

UNIVERSIDADE FEDERAL DE SANTA CATARINA

Felipe Vieira Roque

**DETECÇÃO DE DEFEITOS EM TECIDOS UTILIZANDO
ALGORITMOS DE CLASSIFICAÇÃO E WAVELETS.**

Araranguá

2017

Felipe Vieira Roque

**DETECÇÃO DE DEFEITOS EM TECIDOS UTILIZANDO
ALGORITMOS DE CLASSIFICAÇÃO E WAVELETS.**

Trabalho de conclusão de Curso submetido à Universidade Federal de Santa Catarina para a obtenção do Grau de Bacharel em Engenharia de Computação.
Orientador: Prof. Tiago Oliveira Weber, Dr.

Araranguá

2017

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Roque, Felipe Vieira

Detecção de defeitos em tecidos utilizando
algoritmos de classificação e Wavelets / Felipe
Vieira Roque ; orientador, Tiago Oliveira Weber,
2017.

89 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Campus
Araranguá, Graduação em Engenharia de Computação,
Araranguá, 2017.

Inclui referências.

1. Engenharia de Computação. 2. Aprendizado de
máquina. 3. Visão Computacional. 4. Wavelets. I.
Weber, Tiago Oliveira. II. Universidade Federal de
Santa Catarina. Graduação em Engenharia de
Computação. III. Título.

Felipe Vieira Roque

**DETECCÃO DE DEFEITOS EM TECIDOS UTILIZANDO
ALGORITMOS DE CLASSIFICAÇÃO E WAVELETS**

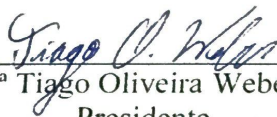
Este Trabalho de Conclusão de Curso foi julgado aprovado para obtenção do Título de “Bacharel em Engenharia de Computação” e aprovado em sua forma final pela Universidade Federal de Santa Catarina

Araranguá, 06 de Dezembro de 2017.

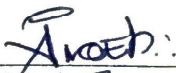


Prof.^a Eliane Pozzebon, Dr.^a
Coordenadora do Curso

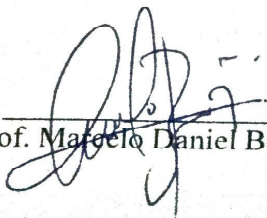
Banca Examinadora:



Prof.^a Tiago Oliveira Weber, Dr.
Presidente



Prof.^a Angela Maria Ruasne da Silva Macedo, M.^a



Prof. Marcelo Daniel Berejuck, Dr.

Este trabalho é dedicado a todos aqueles que de alguma forma, direta ou indiretamente, colaboraram para sua execução.

AGRADECIMENTOS

À minha família, que durante todo o período do curso deu todo o apoio necessário para que o mesmo fosse possível. À Universidade Federal de Santa Catarina, pelas oportunidades que me foram concedidas. Ao professor e orientador Tiago de Oliveira Weber, por todo o apoio, conhecimentos compartilhados, paciência e disponibilidade me oferecidos nesse período, ainda um exemplo de profissional e pessoa a ser seguido. À todos os professores da universidade pelos conhecimentos me passados durante o curso. À todos meus amigos que durante toda a realização desse trabalho me deram apoio e ajuda em diversos aspectos do trabalho, em especial, Roger, Luiz Antonio, Luan e Yuri. À minha namorada, Jaqueline, pelo apoio, motivação e compreensão nesse período de execução e finalização do trabalho. Aos demais colegas e amigos que estiveram presentes nesse aprendizado constante ao longo dos anos na universidade.

Não há fatos eternos, como não há verdades absolutas.

(Friedrich Nietzsche)

RESUMO

O uso da inteligência artificial em processos que antes eram feitos por seres humanos está se tornando cada vez mais comum. Algoritmos de aprendizagem de máquina têm a capacidade de reconhecer padrões e assim tomar decisões a partir de eventos passados. Um dos processos que podem ser automatizados é a detecção de defeitos na indústria têxtil. O controle de qualidade é um dos processos mais importantes numa indústria, com ele pode-se agilizar a produção e ainda reduzir custos. Encontrar as características que representam os problemas muitas vezes pode ser difícil. No presente trabalho, foi utilizada a abordagem espectral para extração de características. A transformada de *Wavelet* é uma técnica poderosa na análise de imagens, bem como na análise de sinais em geral. Este trabalho apresenta um sistema de visão computacional que possa auxiliar na tomada de decisão na indústria têxtil. O trabalho ainda aborda diferentes técnicas de classificação de modo a identificar o melhor classificador. Para obtenção dos parâmetros dos classificadores, foi utilizado o otimizador de Bayes. A acurácia média encontrada no melhor classificador, a técnica dos vizinhos mais próximos, foi de cerca de 91%.

Palavras-chave: Aprendizado de máquina. Visão Computacional. Wavelets.

ABSTRACT

Artificial Intelligence is becoming more common in processes that were once only made by humans. Machine learning algorithms can recognize patterns and make decisions based on past events. One of the processes that can be automated is the detection of defects in the textile industry. Quality control is one of the most critical processes in industry, improving it can increase the production and reduce costs. Finding the features that represent the problems can often be difficult. In this work, the spectral approach was used to extract characteristics. The Wavelet transform is a powerful technique in image analysis as well as in signal analysis in general. This work presents a computer vision system that can help in decision-making in the textile industry. The work also addresses different classification techniques to identify the best classifier. Bayes Optimization was used to obtain the classifier parameters. The average accuracy for the best classifier, the nearest neighbor technique, was 91%.

Keywords: Machine Learning. Computer Vision. Wavelets.

LISTA DE FIGURAS

Figura 1	Especialista fazendo a inspeção. (MALEK, 2012)	22
Figura 2	Hierarquia de técnicas de aprendizado de máquina.	32
Figura 3	Etapas de um sistema classificação com aquisição e extração de características.	34
Figura 4	Estrutura geral para a generalização de um modelo e predição	34
Figura 5	Árvore de Decisão.	39
Figura 6	SVM linear. (RAJPOOT; RAJPOOT, 2004)	40
Figura 7	Esquema em cascata da Transformada Discreta de Wavelet com banco de filtros.	50
Figura 8	Etapas da montagem da base de dados até o classificador.	52
Figura 9	Imagem de controle inicial testada	53
Figura 10	Imagem Cortada	54
Figura 11	Esquema da transformada de <i>Wavelet</i> de duas dimensões com banco de filtros.	55
Figura 12	Saída da transformada.	56
Figura 13	Otimização do KNN.	60
Figura 14	Matriz de confusão para o KNN.	62
Figura 15	Otimização do Bayes Ingênuo.	64
Figura 16	Matriz de confusão para o classificador de Bayes.	65
Figura 17	Otimização do SVM.	65
Figura 18	Matriz de confusão para o SVM.	67
Figura 19	Otimização da Árvore de Decisão.	68
Figura 20	Matriz de confusão para a Árvore de Decisão.	69
Figura 21	Acurácia média dos classificadores.	70
Figura 22	Comparação das taxas de verdadeiros positivos para cada classe.	71
Figura 23	Comparação das taxas de classificação correta de cada classe.	72
Figura 24	Taxas do KNN.	83
Figura 25	Tipos de defeito.	87

LISTA DE TABELAS

Tabela 1	Características Gerais dos Classificadores. Adaptada da página online do Matlab.	41
Tabela 2	Porcentagens e quantidades dos tipos de defeitos. Tabela elaborada pelo autor.	59
Tabela 3	Parâmetros encontrados pelo Otimizador.	61
Tabela 4	Resultados gerais do KNN.	61
Tabela 5	Parâmetros encontrados para o Bayes Ingênuo.	63
Tabela 6	Resultados gerais do Bayes Ingênuo.	64
Tabela 7	Parâmetros encontrados para o SVM.	66
Tabela 8	Resultados gerais do SVM.	66
Tabela 9	Parâmetros encontrados para a Árvore de Decisão.	68
Tabela 10	Resultados gerais da Árvore de Decisão.	69

LISTA DE ABREVIATURAS E SIGLAS

GP	Gaussian process
RNA	Redes Neurais Artificiais
STFT	Short Time Fourier Transform
CWT	Continuous Wavelet Transform
FFT	Fast Fourier Transform
DWT	Discrete Wavelet Transform
Abit	Associação Brasileira da Indústria Têxtil e de Confecção
AM	Aprendizado de Máquina
IA	Inteligência Artificial
SVM	Supported Vector Machine
KNN	K-Nearest Neighbor
CA	Coefficientes De Aproximação
CD	Coefficientes De Detalhes

SUMÁRIO

1	INTRODUÇÃO	21
1.1	ESTADO DA ARTE	24
1.1.1	Defeitos	25
1.1.2	Reconhecimento de Defeitos, Visão Computacional	26
1.2	OBJETIVOS	27
1.2.1	Objetivo Geral	27
1.2.2	Objetivos Específicos	27
1.3	JUSTIFICATIVA	27
1.4	HIPÓTESE	29
1.5	METODOLOGIA	29
2	APRENDIZADO DE MÁQUINA	31
2.1	APRENDIZADOS SUPERVISIONADO E NÃO SUPERVISIONADO	32
2.2	CLASSIFICAÇÃO E RECONHECIMENTO DE PADRÕES	33
2.2.1	Algoritmo do Vizinho mais Próximo (KNN)	35
2.2.1.1	Conjunto De Treinamento	36
2.2.1.2	Medida De Distância	36
2.2.1.3	Número de Vizinhos Utilizados	36
2.2.1.4	Aspectos Do Algoritmo	37
2.2.2	Naive Bayes	37
2.2.3	Árvores De Decisão	38
2.2.4	Máquinas De Vetores De Suporte (SVM)	39
2.3	OTIMIZAÇÃO	42
2.3.1	Otimizador Bayesiano	42
2.4	AVALIAÇÃO DO DESEMPENHO DOS MODELOS ...	43
2.4.1	Validação Cruzada	44
3	VISÃO COMPUTACIONAL E PROCESSAMENTO DIGITAL DE IMAGEM	45
3.1	MÉTODOS DE EXTRAÇÃO DE CARACTERÍSTICAS DE IMAGENS	45
3.1.1	Transformada De Wavelet	46
3.1.1.1	Hitória	46
3.1.1.2	Comparativo Com a Transformada De Fourier	47
3.1.1.3	Tranformada Continua De Wavelet	48
3.1.1.4	Tranformada Discreta De Wavelet	48
4	METODOLOGIA	51
4.1	AQUISIÇÃO E TRATAMENTO DAS IMAGENS	52

4.2	USO DA TRANSFORMADA DE WAVELET.....	55
4.3	BASE DE DADOS	57
5	RESULTADOS OBTIDOS.....	59
5.1	RESULTADOS PARA O KNN.....	60
5.2	RESULTADOS PARA O BAYES INGÊNUO	62
5.3	RESULTADOS PARA O SVM.....	64
5.4	RESULTADOS PARA A ÁRVORE DE DECISÃO	67
5.5	COMPARAÇÃO ENTRE OS CLASSIFICADORES E DIS- CUSSÃO DA APLICAÇÃO	70
5.6	COMPARAÇÃO COM TRABALHOS RELACIONADOS	72
6	CONCLUSÃO	75
6.1	TRABALHOS FUTUROS.....	76
	REFERÊNCIAS	77
	APÊNDICE A – Taxas do classificador KNN	84

1 INTRODUÇÃO

A qualidade nos processos industriais vem se tornando cada vez mais importante e a redução de custos e otimização de processos se tornam cada vez mais necessários (KUMAR, 2008). O controle da qualidade traz aumento de produção e ainda aumento nos lucros para um processo. Pode-se dizer, então, que ela é a métrica mais importante quando se trata de produção. É extremamente difícil ter um processo de fabricação 100% livre de defeitos. Um dos processos industriais que vem recebendo tal atenção quanto a defeitos é o processo de tecelagem. Segundo Malek (2012), o sucesso da tecelagem está significativamente atrelado ao seu sucesso em reduzir defeitos de fabricação.

A indústria têxtil é um importante setor da economia brasileira e mundial. A finalidade primordial dos tecidos é vestir e proteger o corpo humano. Por esse e outros motivos a qualidade da fabricação é tema de inúmeros trabalhos (CHO; CHUNG; PARK, 2005), (JAIN; DUIN; MAO, 2000), (DORRITY; VACHTSEVANOS; JASPER, 1996).

Dois fatores são importantes na tecelagem de um tecido. Primeiramente, a velocidade do tear, que impacta diretamente na produtividade e, por consequência, no faturamento da empresa. Por outro lado, a qualidade também é um importante fator. Segundo Dastoor et al. (1994), o preço de um tecido com defeito reduz cerca de 45% a 65% do valor de venda.

Existem tecidos de primeira e de segunda qualidade de acordo com Malek (2012). Os primeiros são livre de defeito, enquanto que os últimos podem conter alguns defeitos em sua textura ou na superfície. Por ser um produto que passa por processos delicados, como a tecelagem, é muito difícil conseguir produtos de primeira qualidade. Assim grande parte da inspeção e verificação de qualidade diz respeito a redução da ocorrência de defeitos nos tecidos nos estágios iniciais de fabricação (MALEK, 2012).

Na fabricação de tecidos, a inspeção é feita primeiramente para localizar um defeito e posteriormente para corrigir o problema que leva a esse defeito (GAJANAN, 2014). Esse processo leva a diminuição de produtos de segunda qualidade. Esses, por sua vez, têm menor valor de mercado uma vez que têm problemas em sua fabricação desde falhas na trama até mesmo manchas, por exemplo. Esse processo para uma pessoa pode se tornar extremamente cansativo, uma vez que a capacidade de processamento visual humana tem suas limitações e fraquezas.

A inspeção realizada por pessoas tem se tornado obsoleta no

que diz respeito a detecção de defeitos. Um dos fatores que levam a essa constatação é a produtividade. Segundo Kumar (2008), uma pessoa consegue apenas inspecionar 20m/min em um rolo de tecido. Isso mostra o gargalo que existe no quesito de produtividade. Sem contar os outros inúmeros problemas que o sistema visual humano pode encontrar, sendo um deles o cansaço do inspetor.

O processo de inspeção e detecção de defeitos nos tecidos é feito por pessoas especializadas (MALEK, 2012). A maneira como o material é avaliado varia entre as fábricas (KUMAR, 2008). Em algumas, os próprios trabalhadores colocam o tecido na mesa para verificação. Contudo, a maioria tem maneiras automatizadas de retirar o rolo da máquina de tecelagem de forma motorizada e desenrola-lo em um ambiente de iluminação controlado. A inspeção é feita em velocidade relativamente alta, entre 8 a 20 metros por minuto (KUMAR, 2008). A figura 1 mostra melhor como é feito a inspeção por um especialista na produção de uma fábrica.



Figura 1 – Especialista fazendo a inspeção. (MALEK, 2012)

A partir do momento que o especialista encontra um defeito, ele pára a máquina, registra o defeito e o seu local e liga novamente

o motor. O registro muitas vezes é no próprio tecido, feito com uma marca de giz no local da falta. Para cada rolo de tecido inspecionado, o número de defeitos por metro é calculado (MALEK, 2012). As taxas de defeitos prévias, ou seja, dos rolos que já foram inspecionados, são deixadas com o operador da máquina (DORRITY; VACHTSEVANOS; JASPER, 1996), (KUMAR, 2008). Uma vez que o operador notar uma taxa muito alta de defeitos, o aviso é feito ao departamento de produção. Assim, devidos reparos podem ser feitos de modo a diminuir o número de erros cometidos.

Em sistemas de reconhecimento de defeitos como os descritos acima, é praticamente impossível reduzir o erro a zero. Quando a velocidade de produção era lenta em comparação a atual, esse sistema de inspeção funcionava bem. A evolução das técnicas e principalmente das máquinas de tecelagem, tornaram essa uma tarefa de difícil gerência. Inúmeros são os problemas relacionados a detecção visual humana (MALEK, 2012).

