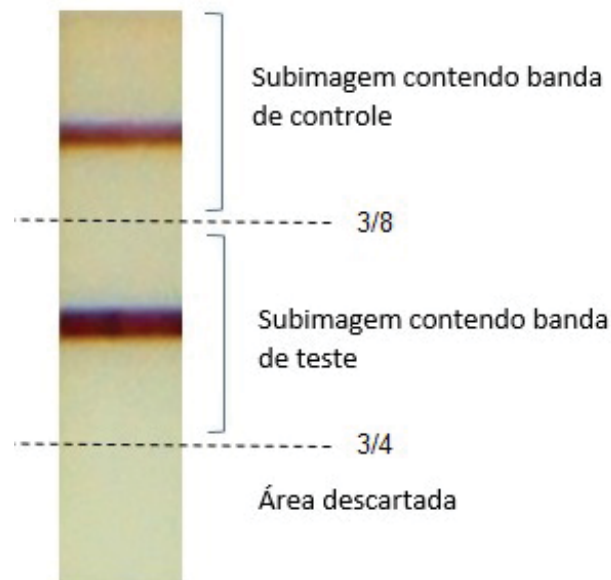


Figura 5.5: O resultado do primeiro passo da segmentação gera dois segmentos.

O segundo passo da segmentação visa agora identificar as linhas dentro de cada segmento gerado. Para isso, cada segmento é convertido para o formato HSV a fim de se isolar o canal V da imagem, que é o canal que contém as informações sobre a intensidade da cor. Então é aplicada uma binarização *OTSU*, visando destacar os *pixels* que compõem a região da linha, uma vez que existe uma diferença de intensidade de cor entre o fundo da imagem e a linha do segmento. Após a binarização, uma operação morfológica de fechamento é aplicada para preencher lacunas geradas na binarização e, assim, manter a proporção de largura e altura original da linha. Com isso, os contornos da imagem podem ser encontrados. Com eles é possível encontrar informações como altura, largura, posição e área da região de contorno. Essas informações são importantes nessa etapa para diferenciar uma possível linha de um ruído na imagem, por exemplo. A informação de área busca reforçar essa hipótese. A posição do contorno também é utilizada para eliminar ruídos muito próximos às extremidades da tira que, geralmente, não correspondem a linhas. Caso a segmentação não encontre nenhuma linha no segmento, a área total do segmento é utilizada para extração de características. As figuras 5.6 e 5.7 e o algoritmo 2 mostram o processo descrito.

---

**Procedure 2** Segmentação 2.
 

---

**Input:** *segmentos* = algoritmo 1

**Output:** *segs*
*caracteristicas* = []

*segs* = []

*kernel* = geraMatriz3x3()

**for** cada segmento de *segmentos* **do**

   *segOtsu* = binarizaOtsu(segmento.copy, 0, 255)

   *segFechamento* = aplicaFechamento(*segOtsu*, *kernel*)

   *contornos* = achaContornos(*segFechamento*)

   *contornos* = ordenaMaiorParaMenor(*contornos*)

   *cnt* = contornos[0]

   *caracteristicas* = adiciona(area(*cnt*), rect(*cnt*))

   **if** analisa(*caracteristicas*) **then**

     *segs* = adiciona(segmento[*cnt*])

   **else**

     *segs* = adiciona(segmento[*segmento*])

   **end if**
**end for**


---

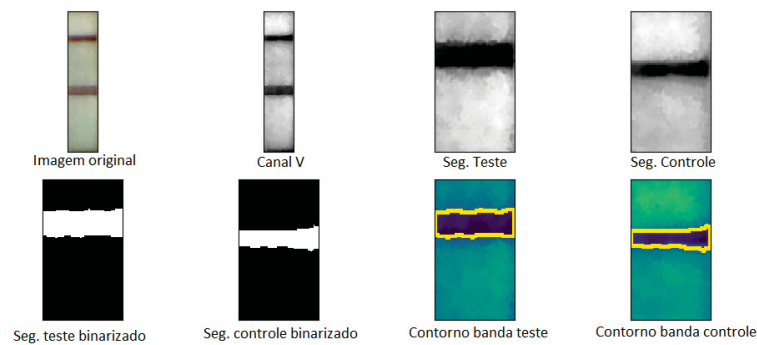


Figura 5.6: Processo de segmentação com uma imagem com 2 linhas. A partir da imagem original, são obtidos 2 segmentos correspondentes as regiões de controle e teste. Os contornos (em amarelo) mostram se o segmento contém uma linha ou não contém.

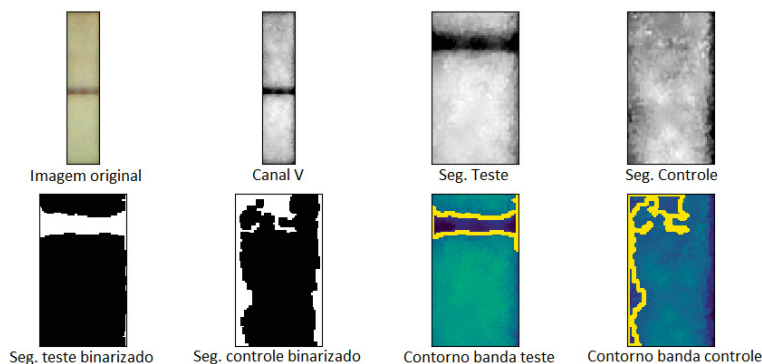


Figura 5.7: Processo de segmentação com uma imagem com 1 linha. Essa imagem mostra que através dos contornos obtidos é possível saber se tal contorno corresponde a uma linha ou não. A última imagem da figura mostra um contorno que não corresponde a uma linha.

A eficiência do processo foi medida através da taxa de detecção de linhas. O resultado da segmentação foi comparado com a base de treinamento, onde foi assumido que rótulos '0' não possuem linha e os demais rótulos ('1' e '-1') possuem linha. A taxa de acerto foi a seguinte:

- Para linhas positivas: 99,29%
- Para linhas negativas: 99,22%
- Para linhas indefinidas: 62,50%

O algoritmo encontrou linhas em quase a totalidade dos segmentos positivos e negativos. O erros ocorrem quando o algoritmo encontra menos ou mais linhas se comparadas com a rotulação. Pode-se atribuir o baixo rendimento nas imagens de tiras indefinidas ao padrão que elas representam, com baixa definição, imagens manchadas ou com linhas fora dos limites permitidos.

### 5.3 Extração de características

As características das imagens são extraídas de duas formas distintas neste trabalho. A primeira forma é selecionando algoritmos de representação de cor, textura e forma. A segunda maneira é delegando essa tarefa para uma CNN, sem interferência humana sobre que tipo de representação deve ser extraída. O método de "*transfer learning*" é aplicado para esse fim, usando um modelo pré-treinado.

Sendo assim, quatro conjuntos de características foram considerados: um conjunto foi extraído com ajuda de uma CNN, 2 grupos com características programadas e um grupo extraído pela leitora, cujos valores foram fornecidos juntamente com a base de imagens. A descrição de cada conjunto é detalhada a seguir. Os conjuntos 1, 2, 3 e o de características fornecidas, são aplicados na metodologia 1.

Os vetores de características foram normalizados usando a regra Min-Max conforme a equação 5.1.

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (5.1)$$

#### 5.3.1 Conjunto de características (originais) extraídas pela leitora.

O primeiro conjunto se refere às características de cor, intensidade e forma extraídas pela leitora no momento da aquisição da imagem. Assume-se que a segmentação para extração destas características foi manualmente definida, ou seja, para cada tira, foi definido manualmente a região onde cada linha se localiza exatamente. Devido a restrições de direitos autorais sobre o algoritmo extrator, nenhum detalhe sobre a implementação será descrita, apenas um breve descritivo sobre cada característica implementada:

- Linha de base do pico: Os dados da linha de base (*baseline*) são encontrados usando o método ALS e armazenados em um vetor multidimensional. Depois é calculado a média dos valores da linha de base na região de pico. Esse valor representa a iluminação da região e pode ser usada para determinar se a exposição das imagens está ocorrendo de maneira satisfatória.
- Intensidade do pico: A intensidade dos picos é calculada somando os valores da curva entre os pontos da largura do pico à meia altura e dividindo este valor pela largura da curva.
- Altura dos 20 *pixels* mais fracos: A partir dos dados dos 20 *pixels* mais fracos, é subtraída a linha de base dos valores da média e é calculada a intensidade na região de pico encontrada. Esta intensidade mostra se o pico analisado possui alguma mancha com a mesma intensidade da linha.
- Intensidade da saturação: Os dados de saturação são ajustados para remover a linha de base e suavizados com o filtro *Savgol* [Savitzky e Golay, 1964]. A partir destes dados, é calculada a intensidade do pico de saturação na região de pico encontrada.
- Média da saturação absoluta na região: Com os dados de saturação, é realizada a média dos valores de saturação na região de pico encontrada. A saturação representa a intensidade da cor na região de pico e é possível distinguir entre os picos que possuem uma cor forte e outros artefatos, como sombras, que não apresentam uma coloração intensa.
- Intensidade do canal *hue*: Os dados de *hue* são ajustados para remover a linha de base e suavizados com o filtro *Savgol*. A partir destes dados, é calculada a intensidade do pico de *hue* na região de pico encontrada.
- Média do *hue* absoluto na região: É realizada a média dos valores de *hue* na região de pico encontrada. O *hue* representa a tonalidade do pico e pode ser um parâmetro para filtrar os picos incorretos caso sua intensidade esteja incorreta.

- Posição do pico: A posição dos picos é determinada a partir dos picos encontrados nos dados das médias dos valores do canal verde das linhas.
- Largura do pico, posição do início do pico e posição do final do pico: Tais valores são determinados a partir da posição dos picos e encontrando as raízes dos dados quando deslocados para a meia-altura do pico. É possível usar a largura como parâmetro para filtrar linhas inválidas.
- Coerência do pico: A coerência do pico é um valor booleano que é verdadeiro caso na região de pico haja variação na saturação e na cor da tira e falso, caso contrário.
- Distância do pico à borda da tira: Esta distância é importante para avaliar picos muito próximos da borda da tira, principalmente nos casos de picos falsos.
- Distância do pico à borda da região de linha: As linhas devem ter uma posição fixa nas tiras, no entanto é possível que por alguma falha, a linha fique numa região duvidosa entre duas regiões de linha, então pode ser avaliada usando este parâmetro.
- Intensidade da maior anomalia no desvio padrão na região de linha : Os dados de desvio padrão são ajustados para remover a linha de base e suavizados com o filtro *Savgol*. A partir destes dados, são encontrados os picos e determinada a maior intensidade para tal região de linha.
- Intensidade do pico com linha de base linear: A linha de base linear diminui a sensibilidade dos dados, mas é um bom parâmetro para determinar caso a linha de base ALS esteja criando um pico artificial. Estes dados são determinados usando o filtro *Detrend*, que remove o valor médio ou tendência linear da linha de base.
- Altura do desvio padrão normalizado: Com os dados ajustados do desvio padrão, é calculada a intensidade do pico de desvio padrão na região de pico encontrada. O desvio padrão é uma grandeza útil para determinar se há muita variação de intensidade na região de pico.

