



UNIVERSIDADE FEDERAL DA BAHIA
DEE - DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Pedro Xavier Alcantara

**Classificação de Obstáculos Baseada no
Classificador *k-Nearest Neighbors* Aplicada a
um Robô de Inspeção de Linha de Transmissão**

Dissertação de Mestrado

Salvador

2018



UNIVERSIDADE FEDERAL DA BAHIA
DEE - DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Pedro Xavier Alcantara

Classificação de Obstáculos Baseada no Classificador *k-Nearest Neighbors* Aplicada a um Robô de Inspeção de Linha de Transmissão

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal da Bahia como requisito parcial para obtenção de título de Mestre em Engenharia, modalidade Engenharia Elétrica

Orientador: Prof. André Gustavo Scolari Conceição

Salvador
2018

Ficha catalográfica elaborada pelo Sistema Universitário de Bibliotecas (SIBI/UFBA),
com os dados fornecidos pelo(a) autor(a).

Alcantara, Pedro
Classificação de Obstáculos Baseada no
Classificador k-Nearest Neighbors Aplicada a um Robô /
Pedro Alcantara. -- Salvador, 2018.
89 f. : il

Orientador: André Conceição.
Dissertação (Mestrado - Engenharia Elétrica) --
Universidade Federal da Bahia, Escola Politécnica,
2018.

1. K-Nearest Neighbor. 2. LiDAR. 3. Inspeção de
linhas de transmissão. I. Conceição, André. II. Título.

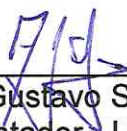
Pedro Xavier Alcântara

**Classificação de Obstáculos Baseada no Classificador K-Nearest Neighbors
Aplicada a um Robô de Inspeção de Linha de Transmissão**

Dissertação apresentada à Universidade Federal da Bahia, como parte das exigências do Programa de Pós-Graduação em Engenharia Elétrica, para a obtenção do título de *Mestre*.

APROVADA em: 19 de Outubro de 2018.

BANCA EXAMINADORA



Prof. Dr. André Gustavo Scolari Conceição
Orientador - UFBA



Prof. Dr. Paulo César Machado de Abreu Farias
UFBA



Prof. Dr. Oberdan Rocha Pinheiro
Senai - Cimatec

Agradecimentos

Essa jornada vem chegando ao fim e isso não seria possível sem o apoio de inúmeras pessoas que merecem meu mais profundo sentimento de gratidão.

Muito obrigado a minha família por ser minha base, meus pais, Humberto, Margareth e meu irmão Vitor, por estarem juntos na minha caminhada desde o primeiro passo. Por propiciar minha condição privilegiada sem a qual não poderia ter completado mais essa jornada, de muitas que já percorri.

Obrigado a minha companheira, Vívian, por todo apoio, compreensão, "puxões de orelhas", paciência e cumplicidade durante esses longos 3 anos. Sem você isso realmente não seria possível de ser completado.

Ao meu orientador André Scolari, pelo encorajamento e supervisão.

A todos os meus colegas do BIR, pelo apoio, sugestões e companheirismo. Todos presentes nesse ambiente de enorme aprendizado e crescimento fazem parte dessa vitória. Em especial a Luís, Branilson, Thiago, Leone, Matheus, Juliana e Marco, por serem parte real e peças construtivas desse trabalho.

Aos meus amigos Diego Cesar e Rafael Saback, por serem referência de competência, amizade e pelo estímulo e suporte.

As minhas cunhadas Larissa e Tais Paim por auxiliarem e darem suporte sempre que preciso, mesmo por muitas vezes não entendendo nada do trabalho.

Enfim, muito obrigado a todos os que participaram e contribuíram de maneira direta ou indiretamente para a conclusão desse mestrado.

“...Be water my friend.”

Bruce Lee

Abstract

Nowadays, our way of life is increasingly more dependent on electric power. This characteristic makes the transmission lines, responsible for making the connection between the generating units and consumers, to be on operating conditions 24 hours a day, 7 days a week. Currently, manned aircrafts are responsible for the inspection of these transmission lines, which is very expensive and extremely dangerous. Given this perspective, robots have become more and more real alternatives to reduce costs and increase safety in inspection activity. In order to seek autonomy in the inspection activity, it is necessary that the robot identify the obstacles in its course, so that it can continue inspecting the line without interruptions. Considering this context, this paper presents an approach to characterize and classify the obstacles found in the line. The applied strategy includes the usage of a LiDAR for data collection, shape features extraction and classification using the K-Nearest Neighbor artificial intelligence algorithm. The approach of this work used two datasets to demonstrate the performance of the proposed method, presenting 100% accuracy when the images had no occlusion and 98.4% with the dataset showing occlusion in the images. Classifier analysis was performed with addition of image noise types salt&pepper and speckle with little change in performance with a minimum performance of 97.6% of accuracy.

Keywords: K-Nearest Neighbor, LiDAR, Powerline Inspection

Resumo

Atualmente, nosso modo de vida esta cada vez mais dependente de energia elétrica. Essa característica faz com que as linhas de transmissão, responsáveis por fazer a conexão entre as unidades geradoras e consumidoras devam estar em condições de operação 24 horas por dia, 7 dias por semana. Normalmente a realização da inspeção das linhas de transmissão é feita por aeronaves tripuladas, condição muito cara e extremamente perigosa. Diante dessa perspectiva, robôs têm se tornado alternativas cada vez mais reais para o barateamento e aumento da segurança dessa atividade de inspeção. Para buscar autonomia nessa atividade, é necessário que o robô identifique os obstáculos no seu percurso para que possa continuar inspecionando a linha sem interrupções. Considerando esse contexto, o presente trabalho apresenta uma abordagem para caracterização e classificação dos obstáculos encontrados na linha. A estratégia abordada utiliza-se de um LiDAR para a coleta de dados, extração de *features* de forma e intensidade e a classificação utilizando o algoritmo de inteligência artificial *k-Nearest Neighbor*. A abordagem desse trabalho utilizou dois *datasets* para demonstrar o desempenho do método proposto, apresentando acurácia de 100% quando as imagens não apresentavam oclusão e 98,4% com o *dataset* que apresenta oclusão nas imagens. Foi realizada análise do classificador com a adição de ruído dos tipos *salt&pepper* e *speckle* com pouca alteração de desempenho, apresentando desempenho mínimo de 97,6%.

Palavras-chave: K-Nearest Neighbor, LiDAR, Inspeção de linhas de transmissão

Lista de ilustrações

Figura 1 – Inspeção de linhas de transmissão realizadas por aeronaves tripuladas.	16
Figura 2 – Realização de manutenção por operador especializado, [DEBENEST et al., 2008].	16
Figura 3 – Manutenção realizada a partir de aeronave, [SENAI - CIMATEC, 2013].	17
Figura 4 – Robôs de Inspeção de Linha de Transmissão.	18
Figura 5 – Obstáculos: amortecedor de vibração (esquerda) e grampo de suspensão (direita).	19
Figura 6 – Protótipo de robô apresentado em [VENTRELLA et al., 2003].	21
Figura 7 – Teste realizado em campo para remoção e instalação de esferas de sinalização, [CAMPOS et al., 2002].	22
Figura 8 – Robô apresentado em [ZHOU et al., 2005], [SARTORI, 2018].	22
Figura 9 – a) LineScout, b) Procedimento de ultrapassagem de obstáculo, [MONTAMBAULT; POULIOT, 2007].	23
Figura 10 – Robô de inspeção híbrido, [KATRASNIK; PERNU; LIKAR, 2008].	23
Figura 11 – Robô de inspeção híbrido, [KATRASNIK; PERNU; LIKAR, 2008].	24
Figura 12 – Processo de ultrapassagem do robô Expliner, [KATRASNIK; PERNU; LIKAR, 2008].	24
Figura 13 – VANT de inspeção juntamente com estação de controle, [RANGEL; KIENITZ; BRANDÃO, 2009].	25
Figura 14 – Teste de robô de inspeção no cabo guarda, [LI; RUAN, 2010].	25
Figura 15 – Robô de inspeção bípede, [WANG et al., 2010].	26
Figura 16 – Modelo 3D de robô de inspeção de linha de transmissão, [GONCALVES; CARVALHO, 2010].	27
Figura 17 – Robô de inspeção de luva de emenda, [LEE; JUNG; CHO, 2011].	27
Figura 18 – Ti se deslocando pela linha de transmissão, [PHILLIPS et al., 2012].	28
Figura 19 – Apresentação conceitual de módulos de sistema de inspeção de linha de transmissão, [GONÇALVES; CARVALHO, 2013].	28
Figura 20 – Robô D311 realizando testes, [II et al., 2014].	29
Figura 21 – Módulos do Robô D311, a) Módulo de Busca, b) Módulo de Tração, c) Módulo de Apoio, [MOURÃO et al., 2015].	30
Figura 22 – PIRO, [MOURÃO et al., 2015].	30
Figura 23 – Apresentação da massa do PIRO de 8,92 kg, [MOURÃO et al., 2015].	31
Figura 24 – Robô Híbrido, [CHANG et al., 2017].	31
Figura 25 – Modelo 3D de estrutura mecânica do robô ELIR, [SARTORI, 2018].	33
Figura 26 – LiDAR : Hokuyo UTM-30LX.	37
Figura 27 – Etapas do processo de classificação.	40

Figura 28 – Representação do: a) objeto, b) classe c) todo o <i>dataset</i> , adaptado de [MEDINA et al., 2013].	43
Figura 29 – Representação esquemática do <i>K-fold cross-validation</i> , [RASCHKA; MIRJALILI, 2015].	44
Figura 30 – Matriz de Confusão, [RASCHKA; MIRJALILI, 2015].	45
Figura 31 – Exemplo de curva ROC com quatro classificadores, [MAJNIK; BOSNIĆ, 2013].	47
Figura 32 – Exemplo de classificação utilizando kNN, [KIM et al., 2017].	48
Figura 33 – Plataforma de Teste.	51
Figura 34 – Pontos de posicionamento dos obstáculos para coleta de imagens.	53
Figura 35 – Amortecedores distorcidos pela vibração da linha de transmissão.	53
Figura 36 – Grampos distorcidos pela vibração da linha de transmissão.	53
Figura 37 – Imagens dos obstáculos e imagens do <i>dataset</i>	54
Figura 38 – Imagens dos obstáculos e imagens do <i>dataset</i> com oclusão.	55
Figura 39 – Fluxograma de geração de imagens.	57
Figura 40 – Etapas do processamento das imagens.	59
Figura 41 – Área dos obstáculos.	62
Figura 42 – Perímetro dos obstáculos.	63
Figura 43 – Intensidade Média dos obstáculos.	64
Figura 44 – Imagens do amortecedor de vibração, da esquerda para a direita: imagem original, imagem binarizada, imagem em escala de cinza apenas do obstáculo.	64
Figura 45 – Imagens do grampo de suspensão, da esquerda para a direita: imagem original, imagem binarizada, imagem em escala de cinza apenas do obstáculo.	65
Figura 46 – Comparação entre média e área dos obstáculos.	65
Figura 47 – Comparação entre média e perímetro dos obstáculos.	66
Figura 48 – Comparação entre área e perímetro dos obstáculos.	67
Figura 49 – Comparação de desempenho com dados normalizados e não normalizados	68
Figura 50 – Curva de validação.	69
Figura 51 – Curva ROC com k e todas as <i>features</i> para classificação do <i>dataset</i> "com oclusão".	72
Figura 52 – Amortecedor de vibração com adição de ruído <i>salt&pepper</i>	73
Figura 53 – Grampo de suspensão com adição de ruído <i>salt&pepper</i>	73
Figura 54 – Amortecedor de vibração com adição de ruído <i>speckle</i>	73
Figura 55 – Grampo de suspensão com adição de ruído <i>speckle</i>	74
Figura 56 – Resultado das etapas de pré-processamento dos obstáculos com inserção de 20% de ruído do tipo <i>salt&pepper</i> , da esquerda para direita: imagem original, imagem binária do obstáculos, imagem em escala de cinza do obstáculo.	75

Figura 57 – Resultado das etapas de pré-processamento dos obstáculos com inserção de ruído do tipo <i>speckle</i> com 0,1 de variância, da esquerda para direita: imagem original, imagem binária do obstáculos, imagem em escala de cinza do obstáculo.	76
Figura 58 – Resultado das etapas de pré-processamento dos obstáculos com inserção de ambos os tipos de ruído, da esquerda para direita: imagem original, imagem binária do obstáculos, imagem em escala de cinza do obstáculo.	77
Figura 59 – Curva ROC com inserção de ruído do tipo <i>salt&pepper</i> com percentual de 5%	77
Figura 60 – Curva ROC com inserção de ruído do tipo <i>salt&pepper</i> com percentual de 20%	78
Figura 61 – Curva ROC com inserção de ruído do tipo <i>speckle</i> com variância de 0,01	78
Figura 62 – Curva ROC com inserção de ruído do tipo <i>speckle</i> com variância de 0,1	78
Figura 63 – Curva ROC com inserção de ruído do tipo <i>speckle</i> com variância de 0,1 e <i>salt&pepper</i> com 20%	79

Lista de tabelas

Tabela 1	– Tabela com Matrizes de Confusão para <i>dataset</i> "sem oclusão".	70
Tabela 2	– Tabela com Matrizes de Confusão para <i>dataset</i> "com oclusão".	71
Tabela 3	– Tabela da média dos valores de acurácia do classificador.	71
Tabela 4	– Tabela da média dos erros de previsão do classificador.	72
Tabela 5	– Tabela da média dos valores de acurácia do classificador com adição de ruído <i>salt&pepper</i>	74
Tabela 6	– Tabela da média dos erros de previsão do classificador com adição de ruído <i>salt&pepper</i>	75
Tabela 7	– Tabela da média dos valores de acurácia do classificador com adição de ruído <i>speckle</i>	75
Tabela 8	– Tabela da média dos erros de previsão do classificador com adição de ruído <i>speckle</i>	76

Lista de abreviaturas e siglas

ACC	acurácia
ERR	taxa de erro ve previsão
TFP	Taxa de Falso Positivo
TVP	Taxa de Verdadeiro Positivo
<i>kNN</i>	k-Nearest Neighbors
ANEEL	Agência Nacional de Energia Elétrica
Aneel	Agência Nacional de Energia Elétrica
AUC	<i>Area under the curve</i>
CEMIG	Companhia Energética de Minas Gerais
CSV	<i>Comma Separeted Value</i>
ELIR	<i>Electrical Inspection Robot</i>
EPRI	<i>Electrical Power Research Institute</i>
EPRI	<i>Eletric Power Reaserch Institute</i>
JPS	Japan Systems Corp.
KEPCO	Kansai Electric Power Corp.
LiDAR	<i>Light Detecttion and Ranging</i>
MB	Megabytes
ms	milisegundos
PIRo	<i>Powerlines Inspection Robot</i>
RNN	<i>rought neural network</i>
ROC	<i>Receiver Operation Characteristics</i>
ROS	<i>Robotic Operation System</i>
SSL	Sistema de Serviço de Linhas de Transmissão
UAV	<i>Unmanned Aerial Vehicle</i>
VANT	Veículo Aéreo Não Tripulado

Lista de símbolos

δ	Letra grega delta
----------	-------------------

Sumário

1	INTRODUÇÃO	14
1.1	Inspeção de Linhas de Transmissão	14
1.2	Objetivo Geral	18
1.3	Objetivos Específicos	19
1.4	Organização do Trabalho	19
2	ROBÔS DE INSPEÇÃO DE LINHA DE TRANSMISSÃO	21
2.1	EIIR (<i>Electrical Inspection Robot</i>)	31
2.2	Técnicas de Identificação de Obstáculos em Linhas de Transmissão	33
2.2.1	Escolha do sensor	35
3	FUNDAMENTAÇÃO TEÓRICA	38
3.1	Classificação de Objetos em Imagens	38
3.2	<i>Features</i> de Forma	40
3.2.1	Área	41
3.2.2	Perímetro	41
3.2.3	Média de intensidade	41
3.3	Classificador	42
3.3.1	Métodos e Métricas de Avaliação	43
3.3.2	<i>k-Nearest Neighbors(kNN)</i>	47
4	METODOLOGIA E EXPERIMENTOS	51
4.1	Estrutura de Teste	51
4.2	Levantamento de <i>Dataset</i>	52
4.3	Processamento das Imagens	57
4.3.1	Pré-Processamento das Imagens e Extração de <i>Features</i>	58
4.3.2	Classificação das Imagens	59
4.3.2.1	Avaliação de Robustez	60
5	RESULTADOS E DISCUSSÕES	62
5.1	Análise das <i>Features</i>	62
5.2	Análise do <i>Dataset</i> no Espaço das <i>Features</i>	63
5.3	Classificação	66
5.3.1	Seleção de Modelo	67
5.3.1.1	Avaliação de normalização	67
5.3.1.2	Avaliação de valores de k	69
5.3.1.3	Avaliação do comportamento das features	69

5.3.2	Avaliação de Robustez	71
6	CONCLUSÃO	81
	REFERÊNCIAS	84

1 Introdução

Nos últimos 30 anos, a robótica tem alcançado grande relevância, principalmente no ambiente industrial. Integrados à linha de produção, esses robôs são capazes de aumentar a produtividade e a repetibilidade da linha, visto que possuem velocidade e precisão. Isso fez com que a produtividade fosse aumentada, juntamente com o aumento de qualidade dos produtos e redução dos custos de produção. A ausência da mobilidade dos manipuladores restringe suas aplicações. Isso abre um espaço variado para soluções que podem ser alcançadas através da aplicação de robôs móveis. Apesar da grande capacidade de locomoção, estes ainda são pouco eficientes quanto a manipulação [SIEGWART; NOURBAKHSH; SCARAMUZZA, 2011].

Além da aplicação em plantas industriais, existe também a necessidade da execução de tarefas em lugares onde a utilização de mão-de-obra humana é difícil, arriscada ou até impossível, como no fundo do mar (a elevadas profundidades) [ALBIEZ et al., 2015], no espaço [GROTZINGER, 2013] ou utilizados em tarefa de localização e resgates de pessoas em catástrofes naturais [MATSUNO; TADOKORO, 2004]. Para realização dessas tarefas, a robótica apresenta-se como alternativa interessante, visto que, há um grande benefício que é a preservação da vida e/ou da saúde das pessoas envolvidas [ROSÁRIO, 2006]. A utilização de robôs móveis para essas aplicações tem sido impulsionada pela melhoria significativa na capacidade de processamento dos computadores e dispositivos eletrônicos além do avanço e redução de preço dos sensores perceptivos.

Uma das tarefas que apresenta elevado nível de periculosidade é a inspeção de linhas de transmissão de alta tensão. Essa atividade é realizada por operadores que caminham pelas linhas a elevadas altitudes ou realizam a inspeção a bordo de aeronaves que sobrevoam bem próximo às linhas. Essas condições extremas favorecem o acontecimento frequente de acidentes, muitas vezes fatais [RANGEL; KIENITZ; BRANDÃO, 2009]. Devido a essas condições, juntamente com a busca pela redução de custos e melhoria da manutenção do sistema elétrico através da inspeção das linhas de transmissão cada vez mais frequentes, a robótica tem sido impulsionada como alternativa para a execução dessa tarefa, sendo realizada de maneira autônoma ou com operação remota [SARTORI, 2018].

1.1 Inspeção de Linhas de Transmissão

Segundo o EMPRESA [2014], no Brasil 17,1% do consumo final de energia é de energia elétrica. Este valor mostra a importância desse tipo de energia para o funcionamento da indústria, dos grandes centros urbanos e da população em geral. O mercado consumidor é de aproximadamente 47 milhões de unidades, sendo, 56% nas regiões Sudeste/Centro-

Oeste, 22% na região Nordeste, 15% na região Sul, 5% na região Norte e 2% em sistemas isolados. Para fazer a interligação entre as unidades geradoras e os centros consumidores, são utilizados cerca de 135.264 km de linhas de transmissão, que devem operar 24 horas por dia, 7 dias por semana, durante todo o ano [MME, 2017].

As linhas de transmissão de energia são formadas por cabos condutores de energia elétrica, isoladores e torres que sustentam os cabos [RANGEL; KIENITZ; BRANDÃO, 2009]. Ainda segundo Rangel, Kienitz e Brandão [2009], “no Brasil, há uma quantidade considerável de linhas de transmissão que já ultrapassou os 40 anos de idade. Com o envelhecimento das linhas de transmissão, a manutenção preventiva é um fator de extrema relevância para garantir o perfeito funcionamento dos sistemas.”. Aliado ao tempo de utilização, os cabos das linhas de transmissão estão sujeitos a intempéries do clima ou a condições do local de instalação como: influência da vegetação, vento, erosão, salitre, chuvas, variação de temperatura, entre outras. Devido às condições citadas e a sua importância, as linhas de transmissão devem receber manutenção frequentemente.

Atualmente, a inspeção de linhas de transmissão é realizada, em sua maioria, por aeronaves tripuladas. Essa atividade tem como objetivo a verificação, de forma preventiva, da integridade física dos componentes da linha em relação a danos que possam comprometer o fornecimento adequado de energia elétrica, como: fissuras nos condutores, corrosão e danificação mecânica, por exemplo. Durante essa inspeção, são avaliados a integridade estrutural das torres, condição dos isoladores, conexões das linhas. Outro tipo de investigação é realizado através da busca de pontos quentes, utilizando-se câmeras térmicas, busca-se por regiões em que haja corrosão em estágio avançado, ruptura ou folga nos componentes. Esses problemas aumentam a resistência dessa região do condutor, ocorrendo, conseqüentemente o aumento da temperatura, sendo assim, detectável por dispositivos de visualização térmica. Para obter essas informações, as aeronaves realizam voos em baixas altitudes e próximo das linhas de transmissão, conforme mostrado na Figura 1. Em determinadas situações, condições climáticas e/ou características geográficas dificultam muito o sobrevoo, o que torna a inspeção uma tarefa bastante arriscada, expondo toda a tripulação à ocorrência frequente de acidentes (RANGEL; KIENITZ; BRANDÃO, 2009 apud OLIVEIRA; LAGES, 2007).

Uma alternativa à inspeção com helicópteros é a utilização de veículos terrestres. Contudo, esse tipo de inspeção é muito limitada, uma vez que boa parte das torres ficam localizadas em áreas de difícil acesso, tornando restrito a capacidade de inspeção da equipe. As inspeções são importantes pois fornecem dados valiosos do estado da linha, o que ajuda os engenheiros a planejar reparos ou substituições necessárias antes da ocorrência de falhas que possam resultar na interrupção do fornecimento de energia elétrica (SARTORI, 2018 apud NAYYERLOO et al., 2009).

Além dos problemas encontrados para realização da inspeção, a manutenção pre-



Figura 1 – Inspeção de linhas de transmissão realizadas por aeronaves tripuladas.

ventiva e corretiva das linhas é uma tarefa ainda mais problemática. Essa atividade apresenta um elevado grau de risco, uma vez que o profissional necessita estar suspenso nas linhas a uma altura de aproximadamente 100 metros do chão, Figura 2, ou a partir de um helicóptero, Figura 3. Na realização da manutenção, também é necessário que a linha seja desligada, causando a sobrecarga de outra linha, ou em alguns casos, interrompendo o fornecimento de energia elétrica para alguma localidade [DEBENEST et al., 2008].



Figura 2 – Realização de manutenção por operador especializado, [DEBENEST et al., 2008].

Considerando os fatores mencionados, juntamente com o aumento da demanda por energia elétrica, os proprietários das redes de transmissão tem sido pressionados para melhorar as condições das linhas, aumentando assim sua confiabilidade, o que implica num aumento da necessidade de manutenção das mesmas. Segundo Toussaint, Pouliot e Montambault [2009], as dificuldades encontradas na realização da inspeção das linhas de transmissão tem incentivado o desenvolvimento de robôs para a realização das tarefas de inspeção e manutenção. Essas alternativas são extremamente importantes tanto para a manutenção das linhas de transmissão, quando para o aumento de segurança do consumidor final, do sistema elétrico em si e das equipes de manutenção, sendo incentivada, inclusive,



Figura 3 – Manutenção realizada a partir de aeronave, [SENAI - CIMATEC, 2013].

como linha de pesquisa de novos projetos da Agência Nacional de Energia Elétrica (Aneel). Essa linha de pesquisa incentiva a busca por soluções que envolvam a utilização de equipamentos robóticos para inspeção e manutenção de linhas de transmissão, de turbinas geradoras e outros componentes da infra-estrutura do sistema elétrico [ANEEL, 2017].

De acordo com Debenest et al. [2008] e Toussaint, Pouliot e Montambault [2009], a utilização de robôs é uma alternativa para reduzir os custos e perigos da inspeção nas linhas de transmissão. Desde 1998 existem estudos e projetos no intuito de desenvolver robôs de inspeção de linha de transmissão. Iniciando com abordagens simples como em [VENTRELLA et al., 2003], os robôs foram evoluindo e sendo capazes de ultrapassar os mais diversos obstáculos como apresentado em [ZHOU et al., 2005], composto por unidades de movimentação, braços flexíveis para permitir a ultrapassagem de obstáculos, garras para se fixar aos cabos e um sistema de controle.

Outro local de desenvolvimento é o instituto Hidro Québec Research Institute, que desde 1998 tem intensificado o desenvolvimento de robôs aplicados à manutenção em linhas de transmissão. Seus protótipos são capazes de ultrapassar obstáculos como: espaçadores, amortecedores e grampos de suspensão [TOUSSAINT; POULIOT; MONTAMBAULT, 2009]. No Japão, o Kansai Electric Power Corp. (KEPCO) e Japan Systems Corp. (JPS) também desenvolveram um robô de inspeção em linhas de transmissão chamado *Expliner* (Figura 4.b), capaz de se mover sobre os cabos de alta tensão, ultrapassar espaçadores de maneira segura e rápida e grampos de suspensão sem tocar na estrutura da torre [DEBENEST et al., 2008].

Apesar da evolução dos robôs de inspeção, a maioria é teleoperada durante toda sua atividade ou ao menos em parte dela. Um exemplo, que apresenta uma abordagem autônoma é o Ti, desenvolvido pelo *Electric Power Research Institute* (EPRI). Ele apresenta uma estratégia de ultrapassagem de obstáculos autônoma, contudo, é necessário a realização de intervenções na linha para adição de *by-pass* nos pontos em que existam obstáculos [PHILLIPS et al., 2012].

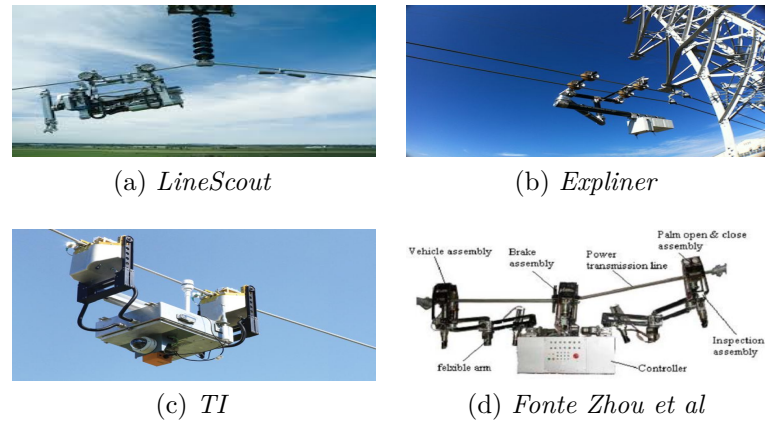


Figura 4 – Robôs de Inspeção de Linha de Transmissão.