Dentre os principais problemas relacionados com detecção por pessoas pode ser citado: Pessoas que são especialistas nessa tarefa são difíceis de se encontrar; Pessoas são mais lentas em relação a máquinas; Os inspetores se cansam rápido, uma vez que essa é uma tarefa dispendiosa; Existem muitos tipos de defeitos; Quando um tecido de 1,4-2 metros de largura a uma taxa de 20 metros por minuto é desenrolado, o acerto do inspetor é de 50% na média e no máximo de 80%.

A inspeção de tecidos é uma tarefa extremamente importante na indústria atual. Pelos motivos citados anteriormente, desenvolver um sistema de visão computacional para esse processo é de grande importância. Obter um sistema confiável de qualidade garante confiabilidade ao produto e por consequência melhores resultados.

Um sistema automatizado de detecção de defeitos melhora a qualidade do produto e sua produtividade (DORRITY; VACHTSEVANOS; JASPER, 1996). A velocidade na produção das máquinas de tecer aumentaram muito ao longo do tempo. O aumento da velocidade leva a demanda de reconhecer e corrigir as falhas a níveis mais difíceis. É necessário, então, diferenciar dois tipos de inspeção. A inspeção *online* e a inspeção *offline*, a primeira ainda é chamada de inspeção de tempo real segundo Dorrity, Vachtsevanos e Jasper (1996). Para Malek (2012), os sistemas *online* provêm dados da produção atual e ainda estão localizados na própria linha de produção de modo que as medidas possam ser tomadas na mesma hora. Já os sistemas *offline* estão localizados no final da linha de produção, ocasionando em perdas de qualidade de tecidos, já que as falhas somente são computadas no final do processo.

O estudo de sistemas de visão computacional tem sido cada vez mais relevantes do ponto de vista industrial e acadêmico. Para imitar a grande variedade das funções humanas, a tecnologia foi a grande propulsora que fez com que a humanidade avançasse do manual para o mecânico e depois do mecânico para o automático (MALEK, 2012). O avanço de técnicas de processamento de imagem e também os algoritmos de aprendizagem de máquina estão tornando os processos industriais cada vez mais inteligentes. A combinação dessas duas práticas tornou possível inúmeras aplicações na indústria, agropecuária, medicina e outros.

É natural que se procure criar máquinas que podem reconhecer padrões (DUDA; HART; STORK, 2012). A capacidade dos seres humanos de aprender é e foi fundamental para a evolução. Sistemas de visão computacional tentam imitar, de certa forma, essa capacidade de aprender e reconhecer padrões. Como o erro humano pode acontecer por diversas razões, como, por exemplo, exaustão, um sistema computacional capaz de fazer esse papel de reconhecimento se faz necessário.

A inteligência artificial é um campo antigo dentro da computação. No passado, esses algoritmos eram tratados de forma somente teórica e não se via grandes aplicações e possibilidades. Hoje, com o aumento do número de problemas e o aumento de dispositivos que fazem de certa forma uma aquisição de dados, esses algoritmos têm se tornado cada vez mais importantes. Uma área que tem se destacado muito nos últimos anos é o Aprendizado de Máquina. Segundo Faceli et al. (2011) o Aprendizado de Máquina é o processo de indução de uma hipótese (ou aproximação de função) a partir da experiência passada.

Assim, a união de duas grandes técnicas, o aprendizado de máquina e o processamento de imagem se tornam uma ferramenta poderosa na resolução de problemas do tipo que este trabalho aborda. Inúmeros são os trabalhos e áreas em que essas duas técnicas são utilizadas para solucionar determinado problema (SCOTT; JAMES; ALI, 2006), (KUMAR, 2008).

1.1 ESTADO DA ARTE

Esta seção apresentará uma investigação do estado da arte de defeitos na indústria têxtil e reconhecimento de defeitos usando visão computacional. O trabalho utiliza como objeto de estudo um tecido plano para as análises.

A literatura, com relação ao reconhecimento de defeitos na in-

dústria têxtil e também em outras áreas afins, é ampla (ZHANG; JING; ZHANG, 2015) (REBHI; ABID; FNAIECH, 2016) (ABDELLAH; AHMED; SLIMANE, 2014) (JMALI; ZITOUNI; SAKLI, 2014) (DROBINA; MACHNIO, 2006) (KUMAR, 2008) (CHO; CHUNG; PARK, 2005) (JAIN; DUIN; MAO, 2000) (KIM; LEE; WOO, 1999) (DASTOOR et al., 1994). Todos esses trabalhos têm um aspecto voltado para a visão computacional.

1.1.1 Defeitos

Existem dois tipos diferentes de defeitos (JMALI; ZITOUNI; SAKLI, 2014). Um tem relação com irregularidades na textura local, principal preocupação das aplicações de detecção. O outro consiste perda de textura e/ou coloração. Para Gajanan (2014) os defeitos são divididos entre Defeitos de Fio e Defeitos na Tecelagem. Os Defeitos de Fio ocorrem geralmente na etapa de enrolamento do tecido. Já os Defeitos de Tecelagem, ocorrem no processo de tecer.

Alguns trabalhos tratam de separar em vários tipos de defeito como em Behera e Mani (2007). Os autores mostram 8 tipos diferentes de defeitos; fio grosso na trama (*Thick Weft Yarn*); trama partida (*Crack*); Fios Misturados (*Mixed Yarn*); Urdidura Quebrada (*Broken Warp*); Urdidura Flutuante (*Warp Float*); Marca (*Starting Mark*); Trama Flutuante (*Weft Float*); Barra na trama (*Weft Bar*). Os resultados para essa quantidade de defeitos foram satisfatórios, mas os autores ressaltam que a quantidade de informação extraída da imagem pode ser um problema.

Uma metodologia mais interessante é encontrada no trabalho de (SENGOTTUVELAN; WAHI; SHANMUGAM, 2008). Nesse trabalho são destacados 2 tipos diferentes de defeitos apenas, Fio Duplo (*Double Yarn*) e Fio Faltante (*Missing Yarn*). Com essa abordagem os autores chegaram a resultados consistentes para um número determinado de amostras. Os autores ainda destacam que estudos serão feitos com mais tipos de defeitos.

Ainda há abordagens que procuram encontrar um meio termo no número de tipos de defeitos. Em Abdellah, Ahmed e Slimane (2014), são listados 4 tipos de defeitos que são frequentes. São eles: Fio Faltante Horizontal (*Horizontal Missin Yarn*), Fio Faltante Vertical (*Vertical Missing Yarn*), Buraco (*Hole*) e Ponto (*Spot*). Com esse tipo de informação, uma fábrica poderia direcionar seus esforços para sanar esses problemas, pois geralmente são defeitos nas máquinas que geram as faltas.

Para este trabalho será empregado uma abordagem mais próxima com os trabalhos de Abdellah, Ahmed e Slimane (2014) e Zhang, Jing e Zhang (2015). O objetivo principal na indústria é encontrar um defeito e posteriormente sanar o problema que o criou. Somente realizar uma distinção entre defeito e não-defeito não seria interessante. Assim, como existem muitos tipos de defeitos (GAJANAN, 2014), é necessário encontrar os defeitos mais comuns e distingui-los.

1.1.2 Reconhecimento de Defeitos, Visão Computacional

Existem diversos trabalhos que abordam o problema de detecção de defeitos em tecidos com o uso de processamento de imagem. Como foi citado anteriormente, diferentes abordagens são empregadas. Segundo Kumar (2008) existem três tipos de técnicas de detecção de defeitos em tecidos: a estatística, a espectral e as baseadas em modelo. De certa forma todos essas abordagens são estatísticas, uma vez que empregam cálculos estatísticos para computar os defeitos.

A abordagem estatística separa a imagem em regiões com comportamento distinto. Já o método espectral se mostra adequado para a detecção de defeitos em imagens uniformes texturizadas, ela é comumente utilizada em aplicações de detecção de defeitos em tecidos. Por último a abordagem baseada em modelo é utilizada onde as anteriores ainda se mostram inúteis, como por exemplo, em superfícies estocásticas (KUMAR, 2008).

Muitos trabalhos foram feitos utilizando abordagens estatísticas, como em Amet, Ertuzun e Erçil (1998) que utiliza matriz de coocorrência para extrair informação de tecidos texturizados. Esse tipo de técnica prove mais informação mas é computacionalmente mais custosa. Em Zhang, Jing e Zhang (2015), o autor se utiliza de matrizes de coocorrência de escala de cinza (GLCM) e padrões locais binários (PLB) para extrair informação, duas técnicas de abordagem estatística.

Os métodos mais comuns são os espectrais. Os trabalhos com essa abordagem são mais numerosos. Técnicas com transformadas e filtros são encontradas. Em Behera e Mani (2007), a transformada do cosseno discreta foi utilizada de modo a encontrar diferentes espectros para os defeitos. Já em Kim, Lee e Woo (1999), outra transformada, a de *Wavelet* foi utilizada para o reconhecimento de defeitos. Outros trabalhos podem ser encontrados na literatura, muitos deles se utilizam da Transformada de Fourier (KUMAR, 2008).

Os trabalhos que se utilizam de metodologias baseadas em mo-

delo são mais raros, mas não menos relevantes. Em Cohen, Fan e Attali (1991), o autor usa a técnica chamada de Campo Randômico de Gauss Markov com sucesso para modelar muitas texturas naturais. Vários outros trabalhos podem ser encontrados com todas as abordagens em Kumar (2008).

Apesar das abordagens serem usadas separadamente nos trabalhos citados, na literatura ainda pode ser encontrado aplicações onde as mesmas foram usadas em conjunto. Segundo Kumar (2008), as três abordagens dão resultados diferentes, assim, a combinação dessas abordagens pode proporcionar melhores resultados do que cada uma individualmente.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

O objetivo deste trabalho é elaborar um sistema capaz de classificar diferentes tipos de defeitos em tecidos através de técnicas de processamento de imagem.

1.2.2 Objetivos Específicos

1. Utilizar *Wavelets* de modo a extrair informação dos tecidos.
2. Usar técnicas de aprendizado de máquina supervisionado de modo a tornar possível a classificação de defeitos.
3. Classificar os tecidos em diferentes tipos de defeitos.
4. Utilizar o algoritmo de otimização de Bayes para melhorar a resposta dos classificadores.
5. Obter o sistema validado com validação cruzada.

1.3 JUSTIFICATIVA

A automatização de processos tem se tornado fundamental para o aumento da produtividade e por consequência competitividade na indústria. Como citado anteriormente, o ser humano vêm gradativamente desde a revolução industrial mecanizando e automatizando pro-

cessos. Trabalhos que são repetitivos e não envolvem algum tipo de criatividade passaram a ser desenvolvidos por máquinas. Como a detecção de defeitos é um processo repetitivo e exaustivo para uma pessoa, um sistema inteligente se faz necessário.

A indústria têxtil na década de 90 representava 14% de todos os postos de trabalho da indústria (KON; COAN, 2009). Fato esse que demonstra o quão importante foi e é esse mercado para o Brasil. Com a abertura do mercado também na década de 1990, por meio de políticas públicas no Brasil, o país se tornou obsoleto no setor têxtil, uma vez que tinha seu parque de máquinas ultrapassado em relação ao restante do mundo e principalmente em relação ao mercado asiático.

Essa defasagem causou ao Brasil uma queda nas exportações, chegando a um déficit de US\$ 1.157.014.000,00 em 1996. O país só conseguiu amenizar o panorama a partir de 1998, onde nos anos seguintes houveram quedas significativas no déficit chegando em 2001 em US\$ 284.087.000,00 (KON; COAN, 2009). A partir desse fato, o setor teve de se modernizar. Somente naquele período foram investidos aproximadamente US\$ 6 bilhões, aplicados a maior parte em máquinas têxteis, onde os recursos foram distribuídos praticamente de forma uniforme pelos setores mais importantes, como na fiação, tecelagem, malharia e beneficiamento (KON; COAN, 2009). Esses pesados investimentos eram necessários já que os produtos vindos de fora do território nacional tinham produtos especializados. Um fato importante é que o número de empregos diminuiu em relação ao que era anteriormente a modernização do parque industrial.

Todo esse investimento e modernização trouxeram ao material produzido mais qualidade, redução de custos com funcionários e velocidade na produção. Esses fatores são abordados nesse trabalho e mostram como a modernização industrial é e foi importante para o mercado mundial. O Brasil também precisa produzir máquinas capazes de produzir melhor, com qualidade e competitividade, para assim obter autonomia tecnológica, conforme a lei da inovação.

Hoje, segundo dados de 2015 da Abit (Associação Brasileira da Indústria Têxtil e de Confecção), o Brasil posiciona-se como o quinto maior produtor têxtil do mundo. O setor é o segundo que mais emprega no país com 1,6 milhão de trabalhadores que geram um faturamento anual de US\$ 23,6 bilhões. São cerca de 33 mil empresas instaladas no ramo. Esses dados mostram como esse é um setor forte no Brasil e que ainda não encontrou a hegemonia já que o déficit ainda é alto, cerca de 4,8 bilhões de dólares. O estado de Santa Catarina possui um polo têxtil, concentrado na região do vale do Itajaí, que possui 900 empresas

do setor.

Os fatos anteriormente citados mostram o quão importante é a pesquisa em novas tecnologias para modernizar os setores econômicos e também para o avanço em pesquisa em si. Isso foi comprovadamente um problema no passado para o Brasil, que teve de se modernizar as pressas ante a concorrência do mercado externo. Assim o desenvolvimento de um sistema de detecção de falhas em tecidos pode ajudar o Brasil a se destacar ainda mais no mercado têxtil, mesmo já sendo um dos maiores participantes atualmente.

O presente trabalho fará um estudo global sobre as técnicas de Aprendizado de Máquina e também sobre as *Wavelets*. Esse estudo poderá servir de base para futuros trabalhos acadêmicos. A aplicação montada no presente trabalho ainda servirá de exemplo de como um sistema de visão computacional pode variar desde o algoritmo classificador utilizado á técnica de extração de característica que neste caso, usará a Transformada de *Wavelet*.

1.4 HIPÓTESE

1. O uso de um sistema de visão computacional pode melhorar o sistema de detecção de defeitos em tecidos na indústria.

1.5 METODOLOGIA

1. Levantar o estado da arte em métodos de reconhecimento de defeitos em tecidos.
2. Criar a base de dados, bem como o conjunto de imagens a ser utilizado.
3. Extrair informações da imagem com a Transformada de *Wavelet*.
4. Testar diferentes algoritmos classificadores de modo a encontrar o de melhor resposta para esse problema.
5. Melhorar o desempenho dos algoritmos classificadores através do Otimizador de Bayes.
6. Validar o sistema através da técnica de validação *k-fold cross validation*.

2 APRENDIZADO DE MÁQUINA

Na computação, programas são escritos com regras bem definidas de modo a solucionar determinados problemas. Praticamente qualquer tarefa pode ser descrita e solucionada por meio de um algoritmo. Contudo, existem certos tipos de problemas que tem resolução complexa por meio de programas, por exemplo, reconhecimento facial e de fala (DUDA; HART; STORK, 2012). No entanto, seres humanos podem resolver esse tipo de problema com muita facilidade. Os melhores reconhecedores de padrões na maioria dos casos são humanos, contudo, ainda não entendemos como os humanos reconhecem padrões (JAIN; DUIN; MAO, 2000).

Na área de inteligência artificial (IA), técnicas de Aprendizado de Máquina (AM) têm sido utilizadas para resolver esse tipo de problema, onde é necessário reconhecer padrões em um conjunto de dados.

Quando se fala em AM, segundo Faceli et al. (2011) algoritmos são construídos de modo que a máquina ou o programa em si aprendam com a experiência passada. Para isso, empregam um princípio de inferência denominado indução, no qual se obtêm conclusões genéricas a partir de um conjunto genérico de dados (FACELI et al., 2011). Com isso, esses algoritmos são capazes de resolver problemas a partir de dados que representam instâncias do problema a ser resolvido.

Uma definição mais formal pode ser encontrada em Russell e Norvig (2004). Um exemplo é um par $(x, f(x))$, onde x é a entrada e $f(x)$ é a saída da função aplicada a x . A tarefa da indução é: Dada uma coleção de exemplos de f , retornar uma função h que se aproxime de f .

