

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA
CAMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA
CURSO DE ENSINO SUPERIOR DE ENGENHARIA MECATRÔNICA**

VICTÓRIA ZANETTI MARÇAL GALLO

**DESENVOLVIMENTO DE UM SISTEMA DE
RECONHECIMENTO DE CARACTERES PARA CALIBRAÇÃO
DE TRENAS A LASER**

FLORIANÓPOLIS, 2019

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA
CAMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA
CURSO DE ENSINO SUPERIOR DE ENGENHARIA MECATRÔNICA**

VICTÓRIA ZANETTI MARÇAL GALLO

**DESENVOLVIMENTO DE UM SISTEMA DE
RECONHECIMENTO DE CARACTERES PARA CALIBRAÇÃO
DE TRENAS A LASER**

Trabalho de Conclusão de Curso
submetido ao Instituto Federal de
Educação, Ciência e Tecnologia de
Santa Catarina como parte dos
requisitos para obtenção do título de
Bacharel em Engenharia Mecatrônica.

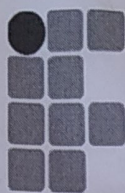
FLORIANÓPOLIS, 2019

Ficha de identificação da obra elaborada pelo autor.

Gallo, Victória
DESENVOLVIMENTO DE UM SISTEMA DE RECONHECIMENTO DE
CARACTERES PARA CALIBRAÇÃO DE TRENAS A LASER / Victória Gallo
; orientação de Cassiano Bonin. - Florianópolis,
SC, 2019. 52 p.

Trabalho de Conclusão de Curso (TCC) - Instituto
Federal de Santa Catarina, Câmpus Florianópolis.
Bacharelado em Engenharia Mecatrônica. Departamento
Acadêmico de Metal Mecânica.
Inclui Referências.

1. Reconhecimento de caracteres. 2. Sistema de visão.
3. Calibração. 4. Processamento de imagens. 5. Opencv.
I. Bonin, Cassiano. II. Instituto Federal de Santa
Catarina. Departamento Acadêmico de Metal Mecânica.
III. Título.



INSTITUTO FEDERAL
SANTA CATARINA

MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS

DECLARAÇÃO DE FINALIZAÇÃO DE TRABALHO DE CURSO

Declaro que a estudante **Victoria Zanetti Marçal Gallo**, matrícula n° **1510051503**, do Curso de Engenharia Mecatrônica, defendeu o trabalho intitulado **Desenvolvimento de um Sistema de Reconhecimento de Caracteres para Otimização do Processo de Calibração de Trenas Laser**, o qual está apto a fazer parte do banco de dados da Biblioteca Hercílio Luz do Instituto Federal de Santa Catarina, Campus Florianópolis.

Florianópolis, 12 de dezembro de 2019.

Prof. Orientador do TCC: Cassiano Bonin

AGRADECIMENTOS

Aos meus pais Jumeri Zanetti e Wagner Marçal Gallo, por me criarem, educarem e me apoiarem. Por terem feito todo possível para que nunca me faltasse nada, o que sou hoje, devo a eles.

Ao professor Cassiano Bonin, por todas as orientações durante o desenvolvimento deste trabalho e por todos os conselhos e ensinamentos durante a graduação.

À minha namorada Júlia Ramos Deloroso, pelos conselhos e por todo incentivo, apoio e companheirismo em todos os momentos, mas principalmente naqueles mais difíceis.

À minha prima Stefani Verônica Zanetti da Cruz e ao meu irmão Arthur Zanetti Marçal Gallo por todos os conselhos e momentos de descontração e alegria.

Aos amigos e colegas de curso especialmente a Luan Florence de Medeiros, Pedro von Hertwig Batista, Rodolfo Schiavi e Luiz Henrique por todo suporte durante a graduação e pelos momentos de distração.

À todos os professores e funcionários do IFSC que, de alguma forma, contribuíram com a minha formação.

À todos os meus colegas e funcionários da CERTI que me acolheram e me ensinaram durante o estágio, especialmente ao meu então supervisor Matheus Dieckman por ter acreditado em mim e por ter me dado a oportunidade de desenvolver este projeto.

RESUMO

Este trabalho apresenta a proposta de um sistema que utiliza técnicas de visão computacional, processamento de imagens e técnicas de reconhecimento de caracteres para otimização de um processo de calibração de trenas a laser. Visando o entendimento do processo a ser otimizado, a primeira parte do trabalho apresenta o procedimento de calibração. O sistema é desenvolvido com a linguagem de programação *Python* utilizando bibliotecas como *OpenCV* na captura, processamento e segmentação das imagens adquiridas do display das trenas a serem calibradas e *Tkinter* para elaboração da interface gráfica. A etapa de reconhecimento dos caracteres constitui-se da utilização dos algoritmos SSOOCR E OCR, porém outras técnicas como algoritmo KNN e *AutoML Vision* são apresentados como possíveis soluções para a detecção e o reconhecimento dos caracteres. Os resultados demonstram o porquê da escolha dos algoritmos SSOOCR e OCR e que, mesmo possuindo limitações, através de correções e restrições conseguem alcançar uma precisão de até 97% no processo de reconhecimento. Além disso, apresentam que o tempo de operação para o procedimento de calibração utilizando o sistema desenvolvido reduziu em mais de 50% comparando ao procedimento feito manualmente.

Palavras-chave: Sistema de visão. Processamento de imagens. Calibração. Reconhecimento de caracteres.

ABSTRACT

This paper presents the proposal of a system that uses computational vision techniques, image processing and character recognition techniques to optimize a laser tape calibration process. For a better understanding the process to be optimized, the first part of the work presents the calibration procedure. The system is developed with the Python programming language using libraries such as OpenCV in capturing, processing and segmenting of images acquired from the display of the trenas to be calibrated and Tkinter to elaborate the graphical interface. The character recognition step consists of not only using the SSOCR and OCR algorithms, but other techniques such as KNN algorithm and AutoML Vision are presented as possible solutions for character detection and recognition. The results demonstrate why the choice of SSOCR and OCR algorithms and that, through corrections and restrictions can achieve an accuracy of up to 97% in the recognition process. The operating time using the developed system was reduce by more than 50% comparing to the manually procedure.

Keywords: Vision system. Image processing. Calibration. Character recognition.

SUMÁRIO

1. INTRODUÇÃO	12
1.1 Justificativa	14
1.2 Objetivo Geral	14
1.3 Objetivos Específicos	14
2. FUNDAMENTAÇÃO TEÓRICA	15
2.1 Trens a laser	15
2.1.1 Funcionamento de trens a laser.....	15
2.1.2 Aplicação das trens	16
2.1.3 Modelos de trens a laser	17
2.1.4 Calibração de trens.....	18
2.2 Sistemas de visão artificial e processamento de imagens	19
2.2.1 Sistema de visão artificial e sua estrutura	19
2.2.2 Aquisição.....	19
2.2.3 Pré-processamento	20
2.2.4 Segmentação.....	21
2.2.5 Extração das características.....	21
2.2.6 Reconhecimento e interpretação.....	22
2.3 Reconhecimento de caracteres	22
2.3.1 Sistema de reconhecimento de caracteres	22
2.3.2 Aplicações	23
2.3.3 Técnicas para reconhecimento de caracteres.....	24
2.4 Python e OpenCV	25
2.4.1 Python	25
2.4.2 OpenCV	26
3. MATERIAIS E MÉTODOS	28
3.1 Processo de calibração de trena a laser	28
3.2 Software	29
3.2.1 Interface Gráfica.....	30
3.2.2 Processamento das imagens	33
3.2.3 Reconhecimento dos caracteres.....	35
4. TESTES E RESULTADOS	38
4.1 Testes experimentais	38
4.1.1 Testes de tempo de captura	38
4.1.2 Teste do reconhecimento dos caracteres	40
4.2 Resultados	42
4.2.1 Viabilidade das técnicas de reconhecimento de caracteres	42
4.2.2 Reconhecimento dos caracteres.....	44

4.2.3 Precisão do sistema.....	45
4.2.4 Tempo de operação.....	48
5. CONCLUSÃO E TRABALHOS FUTUROS	50
Referências	51

LISTA DE FIGURAS

Figura 1- Tempo de voo.....	15
Figura 2- Aplicação trena a laser construção civil.....	16
Figura 3 - Mostradores: (a) Tipo A; (b) Tipo B; (c) Tipo C	18
Figura 4- Principais etapas de um sistema de visão artificial.....	19
Figura 5- Imagem digitalizada do display da trena.....	20
Figura 6- Imagem display da trena pré-processada	20
Figura 7- Imagem segmentada. (a) - Limiarização; (b) - Limiarização inversa; (c) - Filtro de Canny	21
Figura 8 - Extração das características.....	22
Figura 9- Processo de calibração. (a) - Posicionamento da trena a laser no suporte; (b) - Medição dos pontos; (c) - Registro dos valores obtidos	28
Figura 10- Matriz funcional do sistema proposto.....	29
Figura 11- Fluxograma do software desenvolvido	30
Figura 12- Interface: janela de acesso.....	31
Figura 13- Interface: janela de execução	31
Figura 14- Janela de ajuste Webcam	32
Figura 15- Processamento de imagem: etapas I e II	33
Figura 16- Processamento de imagens etapas III e IV.....	34
Figura 17- Região de interesse: ajuste da câmera.....	35
Figura 18 - Imagem a ser reconhecida	35
Figura 19 - Análise do dígito.....	37
Figura 20 – Cenário Teste I.....	39
Figura 21- Cenário Teste I.....	39
Figura 22- Mudança de valor no display.....	40
Figura 23- Cenário Teste II: trena.....	41
Figura 24- Cenário Teste II: Trena Leica.....	41
Figura 25- Curva de aprendizagem.....	43
Figura 26- Gráfico I	46
Figura 27- Gráfico II.....	47
Figura 28- Gráfico III.....	47

LISTA DE TABELAS

Tabela 1- Tabela comparativa dos modelos de trenas a laser selecionados.....	18
Tabela 2- Análise do reconhecimento dos caracteres	36
Tabela 3- Registro do Teste 1.....	40
Tabela 4- Resultado Teste II : Trena 1.....	44
Tabela 5- Resultado Teste II : Trena 2.....	45
Tabela 6- Duração do procedimento manual.....	48
Tabela 7- Duração do procedimento otimizado	48

1. INTRODUÇÃO

Trenas a laser são instrumentos desenvolvidos para medições à distância. Por serem leves, compactas e ainda apresentarem agilidade, precisão e praticidade, estão ganhando mais espaço no mercado e na indústria (FG, 2018). Disponíveis em diversos modelos, possibilitam medições para diferentes tipos de ambientes. Trenas a laser podem ser utilizadas para o cálculo de área, volume, altura e ângulos, além da medição linear. Isso faz com que sejam utilizadas por profissionais de diversas áreas, como, pintores, marceneiros, agrimensores, encanadores, montadores, domésticos, automobilísticos, mecânicos (CAMILO, 2018). Trenas a laser são equipamentos que necessitam de calibrações e este processo ainda é feito de forma manual.

Os processos de calibração de instrumentos são de suma importância para a indústria e para o seu sistema de qualidade (FARUOLO, 2015). Segundo INMETRO (2012, p. 27) um processo de calibração pode ser definido como uma “operação que estabelece, sob condições especificadas, numa primeira etapa, uma relação entre os valores e as incertezas de medição fornecidos por padrões e as indicações correspondentes com as incertezas associadas;”. Apesar de todo avanço tecnológico, é difícil encontrar sistemas automáticos de calibração de instrumentos ou afins.

Segundo Gonçalves e Centeno (2017, p. 61) “o crescimento avançado da tecnologia digital, em conjunto com o desenvolvimento de novos algoritmos, tem permitido um número cada vez maior de aplicações que envolvem processamento digital de imagens (PDI)”. Essa nova era digital vem evoluindo rapidamente, fazendo com que novas ferramentas surjam para tornar os processos produtivos mais ágeis, simples e eficientes (GALLON, 2014). Visão computacional e aprendizado de máquina (*machine learning*) podem ser citados como exemplos desses grandes avanços, os quais desempenham um papel relevante para história da tecnologia, da automação industrial (SANTOS, 2006)e, ainda, dos métodos de calibração e instrumentação.