Dentro desse contexto, este trabalho se apresenta como parte do projeto do robô *Electrical Inspection Robot* (ELIR), desenvolvido pelo BIR (*Brazilian Institute of Robotics*), no Senai CIMATEC. O projeto propõe-se a desenvolver um robô autônomo para inspeção visual e térmica de linhas de transmissão formada por cabos do tipo LINNET, com 18,3 mm de diâmetro e com condição de operação em 138kV. O robô deve ter a capacidade de transpor obstáculos, sendo considerado obstáculo todo objeto que interfira na trajetória do robô, monitorar a faixa de servidão e de transportar equipamentos e sensores para realização de inspeção. Sua estrutura deve ser de fácil montagem e fabricação, possuir massa reduzida e potência motriz suficiente para viabilizar a movimentação com inclinações de até 60°.

1.2 Objetivo Geral

O objetivo desse trabalho é projetar e implementar, em ambiente de teste, um classificador dos obstáculos encontrados durante a navegação autônoma do robô de inspeção de linha de transmissão, ELIR. Os obstáculos que serão analisados nesse trabalho serão o amortecedor de vibração e grampo de suspensão, ver Figura 5. Estes objetos são comumente encontrados nas linhas de transmissão e é necessário que sejam ultrapassados pelo robô para possibilitar a navegação contínua pelas linhas, por isso tem sido analisada sua detecção e classificação na literatura por diversos trabalhos como: [PETERS; AHN; BORKOWSKI, 2002], [ZUO et al., 2009], [WU et al., 2010], [POULIOT; RICHARD; MONTAMBAULT, 2012] e [RICHARD; POULIOT; MONTAMBAULT, 2014]. Como classificador será utilizado o algoritmo de classificação supervisionada *k*-Nearest Neighbors (*kNN*) juntamente como as *features* de forma: área e perímetro e a média da intensidade da imagem 2D em escala de cinza dos obstáculos citados, obtidas através de sensor do tipo *Light Detection and Ranging* (LiDAR).



Figura 5 – Obstáculos: amortecedor de vibração (esquerda) e grampo de suspensão (direita).

1.3 Objetivos Específicos

- Desenvolvimento de uma plataforma de teste para levantamento de dados baseada em parte do robô ELIR;
- Estudo de técnicas utilizadas para detectar e classificar obstáculos em linhas de transmissão;
- Geração de *dataset* de imagens dos obstáculos considerando distúrbios de trepidação, oclusão, diversidade de posição;
- Estudo comparativo entre as *features* de forma e intensidade para aplicação no algoritmo de classificação;
- Avaliação de desempenho do classificador *k-Nearest Neighbors* para classificação dos obstáculos utilizando *dataset* gerado e possível expansão para outros tipos de obstáculos;
- Avaliação de desempenho do classificador *k-Nearest Neighbors* para classificação dos obstáculos utilizando *dataset* gerado com adição de ruído nas imagens simulando problemas em aplicação em ambiente real;

1.4 Organização do Trabalho

Esta dissertação é composta por esse capítulo que apresenta uma breve introdução sobre o processo e a necessidade de inspeção de linhas de transmissão de energia elétrica, além das motivações e aspectos da introdução da utilização de robôs como alternativa para a realização dessa perigosa tarefa.

O Capítulo 2 apresenta uma revisão bibliográfica sobre os robôs de inspeção de linha de transmissão e de técnicas de detecção e classificação dos obstáculos encontrados na linha. Apresenta também as características do projeto ELIR e as justificativas para a escolha do sensor do tipo LiDAR, adotado na realização desse trabalho.

No Capítulo 3 apresenta-se os aspectos envolvidos na classificação de objetos em imagens, as *features* utilizadas nesse trabalho e o funcionamento e características da técnica de classificação *kNN*.

O Capítulo 4 explica o processo de desenvolvimento da estrutura para a realização dos testes juntamente com a metodologia seguida para o levantamento dos *datasets*, tratamento das imagens e classificação dos objetos nas imagens.

O Capítulo 5 apresenta os estudos do comportamento das *features* e os resultados de classificação juntamente com suas análises.

O Capítulo 6 mostra as conclusões advindas desse trabalhos e sugestões de pontos a serem estudados em trabalhos futuros.

2 Robôs de Inspeção de Linha de Transmissão

Como mencionado em Debenest et al. [2008] e Toussaint, Pouliot e Montambault [2009] os robôs tem ganhado destaque como solução para inspeção e manutenção das linhas de transmissão. Na literatura, podem ser encontrados diversos conceitos, protótipos e produtos a fim de oferecer soluções para essa tarefa. Nesta seção serão apresentados algumas das abordagens presentes na literatura.

Em Ventrella et al. [2003] é apresentado o projeto Sistema de Serviço de Linhas de Transmissão (SSL), esse projeto tem como objeto o desenvolvimento de um robô móvel para realizar inspeção e manutenção das linhas aéreas, tanto do cabo guarda quanto dos cabos condutores. Esse robô deve ser capaz de se locomover pela cabo guarda com controle de avanço, retorno e parada através do seu sistema de controle. Através de atividade tele operada, o sistema ainda deve ser capaz de realizar diferentes tipos de operações, como: inspecionar a integridade física e mecânica do cabo guarda quanto aos defeitos provenientes de descargas atmosféricas, possibilitando a gravação dessa inspeção; inspecionar visualmente os cabos condutores, espaçadores, isoladores de vidro e porcelana, vegetação sobre a linha e problemas como assoreamento na base das torres; instalação ou retirada de esferas de sinalização aérea do cabo guarda, mesmo com a linha em operação. Durante o deslocamento, também deve ser capaz de, com auxílio do piloto, realizar a transposição de grampos de suspensão. A Figura 6 mostra o protótipo do robô de inspeção de linha de transmissão.



Figura 6 – Protótipo de robô apresentado em [VENTRELLA et al., 2003].

Outro robô que se movimenta pelo cabo de guarda é apresentado em Campos et al. [2002]. O robô foi desenvolvido para a instalação e remoção de esferas sinalizadoras aéreas das linhas de transmissão. Com o computador a bordo, o sistema obtém os dados

dos sensores e envia para o operador em solo através de ondas de rádio, tendo por isso, um alcance de operação de 2 km. O equipamento foi testado em campo em condições reais, mostrando-se eficiente e robusto (SARTORI, 2018 apud CAMPOS et al., 2002). Imagem desse teste podem ser vista da Figura 7.



Figura 7 – Teste realizado em campo para remoção e instalação de esferas de sinalização, [CAMPOS et al., 2002].

Em Zhou et al. [2005] é apresentado um robô capaz de ultrapassar qualquer tipo de obstáculo, inclusive torres, trafegando pela linha de transmissão com tensão de alimentação de 110 kV. A Figura 8 mostra o robô com seus sistemas. Segundo Sartori [2018], o robô é composto por 5 partes principais: mecanismos de movimentação (*vehicle driving mechanism*), sistema de parada (*brake system assembly*), braços flexíveis (*flexible arms*), garras (*palm gripper*), fonte de alimentação (*power supply*) e sistema de controle (*controller*). A flexibilidade para ultrapassagem de obstáculos é obtida com os 16 eixos de movimentação presentes nesse robô. Tendo sido financiado pelo Plano Nacional Chinês 836, o robô também possui sistemas de câmera para realizar inspeção visual, enviando os dados para uma estação em solo através de sistema de comunicação sem fio.

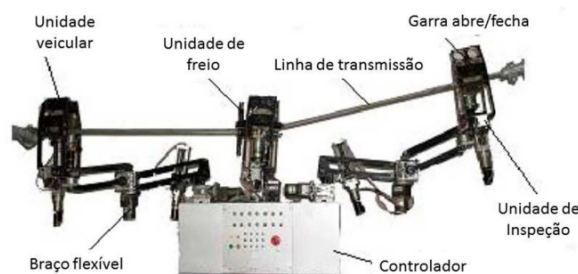


Figura 8 – Robô apresentado em [ZHOU et al., 2005], [SARTORI, 2018].

No Canadá, o *Hydro-Quebec Research Institute*, vem desenvolvendo pesquisas na área de aplicação robótica para inspeção e manutenção de linhas de transmissão. Em Montambault e Pouliot [2007] é apresentado o LineScout, o robô desenvolvido para realizar inspeção e manutenção em linhas de transmissão de 375 kV. Pesando 100Kg, esse robô foi validado em campo, com diferentes configurações de linha e ultrapassando

diferentes obstáculos, como: espaçadores, amortecedores de vibração e grampo de suspensão [TOUSSAINT; POULIOT; MONTAMBAULT, 2009]. Apoiado com rodas sobre a linha, o LineScout é capaz de se locomover de maneira rápida e eficiente, sendo capaz de passar por cima de alguns obstáculos. Para outras condições em que é necessário a ultrapassagem, com o auxílio do operador, o robô executa o procedimento de ultrapassagem mostrado na Figura 9.

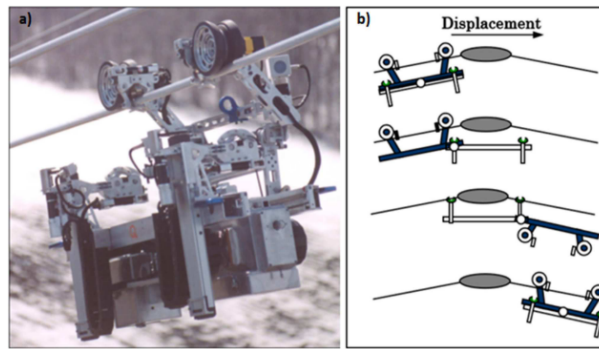


Figura 9 – **a)** LineScout, **b)** Procedimento de ultrapassagem de obstáculo, [MONTAMBAULT; POULIOT, 2007].

Em Katrasnik, Pernus e Likar [2008], apresentou-se um conceito novo de robô híbrido, uma combinação de robô móvel e Veículo Aéreo Não Tripulado (VANT). Segundo Katrasnik, Pernus e Likar [2008], após a investigação e comparação entre abordagens envolvendo veículos aéreos e robôs móveis, chegou-se a conclusão de que uma abordagem mista traria mais benefícios como um menor consumo de energia quando comparado a inspeção realizada apenas por VANT e uma maior flexibilidade para ultrapassagem de obstáculos, quando comparado aos robôs móveis que se deslocam pelas linhas. A Figura 10 mostra o diagrama conceitual desse robô híbrido

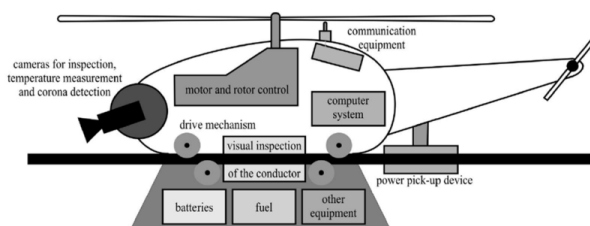


Figura 10 – Robô de inspeção híbrido, [KATRASNIK; PERNUS; LIKAR, 2008].

No Japão, o KEPCO e JPS também desenvolveram um robô de inspeção chamado Expliner, cujo sistemas e juntas são ilustrados na Figura 11. Operando em linhas de transmissão com configuração de 2 condutores por fase, este robô tele operado se desloca através de rodas que ficam apoiadas sobre os condutores. Contando com 2 pontos de apoio e um contra-peso é capaz de ultrapassar espaçadores e grampos de suspensão de maneira rápida e segura [DEBENEST et al., 2008]. O processo de ultrapassagem é realizado utilizando o contra-peso para levantar o ponto de apoio (P1) e se equilibrar nessa condição,

em seguida, o robô desloca-se em direção ao obstáculo até que o ponto P1 possa ser colocado sobre o cabo novamente. Para completar a ultrapassagem, o mesmo procedimento é repetido para o ponto de apoio P2, como ilustrado na Figura 12. Através de um sistema de comunicação sem fio, o operador tem sempre controle das tarefas, contudo, não precisa operar todas elas, alguns procedimentos encontram-se pré-programados na memória do robô, sendo necessário apenas a autorização do operador.

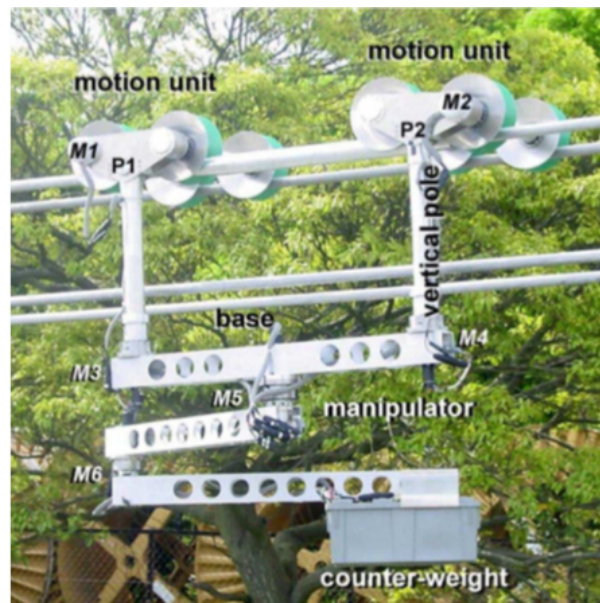


Figura 11 – Robô de inspeção híbrido, [KATRASNIK; PERNUS; LIKAR, 2008].

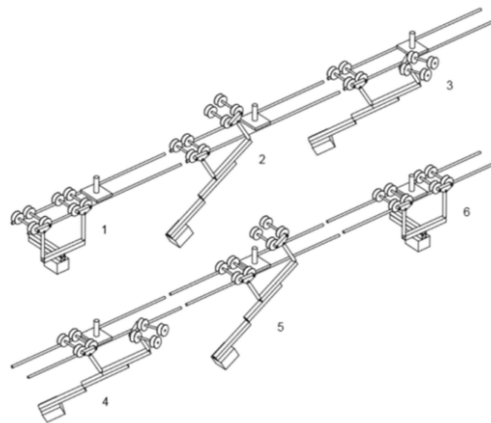


Figura 12 – Processo de ultrapassagem do robô Expliner, [KATRASNIK; PERNUS; LIKAR, 2008].

Rangel, Kienitz e Brandão [2009] utiliza um VANT para realizar inspeção das linhas de transmissão. Segundo Sartori (2018 apud RANGEL; KIENITZ; BRANDÃO, 2009) a integridade da linha é verificada utilizando-se câmeras e sensores instalados no equipamento que, através do sistema de controle, enviam os dados georreferenciados para a estação em solo, sendo assim processado e armazenado.

A utilização de VANTs para inspeção apresenta algumas desvantagens, como o fato de não substituir completamente a utilização de aeronaves tripuladas. Isso ocorre pois, devido a interferência eletromagnética gerada pelas linhas de transmissão, os VANTs não podem se aproximar muito das linhas. Outro ponto apontado nesse trabalho é de que, esse tipo de abordagem permite apenas a realização de inspeção, não sendo possível atividades de manutenção corretiva [SARTORI, 2018]. A Figura 13 mostra a aeronave de inspeção junto com a estação de controle em solo.



Figura 13 – VANT de inspeção juntamente com estação de controle, [RANGEL; KIENITZ; BRANDÃO, 2009].

Em Li e Ruan [2010] é apresentado um robô móvel que se desloca pelo cabo guarda. Segundo Sartori (2018 apud LI; RUAN, 2010), este é capaz de ultrapassar obstáculos como contrapeso, torre de ancoragem e torre de torção. O robô conta com 13 motores e seu movimento na linha é inspirado no movimento dos macacos. Ainda em fase de desenvolvimento, o robô apresentado na Figura 14, necessita de mais testes para aplicação em inspeção real.



Figura 14 – Teste de robô de inspeção no cabo guarda, [LI; RUAN, 2010].

Uma proposta diferenciada é apresentada em Wang et al. [2010], ao invés de se deslocar pela linha com rodas, esse robô apresenta uma estrutura bípede para "caminhar" sobre a linha. Os pés ficam presos na linha e sua movimentação ocorre através de um ciclo alternado entre momentos onde há apenas um apoio no cabo e momentos onde o robô encontra-se com dois apoios. Cada "perna" possui uma junta prismática que permite o ajuste do seu tamanho, fazendo assim com que o eixo da articulação central (*hip joint*) coincida com o centroide, o que reduz o consumo de energia e melhora a estabilidade do robô nos momentos que há apenas um ponto de apoio. Em trabalhos futuros, pretende-se avaliar a inclusão de sensores para detecção de linha e controlador *on-board*, além de realizar testes em ambiente real e com presença de obstáculos (SARTORI, 2018 apud WANG et al., 2010). A Figura 15 mostra um modelo 3D do robô juntamente com suas partes.

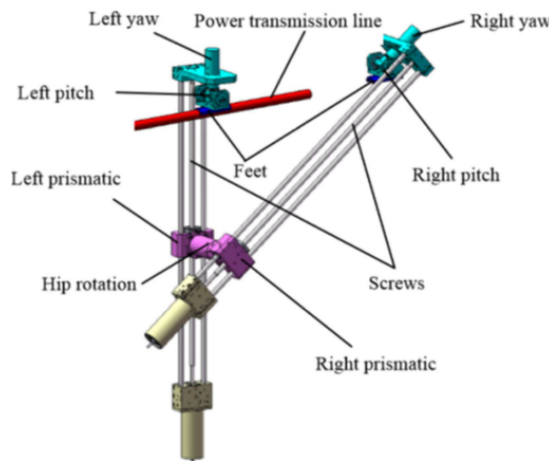


Figura 15 – Robô de inspeção bípede, [WANG et al., 2010].

Outro robô que se locomove pelo cabo guarda é apresentado em Gonçalves e Carvalho [2010]. De acordo com (SARTORI, 2018 apud GONCALVES; CARVALHO, 2010), o robô apresentado é capaz de transpor obstáculos como grampos e esferas de sinalização. Dotado de quatro pernas que podem variar o comprimento, apresenta um sistema de fácil controle, uma vez que as quatro pernas são movimentadas por apenas um motor, associado a um conjunto de polias e correias. A Figura 16 apresenta o modelo 3D dessa abordagem.

Com o intuito de inspecionar luva de emenda de cabos, o Instituto Coreano de Pesquisa em Energia Elétrica, através de Lee, Jung e Cho [2011], propõe um robô com movimento semelhante a uma lagarta. Através de operação remota, o robô fornece informações da sua condição para o operador juntamente com os dados fornecidos pelo sensor de campo magnético utilizado para verificar as condições da luva de emenda e sua concentricidade.

Ilustrado na Figura 17, o robô apresenta uma parte responsável pela realização

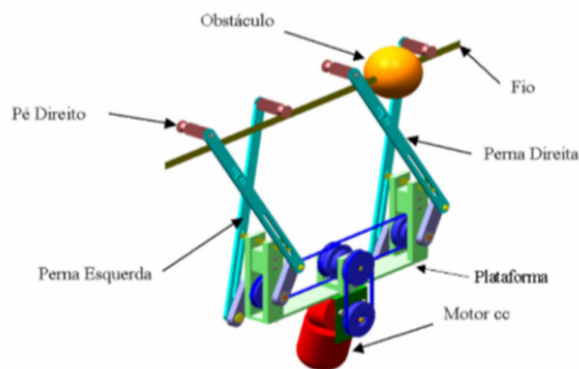


Figura 16 – Modelo 3D de robô de inspeção de linha de transmissão, [GONCALVES; CARVALHO, 2010].



Figura 17 – Robô de inspeção de luva de emenda, [LEE; JUNG; CHO, 2011].

da inspeção e outra com o objetivo de realizar a movimentação. Segundo Gonçalves e Carvalho [2013], o robô desliza pela linha onde fica aderido durante sua locomoção. Essa aderência é possibilitada por uma série de sulcos e ranhuras presentes no seu sistema de movimentação.

O *Electrical Power Research Institute* (EPRI) apresenta em Phillips et al. [2012] o robô Ti, Figura 18. Com atividade autônoma, esse robô necessita da instalação de um *by-pass* nas torres para que seja possível a ultrapassagem de obstáculos. Com o objetivo de oferecer informações a fim de otimizar a manutenção das linhas de transmissão e melhorar a confiabilidade das mesmas. Esse robô possui câmera de alta definição, câmeras de espectro infravermelho, detectores de interferência eletromagnética, além de outros sensores de radiofrequência.

Em Gonçalves e Carvalho [2013] é apresentado um conceito de robô híbrido e modular. Baseado na proposta apresentada por Katrasnik, Pernus e Likar [2008], é proposto um conceito de solução com diversos módulos, sendo cada um responsável por uma atividade em específico. Cada módulo deveria possuir seu próprio sistema de locomoção e fonte de energia. O conceito possui um módulo principal equipado com um *Unmanned Aerial Vehicle* (UAV) responsável por transportar todos os outros módulos na

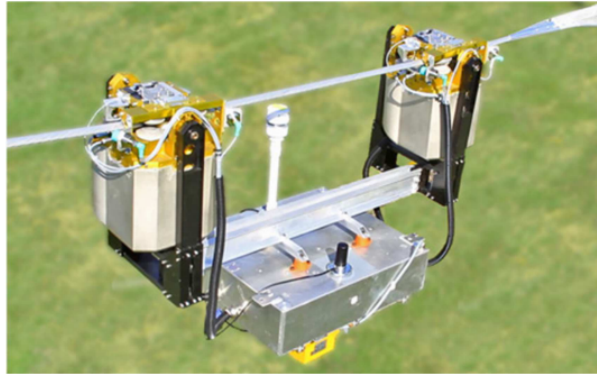


Figura 18 – Ti se deslocando pela linha de transmissão, [PHILLIPS et al., 2012].

presença de obstáculos, de maneira tele operada. Além desse módulo principal, existiriam módulos de: inspeção, bateria, manutenção, dentre outros. A Figura 19 mostra o conceito juntamente com alguns módulos.

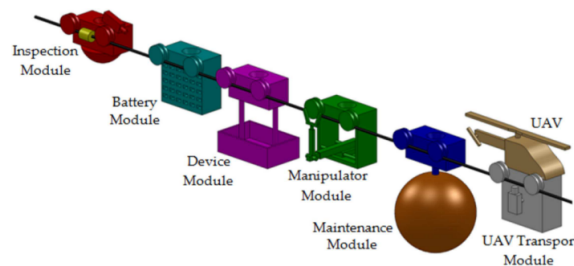


Figura 19 – Apresentação conceitual de módulos de sistema de inspeção de linha de transmissão, [GONÇALVES; CARVALHO, 2013].

Segundo Gonçalves e Carvalho [2013], a vantagem da modularidade é a otimização do peso, uma vez que só seriam utilizados os módulos necessários para uma tarefa em específico. Isso aumenta a flexibilidade do robô quanto a tarefas, como por exemplo: inspeção, limpeza, manutenção, captura de imagens, dentre outras, sem a necessidade de aumento de peso de maneira proporcional, quando comparado a uma solução onde todas essas tarefas fossem realizadas pelo mesmo robô.

Considerando esse contexto, o SENAI-CIMATEC/BIR, em parceria com a Companhia Energética de Minas Gerais (CEMIG) e a Universidade Federal de Minas Gerais, desenvolveu um robô para inspeção de linhas de transmissão. Financiado pelo Programa de Pesquisa e Desenvolvimento Tecnológico do Setor de Energia Elétrica, regulado pela Agência Nacional de Energia Elétrica (ANEEL), o projeto denominado como D311, Figura 20, deveria atingir os seguintes requisitos [II et al., 2014]:

1. O robô deve ser capaz de trabalhar na faixa de tensão entre 124,4 kV e 151,8 kV com corrente trifásica de 500A.

2. O robô deverá ser autônomo, dependendo de operador apenas para instalação e remoção do mesmo da linha ou para eventuais operações ocorridas devido a emergência.
3. O robô deverá operar em cabo condutor do tipo LINNET com diâmetro de 18,3mm.
4. A massa do robô deverá ser igual ou inferior a 14 Kg.
5. O robô deve possuir blindagem eletromagnética para assegurar seu funcionamento em ambiente com condições de interferência eletromagnética como as linhas de transmissão.
6. O robô deverá realizar inspeção visual e térmica, detectando possíveis falhas que venham a causar danos ao fornecimento de energia.

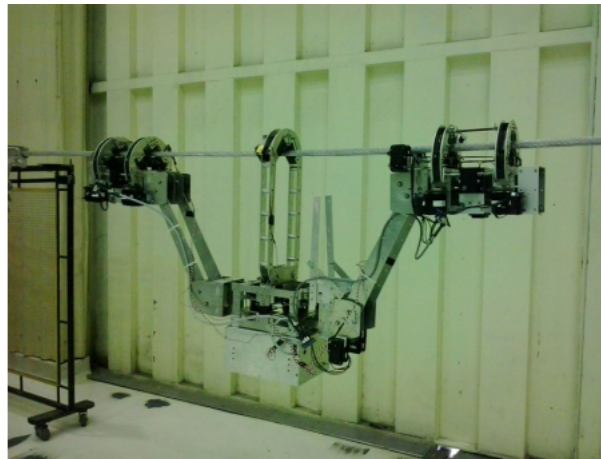


Figura 20 – Robô D311 realizando testes, [II et al., 2014].

Para realizar a operação autônoma, o robô deve ser capaz de ultrapassar obstáculos como grampos de suspensão, amortecedores de vibração e luvas de emenda. Objetivando a realização de tais tarefas, o conceito mecânico foi dividido em três módulos: módulo de busca, módulo de tração e módulo de apoio, ilustrado na Figura 21. O módulo de busca tem a função de conter os sensores utilizados na tarefa de inspeção e na identificação de obstáculos e do cabo no processo de ultrapassagem. O módulo de tração possui os motores responsáveis pelo movimento do robô ao longo da linha e o módulo de apoio atua como referência para o robô no processo de ultrapassagem de obstáculos. O sistema mecânico foi inspirado numa lagarta Caterpillar para realização da ultrapassagem de obstáculos [II et al., 2014].