Para Duda, Hart e Stork (2012) existem critérios diferentes para a organização do AM. Esses critérios dependem explicitamente da tarefa a ser realizada. São eles: Aprendizado de Máquina Supervisionado e Aprendizado de Máquina Não Supervisionado. Enquanto um, como o nome já diz, depende de um "supervisor" externo que conhece a saída desejada. O outro explora os dados sem a necessidade de conhecimento prévio das saídas. A figura 2 mostra melhor a hierarquia das grandes áreas do AM.

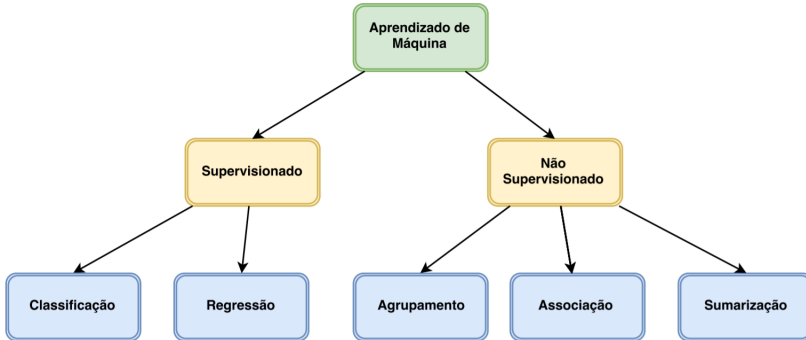


Figura 2 – Hierarquia de técnicas de aprendizado de máquina.

Mesmo que o AM seja associado a IA, outras áreas são de grande importância para o avanço do AM. A probabilidade, a estatística e a teoria da computação para citar alguns. O AM é uma das áreas que mais cresce na computação, sendo que os algoritmos usados para AM são inúmeros, com muitas variações e ainda sem restrições nas aplicações.

2.1 APRENDIZADOS SUPERVISIONADO E NÃO SUPERVISIONADO

Quando se fala em aprendizado supervisionado, uma palavra tem que estar presente, os rótulos ou etiquetas. O AM supervisionado consiste em treinar um conjunto de dados com base nesses rótulos, que podem também ser chamados de classes (DUDA; HART; STORK, 2012). Já o aprendizado não supervisionado não possui esses rótulos, assim, esses métodos são chamados de exploratórios. A técnica mais utilizada de AM não supervisionada é a técnica de agrupamento (JAIN; DUIN; MAO, 2000).

Neste trabalho as técnicas utilizadas basicamente serão de AM supervisionado, uma vez que os defeitos serão organizados em classes predefinidas. O objetivo não é entender o conjunto de dados, já que os defeitos são conhecidos. As imagens serão classificadas previamente de acordo com seu defeito, para depois serem usadas em um conjunto de treinamento e posteriormente de teste.

As técnicas de AM supervisionado são inúmeras, mas esse trabalho trará um estudo apenas dos classificadores. Um classificador é

uma função que dado um exemplo não rotulado, atribui esse exemplo a uma das possíveis classes (DIETTERICH, 1998).

Este trabalho trará 4 técnicas de classificação, o Algoritmo do Vizinho mais Próximo, Máquinas de Vetores de Suporte (SVM), do inglês, *Supported Vector Machine*, Árvores binárias de decisão e o Bayes ingênuo, do inglês, *Naive Bayes*. Elas são técnicas diferentes, que usam abordagens distintas para treinamento e predição de saídas. Esses algoritmos serão estudados mais a fundo nas seções subsequentes.

2.2 CLASSIFICAÇÃO E RECONHECIMENTO DE PADRÕES

Segundo Michie, Spiegelhalter e Taylor (1994) a classificação ocorre em uma grande variedade das atividades humanas. Desde simples tomadas de decisões e até mesmo previsão, de clima, por exemplo. Para que a tarefa de classificação ocorra é necessário existir um conjunto de dados, também chamado de base de dados ou do inglês *database*. Essa base de dados possui entradas ou observações, cada uma das observações possui características. Essas entradas com suas respectivas características possuem uma das possíveis saídas, também chamadas classes.

As características são fornecidas pelo extrator de características. O limite conceitual entre um extrator de características e um classificador é quase arbitrário. Um extrator de características ideal faria uma representação de dados que reduziria o trabalho do classificador; inversamente, um classificador universal não necessitaria de um extrator de características sofisticado (DUDA; HART; STORK, 2012).

A tarefa do classificador é usar o vetor de características proveniente do extrator e classificar a entrada numa das possíveis saídas. A parte do extrator de características se dará por técnicas de processamento de imagem, como dito anteriormente. A figura 3 mostra melhor como funciona a tarefa de um sistema de classificação.

Dentro da etapa de classificação existe uma das mais importantes tarefas, se não a mais importante, a etapa de treinamento. A etapa de treinamento consiste em gerar um modelo que represente os dados, dado uma base de dados com observações conhecidas e com saídas também conhecidas. Treinar um modelo por indução para gerar previsões razoáveis para dados novos que não possuem saída conhecida. A figura 4 mostra melhor como funciona essa tarefa.

A qualidade do sistema de classificação como um todo depende de variáveis como técnicas de pré-processamento e qualidade de senso-

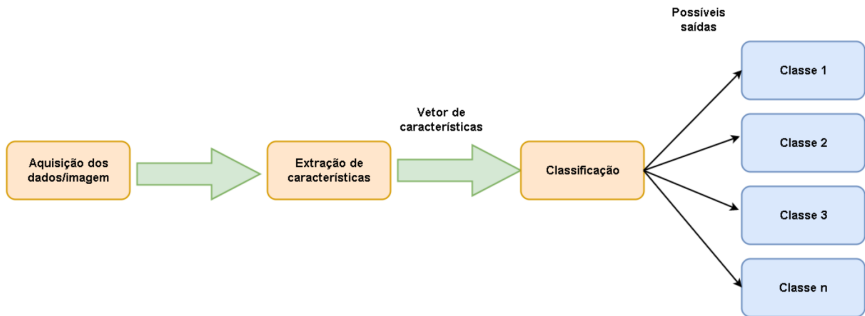


Figura 3 – Etapas de um sistema de classificação com aquisição e extração de características.

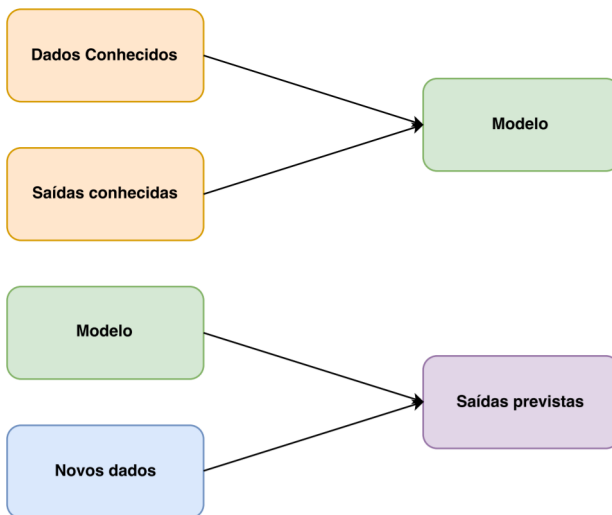


Figura 4 – Estrutura geral para a generalização de um modelo e predição .

res por exemplo. É geralmente acordado que um problema de reconhecimento bem definido e suficientemente limitado (pequenas variações intraclasses e grandes variações interclasses) conduzirá a uma repre-

sentação de padrões e uma simples estratégia de tomada de decisão (JAIN; DUIN; MAO, 2000).

Para Russell e Norvig (2004) um algoritmo de aprendizagem é bom se produz hipóteses que fazem um bom trabalho de previsão das classificações de exemplos não conhecidos. Pode-se então avaliar a qualidade de uma hipótese comparando suas previsões contra a classificação correta. Isso é feito com um conjunto de exemplos conhecido como teste. É conveniente então dividir o conjunto total de dados em um conjunto de teste e um conjunto de treinamento. Tal tarefa será melhor discutida posteriormente.

Segundo Michie, Spiegelhalter e Taylor (1994) existem diversos problemas relacionados a classificação. O mais importante deles é a acurácia, ela representa a proporção de acertos na classificação. A velocidade também é importante em algumas circunstâncias, como nesse trabalho, que existe um linha de produção. A compreensibilidade de um classificador é necessária para um melhor entendimento do mesmo. Ainda, o tempo de treinamento também conta na hora da decisão do classificador, uma vez que alguns demoram muito mais para realizar tal tarefa do que outros.

De acordo com Jain, Duin e Mao (2000) existem quatro tipos de abordagens para reconhecimento de padrões: Template Matching, a estatística, a sintática e a neural. No presente trabalho será abordada apenas a estatística. Na abordagem estatística os limites de decisão são determinados por distribuições de probabilidade (DUDA; HART; STORK, 2012).

2.2.1 Algoritmo do Vizinho mais Próximo (KNN)

Este é um algoritmo que se baseia na proximidade entre os dados na realização das predições. A hipótese base é de que dados similares tendem a estar concentrados em uma mesma região no espaço de entrada (FACELI et al., 2011). Um dos métodos mais comuns que utiliza essa abordagem é o algoritmo do vizinho mais próximo ou do inglês *k-nearest neighbor*.

O algoritmo foi proposto por Aha, Kibler e Albert (1991) e é uma abordagem estatística. Ele é considerado um algoritmo preguiçoso, uma vez que não aprende um modelo compacto dos dados, apenas memoriza os dados de treinamento. Assim três são os parâmetros que influenciam diretamente a execução desse algoritmo:

1. Quais entradas devem ser memorizadas.

2. A medida de distância usada.

3. O número de vizinhos utilizados.

2.2.1.1 Conjunto De Treinamento

Este é um parâmetro que aparece em todos os algoritmos. Ele influencia tanto no tempo de aprendizado quanto na generalidade do modelo. Assim em Aha, Kibler e Albert (1991), o autor mostra algumas técnicas de escolha no número de entradas utilizadas para teste e treinamento. O ideal é utilizar o menor número possível de exemplos de modo a minimizar o tempo de treinamento.

2.2.1.2 Medida De Distância

Também chamada de medida de similaridade entre as entradas (JAIN; DUIN; MAO, 2000). Assim, uma menor distância entre os exemplos equivale a uma maior similaridade entre eles. Diversas métricas dessa medida podem ser usadas, uma das mais comuns é a distância euclidiana que é mostrada na equação 2.1:

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^d (x_i^l - x_j^l)^2} \quad (2.1)$$

onde x_i e x_j são duas entradas representadas por vetores no espaço \mathbb{R}^d , e x_i^l e x_j^l são os elementos desses vetores que, que correspondem aos valores da coordenada l , ou seja os atributos.

2.2.1.3 Número de Vizinhos Utilizados

O número de vizinhos a ser utilizados talvez seja o parâmetro mais difícil a ser encontrado. Não existe um único número de vizinhos que seja apropriado para cada problema em particular.

É importante ressaltar que todos esses parâmetros serão encontrados com o uso de um otimizador. Esse otimizador será melhor estudado nas seções seguintes.

2.2.1.4 Aspectos Do Algoritmo

O algoritmo KNN tem um treinamento bastante simples, o armazenamento das entradas. Ele ainda constrói aproximações locais da função objetivo. Característica vantajosa quando a função objetivo é muito complexa, uma vez que pode ser descrita por uma coleção de aproximações locais de menor complexidade (MITCHELL, 1997). Assim ele pode ser aplicado mesmo em problemas complexos. Ele ainda é de fácil entendimento.

Segundo Faceli et al. (2011), pontos negativos do algoritmo começam pelo fato de ser do tipo *lazy*. Isso representa um pequeno esforço computacional no treinamento, mas um esforço grande na etapa de classificação. Deve ser calculado a distância da entrada a todas as entradas do treinamento. Ainda é afetado pela presença de atributos redundantes. Assim o KNN é considerado um bom algoritmo de classificação, uma vez que ele pode ser usado em qualquer tipo de problema. Existem outros algoritmos que são considerados mais robustos e que tem melhor desempenho onde o KNN falha.

2.2.2 Naive Bayes

Considerado um método probabilístico e usado quando os dados são imprecisos ou incompletos, o Naive Bayes é outro algoritmo de classificação. Ele recebe o nome "ingênuo" porque supõe que os atributos são condicionalmente independentes uns dos outros, dada a classe (RUSSELL; NORVIG, 2004). Inúmeros são os estudos a respeito desse classificador. Por exemplo, Michie, Spiegelhalter e Taylor (1994) mostram em seu estudo detalhado que o Bayes ingênuo é competitivo em comparação a outras técnicas, incluindo redes neurais e árvores de decisão.

Por ser uma abordagem que usa conceitos de probabilidade, esse algoritmo requer conhecimentos nessa área. Segundo Zhang (2004), tipicamente, um exemplo ou entrada E , é representado por uma tupla de valores dos atributos (x_1, x_2, \dots, x_n) , onde x_i é o valor do atributo X_i . Ainda, C representa a variável de classificação, e c é o valor de C . Para ilustrar assume-se que existem somente duas classes: 1 (a classe positiva) ou 0 (a classe negativa).

De acordo com a regra de Bayes, a probabilidade de um exemplo $E = (x_1, x_2, \dots, x_n)$ pertencer a classe c é dada pela equação 2.2.

$$p(c|E) = \frac{p(E|c)p(c)}{p(E)}. \quad (2.2)$$

Ainda, E é classificado como da classe 1 se e somente se

$$f_b(E) = \frac{p(1|E)}{p(0|E)} \geq 1, \quad (2.3)$$

onde $f_b(E)$ é chamado de classificador Baysiano (ZHANG, 2004).

Como dito anteriormente, todos os atributos são independentes entre si, dada uma classe, implica que,

$$p(E|c) = p(x_1, x_2, \dots, x_n|c) = \prod_{i=1}^n p(x_i|c) \quad (2.4)$$

O classificador de bayes ingênuo pode então ser expresso na forma da equação 2.5 a seguir:

$$NB(E) = \frac{p(C = 1)}{p(C = 0)} \prod_{i=1}^n \frac{p(x_i|C = 1)}{p(x_i|C = 0)} \quad (2.5)$$

Todas as probabilidades necessárias para a obtenção do classificador naive Bayes são computados a partir dos dados de treinamento. Segundo Rish (2001), esse algoritmo é robusto a dados irrelevantes e ruídos. Também foi notado que as teorias aprendidas são fáceis de compreender pelos especialistas do domínio. Sempre lembrando que essa é uma característica desejável quando se trata de problemas de classificação.

2.2.3 Árvores De Decisão

Uma outra abordagem de classificação são os métodos baseados em árvores. Em todos os outros métodos foram baseados em vetores de características com dados discretos ou contínuos. Como por exemplo, no algoritmo do vizinho mais próximo que existia uma distância natural entre os vetores. A maioria dos classificadores solucionam problemas desse tipo, onde existe alguma noção de métrica. Quando um problema de classificação envolve dados nominais, que são descrições e que não tem nenhuma noção de similaridade ou mesmo ordem, chamados ainda de não métricos, são utilizados métodos de árvore para resolve-lo (DUDA; HART; STORK, 2012).

A ideia principal de uma árvore de decisão é dividir para conquistar. O problema mais complexo é dividido em subproblemas. Esses subproblemas são avaliados recursivamente utilizando a mesma estratégia. Dividindo assim o espaço de instâncias em subespaços. Esse tipo de modelo é utilizado em algoritmos como: (BREIMAN et al., 1984) e (QUINLAN, 1993). Esses trabalhos são referência na área de Árvores de Decisão. Assim uma árvore de decisão é um grafo acíclico direcionado em que cada nó ou é um nó de divisão, com dois ou mais sucessores, ou um nó folha (FACELI et al., 2011).