Embora existam muitos equipamentos com diferentes tipos de tecnologia para compartilhamento de dados, os sistemas de visão computacional e de processamento de imagem são soluções para coletar dados que são feitos de forma manual (GONÇALVES, Julio Cesar e CENTENO, 2017). Esses sistemas, possuem um campo de aplicações, sendo utilizados para reconhecimento de padrões, de caracteres, monitoramento, inspeções, entre outros (MARENGONI e STRINGHINI, 2009). Desta forma, é capaz de auxiliar em diversas áreas, como a de calibração de padrões e instrumentos.

A partir do que foi apresentado, o presente trabalho apresenta o desenvolvimento de um sistema de visão computacional para reconhecer caracteres numéricos, com o propósito de otimizar o processo de calibração de trenas a laser. Está dividido em 5 capítulos, sendo o primeiro a introdução, seguido da fundamentação teórica, materiais e métodos utilizados durante o desenvolvimento, testes e resultados e por fim o capítulo de conclusão e trabalhos futuros.

Este trabalho é um Trabalho de Conclusão de Curso como parte dos requisitos para obtenção do título de Bacharel em Engenharia Mecatrônica.

1.1 Justificativa

Por conta da dificuldade encontrada em relação a diversidade dos modelos de trenas a laser, possuir um sistema de calibração para um modelo específico de trena acaba não sendo vantajoso para uma empresa. Embora existam inúmeros modelos de trenas, com diversas tecnologias, como *bluetooth* e *Wi-Fi*, a sua calibração ainda é feita manualmente.

Considerando que os métodos utilizados para calibração de trena a laser são executados manualmente e com dois operadores, sendo um responsável pela leitura e registro da medição da trena a laser e o outro pelo deslocamento do alvo a ser medido, a relevância deste trabalho está em automatizar este processo de leitura e registro, reduzindo custos e tempo em um processo de calibração e certificação de trenas a laser.

1.2 Objetivo Geral

Elaborar um sistema de reconhecimento de caracteres numéricos utilizando visão computacional e técnicas de reconhecimento de caracteres, para auxiliar o processo de calibração de trenas a laser.

1.3 Objetivos Específicos

Para atingir o objetivo geral foram traçados os seguintes objetivos específicos:

- a) desenvolver um sistema de visão computacional que faça a aquisição e o tratamento da imagem;
- b) desenvolver uma interface gráfica que permita orientar o procedimento de calibração;
- c) selecionar uma técnica para o reconhecimento dos caracteres numéricos;
- d) validar o sistema desenvolvido através de testes;

- e) comparar e analisar o processo manual com o automatizado, avaliando velocidade e qualidade.

2. FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão apresentadas as fundamentações teóricas utilizadas para o desenvolvimento deste trabalho.

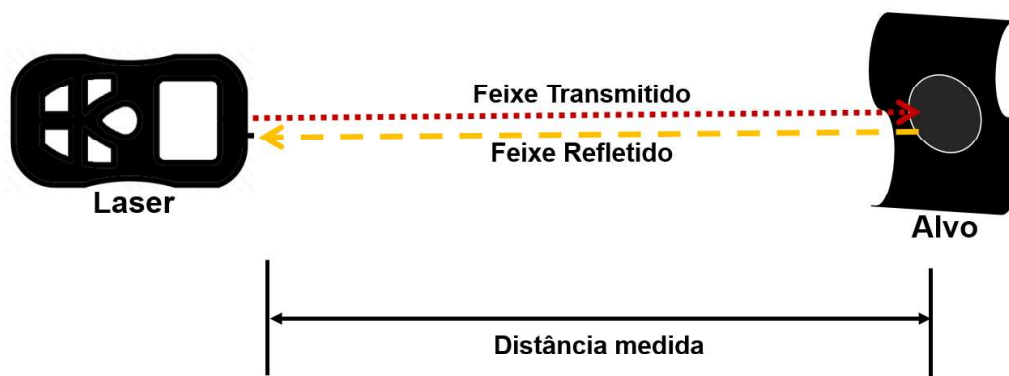
2.1 Trens a laser

2.1.1 Funcionamento de trenas a laser

RESENDE e colab., 2017 define que o “Sistema foi batizado por Gold de LASER, abreviatura para expressão inglesa *light amplification by simulated emission of radiation* (ou em português, amplificação de luz por emissão estimulada)”. Sistemas de medição a laser utilizam o princípio do “tempo de voo” para determinar a distância da qual se quer medir.

BRANCO e colab., (2016, p. 100) apresenta que o princípio de tempo de voo (Figura 1) ocorre quando “um curto e intenso pulso de radiação é emitido pelo transmissor ao alvo, que reflete um sinal paralelo até o receptor. Assim, a distância é calculada pelo tempo decorrido entre a emissão e a recepção do sinal, o qual é multiplicado pela velocidade da luz.”.

Figura 1- Tempo de voo



Fonte: autor

Em uma trena a laser é possível encontrar diferentes funções de medição, como medição de comprimento, área, volume, ângulo, entre outros. Segundo ANDREW e colab.(2015) apesar de possuírem funções robustas, a utilização desses dispositivos, e de suas respectivas funções, é relativamente simples. Considerando que o operador necessita apenas ligar o aparelho e o laser, posicioná-lo, direcioná-lo até o seu alvo e, por fim, capturar os dados indicados no mostrador da trena.

2.1.2 Aplicação das trenas

Trenas a laser podem ser aplicadas de diferentes maneiras e, por possuírem muitas funções, estes dispositivos podem ser utilizados em diversos ambientes. Na indústria, as trenas podem ser utilizadas no controle de qualidade de produtos e dos equipamentos através da medição linear. Além disso, com as funções de cálculo de área, volume e nível, é possível utilizá-las no chão de fábrica (SCHOENKNECHT, 2010).

São muito utilizadas na área da construção civil, pois possibilitam uma maior precisão na medição linear, superficial e volumétrica e por possibilitar e facilitar a medição em espaços com difícil acesso. São consideradas mais ágeis por necessitar somente um único operador e por não precisar ser esticada, diferentemente da trena de fita tradicional (INTRUSUL, 2019). Na Figura 2, é possível visualizar a aplicação da trena a laser na construção civil.

Figura 2- Aplicação trena a laser construção civil



Fonte: <https://hardwarespecialist.com.sg/product/bosch-professional-glm-250-vf/>

Pela variedade de funcionalidade já citada, equipamentos como trenas a laser são utilizados, também, por profissionais como: pintores, marceneiros, agrimensores, encanadores, montadores, domésticos, automobilísticos, mecânicos, entre outros (CAMILO, 2018).

Apesar de possuir um campo de aplicação muito grande, a finalidade da utilização de uma trena a laser aponta para dimensionamento e medição de superfícies ou objetos, o que torna de grande relevância que estes equipamentos sejam precisos e devidamente calibrados.

2.1.3 Modelos de trenas a laser

Existe uma diversidade de modelos de trenas a laser no mercado, que podem variar desde suas funcionalidades até as suas marcas fabricantes. Para melhor entendimento, esta seção apresenta alguns modelos de trena a laser e as compara, tendo como parâmetros: as funcionalidades, faixa de medição, precisão de medição, tempo de medição e tipo mostrador.

Na Tabela 1 é possível visualizar os dados técnicos dos modelos de trena a laser selecionados. Todas as informações foram retiradas dos respectivos manuais de operação, anexados neste trabalho (Anexo I ao VII).

Para melhor organização e visualização da tabela, foram utilizadas as seguintes letras para indicar a característica do mostrador, é possível visualizar os mostradores na Figura 3:

- Mostrador do tipo **A** (Figura 3. a): Mostrador de cristal líquido com tela de fundo escuro e caracteres numéricos convencionais.
- Mostrador do tipo **B** (Figura 3. b): Mostrador de cristal líquido com tela de fundo claro e caracteres numéricos convencionais.

Mostrador do tipo **C** (Figura 3. c): Mostrador de cristal líquido com caracteres numéricos de 7 segmentos.

Figura 3 - Mostradores: (a) Tipo A; (b) Tipo B; (c) Tipo C



Fonte: autor

Tabela 1- Tabela comparativa dos modelos de trenas a laser selecionados

Modelo	Faixa de medição	Precisão	Tempo medição	Mostrador
GLM80	0,05-80m	±1,5mm	4s	B
DISTOD8	0,05-200m	±1,0mm	2s	A
SW-T100	0,05-100m	±2,0mm	-	C
TLM99	1-30m	±2,0mm	4s	C
LD030P	0,2-30m	±3,0mm	6s	C
GLM250	0,05-250m	±1,0mm	4s	B
DISTOD510	0,05-200m	±1,0mm	2s	B

Fonte: autor

2.1.4 Calibração de trenas

Não existe um procedimento de calibração padrão para trenas a laser, isso porque não há uma norma para este, o que faz com que cada empresa de calibração e instrumentação tenha o seu procedimento. Um dos processos, do qual foi utilizado como base nessa pesquisa, é feito de maneira manual. O procedimento consiste, basicamente, em alinhar a trena a laser com o alvo e para cada ponto, anotar e comparar os valores obtidos na trena a laser e os valores reais dos alvos.

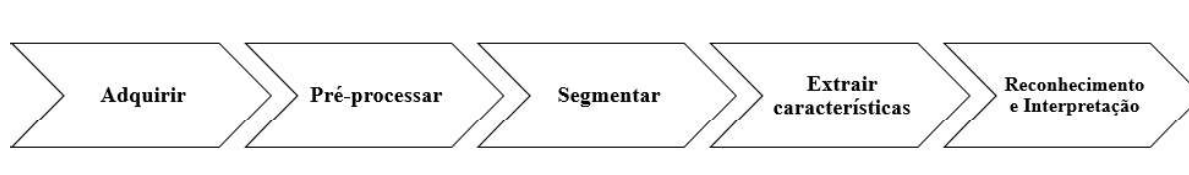
2.2 Sistemas de visão artificial e processamento de imagens

2.2.1 Sistema de visão artificial e sua estrutura

FILHO e NETO (1999) definem um Sistema de visão artificial como sendo “um sistema computadorizado capaz de adquirir, processar e interpretar imagens correspondentes a cenas reais.”. Determinam também, a estrutura de um sistema de visão, separando em 5 etapas: aquisição, pré-processamento, segmentação, extração de características e reconhecimento e interpretação. Cada etapa tem papel fundamental no processamento de imagens e para melhor entendimento, serão explicadas a partir de um problema prático, a leitura e reconhecimento de dados de uma trena a laser digital.

Partindo do que já foi dito, a Figura 4 representa a estrutura funcional do processamento de imagem de um sistema de visão artificial.

Figura 4- Principais etapas de um sistema de visão artificial



Fonte: Adaptado de Filho e Neto (1999)

2.2.2 Aquisição

A primeira etapa consiste na Aquisição, definida por Filho e Neto (1999) como “o processo de conversão de uma cena real tridimensional em uma imagem analógica, ou seja, transdução optoeletrônica.”. Que para o exemplo prático, equivaleria a uma imagem digitalizada do display da trena a laser (Figura 5).

Para esta etapa, alguns aspectos precisam ser estabelecidos como: escolha do conversor (câmera fotográfica ou de vídeo que irá capturar a imagem), a velocidade de aquisição, resolução, dimensão e quantização dos níveis de cinza. A imagem monocromática, ou em escalas de cinza, se apresenta em um formato adequado para ser processada, pois os níveis de cinza podem ser tratados computacionalmente como matrizes e, assim, facilitar o processamento da imagem. O objetivo de definir estes pontos é apresentar, na imagem digitalizada, melhor qualidade do que imagem original apresenta (FILHO e NETO, 1999).

Figura 5- Imagem digitalizada do display da trena



Fonte: autor

2.2.3 Pré-processamento

Esta etapa consiste na correção e aprimoramento da qualidade da imagem digitalizada, reduzindo e filtrando ruídos, alterando os níveis de contraste e brilho, corrigindo distorções geométricas entre outros (ACHARYA e AJOY, 2005; FILHO e NETO, 1999) .