Uma evolução do D311 foi desenvolvida e apresentada em [MOURÃO et al., 2015], agora com o nome de *Powerlines Inspection Robot* (PIRo). Além da busca pelos requisitos apresentados em [II et al., 2014], nesse robô buscou-se a melhoria de aspectos como redução de peso, componentes com blindagem eletromagnética para proteção dos componentes

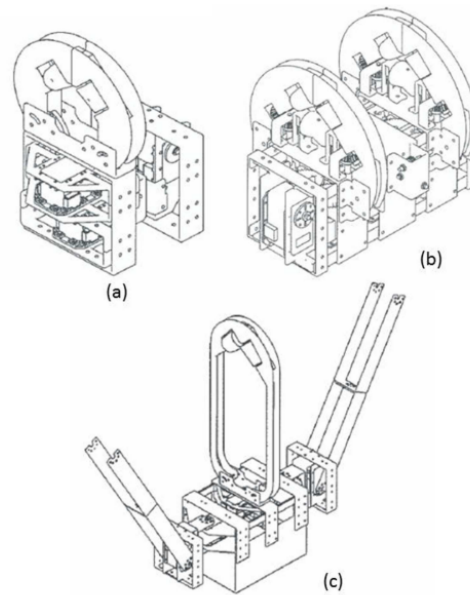


Figura 21 – Módulos do Robô D311, **a)** Módulo de Busca, **b)** Módulo de Tração, **c)** Módulo de Apoio, [MOURÃO et al., 2015].

eletrônicos além de um sistema de comunicação sem fio. A Figura 22 mostra o PIRO juntamente com a denominação de suas partes. O robô apresenta uma estrutura simétrica composta por: dois braços, duas unidades de tração e uma unidade de apoio. A unidade de apoio funciona como referência no processo de ultrapassagem de obstáculos e suporte para a caixa onde ficam protegidos os equipamentos eletrônicos, os dois braços tem a função de unir a unidade de apoio com as unidades de tração e possibilitar os graus de liberdade necessário para a ultrapassagem de maneira autônoma. Segundo [MOURÃO et al., 2015] este robô alcançou uma inovação ímpar dentro do estado-da-arte devido ao sua massa total ser de aproximadamente 9 Kg, ver Figura 23.

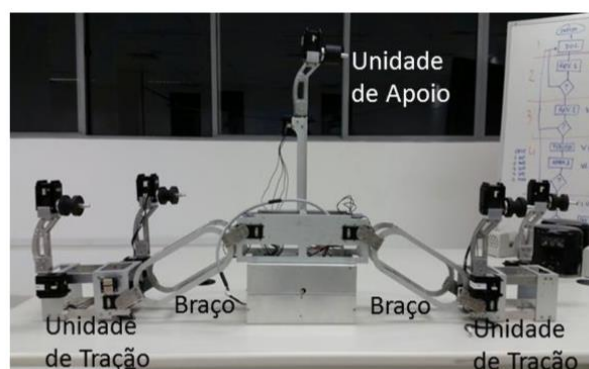


Figura 22 – PIRO, [MOURÃO et al., 2015].

Em [CHANG et al., 2017] é apresentado um robô híbrido. Essa denominação é usada pelos autores devido ao robô possuir capacidade de se deslocar pelo cabo para-raio e voar sobre os obstáculos para ultrapassá-los. A configuração híbrida se baseia na ideia de unir os benefícios dos robôs que se deslocam pelas cabos para realizar inspeção juntamente

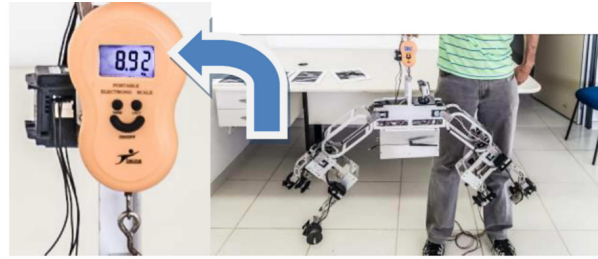


Figura 23 – Apresentação da massa do PIro de 8,92 kg, [MOURÃO et al., 2015].

com os robôs aéreos. O aspecto de se deslocar pela linha trás a vantagem da redução de consumo de energia, melhorando na autonomia do robô quanto ao tempo de inspeção, além de possibilitar uma inspeção mais precisa e com melhor qualidade. Já a característica de voo sobre os obstáculos faz com que a ultrapassagem não seja limitada pela geometria do robô, sendo possível ultrapassar todos os tipos de obstáculos encontrados na linha.

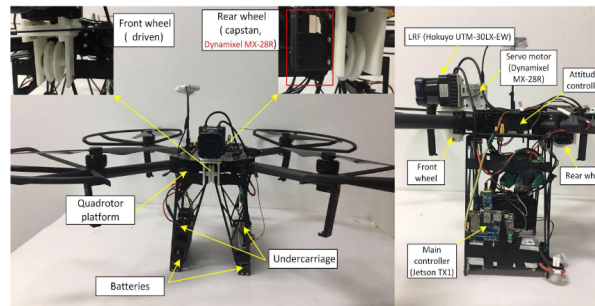


Figura 24 – Robô Híbrido, [CHANG et al., 2017].

A Figura 24 mostra o robô, sua estrutura mecânica e sensores. Pesando um total de 4,235 Kg, o robô possui estrutura de aterrissagem em forma de cone, o que possibilita guiar o cabo para o centro, onde estão as roldanas utilizadas para o apoio e deslocamento na linha. Essa configuração auxilia o ajuste de posição, de maneira passiva, na aterrissagem no cabo, reduzindo o controle e precisão necessária para execução dessa tarefa. Para detectar obstáculos e realizar a ultrapassagem de maneira autônoma, utiliza-se um Lidar Hokuyo UTM-30LX-EW juntamente com um motor Dynamixel MX-28R para fazer o escaneamento 3D do obstáculo e para encontrar a linha no processo de ultrapassagem. Segundo os autores, o robô apresentou resultados bons e robustos na detecção, ultrapassagem e algoritmo de pouso em uma linha montada em ambiente externo, mesmo na presença de agentes como vento e luz solar.

2.1 EIIR (*Electrical Inspection Robot*)

O projeto ELIR, sendo uma evolução do projeto PIro, tem o objetivo de desenvolver um robô capaz de realizar inspeção visual e térmica de linhas de alta tensão (138kV) sem a necessidade da interrupção do funcionamento da linha. Pretende-se que o robô seja autônomo, necessitando de utilização de operador apenas para instalação e remoção do

cabo ou para atuação em eventuais condições de emergência e que possua desempenho médio de 14,4 km por dia, considerando oito horas de atividade.

Durante realização da inspeção o robô deve filmar a linha de transmissão e os equipamentos instalados na mesma. Para realizar a inspeção de maneira contínua, o robô também deve ser capaz de detectar, identificar e transpor os obstáculos que encontrem-se no seu caminho como grampos de suspensão e amortecedor de vibração. Além da inspeção das linhas, deve-se inspecionar a faixa de servidão, fornecendo dados que possam indicar violação na sua condição de segurança. Para executar essas tarefas, o robô deve contar com câmera de espectro visível e infravermelho além de uma série de sensores que possibilitem adquirir informações do seu funcionamento e do meio.

Juntamente com a capacidade de ultrapassagem de obstáculos, o desenvolvimento mecânico objetiva possuir massa reduzida, aproximadamente 8,5 kg, além de capacidade para vencer inclinações dos cabos com angulações de até 60°. Além desses pontos, buscou-se a facilidade na montagem e fabricação, através de número reduzido e simplicidade das peças que compõem o robô.

Dessa forma, são pré-requisitos do robô:

1. Realizar inspeção visual e térmica de linha de transmissão.
2. Monitorar a faixa de servidão.
3. Detectar e ultrapassar, de maneira autônoma, obstáculos como grampo de suspensão e amortecedor de vibração.
4. Trafegar em linha energizada com tensão de até 138kV, formada por condutor do tipo LINNET com 18,3mm de diâmetro.
5. Necessitar de operador apenas para instalação e remoção de linha.
6. Possuir massa inferior ou igual a 14 kg, possibilitando a instalação em campo através da utilização de hastes por apenas 2 operadores.
7. Possuir sensores e equipamentos necessário para realização das inspeções.
8. Ter blindagem eletromagnética para proteção de componentes eletrônicos quanto ao efeito da interferência eletromagnética provocada pelas linhas de transmissão.

A Figura 25 mostra o conceito mecânico do robô ELIR juntamente com suas partes e denominações.

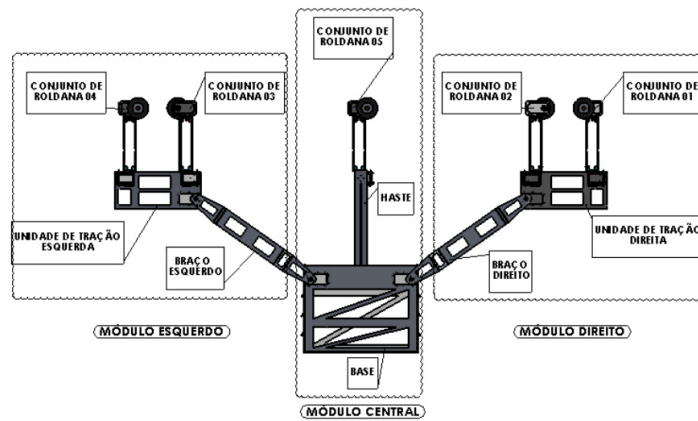


Figura 25 – Modelo 3D de estrutura mecânica do robô ELIR, [SARTORI, 2018].

2.2 Técnicas de Identificação de Obstáculos em Linhas de Transmissão

Segundo Phillips et al. [2012], os robôs de inspeção de linha de transmissão estão caminhando para uma atuação cada vez mais autônoma. Um dos grandes problemas para obtenção da autonomia é a tarefa de ultrapassagem de obstáculos. Já existem robôs capazes de ultrapassar obstáculos, como por exemplo os apresentados em [DEBENEST et al., 2008] e [TOUSSAINT; POULIOT; MONTAMBAULT, 2009]. Estes possuem os mecanismos e configurações necessárias para a transposição dos obstáculos, mas não conseguem identificá-los como tal, precisando de auxílio humano nessa tarefa. Nesse contexto, diversos estudos tem sido feitos na literatura a fim de inserir a capacidade de identificar obstáculos aos robôs de inspeção de linha.

As linhas de transmissão são consideradas ambientes semi-estruturados, uma vez que a própria configuração da linha limita a região onde os obstáculos podem ser encontrados [POULIOT; RICHARD; MONTAMBAULT, 2012]. Apesar desta simplificação, os obstáculos ainda precisam ser detectados e identificados para que possa ocorrer o processo de ultrapassagem. Várias técnicas e sensores tem sido utilizados para o reconhecimento dos obstáculos.

Em Peters, Ahn e Borkowski [2002] utilizou diversos sensores ultrassônicos de proximidade para detectar os obstáculos. Os dados fornecidos pelos sensores foram analisados por um sistema neural de classificação utilizando medição de distância como dados de entrada. Os sensores de distância foram posicionados de forma que captassem objetos em 3 regiões: acima do cabo, no cabo e abaixo do cabo. Essas medidas foram utilizadas como dados de entrada para alimentar uma rede neural do tipo *rough neural network* (RNN). Foram utilizados cerca de 7000 medições para treinamento e teste do classificador que, após 501 épocas apresentou acurácia de 51,2%, melhorando o desempenho para 95,6% após 1841 épocas.

Outra abordagem também com sensores ultrassônicos foi utilizada em [HAN; LEE, 2009]. Nesse trabalho é apresentado uma técnica de detecção dos postes que são detectados utilizando sensores ultrassônicos de distância. Ao trafegar pelo cabo para-raio e após a detecção do poste, é realizada a transposição do mesmo. No processo de reacoplamento das garras, são fundidas as informações de sensores ultrassônicos e infravermelho para melhorar a medida de distância e aprimorar a tarefa de acoplamento da garra.

Câmeras foram utilizadas em [FU et al., 2006] e [ZUO et al., 2009] para o desenvolvimento de um robô de inspeção de linha da Academia Chinesa de Ciência. Através do processamento de imagem foram extraídas *features* de detecção 2D como linhas retas, círculos e elipses, para a detecção de grampos de ancoragem, grampos de suspensão, cadeia de isoladores e amortecedores de vibração. Após a detecção de linhas retas, o cabo é identificado através da largura entre as linhas obtidas por valores experimentais. Busca-se, então, por *features* circulares para a detecção dos obstáculos. A classificação é realizada pela combinação entre as *features*. Após a detecção do cabo, sendo encontrado também uma elipse com o cabo relativamente centralizado, caracteriza-se uma cadeia de isoladores e, conseqüentemente abaixo, um grampo de suspensão, mostrando uma detecção indireta do obstáculo uma vez que o objeto detectado é o isolador enquanto que o obstáculo é o grampo de suspensão. Para a classificação do grampo de ancoragem busca-se pela detecção de cabo entre dois círculos. No caso da detecção do cabo com dois círculos, um acima e outro abaixo, representa-se a classificação de amortecedores. Os testes foram realizados em ambiente interno e externo, apresentado bons resultados segundo os autores, contudo, não se apresentou nenhuma métrica acerca do que seria um bom desempenho, bem como não houve a apresentação da quantidade de dados analisados.

Outro tipo de sensor foi utilizado em [WU et al., 2008] e [WU et al., 2010]. Utilizando sensores de campo magnético, foi possível realizar a detecção, classificação e auxílio no planejamento de trajetória para ultrapassagem de obstáculos. Através da medição do campo magnético e da comparação entre obstáculos que são magnetizáveis, grampo de suspensão e amortecedor de vibração e os que não são, grampo de ancoragem, juntamente com a posição dos obstáculos foi possível a classificação e auxílio na trajetória da transposição dos obstáculos. A detecção e classificação, apesar de ser realizada em ambiente real, foi baseada apenas nos valores de medida comparativa através de *threshold*, não apresentando para esse processo, nenhuma métrica de desempenho ou auxílio de classificador que possa tratar medições e valores fora do previsto.

Já em [POULIOT; RICHARD; MONTAMBAULT, 2012] e [RICHARD; POULIOT; MONTAMBAULT, 2014] utilizou-se um LiDAR para a detecção e descrição dos obstáculos na linha. Através dos dados coletados, foi possível levantar características como largura, e perfil de distância, utilizando um *threshold* dessas informações, conseguiu-se realizar a diferenciação dos componentes na linha entre grampo de suspensão, amortecedor de vibração e luva de emenda. Apesar de testes em ambiente interno e externo, não

apresentou-se o número de vezes que os obstáculos foram classificados utilizando-se a metodologia apresentada, bem como uma métrica para representar o desempenho da tarefa de classificação.

No PIRO [MOURÃO et al., 2015], inicialmente utilizou-se sensor ultrassônico para a detecção de obstáculos. Contudo durante a realização dos testes, percebeu-se que este era muito instável. Mesmo possuindo um "cone de detecção" não muito estreito, como o caso de sensores infravermelho, por diversas vezes, o obstáculos não foi detectado devido a não estar perfeitamente alinhado com a região de detecção do sensor.

Em [POULIOT; RICHARD; MONTAMBAULT, 2012] e [RICHARD; POULIOT; MONTAMBAULT, 2014] foi realizado um extenso estudo com a aplicação do LiDAR Hokuyo UTM-30LX para detecção e identificação dos componentes na linha a fim de tornar o robô LineScout autônomo. Os resultados apresentados foram bastante interessante e promissores.

2.2.1 Escolha do sensor

LiDAR é um sensor do tipo *laser range finder* e é amplamente utilizado em aplicações de robótica móvel para detecção de obstáculos e mapeamento de ambiente [RICHARD; POULIOT; MONTAMBAULT, 2014]. Nos últimos 10 anos, o LiDAR se tornou uma escolha preferencial na robótica móvel por apresentar funcionalidades como: detecção de longo alcance, processamento simples e boa velocidade de aquisição. Inicialmente, as aplicações concentravam-se em utilizações em ambiente interno, conduto, com o aumento da resolução e robustez, aplicações em ambiente externo ganharam espaço, tendo sido influenciado também pela redução de dimensões, peso e preço [POULIOT; RICHARD; MONTAMBAULT, 2012].

Estes tipos de sensores tem demonstrado sua qualidade em diversos ambientes: em ruas pavimentadas [KANG et al., 2012], em ambientes rurais [MANZ et al., 2011], dentro de florestas [MCDANIEL et al., 2010] e em ambiente cheio de pedestres [PREMEBIDA; LUDWIG; NUNES, 2009].

Na aplicação para robôs de inspeção de linha de transmissão, o LiDAR apresenta diversas vantagens. Uma vez que mede a distância ao anteparo, problemas dos objetos presentes no plano de fundo podem ser resolvidos de maneira simples. Esta é uma vantagem quando comparado com a utilização de câmeras, onde é necessário uma etapa de pré-processamento para isolar o obstáculo a ser identificado na imagem dos objetos presentes no plano de fundo. Considerando que o ambiente que passam as linhas de transmissão são bastante dinâmicos e variados, realizar essa separação é apresentada como uma difícil tarefa em [FU et al., 2006]. Ainda em [FU et al., 2006], a solução apresentada para a identificação dos obstáculos passa por uma detecção indireta de objetos. No caso dos grampos de suspensão, por exemplo, foi detectada os pratos de isolamento, não o grampo, que

é o obstáculo propriamente. Essa detecção indireta pode aumentar o erro em aplicações reais. Esse problema não é encontrado em aplicações com LiDAR, uma vez que os feixes refletem diretamente dos obstáculos.

Outra vantagem desse tipo de sensor é a aplicação para inspeção da faixa de servidão. A faixa de servidão é a área abaixo das linhas de transmissão que precisa ser inspecionadas uma vez que, por norma, não pode haver nenhuma construção ou objeto a uma determinada distância abaixo da linha. Com a utilização do LiDAR, é possível realizar a reconstrução 3D dessa região, aumentando as informações das concessionárias em relação as condições de suas instalações. Essa inspeção é um dos tipos de inspeção a serem realizados pelo ELIR, assim, a possibilidade da utilização do mesmo sensor para diversos fins, classificação dos obstáculos e inspeção da faixa de servidão, é um fator que influencia de maneira positiva a utilização desse sensor neste trabalho.

Segundo Richard, Pouliot e Montambault (2014 apud PASCOAL; MARQUES; ALMEIDA, 2008), sensores desse tipo, como o Hokuyo UTM-30LX, são compactos, leves, baratos e robustos suficientes para trabalhar sob luz do sol, poeira, chuva e outras condições de um ambiente externo. Em [POULIOT; RICHARD; MONTAMBAULT, 2012] foram realizados uma série de testes para caracterização deste sensor que apresentou-se com uma boa resposta em relação a orientação, aspecto interessante visto que as linhas formam uma catenária, variando o ângulo de incidência do feixe laser no cabo. Outra característica desse modelo de LiDAR é a boa precisão nas medições de distância. Mesmo para medições em cabos, que apresentam um formato cilíndrico e fino, os resultados foram satisfatórios. Neste ponto é importante destacar um desempenho diferenciado do Hokuyo quando o cabo apresenta-se em condições de oxidação, isso é interessante, uma vez que, devido as intempéries, será a condição da maioria dos cabos nas linhas de transmissão em aplicação real, fortalecendo mais ainda a escolha pela sua utilização.

Considerando condições reais de aplicação, passam pelas linhas uma elevada corrente elétrica, o que gera um campo magnético. Visto que os sensores possuem uma série de componentes eletrônicos, o efeito do campo magnético deve ser avaliado, para garantir que não haverá problemas na medição ou até a possibilidade de completa inutilização do sensor. Em [RICHARD; POULIOT; MONTAMBAULT, 2014] o Hokuyo foi submetido a teste sob um campo eletromagnético equivalente, não apresentando nenhuma alteração no seu funcionamento. Outro aspecto importante apresentado em [RICHARD; POULIOT; MONTAMBAULT, 2014] é que, apesar de a medida de distância ter influência da incidência direta da luz do sol, as medidas de largura e forma dos obstáculos interceptados pelo feixe não se alteravam, mostrando outro ponto positivo do LiDAR, quando comparado com câmeras, por exemplo, que podem sofrer saturação da imagem quando submetidos a luminosidade intensa. Em [POULIOT; RICHARD; MONTAMBAULT, 2015] esse sensor é apresentado como alternativa futura para aumentar a autonomia na ultrapassagem de obstáculos pelo robô Linescout, sendo utilizado para detectar e classificar obstáculos

presentar nos cabos condutores das linhas de transmissão.

Um sensor da mesma família foi utilizada em [CHANG et al., 2017] para detectar obstáculos da linha de transmissão e localizar o cabo utilizado como referência para o pouso do robô quando operando em modo aéreo. Os resultados da detecção de obstáculos e do cabo se mostraram robustos, mesmo em uma linha instalada em ambiente externo.

Dessa forma, com base nas informações e considerações citadas, foi escolhido como sensor para a tarefa de obtenção dos dados para classificação dos obstáculos, realizado nesse trabalho, o LiDAR, modelo UTM-30LX, do fabricante Hokuyo, ver Figura 26.



Figura 26 – LiDAR : Hokuyo UTM-30LX.

Segundo Hokuyo [2013], este LiDAR é um sensor 2D com ângulo de detecção ajustável de no máximo 270° com 1080 passos de medição, o que dá uma resolução angular de $0,25^\circ$ a cada 25 ms. Possui distância de medição entre 0,1 m e 30 m, disponibilizando informação de distância e intensidade do feixe refletido.

3 Fundamentação Teórica

3.1 Classificação de Objetos em Imagens

Classificação é uma tarefa que se relaciona com diversos aspectos da vida, além da sua fundamental relação com outros processos como seleção e tomada de decisão [SOLOMON; BRECKON, 2000]. Atualmente há um aumento da automação em diversas áreas, nesse contexto a classificação automática se insere de maneira importante visto que é uma atividade relacionada a diferentes processos de produção, manufatura e de diversos aspectos da vida cotidiana. Dentre a classificação automática de dados de várias naturezas pode-se destacar a classificação automática de imagens. Essas aplicações tem evoluído devido a evolução de computadores e câmeras tornarem cada vez mais possível a utilização desse sensor. A utilização da câmera como sensor para diversos fins, tem ganhado importância e destaque tendo em vista que, sendo a visão o sentido mais utilizado pelo ser humano, pode-se atribuir essa característica para robôs e processos automáticos.

As aplicações de classificação automática de imagens se encontram nas mais diversas áreas como: classificação de células como normais ou anormais, ou em busca de algum problema específico, seleção de frutas numa planta de empacotamento, classificação de obstáculos em aplicações robóticas, dentre outras [SOLOMON; BRECKON, 2000]. Segundo Solomon e Breckon [2000], o processo de desenvolvimento de um classificador automático é formado pelos seguintes passos:

1. **Definição de classes:** O primeiro passo para desenvolver um sistema de classificação é especificar o problema, assim, nessa etapa é necessário definir se o sistema irá apenas classificar entre duas classes, como o exemplo desse trabalho, definir se o obstáculo é um amortecedor de vibração ou grampo de suspensão, ou se ele irá classificar entre um maior número de classes.
2. **Exploração dos dados:** Nessa etapa se faz a análise das classes a fim de determinar características que as descrevam e possam ser utilizadas para diferenciar uma da outra. Atributos comumente utilizados são relacionados com intensidade dos pixels, forma, cor, textura ou combinações dessas características. Para realizar essa análise, não existe uma abordagem ou procedimento específico, geralmente é baseado na experiência e observação do projetista. Também depende muito da aplicação, por exemplo, numa aplicação em que deseja-se classificar entre limões e laranjas, a cor e textura são aspectos importantes, enquanto que a forma é bem parecida e não deve ajudar muito na diferenciação. Por outro lado, na aplicação desse trabalho por exemplo, onde deseja-se classificar entre os obstáculos presentes na linha de

transmissão, os principais pontos de destaque e diferenciação estão relacionados com a forma e a intensidade, a cor não se apresenta como uma característica promissora, tanto pelo fato da utilização do LiDAR não fornecer informação de cor, como, mesmo na utilização de câmeras, os obstáculos não possuem cores significativamente diferentes.

3. **Seleção e extração de *features*:** Segundo [RASCHKA; MIRJALILI, 2015], dados brutos raramente se encontram na forma e condição necessárias para utilização apropriada em algoritmos de classificação, devido a isso, a etapa de pré-processamento dos dados é uma das mais cruciais numa aplicação de classificação utilizando algoritmos de aprendizado de máquina. Dentro do contexto de classificação de imagens, as imagens são os dados brutos, é necessário então, após a exploração dos dados e observação de quais características são mais relevantes para a separação na classificação, extrair as *features* que irão definir o espaço de características formado pelo vetor de dados com valores referentes a cada objeto e que, posteriormente, servirá de informação para o processo de classificação. Esse processo de extração das *features* é obtido através de técnicas de processamento das imagens com a função de, dado uma matriz de pixels que formam uma imagem, extrair dados como: área, cor, descritores de textura e diversas outras métricas existentes no âmbito do processamento de imagens. Nesse trabalho, utilizou-se *features* de forma como: área e perímetro dos obstáculos, juntamente com medida referente a intensidade dos pixels, essas *features* serão melhor abordadas posteriormente.
4. **Construção do classificador com dados de treinamento:** Para a construção do classificador, geralmente se escolhe, dentre as diversas técnicas de inteligência artificial ou processamento estatístico qual a que mais se adapta a sua condição de aplicação. Com base nesse algoritmo, será fornecido a ele dados de treinamento, Esses dados são vetores das *features* extraídas anteriormente e que caracterizam cada objeto. Utilizando esses vetores de *features*, o algoritmo deverá ser capaz de encontrar um modelo que seja capaz de realizar a diferenciação entre as classes apresentadas.
5. **Teste do classificador:** O real objetivo de qualquer classificador é generalizar a capacidade de classificação a novos exemplo. Para analisar esse capacidade de generalização é necessário testar o classificador. Para isso, são fornecidos dados inteiramente novos e analisada, através de métricas de avaliação, o desempenho do classificador.

Os passos citados são, de maneira geral, os passos utilizados para o desenvolvimento de um sistema de classificação, contudo, esse não é um processo estático, geralmente ocorre de maneira iterativa. Caso, no momento de teste, o classificador não apresente desempenho satisfatório, sendo o conceito de satisfatório dependente da aplicação, ocorrem outras

iterações entre os passos a fim de se alcançar o objetivo desejado. A Figura 27 ilustra de maneira esquemática esse ciclo.

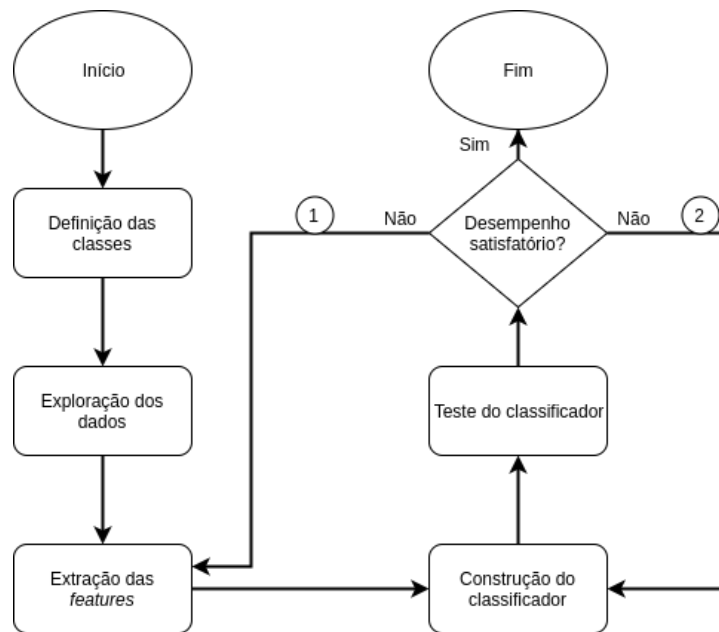


Figura 27 – Etapas do processo de classificação.