Para cada segmento são gerados 17 valores, 34 valores por tira.

### 5.3.2 Conjunto 1: Características de cor.

Esse conjunto de características visa replicar o conjunto anterior, porém, usando o algoritmo de segmentação para identificação automática das linhas nas tiras. A partir do descritivo, tentou-se achar a fórmula que mais se aproximasse da original.

Para o cálculo da linha de base é utilizado o canal V da imagem não segmentada. Para cada linha da imagem é calculado a média dos seus *pixels*, resultando em um vetor de tamanho 1 x 280 (altura da imagem original). Este vetor é usado como entrada do algoritmo 3, que foi baseado em [Eilers e Boelens, 2005]. Os dados da linha de base são encontrados usando o método ALS. A linha de base converge quando o vetor de pesos não sofre mais alterações, na prática, 10 iterações são suficientes para isso.

Após o cálculo da linha de base, a imagem é segmentada.

---

#### Procedure 3 Calcula Linha de Base

---

**Input:**  $y, \lambda, p$

**Output:**  $z$

```

 $m = tamanho(y);$ 
 $D = calculaDiferenca(criaMatrizIndentidade(m), 2)$ 
 $w = criaMatrizDe1s(m)$ 
for  $i = 0$  até 10 do
   $W = spdiags(w, 0, m, m);$ 
   $A = W + \lambda * D' * D$ 
   $C = fatoraCholeski(A)$ 
   $z = C (C' (w . * y));$ 
   $w = p * (y > z) + (1 - p) * (y < z);$ 
if  $verifica\_convergencia(w)$  then
  break
end if
end for

```

---

---

**Procedure 4** Fatoração de Cholesky
 

---

**Input:**  $A$ **Output:**  $L$  $L = \text{Faça}([0.0] * \text{tamanho}(A))$  Para todo elemento da matriz  $A$ **for**  $k = 0$  até  $\text{tamanho}(A)$  **do**  **for**  $l = 0$  até  $k + 1$  **do**     $s = \text{soma}(\text{Faça}(L[k][l])^2$  para todo elemento da matriz  $L$  )    **if**  $k == l$  **then**       $L[k][l] = \text{sqr}(A[k][k] - s)$     **else**      **for**  $i = k + 1$  até  $\text{tamanho}(A)$  **do**        **for**  $j = 0$  até  $k + 1$  **do**           $(1.0/L[j][j] * (A[i][j] - s)$         **end for**      **end for**    **end if**  **end for****end for**

- Média da linha de base (*baseline*). Valor obtido pela média dos valores da linha de base que correspondem ao segmento da imagem. Ou seja, apenas uma fração da linha de base é utilizada.
- Intensidade do pico. A partir do canal V da imagem, é extraído um vetor contendo os valores médios do somatório dos *pixels* de cada linha da região. Este vetor é ordenado e, então, exclui-se 50% dos valores centrais deste vetor, que correspondem aos valores do pico a meia altura. Do vetor restante, é calculado a média e dividido pela largura da região. A largura é o número de linhas da imagem de entrada.
- Altura dos *pixels* mais 20 fracos. É a média dos 20 *pixels* de menor valor da imagem subtraída da média da linha de base.
- Altura, largura, área e posição da linha. Inicialmente são achados os contornos da imagem. Caso seja encontrado mais de um contorno, é selecionado o de maior área, pois geralmente, corresponde a linha de teste ou de controle. Com os dados do contorno, é possível definir os valores de altura, largura, área e posição (coordenadas  $x$  e  $y$  do canto superior esquerdo).
- Intensidade da saturação. Seja  $Vt_{sat}$  um vetor contendo os valores médios do somatório dos *pixels* de cada linha do canal S (projeção lateral) da imagem e  $Bsl_{sat}$  o vetor que representa sua fração da linha de base. A intensidade da saturação é obtida subtraindo  $Vt_{sat}$  de  $Bsl_{sat}$  e aplicando o filtro *Savgol* [Savitzky e Golay, 1964] no vetor resultante.
- Média da saturação absoluta na região. Média dos valores de intensidade dos *pixels* do canal S da imagem.
- Intensidade do canal *hue*. Mesmo processo descrito para "intensidade de saturação", porém, utilizando o canal H.
- Média do *hue* absoluto na região. Mesmo processo descrito para "média da saturação absoluta na região", porém, utilizando o canal H.
- Distância do pico à borda da tira e distância do pico à borda da região de linha. Os picos da tira são encontrados usando a biblioteca *PeakUtils* [Duarte, 2017], que contém métodos para detecção de picos em vetores unidimensionais. A aplicação do método depende das projeções laterais do histograma da imagem, tanto vertical como horizontal. A projeção lateral vertical é encontrada por meio da média dos *pixels* em cada linha da imagem, enquanto a projeção horizontal é aplicada sobre cada coluna da imagem. O resultado pode ser visualizado na figura 5.8.

Assim, para cada segmento são gerados 13 valores.

### 5.3.3 Conjunto 2: Características de cor e textura.

Neste conjunto, há uma mescla de descritores de cor e de textura. A extração das características é feita no segmento inteiro, e não apenas nos *pixels* que compõe a linha do segmento, ou seja, apenas o primeiro passo da

segmentação é aplicado na imagem. Para cor, são extraídos o histograma, sua média e seu desvio padrão. Para textura, são extraídos os descritores *Local Binary Pattern (LBP)* e *Grey Level Co-Occurrence Matrix (GLCM)*. Ambos os descritores usam a imagem em intensidades de cinza. As propriedades definidas no GLCM foram *energia*, *contraste* e *homogeneidade* devido a afinidade com o problema. A altura e a largura da linha também são calculadas e inseridas no vetor. A medição da altura e da largura da linha é feita a partir da projeção vertical e horizontal do histograma de sua imagem binarizada como mostra a figura a seguir.

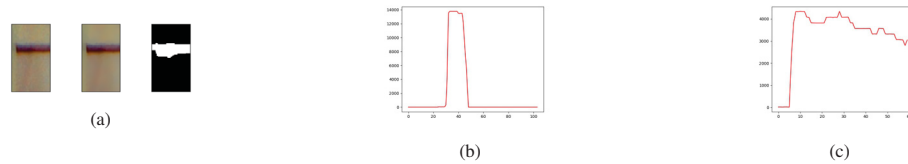


Figura 5.8: a) A imagem binarizada. b) Projeção de histograma lateral. c) Projeção de histograma vertical.

Dentre as características de cor, o cálculo do histograma é feito nos três canais de cores RGB, onde cada canal é quantificado em 8 bins, com a faixa de valores de intensidade variando de 0 a 256. Depois de calculados os três histogramas, eles são combinados em um vetor, formado por 512 valores, valor obtido pela combinação dos bins dos canais ((8x8x8)). A média de intensidades e o desvio padrão dos histogramas também são calculadas para cada canal.

A tabela 5.1 mostra a quantidade de características extraídas de cada descritor.

Tabela 5.1: Quantidade de características por tipo

	Histograma	Média	Desvio	Projeções	LBP	GLCM
# características	512	3	3	164	10	36

### 5.3.4 Conjunto 3: *Transfer Learning CNN*

Para a extração de características das imagens a partir de uma CNN pré-treinada, foi usada a arquitetura *Inception V3* [Szegedy et al., 2016], que é um modelo treinado na base "*ImageNet Large Visual Recognition Challenge*" [Russakovsky et al., 2015]. Para isso é feita a captura do vetor de características da penúltima camada da rede neural artificial, nesse caso, antes da camada de classificação denominada *softmax* conforme figura 5.9.

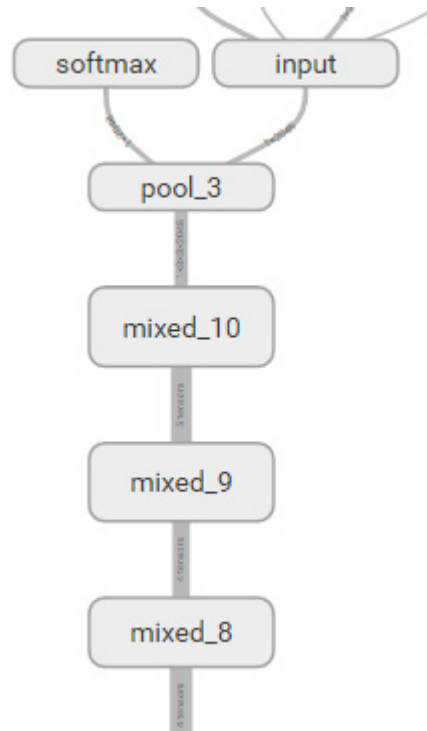


Figura 5.9: Camadas da CNN

Devido a *Imagenet* ser um modelo de rede com milhões de parâmetros, o retreinamento é feito apenas na última camada da rede. Um vetor com 2048 posições é extraído para cada segmento, totalizando 4096 posições para cada imagem.

## 5.4 Classificação

A previsão de classes é feita por algoritmos de aprendizagem supervisionada, seu aprendizado é baseado em exemplos rotulados da base de treinamento. A escolha do algoritmo ideal para cada problema depende de uma série de fatores como o tamanho, a qualidade e a natureza dessa base. Mas, mesmo considerando os fatores, não seria possível determinar o(s) melhor(es) algoritmo(s) sem antes testá-lo(s). Dentro do escopo deste trabalho, alguns algoritmos foram avaliados por seu desempenho em testes preliminares como, modelos de regressão linear, máquina de vetor de suporte (*SVM*), baseado em medidas de proximidade (*KNN*) e métodos de *ensemble* como *Random Forest*. Então, baseado na acurácia, no tempo para treinamento, como também pelo número de características necessárias para prever cada instância, os algoritmos *SVM* e *Random Forest* foram selecionados para os experimentos neste trabalho. Além disso, uma combinação de múltiplos classificadores, utilizando estes dois algoritmos em conjunto com um algoritmo de regressão linear, foi experimentado. Quando utilizada a metodologia 'end-to-end', a classificação é feita pela própria CNN, em suas camadas de decisão.