Um exemplo prático pode ser visualizado na figura 5. Os atributos X_1 e X_2 definem os espaços da árvore. Cada nó da árvore corresponde a uma região nesse espaço. As regiões definidas pelas folhas são mutuamente excludentes. Por outro lado a união dessas regiões cobre todo o espaço definido pelos atributos (FACELI et al., 2011).

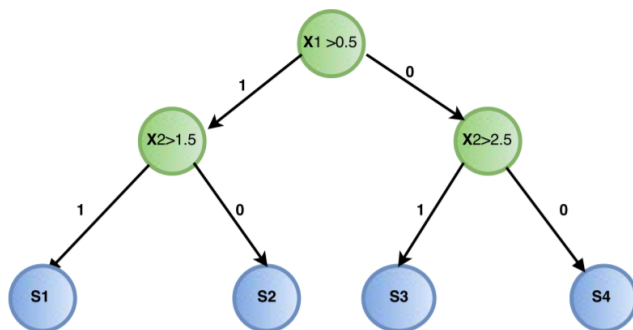


Figura 5 – Árvore de Decisão.

A construção do algoritmo que cria a árvore é bastante simples. Consiste em divisões sucessivas dos dados, escolhendo atributos que maximiza alguma medida de impureza. Contudo, geralmente são encontrados maximizações locais e não globais do problema. Sendo assim considerado um problema *NP completo* (QUINLAN, 1993).

2.2.4 Máquinas De Vetores De Suporte (SVM)

Uma das técnicas mais utilizadas nos últimos anos é de Máquina de Vetores de Suporte. Especialmente robusta para casos de classificação binária, onde existem apenas duas classes a serem classificadas. Inúmeros são os trabalhos que se utilizam deste classificador. Como

por exemplo em (ABDELLAH; AHMED; SLIMANE, 2014), que tem um problema de classificação de defeitos em tecidos com várias classes, similar ao encontrado neste trabalho.

Treinar uma máquina de vetores de suporte consiste em encontrar o hiperplano ótimo. Assim como uma Rede Neural, o objetivo de uma SVM é a otimização de alguma função. Por exemplo, um hiperplano com a distância máxima dos padrões aproximados. Os vetores de suporte são os vetores que definem o plano (DUDA; HART; STORK, 2012).

Como outros classificadores lineares, o SVM tenta avaliar um limite de decisão linear (assumindo que os dados são linearmente separáveis) ou em um hiperplano linear entre as duas classes (figura 5a) (RAJPOOT; RAJPOOT, 2004). Assim, em teoria existem infinitos planos que podem classificar corretamente os dados de treinamento, como mostra a (figura 5b). O SVM tenta então encontrar o hiperplano ótimo para um dado problema (figura 5c).

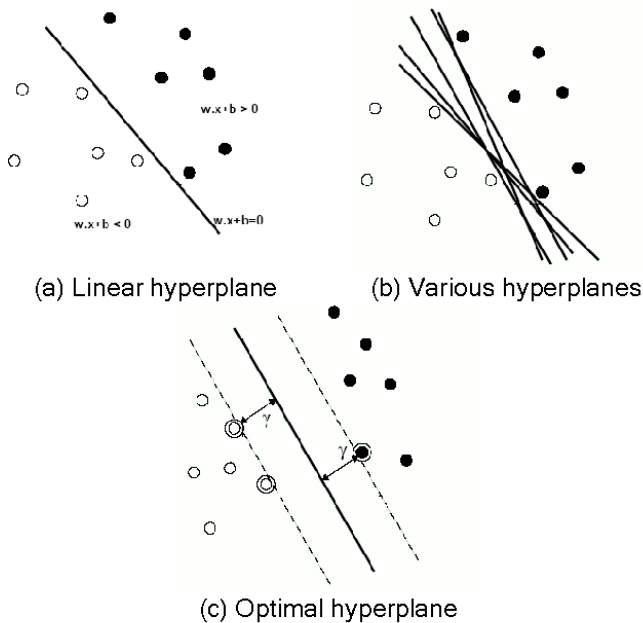


Figura 6 – SVM linear. (RAJPOOT; RAJPOOT, 2004)

Segundo (RAJPOOT; RAJPOOT, 2004), os SVMs possuem várias

características desejáveis em um classificador, sendo assim, uma das técnicas mais robustas. Pode-se destacar por exemplo:

- O tempo computacional em comparação as Redes Neurais;
- Melhor performance com dados de dimensão maiores;
- Comportamento melhor com ruído em comparação com uma rede neural.

Há de se destacar também que esse algoritmo tem resultado de entendimento difícil, uma característica que penaliza sua utilização. Ainda assim, os SVMs em problemas de classificação binária são consenso por sua robustez e bons resultados em uma grande gama de aplicações. Por ser um classificador binário, é necessário utilizar uma abordagem de transformar o algoritmo de forma que ele classifique mais de duas classes. Contudo, isso ocasiona em um tempo computacional maior em relação a outros algoritmos.

Para finalizar, a tabela 1 mostra as diferenças sumarizadas entre os classificadores. Essa tabela ajudaria por exemplo, na decisão de qual classificador utilizar numa aplicação inicialmente. Pode-se notar que o classificador de melhor relação VelocidadeXMemóriaXInterpretabilidade é a Árvore de Decisão. Contudo, essa tabela é apenas uma forma de comparação crua, para um problema específico é melhor testar diferentes classificadores.

Classificador	Vel. de classificação	Uso de memória	Interpretabilidade
Árvores de Decisão	Rápido	Pouco	Fácil
SVM	Médio para linear. Devagar para outros.	Médio para linear. Outros: Médio para multiclases, Bastante para binário.	Fácil para SVMs lineares. Difícil para todos os de tipo kernel.
KNN	Devagar para Cúbico. Médio para outros.	Médio.	Difícil.
Bayes Ingênuo	Pouco para distribuições simples. Devagar para dados de grande dimensão.	Pouco para distribuições simples. Médio para dados de grande dimensão.	Fácil

Tabela 1 – Características Gerais dos Classificadores. Adaptada da página online do Matlab.

2.3 OTIMIZAÇÃO

Praticamente nenhum classificador é livre de parâmetros (SNOEK; LAROCHELLE; ADAMS, 2012). No presente trabalho todos os algoritmos apresentam esse tipo de característica. Por exemplo, pode-se citar o KNN, que além de uma função de similaridade, já mencionada, possui ainda o número de vizinhos a ser escolhido. O fato é que encontrar a melhor combinação desses parâmetros é difícil e requer conhecimentos específicos e experiência com o algoritmo utilizado. Assim, um especialista em encontrar os melhores parâmetros em uma SVM não poderá fazer o mesmo numa RNA. Outro ponto importante é que os parâmetros também dependem do problema a ser tratado, mais precisamente da base de dados, portanto, para cada problema um dado conjunto de parâmetros funcionará melhor do que outros.

2.3.1 Otimizador Bayesiano

O otimizador Bayesiano visa, como os outros métodos de otimização, o mínimo de uma função $f(x)$ em algum conjunto de dados limitado X (SNOEK; LAROCHELLE; ADAMS, 2012). O que torna o algoritmo diferente de outros métodos é que ele constrói um modelo probabilístico para $f(x)$. Esse algoritmo é bastante eficiente, onde grande parte dessa eficiência está na habilidade da otimização Bayesiana de incorporar comportamento de previsão ao problema, daí o nome do próprio algoritmo, uma vez que utiliza o teorema de Bayes em seu método. Assumindo assim que há conhecimento prévio nos dados, algumas funções objetivo são mais plausíveis que as outras. Os principais elementos da minimização são:

- Um modelo Gaussiano de $f(x)$;
- Uma atualização para modificar o modelo Gaussiano a cada nova estimativa de $f(x)$;
- Uma função de aquisição, baseada no processo do modelo Gaussiano. Quando maximizada determina o próximo ponto de avaliação.

O modelo Gaussiano, do inglês, *Gaussian process* é usado para modelar a função objetivo. Um GP é uma generalização de uma distribuição normal multivariada para conjuntos de índices arbitrários,

incluindo vetores de comprimento infinito ou funções. Assim, isso permite que os dados observados possam ser traçados para computar uma distribuição posterior do modelo para qualquer número finito de pontos consultados (GELBART; SNOEK; ADAMS, 2014). Portanto a distribuição marginal em qualquer ponto é Gaussiana, com média e variância conhecidas.

As funções de aquisição exploram regiões onde o modelo acredita que a função objetivo é baixa e regiões onde a incerteza é alta (GELBART; SNOEK; ADAMS, 2014). Esse tipo de característica é encontrado na função de aquisição melhoria esperada, do inglês, *expected improvement*. Essas funções avaliam o quão bom um ponto x é com base na função de distribuição. Por fim existem três formas de parar o algoritmo, primeiramente pelo número de iterações que pode ser definida, pode também ser por tempo limitado ou ainda por um critério arbitrário fornecido previamente.

De forma análoga, o otimizador Bayesiano pode ser sumarizado pela equação 2.6

$$w = f(X, y, \lambda, z). \quad (2.6)$$

onde X é a base de dados, y são as classes corretas para cada exemplo das amostras, λ é o conjunto de todos os *hiper-parâmetros* de um classificador e z é o ruído randômico.

O objetivo é encontrar um λ_{opt} que maximiza a precisão esperada dado uma base de dados, como mostra a equação 2.7 .

$$\lambda_{opt} = \operatorname{argmax}\{E[f(X, y, \lambda, z)|X, y]\} \quad (2.7)$$

2.4 AVALIAÇÃO DO DESEMPENHO DOS MODELOS

Uma das partes mais importantes em um sistema de classificação é a avaliação dos modelos gerados pelos classificadores. Para Faceli et al. (2011), não existe técnica que seja universal na resolução de um problema específico. Enquanto que um SVM pode resolver um problema com dados de grande dimensão, para esse mesmo problema não seria sensato usar o KNN. Contudo, a única forma de realmente saber qual o melhor algoritmo é testando e validando os resultados.

Uma das métricas comumente utilizadas é a taxa de erro do classificador, que representa as classificações incorretas. Assim o erro e de um classificador g pode ser dado pelo somatório das predições da equação $C(a)$, dividido pelo número de objetos n do modelo. Onde

$C(a) = 1$ se a é verdadeiro e 0 se é falso. Essa taxa representa a proporção de exemplos do conjunto que foram classificados errado por g , ela é obtida comparando a classe conhecida de x_i, y_i , com a classe que o classificador deu como saída, $g(x_i)$. A equação 2.8 mostra a taxa de erro.

$$e(g) = \frac{1}{n} \sum_{i=1}^n C(y_i \neq g(x_i)) \quad (2.8)$$

O complemento da taxa de erro é a chamada acurácia do modelo dada pela equação 2.9. Ela representa a taxa de acertos do classificador, e quanto mais próxima a 1 melhor, diferentemente da taxa de erro.

$$ac(g) = 1 - e(g) \quad (2.9)$$

Uma outra forma interessante de avaliar o desempenho do classificador é a matriz de confusão. Essa matriz ilustra o número de predições corretas e incorretas em cada classe (FACELI et al., 2011). Na matriz, as linhas representam as classes verdadeiras, e as colunas, as classes classificadas pelo classificador. Esses conceitos serão melhor explorados a medida que os classificadores forem analisados.

2.4.1 Validação Cruzada

Para (FACELI et al., 2011), avaliar o modelo com o conjunto n total dos dados pode tornar a avaliação no chamado erro ou acerto aparente. Isso acontece devido ao fato de que predizer com exemplos que pertencem ao conjunto de treinamento torna o modelo superajustado ao conjunto de dados ao qual ele foi treinado. Para evitar isso, o conjunto de dados n é então dividido entre teste e treinamento. Existem diferentes técnicas de *amostragem* para dividir esses dados. No presente trabalho será usada a técnica *K-fold Cross Validation*, que é uma técnica bastante usada de validação cruzada.

Segundo Jain, Duin e Mao (2000), o *K-fold Cross Validation* é a divisão dos exemplos em K diferentes sub exemplos. Esses exemplos são então rotacionados de forma a usar um como teste e o restante como treinamento, até que todos os subconjuntos sejam usados como teste.

3 VISÃO COMPUTACIONAL E PROCESSAMENTO DIGITAL DE IMAGEM

O limiar entre o processamento de imagens e um sistema de visão computacional é difícil de se encontrar. Uma boa diferença entre as duas situações é que enquanto o processamento de imagens tem uma imagem como entrada e como saída, um sistema de visão computacional, simula, muitas vezes, a visão humana, incluindo um aprendizado e a realização inferências. Portanto, sistemas de visão computacional estão inseridos juntamente com a Inteligência Artificial.

Uma imagem pode ser definida como uma função de duas dimensões $f(x, y)$, onde x e y são coordenadas espaciais (planas), e a amplitude de f em qualquer par de coordenadas é chamado de intensidade da imagem em dado ponto. O termo escala de cinza é usado para descrever a intensidade da imagem em um ponto (GONZALEZ; WOODS; EDDINS, 2010). Quando x , y e a amplitude são finitos, a imagem é chamada *imagem digital*. Assim, processamento digital de imagem se refere a processar uma imagem através de um computador. Analogamente, seres humanos processam imagens, contudo, existem limitações nessa tarefa, como por exemplo no espectro eletromagnético visual.

Ainda de acordo com Gonzalez, Woods e Eddins (2010), imagens coloridas são formadas por uma combinação de imagens individuais. No sistema de cor RGB, uma imagem colorida consiste de três imagens monocromáticas individuais. As três cores são vermelho, verde e azul, do inglês *red*(R), *green*(G) e *blue*(B) respectivamente. Converter os valores das coordenadas x e y para forma digital, é conhecido como *amostragem* e converter a amplitude é chamado *quantização*.

3.1 MÉTODOS DE EXTRAÇÃO DE CARACTERÍSTICAS DE IMAGENS

Em um sistema de visão computacional, é necessário fazer a extração e o reconhecimento de padrões, como descrito anteriormente. Para isso são estabelecidos certos parâmetros quantificáveis para poder ser realizado computacionalmente. Os valores dependem de variáveis como cor, posição, dimensões, textura ou ainda outros que descrevam os objetos (AZEVEDO; CONCI, 2003).

Uma textura pode ser definida como mudanças na intensidade da imagem que formam determinados padrões repetitivos. Essas ca-

racterísticas resultam de propriedades físicas da superfície do objeto, ou ainda pode ser resultado de diferenças de reflexão, como a cor, por exemplo. O problema de inspeção de defeitos em tecidos é um problema de análise de textura (KUMAR, 2008). Assim, para este trabalho foi decidido procurar uma abordagem com uma técnica que fosse capaz de encontrar defeitos em texturas. Uma técnica que foi e é amplamente utilizada para tratar esse tipo de problema é a transformada de *Wavelet*.

3.1.1 Transformada De Wavelet

Wavelets são funções matemáticas que separam um sinal em diferentes componentes de frequência. A Transformada de *Wavelet*, ou onduleta, em tradução livre, foi inicialmente aplicada nas áreas de Geologia, Engenharia elétrica, Matemática, Física Quântica, Processamento de Sinais, entre outros, a partir da década de 1980 (AZEVEDO; CONCI, 2003). O objetivo da decomposição é estudar cada componente com uma resolução combinada a sua escala. Sua vantagem em relação a análise de Fourier é que elas são eficazes quando o sinal contém descontinuidades e picos.

3.1.1.1 História

Segundo Azevedo e Conci (2003), as *wavelets* têm como principal condutor Joseph Fourier (1807), com sua teoria de análise de frequência, chamada de análise de Fourier. Fourier mostrou que qualquer função $f(x)$ com período 2, pode ser reescrita como somas dos termos da série de Fourier, a equação 3.1 a seguir mostra essa equação:

$$a_0 + \sum_{n=1}^{\infty} a_n \cos(nx) + b_n \sin(nx) \quad (3.1)$$

Onde a_0 , a_n e b_n são dados pelas equações 3.2, 3.3 e 3.4, respectivamente.

$$a_0 = \frac{2}{T} \int_0^{2\pi} f(x) dx \quad (3.2)$$

$$a_n = \frac{2}{T} \int_0^{2\pi} f(x) \cos(nx) dx \quad (3.3)$$

$$b_n = \frac{2}{T} \int_0^{2\pi} f(x) \text{sen}(nx) dx \quad (3.4)$$

As contribuições de Fourier tiveram grande influência na matemática no que diz respeito a funções. Com isso, pode-se analisá-las com uma estrutura matemática que varia em escala (AZEVEDO; CONCI, 2003). Para tal, é usado uma função deslocada e com escala modificada de uma certa quantidade. Esse tipo de análise se mostra eficiente por ser menos sensível a ruído, uma vez que mede a variação média do sinal em diferentes escalas.