Quando não há um desempenho satisfatório, é necessário realizar a sintonia do classificador novamente, alterando os parâmetros de controle, demonstrado no ciclo "2". Contudo há situações em que é necessário ir mais fundo e rever as *features* utilizadas para melhorar o desempenho do classificador, processo ilustrado pelo ciclo "1".

3.2 *Features* de Forma

Como já mencionado, uma das etapas do processo de classificação é a exploração dos dados com o objetivo de identificar características das classes que possam descrevê-las e discriminá-las das demais. No contexto da inspeção das linhas de transmissão, são encontrados basicamente dois tipos de obstáculos, grampo de suspensão e amortecedor de vibração, Figura 5. Analisando os obstáculos a serem classificados, observou-se que suas formas geométricas (área e perímetro) e suas posições relativas ao cabo poderiam ser discriminantes. O amortecedor situa-se abaixo do cabo, enquanto o grampo de suspensão localiza-se no nível ou acima do cabo, essa característica também foi utilizada em [WU et al., 2010], para diferenciação desses obstáculos. Diante dessa observação, optou-se por utilizar como *features* para classificação a área, o perímetro e a média das intensidades da imagem (distância do objeto ao LiDAR convertida em intensidade dos pixels na imagem) apenas com os obstáculos.

3.2.1 Área

Os Momentos Invariantes são conceitos análogos ao momento dos corpos rígidos na Física e indicam uma distribuição de componentes na imagem [YANG; KPALMA; RONSIN, 2008]. Sua representação é demonstrada através da Equação 3.1.

$$M_{pq} = \sum_x^{nx} \sum_y^{ny} x^p y^q f(x, y) \quad (3.1)$$

Onde $p = 1, 2, 3, \dots$ e $q = 1, 2, 3, \dots$ são a ordem do momento, nx é a largura da imagem e ny é a altura da imagem.

O valor da área pode ser obtido do momento de ordem 0 (M_{00}), representado na Equação 3.2.

$$M_{00} = \sum_x^{nx} \sum_y^{ny} f(x, y) \quad (3.2)$$

Assim, a área é obtida através do somatório do valor dos pixels da imagem na vertical e horizontal. Sendo $f(x, y)$ a intensidade dos pixels da imagem. Em imagens binárias $f(x, y) = 1$ para os pixels pertencentes ao objeto selecionado e $f(x, y) = 0$ para todo o resto da imagem. A área é calculada como o somatório dos pixels não nulos da imagem, ou seja, os pixels pertencentes ao objeto selecionado.

3.2.2 Perímetro

Em uma imagem binarizada o perímetro de um objeto é dado pela contagem dos pixels em seu contorno. Matematicamente o perímetro é calculado como mostrado na Equação 3.3. Onde $g(x, y) = 1$ se (x, y) pertence ao contorno de algum objeto na imagem, e $\delta(x, y) = 0$ no caso contrário.

$$P = \sum_x^{nx} \sum_y^{ny} g(x, y) \quad (3.3)$$

3.2.3 Média de intensidade

Seja $f(x, y)$ a intensidade do pixel (x, y) de uma imagem representada em escala de cinza. O cálculo da intensidade média desta imagem é dada pela Equação 3.4, onde N é o número total de pixels da imagem.

$$m = \frac{1}{N} \sum_x^{nx} \sum_y^{ny} f(x, y) \quad (3.4)$$

3.3 Classificador

O aprendizado de algoritmos de aprendizado de máquina pode ser agrupada em dois grupos principais: **não supervisionado** e **supervisionado**. No aprendizado não supervisionado os exemplos de treinamento não são rotulados e busca-se identificar as classes através dos conjuntos das *features* que possibilitam o agrupamento de exemplos semelhantes e a separação de exemplos diferentes através do padrão desses dados. Técnicas de aglomeração (*clustering*) são exemplos de classificação não supervisionada [SOLOMON; BRECKON, 2000]. Já no aprendizado supervisionado, as classes de cada objeto são conhecidas e informadas para o classificador a fim de levantar-se o modelo de classificação.

O principal objetivo no aprendizado supervisionado é aprender um modelo baseado em dados utilizados para treinamento, possibilitando futura previsão de dados nunca antes vistos, sendo o termo "supervisionado" referente ao fato de que, os dados oferecidos para o aprendizado pertencem a uma grupo conhecido e possuem um rótulo que os indica pertencentes a esse grupo. Classificação é uma sub-categoria do aprendizado supervisionado onde o objetivo é prever o rótulo da classe do dado novo baseado em observações passadas [RASCHKA; MIRJALILI, 2015]. Por exemplo, é dado a um sistema a tarefa de classificação de imagens de carros e caminhões. Nesses contexto as imagens são os dados de entrada que, por ser um aprendizado supervisionado, irão possuir os rótulos das classes, ou seja, irão possuir informações se são imagens de carros ou caminhões. Dessa forma, com base nas imagens previamente fornecidas, o sistema deverá ser capaz de, dada uma imagem nova, informar qual a classe desse novo dado, ou seja, se é uma imagem de carro ou caminhão. O sistema que irá executar a tarefa de classificação recebe o nome de classificador. Assim, de maneira simplificada, classificador é um sistema computacional que tem a capacidade de inferir a classe de novo dado baseada nas informações obtida de dados fornecidos anteriormente [FOSSENG, 2013].

Classificação supervisionada se propõe a resolver o problema de inferir uma função baseada num grupo de treinamento. Em muitas abordagens a função utilizada para realizar a classificação é uma função de distância e, frequentemente a função de distância utilizada é a distância Euclidiana [FOSSENG, 2013]. A vantagem de utilizar essa função de distância se deve ao fato de ser uma função simples. Uma vez obtida a função de classificação, está é comumente avaliada referente a sua capacidade de generalização em relação a dados novos. Para essa, utiliza-se um grupo de teste e uma métrica de avaliação, acurácia, por exemplo, para determinar a qualidade do classificador na tarefa de classificação [FOSSENG, 2013].

Para a classificação supervisionada é necessário um conjunto de dados contendo as instâncias da classe representada pelo vetor de *features* juntamente com o rótulo da classe de cada instância. Seja um objeto descrito por J variáveis (*features*) $\mathbf{x} = [x_1, x_2, \dots, x_J]$. Dada a classe $c (c = 1, 2, 3, \dots, C)$ do objeto I_c , tem-se que a matriz $\mathbf{X}_c$ é a matriz formada pelo conjunto de objetos e *features* e $\mathbf{y}_c$ é o vetor coluna formado pelos rótulos de cada objeto

da classe c . Todo o *dataset* de treino é agrupado na matriz $\mathbf{X}(IxJ) = [X_1, X_2, \dots, X_C]$ contendo todas as classes juntamente com a matriz coluna $\mathbf{y}(Ix1) = [y_1, y_2, \dots, y_C]$ contendo os rótulos dos objetos, sendo $I = \sum_{c=1}^C I_c$ o número total de objetos. A Figura 28 ilustra a junção dos objetos na classe o no *dataset* de treinamento.

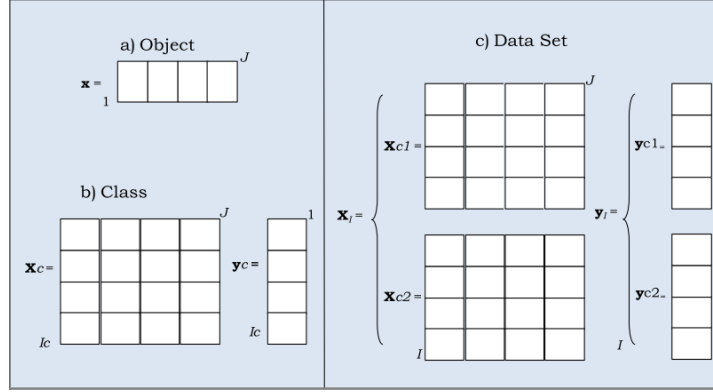


Figura 28 – Representação do: a) objeto, b) classe c) todo o *dataset*, adaptado de [MEDINA et al., 2013].

3.3.1 Métodos e Métricas de Avaliação

Para avaliar um classificador são necessários métodos e métricas de avaliação. Os métodos de avaliação são como o *dataset* será dividido de forma que a etapa de treinamento não seja feita com os mesmos dados que a etapa de teste, onde busca-se avaliar a capacidade de generalização do classificador para determinar o tipo de dado, frente a um dado nunca antes analisado. Existem diferentes métodos como *holdout*, *k-fold cross-validation*, *leave-one-out*, dentre outros. Os métodos utilizados nesse trabalho foram o *holdout* e o *k-fold cross validation*.

O processo de desenvolvimento de um classificador envolve a etapa de seleção dos parâmetros para otimizar o seu desempenho e a etapa de teste do desempenho do classificador, para isso são utilizados os métodos de validação. Nesse processo é comum a separação do *dataset* em grupo de teste e treino e, esse último, separado novamente em treino e validação. Essa separação é utilizada para que não se utilize o grupo de teste várias vezes durante a etapa da seleção dos parâmetros, o que pode favorecer a ocorrência de *overfit*, além de não ser uma boa prática no desenvolvimento de algoritmos de aprendizagem de máquina [RASCHKA; MIRJALILI, 2015].

O método *holdout* é uma abordagem clássica e popular para validação de performance quanto a verificação da generalização do algoritmo de Inteligência Artificial [RASCHKA; MIRJALILI, 2015]. Nessa abordagem o *dataset* é dividido em grupo de treino e teste e dentro desses grupos, os dados são distribuídos aleatoriamente. Como o nome sugere, o grupo de treino é utilizado para treinar o algoritmo para levantar o modelo de seleção, posteriormente, o grupo de teste, com dados inteiramente novos, é utilizado

para avaliar a performance do classificador frente a classificação de dados que não foram utilizados anteriormente.

No método *K-fold cross validation* o *dataset* é, aleatoriamente, dividido em K grupos de mesmo tamanho. Para avaliação do modelo, são utilizados K-1 grupos para treinamento do algoritmo e 1 grupo para avaliação da performance. Esse procedimento é repetido K vezes, sem que haja a repetição do mesmo grupo para avaliação. Nessa forma, no final, são obtidas K valores de métricas de avaliação, acurácia por exemplo, em que a performance final é atribuída a média entre as K métricas obtidas nas iterações, a Figura 29 mostra esquematicamente o funcionamento do método. Segundo Raschka e Mirjalili [2015] a razão por utilizar todo o *dataset* de treinamento diversas vezes através do *K-fold cross validation* é que oferecer mais amostras de treinamento, geralmente leva a um resultado mais preciso e um modelo mais robusto.



Figura 29 – Representação esquemática do *K-fold cross-validation*, [RASCHKA; MIRJALILI, 2015].

Já as métricas de avaliação são valores numéricos quantitativos utilizados para expressar o desempenho do classificador frente a algum critério. As métricas de avaliação são, em sua maioria, obtidas através dos valores fornecidos pela Matriz de Confusão. Dentre as diversas métricas existentes, serão utilizadas nesse trabalho a acurácia, a taxa de erro de previsão e a curva ROC juntamente com sua área que será melhor detalhada a seguir.

A Matriz de Confusão, ver Figura 30, ilustra a performance dos algoritmos de Inteligência Artificial e, conseqüentemente, dos classificadores. Nela são expressas as relações entre as classes reais dos dados e as classes previstas pelo algoritmo através de 4 valores: Verdadeiro Positivo (*True Positive*), Falso Negativo (*False Negative*), Falso Positivo (*False Positive*) e Verdadeiro Negativo (*True Negative*) [RASCHKA; MIRJALILI, 2015]. Sendo:

Nesse trabalho não há condições "negativas" e "positivas" como quando deseja-se

		Predicted class	
		<i>P</i>	<i>N</i>
Actual class	<i>P</i>	True positives (TP)	False negatives (FN)
	<i>N</i>	False positives (FP)	True negatives (TN)

Figura 30 – Matriz de Confusão, [RASCHKA; MIRJALILI, 2015].

detectar a presença ou não de um tumor em um conjunto de células, por exemplo. As relações ocorrem pelas classes, ou seja, a classe "principal", comparada com o positivo é o grampo de suspensão, enquanto que a classe que é análoga ao negativo é o amortecedor de vibração. Assim, o valor dado pelo Verdadeiro Positivo é composta pelos dados que são grampos e foram classificados como grampos, enquanto que o valor dado pelo Verdadeiro Negativo é composto pelos dados que não são grampos, portanto são amortecedores, e não foram classificados como grampos. Já os Falsos Negativos foram classificados como amortecedores mas eram grampos e o valor de Falso Positivo foram os dados classificador como grampo sendo amortecedores.

- **Verdadeiro Positivo (VP)**: Dados classificados **corretamente** como **sendo** da classe "principal";
- **Falso Negativo (FN)**: Dados classificados **incorretamente** como **não sendo** da classe "principal";
- **Falso Positivo (FP)**: Dados classificados **incorretamente** como **sendo** da classe "principal";
- **Verdadeiro Negativo (VN)**: Dados classificados **corretamente** como **não sendo** da classe "principal";

A acurácia (**ACC**) pode ser entendida como a proporção entre as previsões corretas em relação ao total de previsões, seu cálculo é dado pela Equação 3.5 através dos valores da Matriz de Confusão.

$$ACC = \frac{VP + VN}{FP + FN + VP + VN} \quad (3.5)$$

A taxa de erro de previsão (**ERR**) é a representação do percentual total de erros na previsão das classes realizadas pelo classificador. Seu cálculo é dado pela Equação 3.6.

$$\mathbf{ERR} = \frac{\mathbf{FP} + \mathbf{FN}}{\mathbf{FP} + \mathbf{FN} + \mathbf{VP} + \mathbf{VN}} = 1 - \mathbf{ACC} \quad (3.6)$$

A curva *Receiver Operation Characteristics* (ROC) é um método de análise, avaliação, comparação e seleção de classificadores baseados em sua performance. Segundo Majnik e Bosnić [2013] a análise através da curva ROC tem ganhado uma intensa popularidade na área de aprendizado de máquina. Atualmente, a utilização como métrica de validação de algoritmos de aprendizado de máquina tem se tornado indispensável e a introdução da utilização nessa área pode ser encontrada em [FAWCETT, 2006].

A curva traz uma relação de desempenho relacionando duas métricas obtidas através da Matriz de confusão, a Taxa de Verdadeiro Positivo (**TVP**) e Taxa de Falso Positivo (**TFP**), detalhadas, respectivamente das Equações 3.7 e 3.8.

$$\mathbf{TVP} = \frac{\mathbf{VP}}{\mathbf{FN} + \mathbf{VP}} \quad (3.7)$$

$$\mathbf{TFP} = \frac{\mathbf{FP}}{\mathbf{VN} + \mathbf{FP}} \quad (3.8)$$

A **TVP** é a relação entre o número de acertos (Verdadeiros Positivos) em relação ao número total de resultados positivos definido pelo classificador. Já a **TFP** é a relação entre o número de "falsos alarmes" em relação número de classificações incorretas do classificador. De forma simplificada, a curva ROC mostra a relação entre a taxa de acertos em relação a taxa de "falsos alarmes" de um classificador. Essa relação, além de mostrar o desempenho do classificador, é especialmente importante em algumas aplicações, como quando utilizada em classificações médicas, por exemplo. Considerando um exemplo da utilização de um classificador para classificar um conjunto de dados referentes as células de tumores de pacientes, é mais importante ter uma **TVP** elevada, mostrando um elevado número de acertos do classificador, contudo, também não é desejável que a **TFP** seja alta, pois haverá inúmeros pacientes sendo diagnosticados com câncer de maneira incorreta, trazendo inúmeros transtornos e desconforto aos pacientes.

Na curva ROC, o **TVP** é representado do eixo Y e a **TFP** é representada no eixo X. Segundo Majnik e Bosnić [2013] a forma clássica de geração da curva é realizada através de diferentes pares cartesianos obtidos variando-se o *threshold* em relação a probabilidade de classificação das amostras, variando-se assim os valores de **TVP** e **TFP** do classificador.

Um exemplo de curva ROC é demonstrado na Figura 31, onde são ilustrado 4 exemplo de classificadores. Segundo Raschka e Mirjalili [2015] a diagonal da curva ROC pode ser interpretado como uma escolha aleatória, classificadores que se encontram abaixo da diagonal apresentam desempenho pior do que uma escolha aleatória. O classificador perfeito encontra-se no canto superior esquerdo da curva com valor de **TVP** igual a 1 e **TFP** igual a 0.

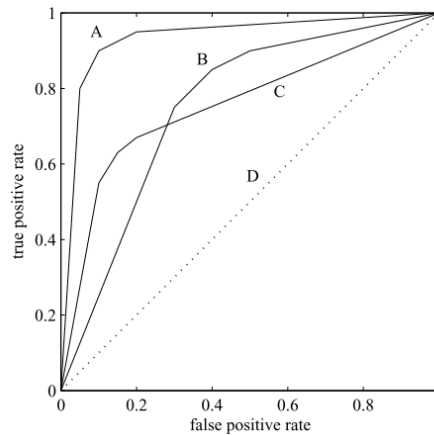


Figura 31 – Exemplo de curva ROC com quatro classificadores, [MAJNIK; BOSNIĆ, 2013].

Analisando os exemplos apresentados na Figura 31 observa-se que o classificador **D** apresenta performance igual de uma escolha aleatória e o que apresenta melhor desempenho é o classificador **A** por estar mais direcionado ao canto superior esquerdo do gráfico. Segundo Majnik e Bosnić [2013] comparar o desempenho de classificadores através da análise da curva ROC pode ser uma tarefa não trivial, devido ao cruzamento entre classificadores, como exemplificado entre os classificadores **B** e **C**. Para auxiliar nessa tarefa, utiliza-se a métrica *Area under the curve* (AUC) em tradução livre, área abaixo da curva. Esse valor é, na maioria da vezes, concordante com o valor de acurácia do classificador e apresenta valores variando entre 0 e 1, sendo este último, o valor máximo [RASCHKA; MIRJALILI, 2015].

3.3.2 *k*-Nearest Neighbors(*k*NN)

O *k*NN é um dos mais antigos e simples métodos de classificação supervisionada, podendo ser utilizado também para regressão multivariada. Contudo, apesar da sua simplicidade, esse algoritmo ainda apresenta, frequentemente, resultados competitivos quando comparados com outras técnicas de classificação [FOSSENG, 2013]. Sua aplicação pode ser encontrada nas mais diversas áreas, como na área médica [RISKY; HARSONO; SIGIT, 2017], robótica [MARKOM et al., 2017] e [NASCIMENTO, 2018], e botânica [ELNEMR, 2017], por exemplo. O método clássico do *k-nearest neighbors* é um método não paramétrico que determina a classe do objeto desconhecido como sendo a mesma classe da maioria dos **k** vizinhos mais próximos. A relação de proximidade é definida segundo

uma métrica de distância [MEDINA et al., 2013]. Esta é uma das mais simples regras de decisão não paramétricas. O termo não paramétrico aqui citado refere-se ao fato de que não há nenhum conhecimento prévio acerca da distribuição estatística dos dados. O kNN assume que as instâncias dos dados são independentes e igualmente distribuídas, assim, as instâncias que estão próximas possuem a mesma classificação [SYED, 2014].

Neste algoritmo, a classificação do vetor de *features* de uma amostra é dada por votação entre as instâncias catalogadas durante o treinamento e presentes entre as k instâncias mais próximas desta amostra. A predição ocorre de acordo com a maioria dos votos e os vetores mais próximos são determinados através de uma métrica de distância. Segundo [FOSSENG, 2013], na literatura são apresentadas diversas métricas de distância como: Manhattan, Minkowski, Mahalanobis, Euclidiana, dentre outras, contudo a métrica mais utilizada é a Euclidiana [KIM et al., 2017]. O processo de classificação juntamente com representação de k pode ser visto na Figura 32

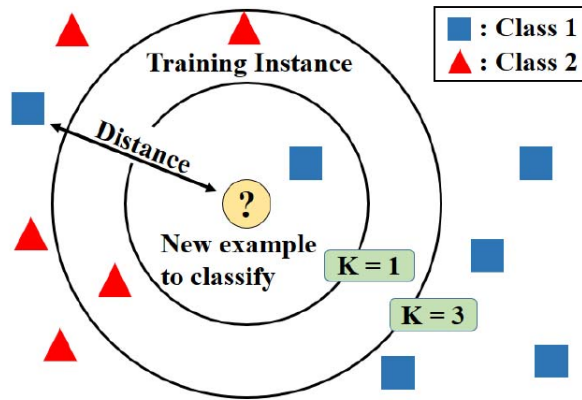


Figura 32 – Exemplo de classificação utilizando kNN, [KIM et al., 2017].

Analisando a Figura 32 é possível observar que a classe do objeto muda com o valor de k . Para $k=1$, o objeto é classificado como a classe 1, enquanto que, considerando o valor de $k=3$, o objeto passa a ser classificado como sendo da classe 2.

Considerando um novo objeto $\mathbf{x}_t$ para ser classificado, segundo [MEDINA et al., 2013], as etapas na classificação com o *k-Nearest Neighbors* são:

1. Calcular a distância entre o objeto $\mathbf{x}_t$ e todos os objetos no *dataset* de treinamento $\mathbf{X}$.
2. Armazenar e organizar os objetos de acordo com a distância em relação ao objeto desconhecido $\mathbf{x}_t$. Para os k objetos mais próximos, contar o número de objetos de cada classe.
3. Classificar o objeto $\mathbf{x}_t$ como pertencente a classe da maioria dos k vizinhos mais próximos.

Como mencionado, a distância mais utilizada é a distância Euclidiana, dada pela Equação 3.9 :

$$d_E(\mathbf{x}_t, \mathbf{x}_i) = \sqrt{\sum_{j=1}^J (\mathbf{x}_{tj} - \mathbf{x}_{ij})^2} \quad (3.9)$$

Onde $d_E(\mathbf{x}_t, \mathbf{x}_i)$ é a distância Euclidiana entre os pontos $\mathbf{x}_t$ e $\mathbf{x}_i$ e J o vetor de *features*.

Como visto anteriormente, o kNN armazena todos os dados de treinamento, isso faz desse tipo de classificador um "*lazy classifier*". O termo *lazy* não refere-se a sua simplicidade, mas ao fato de que, diferentes de outros classificadores, o kNN não aprende uma função discriminante para realizar a separação entre as classes, ele apenas memoriza todo o conjunto de dados de treinamento [RASCHKA; MIRJALILI, 2015]. Também chamados de métodos de aprendizado baseado em instância, os "*lazy classifier*" são definidos por 3 propriedades [SYED, 2014]:

1. Armazenam todos os dados durante o processo de treinamento.
2. A generalização que deveria ocorrer durante o treinamento é postergada até a predição de um novo valor pois, qualquer nova consulta é respondida comparando-se o novo caso com os dados de treinamento.
3. Dos dados de treinamento são inferidas semelhanças com a nova instância a ser avaliada.

O kNN apresenta inúmeras vantagens, podem ser destacadas:

1. É um classificador não paramétrico, o que significa que a distribuição dos dados não precisa ser conhecida [MEDINA et al., 2013].
2. É conceitual e computacionalmente simples pois é baseado em distância [KOWALSKI; BENDER, 1972].
3. É multi-classe e não assume separabilidade linear entre os dados [MEDINA et al., 2013].
4. É bastante estável, pequenas mudanças nos dados de treinamento não impactam significativamente nos resultados de classificação [BREIMAN, 1996].
5. Realiza aprendizado com pequena quantidade de dados podendo ter a performance melhorada pela incremento de dados [BAY, 1998].

6. Uma vez que o aprendizado é baseado na memória, a classificação se ajusta imediatamente com a adição de novos dados de treinamento [RASCHKA; MIRJALILI, 2015]

Contudo, há também as desvantagens apresentados pelo kNN, as principais são:

1. O modelo clássico do kNN não apresenta bom desempenho quando o *dataset* é desbalanceado quanto ao número de amostras por classe. Caso os dados de treinamento apresentem muita diferenças entre o número de amostras de uma classe 1 em detrimento de uma classe 2, o desempenho do classificador será comprometido pois haverá maior probabilidade de um novo objeto ser classificado como da classe 1 [MEDINA et al., 2013].
2. Ocorrência de *overfit* devido a susceptibilidade a "maldição da dimensionalidade" [RASCHKA; MIRJALILI, 2015]. Esse efeito pode ser resumido como o decréscimo da acurácia de classificação com o aumento do número de variáveis, considerando um número constante de amostras. Isso se deve ao fato de que, com o aumento de dimensões do espaço das *features*, a amostra fica com uma métrica de distância tão elevada que dificulta a estimação dos vizinhos mais próximos. Como solução pra esse problema, aumenta-se o tamanho do *dataset* ou aplica-se técnicas de redução de dimensionalidade, como o LDA (*Linear Discriminant Analysis*), por exemplo [MEDINA et al., 2013].
3. Para o modelo clássico, a complexidade computacional aumenta linearmente com o aumento do número de amostras do *dataset*, ao menos que utilize-se outras técnicas de implementação mais eficiente como o KD-tree, por exemplo [RASCHKA; MIRJALILI, 2015].

4 Metodologia e Experimentos

4.1 Estrutura de Teste

Os experimentos desse trabalho foram realizados no laboratório de robótica do BIR, no SENAI CIMATEC. Nele foi montado uma estrutura de testes possuindo um condutor do tipo LINNET com diâmetro de 18,3 mm, o mesmo encontrado em linhas de transmissão com 1 condutor por fase. Para coleta dos dados, montou-se uma plataforma de testes que pode ser visto na Figura 33. Essa plataforma foi montada com uma possível configuração da unidade de tração do ELIR onde o Hokuyo foi montado a frente da estrutura, centralizado e inclinado para que fosse possível fazer um escaneamento de todo o obstáculo antes que houvesse colisão com o mesmo. Essa configuração permite a coleta dos dados com o LiDAR para posterior tratamento, extração de *features* e aplicação do classificador para classificação dos obstáculos encontrados na linha de transmissão. Na Figura 33 a linha vermelha representa o feixe laser emitido pelo Hokuyo que intercepta, dependendo da posição da plataforma no cabo, o cabo ou partes dos obstáculos para formar a imagem que será posteriormente utilizada para classificação. O feixe do LiDAR foi posicionado de baixo para cima para que, em condições reais, o plano de fundo fosse o céu, não existindo muitos objetos e facilitando o tratamento do plano de fundo, como feito também em [POULIOT; RICHARD; MONTAMBAULT, 2012] e [RICHARD; POULIOT; MONTAMBAULT, 2014].