Para o treinamento, são utilizadas as bases de imagens originais e as bases de dados aumentados, conforme descrito no capítulo 4.

## 6 Resultados Experimentais

Este capítulo mostra os resultados dos experimentos que foram realizados com as duas metodologias citadas no capítulo 5 da proposta, respectivamente nas seções 6.1 e 6.6.

Inicialmente, foi calculada a acurácia da classificação das imagens, feito pelo algoritmo proprietário das leitoras, que esta fora do escopo deste trabalho, mas cujo desempenho serve de referência para os outros experimentos. O cálculo foi possível devido a disponibilidade dos rótulos originais de cada banda em cada tira e o valor previsto pelo algoritmo classificador. Tais valores foram fornecidos juntamente com as imagens. Sendo assim, a acurácia calculada foi **74,54** pontos percentuais. Este valor servirá como parâmetro de comparação para os resultados dos experimentos, visto que, um dos objetivos deste trabalho é a comparação do modelo aqui proposto com os métodos de classificação tradicionais. Além disso e, também para servir como parâmetro, as características extraídas por esse mesmo algoritmo foram submetidas aos algoritmos de aprendizagem de máquina, propostos neste trabalho, SVM e Random Forest. A subseção 6.1.1 detalha este experimento.

Em seguida, foram realizados os experimentos utilizando as metodologias propostas. Os experimentos foram divididos em vários modelos, sendo que, cada modelo utiliza uma combinação com um algoritmo de aprendizado e um conjunto de características, aplicando uma das metodologias propostas. Nesta metodologia, todos os experimentos foram efetuados usando apenas a base de imagens originais.

Assim, para a etapa de classificação as bases foram balanceadas, sendo que, o balanceamento acontece na divisão da amostragem, que pega o mesmo número de amostras para cada classe. Na extração de características pela CNN, usando *transfer learning*, foi utilizada a arquitetura *Inception V3* com retreinamento da última camada da rede. Até a presente data, uma nova versão desta arquitetura está disponível, a *Inception V4*, que foi testada, porém, sem ganhos substanciais para justificar a troca de arquitetura. Aqui, cada vetor de características é composto de 4.096 posições por imagem devido a concatenação dos vetores dos segmentos.

Os dados aumentados, descritos no capítulo 4, são usados como parte da base de treinamento. Para cada imagem que compõe a base de treinamento, são inseridas suas respectivas imagens aumentadas. A base de testes não contém dados aumentados. O experimento usando dados aumentados é reportado na seção 6.2.

E finalmente, para testar a qualidade da aquisição das imagens, cada leitora teve suas imagens testadas em um dos modelo de classificação. O experimento é descrito na seção 6.3.

### 6.1 Resultados da metodologia 1

Além do conjunto de características extraído pelas leitoras, os conjuntos 1, 2 e 3 foram testados para esta metodologia. As amostras são divididas em base treinamento e base teste numa proporção de 66,3% / 33,3% e, então, submetidas ao processo de classificação de cada modelo. Para computar os resultados, o procedimento é repetido dez vezes, sempre utilizando uma amostragem diferente da base. O valor médio dos resultados representa o desempenho do modelo. Os parâmetros utilizados para todos os modelos foram definidos usando um algoritmo de busca exaustiva (*grid search*) que combina diferentes configurações de valores candidatos através da técnica de validação cruzada.

A tabela 6.1 resume os resultados de acurácia obtidas nos experimentos desta metodologia. O melhor resultado fica com o conjunto de características extraídas pela CNN (conjunto 3), usando o algoritmo de aprendizagem SVM. Nesta metodologia, todos os resultados superam a acurácia da classificação original, obtido com a leitora.

Tabela 6.1: Resultados da acurácia de cada modelo da metodologia 1 com respectivos valores de desvio padrão.

	Random Forest	SVM
Características originais	<b>92,39</b> $\pm 1,1$	90,97 $\pm 1,2$
Conjunto 1	90,66 $\pm 1,5$	87,54 $\pm 0,9$
Conjunto 2	90,28 $\pm 1,3$	86,98 $\pm 0,5$
Conjunto 3	91,34 $\pm 0,8$	<b>94,17</b> $\pm 0,4$

As próximas seções detalham cada modelo. Será apresentada a matriz de confusão dos resultados, as métricas de desempenho e o respectivo gráfico ROC.

### 6.1.1 Conjunto de características extraídos pelas leitoras

Este conjunto de características se mostrou bastante eficiente usando algoritmos de aprendizado de máquina, com resultados acima dos 90 pontos percentuais. O modelo que utiliza *random forest* teve o segundo melhor desempenho dentre os modelos testados, alcançando 92,39 pontos percentuais de acurácia, sendo que o SVM também superou os 90 pontos. Ambos os modelos estão bem equilibrados com uma taxa de *Sensibilidade* na casa dos 98 pontos percentuais para as classes dos tipos 0 e 1, conforme tabelas 6.4 e 6.5. Um dado relevante já que as tiras positivas e negativas (válidas para diagnóstico), foram rotuladas corretamente. De acordo com as matrizes de confusão, tabelas 6.2 e 6.3, ambos os algoritmos não confundem a classe do tipo 0 com a do tipo 1 e vice versa, porém, o algoritmo de *random forest* consegue melhor desempenho na classe do tipo 2. A AUC nas figuras 6.1 e 6.2 estão muito similares, com valores próximos a 0,98 para os dois algoritmos. Os gráficos 6.3 e 6.4 mostram as curvas de rejeição e erro das três classes que compõe o problema. Se percebe que as classes 0 e 1 tem maior taxa de rejeição, à medida que o erro é deslocado para baixo. Para uma taxa de erro de 1 ponto percentual, a rejeição nesse modelo é, em média, de 10 pontos percentuais com o algoritmo RF e um pouco maior para o algoritmo SVM. Ao longo da curva, do total de classes que foram rejeitadas, mais de 70% das instâncias pertencem as classes 0 e 1, para RF. Essa taxa sobe para mais de 80% utilizando o SVM.

Os parâmetros de entrada para o RF foram  $n\_estimators = 63$ ;  $max\_depth = 25$ ;  $min\_samples\_leaf = 2$  e  $max\_features = "sqrt"$  e para o SVM foram  $C = 10$ ;  $kernel = "rbf"$  e  $gamma = "0.0001"$ .

Tabela 6.2: Matriz de confusão do modelo com RF e conjunto original.

	0	1	2
0	0,98	0,0	0,02
1	0,0	0,98	0,02
2	0,09	0,13	0,78

Tabela 6.3: Matriz de confusão do modelo com SVM e conjunto original.

	0	1	2
0	0,98	0,0	0,02
1	0,0	0,98	0,02
2	0,12	0,14	0,74

Tabela 6.4: Métricas de desempenho do modelo com RF e conjunto original.

	Precisão	Sensibilidade
0	0,92	0,98
1	0,90	0,98
2	0,95	0,78

Tabela 6.5: Métricas de desempenho do modelo com SVM e conjunto original.

	Precisão	Sensibilidade
0	0,89	0,98
1	0,89	0,98
2	0,94	0,74

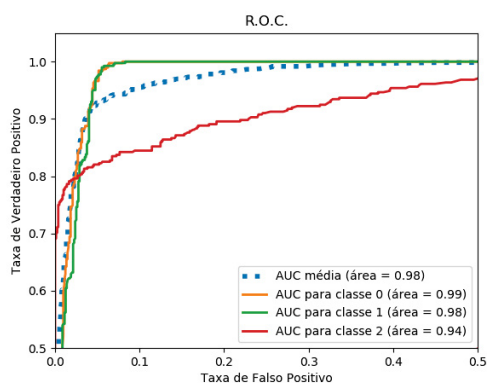


Figura 6.1: Curvas R.O.C. do modelo com RF e conjunto original.

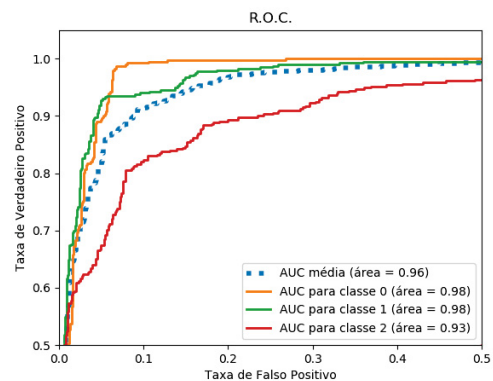


Figura 6.2: Curvas R.O.C. do modelo com SVM e conjunto original.

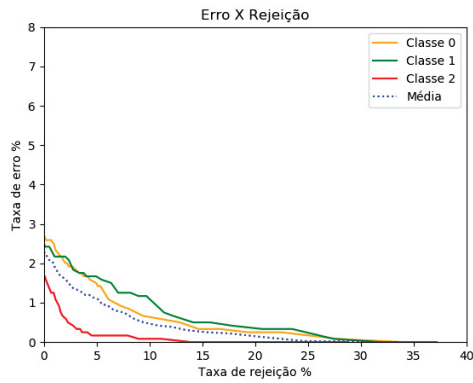


Figura 6.3: Gráfico de rejeição e erro do modelo com RF e conjunto original.

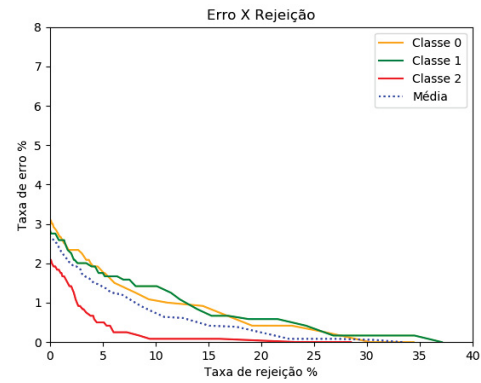


Figura 6.4: Gráfico de rejeição e erro do modelo com SVM e conjunto original.

### 6.1.2 Conjunto de características 1

Este conjunto reflete o esforço em se implementar um modelo cujas características se aproximem do conjunto anterior, que foi extraído pelas leitoras.

O modelo que utiliza *random forest* alcançou a acurácia de 90,66 pontos percentuais. Consegue identificar exemplos da classe do tipo 0 com uma taxa de 99 pontos e para classes do tipo 1, 95 pontos. Também não houve confusão entre estas classes. Assim, sua AUC média ficou em 0,97. Foram usados como parâmetros de entrada  $n\_estimators = 45$ ;  $max\_depth = 30$ ;  $min\_samples\_leaf = 1$  e  $max\_features = "sqrt"$ .