A correção é feita diretamente com os valores de intensidade dos pixels. No exemplo prático, isso corresponderia a alterações na imagem sem o conhecimento prévio sobre os caracteres presentes no display. Após as operações feitas, espera-se que o resultado seja uma imagem com qualidade maior que a original (FILHO e NETO, 1999).

A Figura 6 apresenta a imagem do display da trena corrigida, com redução de brilho e com aplicação do filtro de escalas de cinza para reduzir os ruídos.

Figura 6- Imagem display da trena pré-processada



Fonte: autor

2.2.4 Segmentação

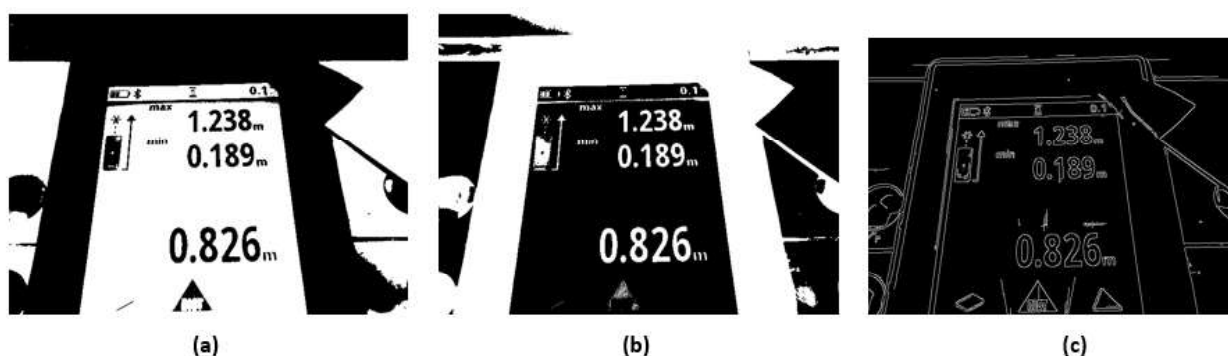
A segmentação pode ser definida como a principal etapa do processo de reconhecimento ou extração de características de uma imagem. Compreendida por ser o processo de subdivisão da imagem em regiões de interesse e não interesse (FILHO e NETO, 1999).

Há diversas técnicas de segmentação e a partir das características que se deseja extrair, da qualidade da imagem e de análises aritméticas e morfológicas, é possível escolher a que melhor se encaixa para o processo.

No caso das trenas, equivaleria ao processo de separar o objeto a ser reconhecido, os números correspondentes a medição, do fundo (*background*) da imagem, que neste caso seria o restante do display.

A Figura 7 apresenta a imagem pré-processada, do display da trena, segmentada através da aplicação de alguns filtros.

Figura 7- Imagem segmentada. (a) - Limiarização; (b) - Limiarização inversa; (c) - Filtro de Canny



Fonte: autor

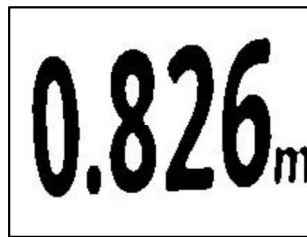
2.2.5 Extração das características

Após o processo de segmentação é realizada a extração das características que consiste na transformação dos dados de entrada em um conjunto de características, através da identificação e na classificação dos objetos que futuramente serão reconhecidos (ANTONELLO, 2017; FACON, 1993).

Filho e Neto (1999, p.10) afirmam que nesta etapa “a entrada ainda é uma imagem, mas a saída é um conjunto de dados correspondentes àquela imagem.”. Esta estrutura de dados pode corresponder a uma matriz, um vetor ou apenas uma coordenada (FACON, 1993).

Ao aplicar o processo explicado no exemplo prático da trena, esta etapa significaria identificar na imagem a região de interesse e os objetos, os caracteres numéricos que posteriormente seriam reconhecidos. A Figura 8 - **Extração das características** apresenta a extração das características a serem reconhecidas da imagem do display da trena a laser.

Figura 8 - Extração das características



Fonte: autor – Trena LEICA DISTO 8

2.2.6 Reconhecimento e interpretação

A etapa de reconhecimento constitui-se da classificação dos objetos extraídos na etapa anterior. Pensando no exemplo das trenas, é nesta etapa que os caracteres numéricos extraídos do display da trena serão identificados. Ou seja, analisando a imagem extraída, o sistema a reconheceria como o número 0.826.

Esta classificação é feita a partir de uma base de dados e de técnicas de reconhecimento, como por exemplo: OCR (*Optical Character Recognition* – Reconhecimento óptico de caracteres), SSOOCR (*Seven Segment Optical Character Recognition* – Reconhecimento óptico de caracteres de sete segmentos), Redes Neurais, K-NN etc. (FILHO e NETO, 1999; GONÇALVES, Julio Cesar e CENTENO, 2017; GONZALEZ, 2002).

A interpretação consiste na análise e validação dos dados obtidos no reconhecimento da imagem (FILHO e NETO, 1999). Após o reconhecimento é necessário interpretá-lo a fim de definir se o resultado é válido ou não. No exemplo, é nesta etapa que verificamos e interpretamos se os valores estão de acordo com a medição da trena.

2.3 Reconhecimento de caracteres

2.3.1 Sistema de reconhecimento de caracteres

Reconhecer um caractere pode ser definido como o processo de segmentação e a classificação de uma imagem (FILHO, 1999). Reconhecer caracteres é uma área de

pesquisa de Visão computacional que segundo SILVA, (2005, p.1) “surgiu a partir da necessidade de auxiliar pessoas portadoras de deficiência visual e a primeira experiência bem sucedida foi realizada pelo cientista russo Tyurin em 1.900.”.

A partir de 1967 as grandes empresas começaram a produzir e comercializar tecnologias de Reconhecimento de caracteres, os chamados de OCR. Com isso, a utilização desses sistemas tornou-se mais comum juntamente com os sistemas de visão, otimizando processos como leitura automática de textos ou de grandes volumes de dados, processamento de linguagens, em máquinas tradutoras, aqueles que envolvem comunicação homem-máquina entre outros processos industriais e comerciais (PATEL e colab., 2012).

2.3.2 Aplicações

Com o avanço das pesquisas e com a evolução da tecnologia, as aplicações de sistemas OCR estão cada vez mais presente no cotidiano e não só em processos industriais (FILHO, 1999). Dentre essas aplicações é possível citar a utilização desses sistemas em processos como leitura de códigos de barra e código em 2D, de placas de carro para controle de tráfego e identificação de veículos, de medidores de consumo de água ou gás, escaneamento de imagens e conversão de arquivos de texto (ALMEIDA, 2018; FILHO, 1999; GONÇALVES, Julio Cesar e CENTENO, 2017; SILVA, 2005).

Os sistemas OCR são capazes de otimizar e automatizar muitos processos que antes eram ditos manuais. Na indústria, por exemplo, podem ser utilizados para inspeção e controle de mercadorias, através da leitura dos códigos de identificação dos produtos, tornando esses processos mais práticos e velozes (GOMES CAROLINE, 2018). A relevância da utilização destes na indústria vai além da substituição homem-máquina, tendo em vista que além da otimização dos processos ainda há a diminuição de falhas que eventualmente podem acontecer durante as leitura ou preenchimentos de planilhas de controle, por exemplo.

Como já mencionado, sistemas OCR não são utilizados somente em indústrias. O LPR (“*License Plate Recognition*” ou traduzido Reconhecimento de Placas de Veículos) é um exemplo de uma aplicação não industrial, utilizado para fiscalização de trânsito, monitoramento do tráfego, pedágios, controle de acesso empresarial e residencial, entre outros (BERNARDI, 2016) . Com o avanço da tecnologia e com o crescimento de estudos na área de visão computacional, criou-se um campo a partir dos sistemas LPR, o ALPR

(do inglês *Automatic License Plate Recognition* ou traduzido Reconhecimento automático de Placas de Veículos (GONÇALVES, Gabriel Resende, 2016).

O sistema de reconhecimento automático de placas de carro (ALPR) é composto por múltiplos processos, seguindo desde a captura e segmentação das imagens até a identificação e reconhecimento da placa do veículo. Existem diversos *softwares* e técnicas para desenvolver sistemas ALPR, visando o crescimento dessa tecnologia, foi desenvolvido pela OpenALPR Technology, Inc, um software chamado *OpenALPR* (CYRUS, 2015; “OpenALPR Documentation — openalpr 2.6.101 documentation”, 2017). O *OpenALPR* é uma biblioteca *Open Source* de reconhecimento de números de placas veiculares escrita em C++ que utiliza as bibliotecas *OpenCV* e *Tesseract*, se tornando disponível para *download* no fim de 2015 (CARDINAL, 2016; “OpenALPR Documentation — openalpr 2.6.101 documentation”, 2017).

2.3.3 Técnicas para reconhecimento de caracteres

Existem diversas técnicas que podem ser utilizadas para a detecção de padrões e reconhecimento de objetos. Dentre essas técnicas e ferramentas, as mais utilizadas por programadores são K-nn, Redes neurais artificiais e *Extreme Learning Machine*. Para aquele que necessitam de uma solução mais prática para o reconhecimento de caracteres e que não envolve tanto conhecimento na área da programação, é a ferramenta do *Google Cloud* o *AutoML*. Além dessas técnicas e do OCR, existe ainda o algoritmo SSOCR (do inglês *Seven segments Optical Character Recognition*), baseado em *OpenCV*, uma biblioteca que permite a implementação das técnicas citadas, tem como objetivo reconhecer dígitos de 7 segmentos (BARRETO, 2002; GUINGO e colab., [S.d.]; SILVA, 2005).

Segundo o site do *Google Cloud*, o *AutoML Vision* é uma ferramenta de aprendizado de máquina do *Google* que pode ser utilizada para desenvolver diversas aplicações sem ter um conhecimento avançado sobre essa tecnologia. Dentre as ferramentas oferecidas pela *Google*, o *AutoML Vision* utiliza as tecnologias de aprendizado de máquina supervisionado para reconhecer padrões e conteúdos em imagens que foram rotulados pelo usuário.

De maneira prática este instrumento proporciona ao usuário o contato direto com o processo de reconhecimento de padrões e objetos em imagens. E através de tutorias e exemplos, o *AutoML Vision* possibilita que, com poucas linhas de código e pouco

conhecimento na área de programação e aprendizado de máquina, o resultado seja satisfatório.

A escolha da técnica se dará a partir do problema a ser resolvido, dos tipos de caracteres a serem reconhecidos, o ambiente, os requisitos do projeto e as tecnologias disponíveis.

2.4 Python e OpenCV

2.4.1 Python

O autor Borges (p.13, 2010) define “*Python*” como “uma linguagem de altíssimo nível (em inglês *Very High Level Language*) orientada a objetos, interpretada e interativa.”. *Python* incluiu diversas estruturas de alto nível, ou seja, listas, dicionários, data, hora, complexos, tuplas, além de possuir uma quantidade grande de módulos e *frameworks* que podem ser adicionados e utilizados (BORGES, 2010; CORREA, 2017).

O *Python* foi criado por Guido Van Rossum, em 1989. É uma linguagem interpretada e possui uma sintaxe mais clara e dinâmica, ao compará-lo com outras linguagens, tornando uma das linguagens mais produtiva. Além disso, é uma linguagem livre (do inglês *open source*) e é possível baixar e instalar gratuitamente o interpretador *Python*. Por possuir licença compatível com *General Public License (GPL)*, é utilizado em muitos produtos proprietários, sendo empregado por empresas como: Google e Yahoo, Microsoft, Nokia e Disney.

A linguagem *Python* possui uma “tipagem” dinâmica, ou seja, as variáveis não precisam ser declaradas, apenas atribuídas, o interpretador define o tipo da variável a partir daquilo que for atribuído a ela. O interpretador *Python* realiza a análise do código, o converte para símbolos, armazena o *bytecode*, um formato binário com instruções para o interpretador, e por fim, compila o código. Além disso, pode ser utilizado de forma interativa, podendo ser evocado em *prompt* de comando, testar e modificar partes de código antes da inseri-las nos programas, analisar objetos, funções, módulos ou até fazer extração ou conversão de dados (BORGES, 2010).