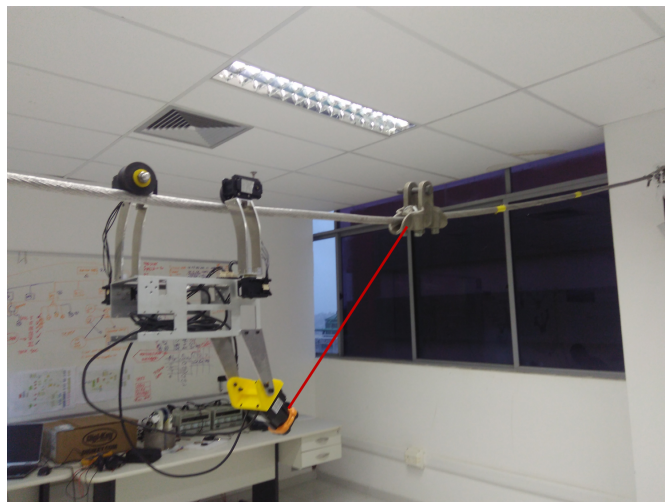


Figura 33 – Plataforma de Teste.

A plataforma é feita em alumínio e possui 4 juntas. Duas delas são juntas rotacionais, utilizadas para movimentar as garras num movimento de "abre e fecha", permitindo o acoplamento e desacoplamento da plataforma no cabo. As outras duas juntas são de revolução, sendo responsáveis pela movimentação linear da plataforma ao longo do cabo.

As juntas são atuadas por motores Dynamixel, fabricados pela empresa Robotis. São utilizados 2 tipos de motores diferentes. Para juntas rotacionais e de revolução são utilizados motores MX-28R e MX-106R, respectivamente. A classe MX da Dynamixel é composta por servo-motores multi-volta que operam com tensão de alimentação na faixa de 10 a 14,8V. A utilização destes motores apresenta vantagens, como: possibilidade de utilização no modo *joint* (como junta rotacional) ou no modo *wheel* (como junta de revolução), sistema de processamento interno e completa integração com o *framework Robotic Operation System* (ROS) .

O ROS possui também integração com o Hokuyo, possibilitando o controle e coleta de dados de maneira simples e direta. O ROS é um *framework* de código aberto, livre e colaborativo. Isso faz com que seu desenvolvimento e expansão seja muito mais rápido, uma vez que usuários de institutos de pesquisa, das indústrias e usuários comuns (aqueles que desenvolvem robótica por *hobby*) possam colaborar, em paralelo, para o aprimoramento dessa plataforma e de seus pacotes.

Outra característica interessante do ROS é a necessidade de pouca capacidade computacional, segundo Quigley et al. [2009] essa característica se deve a reutilização de outros projetos de código aberto, como os *drivers*, sistemas de navegação, simuladores, algoritmos de visão computacional, dentre outros. Dentre esses projetos que possuem interação com o ROS pode-se citar o *OpenCV*, *OpenRAVE* e o Gazebo.

Com a construção dessa plataforma, foi possível a geração do *dataset* com as imagens geradas dos obstáculos pelo escaneamento com o LiDAR e posterior processamento e classificação utilizando a técnica proposta neste trabalho.

4.2 Levantamento de *Dataset*

Para a geração do *dataset* buscou-se uma maior variedade e representatividade, variando o local e a posição dos obstáculos no cabo. Os obstáculos foram posicionados em 4 locais diferentes, com o objetivo de variar a inclinação das posições dos obstáculos juntamente com o cabo, ver Figura 34. Com o intuito de ter uma representação mais próximas de condições reais, foram geradas também imagens inserindo trepidação do cabo, buscando simular a movimentação ocorrida devido a influência de vento no robô de inspeção, Figuras 35 e 36. Outra variação utilizada para a diversificação do *dataset* foi a de posicionamento dos obstáculos, variando-se lateralmente a disposição dos obstáculos em relação ao cabo, demonstrados na Figura 37. As Figuras 37(a) e 38(a) mostram a imagem dos obstáculos pela perspectiva do LiDAR.

Para a geração do *dataset* foram utilizados obstáculos reais posicionados no mesmo modelo de cabo presente nas linhas de transmissão de 1 condutor por fase encontradas em inúmeras linhas de transmissão aqui no Brasil.

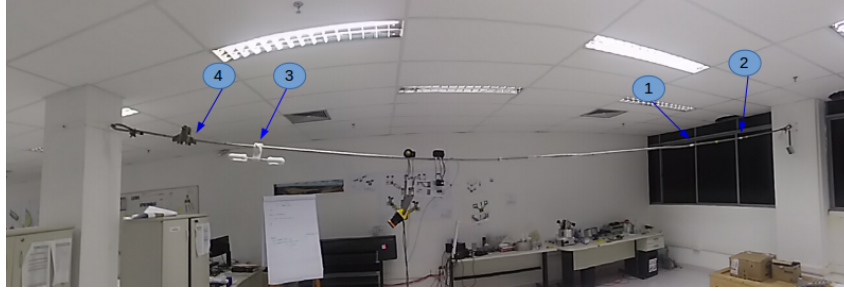


Figura 34 – Pontos de posicionamento dos obstáculos para coleta de imagens.

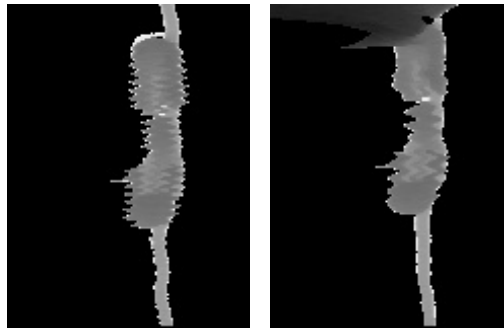


Figura 35 – Amortecedores distorcidos pela vibração da linha de transmissão.

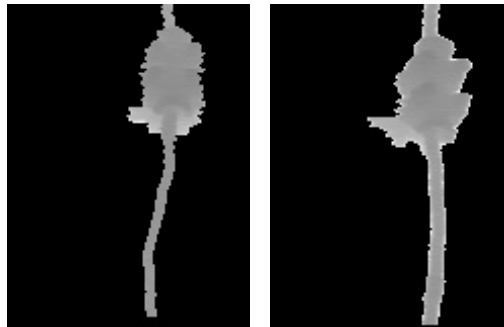
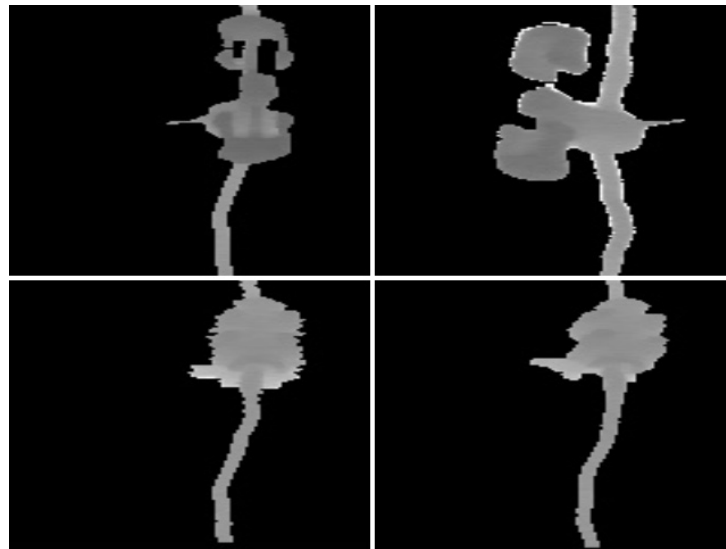


Figura 36 – Grampos distorcidos pela vibração da linha de transmissão.

Para variar ainda mais o *dataset* gerado, houve a adição de objetos que podem, em uma condição real, se prender aos obstáculos, como por exemplo, pipas e sacos plásticos. Esse objetos, ao se prender geram oclusão de parte do obstáculo, como pode ser visualizado nas Figura 38.

As imagens do *dataset* foram divididas em 2 grupos. O primeiro, chamado de *dataset* "sem oclusão", contém 104 de cada obstáculo, totalizando 208 imagens e foi gerado a partir das imagens dos objetos com diferentes posicionamentos, porém, retirando as imagens dos objetos com oclusão. O segundo grupo de imagens, chamado de *dataset* "com oclusão", contém 370 imagens, dividido igualmente entre as classes. Este *dataset* é composto por todas as diversidades de imagens geradas, incluindo trepidação no cabo, diferente posicionamento dos obstáculos, imagens geradas com os obstáculos posicionados em diferentes posições do cabo e com ocorrência de oclusão.

Como mencionado, as imagens do *dataset* foram geradas através do escaneamento

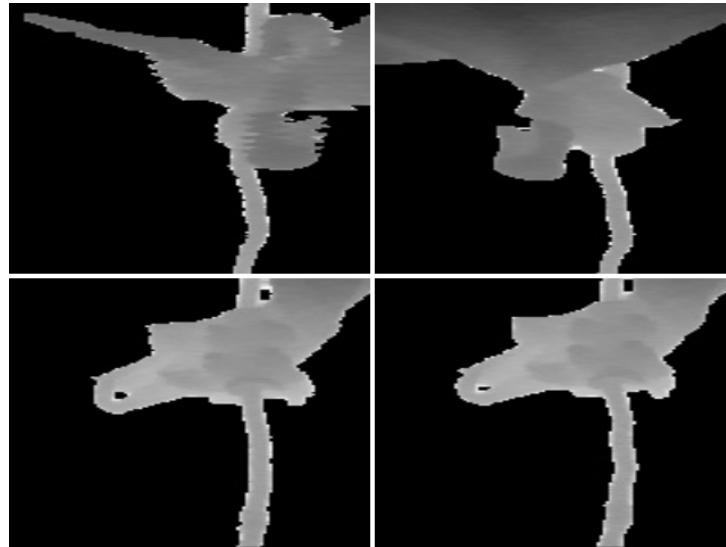
(a) *imagens reais*(b) *imagens geradas*Figura 37 – Imagens dos obstáculos e imagens do *dataset*.

realizado pelo LiDAR da marca Hokuyo. Este é um LiDAR 2D com abertura de 270° e que mede a distância entre o sensor e o ponto onde há reflexão do laser emitido (entre 0.1 m e 30 m) uma vez a cada 25 mili-segundos. Esse mesmo sensor foi utilizado em [POULIOT; RICHARD; MONTAMBAULT, 2012] e [RICHARD; POULIOT; MONTAMBAULT, 2014] para a detecção de obstáculos em linhas de transmissão. Nestes artigos foram realizados vários testes e esse sensor apresentou um resultado satisfatório em relação a aspectos como: interferência eletromagnética, medição com luz natural, vibração, dentre outros pontos relevantes a serem considerados na utilização em inspeção de linhas de transmissão.

Uma vez que é conhecida a área onde o cabo e os obstáculos podem ser encontrados, reduziu-se o ângulo de abertura do Hokuyo de 270° para 30° , diminuindo assim a quantidade de dados processados e, consequentemente, o esforço de processamento. Como a resolução do sensor é de $0,25^\circ$, utilizando-se uma abertura de 30° tem-se um vetor



(a) imagens reais com oclusão



(b) imagens geradas com oclusão

Figura 38 – Imagens dos obstáculos e imagens do *dataset* com oclusão.

horizontal com 120 medidas. Para converter esse vetor de medidas em imagens, utilizou-se o algoritmo ilustrado da Figura 39. Primeiro aplicou-se um "filtro de distância" o que implica na seleção de um *threshold* de distância das medidas fornecidas pelo LiDAR, mantendo as medidas abaixo do valor de *threshold* e zerando o resto das medidas. Uma vez que, pela configuração da plataforma de teste, sabe-se que a distância dos obstáculos e do cabo para o sensor é menor que 1 metro, utilizou-se esse valor como *threshold* de distância do "filtro". Assim, mesmo que a distância de medição do sensor seja de 30m, as medidas são filtradas e o cabo e os obstáculos presentes são isolados do plano de fundo que, no caso do laboratório consistiam no teto juntamente com as luminárias, mas, em uma situação de inspeção em ambiente externo, podem ser: galhos de árvores ou a estrutura das torres de transmissão, por exemplo. Em seguida são calculados a distância média D_m e a largura L do objeto interceptado pelo feixe do LiDAR, demonstrada na Equação 4.1.

Em Pouliot, Richard e Montambault [2012] apresentou-se a medição da largura e da distância média em relação ao Hokuyo. Utilizou-se então a medição da largura para detectar a existência de obstáculo na linha. A distância média $\mathbf{D_m}$ é calculada pela média entre as medidas de distância fornecidas pelo LiDAR. O largura do objeto interceptado pelo feixe do sensor é dada por:

$$\mathbf{L} = \mathbf{D_m} \delta \mathbf{n} \quad (4.1)$$

Onde $\mathbf{L}$ é a largura do objeto em metro, δ é a resolução angular do Hokuyo ($0,25^\circ$) e $\mathbf{n}$ o número de medidas não zeradas após a aplicação do "filtro de distância".

Após o cálculo da largura, o vetor de medidas fornecida pelo LiDAR é convertido em um valor de intensidade de imagem em escala de cinza. Cada medida é transformada em um pixel que vai possuir valor de intensidade entre 0 (preto) e 255 (branco), de acordo com a Equação 4.2 de normalização.

$$\mathbf{I_p} = 255 \left(\frac{\mathbf{d}}{\mathbf{th}} \right) \quad (4.2)$$

Sendo $\mathbf{I_p}$ a intensidade do pixel, $\mathbf{d}$ o valor de distância fornecido pelo LiDAR e $\mathbf{th}$ o valor do *threshold* de distância escolhido.

Criou-se então uma matriz $\mathbf{M}_{160 \times 120}$, com 160 linhas e 120 colunas que irá ser convertida em imagem, as dimensões foram escolhidas para manter a proporção de 640 x 480, proporção comum em imagens. Após a conversão do vetor de medidas em intensidade, esse vetor é concatenado como linha da matriz $\mathbf{M}$ até a formação da imagem.

Considerando que a aplicação do robô é em um ambiente semi-estruturado, ou seja, conhece-se inicialmente as condições do ambiente em que o robô será inserido, estipulou-se um *threshold* de 35 mm para se considerar que há obstáculos no cabo. A diferença entre a largura do cabo (18,3 mm) e os obstáculos, 45mm como largura inicial do amortecedor de vibração e 50mm do grampo de suspensão, possibilitaram a utilização com sucesso desse método. As linhas das imagens foram adicionadas a cada 4 mm de deslocamento linear da plataforma. Essa medida foi obtida através da odometria da plataforma. Para finalizar a geração da imagem, utilizou-se novamente a medida da largura do obstáculo. Uma vez que o obstáculo foi detectado e a largura volta as dimensões do cabo, movimenta-se por mais 10 mm e então salva-se a imagem. No caso onde não foi possível a geração de imagens de maneira automática, quando houve oclusão por parte de algum objeto, por exemplo, utilizou-se comando manual para finalizar a geração da imagem e salvar o arquivo.

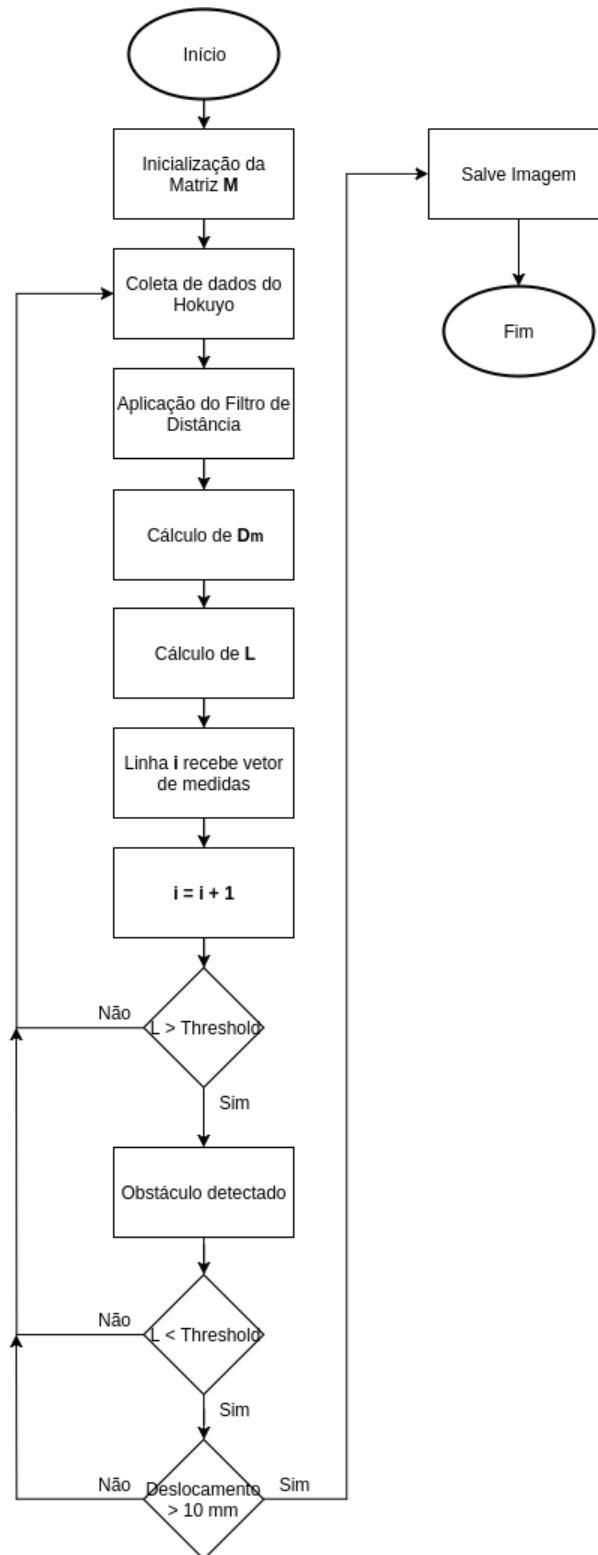


Figura 39 – Fluxograma de geração de imagens.

4.3 Processamento das Imagens

Como mencionado por [RASCHKA; MIRJALILI, 2015], a etapa de pré-processamento é fundamental para o processo de classificação. É necessário que as imagens sejam condicionadas para que a extração das *features*, que servirão de características representativas de

cada classe, seja realizada da melhor maneira.

4.3.1 Pré-Processamento das Imagens e Extração de *Features*

A etapa de pré-processamento das imagens consistiu em 3 etapas:

- Binarização da imagem;
- Isolamento dos obstáculos na imagem (retirada do cabo);
- Tratamento de degradação provocado na etapa de isolamento dos obstáculos;

Para o processamento das imagens utilizou-se a biblioteca OpenCV¹. OpenCV é uma biblioteca *open source* de visão computacional e aprendizado de máquina. Composta por mais de 500 algoritmos, é amplamente utilizada por empresas, grupos de pesquisas e instituições governamentais para aplicações como: auxílio na navegação de robôs, reconhecimento de faces em sistemas de vigilância, monitoramento de equipamentos em minas, verificação de qualidade de peças em fábricas, dentre outras. Dentre os usuários dessa ferramenta estão mais de 47 mil pessoas na comunidade e empresas como Google, Microsoft, Sony, IBM, dentre outras, [OPENCV, 2018]. Entre os algoritmos de processamento de imagem fornecidos pela OpenCV estão as funções para encontrar a área e o perímetro de uma imagem, realizar binarização e outras funções que foram necessárias nesse trabalho.

Para a binarização das imagens foi necessário primeiramente a conversão de plano de fundo. Após a aplicação do filtro de distância ele foi zerado, contudo, seu valor deveria possuir o maior possível (255), visto que são os pontos que estão mais distantes do LiDAR. Essa conversão foi necessária para que fosse possível a binarização da imagem utilizando o *threshold* da distância referente ao cabo, fazendo assim com que, o que estiver abaixo do cabo, valor com intensidade menor, receba o valor de 255 enquanto o que tiver acima do cabo, valor de intensidade maior, receba o valor 0. Observando as imagens e o valor de intensidade dos pixels que formavam o cabo, utilizou-se o valor de *threshold* de intensidade igual a 150.

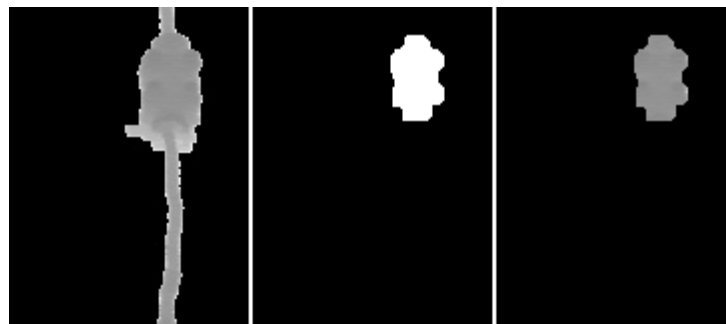
Após a binarização, ainda permaneceram alguns pedaços de cabos nas imagens. Esses pedaços foram retirados utilizando um algoritmo desenvolvido para remoção do cabo nas imagens. Sabendo-se que o cabo possui na imagem aproximadamente 10 pixels, analisou-se todas as linhas da imagem e, para as que possuíam quantidade de pixels não nulos igual ou inferior a 10, zerou-se todos os pixels da linha, convertendo numa linha toda preta a apagando o cabo. Após a utilização desse algoritmo, houve degradação de alguns pontos do obstáculo, utilizou-se então a operação morfológica de fechamento, operação utilizada em imagens binárias para o fechamento de falhas e buracos na imagem e correção de contorno [SOLOMON; BRECKON, 2000].

¹ www.opencv.org

Assim, chegou-se as imagens em condições para utilização das funções da OpenCV para a extração das *features* já descritas, medição de área, perímetro e para a geração de uma máscara com o intuito de calcular a média de intensidade apenas do obstáculos nas imagens originais em escala de cinza. A Figura 40 mostra as etapas do processamento das imagens para extração das *features*. Da esquerda para a direita tem-se: a imagem original, imagem binarizada para extração dos valores de área e perímetro e a imagem apenas do obstáculo em escala de cinza para obtenção da média das intensidades.



(a) Etapas de processamento de imagem do amortecedor de vibração



(b) Etapas de processamento de imagem do grampo de suspensão

Figura 40 – Etapas do processamento das imagens.

4.3.2 Classificação das Imagens

Para a classificação dos obstáculos através do algoritmo kNN utilizou-se a biblioteca de aprendizado de máquina em Python, Scikit-Learn². Esta é uma biblioteca que fornece implementações de vários algoritmos do estado da arte na área de aprendizado de máquina, através de uma interface de fácil utilização, inclusive para não especialistas na área de *software*, como pesquisadores na área de Biologia e Física, por exemplo. Sua utilização é favorecida pelas mais de 300 páginas web de documentação, contendo explicações sobre funcionamento de algoritmos, tutoriais, instruções de instalação e mais de 60 exemplos de aplicações em contexto real [PEDREGOSA et al., 2011].

Para a classificação, as *features* extraídas das imagens foram organizadas em um arquivos no formato CSV (*comma-separated value*), separados nos diferentes *datasets*

² www.scikit-learn.org

a serem avaliados na seção 5. Além das *features*, como o kNN foi utilizado em modo supervisionado, também foram adicionados os indicadores de classe para posterior validação, sendo o valor 1 atribuído para grampos de suspensão e 2 para amortecedores de vibração.

Os *datasets* comumente utilizados nos algoritmos do Scikit-Learn são arquivos no formato *Comma Separated Value* (CSV) (em tradução livre, "valores separados por vírgula"). Estes arquivos de *dataset* são formados por uma lista em que as linhas são as amostras e as *features* são organizadas nas colunas, com os valores separados por vírgula. Esses valores foram então avaliados e validados na seção 5.

4.3.2.1 Avaliação de Robustez

Com o objetivo de investigar ainda mais o desempenho e robustez do classificador, realizou-se uma análise com a inserção de ruído nas imagens. Esses ruídos tem a função de simular possíveis problema que possam ocorrer em testes realizados em ambiente externo e, por isso, mais próximo da aplicação real. Essa avaliação foi realizada a fim de reduzir as simplificações e limitações impostas pela realização em teste de laboratório

Ruídos são efeitos aleatórios que causam variação nas imagens e podem ser visualizados como grãos. Esses distúrbios podem ocorrer por efeitos de luminosidade ou térmicos nos componentes do sensor e podem ocorrer no momento de captura dos dados ou durante a transmissão. Assim, ruído pode ser resumido como, quando pixels na imagens apresentam valores de intensidade diferentes do valor real [VERMA; ALI, 2013]. Esses valores distorcidos produzem efeitos indesejáveis nas imagens como: bordas irreais, "borramento" dos objetos, distúrbios no plano de fundo da imagem, quinas inexistentes [BOYAT; JOSHI, 2015]. Segundo Verma e Ali [2013], dentre as principais fontes de ruído, pode-se destacar:

- Efeitos ambientais interferindo durante a aquisição dos dados;
- Alteração na temperatura dos sensores gerando ruído nos valores coletados;
- Inserção de ruído durante a transmissão dos dados;
- Presença de poeira na interface de captura de dados, pode favorecer a ocorrência de ruídos;

Dentre os diversos tipos de ruídos existentes, escolheu-se o *salt&pepper* e o *speckle*. Esses tipos foram escolhidos devido aos fatores que os causam poderem ser encontrados nas condições reais de aplicação para inspeção de linha de transmissão. De acordo com Boyat e Joshi [2015] o ruído *salt&pepper* ocorre comumente na transmissão de dados ou no erro de conversores analógicos digitais. Esses fatores podem ocorrer com frequência num ambiente com intensa interferência eletromagnética, como encontrado nas linhas de transmissão,

principalmente quando há falha nas blindagens eletromagnéticas dos componentes e dos cabos no robô. Já o *speckle* é encontrado em sistemas de medição utilizando laser, radar ou sensores acústicos [BOYAT; JOSHI, 2015], por isso acreditou-se ser representativo no contexto da medição de distância utilizando o LiDAR.

O ruído *salt&pepper*, também conhecido como ruído independente ou ruído aleatório, se caracteriza pela ocorrência de pontos brancos (*salt*) ou pretos (*pepper*) espalhados de maneira aleatória na imagem, causando distorções nas linhas e bordas, por exemplo.

Já o ruído *speckle* é um ruído multiplicativo, ou seja, esse ruído pode ser modelado como valores aleatórios multiplicados pelos valores de intensidade dos pixels da imagem original. A imagem com ruído pode ser expressa pela Equação 4.3 , [VERMA; ALI, 2013].

$$\mathbf{J} = \mathbf{I} + \mathbf{n} * \mathbf{I} \quad (4.3)$$

Onde $\mathbf{J}$ é a imagem com ruído, $\mathbf{I}$ é a imagem original e $\mathbf{n}$ é um ruído uniforme com média $\mathbf{o}$ e variância $\mathbf{v}$.