Ja o modelo que utiliza SVM teve um desempenho inferior para as classes tipos 0 e 1, mas consegue uma taxa maior na classe do tipo 2. Os resultados são detalhados nas matrizes de confusão, conforme tabelas 6.6 e 6.7. Os dois algoritmos tiveram um *f-score* de 95 pontos na classe tipo 0, mas o SVM foi inferior para os tipos 1 e 2, tabelas 6.8 e 6.9. Para o SVM foram usados como parâmetros de entrada  $C = 1000$ ;  $kernel = "rbf"$  e  $gamma = "0.01"$ , alcançando uma AUC média de 0,96.

Nos gráficos 6.7 e 6.8, a curva da classe 1 apresenta maior taxa de rejeição em relação ao erro, enquanto as curvas relativas às outras classes tem taxas menores e mais equidistantes uma da outra. Para uma taxa de erro de 1 ponto percentual, por exemplo, a rejeição das classes 0 e 2 fica abaixo de 10 pontos percentuais em ambos algoritmos, enquanto a classe 1 supera os 20 pontos percentuais.

Tabela 6.6: Matriz de confusão do modelo com RF e conjunto 1.

	0	1	2
0	0,99	0,0	0,01
1	0,0	0,95	0,04
2	0,08	0,19	0,73

Tabela 6.7: Matriz de confusão do modelo com SVM e conjunto 1.

	0	1	2
0	0,97	0,0	0,02
1	0,01	0,90	0,1
2	0,06	0,18	0,76

Tabela 6.8: Métricas de desempenho do modelo com RF e conjunto 1.

	Precisão	Sensibilidade
0	0,92	0,99
1	0,86	0,95
2	0,92	0,73

Tabela 6.9: Métricas de desempenho do modelo com SVM e conjunto 1.

	Precisão	Sensibilidade
0	0,93	0,97
1	0,85	0,90
2	0,85	0,76

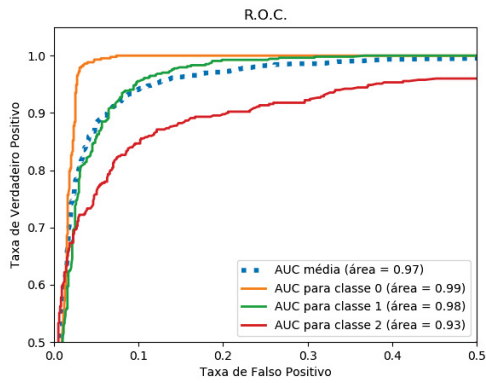


Figura 6.5: Curvas R.O.C. do modelo com RF e conjunto 1.

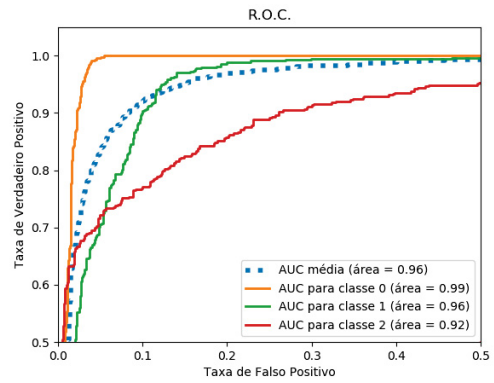


Figura 6.6: Curvas R.O.C. do modelo com SVM e conjunto 1.

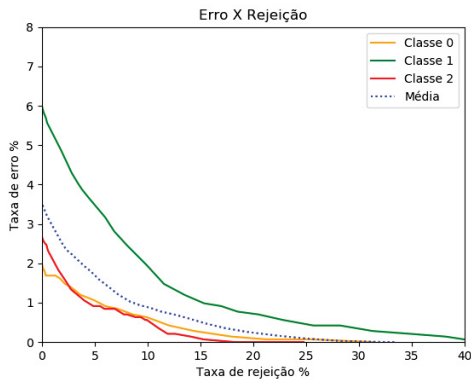


Figura 6.7: Gráfico de rejeição e erro do modelo com RF e conjunto 1.

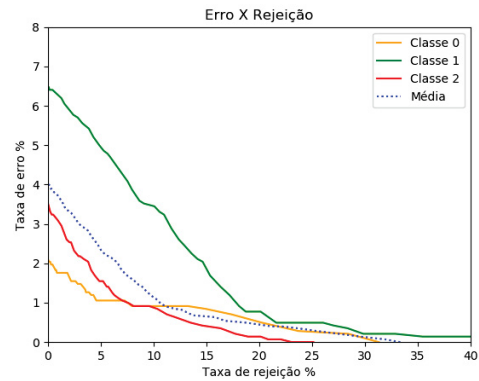


Figura 6.8: Gráfico de rejeição e erro do modelo com SVM e conjunto 1.

### 6.1.3 Conjunto de características 2

Assim como no conjunto 1, a melhor taxa de acurácia deste modelo é com o uso do algoritmo *random forest*, que supera os 90 pontos percentuais.

Os conjuntos de características 1 e 2 obtiveram resultados muito semelhantes, comprovando que a inclusão de características de textura, não modificam o desempenho do modelo. A diferença ocorre na classificação da classe do tipo 2 que, para este conjunto, o algoritmo de *random forest* tem desempenho superior ao SVM. O gráfico 6.12 apresenta uma inversão das curvas das classes 0 e 2. Para este modelo, a classe 2 tem alta taxa de erro para baixa rejeição, enquanto a classe 0 tem menor taxa de erro para a mesma taxa de rejeição. Essa tendência é invertida na altura dos 1,3 pontos percentuais de erro, onde as curvas referentes a estas duas classes se cruzam. Foram usados como parâmetros de entrada do RF  $n\_estimators = 63$ ;  $max\_depth = 50$ ;  $min\_samples\_leaf = 1$  e  $max\_features = "sqrt"$  e para o SVM  $C = 1000$ ;  $kernel = "rbf"$  e  $gamma = "0.01"$ .

Tabela 6.10: Matriz de confusão do modelo com RF e conjunto 2.

	0	1	2
0	0,99	0,0	0,01
1	0,0	0,95	0,05
2	0,07	0,16	0,77

Tabela 6.11: Matriz de confusão do modelo com SVM e conjunto 2.

	0	1	2
0	0,98	0,0	0,02
1	0,01	0,90	0,08
2	0,07	0,2	0,73

Tabela 6.12: Métricas de desempenho do modelo com RF e conjunto 2.

	Precisão	Sensibilidade
<b>0</b>	0,93	0,99
<b>1</b>	0,87	0,95
<b>2</b>	0,92	0,77

Tabela 6.13: Métricas de desempenho do modelo com SVM e conjunto 2.

	Precisão	Sensibilidade
<b>0</b>	0,92	0,98
<b>1</b>	0,84	0,90
<b>2</b>	0,86	0,73

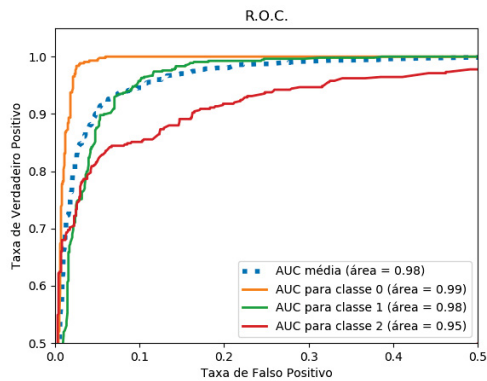


Figura 6.9: Curvas R.O.C. do modelo com RF e conjunto 2.

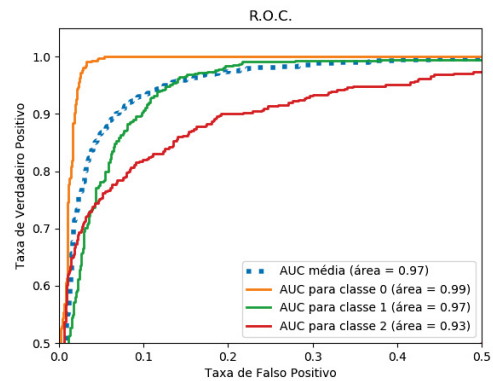


Figura 6.10: Curvas R.O.C. do modelo com SVM e conjunto 2.

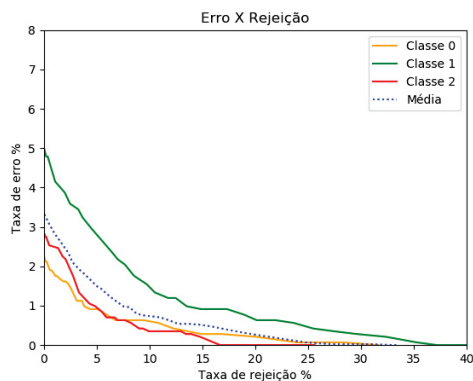


Figura 6.11: Gráfico de rejeição e erro do modelo com RF e conjunto 2.

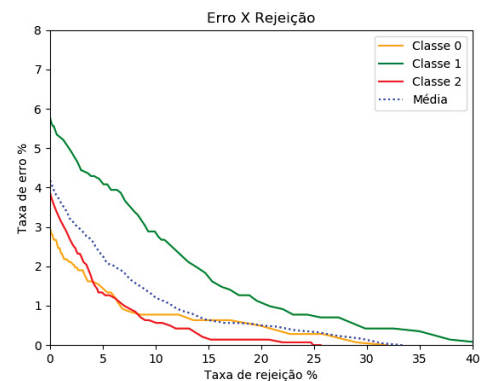


Figura 6.12: Gráfico de rejeição e erro do modelo com SVM e conjunto 2.