Mesmo sendo uma linguagem interativa, há diversas ferramentas para desenvolver códigos, como as IDE (da tradução em inglês “Ambiente Integral de Desenvolvimento”), editores e as *shells*. As IDEs ajudam o desenvolvedor otimizando e aumentando a produtividade com recursos como por exemplo, completar ou sugerir palavras e funções, coloração no código fonte que auxiliam na verificação da sintaxe, entre outras funções.

Os editores e as *shells* possuem funcionalidades que permitem a exportação, conversão de textos e a execução de pequenas partes do código, respectivamente (BORGES, 2010).

A linguagem *Python* é muito conhecida pela variedade de *frameworks* que possui. Um *framework* pode ser considerado coleções de funções, uma estrutura base que possui ferramentas, guias e componentes que auxiliam e otimizam o processo de desenvolvimento do *software*. A instalação dos módulos e *frameworks* é consideravelmente simples e totalmente gratuitas (ASSIS e MACIEL, 2008; BORGES, 2010; MATTSSON, 2000).

Além dos *frameworks*, o *Python* possui diversas bibliotecas que auxiliam em diferentes aplicações e atividades e que permitem a construção de projetos mais complexos e comerciais. Como por exemplo a biblioteca *Tkinter*, que vem acompanhada na instalação padrão da linguagem e que permite, de maneira prática, o desenvolvimento de interfaces gráficas. A biblioteca *Matplotlib* e *numpy*, que auxiliam na visualização de dados através de gráficos e na manipulação de dados com operações matemáticas, respectivamente. A *PIL*, uma biblioteca que trabalha com imagens, facilitando o processo de manipulação de imagens (BORGES, 2010).

2.4.2 OpenCV

“O *OpenCV* (*open source Computer Vision*) é uma biblioteca de programação de código aberto, desenvolvido inicialmente pela Intel Corporation” (MARENGONI e STRINGHINI, 2009, p. 126). É uma biblioteca de visão computacional acessível e que pode ser utilizada para interação humana-computador e na área da robótica. O *OpenCV* possui mais de 500 funções para auxiliar o programador. Apesar de ter sido desenvolvida em C++ o *OpenCV* susporta várias linguagens, inclusive *Python*, o *OpenCV-Python*, a API de *Python* para *OpenCV*, que combina as melhores qualidades do *OpenCV* C++ e a linguagem *Python* (ANTONELLO, 2017; GALLON, 2014).

O *OpenCV* é dividido em cinco grupos principais de funções, cada um responsável por uma área específica para manipulação de imagem, são eles: processamento de imagens, análise estrutural, análise de movimento e rastreamento de objetos, reconhecimento de padrões e calibração de câmera e reconstrução 3D (ANTONELLO, 2017; MARENGONI e STRINGHINI, 2009).

Os grupos de funções são utilizados para específicas aplicações, o que torna a utilização da biblioteca simples e prática. O grupo de processamento de imagens, o mais

utilizado durante todo o desenvolvimento deste trabalho, torna capaz a realização do "pré-processamento", com funções que manipulam imagens, diminuindo ruídos, aplicando filtros, convertendo, formatando tamanhos e formas para posteriormente serem analisadas.

Mesmo com o surgimento de outras técnicas e bibliotecas para análise e reconhecimento de padrões, análises de movimento e rastreamento de objetos, o *OpenCV* continua sendo a solução mais adequada para o pré-processamento e manipulação de imagens e, por conta disso, a comunidade mantém a biblioteca atualizada. Estas informações podem ser comprovadas através do repositório do GitHub do *OpenCV*.

3. MATERIAIS E MÉTODOS

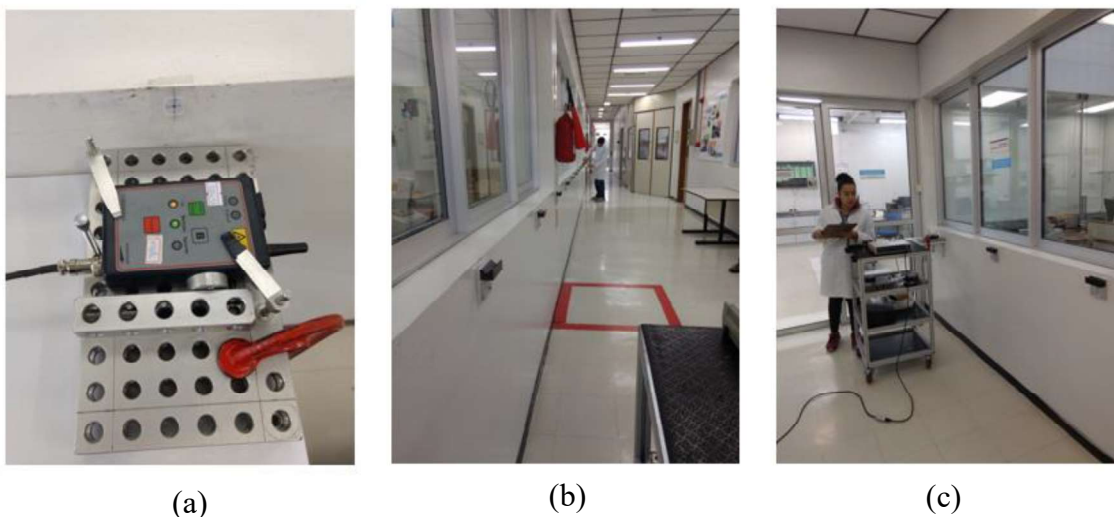
Esta seção apresentará o desenvolvimento da proposta do sistema, indicando, também, os materiais e os métodos de pesquisa utilizados durante todo o processo.

3.1 Processo de calibração de trena a laser

O processo de calibração de trena a laser a ser otimizado consiste em posicionar a trena no suporte, alinhar o laser apontando para o primeiro e para último alvo, apontar o laser para o alvo posicionado no ponto de medição, anotar os valores obtidos durante a medição e, por fim, cadastrá-los no software gerador de certificados.

A Figura 9 a seguir demonstram o processo de calibração descrito, de maneira ilustrativa.

Figura 9- Processo de calibração. (a) - Posicionamento da trena a laser no suporte; (b) - Medição dos pontos; (c) - Registro dos valores obtidos



Fonte: autor

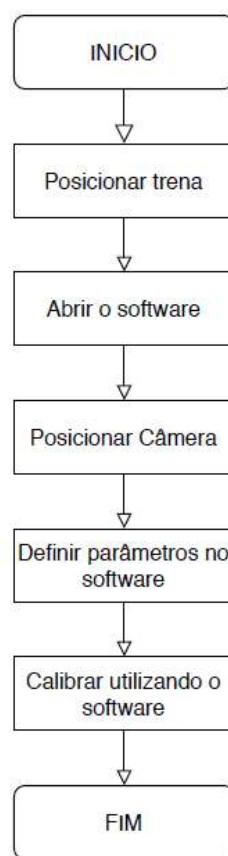
O procedimento descrito é feito por dois operadores, sendo um responsável pelo deslocamento do alvo para cada ponto de medição e o outro por observar e anotar os valores obtidos, através do display da trena, na planilha. Com a implementação do sistema proposto neste trabalho, o processo de calibração de trenas a laser poderá ser feito por apenas um operador, pois a etapa de visualização e registro dos valores obtidos na medição na planilha, que irá adquirir a imagem do display, reconhecerá os caracteres numéricos e gerar uma planilha de extensão xlsx.

O sistema proposto é composto por uma webcam, um notebook. A webcam é o dispositivo responsável pela aquisição das imagens dos displays das trenas que serão calibradas, através de um cabo USB as imagens são digitalizadas e, por fim, o algoritmo

as adquirir, detecta os caracteres e o reconhece através do algoritmo OCR e SSOCR. Para que os números possam ser lidos e reconhecidos corretamente, é necessário que as imagens passem por alguns processamentos digitais, como correção de imagem, diminuição de ruídos aplicação de filtros, limiarização, entre outros. Isso porque a imagem real (sem filtros e correções) poderia apresentar falhas ou ruídos, o que dificultaria o reconhecimento dos objetos presentes na imagem.

A Figura 10 representa a sequência de atividades durante a calibração utilizando o sistema de otimização proposto.

Figura 10- Matriz funcional do sistema proposto



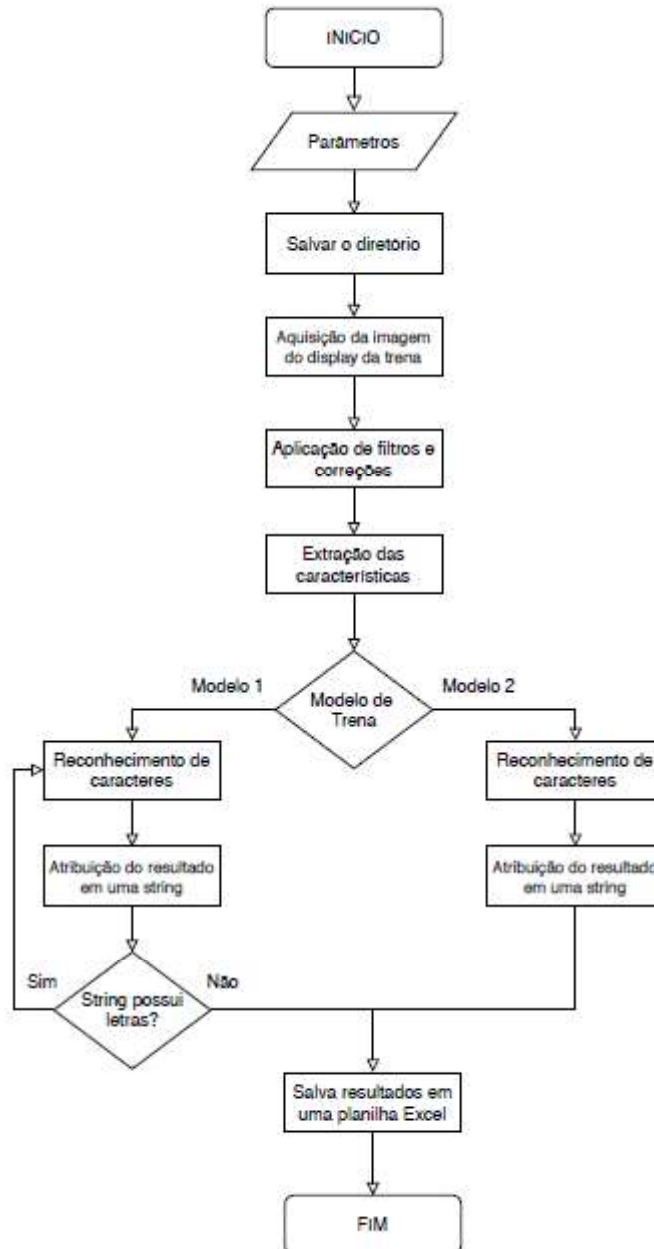
Fonte: autor

3.2 Software

O *software* foi desenvolvido com a linguagem *Python*, versão 3.7, com a IDE do *pycharm*. Esta seção está separada de acordo com as divisões funcionais do *software*, para melhor compreensão.

O fluxograma apresentado na Figura 11 representa as funções e as atividades que o software desenvolve, de forma lógica.

Figura 11- Fluxograma do software desenvolvido



3.2.1 Interface Gráfica

A interface gráfica tem como principal objetivo fazer a conexão entre o operador, quem está executando a calibração da trena a laser, e o reconhecimento dos caracteres. A interface do sistema foi desenvolvida com a linguagem *Python*, com o auxílio da biblioteca *Tkinter*.

As figuras a seguir demonstram a versão de teste da interface do sistema.