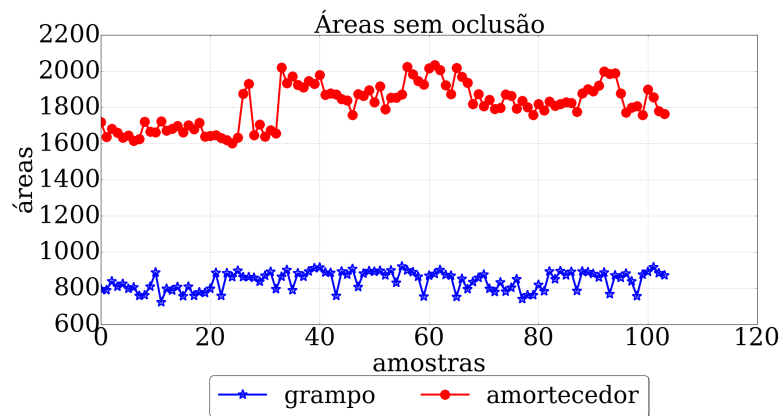
Para inserção do ruídos nas imagens, utilizou-se a biblioteca aberta Scikit-Image³. Nessa biblioteca existe a função *random_noise* que insere ruído nas imagens desejadas.

³ www.scikit-image.org

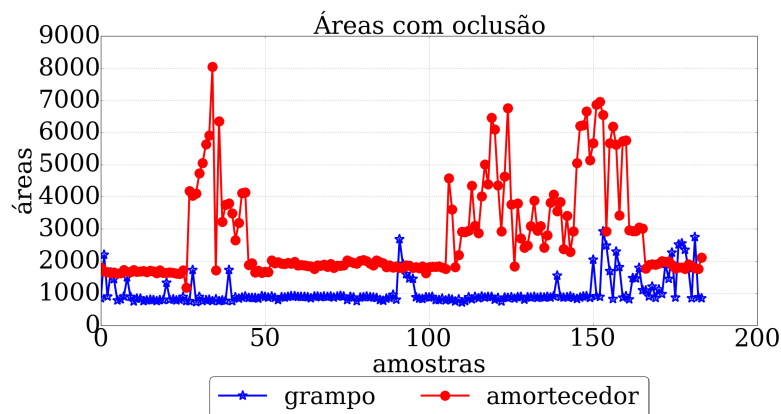
5 Resultados e Discussões

5.1 Análise das *Features*

Os testes foram realizados comparando os *datasets* "sem oclusão" e "com oclusão". Após a etapa de processamento das imagens as *features* de área, perímetro e média de intensidade foram comparadas em ambos os *datasets*. As Figuras 41, 42 e 43 mostram o comportamento desses valores das *features* em relação as amostras levantadas. Os valores de área e perímetro são calculados em pixels, enquanto que a média de intensidade é dada em intensidade de pixel. Esses gráficos foram utilizados para analisar o comportamento de cada *feature* individualmente quando comparadas entre os diferentes obstáculos e as distorções causada pela ocorrência de oclusão.



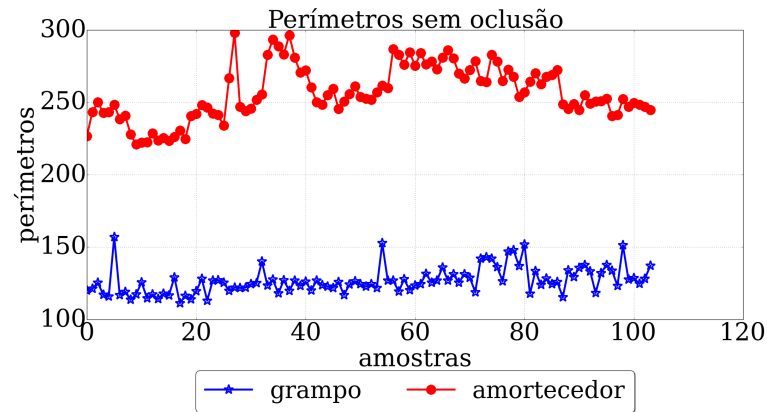
(a) Área das imagens sem oclusão



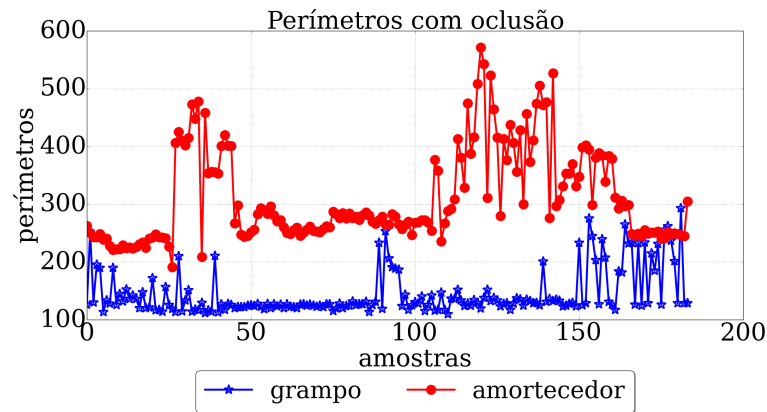
(b) Área das imagens com oclusão

Figura 41 – Área dos obstáculos.

Quando analisado as imagens sem oclusão, os valores de área, perímetro e média de intensidade são consideravelmente diferentes entre os obstáculos. Contudo, quando há



(a) Perímetro das imagens sem oclusão



(b) Perímetro das imagens com oclusão

Figura 42 – Perímetro dos obstáculos.

oclusão dos obstáculos, esses valores são distorcidos ocorrendo, em alguns momentos, uma sobreposição entre os gráficos do grampo de suspensão e amortecedor de vibração. Essas distorções nas imagens podem ser vistas na comparação entre as Figuras 37(b) e 38(b) e nas imagens binarizadas e em escala de cinza das Figuras 44 e 45.

5.2 Análise do *Dataset* no Espaço das *Features*

Considerando que o algoritmo de kNN recebe o vetor de *features*, também foram analisados o comportamento das possíveis combinações de *features* duas a duas. Nas Figuras 46, 47 e 48 são mostrados as *features* comparadas duas a duas. Através da análise dos gráficos é possível notar a separabilidade linear entre as classes de obstáculos, quando se observa o *dataset* "sem oclusão". Ao analisar o *dataset* "com oclusão", observa-se que este não é linearmente separável, por haver uma sobreposição entre as classes no espaço de vetores das *features* utilizadas. A sobreposição ocorre devido distorção ilustrada na secção 5.1 que causa também a sobreposição entre o comportamento individual das *features*. Contudo, como o kNN não prescinde de uma separabilidade linear entre as

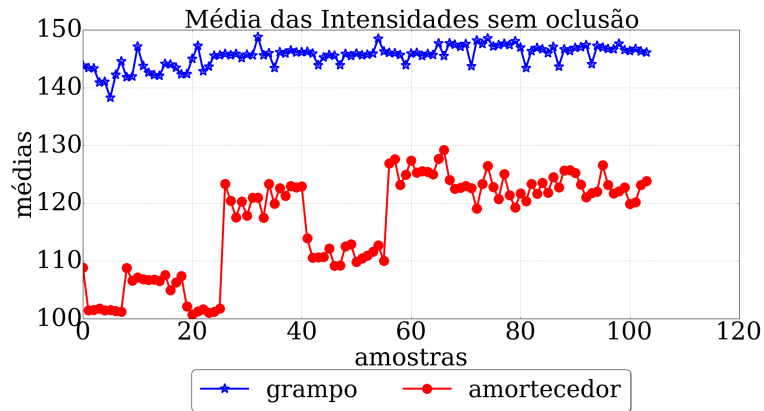
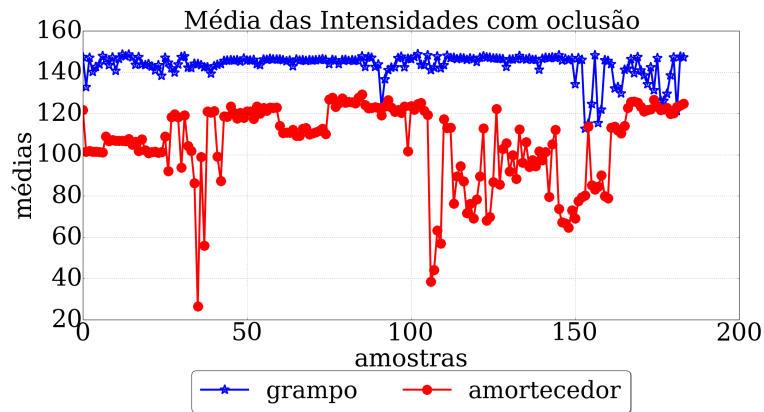
(a) *Intensidade Média das imagens sem oclusão*(b) *Intensidade Média das imagens com oclusão*

Figura 43 – Intensidade Média dos obstáculos.



Figura 44 – Imagens do amortecedor de vibração, da esquerda para a direita: imagem original, imagem binarizada, imagem em escala de cinza apenas do obstáculo.

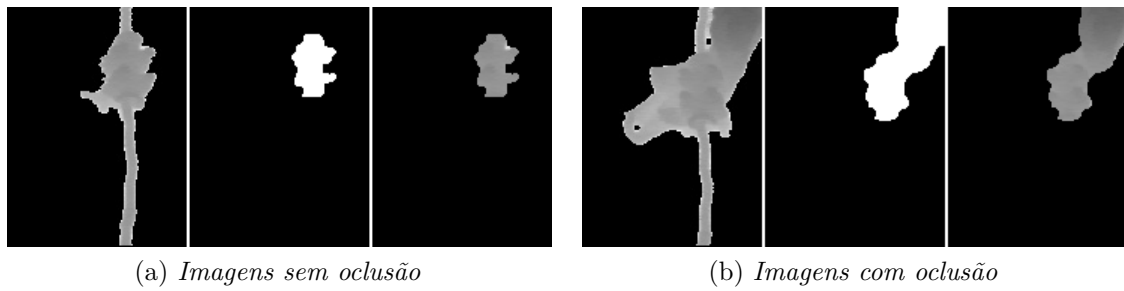
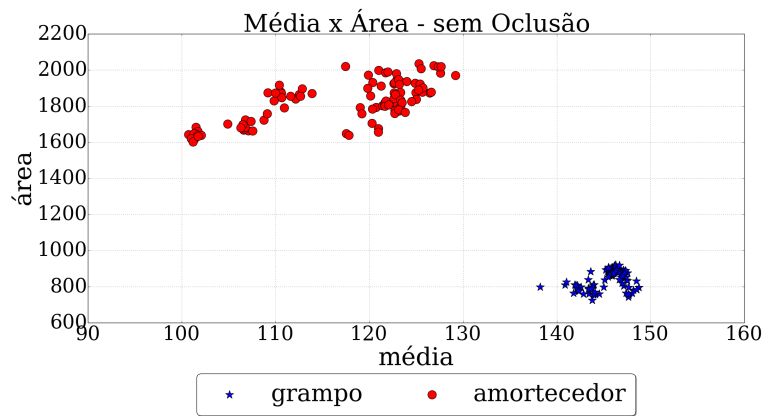
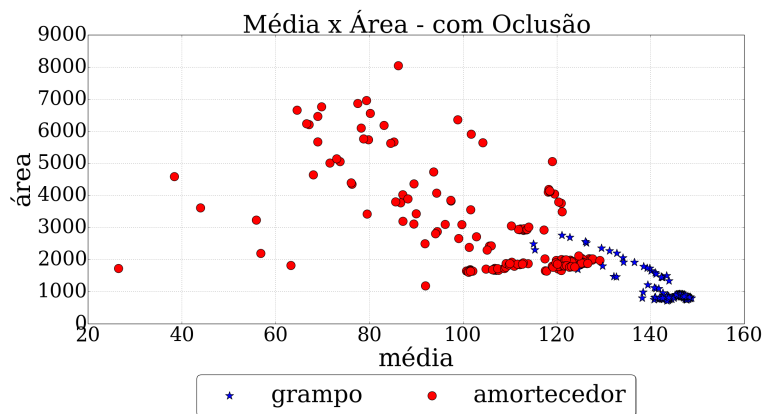


Figura 45 – Imagens do grampo de suspensão, da esquerda para a direita: imagem original, imagem binarizada, imagem em escala de cinza apenas do obstáculo.

features, consegue superar bem essa sobreposição e apresentar bons resultados, como será evidenciado posteriormente na seção 5.3.



(a) *Média $\times$ Área das imagens normais*



(b) *Média $\times$ Área das imagens com oclusão*

Figura 46 – Comparação entre média e área dos obstáculos.

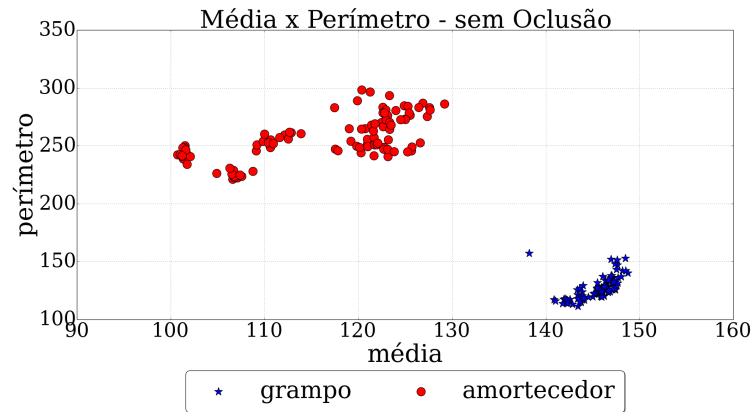
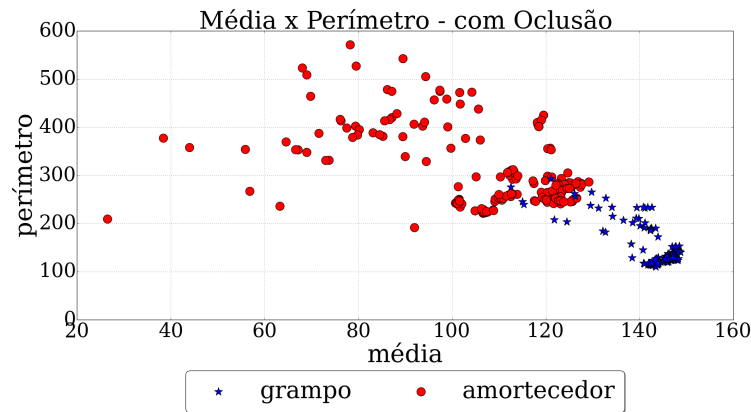
(a) Média $\times$ Perímetro das imagens normais(b) Média $\times$ Perímetro das imagens com oclusão

Figura 47 – Comparação entre média e perímetro dos obstáculos.

5.3 Classificação

Neste trabalho primeiramente utilizou-se o método *holdout* para separação do *dataset* em parte de treino e teste, sendo 70% para treino e 30% para teste. A parte de treino foi separada novamente através do método *K-fold cross-validation* com $K=10$ para realização de avaliação e seleção dos parâmetros do classificador.

Na etapa de avaliação do classificador, utilizou-se todo o *dataset* sem a separação realizada na etapa de avaliação do modelo e utilizou-se o método *K-fold cross-validation* com valor de $K=10$. A escolha pela utilização do *K-fold cross-validation* é justificada pelo tamanho do *dataset*, sendo esse método indicado para *datasets* com tamanho médio, de centenas de dados, segundo Syed [2014]. Já o valor de $K=10$ é o padrão mais utilizado segundo Raschka e Mirjalili [2015] por apresentar melhor relação entre variância e enviesamento dos dados.

Dentre as diversas métricas existentes, serão utilizadas nesse trabalho a acurácia e a taxa de erro de previsão, além de apresentada a curva ROC das configurações dos

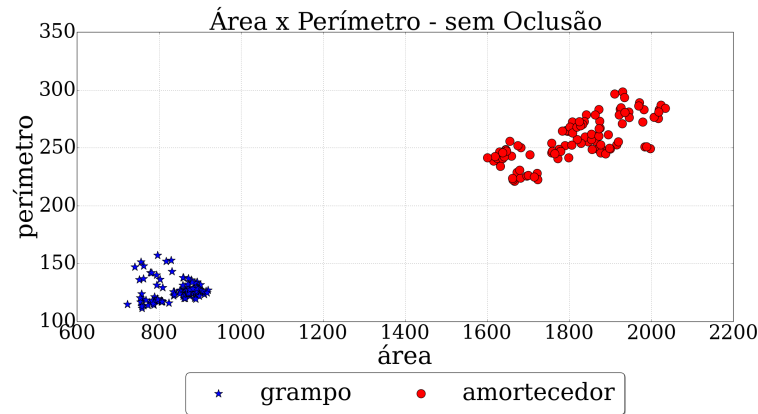
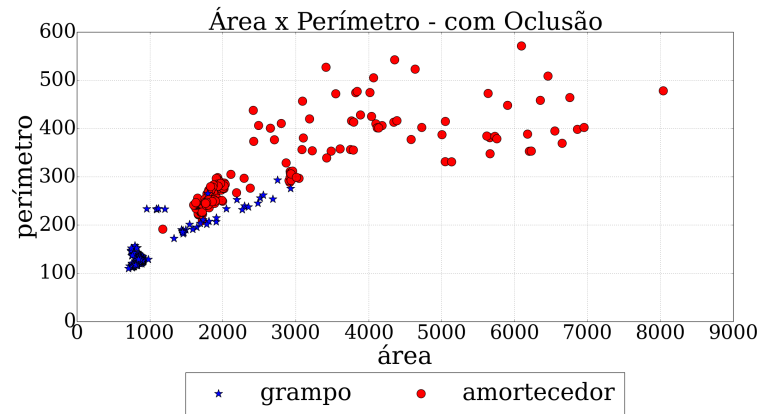
(a) $\text{Área} \times \text{Perímetro}$ das imagens normais(b) $\text{Área} \times \text{Perímetro}$ das imagens com oclusão

Figura 48 – Comparação entre área e perímetro dos obstáculos.

classificadores tendo como métrica o valor de AUC da curva média obtida após as 10 iterações.

5.3.1 Seleção de Modelo

5.3.1.1 Avaliação de normalização

O cálculo dos valores das *features* são realizados de formas diferentes e, por isso, podem apresentar discrepância de valor e escala de grandeza. Devido isso, é necessário que os dados passem por uma etapa de normalização. Segundo Elnemr [2017] é uma estágio importante no processo de classificação e garante que os valores maiores não irão dominar ou afetar os resultados. A normalização das *features* melhora o desempenho da maioria dos algoritmos de aprendizado de máquina utilizados para classificação, dentre eles, se encontra o kNN. A importância da utilização da normalização pode ser melhor entendida com um exemplo. Assumindo que a classificação será realizada com 2 *features*, uma com escala de 1 a 10 e outra de 1 a 1.000, quando a métrica de classificação for calculada, no caso do kNN a distância Euclidiana, os valores de 1 a 1.000 terão predominância em relação aos de

menor escala, fazendo com que esses dados sejam mais "importantes" do ponto de vista da avaliação pela métrica, o que não é necessariamente verdadeiro, isso faz com que haja uma análise comprometida dos dados no processo de classificação [RASCHKA; MIRJALILI, 2015].

Devido a sua importância, a normalização dos dados é comumente utilizada juntamente com a classificação através do kNN, como pode ser visto em [ELNEMR, 2017], [STEENWIJK et al., 2013] e [ZAHED; NAJIB; TAJUDDIN,]. Existem diferentes formas de normalizar as *features*. Segundo [RASCHKA; MIRJALILI, 2015] umas das principais formas de normalização é a ***min-max scaling***, técnica utilizada em [ELNEMR, 2017] e [ZAHED; NAJIB; TAJUDDIN,]. Como o nome sugere, essa normalização leva em consideração os valores mínimos e máximos da *feature* e normaliza para que o valor fique na faixa entre 0 e 1. O valor normalizado é obtido pela Equação 5.1, onde $\mathbf{x}_{norm}^i$ é o novo valor normalizado, $\mathbf{x}^i$ é o valor original e $\mathbf{x}_{min}$ e $\mathbf{x}_{max}$ são, respectivamente, os valores mínimos e máximos apresentado pelos dados.

$$\mathbf{x}_{norm}^i = \frac{\mathbf{x}^i - \mathbf{x}_{min}}{\mathbf{x}_{max} - \mathbf{x}_{min}} \quad (5.1)$$

Quando as *features* são normalizadas, o classificador apresenta uma melhoria de desempenho como ilustrado na Figura 49. Nesta imagem pode-se observar que o classificador com as *features* normalizadas apresentou desempenho igual ou superior, todas as vezes, quando comparado a condição das *features* não normalizadas. A média de acurácia com os dados normalizados foi de **98,8%** enquanto que, com os dados não normalizados, foi de **94,6%**. Para essa validação, utilizou-se os 70% do *dataset* "com oclusão" separados inicialmente para treino, utilizando-se para isso o método *10-Fold cross-validation*.

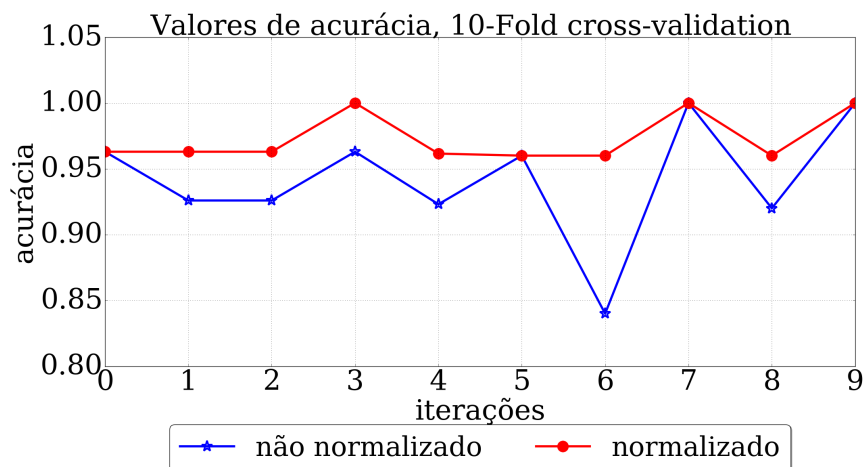


Figura 49 – Comparação de desempenho com dados normalizados e não normalizados

5.3.1.2 Avaliação de valores de k

Um método para avaliar o desempenho do classificador em relação a mudança de valor dos parâmetros é a **Curva de Validação**. No algoritmo kNN o único parâmetro a ser sintonizado é o valor de k , para gerar a curva, variou-se esse parâmetro de 1 a 21. Apenas valores ímpares foram selecionados devido a presença de 2 classes, dessa forma, não haveria possibilidade de ocorrer empate na votação para definir a classe do dado. A Figura 50 mostra a comparação entre o desempenho do grupo de treino e validação para cada valor de k . O gráfico foi gerado utilizando-se a média da acurácia, uma vez que para cada valor de k foi utilizado o método *10-fold cross-validation* com os 70% separado para treino do *dataset* "com oclusão".

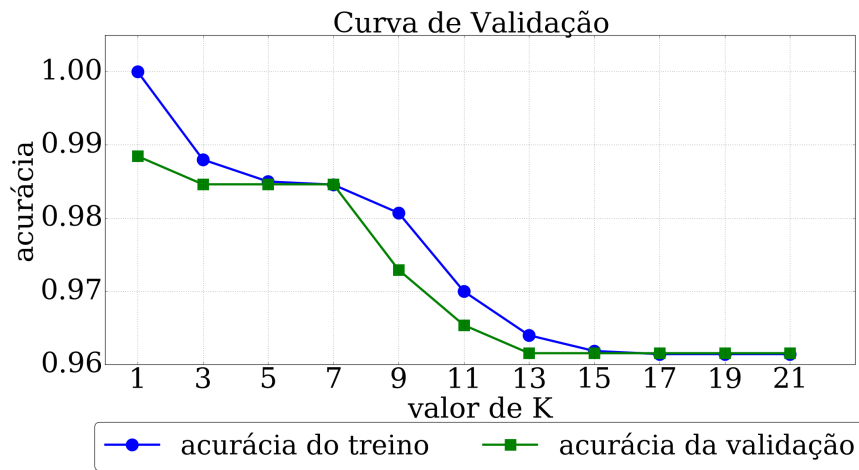


Figura 50 – Curva de validação.

Analisando a Figura 50 pode-se observar que os valores de k que apresentaram melhores desempenhos foram 5 e 7. Para valores pequenos ocorre *overfitting*, com bons valores de desempenho mas com diferença maior entre o grupo de treino e de validação. Entretanto, quando os valores de k aumentam, há a ocorrência de *underfitting*, demonstrado na proximidade de desempenho entre o grupo de treino e validação, contudo, o valor de acurácia cai de maneira geral.

5.3.1.3 Avaliação do comportamento das features

Com base nas avaliações desenvolvidas anteriormente, chegou-se que a configuração com melhor desempenho para o classificador é utilizando normalização das *features* e com os valores de k igual a 5 e 7. Assim foram analisadas as *features* e suas possíveis combinações para determinar o melhor desempenho do classificador. Primeiramente levantou-se a Matriz de Confusão das 14 combinações de configurações, ilustrados nas Tabelas 1 e 2.

Para geração das Matrizes de Confusão utilizou-se o método *holdout*, separando o *dataset* em 70% para treinamento e 30% para teste. A separação do *dataset* é feita de

maneira aleatória, então o valor de cada classe muda, contudo, o valor total de dados, referente aos 30% de teste, é mantido constante, sendo de 63 imagens para o *dataset* "sem oclusão" e 111 imagens para o *dataset* "com oclusão", devido ao tamanho diferente dos *datasets*, 208 e 370 imagens, respectivamente.

Durante o geração das Matrizes de Confusão, percebeu-se que os valores de acurácia para cada configuração variava, devido a separação aleatória gerada pela função do *Scikit-Learn*. Assim, para obter resultado mais uniforme, utilizou-se o método *10-fold cross validation* com o *dataset* completo incluindo os 30% separados anteriormente para teste, a Tabela 3 foi preenchida com as médias obtida nas 10 iterações.

Matriz de Confusão para <i>dataset</i> sem oclusão					
<i>Features</i>	k	VP	FN	FP	VN
Área	5	32	0	0	31
	7	34	0	0	29
Perímetro	5	31	0	0	32
	7	30	0	0	33
Média	5	32	0	0	31
	7	30	0	0	33
Média+Área	5	33	0	0	30
	7	32	0	0	31
Média+Perímetro	5	29	0	0	34
	7	33	0	0	30
Área+Perímetro	5	33	0	0	30
	7	37	0	0	26
Média+Área+Perímetro	5	26	0	0	37
	7	31	0	0	32

Tabela 1 – Tabela com Matrizes de Confusão para *dataset* "sem oclusão".

A Tabela 3 mostra a acurácia do classificador para as diferentes configurações de combinação de *features* e dois valores de **k** diferentes. São apresentados também o desvio padrão da acurácia de classificação representado pela letra σ . Os valores de σ apresentados foram apenas os diferentes de 0%, ou seja, para todos os valores que não possuam valor σ , subte-se que o desvio padrão foi de 0%, como em todas as classificações com o *dataset* "sem oclusão".