### 6.1.4 Conjunto de características 3

SVM é o algoritmo que melhor tirou proveito deste conjunto de características, obteve o melhor resultado dentre todos os modelos com acurácia superior a 94 pontos percentuais. A matriz de confusão, tabela 6.15, mostra que as classes dos tipos 0 e 1 tem boas taxas de precisão e não houve confusão entre elas. A classe 2 também tem a melhor taxa de distribuição dos erros, respectivamente 6 pontos percentuais e 9 pontos percentuais para as classes tipos 0 e 1. A qualidade do modelo é confirmada através de suas curvas ROC, que mostram uma AUC de 0,97, figura 6.14, e por suas taxas de sensibilidade (*Sensibilidade*), tabela 6.17. Em ambos os modelos, RF e SVM, novamente as curvas de rejeição e erro referentes as classes 0 e 2 se cruzam, em baixas taxas de erro, enquanto a curva da classe 1 apresenta elevada taxa de erro. Para uma rejeição de 5% das amostras, por exemplo, o erro apresentado pelo classe 1 supera os 3 pontos percentuais. Neste mesmo nível de rejeição, as demais classes apresentam erro menor que 1 ponto percentual, de acordo com os gráficos 6.15 e 6.16. Os parâmetros de entrada utilizados foram  $C = 1000$ ;  $kernel = "rbf"$  e  $gamma = "0.01"$ . O algoritmo *random forest* também obteve bons resultados. Sua alta taxa de sensibilidade na classe 0 é um dado muito relevante para o modelo, significa que quase a totalidade dos resultados negativos foram diagnosticados corretamente. Para o RF foram usados como parâmetros de entrada  $n\_estimators = 63$ ;  $max\_depth = 50$ ;  $min\_samples\_leaf = 2$  e  $max\_features = "sqrt"$ .

Tabela 6.14: Matriz de confusão do modelo com RF e conjunto 3.

	0	1	2
0	0,99	0,0	0,01
1	0,0	0,96	0,04
2	0,06	0,16	0,79

Tabela 6.16: Métricas de desempenho do modelo com RF e conjunto 3.

	Precisão	Sensibilidade
0	0,94	0,99
1	0,88	0,96
2	0,93	0,79

Tabela 6.15: Matriz de confusão do modelo com SVM e conjunto 3.

	0	1	2
0	0,99	0,0	0,01
1	0,0	0,98	0,02
2	0,06	0,09	0,85

Tabela 6.17: Métricas de desempenho do modelo com SVM e conjunto 3.

	Precisão	Sensibilidade
0	0,94	0,99
1	0,93	0,97
2	0,95	0,85

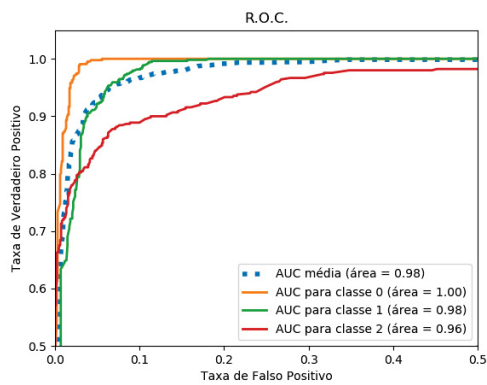


Figura 6.13: Curvas R.O.C. do modelo com RF e conjunto 3.

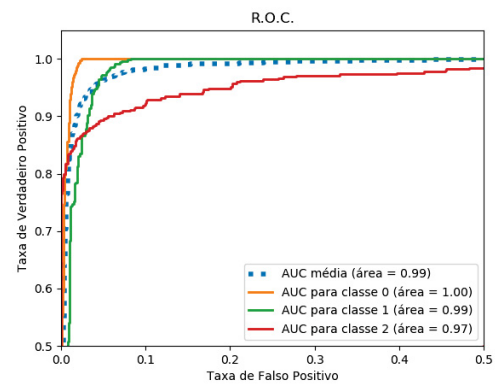


Figura 6.14: Curvas R.O.C. do modelo com SVM e conjunto 3.

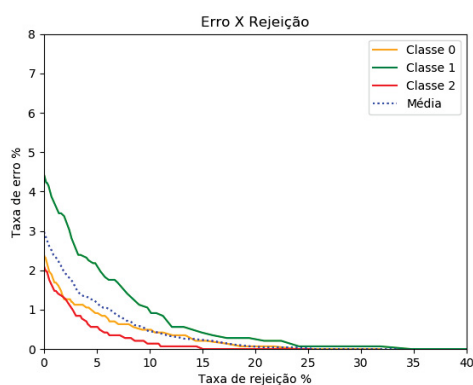


Figura 6.15: Gráfico de rejeição e erro do modelo com RF e conjunto 3.

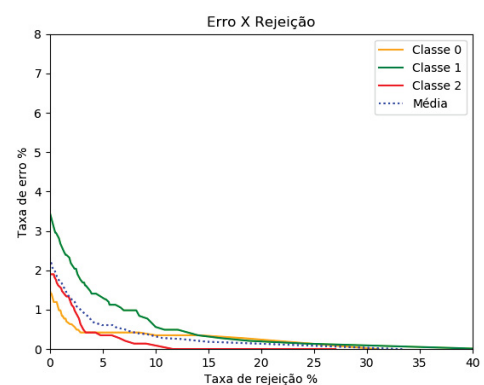


Figura 6.16: Gráfico de rejeição e erro do modelo com SVM e conjunto 3.

## 6.2 Base com dados aumentados

Para medir o impacto do uso de dados aumentados, este experimento utiliza o modelo de melhor desempenho nos experimentos anteriores, o modelo 6.1.4 usando SVM. Os dados aumentados formam duas bases distintas. Uma das bases é formada por imagens aumentadas para simular a variação de iluminação no ambiente, e a outra base formada por imagens aumentadas pela aplicação de filtros. Ambas as bases descritas no capítulo 4.

Inicialmente a base de imagens originais é dividida em base de treinamento e base de teste na proporção de 66,6% / 33,3%. Então, as imagens aumentadas são inseridas na base de treinamento. A base de teste não contém dados aumentados. Esse procedimento é repetido dez vezes e a média dos resultados é calculada.

Os resultados de acurácia estão expostos na tabela 6.18. Quando a primeira base aumentada (a base que simula a variação da iluminação) é inserida no treinamento, o modelo teve acurácia de 94,22 pontos percentuais e ficou praticamente estável em relação ao modelo de referência. As taxas de sensibilidade tiveram melhora de 0,1 ponto percentual nas classes 1 e 2, conforme tabelas 6.19 e 6.20. Considerando que o desvio padrão dos modelos é de 1 ponto percentual, é possível dizer que a utilização das bases de dados aumentados não acrescentou nem diminuiu a eficiência dos modelos.

Tabela 6.18: Acurácias do modelo utilizando as duas bases de dados aumentados. Ambas as bases são compostas por dados originais e aumentados.

Acurácia / desvio padrão	
base aumentada 1	94,22 ±0,5
base aumentada 2	93,52 ±0,5

Tabela 6.19: Matriz de confusão do modelo com SVM com base aumentada 1.

	0	1	2
0	0,99	0,0	0,01
1	0,0	0,97	0,03
2	0,06	0,08	0,86

Tabela 6.20: Matriz de confusão do modelo com SVM com base aumentada 2.

	0	1	2
0	0,99	0,0	0,01
1	0,0	0,97	0,03
2	0,06	0,9	0,85

Com a segunda base de dados aumentados, o modelo alcançou uma acurácia de 93,52 pontos percentuais. Valore abaixo do valor do modelo de referência.

### 6.3 Base separada por leitora.

Diferentes leitoras foram utilizadas para aquisição das imagens da base deste trabalho. As imagens produzidas por cada uma delas possuem diferenças, devido as condições de iluminação. Os modelos de classificação devem ser capazes de lidar com essas diferenças, uma vez que foram treinados com exemplos de imagens de todas as leitoras. Para testar esta hipótese, as imagens das três leitoras foram testadas separadamente. O modelo utilizado nos experimentos foi o de melhor desempenho desta metodologia, modelo 6.1.4 usando SVM. Os resultados são apresentados nas tabelas a seguir.

Tabela 6.21: Acurácias do modelo usando bases formadas com imagens de cada leitora.

acurácia	
leitora preta	95,98 ±1,4
leitora cinza	93,59 ±1,1
leitora verde	86,40 ±0,5

#### 6.3.1 Leitora preta

Tabela 6.22: Matriz de confusão do modelo com imagens da leitora preta.

	0	1	2
0	1,00	0,0	0,0
1	0,0	0,98	0,02
2	0,03	0,04	0,93

Tabela 6.23: Métricas de desempenho do modelo com imagens da leitora preta.

	<b>Precisão</b>	<b>Sensibilidade</b>
<b>0</b>	0,98	1,00
<b>1</b>	0,97	0,98
<b>2</b>	0,97	0,93

### 6.3.2 Leitora cinza

Tabela 6.24: Matriz de confusão do modelo com imagens da leitora cinza.

	<b>0</b>	<b>1</b>	<b>2</b>
<b>0</b>	0,98	0,0	0,02
<b>1</b>	0,0	0,98	0,02
<b>2</b>	0,05	0,12	0,89

Tabela 6.25: Métricas de desempenho do modelo com imagens da leitora cinza.

	<b>Precisão</b>	<b>Sensibilidade</b>
<b>0</b>	0,93	0,98
<b>1</b>	0,86	0,98
<b>2</b>	0,97	0,89

### 6.3.3 Leitora verde

Tabela 6.26: Matriz de confusão do modelo com imagens da leitora verde.

	<b>0</b>	<b>1</b>	<b>2</b>
<b>0</b>	0,95	0,0	0,05
<b>1</b>	0,0	0,95	0,05
<b>2</b>	0,11	0,2	0,69

Tabela 6.27: Métricas de desempenho do modelo com SVM e conjunto 3 da metodologia 2.

	<b>Precisão</b>	<b>Sensibilidade</b>
<b>0</b>	0,92	0,95
<b>1</b>	0,95	0,95
<b>2</b>	0,73	0,69

As imagens das leitoras preta e cinza tiveram resultados semelhantes. A base de imagens da leitora verde teve um desempenho inferior. Esta base não possui a mesma quantidade de amostras das outras leitoras, é 1/3 menor, fator que pode ter contribuído para o desempenho inferior desta base. Também, a taxa de sensibilidade na classe 2 teve muito impacto, enquanto as bases cinza e preta obtiveram valores próximos a 90 pontos percentuais, a base verde ficou próxima dos 70 pontos.

## 6.4 Combinação de classificadores

Cada conjunto de características também foi testado por meio da combinação dos algoritmos SVM, RF e de Regressão logística, e tiveram seus resultados calculados por meio de duas regras de decisão, voto majoritário e média das probabilidades. Os parâmetros utilizados em cada algoritmo e a proporção de divisão da base, foram os mesmos utilizados no conjunto, quando experimentado apenas com um algoritmo. Todos estão descritos nas seções 6.1.2, 6.1.3 e 6.1.4. Ainda, de acordo com o gráfico 6.19, é possível se obter taxas de erro menores que 1 ponto

percentual com taxas de rejeição inferiores a 7 pontos percentuais. E finalmente, todos os algoritmos utilizam os mesmos subconjuntos de treinamento e teste em cada rodada.