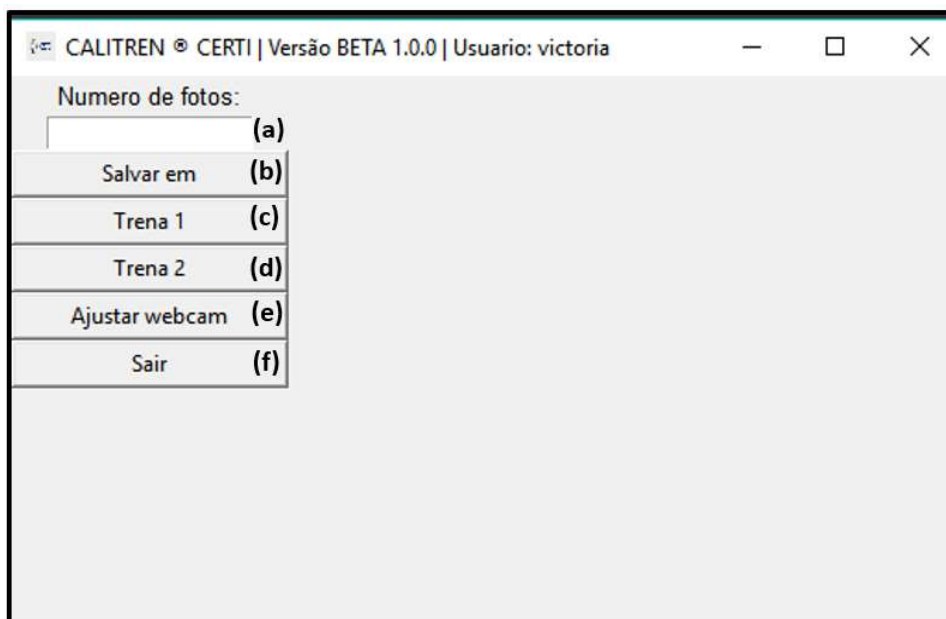
Figura 12- Interface: janela de acesso



Fonte: autor

A Figura 12 apresenta a primeira parte da interface, a janela de acesso ao *software*. O controle de acesso é feito através do nome do usuário e de uma senha padrão.

Figura 13- Interface: janela de execução



Fonte: autor

A segunda janela, denominada de “janela de execução” é composta por uma caixa de entrada (Figura 13– a), onde o usuário deverá determinar a quantidade de fotos deverão ser tiradas. Este parâmetro deverá ser definido pelo operador, devido a diversidade de pontos a serem calibrados, dependendo da trena.

Os botões possuem a função de:

- Figura 13. b – Escolher o diretório para salvar as imagens e a planilha de resultados;
- Figura 13. C e Figura 13. D – Escolher o tipo de trena a ser analisada;
- Figura 13. E – Ajustar a Webcam;
- Figura 13. F – Sair do sistema.

O botão “Ajustar a Webcam” Figura 14 , abre uma janela de vídeo para que o usuário possa ajustar a webcam de tal forma que os dados sejam capturados adequadamente.

Figura 14- Janela de ajuste Webcam



Fonte: autor

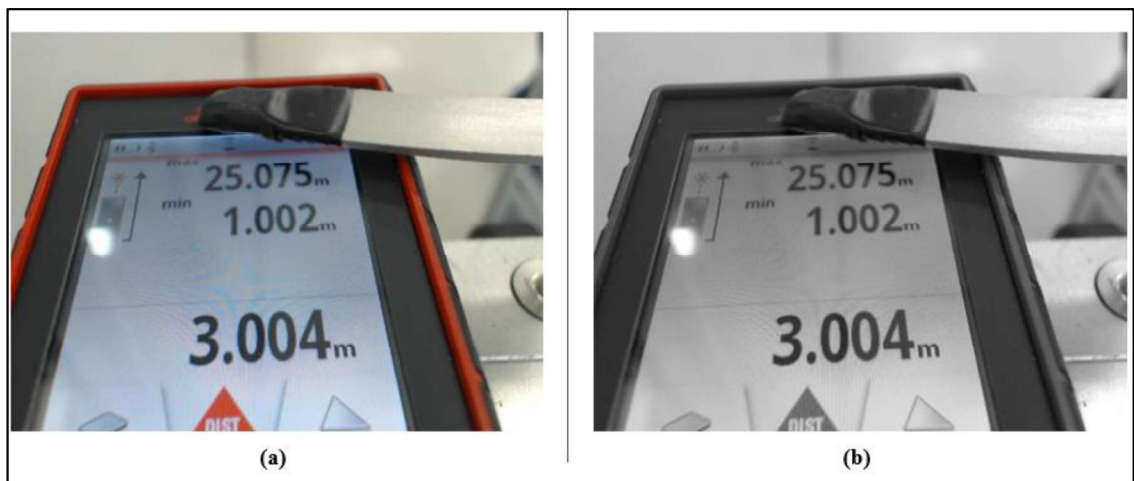
Por fim os botões “Trena 1” e “Trena 2” dão o comando para que o *software* faça a captura do display da trena e enviá-la para ser reconhecida.

3.2.2 Processamento das imagens

O processamento das imagens no software possui o objetivo de reduzir o máximo possível de ruídos na imagem adquirida. Este processamento foi feito através da aplicação de filtros de cores, limiarização, análise de contornos e outras técnicas de processamento de imagem, disponíveis na biblioteca *OpenCV*.

As Figuras a seguir demonstram esta etapa, da aplicação de filtros, e a imagem final a ser segmentada e reconhecida:

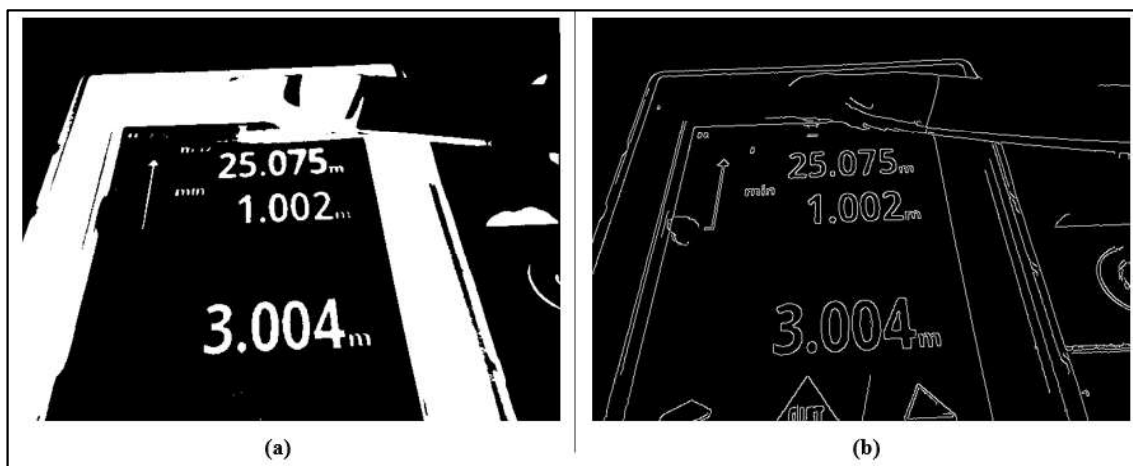
Figura 15- Processamento de imagem: etapas I e II



Fonte: autor

A Figura 15– a apresenta a imagem adquirida, sem filtros ou segmentação, sendo considerada como a primeira etapa do processamento de imagens. Em seguida, na Figura 15– b, temos a II etapa do processamento de imagens, aplicação do filtro de escala de cinza, para torná-la monocromática. A partir da escala de cinza é possível aplicar outros filtros, mostrados na Figura 16.

Figura 16- Processamento de imagens etapas III e IV



Fonte: autor

A Figura 16– a apresenta a imagem do display da trena a laser com um filtro de limiarização aplicado. A partir da escala de cinza e de um limite que foi determinado, os objetos da imagem serão separados em dois grupos: o grupo de pixel que possui níveis de cinza abaixo deste limite e o que possui níveis acima. Desta forma, a imagem se tornará binária, pois terá somente dois valores possíveis para cada pixel.

A Equação 1 define a limiarização da imagem:

Equação 1- Limiarização

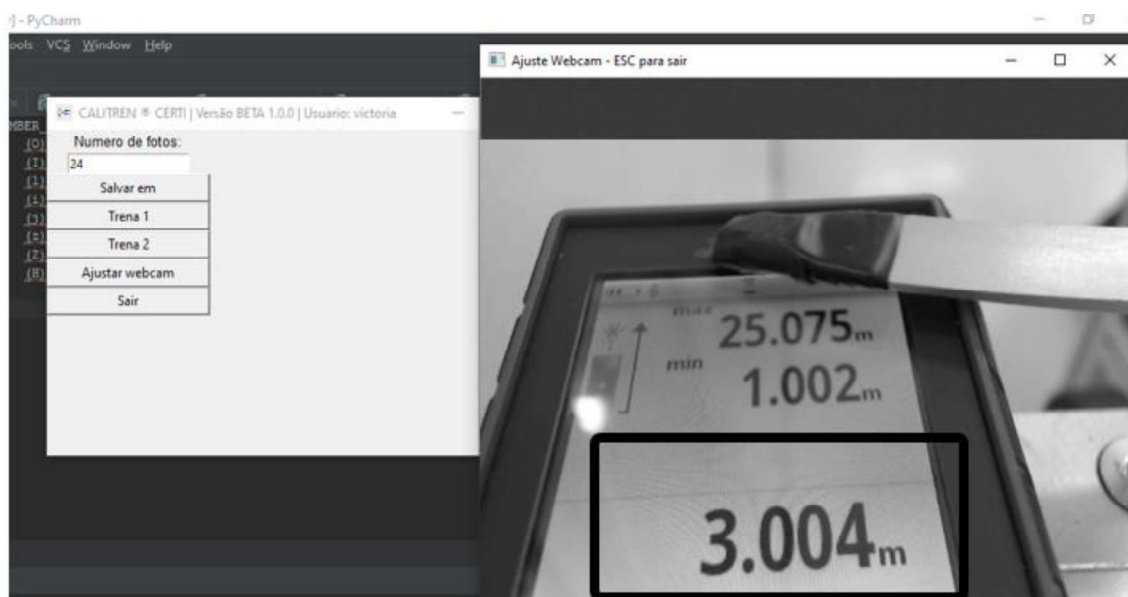
$$L(x, y) = \begin{cases} 1 & \text{se } f(x, y) > T \\ 0 & \text{se } f(x, y) \leq T \end{cases}$$

Onde temos que T é o limite determinado e $f(x, y)$ representa o pixel analisado. Para este caso, o limite T foi definido como 55 e todos os pixels abaixo deste valor foram segmentados para branco e os acima para preto.

A Figura 16– b apresenta a imagem com um filtro de detecção de bordas aplicado. Este filtro foi utilizado para verificar a descontinuidade dos números no display, analisando todas as bordas situadas na imagem. Esta etapa se fez necessária para todas as formas presentes na imagem e posteriormente determinar a região de interesse com os caracteres a serem reconhecido.

Durante os testes foram utilizadas 3 trenas de diferentes tamanhos e assim definido uma região (Figura 17) que comportasse todos os tipos de trenas possíveis recebidas no laboratório. A região de interesse foi determinada, com o auxílio de uma das funções do *OpenCV*, a partir da análise do tamanho do display das trenas a serem calibradas.

Figura 17- Região de interesse: ajuste da câmera



Fonte: autor

Esta região de interesse (ROI) se fez necessária por conta do fato da trena apresentar mais valores além daquele que está sendo medido, para que somente o valor da medição fosse reconhecido. A partir da ROI determinada, o software faz a captura das imagens, salvando o arquivo já com os filtros aplicados e com o corte da região de interesse pronto para ser reconhecido, pode ser visualizado na **Figura 18**.

Figura 18 - Imagem a ser reconhecida



Fonte: autor

3.2.3 Reconhecimento dos caracteres

Após determinar a região de interesse a ser analisada e reconhecida, o *software* segue para a etapa de reconhecimento de caracteres. Esta etapa consiste na análise dos

caracteres adquiridos via OCR, juntamente com a biblioteca de *Python pytesseract* e o *software* Tesseract. Para as trenas com caracteres numéricos estilo “7 segmentos” foi desenvolvido um algoritmo tendo como base o SSOCR (*seven segments optical character recognition*) desenvolvido por Erik Auerswald, pois o OCR comum não era capaz reconhecer, isso porque a estrutura do número estilo “7 segmento” é descontínua, fazendo com que o OCR só reconhecesse, o que seria cada bit, na letra “1”.