Analisando-se a Tabela 3 percebe-se que os resultados foram bastante próximos. Quando são avaliadas apenas as imagens que não possuem oclusão, o desempenho do classificador é de **100%** em todas as configurações, porém, quando há oclusão nas imagens, ocorre uma variação de desempenho. Nessa contexto, os desempenhos apresentam valores próximos, todos acima de **92,0%** de acurácia. A configurações que apresentaram melhor desempenho na presença de oclusão foram: a combinação da utilização de apenas a média das intensidades e a área, independente do valor de **k** juntamente com a utilização de todas as *features* e valor de **k** igual a 5, com valor de acurácia de **98,4% +/- 1,8%** e erro de

Matriz de Confusão para <i>dataset</i> "com oclusão"					
<i>Features</i>	k	VP	FN	FP	VN
Área	5	52	6	2	51
	7	44	9	2	56
Perímetro	5	55	4	0	52
	7	52	5	1	53
Média	5	49	5	0	57
	7	49	3	0	59
Média+Área	5	55	5	0	51
	7	64	1	0	46
Média+Perímetro	5	49	0	0	62
	7	54	3	0	54
Área+Perímetro	5	59	2	1	49
	7	43	3	0	65
Média+Área+Perímetro	5	51	3	0	57
	7	51	2	0	58

Tabela 2 – Tabela com Matrizes de Confusão para *dataset* "com oclusão".

Tabela 3 – Tabela da média dos valores de acurácia do classificador.

1: *dataset* "sem oclusão" 2: *dataset* "com oclusão"

Média das medidas de acurácia				
	k=5		k=7	
<i>dataset</i>	1	2	1	2
Área	100%	93,0% σ :2,9%	100%	92,2% σ :3,3%
Perímetro	100%	95,9% σ :3,0%	100%	95,7% σ :3,1%
Média	100%	97,3% σ :2,1%	100%	97,3% σ :2,8%
Média+Área	100%	98,4% σ:2,1%	100%	98,4% σ:1,9%
Média+Perímetro	100%	97,6% σ :1,9%	100%	97,3% σ :2,5%
Área+Perímetro	100%	98,1% σ :1,8%	100%	97,3% σ :2,4%
Média+Área+Perímetro	100%	98,4% σ:1,8%	100%	98,1% σ :2,5%

1,6%. Considerando os valores de acurácia, a configuração com todas as *features* e valores de **k** igual a 5 será adotada nesse trabalho como a melhor configuração, por apresentar um desvio padrão menos que as outras de mesma acurácia, e será melhor avaliada nas seções seguintes. A Figura 51 mostra a curva ROC e o valor de AUC dessa configuração. Nesta imagem é possível analisar que a curva média esta bem a esquerda e acima no gráfico, ilustrado o bom desempenho do classificador, sendo representado pelo valor de AUC de **0,991**. Só será apresentada com desempenho de classificadores inferiores a 100%, por acreditar que esse valor já é bastante representativo.

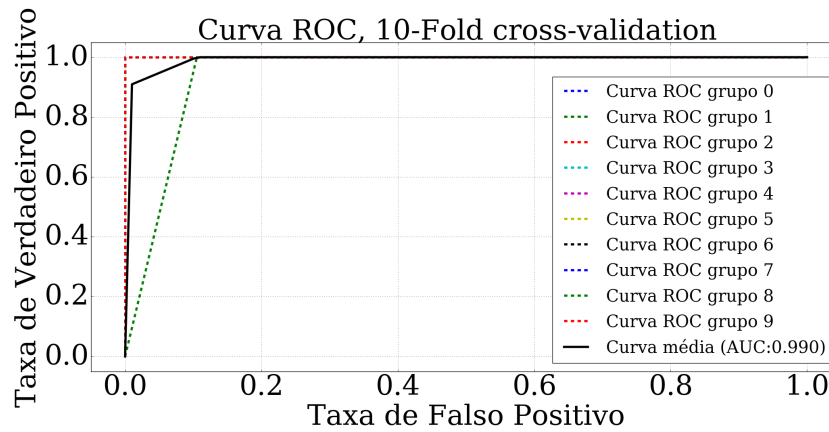
5.3.2 Avaliação de Robustez

Como apresentado na seção 4.3.2.1, foram inseridos ruídos nas imagens do tipo *salt&pepper* e *speckle* a fim de avaliar a robustez do classificador sob condições mais adversas

Tabela 4 – Tabela da média dos erros de previsão do classificador.

1: *dataset* "sem oclusão" 2: *dataset* "com oclusão"

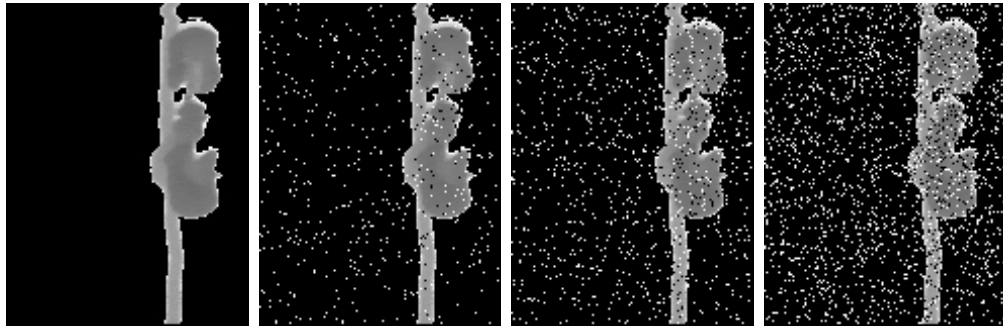
Média dos erros de previsão				
	k=5		k=7	
<i>dataset</i>	1	2	1	2
Área	0%	6,5%	0%	7,5%
Perímetro	0%	4,1%	0%	4,3%
Média	0%	2,7%	0%	3,0%
Média+Área	0%	1,6%	0%	1,6%
Média+Perímetro	0%	2,4%	0%	2,4%
Área+Perímetro	0%	1,9%	0%	1,9%
Média+Área+Perímetro	0%	1,6%	0%	1,9%

Figura 51 – Curva ROC com k e todas as *features* para classificação do *dataset* "com oclusão".

e incontrolladas. Na inserção do *salt&pepper* ajustou-se a proporção de pixels substituídos pelo ruído, a proporção de *salt* e *pepper* foi mantida igual e variou-se a proporção entre os pixels das imagens e com ruído, esses percentuais foram de : 5%, 10% e 20%, exemplos das imagens dos obstáculos após a adição de ruído pode ser vista nas Figuras 52 e 53. Por sua vez, quando deseja-se adicionar ruído do tipo *speckle* os parâmetros de controle são a média e a variância da função que será multiplicada pela imagem. A média foi mantida como padrão, valor 0 e para os valores de variância utilizou-se: 0,01, 0,5 e 0,1, nas Figuras 54 e 55 são apresentadas as imagens dos obstáculos com a inserção do ruído *speckle*.

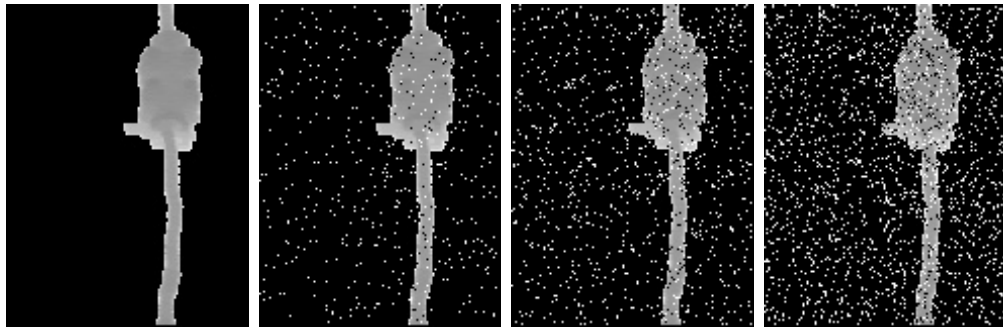
Analisando as Figuras 52 a 55 é possível observar que houve uma degradação significativa das imagens após a adição de ruído.

As Tabelas 5 e 7 mostram o desempenho dos classificador para ambos os *datasets* com a configuração com ruído. As Tabelas 6 e 8 apresentam de maneira explícita os valores de erro calculados a partir da Equação 3.6, a partir dos valores de acurácia obtidos pela



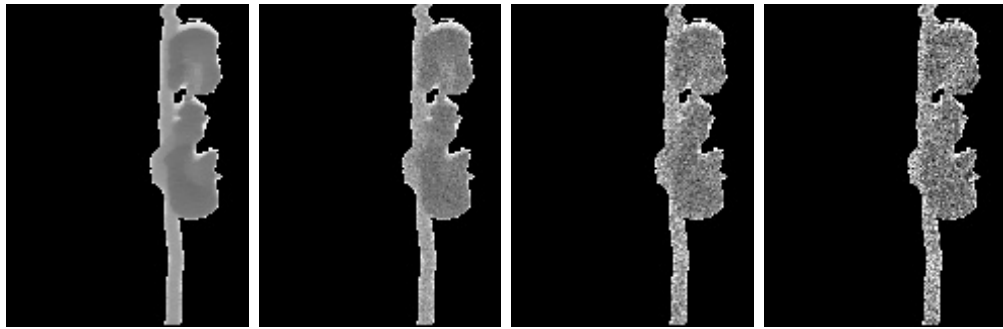
(a) *amortecedor sem ruído* (b) *salt&pepper 5%* (c) *salt&pepper 10%* (d) *salt&pepper 20%*

Figura 52 – Amortecedor de vibração com adição de ruído *salt&pepper*.



(a) *grampo sem ruído* (b) *salt&pepper 5%* (c) *salt&pepper 10%* (d) *salt&pepper 20%*

Figura 53 – Grampo de suspensão com adição de ruído *salt&pepper*.



(a) *amortecedor sem ruído* (b) *speckle com variância de 0,01* (c) *speckle com variância de 0,05* (d) *speckle com variância de 0,1*

Figura 54 – Amortecedor de vibração com adição de ruído *speckle*.

classificação.

Com os resultados expostos nas Tabelas 5 e 7 pode-se notar que o classificador apresentou pouco ou nenhuma degradação na presença de ruído. A melhor configuração continuou sendo com a presença das 3 *features*. Com a inserção do ruído *salt&pepper* não houve alteração significativa dos resultados de classificação, mantendo a acurácia média de **98,4%** e ocorrendo um pequeno aumento do desvio padrão nas classificações em

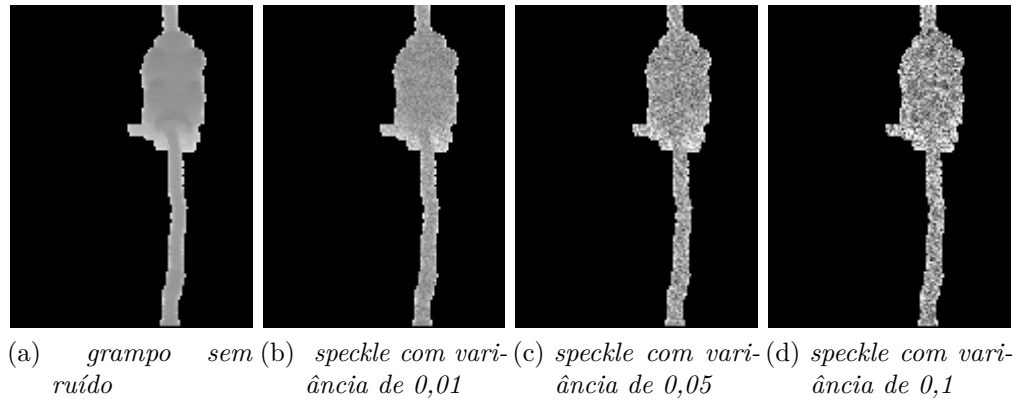


Figura 55 – Grampo de suspensão com adição de ruído *speckle*.

Tabela 5 – Tabela da média dos valores de acurácia do classificador com adição de ruído *salt&pepper*.

1: *dataset* sem oclusão 2: *dataset* com oclusão

Média das medidas de acurácia com adição de ruído <i>salt&pepper</i>						
	5%		10%		20%	
<i>dataset</i>	1	2	1	2	1	2
Área	100%	92,4% σ :3,3%	100%	92,7% σ :3,0%	100%	91,4% σ :3,5%
Perímetro	100%	95,4% σ :3,0%	100%	93,8% σ :4,2%	100%	94,9% σ :3,6%
Média	100%	97,0% σ :2,3%	100%	97,3% σ :2,9%	100%	97,6% σ :2,6%
Média+Área	100%	98,4% σ :2,2%	100%	98,3% σ :1,8%	100%	98,4% σ :1,8%
Média+Perímetro	100%	96,2% σ :2,4%	100%	96,7% σ :2,7%	100%	97,3% σ :2,5%
Área+Perímetro	100%	98,1% σ :2,5%	100%	98,1% σ :2,2%	100%	98,1% σ :2,7%
Média+Área+Perímetro	100%	98,4% σ :1,8%	100%	98,4% σ :1,8%	100%	98,4% σ :2,2%

geral, contudo não houve diferença significativa com as diferentes porcentagens de ruído. Acredita-se que a razão disso é o fato de que, uma vez que esse tipo de ruído é aplicado em toda a imagem e, como a proporção do obstáculo em relação a toda a imagem não é muito grande, boa parte do ruído é perdido nas etapas de pré-processamento, como mostra a Figura 56. Nestas imagens pode-se perceber que a forma da imagem binária não se altera e na imagens onde está apenas o obstáculo em escala de cinza, os pontos não distorcem significativamente o obstáculo, quando comparado as etapas ilustradas nas imagens 44 e 45, apesar de a imagem original apresentar distorções.

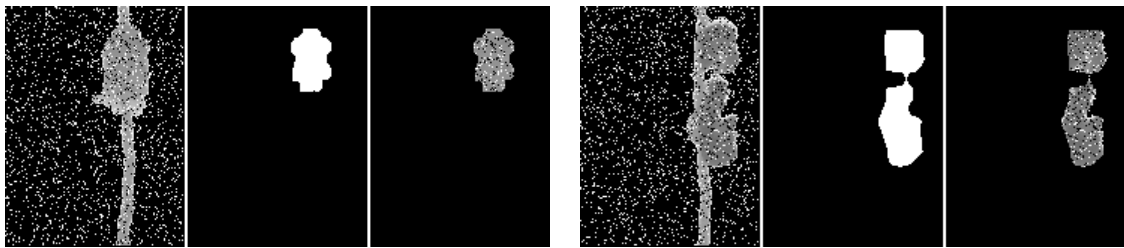
Já o ruído *speckle* degrada um pouco o desempenho do classificador e aumenta, de maneira geral, o desvio padrão das medições, apresentando acurácia média máxima de **97,8%** com variância de 0,01 e uma degradação do desempenho com o aumento do ruído. Uma vez que esse ruído é multiplicativo, como o fundo da imagem possui valor zero, o ruído só afeta realmente o cabo e obstáculo, assim, afeta mais a imagem do obstáculo e, conseqüentemente, o desempenho do classificador. Com a inserção desse ruído são apresentados, de maneira geral, as menores acurácias e maiores desvio padrão. Pode-se observar também, que para a configuração onde há apenas a utilização do perímetro ou

Tabela 6 – Tabela da média dos erros de previsão do classificador com adição de ruído *salt&pepper*.1: *dataset* sem oclusão 2: *dataset* com oclusão

Média dos erros de previsão com adição de ruído <i>salt&pepper</i>						
	5%		10%		20%	
<i>dataset</i>	1	2	1	2	1	2
Área	0%	8,5%	0%	7,0%	0%	8,9%
Perímetro	0%	4,5%	0%	6,4%	0%	4,9%
Média	0%	3,0%	0%	2,7%	0%	2,4%
Média+Área	0%	3,0%	0%	2,7%	0%	2,4%
Média+Perímetro	0%	2,7%	0%	3,3%	0%	2,2%
Área+Perímetro	0%	1,9%	0%	1,9%	0%	1,9%
Média+Área+Perímetro	0%	1,6%	0%	1,6%	0%	1,6%

Tabela 7 – Tabela da média dos valores de acurácia do classificador com adição de ruído *speckle*.1: *dataset* sem oclusão 2: *dataset* com oclusão

Média das medidas de acurácia com adição de ruído <i>speckle</i>						
	0,01		0,05		0,1	
<i>dataset</i>	1	2	1	2	1	2
Área	100%	90,7% σ :3,9%	100%	88,8% σ :3,8%	99,1% σ :1,4%	88,7% σ :4,7%
Perímetro	100%	92,7% σ :4,9%	100%	91,6% σ :2,9%	99,5% σ :1,8%	90,7% σ :4,2%
Média	100%	97,0% σ :3,3%	100%	97,0% σ :2,9%	100%	97,0% σ :3,5%
Média+Área	100%	96,5% σ :2,1%	100%	97,0% σ :3,2%	100%	97,0% σ :2,5%
Média+Perímetro	100%	96,5% σ :2,9%	100%	96,5% σ :2,9%	100%	97,3% σ :3,0%
Área+Perímetro	100%	94,6% σ :2,9%	100%	92,7% σ :3,0%	100%	91,3% σ :3,5%
Média+Área+Perímetro	100%	97,8% σ:2,3%	100%	96,7% σ :3,9%	100%	97,5% σ :2,6%



(a) Resultado das etapas de pré-processamento do grampo com ruído

(b) Resultado das etapas de pré-processamento do amortecedor com ruído

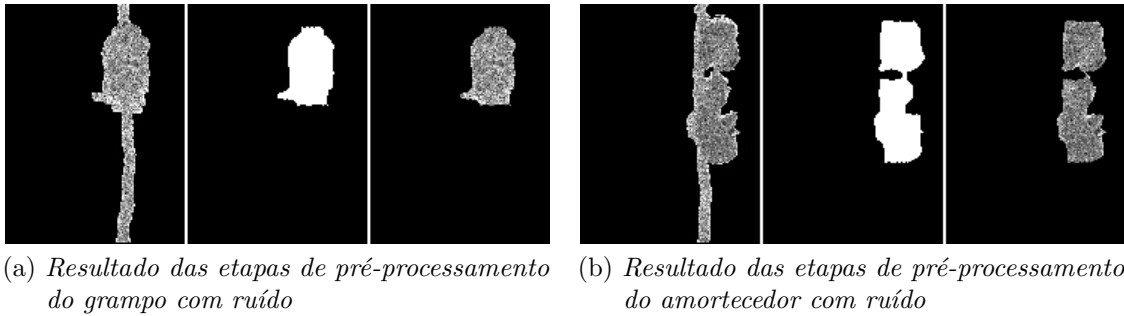
Figura 56 – Resultado das etapas de pré-processamento dos obstáculos com inserção de 20% de ruído do tipo *salt&pepper*, da esquerda para direita: imagem original, imagem binária do obstáculos, imagem em escala de cinza do obstáculo.

da área, o valor de acurácia para o *dataset* "sem oclusão" é inferior a 100%. A Figura 58 ilustra os resultados das etapas de pré-processamento das imagens após inserção de ruído do tipo *speckle*.

Os valores de acurácia de 98,4% com a inserção do ruído *salt&pepper* e de melhor

Tabela 8 – Tabela da média dos erros de previsão do classificador com adição de ruído *speckle*.1: *dataset* sem oclusão 2: *dataset* com oclusão

Média dos erros de previsão com adição de ruído <i>speckle</i>						
	0,01		0,05		0,1	
<i>dataset</i>	1	2	1	2	1	2
Área	0%	9,0%	0%	11,7%	1,0%	11,4%
Perímetro	0%	3,2%	0%	2,7%	0,5%	3,0%
Média	0%	2,7%	0%	3,0%	0%	7,5%
Média+Área	0%	3,3%	0%	2,4%	0%	3,9%
Média+Perímetro	0%	3,3%	0%	3,2%	0%	2,7%
Área+Perímetro	0%	5,4%	0%	3,3%	0%	7,6%
Média+Área+Perímetro	0%	2,2%	0%	3,0%	0%	2,5%

Figura 57 – Resultado das etapas de pré-processamento dos obstáculos com inserção de ruído do tipo *speckle* com 0,1 de variância, da esquerda para direita: imagem original, imagem binária do obstáculos, imagem em escala de cinza do obstáculo.

desempenho de 97,8% com o ruído *speckle* mostra a robustez da proposta de classificador apresentada neste trabalho, tanto no que se refere as etapas de pré-processamento das imagens, quando da escolha das *features* juntamente com o desempenho do classificador. Com o objetivo de investigar um pouco mais a robustez do classificador apresentado, inseriu-se ambos os ruídos nas imagens com os maiores valores utilizados nesse trabalho, ou seja, 20% para o ruído *salt&pepper* e variância de 0,1 para o *speckle*.

Apesar da inserção de ambos os ruídos, o desempenho do classificador se manteve apenas um pouco inferior do que quando foi adicionado apenas o ruído *speckle* em variância de 0,1, apresentando acurácia média de 97,6% com $\sigma=2,8\%$, reforçando a robustez do classificador que apresenta valores satisfatórios mesmo na presença de 2 tipos de ruído e interferência na geração das imagens como: trepidação do cabo, variação da posição dos obstáculos e oclusão.

Foi analisado também a curva ROC para o classificador com a inserção de ruído. Avaliou-se apenas as condições de menor ruído e maior ruído, como apresentado nas

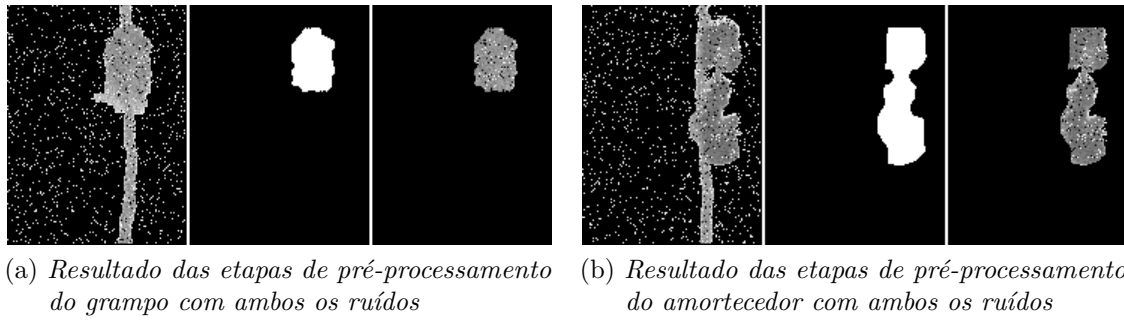


Figura 58 – Resultado das etapas de pré-processamento dos obstáculos com inserção de ambos os tipos de ruído, da esquerda para direita: imagem original, imagem binária do obstáculos, imagem em escala de cinza do obstáculo.

Figuras 59, 60, 61 e 62. Além do desempenho com a inserção dos dois tipos de ruídos apresentados na Figura 63.

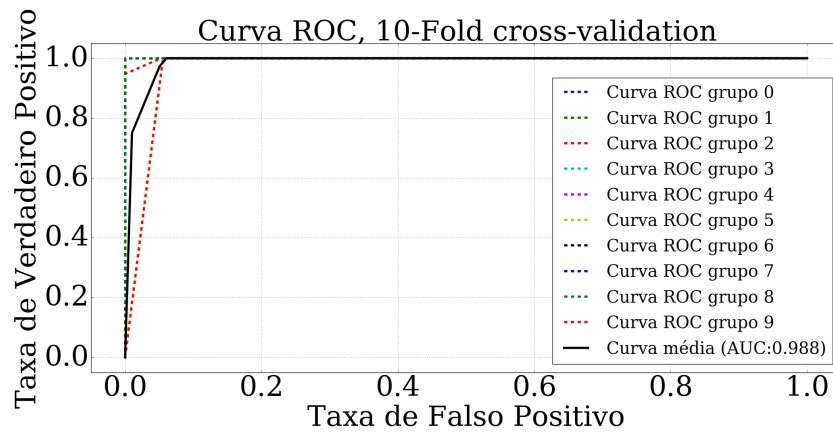


Figura 59 – Curva ROC com inserção de ruído do tipo *salt&pepper* com percentual de 5%

Pode-se observar que todas as curvas apresentam comportamento satisfatório se aproximando do canto superior esquerdo, como era de se esperar devido as valores de acurácia apresentados. Estes, por sua vez, também apresentam valores próximos dos valores de AUC de cada curva para sua respectiva configuração.

Como o objetivo desse trabalho é desenvolver um algoritmo de classificação para ser implementado no ELIR, é interessante possuir informações sobre tempo de execução do algoritmo e consumo de memória, para analisar a viabilidade de implementação, tanto em relação a tempo quanto a *hardware* possível de ser utilizado. Desenvolveu-se então um código da mesma forma que poderia ser implementado no robô. A partir da leitura de uma imagem fornecida por outra parte do sistema, realizou-se todas as etapas necessárias para a classificação: pré-processamento da imagem, extração de *features* e classificação, resultando na predição da classe da imagem utilizada. Uma vez que o tempo varia entre execuções diferentes, repetiu-se todo o processo de classificação dos obstáculos com 100 imagens

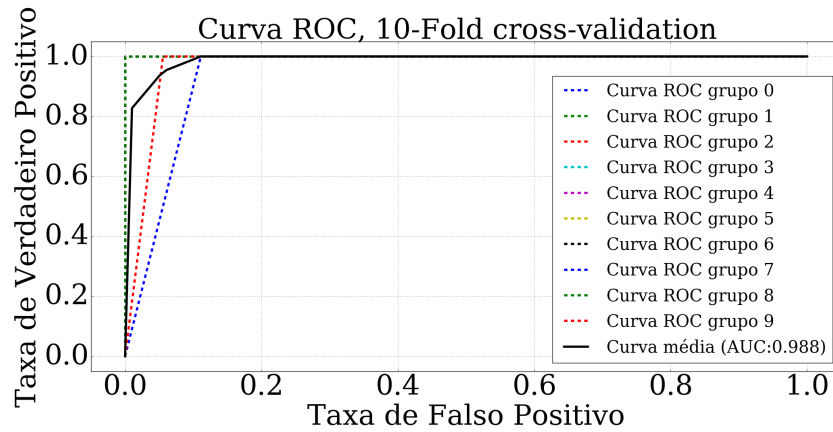


Figura 60 – Curva ROC com inserção de ruído do tipo *salt&pepper* com percentual de 20%

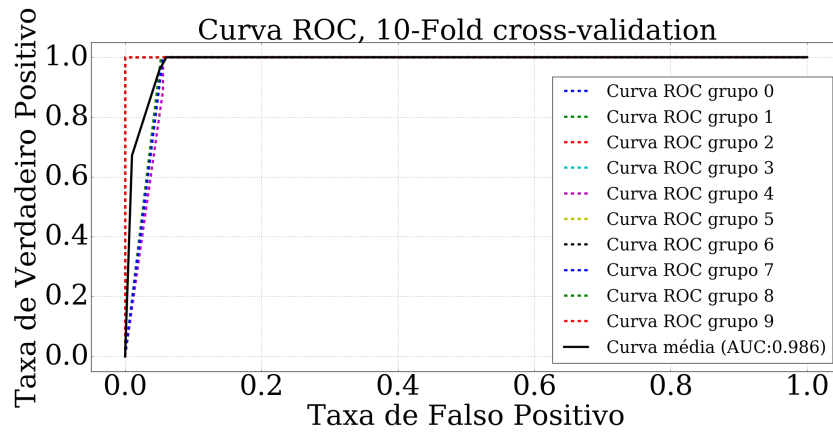


Figura 61 – Curva ROC com inserção de ruído do tipo *speckle* com variância de 0,01

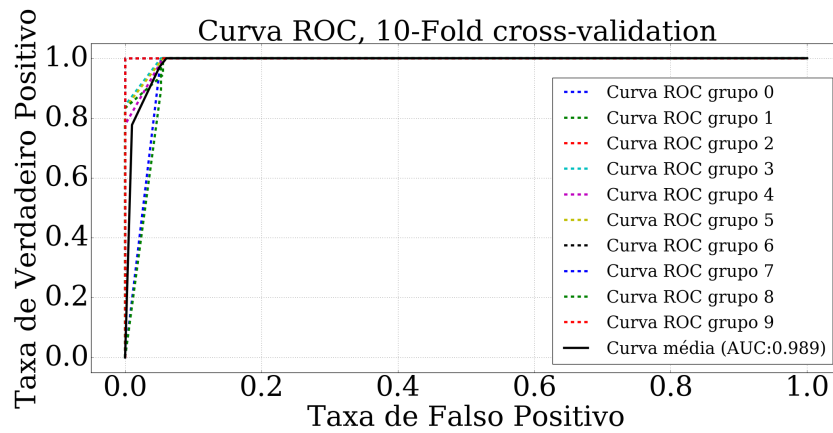


Figura 62 – Curva ROC com inserção de ruído do tipo *speckle* com variância de 0,1

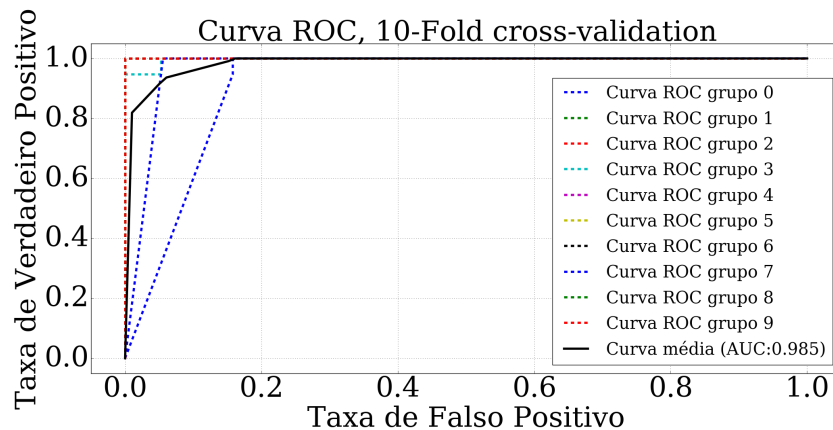


Figura 63 – Curva ROC com inserção de ruído do tipo *speckle* com variância de 0,1 e *salt&pepper* com 20%

diferentes, cerca de 27% do *dataset* total, para possuir um resultado mais representativo. Dessa forma, o tempo médio gasto foi de **64,40 +/- 10,00** mili-segundos (ms) com gasto de memória de **134,26 +/- 0,07** Megabytes (MB). O computador utilizado possui como configuração processador Intel i5 (3ª geração, 2,5 GHz), memória RAM de 8GB com sistema operacional Linux Mint 18.3.