Tabela 6.28: Acurácias dos conjuntos de características usando uma combinação de classificadores por meio de voto majoritário da média de probabilidades.

	voto majoritário	média de probabilidades
Conjunto 1	87,29 $\pm$ 0,6	88,35 $\pm$ 0,3
Conjunto 2	86,98 $\pm$ 0,5	88,26 $\pm$ 0,7
Conjunto 3	94,40 $\pm$ 0,9	93,84 $\pm$ 0,5

O conjunto 3 obteve uma acurácia maior em relação do que quando é usado um único classificador, mas considerando a margem do desvio padrão, pode-se dizer que o desempenho é equivalente.

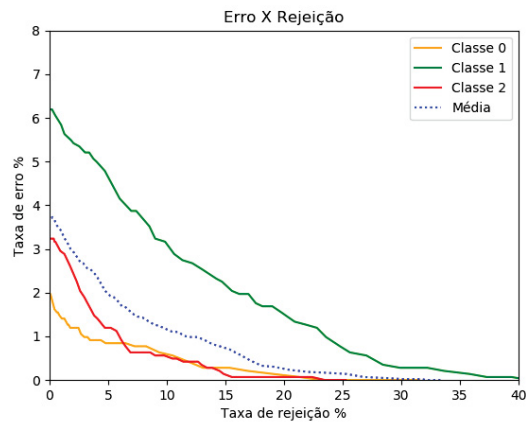


Figura 6.17: Gráfico de rejeição e erro do modelo combinado e conjunto 1.

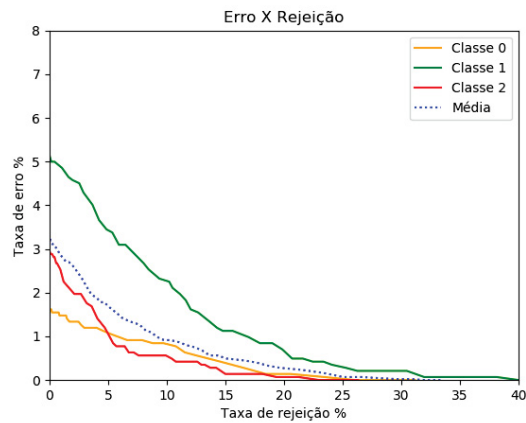


Figura 6.18: Gráfico de rejeição e erro do modelo combinado e conjunto 2.

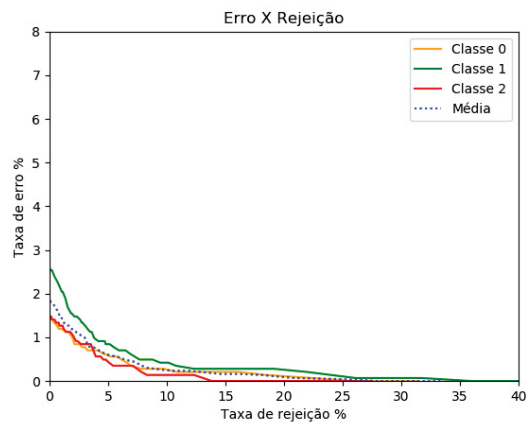


Figura 6.19: Gráfico de rejeição e erro do modelo combinado e conjunto 3.

## 6.5 Combinação de classificadores em dois níveis

Este modelo utiliza dois níveis de classificação, em cascata. O primeiro nível faz a classificação das instâncias positivas e negativas, por meio do modelo 6.1.4 modificado, que inclui um limiar de aceitação. Todas as imagens classificadas como 0 e 1 devem ter no mínimo 85% de probabilidade. Todas as instâncias classificadas

como indefinidas (classe 2) no primeiro nível passam para o próximo nível. O segundo nível de classificação utiliza a CNN apresentada na seção 6.6, que tem a melhor taxa de acerto da classe 2 dentre os modelos experimentados.

A base de treinamento do primeiro nível é composta por 2.883 imagens devidamente balanceada entre as 3 classes, enquanto sua base de teste é composta por 1.421 imagens e também esta balanceada. As tabelas 6.29 e 6.30 mostram os resultados do primeiro nível do modelo. Não houve confusão entre as classes 0 e 1, mas 27% dessas classes foram confundidas com a classe 2, desta maneira estas instâncias seguirão para o segundo nível, juntamente com os 94% das imagens classificadas indefinidas. Os 6% restantes da classe 2 foram confundidos com as classes 0 e 1 e são contabilizados como erro.

Tabela 6.29: Matriz de confusão do primeiro nível do modelo em cascata, com o algoritmo SVM.

	0	1	2
0	0,93	0,0	0,07
1	0,0	0,80	0,20
2	0,03	0,03	0,94

Tabela 6.30: Métricas de desempenho do primeiro nível do modelo em cascata.

	Precisão	Sensibilidade
0	0,97	0,93
1	0,97	0,80
2	0,77	0,94

36,8% das imagens da base de testes passaram para o segundo nível. Deste total, 77% pertencem a classe 2, e o restante às classes 0 e 1. Todas as imagens negativas foram corretamente classificadas pela CNN, assim como 93% das imagens positivas também. A taxa de sensibilidade da classe 2 ficou em 92% pontos percentuais. A base de treinamento do segundo nível é composta por 4.227 imagens com um número maior de imagens da classe 2.

Tabela 6.31: Matriz de confusão do segundo nível do modelo em cascata, utilizando CNN.

	0	1	2
0	1,0	0,0	0,0
1	0,0	0,93	0,06
2	0,02	0,06	0,92

Tabela 6.32: Métricas de desempenho do segundo nível do modelo em cascata.

	Precisão	Sensibilidade
0	0,74	0,92
1	0,78	0,93
2	0,98	1,00

A tabela 6.33 resume as métricas do modelo como um todo. Conforme esperado, as classes 0 e 1 tiveram altas taxas de reconhecimento, enquanto a classe 2 teve um desempenho de 87 pontos percentuais. A taxa de acurácia total do modelo em cascata ficou em **95,3 pontos percentuais**.

Tabela 6.33: Taxas de precisão e sensibilidade finais do modelo em cascata, cuja acurácia final ficou em 95,3 pontos percentuais.

	Precisão	Sensibilidade
0	0,71	1,00
1	0,75	0,99
2	0,75	0,87

## 6.6 Resultados da metodologia 2

Esta metodologia utiliza uma CNN para processar e classificar as imagens. Foram feitos dois experimentos, no primeiro, foi utilizado apenas a base de imagens originais que, após o balanceamento, ficou com 1.295 amostras por classe. Para o segundo experimento, a base de dados aumentados foi adicionada junto a base original, totalizando 6.485 imagens por classe. Conforme descrito no capítulo 4, há duas bases de dados aumentados mas, neste experimento, apenas a base que simula as condições de iluminação foi utilizada. Para o retreinamento da rede, a base foi dividida nas seguintes proporções: 10% para teste, 10% para validação e o restante para treinamento. Os parâmetros de otimização foram  $training-steps = 50.000$  e  $learning-rate = 0.5$ . A camada de classificação da rede é composta por uma camada *fully connected* com 2046 neurônios e camada de saída (*softmax*), que gera uma resposta probabilística para as 3 classes do problema. Para computar os resultados, cada procedimento foi repetido dez vezes, sempre utilizando amostragem diferente da base. O valor médio entre as repetições representa o desempenho do modelo. A tabela 6.1 resume os resultados das acurácias obtidas nos experimentos desta metodologia.

Conforme mostra a tabela 6.34, a acurácia do modelo que usa a base de imagens originais é ligeiramente superior ao modelo que usa a base de dados aumentados. Ao se analisar as tabelas abaixo, interessante notar que, com a base original, a sensibilidade da classe 2 alcançou 93 pontos percentuais, o maior valor dentre todos os modelos. Quando o experimento é repetido, mas com a base de dados aumentados, esse valor cai para 88 pontos percentuais, queda que é compensada pela melhora nos valores de sensibilidade das classes 0 e 1, com ganhos de 0,2 e 0,1 pontos, respectivamente. As curvas de rejeição deste modelo apresentam uma ordem diferente do que foi apresentado até agora, a curva da classe 0 com alta taxa de erro, comparado as curvas das classes 1 e 2, conforme gráfico 6.22.

Tabela 6.34: Resumo dos resultados da metodologia 2 com respectivos valores de desvio padrão.

Acurácia / desvio padrão	
Base original	93,6 ±0,5
Base aumentada	93,2 ±0,2

Tabela 6.35: Matriz de confusão do modelo utilizando a base de imagens originais.

	0	1	2
0	0,93	0,02	0,04
1	0,0	0,95	0,04
2	0,0	0,07	0,93

Tabela 6.36: Matriz de confusão do modelo utilizando a base de imagens aumentadas.

	0	1	2
0	0,95	0,01	0,04
1	0,0	0,96	0,04
2	0,06	0,05	0,88

Tabela 6.37: Métricas de desempenho do modelo utilizando a base de imagens originais.

	Precisão	Sensibilidade
0	0,96	0,93
1	0,98	0,95
2	0,86	0,93

Tabela 6.38: Métricas de desempenho do modelo utilizando a base de imagens aumentadas.

	Precisão	Sensibilidade
0	0,94	0,95
1	0,94	0,96
2	0,92	0,88

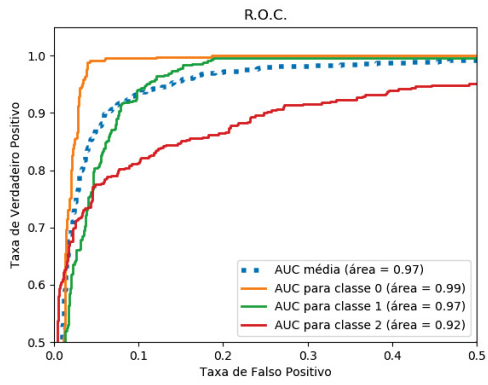


Figura 6.20: Curvas R.O.C. do modelo utilizando base de imagens originais.

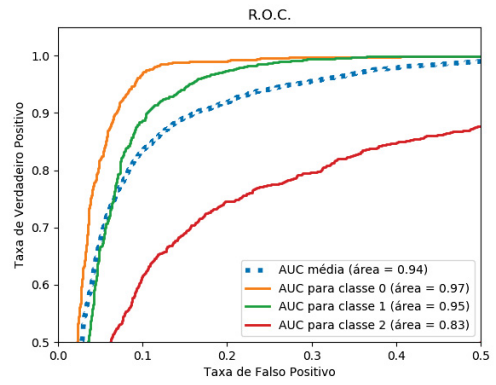


Figura 6.21: Curvas R.O.C. do modelo utilizando base de imagens Aumentadas.