O conceito de *wavelets* surgiu em um apêndice da tese do matemático Alfred Haar no ano de 1909. O matemático possui um grupo de bases que leva seu nome. Contudo, as bases de Haar não são continuamente deriváveis, fato que as torna limitadas em muitas aplicações.

3.1.1.2 Comparativo Com a Transformada De Fourier

A análise clássica de Fourier permite que um sinal possa ser representado por uma soma infinita de termos em seno e cossenos. Esse tipo de análise tem muitas vantagens e existem várias aplicações para a mesma na análise de sinais em frequência. Contudo, segundo Oliveira (2007), a desvantagem dessa análise é que a relação frequência e tempo inexistente. Assim, a transformada de Fourier não fornece uma análise temporal, apenas frequencial.

Uma das soluções para tal problema foi a criação da transformada de Fourier de tempo curto, do inglês (*STFT - Short Time Fourier Transform*), também conhecida como transformada de Gabor. A ideia por trás dessa transformada foi introduzir um novo parâmetro de frequência local. Desse modo, é possível observar o sinal através de uma janela curta dentro da qual o sinal permanece estacionário (OLIVEIRA, 2007).

Ainda de acordo com Oliveira (2007), o problema da resolução do tempo e da frequência são resultados de um fenômeno físico conhecido pelo nome de *Princípio de Heisenberg*, onde em um sinal é impossível conhecer a frequência exata e o tempo exato de ocorrência desta frequência. O fenômeno independe da transformada usada. A Análise Multi-Resolucional é a forma de se contornar esse problema, analisando o sinal em frequências diferentes com diferentes resoluções.

3.1.1.3 Transformada Contínua De Wavelet

A transformada de *wavelet* é caracterizada por a soma ao longo do tempo de um sinal multiplicado por uma escala e deslocado por uma função *wavelet* Ψ , chamada também de base, ou *wavelet* mãe (AZEVEDO; CONCI, 2003). A transformada de *wavelet* contínua (*CWT* - *Continuous Wavelet Transform*) pode ser escrita como a equação 3.5 a seguir:

$$CWT(a, b) = \int_{-\infty}^{\infty} f(t)\Psi_{a,b}(t)dt \quad (3.5)$$

A realização do cálculo dá como resultado vários coeficientes, estes, função da escala e da posição. A variável a pode ser definida como a escala, ou contração, já o parâmetro b tem por função a localização ou deslocamento. A função $\Psi_{a,b}(t)$, chamada *wavelet* é definida como segue na equação 3.6.

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \quad (3.6)$$

Os passos para a construção da CWT, segundo Azevedo e Conci (2003) são mostrados a seguir:

1. Escolha da *wavelet* e a seguir fazer a comparação em uma parte inicial do sinal original.
2. Cálculo de um valor C , este representa a relação da *wavelet* com a parte do sinal analisado.
3. Desloca-se então a *wavelet* para a direita e repetem-se os passos 1 e 2 até percorrer todo o sinal com a *wavelet*.
4. Dilata-se a *wavelet* e repetem-se os passos 1 a 3.
5. Repetição de todos os passos até o passo 4 em todas as escalas.

3.1.1.4 Transformada Discreta De Wavelet

Analogamente a Transformada Rápida de Fourier (*FFT* - *Fast Fourier Transform*) a Transformada de *wavelet* também possui a sua representante no domínio discreto. A transformada discreta de *wavelet*

(*DWT - Discrete Wavelet Transform*) é mostrada na equação 3.7 a seguir:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi \left(\frac{t-b}{a} \right), a = 2^j, b = k2^j, (j, k) \in \mathbb{Z}^2 \quad (3.7)$$

A grande questão da análise discreta como na *FFT* é não realizar cálculos infinitesimais custosos com grande tempo computacional. Assim tanto a *DWT* quanto a *FFT* são operações lineares que geram uma estrutura de dados que contém $\log_2 n$ segmentos de vários comprimentos, normalmente formadas por diferentes dados de comprimento 2^n (AZEVEDO; CONCI, 2003).

A implementação funcional e eficiente da *DWT* tem como intermediário um banco de filtros em quadratura conjugada. Os coeficientes do filtro passa-alta são determinados pela função *wavelet* mãe Ψ . Os coeficientes do filtro passa-baixa são determinados pela função escala Φ . Maiores detalhes dessa implementação podem ser encontrados no trabalho de (OLIVEIRA, 2007).

Essa é uma análise em cascata. Dado um sinal X , a primeira etapa produz dois conjuntos de coeficientes. Os coeficientes de aproximação CA , derivados da convolução do sinal X com o filtro passa baixa PB . Por outro lado os coeficientes de detalhes CD , são derivados da convolução do sinal X com o filtro passa alta PA . A seguir, o algoritmo troca o sinal X pelos coeficientes de aproximação CA e repete o que foi realizado anteriormente. A figura 7 ilustra essas etapas.

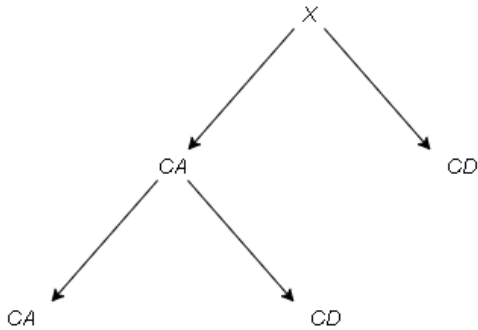


Figura 7 – Esquema em cascata da Transformada Discreta de Wavelet com banco de filtros.

O algoritmo continua indefinidamente em quantos níveis forem necessários. Quanto maior for a decomposição, mais detalhes do sinal em questão são visualizados, portanto mais coeficientes são mostrados. A abordagem para processamento de imagens depende muito do que está se analisando. Como o presente trabalho procura valores que funcionem como padrões para o classificador, avançar demasiadamente nos níveis pode acarretar que o algoritmo de classificação não possa fazer seu trabalho corretamente.

4 METODOLOGIA

Os resultados do presente trabalho foram obtidos a partir de uma aplicação. Este pequeno experimento teve o objetivo de simular um sistema completo de visão computacional. Os materiais utilizados não foram ideais, uma vez que é uma tarefa difícil construir um esquema de aquisição de imagem totalmente isolado e ainda com condições de iluminação perfeitas. A aplicação completa é mostrada como grandes etapas na figura 2.2.

Pelos motivos elucidados acima, a grande dificuldade no trabalho como um todo foi encontrar uma configuração geral que tornasse a aquisição da imagem imune a ruído. Isso significa dizer que ao realizar a foto varias vezes, as imagens deveriam seguir um padrão de uniformidade, textura e iluminação. Esse sistema perfeito se mostrou especialmente difícil de se encontrar, uma vez que uma simples troca de posição resultava em mudanças indesejadas. Essa configuração ideal não foi alcançada, contudo, foi possível encontrar uma que mais se aproximasse da mesma. Para isso foram feitos inúmeros testes com variadas configurações.

O software Matlab foi utilizado para desenvolver todos os procedimentos do trabalho. Um dos motivos pela sua escolha foi a possibilidade de utilizar suas *Toolboxes*, que são funcionalidades implementadas previamente pela própria equipe do software. A *Toolbox* utilizada foi de estatística e Aprendizado de Máquina. A figura 8 mostra as etapas do projeto como um todo. Essas etapas serão melhor discutidas nas seções seguintes do trabalho.

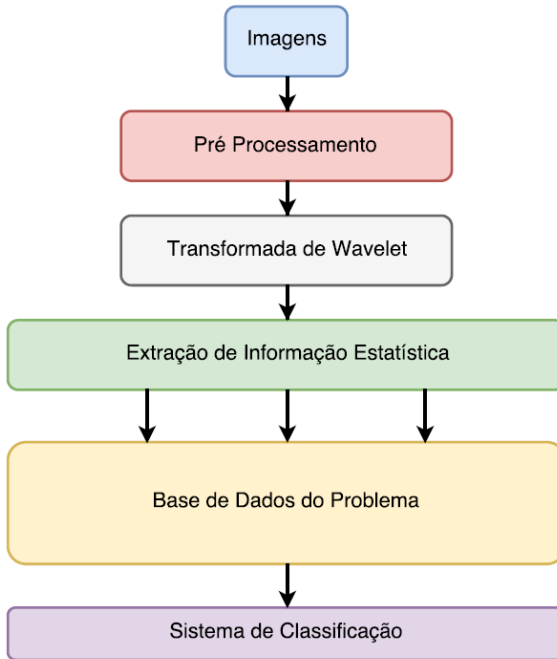


Figura 8 – Etapas da montagem da base de dados até o classificador.

4.1 AQUISIÇÃO E TRATAMENTO DAS IMAGENS

A parte que torna o processo como um todo melhor ou pior é a de aquisição das imagens. Tanto o instrumento de aquisição, ou seja, a câmera nesse caso e até mesmo posição e iluminação podem interferir nesse aspecto. A câmera utilizada no experimento foi a de um celular com resolução máxima de 3264 x 2448 pixels, abertura de sensor de F/2 de 8 Mega pixels.

Desse modo, para uma melhor qualidade do experimento, decidiu-se por realizar a aquisição num ambiente controlado. Isso significa que as imagens não terão elementos externos atuando sobre elas. Esses elementos podem ser variados, mas o principal é a iluminação, que impacta diretamente na qualidade das imagens.

Decidiu-se então utilizar uma caixa para isolar uma determinada

área do tecido. Dentro da caixa foi colocado um sistema de iluminação próprio, assim, nenhuma iluminação externa foi necessária. Contudo, mesmo com esse sistema que isola uma determinada parte do tecido, foi encontrado dificuldades na análise da imagem. Para reduzir os efeitos de imperfeições, tais como dobras e sujeira no tecido, foi primeiramente analisado partes, em teoria, perfeitas do tecido. A imagem inicial do tecido é mostrado na figura 9.



Figura 9 – Imagem de controle inicial testada

O maior problema nessa primeira abordagem foi a diferença entre os exemplos bons. Na extração de características ficou evidente que o protótipo de aquisição de imagem não estava 100% evitando os ruídos, mesmo por movimentação da caixa ou iluminação. Assim, foi decidido analisar somente uma porção da imagem, o centro, onde existe menos ruído. Essas constatações foram feitas através de testes iniciais para a configuração ideal. Desse modo, a imagem foi recortada e somente depois dessa etapa é que a extração acontece. Isso melhorou o desempenho do sistema como um todo. Outro fato importante resultante do corte, foi a diminuição do tempo computacional, já que a imagem foi drasticamente reduzida. A imagem que foi utilizada como padrão bom é mostrada na figura 13.

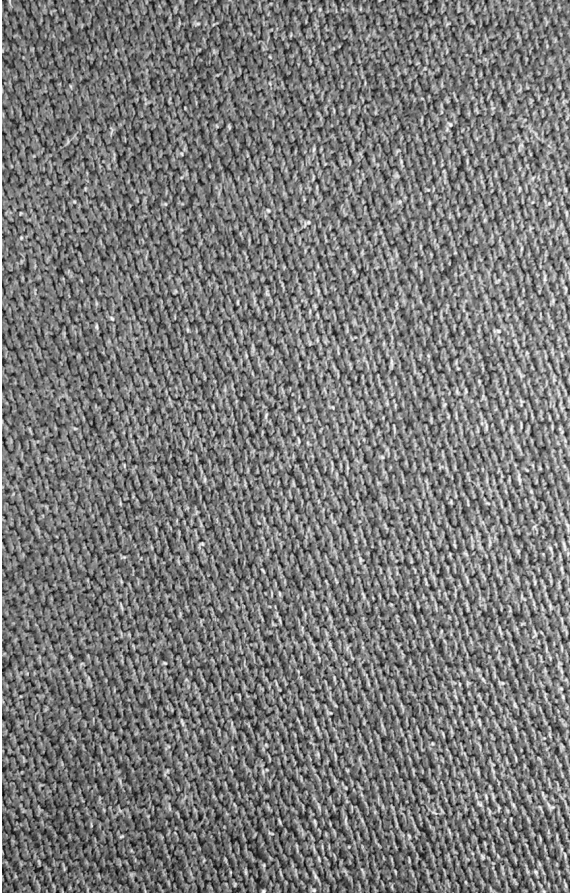


Figura 10 – Imagem Cortada

É possível ver claramente que as imperfeições presentes na imagem inicial foram evitadas a partir do corte. É necessário destacar que recortar a imagem foi necessário neste experimento devido a dificuldade de criar um sistema de iluminação capaz de fornecer imagens de uniformidade semelhante umas as outras. Numa aplicação industrial, seria necessário construir um equipamento muito mais robusto, com sistema de iluminação ideias, ou próximos a isso.

4.2 USO DA TRANSFORMADA DE WAVELET

Como já mencionado, a implementação da transformada de *Wavelet* de duas dimensões, se utiliza de bancos de filtros, tanto para decompor, quanto para recompor a imagem, se a intenção for fazer a compressão da imagem. Aqui, a transformada será usada por dois motivos principais, primeiro para reduzir a quantidade de dados de análise, por consequência a diminuição do custo computacional agregados. E em segundo lugar, mas com relevância maior, a extração de características. A figura 11 mostra um esquema da transformada utilizando filtros. A redução no tamanho se dá pelo fato de ser feita uma amostragem nas linhas, pegando somente os índices pares. O mesmo ocorre para as colunas. Aqui fica evidente a estrutura em cascata dessa implementação, já que a partir do sinal de Coeficientes de Aproximação CA, é possível inseri-lo como uma nova entrada, em X, fazendo assim toda a decomposição novamente.

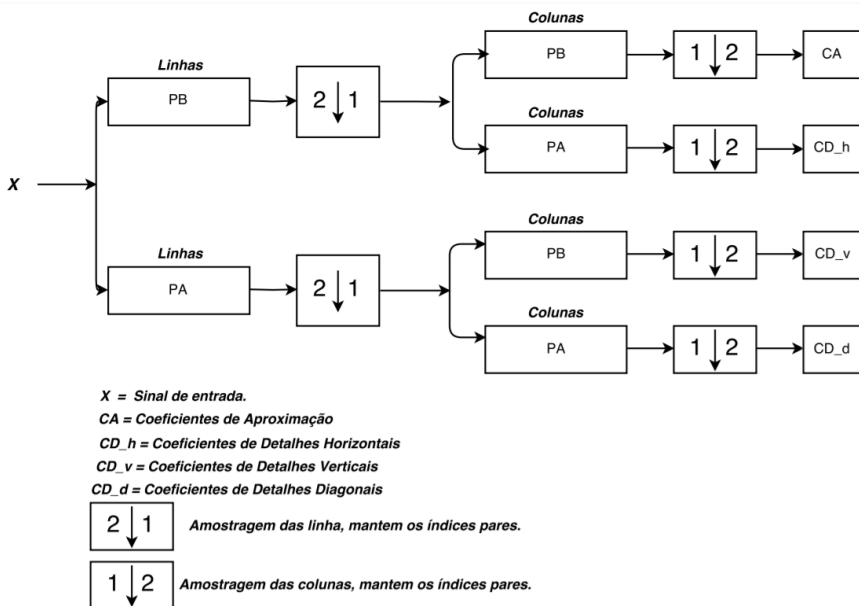


Figura 11 – Esquema da transformada de *Wavelet* de duas dimensões com banco de filtros.