Como já mencionado, o OCR reconhece todos os caracteres na imagem, sendo letras ou números. Este fato precisa ser considerado e, para não haver conflitos e erros de reconhecimento, foi criado um dicionário para resolver estes problemas. Por exemplo, os números 0, 2 e 8 foram reconhecidos como O, z e B, em alguns testes e após uma análise da frequência dos erros de reconhecimento que estavam acontecendo (Tabela 2) criou-se um dicionário para resolver este problema.

Tabela 2- Análise do reconhecimento dos caracteres

Caracteres reconhecidos	Números									
	0	1	2	3	4	5	6	7	8	9
	o	l	z	H	A	S	g	T	B	q
	O	l	Z	E		s		/	%	
	Q		t			\$			&	
	D	:								
	t									

Fonte: autor

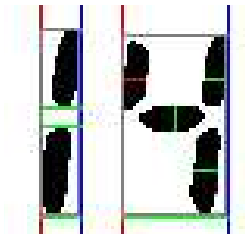
O dicionário funcionou de tal forma que, caso o *software* reconhecesse alguma dessas letras ou caracteres alfanuméricos, ele retornaria os valores determinados no dicionário. Criou-se, também, uma restrição em relação ao tamanho da *string* retornada para até 6 caracteres e à letra “m”, que corresponderia aos valores possíveis a serem reconhecidos: 1,000 a 25,000. Desta forma o algoritmo, mesmo reconhecendo a letra “m”, não a escreveria no arquivo final como um dígito reconhecido.

Como já mencionado, foi utilizado o algoritmo SSOCR como base para realizar o reconhecimento dos caracteres numéricos do estilo “7 segmentos”. Para este algoritmo o único ajuste feito foi a detecção do ponto, que a classe não reconhecia durante os testes. O processo de reconhecimento foi feito a partir de uma varredura vertical dos pixels seguindo a direção superior central do dígito para encontrar os quatro segmentos horizontais. A análise do dígito, apresentada na Figura 19, foi feita a partir da segmentação das linhas verticais e horizontais encontradas que foram identificadas com

o auxílio do dicionário criado, a ordem da identificação dos segmentos influencia diretamente.

Este último dicionário foi criado tendo como base o algoritmo de acionamento de displays de 7 segmentos, ou seja, cada linha vertical identificada corresponderia ao que seria um segmento do display. Assim, para cada conjunto de segmentos um número é formado.

Figura 19 - Análise do dígito



Fonte: Adaptado de <https://www.unix-ag.uni-kl.de/~auerswal/ssocr/>

Neste caso, foram encontrados 2 segmentos, no primeiro algarismo e 4 segmentos no segundo, porém é necessário determinar quais segmentos são esses, para que se possa determinar o número de forma correta de acordo com o dicionário criado. Após determinar os segmentos e verificá-los no dicionário, o dígito estará reconhecido.

4. TESTES E RESULTADOS

4.1 Testes experimentais

Os primeiros testes foram feitos para analisar o desempenho do software desenvolvido. Nestes testes o algoritmo analisava e reconhecia imagens que eram enviadas de forma manual, desta forma já era possível concluir se o algoritmo desenvolvido estava funcionando de acordo com o que era previsto e fazer as mudanças e restrições necessárias. Posteriormente outros testes foram realizados, já no ambiente de calibração das trenas e simulando todo o cenário do processo de calibração.

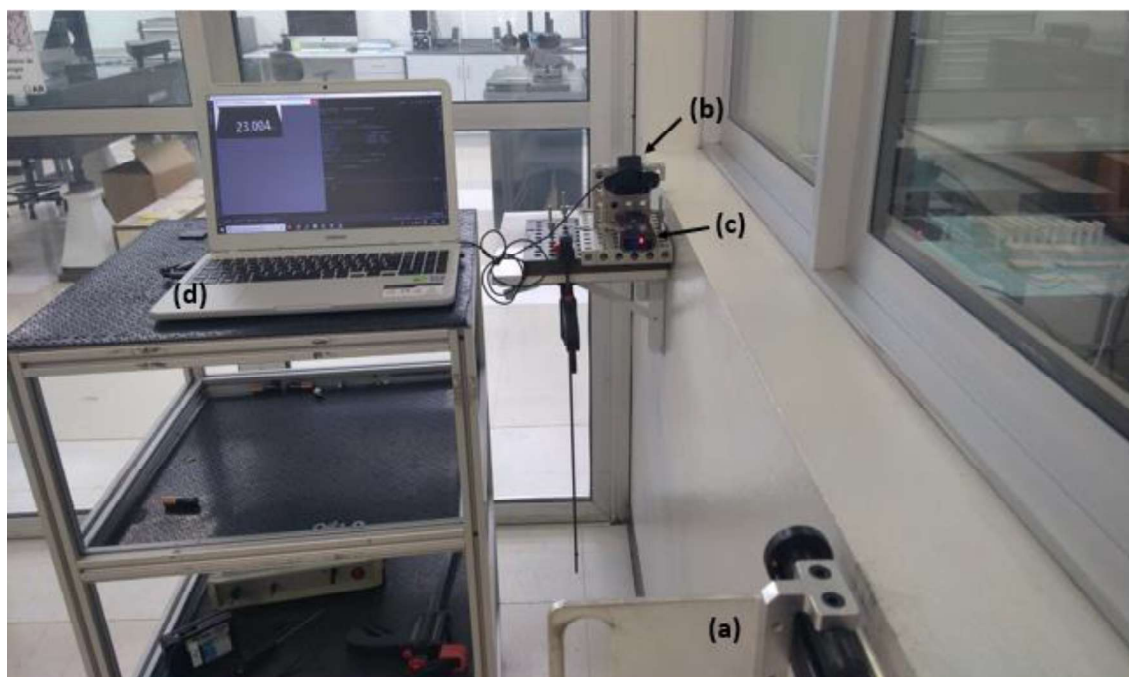
Foi utilizado um notebook, uma webcam e 2 trenas, uma para cada teste (OCR e SSOOCR), para o primeiro foi utilizado a trena LEICA DISTO 8 e para o segundo a trena Sndway. Os testes iniciais, com ambas as trenas, tinham como objetivo verificar se o tempo de captura de imagem era correspondido ao tempo de troca de posição dos alvos e o quanto a velocidade da troca de valores no display influenciava na captura das imagens. Por fim, foram feitos testes do reconhecimento dos caracteres com a finalidade de verificar a taxa de erro e analisar, para validar, a resposta final do sistema.

4.1.1 Testes de tempo de captura

Para a realização deste teste foi necessário posicionar a trena no ambiente de calibração, contar o tempo levado para fazer a troca de posições do alvo e analisar o quanto a mudança dos números no display da trena influenciava na imagem capturada.

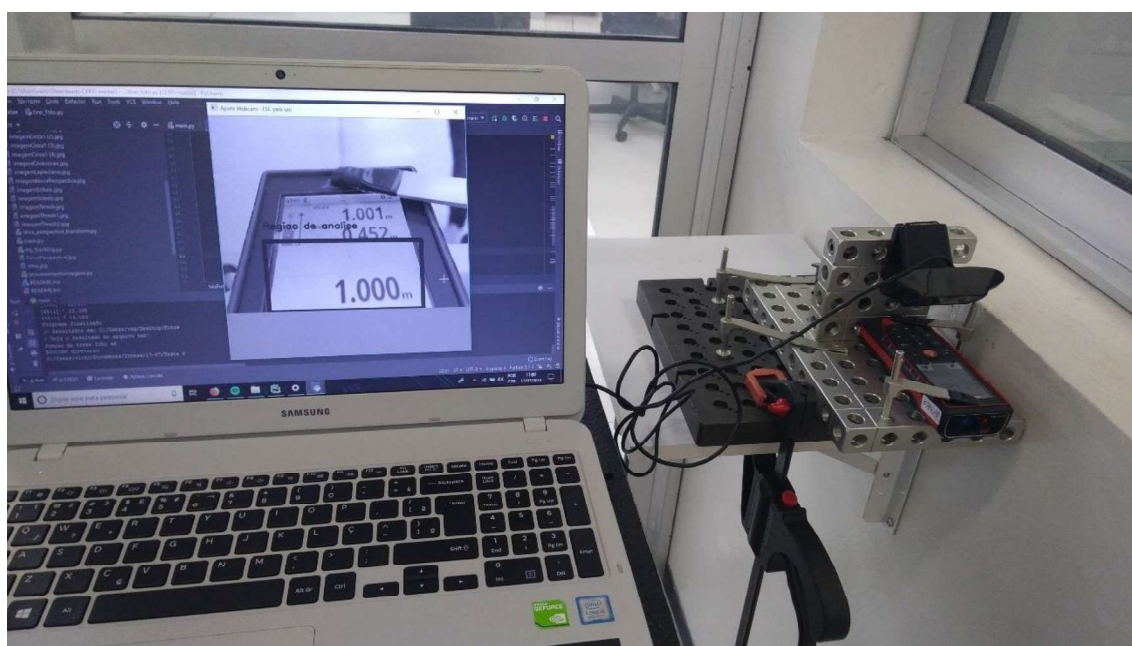
As Figura 20 e Figura 21 apresentam o cenário do teste I com o alvo posicionado (Figura 20 – a), a webcam (Figura 20 – b), trena a laser (Figura 20 – c) e com o notebook (Figura 20 – d) virado para o corredor, para que o operador que faz o deslocamento do alvo, consiga analisar a imagem.

Figura 20 – Cenário Teste I



Fonte: autor

Figura 21- Cenário Teste I



Fonte: autor

Os testes para analisar o tempo da troca de posicionamento do alvo foram feitos com 3 operadores trocando o alvo em 22 posições diferentes, por fim o tempo utilizado no sistema foi a média de todos os valores obtidos.

A Tabela 3- Registro do Teste 1 Tabela 3 apresenta os valores obtidos no teste citado acima.

Tabela 3- Registro do Teste 1

Operadores	Tempo da troca de posição	Incerteza
Operador 1	3s	±1s
Operador 2	4s	±1s
Operador 3	2s	±1s
Média	3s	±1s

Fonte: autor

Além da mudança de valor no display da trena, o tempo de estabilização do laser deve ser considerado. Isso porque enquanto o laser não estabilizar, aos valores da trena ficarão oscilando e a imagem ficará corrompida, como é possível visualizar na imagem Figura 22.

Figura 22- Mudança de valor no display



Fonte: autor

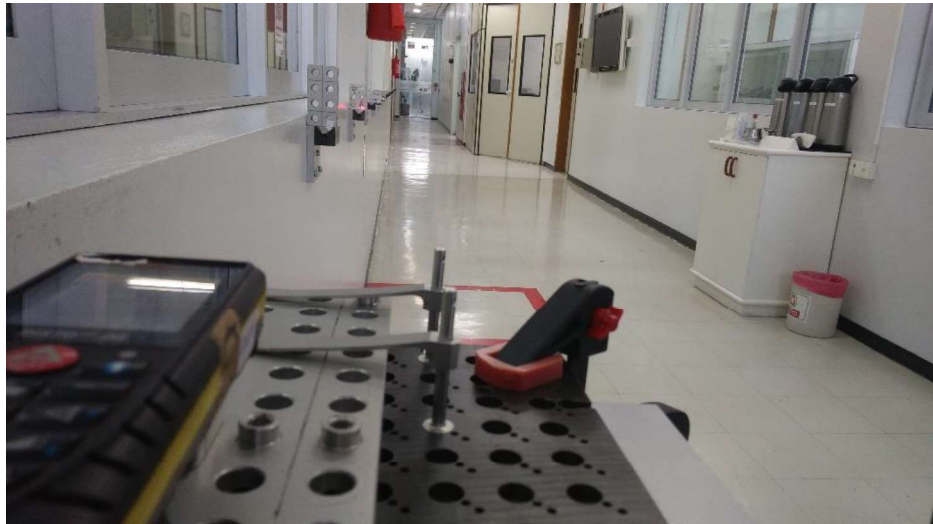
Para corrigir a estabilização do laser, foi determinado ao *software* fazer a captura de 4 imagens para cada ponto de medição com um intervalo de 1 segundo entre elas. Onde ele compara o valor reconhecido dos três e analisa o desvio padrão, caso não houver desvio, ele entende que aquele valor foi reconhecido corretamente. O intervalo entre as capturas das imagens em relação aos pontos de medição foi de 3.5 segundos, valor adaptado dos resultados obtidos no Teste 1, visto na Tabela 3.