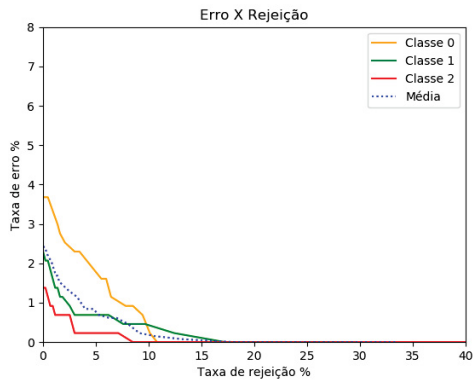


Figura 6.22: Gráfico de rejeição e erro do modelo utilizando a metodologia 2 com as imagens originais.

## 7 Conclusão

Inicialmente, foi constatado que os modelos propostos, baseados em aprendizado de máquina, tem desempenho superior em relação ao modelo tradicional, das leitoras, cuja acurácia é de 74 pontos percentuais para a base de imagens deste trabalho. Em todos os experimentos realizados, a taxa de acurácia supera em, pelo menos, 12 pontos percentuais a taxa de classificação das leitoras.

Depois, foi verificado que o conjunto de características originais, extraídos pelo algoritmo das leitoras, tem boa representatividade do problema. Isso foi demonstrado quando este conjunto foi submetido aos algoritmos RF e SVM. A acurácia saltou dos 74 pontos percentuais, para mais de 92 pontos percentuais. Porém, uma vez que este conjunto não pode ser replicado, seu desempenho serve apenas como parâmetro para o conjunto 1, criado para simular o conjunto de características originais.

O conjunto 1, então, se mostrou bem alinhado ao conjunto original. O comparativo pode ser visto na tabela 7.1. O algoritmo RF junto do conjunto 1 apresentou uma acurácia de 90,66 pontos percentuais, diferença de apenas 1,75 pontos de acurácia.

Tabela 7.1: Comparativo entre o conjunto de características originais e as implementadas para simular as mesmas.

	Random Forest	SVM	Algoritmo leitora
Características originais	92,39	90,97	74,0
Conjunto 1	<b>90,66</b>	87,54	-

Outros modelos, com diferentes configurações, também foram testados. A escolha do melhor deles nada mais é que o de melhor acurácia de predições em equilíbrio com a sensibilidade na predição das classes. O modelo da metodologia 1 usando SVM combinado ao conjunto representativo 3 obteve mais de 94 pontos percentuais de acurácia e métricas bem equilibradas. A tabela 7.2 mostra que a média das acurácias do algoritmo *random forest* é superior, apesar do SVM ter conseguido a melhor taxa de acurácia individual.

Tabela 7.2: Comparativo de acurácia dos algoritmos de classificação entre os modelos.

	Random Forest	SVM
modelo 01	90,66	87,54
modelo 02	90,28	86,98
modelo 03	91,34	94,17
<b>média</b>	<b>90,76</b>	<b>89,56</b>

O uso de uma CNN para extração de características se mostrou eficiente em todos os algoritmos em ambas as metodologias. O conjunto adquirido usando *transfer learning* da CNN, na metodologia 1, teve bons resultados com os algoritmos RF e SVM, com taxas de 91,34 e 94,17 pontos percentuais, respectivamente. Quando aplicada a metodologia 2, a acurácia de classificação se manteve no mesmo nível, 93,6 pontos percentuais.

As bases de imagens aumentadas foram experimentadas utilizando o modelo que obteve o melhor desempenho com o conjuntos das originais. Como resultado, tanto na metodologia 1 como na metodologia 2, usando CNN, as taxas de acurácia praticamente se mantiveram estáveis, considerando os valores de desvio padrão. A base de imagens originais possui quantidade e diversidade o suficiente para representar o problema, sem que o aumento de dados seja necessário.

A combinação de classificadores, em muitos casos, eleva a acurácia e a precisão do modelo, balanceando erros dos classificadores individuais. Para verificar essa hipótese, este experimento foi conduzido e se mostrou viável. Por meio da regra de voto majoritário alcançou 94,4 pontos percentuais de acurácia, superando em 0,3

pontos o modelo de melhor desempenho individual, mas empatado dentro da margem de erro. A combinação de classificadores em cascata se mostrou ainda mais eficiente, alcançando 95,3 pontos percentuais de acurácia.

Quando separadas por leitora, a classificação das imagens indica que existe uma diferença de desempenho entre elas. Imagens capturadas pela leitora cinza tiveram melhor o desempenho, e as imagens da leitora verde, o pior. Nesse sentido, uma comparação com o desempenho do modelo treinado com imagens de todas as leitoras, mostra que as variações de iluminação são minimizadas quando inserimos maior variabilidade de amostras.

E finalmente, uma análise dos erros indica uma possível margem para erros de rotulação. Uma base de aprendizado sempre esta suscetível a erros de rotulação, em um grau maior ou menor, dependendo da experiência do profissional. As figuras abaixo são erros de classificação do modelo 2 da metodologia 1. O ponto dessa questão é que talvez hajam instâncias onde a classificação foi correta, mas sua rotulação esta errada, diminuindo a real acurácia do modelo. Como exemplo, as tiras abaixo foram rotuladas indefinidas, mas possuem fortes características de tiras negativas.



Figura 7.1: Exemplos de tiras indefinidas classificadas como negativas.

Os gráficos CER, que mostram a taxa erro em relação a rejeição, são úteis pois permitem observar o custo (rejeição) de se obter um sistema com uma determinada taxa de erro. Modelos com taxas de precisão e acurácia semelhantes apresentam diferentes comportamentos dentro do gráfico CER. Em geral, modelos que classificam seus erros com alta taxa de probabilidade, tendem a ser menos flexíveis a taxa de erro. Ou seja, exigem alto custo de rejeição em relação a diminuição da taxa de erro.

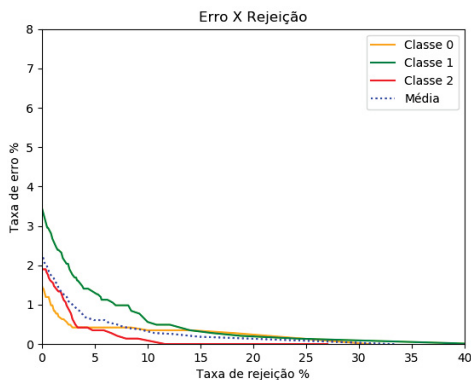


Figura 7.2: Gráfico de rejeição e erro do modelo com SVM e conjunto 3.

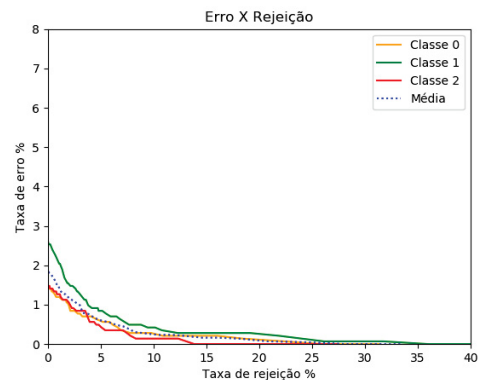


Figura 7.3: Gráfico de rejeição e erro do modelo com classificadores combinados e conjunto 3.

Os gráficos 7.2 e 7.3 pertencem a modelos que obtiveram resultados muito semelhantes. Um modelo utiliza SVM e outro classificação combinada e ambos utilizando o conjunto de características 3. Nesse caso as curvas de erro podem servir como critério de escolha do classificador. Para este trabalho, cada classe apresenta uma curva distinta dentro do mesmo modelo, indicando que é possível ajustar limiares diferentes para cada classe, e assim, otimizar a taxa de erros do sistema.

## REFERÊNCIAS

- [Asuncion e Newman, 2007] Asuncion, A. e Newman, D. J. (2007). Uci machine learning repository [http://www.ics.uci.edu/~mllearn/mlrepository.html]. irvine, ca: University of california. *School of Information and Computer Science*, 12.
- [Aveyard et al., 2007] Aveyard, J., Mehrabi, M., Cossins, A., Braven, H. e Wilson, R. (2007). One step visual detection of pcr products with gold nanoparticles and a nucleic acid lateral flow (nalf) device. *Chemical Communications*, (41):4251–4253.
- [Barbosa et al., 2015] Barbosa, A. I., Gehlot, P., Sidapra, K., Edwards, A. D. e Reis, N. M. (2015). Portable smartphone quantitation of prostate specific antigen (psa) in a fluoropolymer microfluidic device. *Biosensors and Bioelectronics*, 70:5–14.
- [Berg et al., 2015] Berg, B., Cortazar, B., Tseng, D., Ozkan, H., Feng, S., Wei, Q., Chan, R. Y.-L., Burbano, J., Farooqui, Q., Lewinski, M., Di Carlo, D., Garner, O. B. e Ozcan, A. (2015). Cellphone-based hand-held microplate reader for point-of-care testing of enzyme-linked immunosorbent assays. *ACS Nano*, 9(8):7857–7866.
- [Bishop, 2006] Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.
- [Breiman, 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.
- [Brun et al., 2018] Brun, A. L., Britto, A. S., Oliveira, L. S., Enembreck, F. e Sabourin, R. (2018). A framework for dynamic classifier selection oriented by the classification problem difficulty. *Pattern Recognition*, 76:175 – 190.
- [Carrio et al., 2015] Carrio, A., Sampedro, C., Sanchez-Lopez, J. L., Pimienta, M. e Campoy, P. (2015). Automated low-cost smartphone-based lateral flow saliva test reader for drugs-of-abuse detection. *Sensors*, 15(11):29569–29593.
- [Ching et al., 2015] Ching, K. H., He, X., Stanker, L. H., Lin, A. V., McGarvey, J. A. e Hnasko, R. (2015). Detection of shiga toxins by lateral flow assay. *Toxins*, 7(4):1163–1173.
- [Chun, 2009] Chun, P. (2009). *Colloidal Gold and Other Labels for Lateral Flow Immunoassays*, páginas 1–19. Humana Press, Totowa, NJ.
- [Contreras-Naranjo et al., 2016] Contreras-Naranjo, J. C., Wei, Q. e Ozcan, A. (2016). Mobile phone-based microscopy, sensing, and diagnostics. *IEEE Journal of Selected Topics in Quantum Electronics*, 22(3):1–14.
- [Copstead e Banasik, 2000] Copstead, L.-E. e Banasik, J. (2000). *Pathophysiology: biological and behavior perspectives*. USA: WB Saunders Company.
- [Danscher, 1984] Danscher, G. (1984). Autometallography. *Histochemistry*, 81(4):331–335.
- [Dietterich, 2000] Dietterich, T. G. (2000). Ensemble methods in machine learning. Em *International workshop on multiple classifier systems*, páginas 1–15. Springer.
- [Duarte, 2017] Duarte, M. (2017). Peak detection in python. [https://github.com/demotu/BMC/blob/master/functions/detect\\_peaks.py](https://github.com/demotu/BMC/blob/master/functions/detect_peaks.py). Acessado em 12/06/2018.
- [Eilers e Boelens, 2005] Eilers, P. H. e Boelens, H. F. (2005). Baseline correction with asymmetric least squares smoothing. *Leiden University Medical Centre Report*, 1(1):5.
- [Erickson et al., 2017] Erickson, B. J., Korfiatis, P., Akkus, Z. e Kline, T. L. (2017). Machine learning for medical imaging. *Radiographics*, 37(2):505–515.
- [Fawcett, 2006] Fawcett, T. (2006). An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874.
- [Gui et al., 2014] Gui, C., Wang, K., Li, C., Dai, X. e Cui, D. (2014). A ccd-based reader combined with cds quantum dot-labeled lateral flow strips for ultrasensitive quantitative detection of caga. *Nanoscale research letters*, 9(1):57.