Para o presente trabalho foi utilizado uma *Wavelet* de um nível da família de Daubechies. A *db1* ou *haar*, da como resultado uma matriz com a metade do número de índices nas colunas e nas linhas. Essa matriz é utilizada então para montar a base de dados desejada. Foram testadas outra *Wavelets*, também da família de Daubechies, contudo, nenhuma apresentou resultado melhor que a *db1*.

A saída da aplicação desses parâmetros na transformada de *Wavelet* pode ser vista na figura 12. Olhando a imagem, é difícil detectar qualquer detalhe nos coeficientes, exceto no do filtro passa baixa nas linhas e colunas (primeira imagem, na primeira linha e coluna da figura 12), onde fica evidente o furo no tecido. Por isso se faz necessário usar métodos estatísticos para retirar dados que serão usados como características pelo classificador.

A segunda imagem (primeira linha, segunda coluna), representa os coeficientes de detalhes horizontais. Já na terceira imagem (linha 2, coluna 1), tem-se os coeficientes de detalhes verticais. Por último, na última imagem apresentada nessa figura (linha 2, coluna 2), é apresentado os coeficientes de detalhes diagonais.

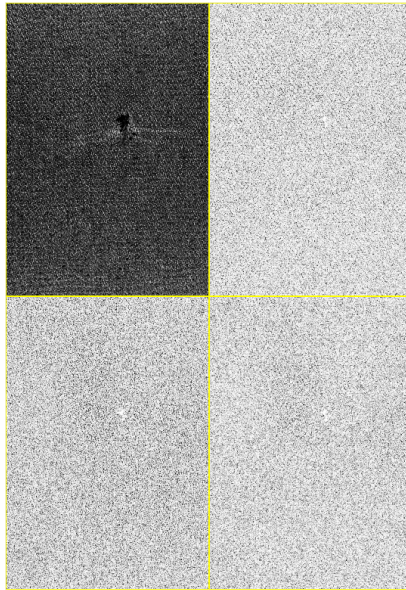


Figura 12 – Saída da transformada.

4.3 BASE DE DADOS

Como frisado anteriormente, a base de dados utilizada foi criada a partir das imagens provenientes da aquisição. Essas imagens passaram por uma seleção previa de modo a retirar imagens que não representassem as classes em questão. Esses imagens que são na verdade ruído para o modelo e tem como característica principal falta de foco. Isso resulta em diferenças significativas na etapa de processamento da *Wavelet*. Assim, com as imagens prontas, o próximo passo foi colocar o rótulo em cada imagem. Etapa essa importante no Aprendizado Supervisionado, já que são esses rótulos que serão usados pelo classificador para gerar o modelo. Essa classificação previa das imagens foi feita por uma pessoa capacitada da empresa Loop Jeans que trabalha com os tecidos diariamente e pode reconhecer um tecido com defeito ou não.

A partir disso pode-se então aplicar a transformada de *Wavelet* às imagens, ela da como resultado os coeficientes, antes já explicados. Com os coeficientes em mãos, é possível realizar diferentes abordagens para a criação de um vetor de características. Para esse trabalho foi decidido utilizar dados estatísticos de média, variância, energia e entropia. Como dito nas seções anteriores, a abordagem utilizada é considerada espectral, contudo, dados estatísticos são usados para uma maior eficiência do classificador. Tanto na parte de tempo computacional, quanto na precisão. As equações da média, da variância, da energia e da entropia são mostradas nas equações 4.1, 4.2, 4.3 e 4.4, respectivamente:

$$V_1 = \frac{1}{n} \sum_{i=1}^n x_i \quad (4.1)$$

$$V_2 = \frac{1}{n} \sum_{i=1}^n (x_i - V_1)^2 \quad (4.2)$$

$$V_3 = \sum_{i=1}^n x_i^2 \quad (4.3)$$

$$V_4 = \sum_{i=1}^n \log(x_i^2) \quad (4.4)$$

onde n representa o número total de dados do conjunto e x_i é um valor de uma posição do conjunto.

Estes dados estatísticos foram computados para cada um dos

coeficientes. Como se trata de uma imagem, os coeficientes de aproximação CA resultante do filtro passa baixa nas linhas e colunas é uma das matrizes computadas. Os Coeficientes de detalhe CD são resultado do filtro passa baixa e passa alta aplicados às linhas e colunas e resultam em três matrizes. Cada uma dessas matrizes terá seus próprios dados estatísticos. Isso resulta que o vetor de característica terá 16 elementos.

Esses vetores concatenados formam uma matriz, que é a base de dados usada para realizar o treinamento e classificação do problema. As linhas da matriz representam os exemplos, já as colunas representam as características de cada exemplo. Esse conjunto representa a base de dados do problema. Os dados podem ser testados por diferentes classificadores, com diferentes técnicas e métodos.

5 RESULTADOS OBTIDOS

Esta seção apresenta, separadamente, os resultados de cada classificador. A metodologia utilizada foi a mesma para todos eles, tanto a base utilizada, quanto o número de iterações para o otimizador. Foram realizadas 1000 iterações para cada classificador. Dentre os resultados analisados tem-se o tempo de treinamento, tempo de predição e análises através da matriz de confusão. Na iteração de otimização é realizado o treinamento e o teste com a validação cruzada, o algoritmo usado é o *k-fold-cross validation*. Essa técnica para k com valor 10 é muito utilizada em vários trabalhos. A técnica ainda tem a grande vantagem em relação à técnicas que usam uma determinada parte do conjunto de dados para teste e outra para treinamento. Aqui, no *k-fold*, todo o conjunto de dados é utilizado, evitando casos de superajustamento dos dados a determinada parte do conjunto.

Como mencionado, a base de dados tem três tipos de defeito e ainda um tecido sem nenhuma falha, chamado de controle. As imagens ilustrando os tipos de defeitos e o padrão sem defeitos estão no Anexo A. A base não é proporcional entre os tipos por dois motivos. O primeiro é que o tecido utilizado tinha muito mais partes boas do que ruins e a segunda, que tem relação com a primeira, é que é necessário fazer com que o modelo se ajuste a identificar mais padrões bons do que ruins. Isso é explicado pelo fato de que defeitos insignificantes não são considerados na produção, pois diminuem a mesma consideravelmente. A tabela 2 a seguir mostra a proporção.

Classe	Tipo	Quantidade	Porcentagem
1	Bons	197	41.13%
2	Furo	122	25.47%
3	Fio Faltante	86	17.95%
4	Mancha	74	15.45%
	Total	479	100%

Tabela 2 – Porcentagens e quantidades dos tipos de defeitos. Tabela elaborada pelo autor.

5.1 RESULTADOS PARA O KNN

Como mencionado na seção de que tratava dos classificadores, o KNN é uma técnica simples, de treinamento rápido, mas que muitas vezes apresenta bons resultados. A acurácia média encontrada para o KNN foi de aproximadamente 91%. Um resultado bastante expressivo, principalmente se tratando de uma técnica simples como esta. A otimização do KNN pode ser vista na figura 13.



Figura 13 – Otimização do KNN.

Como pode-se observar o uso da otimização foi justificado, uma vez que a função objetivo saiu de 0.39 para chegar ao valor de 0.089, obtendo assim, uma melhoria significativa de cerca de 50%. Fica claro que em determinado ponto, entre a iteração 300 e 400, chegou-se a esse valor. Contudo, permitir que algoritmo rodasse 1000 iterações se fez necessário para manter o mesmo padrão entre todos os classificadores. Outro ponto importante, também abordado anteriormente, é a dificuldade de encontrar esses parâmetros que melhoram o desempenho do classificador para determinada base de dados. A tabela 3 mostra os parâmetros que o otimizador encontrou após realizar as iterações.

A tabela 4 mostra alguns dados relativos tanto a otimização, quanto ao classificador em si. Esses resultados serão depois melhor discutidos, comparando com o resultado dos outros classificadores. Aqui ressaltasse o fato de tempo de otimização, que foi de 618.86 segundos. Esse tempo pode parecer muito se num primeiro momento se pensar em uma indústria que precisa de velocidade de produção. Contudo, esse treinamento é realizado apenas uma vez para a produção de vários quilômetros de um mesmo tipo de tecido. Desse modo, perder um

Número de Vizinhos	2
Função de Distância	Mehalanobis
Peso da Distância	Raiz Inversa
Expoente	Não
Padronizar	Falso

Tabela 3 – Parâmetros encontrados pelo Otimizador.

pouco mais de tempo nessa etapa pode fazer com que o sistema tenha uma acurácia maior, se isso fosse o desejado.

Outra importante métrica apresentada nesta tabela é o tempo de predição, que nesse caso foi de 1800 predições por segundo. Esse número é mais do que suficiente para poder usar o algoritmo na indústria. É claro que esse número serve apenas como base, uma vez que a implementação para uma aplicação real teria de ser diferente da apresentada neste trabalho. O tempo de treinamento do modelo também é apresentado, 5.59 segundos, um tempo muito bom, que condiz com a implementação do KNN, onde a etapa de treinamento consiste apenas em armazenar os exemplos.

Tempo de Treinamento(s)	5.59
Acurácia	91%
Tempo de Predição	1800 predições/seg
Objetivo mínimo	0.089
Tempo de Otimização(s)	618.86

Tabela 4 – Resultados gerais do KNN.

Considerada uma das melhores análises para avaliar um classificador, a matriz de confusão na figura 14 mostra as taxas de verdadeiros positivos e falsos negativos de cada classe. As linhas da matriz representam as verdadeiras observações, enquanto que as colunas representam as predições feitas pelo classificador. Desse modo, pode-se perceber que para a classe 1, que é o padrão sem defeitos, chamado de bons, apresentou uma taxa de verdadeiros positivos de 96%, que foi o valor mais alto. Isso equivale a dizer de que de todos os exemplos do padrão bom, apenas aproximadamente 5% não foram corretamente classificadas. O restante está distribuído nas outras classes, esse valor pode ser encontrado na ultima linha da matriz em cinza. O mesmo pensamento pode ser feito para as outras classes, onde o pior desempenho pode ser visto na classe 3 que apresentou taxa de verdadeiros positivos de 82.6%.

Matriz de confusão

Classificação	1	189 39.5%	5 1.0%	10 2.1%	8 1.7%	89.2% 10.8%
	2	4 0.8%	111 23.2%	5 1.0%	2 0.4%	91.0% 9.0%
	3	4 0.8%	5 1.0%	71 14.8%	1 0.2%	87.7% 12.3%
	4	0 0.0%	1 0.2%	0 0.0%	63 13.2%	98.4% 1.6%
		95.9% 4.1%	91.0% 9.0%	82.6% 17.4%	85.1% 14.9%	90.6% 9.4%
	1	2	3	4	Saídas verdadeiras	

Figura 14 – Matriz de confusão para o KNN.

Uma interpretação desses resultados pode ser de que como o número de observações do padrão 1 é maior, ele realmente tende a acertar mais nesse padrão. Da mesma forma que a classificação do tipo 1 é a que o classificador mais erra, observado em cinza na primeira linha na ultima coluna, com um valor de 89.2%, a análise pode parecer confusa, mas deve-se atentar de que são coisas distintas. Enquanto uma analisa o quanto o classificador acerta com relação as classificações, chamada de taxa de predições certas, a outra análise se refere o quanto dos realmente verdadeiros ele acertou.

5.2 RESULTADOS PARA O BAYES INGÊNUO

Contrastando com o resultado do KNN, o classificador baseado em Bayes teve resultados ruins. Com uma acurácia média de 58.9%, o classificador esteve abaixo do esperado. Esse resultado confirma a inobservância anterior em outros problemas da qualidade deste classificador. Vale lembrar que essa implementação do Bayes Ingênuo é uma das mais simples, que não considerada a dependência entre as características.

Quando se trata de otimização, a primeira estimativa do otimizador foi de pouco mais de 0.42, enquanto que a melhor foi de cerca de 0.41. Esse resultado pode ser visto na figura 15. Mesmo sendo um ganho pequeno em relação ao anterior do KNN, ainda é considerável, mostrando mais uma vez a importância de se usar um método de otimização para os parâmetros possíveis. Diferentemente do que ocorreu no KNN, o otimizador melhorou o desempenho do classificador por aproximadamente 800 iterações, evidenciando a importância de fazer um número maior de iterações.

Na tabela 5 podem ser vistos os melhores parâmetros encontrados para o classificador de Bayes. Mesmo com os melhores parâmetros encontrados após 1000 iterações, os resultados não foram bons. Como mencionado anteriormente, a mudança na implementação do algoritmo seria necessário, implementando dependência entre os atributos. De forma geral, fica descartado o uso deste algoritmo.

Parâmetro	Escolha
Distribuição	Kernel
Largura	1.59×10^{-14}
Kernel	Triangular

Tabela 5 – Parâmetros encontrados para o Bayes Ingênuo.

Os resultados apresentados na tabela 6 ajudam a melhorar a avaliação do desempenho deste classificador. Primeiramente, percebe-se que o tempo de treinamento foi similar ao encontrado anteriormente, neste, cerca de 7.99 segundos. O tempo total de otimização foi consideravelmente maior, fato que não interfere na escolha do otimizador, pelo motivo já elucidado anteriormente. Ele ainda é capaz de realizar aproximadamente 840 predições por segundo, número inferior ao encontrado no KNN, mas que pode ser considerado como razoável. A métrica mais importante, a acurácia, como já foi mencionado acima, ficou longe do desejado, logo, esse classificador seria descartado para utilização, sua precisão foi muito abaixo do que é necessário para uma indústria. Foi importante a análise deste algoritmo para mostrar que nem todo classificador responde bem a determinado problema.

As taxas de acerto são apresentadas na figura 16. Similar ao que aconteceu no KNN, a classe com maior número de verdadeiros positivos foi a classe 1, dos tecidos considerados sem defeito. Contudo, as outras classes apresentaram resultados muito ruins em acertos de verdadeiros positivos. Se for analisado a taxa de classificações corretas de cada classe fica evidente que a classe que mais apresenta erros é a classe 1,

Tempo de Treinamento(s)	7.99
Acurácia	58.7%
Tempo de Predição	840 predições/seg
Objetivo mínimo	0.4071
Tempo de Otimização(s)	2532.2

Tabela 6 – Resultados gerais do Bayes Ingênuo.

fato já esperado. De todas as classificações da classe 1, a porcentagem de acerto foi de apenas 51.2%, onde a classe que ele mais erra é a 2, de furo, onde existem diferenças mínimas de características gerais no tecido.



Figura 15 – Otimização do Bayes Ingênuo.

5.3 RESULTADOS PARA O SVM

Dentre todos os classificadores usados neste trabalho, o mais robusto, em teoria, é o SVM. Usado com êxito em vários trabalhos de classificação, o SVM apresentou acurácia média de 89.4%. Resultado próximo ao KNN e melhor do que o Bayes Ingênuo. Quanto à otimização, similar ao que aconteceu com o KNN, o mínimo foi encontrado nas primeiras iterações como pode ser visto na figura 17. A variação da melhor e da pior estimativa foi grande, o mínimo foi de 0.1002, já o pior resultado, para essas estimativas, foi de cerca de 0.59. Um ganho considerável, a acurácia do mínimo seria cerca de 41%, já a do máximo seria de 89,98%, mostrando assim um ganho de aproximadamente 48%.

Matriz de confusão

Classificação	1	191 39.9%	83 17.3%	61 12.7%	38 7.9%	51.2% 48.8%
	2	3 0.6%	32 6.7%	1 0.2%	1 0.2%	86.5% 13.5%
	3	0 0.0%	6 1.3%	24 5.0%	0 0.0%	80.0% 20.0%
	4	3 0.6%	1 0.2%	0 0.0%	35 7.3%	89.7% 10.3%
		97.0% 3.0%	26.2% 73.8%	27.9% 72.1%	47.3% 52.7%	58.9% 41.1%
		1	2	3	4	Saídas Verdadeiras

Figura 16 – Matriz de confusão para o classificador de Bayes.