4.1.2 Teste do reconhecimento dos caracteres

Para realizar os testes do reconhecimento dos caracteres obtidos no display da trena, foi utilizado o mesmo cenário do Teste 1 que pode ser visto na Figura 23 e na Figura 24. O teste consistiu em fazer o processo de calibração utilizando o *software*

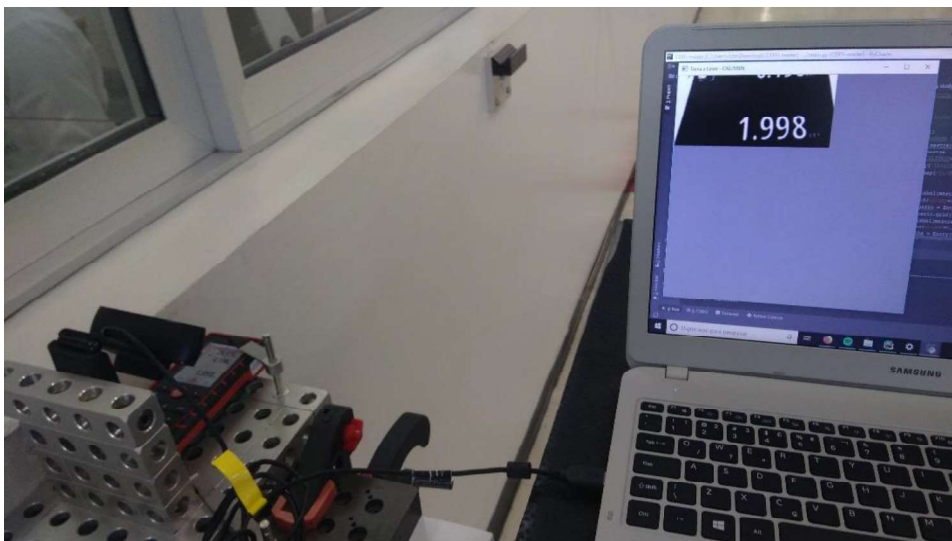
desenvolvido, ou seja, ligar o sistema, ajustar a câmera, inserir o número de capturas, neste caso foram 22 capturas, pois são 22 pontos, fazer a troca de posição do alvo e por fim importar a planilha gerada no *software* de certificação do laboratório.

Figura 23- Cenário Teste II: trena



Fonte: autor

Figura 24- Cenário Teste II: Trena Leica



Fonte: autor

A parte mais importante deste teste está na etapa de aquisição, reconhecimento dos números, desempenho do sistema e na verificação dos valores retornados. Além do reconhecimento de caracteres, o tempo de operação do processo de calibração também foi medido durante este teste.

4.2 Resultados

4.2.1 Viabilidade das técnicas de reconhecimento de caracteres

Durante o desenvolvimento deste trabalho, outras tecnologias além do OCR e do SSOOCR foram estudadas e utilizadas para o processo de reconhecer caracteres. A princípio a API do *google, AutoML Vision*, se apresentava como a melhor solução. Sem precisar de muitos conhecimentos na área de *machine learning* ou de programação, a ferramenta se mostrou simples e objetiva, sendo capaz de treinar o modelo em menos de 30 minutos, onde o único trabalho a ser realizado era a rotulação das imagens e a inserção destas no *Cloud Vision*.

Esta ferramenta faz apenas o processo de reconhecimento dos caracteres individualmente, não os detecta na imagem, fazendo com que fosse necessário a utilização de um algoritmo auxiliar para a detecção dos números na imagem. Isso porque os valores obtidos na trena a laser possuem, pelo menos, 5 caracteres, se não fosse este problema o *AutoML* seria suficiente.

Para a etapa de detecção dos números nas imagens foi estudada a técnica denominado “k vizinho mais próximo” (KNN). Porém durante os testes pôde se perceber que o modelo não possuía um *dataset* grande o suficiente para treinar o modelo de forma eficiente e que consumia muito processamento da máquina, o que deixava o processo muito lento. Além disso, houve algumas dificuldades durante a programação e o desenvolvimento do algoritmo.

Com base no que já foi dito anteriormente, a Tabela 4 apresenta a comparação dos critérios utilizados para a escolha da técnica de reconhecimento dos caracteres durante o estudo e o desenvolvimento do sistema, tendo como base a experiência pessoal do desenvolvedor e de seu desempenho.

Tabela 4 - Critérios de avaliação

Técnica de reconhecimento	Vantagens		Desvantagens		Total
	Fácil utilização	Versatilidade operacional	Conhecimento avançado de programação	Tempo de processamento	
OCR	3	3	-3	-2	1
SSOOCR	3	3	-3	-2	1
<i>AutoML Vision</i>	5	5	-2	-3	5
<i>KNN</i>	1	4	-5	-5	-5

Fonte: autor

A Figura 25 apresenta uma curva de aprendizagem que tem como objetivo apresentar a relação entre a curva de aprendizado adquirido durante o estudo e a motivação necessária para o desenvolvimento das atividades. A parte da curva, que pode ser observada entre os pontos destacados na imagem, representa o período de maior dificuldade de aprendizagem, do qual os testes com os algoritmos de *deep learning* estavam sendo feitos e os resultados não estavam sendo satisfatórios. Durante esta etapa o nível de aprendizado se estabilizou e isso ocorreu porque houve muitas dificuldades relacionadas a geração do modelo para detectar os caracteres e, mesmo após compreender e gerar um modelo, os resultados não foram satisfatórios. Somente após novos estudos e testes com alternativas de solução diferente, OCR e SSOCR, que foi possível a finalização da proposta de solução.

Figura 25- Curva de aprendizagem



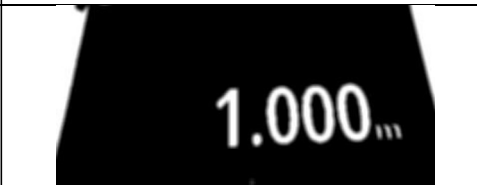
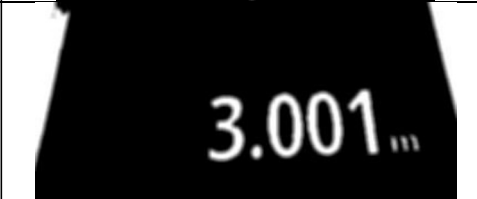
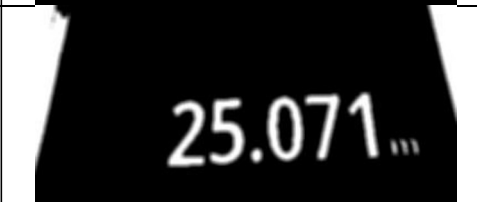
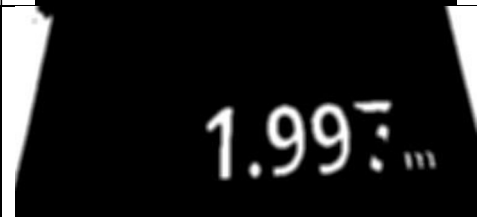
Fonte: autor

4.2.2 Reconhecimento dos caracteres

O algoritmo desenvolvido foi testado inúmeras vezes, para que a leitura e o funcionamento fossem verificados constantemente. Embora o sistema já estivesse configurado, durante os testes algumas mudanças ainda eram feitas, como ajustar o dicionário e alguns parâmetros dos filtros durante o processamento de imagens, visando melhorar o desempenho do sistema. Mesmo após os testes, todas as imagens que eram capturadas foram conferidas com o intuito de verificar a ocorrência de falhas.





A Tabela 5 e Tabela 6 apresentam algumas das leituras e resultados obtidos durante os testes com trena Leica e Sdneway respectivamente:

Tabela 5- Resultado Teste II : Trena 1

Imagens	Número Reconhecido
	1.000
	3.001
	25.071
	1.99/

Fonte: autor

Tabela 6- Resultado Teste II : Trena 2

Imagens	Número reconhecido
	0000
	0.291
	0.453
	1.188

Fonte: autor

Analisando a Tabela 5 onde os testes foram feitos com a trena LEICA DISTO 8 é possível visualizar que os resultados foram satisfatórios. Erros como aquele apresentado na quarta coluna da tabela foram corrigidos ao longo dos testes e, tirando esses ajustes, o software funcionou como esperado para o modelo de trena I. A Tabela 6 apresenta os valores obtidos durante os testes com a trena do modelo 2, Sndway. Pelo que apresentou, *software* reconheceu essas e as outras imagens, os resultados foram satisfatórios.

O algoritmo SSOCR alcançou com mais êxito considerando que, durante os testes, não foram feitas muitas alterações. O OCR precisar de mais ajustes pode ser justificado pelo fato de que este foi criado para reconhecer quaisquer caracteres, não só números, reconhecendo quaisquer caracteres, numéricos ou alfanuméricos, não sendo direcional.

4.2.3 Precisão do sistema

Para analisar o desempenho do sistema foi feito uma planilha para registrar os valores reconhecidos da imagem capturada, verificando se foi reconhecido ou não corretamente, e posteriormente estes dados foram colocados em um gráfico para examinar a precisão do sistema, calculada a partir da quantidade de erros obtidos em cada tentativa de teste.

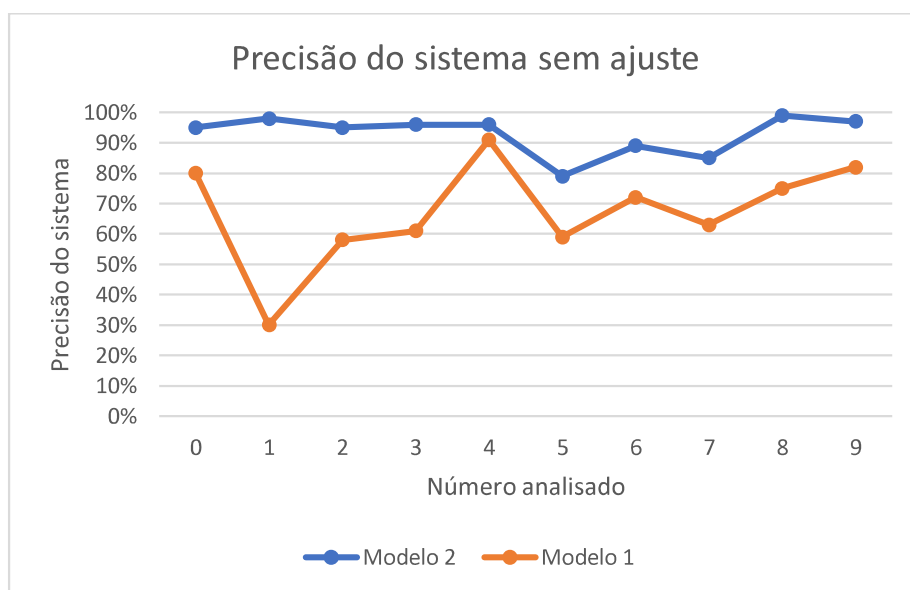
Os gráficos abaixo apresentam a precisão do sistema, calculada a partir da Equação 2, durante os testes, sendo a cor amarela representando os valores obtidos nos testes com a trena de Modelo 1 e a de cor azul os valores obtidos com a trena do Modelo

2. O - Gráfico I apresenta a precisão do sistema sem qualquer ajuste ou correção, o Gráfico II apresenta a precisão do sistema após ajustes no algoritmo como: o acréscimo de caracteres a serem verificados no dicionário e definição de parâmetros nos filtros para manipulação das imagens. Por fim, o Gráfico III apresenta a precisão do sistema final.