Estes números mostram a viabilidade da utilização durante a navegação de robôs de inspeção de linha de transmissão, no caso deste trabalho especificamente, mas não exclusivamente, o ELIR, além de um consumo de memória relativamente pequeno, não sendo necessário *hardware* com elevada capacidade de processamento, o que torna possível a utilização de computadores mais simples, barateando a implementação da abordagem utilizada nesse trabalho.

Assim o método apresentado neste trabalho apresenta desempenho satisfatório com valor de acurácia máximo de 98,4%, mesmo na presença de ruído. Este valor de desempenho de classificação apresenta melhor resultado do que o apresentado em [PETERS; AHN; BORKOWSKI, 2002]. Esta trabalho também apresenta vantagens em relação aos trabalhos apresentados na literatura pois, a maioria, apesar de citar bons resultados até em condições de aplicação em linha de transmissão real, não apresentam informações sobre métrica de desempenho ou quantidade de amostras analisadas na tarefa de classificação dos obstáculos, apresentando métodos muitas vezes baseados em *threshold* sem a presença de um classificador que possa generalizar a classificação para condições fora da condição planejadas.

Outro ponto a se destacar desse trabalho é a representatividade das *features* e possivelmente maior robustez na classificação utilizando LiDAR quando comparada com a técnica apresentada em [RICHARD; POULIOT; MONTAMBAULT, 2014], uma vez que nesse mesmo artigo é citado que a medição do Hokuyo é significativamente afetada

pela variação de temperatura em ambientes externos, oq altera a medição de largura, parâmetro utilizado para a detecção e classificação dos obstáculos. Contudo, mesmo com a variação do valor da medição, a forma dos obstáculos não se altera, além de, como visto na avaliação com inserção de ruído, o classificador apresentar bom desempenho de classificação mesmo na presença de variações de medidas.

6 Conclusão

Neste trabalho foi desenvolvido um sistema para classificação de obstáculos encontrados por robôs de inspeção de linha de transmissão utilizando um sensor do tipo LiDAR para geração de imagens em escala de cinza dos obstáculos. Inicialmente, introduziu-se a necessidade de inspeção frequente das linhas de transmissão frente a demanda e dependência crescente em relação a energia elétrica. Foram abordados também os métodos adotados atualmente e os perigos e limitações apresentadas pelos mesmos, o que faz com que a utilização de robôs seja uma alternativa cada vez mais viável e necessária.

Posteriormente, apresentou-se a revisão na bibliografia dos robôs de inspeção de linha juntamente com as técnicas de detecção e classificação dos obstáculos. Nas técnicas de classificação apresentadas na literatura, a maioria cita bons desempenhos mas não apresenta métricas de desempenho nem a utilização de algum tipo de classificador para realizar a classificação dos diferentes tipos de obstáculos. Em [POULIOT; RICHARD; MONTAMBAULT, 2012] é apresentada uma extensa caracterização do sensor do tipo LiDAR modelo Hokuyo UTM-30LX, apresentando bons desempenhos para aplicação em robôs de inspeção de linha de transmissão, baseado nos resultados desse estudo e em vantagens como a possibilidade de inspeção simultânea da faixa de servidão e o tratamento simplificado do plano de fundo dos obstáculos optou-se por utilizar esse mesmo sensor neste trabalho. Este sensor foi utilizado para escaneamento dos obstáculos e, através das medidas fornecidas, foram geradas imagens em escala de cinza para extração das *features* clássicas de forma: área e perímetro, e a obtenção da média de intensidade da imagem dos obstáculos. Essas *features* foram utilizadas como dados de entrada do classificador clássico *k-Nearest Neighbors* que, apesar de ser um dos mais antigos e simples algoritmos de classificação, ainda apresenta e apresentou bons resultados na classificação dos obstáculos.

Para levantamento das imagens, o sistema desenvolvido foi testado em ambiente real sob condições internas de laboratório. Para o levantamento do *dataset* foi montada uma plataforma com parte do robô ELIR para a geração de 370 imagens dos obstáculos grampo de suspensão e amortecedor de vibração. Estes obstáculos reais, são comumente encontrados na linha de transmissão e precisam ser ultrapassados pelo robô para a realização de uma inspeção autônoma. Apesar de serem estes os principais obstáculos a serem ultrapassados pelo ELIR, e por isso, selecionados pra classificação nesse trabalho, o procedimento utilizado pode ser facilmente adaptado e estendido para outros obstáculos. O *dataset* foi gerado com diversidade, utilizando variações como diferentes posições no cabo, diferentes posições dos obstáculos, trepidação do cabo simulando oscilação pelo vento e inserção de oclusão com objetos que podem se prender aos obstáculos mudando sua forma.

Os dados levantados nos experimentos foram analisados chegando-se a conclusão de

que as três *features* escolhidas apresentam boa representatividade das classes e distinções, principalmente quando as imagens não apresentavam oclusão, mesmo tendo um bom grau de diversidade. Essa separabilidade se mostrou importante e foi representada no desempenho do classificador que apresentou acurácia de 100% para todas as configurações utilizadas quando se considerou apenas as imagens "sem oclusão". Através da avaliação do comportamento da utilização da normalização das *features* e da variação dos valores de k , chegou-se a conclusão que o melhor desempenho é alcançado quando utiliza-se a normalização, o valor de k utilizado é igual a 5 e os valores de área, perímetro e média de intensidade são adotados para a classificação com o kNN, sendo essa configuração a configuração padrão de investigação neste trabalho. O valor de acurácia de 98,4% foi bastante satisfatório quando considerado o *dataset* "com oclusão" que, através das imagens apresentadas no trabalho, apresentavam boa variação nas imagens que, em algumas situações, possuem uma quantidade significativa de oclusão do obstáculo.

Uma vez que acreditasse que o sistema de classificação esta apto para implementação do robô, analisou-se o tempo de execução da tarefa de classificação e o consumo de memória. Apesar de ser um classificador simples, o sistema de classificação apresentou bons resultados com o tempo médio entre leitura da imagens e classificação de 64,4 milisegundos e consumo de memória de 134,26 Megabytes, comprovando a factibilidade da utilização embarcada no robô e minimizando a necessidade de *hardware* de alto desempenho.

O classificador mostrou-se também robusto para condições onde há presença de ruído nas imagens. Mesmo na presença dos dois tipos de ruído utilizados, quando não havia oclusão, a acurácia permaneceu em 100% na maioria das configurações de classificação e se manteve em 100% na configuração padrão. Já quando eram consideradas as imagens "com oclusão", o desempenho se manteve no mesmo nível com a ausência de ruído, sendo o maior valor de acurácia igual a 98,4%. Mostrando a qualidade e robustez das *features* e da técnica de classificação escolhida nesse trabalho.

Portanto, a técnica e o sistema adotado nesse trabalho apresenta melhor desempenho e um diferencial quanto aos trabalhos apresentados na literatura por apresentar uma metodologia e informações acerca do número de dados analisados bem como uma métrica de desempenho para avaliar a qualidade do classificador, além da própria utilização de um algoritmo de aprendizagem de máquina para classificação, o que possibilita uma maior robustez quanto ao tratamento de dados inteiramente novos, como ilustrado no desempenho na presença de ruído. Outro diferencial desse trabalho é a geração de um *dataset* com imagem gerada dos obstáculos e arquivo CSV com as *features* extraídas que será disponibilizado em repositório aberto para a comunidade, no github do autor ¹.

Apesar dos bons desempenhos, ainda é necessário um estudo mais extenso da aplicação desse sistema de classificação. Portanto, como trabalhos futuros sugere-se a

¹ <https://github.com/pxalcantara>

aplicação e implementação no robô em ambiente externo e real, com condições reais de temperatura, iluminação e exposição ao sol, além do posicionamento dos obstáculos em linha de transmissão real e os efeitos eletromagnéticos causados pelas condições de tensão e corrente da linha. Acredita-se também que a aplicação dos outros tipos de amortecedores e grampos, bem como avaliação com outros tipos de obstáculos pode trazer informações interessantes sobre um desempenho mais abrangente desse classificador.

Referências

ALBIEZ, J. et al. Flatfish-a compact subsea-resident inspection auv. In: IEEE. *OCEANS'15 MTS/IEEE Washington*. [S.l.], 2015. p. 1–8. Cited on page 14.

ANEEL. *Tema para Investimento em P&D*. 2017. Disponível em: <http://www.aneel.gov.br/programa-de-p-d/-/asset_publisher/ahiml6B12kVf/content/temas-para-investimentos-em-p-1/656831?inheritRedirect=false&redirect=http%3A%2F%2Fwww.aneel.gov.br%2Fprograma-de-p-d%3Fp_p_id%3D101_INSTANCE_ahiml6B12kVf%26p_p_lifecycle%3D0%26p_p_state%3Dnormal%26p_p_mode%3Dview%26p_p_col_id%3Dcolumn-2%26p_p_col_pos%3D1%26p_p_col_count%3D3>. Acesso em: 25 Jul. 2017. Cited on page 17.

BAY, S. D. Combining nearest neighbor classifiers through multiple feature subsets. In: CITESEER. *ICML*. [S.l.], 1998. v. 98, p. 37–45. Cited on page 49.

BOYAT, A. K.; JOSHI, B. K. A review paper: noise models in digital image processing. *arXiv preprint arXiv:1505.03489*, 2015. Cited 2 times on pages 60 e 61.

BREIMAN, L. Bagging predictors. *Machine learning*, Springer, v. 24, n. 2, p. 123–140, 1996. Cited on page 49.

CAMPOS, M. F. et al. A mobile manipulator for installation and removal of aircraft warning spheres on aerial power transmission lines. In: IEEE. *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*. [S.l.], 2002. v. 4, p. 3559–3564. Cited 3 times on pages 6, 21 e 22.

CHANG, W. et al. Development of a power line inspection robot with hybrid operation modes. In: IEEE. *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. [S.l.], 2017. p. 973–978. Cited 4 times on pages 6, 30, 31 e 37.

DEBENEST, P. et al. Expliner-robot for inspection of transmission lines. In: IEEE. *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. [S.l.], 2008. p. 3978–3984. Cited 6 times on pages 6, 16, 17, 21, 23 e 33.

ELNEMR, H. A. Feature selection for texture-based plant leaves classification. In: IEEE. *Advanced Control Circuits Systems (ACCS) Systems & 2017 Intl Conf on New Paradigms in Electronics & Information Technology (PEIT), 2017 Intl Conf on*. [S.l.], 2017. p. 91–97. Cited 3 times on pages 47, 67 e 68.

EMPRESA, D. P. E. Balanço energético nacional 2014: ano base 2013. *Rio de Janeiro: EPE*, 2014. Cited on page 14.

FAWCETT, T. An introduction to roc analysis. *Pattern recognition letters*, Elsevier, v. 27, n. 8, p. 861–874, 2006. Cited on page 46.

FOSSENG, S. *Learning Distance Functions in k-Nearest Neighbors*. Dissertação (Mestrado) — Institutt for datateknikk og informasjonsvitenskap, 2013. Cited 3 times on pages 42, 47 e 48.

- FU, S. et al. Structure-constrained obstacles recognition for power transmission line inspection robot. In: IEEE. *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. [S.l.], 2006. p. 3363–3368. Cited 2 times on pages 34 e 35.
- GONCALVES, R. S.; CARVALHO, J. C. M. Graphical simulations of a mobile robot suspended on transmission lines using multibody systems software. In: IEEE. *Robotics Symposium and Intelligent Robotic Meeting (LARS), 2010 Latin American*. [S.l.], 2010. p. 1–6. Cited 3 times on pages 6, 26 e 27.
- GONÇALVES, R. S.; CARVALHO, J. C. M. Review and latest trends in mobile robots used on power transmission lines. *International Journal of Advanced Robotic Systems*, SAGE Publications Sage UK: London, England, v. 10, n. 12, p. 408, 2013. Cited 3 times on pages 6, 27 e 28.
- GROTZINGER, J. P. Analysis of surface materials by the curiosity mars rover. *Science*, American Association for the Advancement of Science, v. 341, n. 6153, p. 1475–1475, 2013. Cited on page 14.
- HAN, S.; LEE, J. Path-selection control of a power line inspection robot using sensor fusion. *Multisensor Fusion and Integration for Intelligent Systems*, Springer, p. 335–348, 2009. Cited on page 34.
- HOKUYO, U. *30-LX, data sheet*. 2013. Cited on page 37.
- II, E. J. L. et al. Robô autônomo para inspeção de linhas de alta tensão. 2014. Cited 3 times on pages 6, 28 e 29.
- KANG, Y. et al. A lidar-based decision-making method for road boundary detection using multiple kalman filters. *IEEE Transactions on Industrial Electronics*, IEEE, v. 59, n. 11, p. 4360–4368, 2012. Cited on page 35.
- KATRASNIK, J.; PERNUS, F.; LIKAR, B. New robot for power line inspection. In: IEEE. *Robotics, Automation and Mechatronics, 2008 IEEE Conference on*. [S.l.], 2008. p. 1195–1200. Cited 4 times on pages 6, 23, 24 e 27.
- KIM, J.-H. et al. License plate detection and recognition algorithm for vehicle black box. In: IEEE. *Automatic Control Conference (CACS), 2017 International*. [S.l.], 2017. p. 1–6. Cited 2 times on pages 7 e 48.
- KOWALSKI, B.; BENDER, C. Pattern recognition. powerful approach to interpreting chemical data. *Journal of the American Chemical Society*, ACS Publications, v. 94, n. 16, p. 5632–5639, 1972. Cited on page 49.
- LEE, J.-K.; JUNG, N.-J.; CHO, B.-H. Development of transmission line sleeve inspection robot. *World Academy of Science, Engineering and Technology*, v. 58, n. 58, p. 157–161, 2011. Cited 3 times on pages 6, 26 e 27.
- LI, Z.; RUAN, Y. Autonomous inspection robot for power transmission lines maintenance while operating on the overhead ground wires. *International Journal of Advanced Robotic Systems*, SAGE Publications Sage UK: London, England, v. 7, n. 4, p. 25, 2010. Cited 2 times on pages 6 e 25.

- MAJNIK, M.; BOSNIĆ, Z. Roc analysis of classifiers in machine learning: A survey. *Intelligent data analysis*, IOS Press, v. 17, n. 3, p. 531–558, 2013. Cited 3 times on pages 7, 46 e 47.
- MANZ, M. et al. Detection and tracking of road networks in rural terrain by fusing vision and lidar. In: IEEE. *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. [S.l.], 2011. p. 4562–4568. Cited on page 35.
- MARKOM, M. et al. Scan matching and knn classification for mobile robot localisation algorithm. In: IEEE. *Robotics and Manufacturing Automation (ROMA), 2017 IEEE 3rd International Symposium in*. [S.l.], 2017. p. 1–6. Cited on page 47.
- MATSUNO, F.; TADOKORO, S. Rescue robots and systems in japan. In: IEEE. *Robotics and Biomimetics, 2004. ROBIO 2004. IEEE International Conference on*. [S.l.], 2004. p. 12–20. Cited on page 14.
- MCDANIEL, M. W. et al. Ground plane identification using lidar in forested environments. In: IEEE. *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. [S.l.], 2010. p. 3831–3836. Cited on page 35.
- MEDINA, J. L. V. et al. *Reliability of classification and prediction in k-nearest neighbours*. Tese (Doutorado) — Universitat Rovira i Virgili, 2013. Cited 5 times on pages 7, 43, 48, 49 e 50.
- MME. MINISTÉRIO DE MINAS E ENERGIA DO BRASIL, *Boletim Mensal de Monitoramento do Sistema Elétrico Brasileiro-Março/2017*. 2017. Cited on page 15.
- MONTAMBAULT, S.; POULIOT, N. Design and validation of a mobile robot for power line inspection and maintenance. In: SPRINGER. *6th International Conference on Field and Service Robotics-FSR 2007*. [S.l.], 2007. v. 42. Cited 3 times on pages 6, 22 e 23.
- MOURÃO, M. et al. Robô de inspeção de linha - d311. 2015. Cited 5 times on pages 6, 29, 30, 31 e 35.
- NASCIMENTO, M. *Detecção e classificação de obstáculos para robô de inspeção de linha de transmissão*. Dissertação (Mestrado), 2018. Cited on page 47.
- NAYYERLOO, M. et al. Cable-climbing robots for power transmission lines inspection. In: *Mobile Robots-State of the Art in Land, Sea, Air, and Collaborative Missions*. [S.l.]: InTech, 2009. Cited on page 15.
- OLIVEIRA, V.; LAGES, W. Comparação entre duas classes de robôs bracejadores subatuados utilizando controle preditivo baseado em modelo, viii simpósio brasileiro de automação inteligente, florianópolis, brasil. 2007. Cited on page 15.
- OPENCV. *About*. 2018. <<https://opencv.org/about.html>>. Accessed: 2018-07-02. Cited on page 58.
- PASCOAL, J.; MARQUES, L.; ALMEIDA, A. T. de. Assessment of laser range finders in risky environments. In: IEEE. *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. [S.l.], 2008. p. 3533–3538. Cited on page 36.
- PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Cited on page 59.

- PETERS, J. F.; AHN, T.-C.; BORKOWSKI, M. Obstacle classification by a line-crawling robot: A rough neurocomputing approach. In: SPRINGER. *International Conference on Rough Sets and Current Trends in Computing*. [S.l.], 2002. p. 594–601. Cited 3 times on pages 18, 33 e 79.
- PHILLIPS, A. et al. Autonomous overhead transmission line inspection robot (ti) development and demonstration. In: IEEE. *Applied Robotics for the Power Industry (CARPI), 2012 2nd International Conference on*. [S.l.], 2012. p. 94–95. Cited 5 times on pages 6, 17, 27, 28 e 33.
- POULIOT, N.; RICHARD, P.-L.; MONTAMBAULT, S. Linescout power line robot: Characterization of a utm-30lx lidar system for obstacle detection. In: IEEE. *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. [S.l.], 2012. p. 4327–4334. Cited 9 times on pages 18, 33, 34, 35, 36, 51, 54, 56 e 81.
- POULIOT, N.; RICHARD, P.-L.; MONTAMBAULT, S. Linescout technology opens the way to robotic inspection and maintenance of high-voltage power lines. *IEEE Power and Energy Technology Systems Journal*, IEEE, v. 2, n. 1, p. 1–11, 2015. Cited on page 36.
- PREMEBIDA, C.; LUDWIG, O.; NUNES, U. Lidar and vision-based pedestrian detection system. *Journal of Field Robotics*, Wiley Online Library, v. 26, n. 9, p. 696–711, 2009. Cited on page 35.
- QUIGLEY, M. et al. Ros: an open-source robot operating system. In: KOBE. *ICRA workshop on open source software*. [S.l.], 2009. v. 3, n. 3.2, p. 5. Cited on page 52.
- RANGEL, R. K.; KIENITZ, K. H.; BRANDÃO, M. P. Sistema de inspeção de linhas de transmissão de energia elétrica utilizando veículos aéreos não-tripulados. In: *3rd CTA-DLR Workshop on Data Analysis & Flight Control ocorrido entre*. [S.l.: s.n.], 2009. p. 14–16. Cited 5 times on pages 6, 14, 15, 24 e 25.
- RASCHKA, S.; MIRJALILI, V. *Python machine learning*. [S.l.]: Packt Publishing Ltd, 2015. Cited 12 times on pages 7, 39, 42, 43, 44, 45, 47, 49, 50, 57, 66 e 68.
- RICHARD, P.-L.; POULIOT, N.; MONTAMBAULT, S. Introduction of a lidar-based obstacle detection system on the linescout power line robot. In: IEEE. *Advanced Intelligent Mechatronics (AIM), 2014 IEEE/ASME International Conference on*. [S.l.], 2014. p. 1734–1740. Cited 7 times on pages 18, 34, 35, 36, 51, 54 e 79.
- RISKY, I. A.; HARSONO, T.; SIGIT, R. Middle finger bone assessment for age identification. In: IEEE. *Knowledge Creation and Intelligent Computing (IES-KCIC), 2017 International Electronics Symposium on*. [S.l.], 2017. p. 288–294. Cited on page 47.
- ROSÁRIO, J. M. *Princípios de mecatrônica*. [S.l.]: Pearson Prentice Hall, 2006. Cited on page 14.
- SARTORI, J. *Robô de inspeção de linhas de transmissão: Análise Geométrica e Cinemática*. Dissertação (Mestrado), 2018. Cited 8 times on pages 6, 14, 15, 22, 24, 25, 26 e 33.
- SENAI - CIMATEC. *Memorial Descritivo - Robô de Inspeção de Linhas Vivas de 138 kV*. 1. ed. Avenida Orlando Gomes, 1845 - Piatã, Salvador, BA, 2013. Cited 2 times on pages 6 e 17.

- SIEGWART, R.; NOURBAKHSI, I. R.; SCARAMUZZA, D. *Introduction to autonomous mobile robots*. [S.l.]: MIT press, 2011. Cited on page 14.
- SOLOMON, C.; BRECKON, T. *Fundamentos de processamento digital de imagens: uma abordagem prática com exemplos em Matlab*. [S.l.]: Grupo Gen-LTC, 2000. Cited 3 times on pages 38, 42 e 58.
- STEENWIJK, M. D. et al. Accurate white matter lesion segmentation by k nearest neighbor classification with tissue type priors (knn-ttps). *NeuroImage: Clinical*, Elsevier, v. 3, p. 462–469, 2013. Cited on page 68.
- SYED, M. E. *Attribute weighting in k-nearest neighbor classification*. Dissertação (Mestrado), 2014. Cited 3 times on pages 48, 49 e 66.
- TOUSSAINT, K.; POULIOT, N.; MONTAMBAULT, S. Transmission line maintenance robots capable of crossing obstacles: State-of-the-art review and challenges ahead. *Journal of Field Robotics*, Wiley Online Library, v. 26, n. 5, p. 477–499, 2009. Cited 5 times on pages 16, 17, 21, 23 e 33.
- VENTRELLA, A. et al. Robô móvel aplicado à inspeção em linhas de transmissão. In: *Anais do II Congresso de Inovação Tecnológica em Engenharia Elétrica*. [S.l.: s.n.], 2003. v. 1. Cited 3 times on pages 6, 17 e 21.
- VERMA, R.; ALI, J. A comparative study of various types of image noise and efficient noise removal techniques. *International Journal of advanced research in computer science and software engineering*, v. 3, n. 10, 2013. Cited 2 times on pages 60 e 61.
- WANG, L. et al. Design, modeling and control of a biped line-walking robot. *International Journal of Advanced Robotic Systems*, SAGE Publications Sage UK: London, England, v. 7, n. 4, p. 33, 2010. Cited 2 times on pages 6 e 26.
- WU, G. et al. A novel self-navigated inspection robot along high-voltage power transmission line and its application. *Intelligent Robotics and Applications*, Springer, p. 1145–1154, 2008. Cited on page 34.
- WU, G. et al. Navigation strategy for local autonomous obstacles-overcoming based on magnetic density detection for inspection robot of single split high voltage transmission line. In: IEEE. *Intelligent Control and Automation (WCICA), 2010 8th World Congress on*. [S.l.], 2010. p. 6555–6561. Cited 3 times on pages 18, 34 e 40.
- YANG, M.; KPALMA, K.; RONSIN, J. *A survey of shape feature extraction techniques*. [S.l.]: In-Tech, 2008. Cited on page 41.
- ZAHED, N.; NAJIB, M. S.; TAJUDDIN, S. N. Categorization of gelam, acacia and tualang honey odor-profile using k-nearest neighbors. Cited on page 68.
- ZHOU, F. et al. Control of an inspection robot for 110kv power transmission lines based on expert system design methods. In: IEEE. *Control Applications, 2005. CCA 2005. Proceedings of 2005 IEEE Conference on*. [S.l.], 2005. p. 1563–1568. Cited 3 times on pages 6, 17 e 22.
- ZUO, Q. et al. The obstacle recognition approach for a power line inspection robot. In: IEEE. *Mechatronics and Automation, 2009. ICMA 2009. International Conference on*. [S.l.], 2009. p. 1757–1761. Cited 2 times on pages 18 e 34.