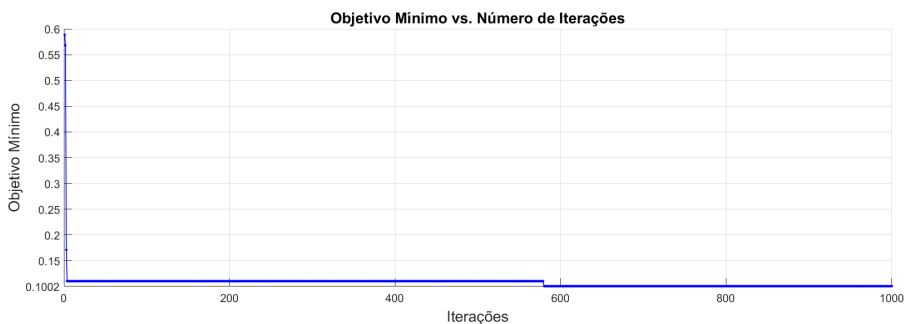


Figura 17 – Otimização do SVM.

A tabela 7 mostra os parâmetros do classificador encontrados para o SVM com a otimização. A codificação é um parâmetro que existe para suprir uma deficiência do SVM, que é um algoritmo de classificação binário. Neste presente problema, tem-se 4 classes possíveis, então é necessário adaptar o classificador de modo que ele aceite mais classes, isso é resolvido com esse parâmetro de codificação. Um contra

um é o tipo escolhido para este problema, que poderia ser também um contra todos. A diferença, como está implícita no nome, é a abordagem empregada quanto a diferenciação das classes. O kernel escolhido foi o Gaussiano, um parâmetro interessante de ser avaliado é o Padronizar, ou normalização dos dados. No KNN esse parâmetro foi falso, enquanto que para o SVM foi verdadeiro, essa escolha para o KNN foi curiosa, uma vez que o mesmo utiliza uma medida de distância e grandes variações nos dados deveriam prejudicar o comportamento do mesmo, ainda assim o parâmetro ótimo diz que não deve padronizar os dados.

Parâmetro	Escolha
Codificação	Um contra Um
<i>BoxConstraint</i>	940.68
Kernel	Gaussiano
Escala do Kernel	3.76
Padronizar	Verdadeiro

Tabela 7 – Parâmetros encontrados para o SVM.

Como já foi mostrado, a acurácia média ficou por volta de 90%, resultado considerado bom. O tempo de treinamento do classificador foi de 3.48 segundos, enquanto que o número de classificações por segundo é de cerca de 2500. O tempo total de otimização foi de 4000.8 segundos, o maior tempo até agora, explicado pelo fato de o SVM ser uma técnica baseada em kernel, muitas vezes custosa computacionalmente. Outro fator que degrada o sistema é o fato de ser uma classificação com mais de duas classes, já que o SVM é um classificador binário. Os resultados de desempenho são mostrados na tabela 8.

Tempo de Treinamento(s)	3.48
Acurácia	89.8%
Tempo de Predição	2500 predições/seg
Objetivo mínimo	0.1002
Tempo de Otimização(s)	4000.8

Tabela 8 – Resultados gerais do SVM.

O resultado da matriz de confusão, apresentado na figura 18, é parecido com o encontrado no KNN. A taxa de verdadeiros positivos para a classe 1 foi de 91.4%, pouco abaixo do encontrado no KNN. Um resultado que chama a atenção é a taxa de verdadeiros positivos da

classe 4 (última linha da tabela, na quarta coluna), com um acerto de 97.3%, com apenas 2 erros dentre os exemplos. Resultado parecido é encontrado na taxa de predições corretas da classe 4 (quarta linha da tabela, última coluna), onde o acerto foi de quase 100%, com apenas um erro de classificação. Difícil precisar o motivo dessa alta taxa de acertos para a classe 4 nesse classificador, o fato é que a classe de manchas (classe 4), é das mais diferentes em termos de características.

Matriz de confusão

Classificação	1	180 37.6%	12 2.5%	9 1.9%	0 0.0%	89.6% 10.4%
	2	6 1.3%	104 21.7%	5 1.0%	2 0.4%	88.9% 11.1%
	3	11 2.3%	5 1.0%	72 15.0%	0 0.0%	81.8% 18.2%
	4	0 0.0%	1 0.2%	0 0.0%	72 15.0%	98.6% 1.4%
		91.4% 8.6%	85.2% 14.8%	83.7% 16.3%	97.3% 2.7%	89.4% 10.6%
	1	2	3	4	Saídas verdadeiras	

Figura 18 – Matriz de confusão para o SVM.

5.4 RESULTADOS PARA A ÁRVORE DE DECISÃO

O classificador de Árvore de Decisão é uma técnica simples, no sentido de entendimento de funcionamento, embora sua implementação envolva criação de árvores e técnicas de poda. Ele apresentou resultados razoáveis, superando o melhor caso da classificação realizada por um ser humano, sua acurácia foi de 83%. Esse resultado é muito bom se for considerado outros aspectos, como por exemplo, sua interpretabilidade que é considerada alta. A figura 19 mostra o resultado da otimização neste classificador. O otimizador pouco após a iteração 200 encontrou o modelo ótimo para 1000 iterações.

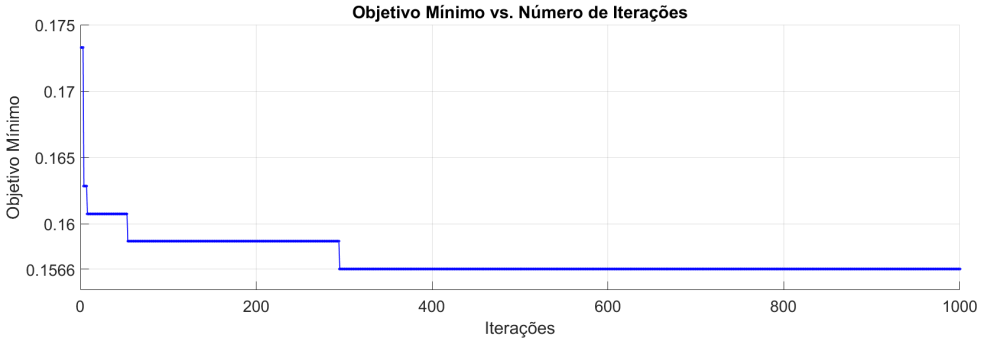


Figura 19 – Otimização da Árvore de Decisão.

A Árvore de decisão para esta implementação possui poucos parâmetros. Os melhores são apresentados na tabela 9.

Parâmetro	Escolha
Tamanho mínimo da folha	5
Número máximo de divisões	190
Critério de divisão	Desvio

Tabela 9 – Parâmetros encontrados para a Árvore de Decisão.

A grande vantagem da Árvore de Decisão em relação à outros algoritmos é sua velocidade, tanto de predição, quanto de treinamento. Esse valores, mostrados na tabela 10, são os melhores entre todos os classificadores aqui estudados. Fazer 4400 predições por segundo mostra que esse classificador se encaixa na resolução do problema aqui abordado, contudo, sua acurácia não foi tão alta, o que pesa contra na avaliação de desempenho, afinal, o objetivo é ter um número de acertos o mais alto possível sem que o modelo esteja super ajustado. Em consequência dos tempos de predição e treinamento, o tempo de otimização total também é o melhor, cerca de 589 segundos.

Tempo de Treinamento(s)	2.18
Acurácia	83%
Tempo de Predição	4400 predições/seg
Objetivo mínimo	0.1566
Tempo de Otimização(s)	589.5

Tabela 10 – Resultados gerais da Árvore de Decisão.

Como já foi mencionado, a taxa de acertos foi ligeiramente baixa em relação ao SVM e ao KNN. Analisando a matriz de confusão na figura 20, pode-se ver que a maior taxa de acerto de verdadeiros positivos é também da classe 1, cerca de 91%. Enquanto que a menor taxa de verdadeiros positivos é da classe 3, com 67.4%. Diferentemente do que ocorre com a taxa de classificações corretas, onde o valor mais alto foi para a classe 4, com 91.8%. A Árvore de decisão é uma técnica muito boa para problemas de classificação, contudo, ela pode gerar super ajustamento dos dados, esse problema pode ser corrigido usando Florestas Randômicas.

Matriz de confusão

Classificação	1	179 37.4%	14 2.9%	13 2.7%	3 0.6%	85.6% 14.4%
	2	5 1.0%	94 19.6%	15 3.1%	4 0.8%	79.7% 20.3%
	3	11 2.3%	10 2.1%	58 12.1%	0 0.0%	73.4% 26.6%
	4	2 0.4%	4 0.8%	0 0.0%	67 14.0%	91.8% 8.2%
		90.9% 9.1%	77.0% 23.0%	67.4% 32.6%	90.5% 9.5%	83.1% 16.9%
	1	2	3	4	Saídas Verdadeiras	

Figura 20 – Matriz de confusão para a Árvore de Decisão.

5.5 COMPARAÇÃO ENTRE OS CLASSIFICADORES E DISCUSSÃO DA APLICAÇÃO

Quando se trata de acurácia, 3 dos 4 classificadores apresentaram valores considerados adequados para o sistema de classificação, a figura 21 mostra esse resultado. O objetivo de um classificador é chegar o mais próximo possível de 100% de acerto, tarefa essa que depende muito do tipo do problema. Como já foi mencionado, uma pessoa consegue, no melhor caso, encontrar cerca de 80% dos defeitos nos tecidos. Desse modo, o KNN, o SVM e a Árvore de Decisão se encaixariam nesse requisito.

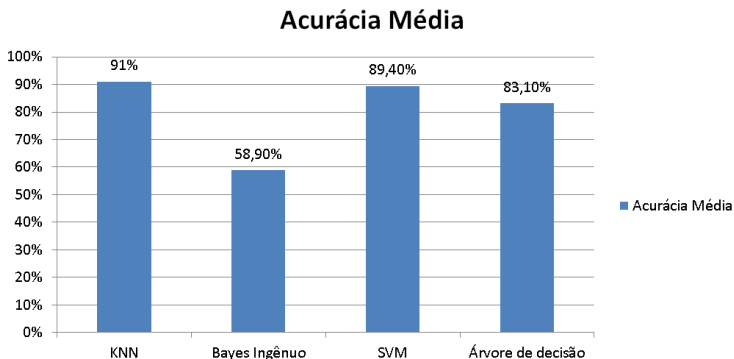


Figura 21 – Acurácia média dos classificadores.

Além da acurácia, outros fatores, com relação a acertos de classificação, devem ser levados em conta e são importantes na avaliação do classificador. A taxa de acertos dos verdadeiramente positivos, mostrada na figura 22, mostra o percentual de acertos de cada classe, dentre os exemplos que eram realmente da mesma, comparando o desempenho dos classificadores. Nesta análise fica mais evidente que o KNN tem desempenho melhor, ou próximo do melhor em quase todas as classes. Por essa análise talvez o SVM parecesse ser uma boa escolha e também fica claro que o desempenho do Bayes Ingênuo foi muito bom na classe 1, mas foi pífio em todas as outras classes, o que prejudicou bastante o desempenho geral do classificador. Em relação à Árvore de Decisão, a classe 3 para esse classificador foi o responsável por prejudicar o desempenho, já que o restante das classes foram próximas a 80% de acerto.

A seguir, é analisado a taxa das classificações corretas dentro de

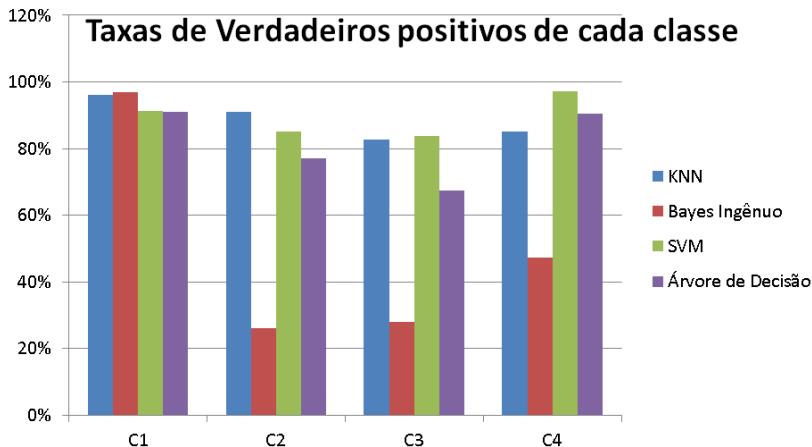


Figura 22 – Comparação das taxas de verdadeiros positivos para cada classe.

cada classe, mostrado na figura 23. Essa taxa mostra dentre as classificações que o classificador fez, qual o percentual ele acertou. Diferentemente da taxa de verdadeiros positivos, aqui são avaliadas as classificações. Uma primeira análise no classificador Bayesiano evidência que ele erra muito ao classificar os exemplos na classe 1, o que degradou completamente sua acurácia. Esse resultado pode ser interpretado que o classificador acha que os exemplos são da classe 1, mas na realidade são de outra, ocasionando em um erro de classificação. O KNN por outro lado, classifica muito bem quase todas as classes, em especial a classe 4, que teve quase 100% de acerto. O SVM teve comportamento semelhante, mas ficou um pouco abaixo. O mesmo pode ser analisado para a Árvore de Decisão. Um resultado interessante é que na classe 4 todos os classificadores se comportaram bem, o que evidência provavelmente sua maior variação de características.

Por fim, a análise pode ser complementada com a avaliação não só dos resultados, mas também das características dos classificadores. Como mencionado anteriormente, o tempo de classificação é um fator importante, assim como a interpretabilidade dos resultados. Juntando todos esses fatores, e complementarmente com o que foi apresentado anteriormente, em relação às taxas de acerto, o KNN foi o melhor classificador para esse problema, isso porque ele apresentou a maior taxa

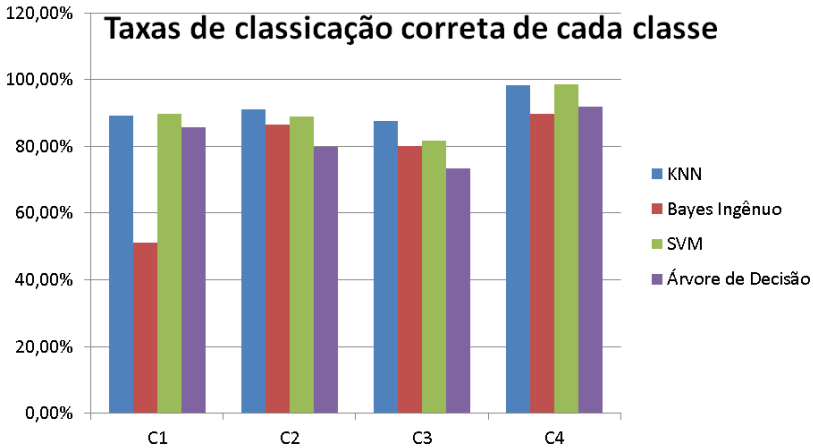


Figura 23 – Comparação das taxas de classificação correta de cada classe.

de acertos e ainda um bom tempo de predição. Neste problema existem poucos exemplos de treinamento, se fossem muitos, a classificação usando o KNN já seria repensada. Outro classificador que apresentou bons resultados foi a Árvore de Decisão, tanto em acertos quanto nos tempos necessários. Por mais que o SVM tenha uma taxa de acertos alta, sua interpretabilidade é considerada complexa, logo ele seria uma terceira opção.

5.6 COMPARAÇÃO COM TRABALHOS RELACIONADOS

A acurácia média do melhor resultado apresentado no presente trabalho, cerca de 91%, é considerada suficiente e capaz de satisfazer uma indústria, substituindo assim, essa parte manual. No trabalho de Abdellah, Ahmed e Slimane (2014), foi encontrada uma taxa de acerto relativamente maior, 96.15%. Para esse resultado foi utilizada um SVM como classificador e técnicas de segmentação de imagem como extrator de características. Levando em consideração que a base de dados do trabalho era relativamente pequena, com apenas 76 exemplos, fica difícil comparar resultados de taxas de acerto. Além disso, a base de dados dos trabalhos usam tecidos diferentes, o que dificulta ainda mais essa análise. No entanto, como as metodologias são semelhantes,

os resultados são robustos.