- [Hearst et al., 1998] Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J. e Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28.
- [Holgate et al., 1983] Holgate, C. S., Jackson, P., Cowen, P. N. e Bird, C. C. (1983). Immunogold-silver staining: new method of immunostaining with enhanced sensitivity. *Journal of Histochemistry & Cytochemistry*, 31(7):938–944.
- [Jain, 1989] Jain, A. K. (1989). *Fundamentals of digital image processing*. Englewood Cliffs, NJ: Prentice Hall,.
- [Karakus e Salih, 2013] Karakus, C. e Salih, B. A. (2013). Comparison of the lateral flow immunoassays (Ifia) for the diagnosis of helicobacter pylori infection. *Journal of immunological methods*, 396 1-2:8–14.
- [Kim et al., 2017] Kim, H., Awofeso, O., Choi, S., Jung, Y. e Bae, E. (2017). Colorimetric analysis of saliva–alcohol test strips by smartphone-based instruments using machine-learning algorithms. *Applied Optics*, 56(1):84–92.
- [Kittler et al., 1998] Kittler, J., Hatef, M., Duin, R. P. e Matas, J. (1998). On combining classifiers. *IEEE transactions on pattern analysis and machine intelligence*, 20(3):226–239.
- [Koczula e Gallotta, 2016] Koczula, K. M. e Gallotta, A. (2016). Lateral flow assays. *Essays in biochemistry*, 60(1):111–120.
- [Kodituwakku e Selvarajah, 2004] Kodituwakku, S. e Selvarajah, S. (2004). Comparison of color features for image retrieval. *Indian Journal of Computer Science and Engineering*, 1(3):207–211.
- [Kuang et al., 2013] Kuang, H., Xing, C., Hao, C., Liu, L., Wang, L. e Xu, C. (2013). Rapid and highly sensitive detection of lead ions in drinking water based on a strip immunosensor. *Sensors*, 13(4):4214–4224.
- [Leuvering et al., 1981] Leuvering, J. H., Thal, P. J., Van der Waart, M. e Schuurs, A. H. (1981). A sol particle agglutination assay for human chorionic gonadotrophin. *Journal of immunological methods*, 45(2):183–194.
- [Luo et al., 2016] Luo, H., Huang, Y., Lai, K., Rasco, B. A. e Fan, Y. (2016). Surface-enhanced raman spectroscopy coupled with gold nanoparticles for rapid detection of phosmet and thiabendazole residues in apples. *Food Control*, 68:229 – 235.
- [Maini e Aggarwal, 2009] Maini, R. e Aggarwal, H. (2009). Study and comparison of various image edge detection techniques. *International journal of image processing (IJIP)*, 3(1):1–11.
- [Mei et al., 2013] Mei, Z., Qu, W., Deng, Y., Chu, H., Cao, J., Xue, F., Zheng, L., El-Nezamic, H. S., Wu, Y. e Chen, W. (2013). One-step signal amplified lateral flow strip biosensor for ultrasensitive and on-site detection of bisphenol a (bpa) in aqueous samples. *Biosensors and Bioelectronics*, 49:457–461.
- [Mirasoli et al., 2012] Mirasoli, M., Buragina, A., Dolci, L. S., Simoni, P., Anfossi, L., Giraudi, G. e Roda, A. (2012). Chemiluminescence-based biosensor for fumonisins quantitative detection in maize samples. *Biosensors and Bioelectronics*, 32(1):283–287.
- [Mudanyali et al., 2012] Mudanyali, O., Dimitrov, S., Sikora, U., Padmanabhan, S., Navruz, I. e Ozcan, A. (2012). Integrated rapid-diagnostic-test reader platform on a cellphone. *Lab Chip*, 12:2678–2686.
- [Ngom et al., 2010] Ngom, B., Guo, Y., Wang, X. e Bi, D. (2010). Development and application of lateral flow test strip technology for detection of infectious agents and chemical contaminants: a review. *Analytical and bioanalytical chemistry*, 397(3):1113–1135.
- [Nielsen et al., 2009] Nielsen, K., Yu, W., Kelly, L., Williams, J., Dajer, A., Gutierrez, E., Cruz, G. R., Renteria, T., Bermudez, R. e Algire, J. (2009). Validation and field assessment of a rapid lateral flow assay for detection of bovine antibody to anaplasma marginale. *Journal of Immunoassay and Immunochemistry*, 30(3):313–321.
- [Oliveira et al., 2002] Oliveira, L. S., Sabourin, R., Bortolozzi, F. e Suen, C. Y. (2002). Automatic recognition of handwritten numerical strings: A recognition and verification strategy. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (11):1438–1454.
- [Omidfar et al., 2011] Omidfar, K., Kia, S. e Larijani, B. (2011). Development of a colloidal gold-based immunochromatographic test strip for screening of microalbuminuria. 30:117–24.
- [Otsu, 1979] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66.

- [O'Farrell, 2009] O'Farrell, B. (2009). Evolution in lateral flow–based immunoassay systems. Em *Lateral flow immunoassay*, páginas 1–33. Springer.
- [Pal, 2008] Pal, M. (2008). Multiclass approaches for support vector machine based land cover classification. *arXiv preprint arXiv:0802.2411*.
- [Panhotra et al., 2005] Panhotra, B., Hassan, Z., Joshi, C. e Bahrani, A. (2005). Visual detection of multiple viral amplicons by dipstick assay: its application in screening of blood donors a welcome tool for the limited resource settings. *Journal of clinical microbiology*, 43(12):6218–6219.
- [Posthuma-Trumpie et al., 2009] Posthuma-Trumpie, G. A., Korf, J. e van Amerongen, A. (2009). Lateral flow (immuno)assay: its strengths, weaknesses, opportunities and threats. a literature survey. *Analytical and Bioanalytical Chemistry*, 393(2):569–582.
- [Russakovsky et al., 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C. e Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.
- [Savitzky e Golay, 1964] Savitzky, A. e Golay, M. J. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639.
- [scikit image, 2018] scikit image (2018). cameramanimage. [http://tonysyu.github.io/scikit-image/\\_images/plot\\_otsu\\_1.png](http://tonysyu.github.io/scikit-image/_images/plot_otsu_1.png). Acessado em 30/08/2018.
- [Sokolova e Lapalme, 2009] Sokolova, M. e Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437.
- [Solmaz et al., 2018] Solmaz, M. E., Mutlu, A. Y., Alankus, G., Kılıç, V., Bayram, A. e Horzum, N. (2018). Quantifying colorimetric tests using a smartphone app based on machine learning classifiers. *Sensors and Actuators B: Chemical*, 255:1967–1973.
- [SoRelle et al., 2016] SoRelle, E. D., Liba, O., Campbell, J. L., Dalal, R., Zavaleta, C. L. e de la Zerda, A. (2016). A hyperspectral method to assay the microphysiological fates of nanomaterials in histological samples. *Elife*, 5:e16352.
- [Szegedy et al., 2016] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. e Wojna, Z. (2016). Rethinking the inception architecture for computer vision. Em *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, páginas 2818–2826.
- [Tuceryan et al., 1993] Tuceryan, M., Jain, A. K. et al. (1993). Texture analysis. *Handbook of pattern recognition and computer vision*, 2:235–276.
- [Wang e Perez, 2017] Wang, J. e Perez, L. (2017). The effectiveness of data augmentation in image classification using deep learning.
- [Yang et al., 2008] Yang, M., Kostov, Y., Bruck, H. A. e Rasooly, A. (2008). Carbon nanotubes with enhanced chemiluminescence immunoassay for ccd-based detection of staphylococcal enterotoxin b in food. *Analytical chemistry*, 80(22):8532–8537.
- [Zangheri et al., 2015] Zangheri, M., Cevenini, L., Anfossi, L., Baggiani, C., Simoni, P., Di Nardo, F. e Roda, A. (2015). A simple and compact smartphone accessory for quantitative chemiluminescence-based lateral flow immunoassay for salivary cortisol detection. *Biosensors and Bioelectronics*, 64:63–68.
- [Zeng et al., 2016] Zeng, N., Wang, Z., Zhang, H., Liu, W. e Alsaadi, F. E. (2016). Deep belief networks for quantitative analysis of a gold immunochromatographic strip. *Cognitive Computation*, 8(4):684–692.
- [Zhang et al., 2010] Zhang, X., Yang, H. e Wang, K. (2010). Image processing for the ccd based lateral flow strip detector. *Nano biomedicine engineering*, 2(4):214–217.
- [Zhao et al., 2012] Zhao, Y., Huang, D.-S. e Jia, W. (2012). Completed local binary count for rotation invariant texture classification. *IEEE transactions on image processing*, 21(10):4492–4497.
- [Zheng et al., 2016] Zheng, Y., Wang, K., Zhang, J., Qin, W., Yan, X., Shen, G., Gao, G., Pan, F. e Cui, D. (2016). Simultaneous quantitative detection of helicobacter pylori based on a rapid and sensitive testing platform using quantum dots-labeled immunochromatographic test strips. *Nanoscale research letters*, 11(1):62.