O trabalho de Rajpoot e Rajpoot (2004) usa *Wavelets* combinado a um SVM de modo a classificar os diferentes tipos de tecidos. O trabalho utiliza taxas de erro nas duas bases de dados como métrica, para uma delas a taxa de erro foi de 13,65%, já para a outra foi de 7,6%. Outro trabalho que se utiliza de *Wavelets* é o de Kim, Lee e Woo (1999). Diferente do que é apresentado nos trabalhos com classificadores, as imagens são testadas com uma *Wavelet* otimizada que ajuda a detectar os defeitos de alta, baixa e média frequência. O autor ainda mostra a diferença significativa entre a relação sinal ruído para as altas e médias frequências.

Combinando duas técnica para encontrar o melhor resultado, o trabalho de Zhang, Jing e Zhang (2015) mostra resultados robustos, com acurácia de 97,6% para o melhor resultado. Essa acurácia foi obtida combinando a extração de características por GLCM e LBP. Os extratores atuando sozinhos tiveram resultados pior do que atuando juntos. Contudo, utilizar mais de um extrator pode aumentar significativamente o tempo de predição. Assim, precisa-se analisar se realmente é necessário a combinação de mais de um método. Outra característica do trabalho de Zhang, Jing e Zhang (2015) é a utilização de uma Rede Neural como classificador. Ressalta-se ainda que os autores utilizaram a base de dados TILDA para realizar o experimento.

Usando também uma Rede Neural, o trabalho de Behera e Mani (2007), tem como extrator a transformada discreta do cosseno. É apresentada a taxa de predições incorretas, dividindo a diferença entre as saídas calculadas da Rede Neural pelas saídas desejadas. A taxa ficou em 1.47, onde o autor destaca que o resultado esta aceitável. A transformada discreta do cosseno se mostrou útil na análise, embora o autor ressalte que em alguns casos pode haver falta de informações para a avaliação.

O trabalho de Jmali, Zitouni e Sakli (2014) tem características diferentes do outro trabalhos. Os autores tentam abordar cada tipo de defeito com técnicas separadas, como por exemplo, contagem da diferença do número de pixel pretos para identificar linhas de falha. Juntamente é utilizada uma Rede Neural para classificar os exemplos. Os resultados do trabalho dão uma ideia clara da alta performance do nosso sistema (JMALI; ZITOUNI; SAKLI, 2014). Não é apresentado taxas de acerto, apenas uma curva de regressão que aproxima os valores desejados pelos classificados pela Rede Neural. Os resultados foram bem próximos dos desejados.

De forma geral, os resultados dos trabalhos mostram que o pre-

sente trabalho obteve resultados considerados satisfatórios. Como evidenciado anteriormente, fica difícil a comparação uma vez que as bases de dados são completamente diferentes. Alguns dos trabalhos ainda não apresentam resultados em forma de taxas de acerto. Outro ponto a se levar conta é o algoritmo de classificação utilizado, muitos dos trabalhos usam uma Rede Neural, uma técnica mais robusta. O melhor classificador neste trabalho foi o KNN, que possui interpretabilidade maior em relação às Redes Neurais.

6 CONCLUSÃO

Sistemas que visam aumentar a qualidade de determinado processo são de extrema importância. Eles aumentam a produção, bem como reduzem custos. Usar um sistema de visão computacional tem tornado esse tipo de tarefa mais fácil, visto que o avanço do Aprendizado de Máquina e do processamento de imagens conseguem imitar, de certa forma, a visão humana. Deste modo, o trabalho mostra que esse tipo de abordagem tem eficácia superior ao que pessoas conseguem realizar a olho nu. Tendo em consideração todas as limitações que são agregadas ao corpo humano e funcionamento da visão.

Os resultados do trabalho ficaram dentro do esperado. O melhor classificador para esse problema foi o KNN, com acurácia média de 91%, pode superar em mais de 10% a taxa de acertos que uma pessoa consegue chegar. Outro ponto importante a se destacar é que a transformada de *Wavelet* se mostrou uma técnica poderosa em análises de imagens, uma vez que a mesma possibilitou uma diferenciação entre as classes. Mesmo assim, seria interessante testar uma combinação com algum método estatístico, aumentando assim a dimensionalidade da base de dados.

Com a acurácia obtida, além dos outros parâmetros, como as taxas de verdadeiros positivos, um sistema de classificação com 4 classes, sendo três de defeitos e um do padrão bom se mostrou possível, com resultado satisfatório. Dessa forma, a indústria teria informação em tempo real para corrigir esses defeitos antes que os mesmos prejudiquem ainda mais a qualidade dos tecidos.

O otimizador de Bayes teve papel fundamental na escolha dos parâmetros dos classificadores. Sem ele, seria muito difícil encontrar os parâmetros que melhor representassem o modelo. Outro ponto importante a respeito do otimizador, foi a rapidez com que esses resultados foram obtidos, em média, cerca 32 minutos. De outro modo teria de ser implementado um método de força bruta, ou manualmente testar algumas combinações, técnica essa que requer conhecimentos específicos por parte do especialista.

Quanto aos resultados gerais dos classificadores, como o tempo de classificação e treinamento, eles se comportaram de maneira esperada, visto que algoritmos mais complexos, como o SVM tem tempo computacional maior em relação a um Árvore de Decisão. Fica difícil tomar como parâmetro esses tempos, uma vez que eles não consideram todo o processo de aquisição da imagem e extração de características. Con-

tudo, eles foram usados para comparar os próprios classificadores.

Por fim, como consideração final, ressalta-se que existem algumas características principais que poderiam melhorar o desempenho em termos de acertos gerais. A principal delas, é o melhor desenvolvimento do sistema de aquisição de imagem, que neste trabalho se limitou a uma caixa com iluminação interna. As variações de imagens, fato indesejado, foram muito difíceis de evitar, assim, a melhora dessa etapa no geral, como sistema de iluminação e câmera, melhoraria bastante o sistema de classificação.

6.1 TRABALHOS FUTUROS

São listados, nesta seção, algumas propostas para trabalhos futuros:

- Desenvolvimento de um sistema de aquisição que garanta uma menor variação intraclasses.
- Combinação de outras técnicas de extração de características.
- Uma etapa de otimização na base de dados.
- Testes usando *Deep Learning*.
- Usar uma Rede Neural como Classificador.
- Implementar um protótipo de baixo custo para o problema.

REFERÊNCIAS

- ABDELLAH, H.; AHMED, R.; SLIMANE, O. Defect detection and identification in textile fabric by svm method. **IOSR Journal of Engineering (IOSRJEN)**, v. 4, n. 12, p. 69–77, 2014.
- AHA, D. W.; KIBLER, D.; ALBERT, M. K. Instance-based learning algorithms. **Machine learning**, Springer, v. 6, n. 1, p. 37–66, 1991.
- AMET, A. L.; ERTUZUN, A.; ERÇIL, A. Texture defect detection using subband domain co-occurrence matrices. In: IEEE. **Image Analysis and Interpretation, 1998 IEEE Southwest Symposium on**. [S.l.], 1998. p. 205–210.
- AZEVEDO, E.; CONCI, A. **Computação gráfica: teoria e prática**. [S.l.]: Elsevier, 2003.
- BEHERA, B.; MANI, M. Characterization and classification of fabric defects using discrete cosine transformation and artificial neural network. **Indian Journal of Fibre & Textile Research**, CSIR, 2007.
- BREIMAN, L. et al. Classification and regression trees. wadsworth & brooks. **Monterey, CA**, 1984.
- CHO, C.-S.; CHUNG, B.-M.; PARK, M.-J. Development of real-time vision-based fabric inspection system. **IEEE Transactions on Industrial Electronics**, IEEE, v. 52, n. 4, p. 1073–1079, 2005.
- COHEN, F. S.; FAN, Z.; ATTALI, S. Automated inspection of textile fabrics using textural models. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 13, n. 8, p. 803–808, 1991.
- DASTOOR, P. et al. Sdas: a knowledge-based framework for analyzing defects in apparel manufacturing. **Journal of the Textile Institute**, Taylor & Francis, v. 85, n. 4, p. 542–560, 1994.
- DIETTERICH, T. G. Approximate statistical tests for comparing supervised classification learning algorithms. **Neural computation**, MIT Press, v. 10, n. 7, p. 1895–1923, 1998.

- DORRITY, J. L.; VACHTSEVANOS, G.; JASPER, W. Real-time fabric defect detection and control in weaving processes. **National Textile Center Annual Report**, p. 113–122, 1996.
- DROBINA, R.; MACHNIO, M. S. Application of the image analysis technique for textile identification. **AUTEX Research Journal**, v. 6, n. 1, p. 40–48, 2006.
- DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern classification**. [S.l.]: John Wiley & Sons, 2012.
- FACELI, K. et al. Inteligência artificial: Uma abordagem de aprendizado de máquina. **Rio de Janeiro: LTC**, v. 2, p. 192, 2011.
- GAJANAN, G. K. Review of defect detection and classification methods for fabric. **International Journal of Scientific & Engineering Research**, 2014.
- GELBART, M. A.; SNOEK, J.; ADAMS, R. P. Bayesian optimization with unknown constraints. **arXiv preprint arXiv:1403.5607**, 2014.
- GONZALEZ, R. C.; WOODS, R. E.; EDDINS, S. L. **Digital Image Processing Using MATLAB®**. [S.l.]: McGraw Hill Education, 2010.
- JAIN, A. K.; DUIN, R. P. W.; MAO, J. Statistical pattern recognition: A review. **IEEE Transactions on pattern analysis and machine intelligence**, IEEE, v. 22, n. 1, p. 4–37, 2000.
- JMALI, M.; ZITOUNI, B.; SAKLI, F. Fabrics defects detecting using image processing and neural networks. In: IEEE. **Information and Communication Technologies Innovation and Application (ICTIA)**. [S.l.], 2014. p. 1–6.
- KIM, S.; LEE, M. H.; WOO, K.-B. Wavelet analysis to fabric defects detection in weaving processes. In: IEEE. **Industrial Electronics, 1999. ISIE'99. Proceedings of the IEEE International Symposium on**. [S.l.], 1999. v. 3, p. 1406–1409.
- KON, A.; COAN, D. C. Transformações da indústria têxtil brasileira: a transição para a modernização. **Revista de economia Mackenzie**, v. 3, n. 3, 2009.
- KUMAR, A. Computer-vision-based fabric defect detection: A survey. **IEEE transactions on industrial electronics**, IEEE, v. 55, n. 1, p. 348–363, 2008.

- MALEK, A. S. **Online fabric inspection by image processing technology**. Tese (Doutorado) — Université de Haute Alsace-Mulhouse, 2012.
- MICHIE, D.; SPIEGELHALTER, D. J.; TAYLOR, C. C. **Machine learning, neural and statistical classification**. [S.l.]: Citeseer, 1994.
- MITCHELL, T. M. Machine learning. 1997. **Burr Ridge, IL: McGraw Hill**, v. 45, n. 37, p. 870–877, 1997.
- OLIVEIRA, H. de. **Análise de Fourier e Wavelets**. [S.l.]: Hélió Magalhaes de Oliveira, 2007.
- QUINLAN, J. R. C4. 5: Programming for machine learning. **Morgan Kaufmann**, v. 38, 1993.
- RAJPOOT, K. M.; RAJPOOT, N. M. Wavelets and support vector machines for texture classification. In: IEEE. **Multitopic Conference, 2004. Proceedings of INMIC 2004. 8th International**. [S.l.], 2004. p. 328–333.
- REBHI, A.; ABID, S.; FNAIECH, F. Fabric defect detection using local homogeneity and morphological image processing. In: IEEE. **Image Processing, Applications and Systems (IPAS), 2016 International**. [S.l.], 2016. p. 1–5.
- RISH, I. An empirical study of the naive bayes classifier. In: IBM NEW YORK. **IJCAI 2001 workshop on empirical methods in artificial intelligence**. [S.l.], 2001. v. 3, n. 22, p. 41–46.
- RUSSELL, S.; NORVIG, P. **Inteligência artificial**. [S.l.]: Elsevier, 2004.
- SCOTT, S. M.; JAMES, D.; ALI, Z. Data analysis for electronic nose systems. **Microchimica Acta**, Springer, v. 156, n. 3-4, p. 183–207, 2006.
- SENGOTTUVELAN, P.; WAHI, A.; SHANMUGAM, A. Automatic fault analysis of textile fabric using imaging systems. **Research Journal of Applied Sciences**, v. 3, n. 1, p. 26–31, 2008.
- SNOEK, J.; LAROCHELLE, H.; ADAMS, R. P. Practical bayesian optimization of machine learning algorithms. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2012. p. 2951–2959.

ZHANG, H. The optimality of naive bayes. **AA**, v. 1, n. 2, p. 3, 2004.

ZHANG, L.; JING, J.; ZHANG, H. Fabric defect classification based on lbp and glcm. **Journal of Fiber Bioengineering and Informatics**, v. 8, n. 1, p. 81–89, 2015.

SUPERVISED Learning Workflow and Algorithms: What is Supervised Learning?. What is Supervised Learning?. [2006]. Disponível em: <https://www.mathworks.com/help/stats/supervised-learning-machine-learning-workflow-and-algorithms>. Acesso em: 05 nov. 2017.

APÊNDICE A - Taxas do classificador KNN

Model 2

True class	Predicted class				True Positive Rate	False Negative Rate
	1	2	3	4		
1	96%	2%	2%		96%	4%
2	5%	90%	4%	1%	90%	10%
3	10%	6%	84%		84%	16%
4	7%	3%		91%	91%	9%

(a) Taxas de verdadeiros positivos para o KNN.

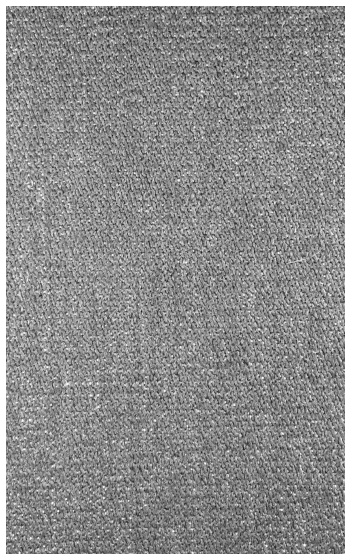
Model 2

True class	Predicted class			
	1	2	3	4
1	90%	3%	5%	
2	3%	92%	6%	1%
3	4%	4%	89%	
4	2%	2%		99%
Positive Predictive Value	90%	92%	89%	99%
False Discovery Rate	10%	8%	11%	1%

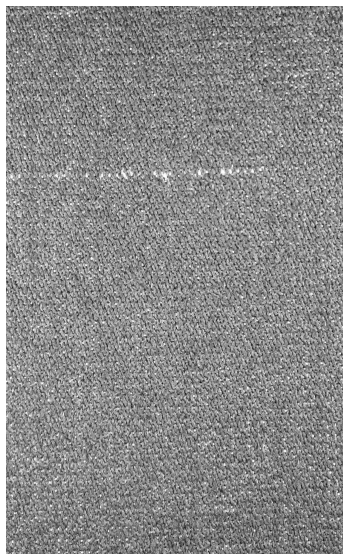
(b) Taxas de classificados corretos para o KNN.

Figura 24 – Taxas do KNN.

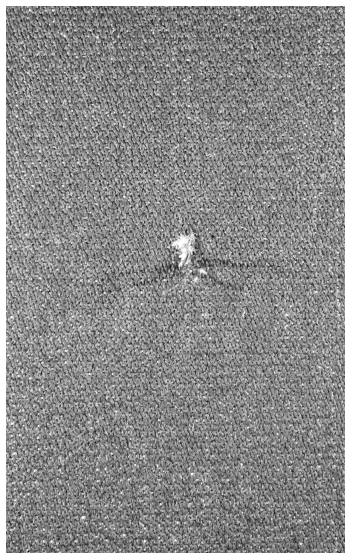
ANEXO A - Imagens dos tipos de defeitos



(a) Sem defeitos



(b) Fio faltante



(c) Furo



(d) Mancha

Figura 25 – Tipos de defeito.