Equação 2 - Precisão do sistema

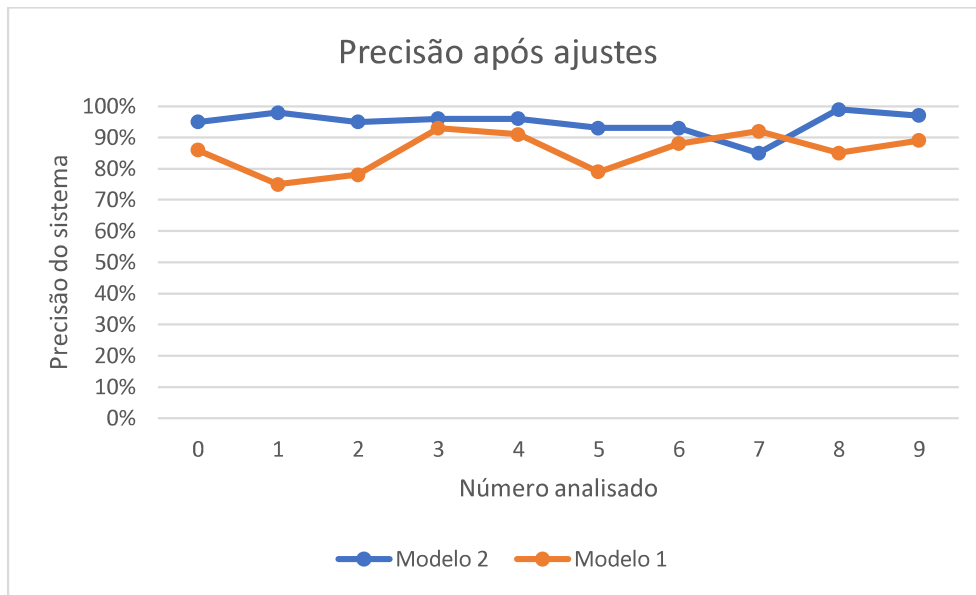
$$\text{Precisão do sistema} = \frac{\text{Reconhecimento Assertivo}}{\text{Numero de testes realizados}} * 100\%$$

Figura 26- Gráfico I



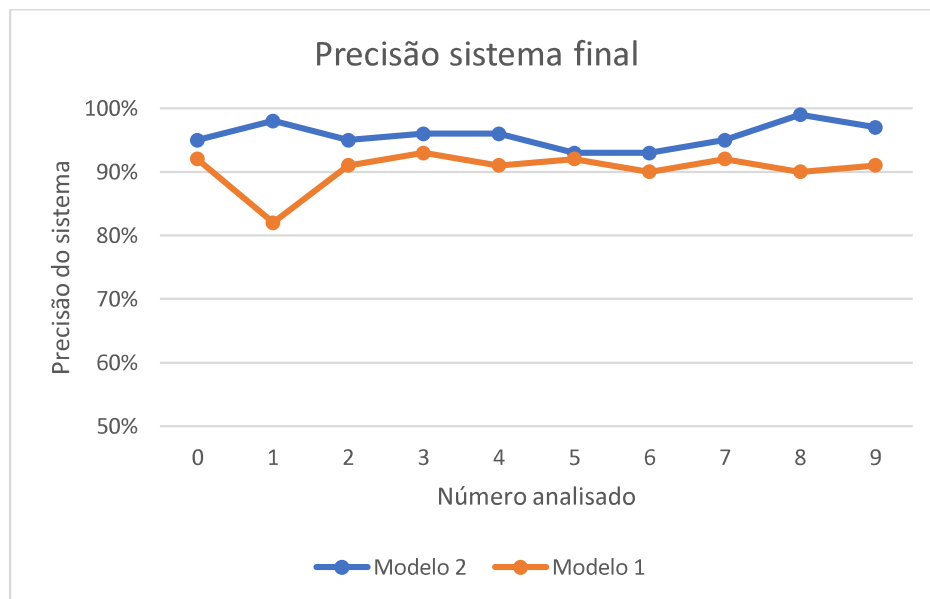
Fonte: autor

Figura 27- Gráfico II



Fonte: autor

Figura 28- Gráfico III



Fonte: autor

Através do- Gráfico I é possível perceber que para a trena de Modelo 1 o sistema não funcionou muito bem, principalmente no reconhecimento do número 1 e já para o Modelo 2 o *software* obteve dificuldades para diferenciar os números 5 e 6. As dificuldades encontradas para o Modelo1 podem ser justificadas pelo fato de o sistema reconhecer outras letras e caracteres alfanuméricos, além de números. Para o Modelo 2, a explicação encontrada e os ajustes se deram no processamento das imagens, ou seja, a partir dos ajustes durante a manipulação das imagens, os erros já foram minimizados.

Após feita as correções, um teste novo foi feito e os resultados podem ser visualizados no Gráfico II. O ajuste feito sobre o Modelo 2 entre os testes foi suficiente para corrigir o problema antes analisado. O Modelo 1 continuou apresentando uma precisão baixa para os números 1, 2 e 5 em relação aos outros e, por isto, mais ajustes foram feitos até que os últimos testes foram realizados conseguindo uma média de aproximadamente 90% de precisão para ambos os modelos. Este resultado pode ser visto no Gráfico III.

4.2.4 Tempo de operação

O tempo total da calibração, considerando todo o ajuste mecânico, de *software* e o procedimento de calibração foi medido durante todos os testes, como estes foram realizados por três operadores diferentes, foi retirada a média, e podem ser visualizados na Tabela 8 . Para comparar com o procedimento manual a Tabela 7 apresenta o tempo gasto durante os ajustes mecânicos, realização dos ciclos de calibração e passagem dos valores para o *software* de certificação.

Tabela 7- Duração do procedimento manual

Atividade	Duração
Ajustes e alinhamento da trena	5 minutos
Realização dos 3 ciclos de calibração.	15 minutos
Passagem dos valores anotados para o <i>software</i> de certificação.	5 minutos
Tempo total:	25 minutos

Fonte: autor

Tabela 8- Duração do procedimento otimizado

Atividade	Duração
Ajustes e alinhamento da trena e da câmera	5 minutos
Realização dos 3 ciclos de calibração.	5 minutos
Importar arquivo para o <i>software</i> de certificação.	1 minuto
Tempo total:	11 minutos

Fonte: autor

Examinando e comparando as duas tabelas é possível afirmar que houve uma redução considerável no tempo de duração do procedimento de calibração de trenas a laser utilizando o sistema desenvolvido.

5. CONCLUSÃO E TRABALHOS FUTUROS

O presente trabalho descreve o desenvolvimento de um sistema de reconhecimento de caracteres numéricos para otimização do processo de calibração de trenas a laser. No *software* desenvolvido foi implementado, a partir de análises e estudos, o processamento e a manipulação das imagens a fim de reconhecer os caracteres numéricos presentes nessa imagem.

Com base nos resultados obtidos nos cenários dos quais o sistema foi testado é possível afirmar que o objetivo geral de desenvolver um sistema de visão para calibração de trenas a laser foi alcançado. Além disso, os objetivos específicos também foram atingidos, visto que foi desenvolvido um sistema com interface gráfica capaz de manipular e processar imagens, reconhecer os caracteres numéricos com uma precisão suficiente para reconhecer, sem que sejam feitos muitos ciclos de operação, foram feitos os testes e houve uma redução de quase 50% do tempo de calibração.

Entre as melhorias e os trabalhos futuros para o sistema, estão a comparação com outros algoritmos de reconhecimento de caracteres, estudar a possibilidade de o sistema reconhecer qualquer tipo de número, possibilitando uma maior flexibilidade, e testá-lo com o maior número de trenas possíveis. Construir um *dataset* para cada modelo de trena, sendo robusto o suficiente para que as técnicas de reconhecimento com *deep learning* possam ser utilizadas juntamente com a API do *Google AutoML Vision* que se demonstraram como uma boa solução para o problema, mas que por conta das dificuldades e do tempo, não foi possível utilizá-las.

Após isso, implementar o algoritmo em um microcomputador e conectá-lo à nuvem ou diretamente ao *software* de certificação, visando tornar o sistema compacto, portátil e com maior desempenho, para que possa ser utilizado em outros lugares além do laboratório.

Referências

ACHARYA, Tinku e AJOY, RAY K. **Image Processing Principles and Applications**. Kharagpur: [s.n.], 2005.

ALMEIDA, Luiz. **Estudo de caso em Reconhecimento Automático de Placas Veiculares usando linguagem de programação Python**. 2018. Disponível em: <http://www.cear.ufpb.br/arquivos/cgee/TCC/TCC_-_Luiz_Eduardo_P._M._Almeida_-_Versão_Final.pdf>.

ANDREW, Anthony e colab. **LASER NA ATIVIDADE DE VISTORIA DO CORPO DE**. 2015.

ANTONELLO, Ricardo. **Introdução a Visão Computacional com Python e OpenCV**. v. 1, p. 46, 2017. Disponível em: <<http://professor.luzerna.ifc.edu.br/ricardo-antonello/wp-content/uploads/sites/8/2017/02/Livro-Introdução-a-Visão-Computacional-com-Python-e-OpenCV.pdf>>.

ASSIS, SR De e MACIEL, RSP. **Um framework de apoio ao desenvolvimento de aplicações baseadas em documentos**. n. January, 2008. Disponível em: <<http://www.uefs.br/erbase2004/documentos/wticgbase/Wticgbase2004ArtigoMonografia006.pdf>>.

BARRETO, Jorge M. **Introdução às Redes Neurais Artificiais**. V Escola Regional de Informática. Sociedade Brasileira de Computação, Regional Sul, Santa Maria, Florianópolis, Maringá, p. 5–10, 2002. Disponível em: <https://06486feb-a-62cb3a1a-sites.googlegroups.com/site/professorrn6/computacao/TC-RedesNeuraisArtificiais.pdf?attachauth=ANoY7cqnnNtEGxQpQmGoiIa2V0GB7reXWZjoCf034xBbKV7LvCug04Iv32_4DYsFaq4wLeLMuELZx6oB1WTAuGbcWkopnY9sIeLXMsn2vgB3XLVAXxQxn2KIT6TbHwtFla9>.

BERNARDI, Ely. **Os sistemas de identificação veicular, em especial o reconhecimento automático de placas**. p. 1–136, 2016. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3138/tde-11052016-162646/>>.

BORGES, Luiz Eduardo. **Python para Desenvolvedores**. 2. ed. Rio de Janeiro: [s.n.], 2010.

BRANCO, Pato e colab. **Laser Scanner Terrestre: teoria, aplicações e prática**. p. 99–109, 2016.

CAMILO, Santelmo; **Trenas a laser oferecem precisão para medir grandes distâncias** | AECweb. Disponível em: <<https://www.aecweb.com.br/cont/m/rev/trenas->

a-laser-oferecem-precisao-para-medir-grandes-distancias_17140_10_22>. Acesso em: 10 jun 2019.

CARDINAL, David. **How to get started with DIY home surveillance systems - ExtremeTech**. Disponível em: <<https://www.extremetech.com/electronics/232428-home-video-monitoring-for-the-do-it-yourselfer-an-overview>>. Acesso em: 21 ago 2019.

CORREA, E. **Meu primeiro livro de Python**. [S.l: s.n.], 2017.

CYRUS, Farivar. **New software watches for license plates, turning you into Little Brother | Ars Technica**. Disponível em: <<https://arstechnica.com/information-technology/2015/12/new-open-source-license-plate-reader-software-lets-you-make-your-own-hot-list/>>. Acesso em: 21 ago 2019.

FACON, J. **Processamento e análise de imagens**. Universidad Nacional de Córdoba, Argentina, p. 1–46, 1993. Disponível em: <<http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Processamento+e+An?lise+de+Imagens#0>>.

FARUOLO, Luciano Bruno. **A importância do ensino de metrologia, com foco na incerteza de medição, na formação de engenheiros**. n. August, 2015.

FERRAMENTAS GERAIS COMÉRCIO E IMPORTAÇÃO DE FERRAMENTAS E MÁQUINAS LTDA. **Trenas a Laser - Confira 5 benefícios para medição em obras**. Disponível em: <<http://conectafg.com.br/trenas-a-laser/>>. Acesso em: 10 maio 2019.

FILHO, Ogê Marques. **UMA PROPOSTA DE RECONHECIMENTO DE CARACTERES UTILIZANDO CORRELAÇÃO COM MAScARAS BINÁRIAS CBINARY TEHPLATE HATCHJNG**). p. 80–89, 1999.

FILHO, Ogê Marques e NETO, Hugo Vieira. **Processamento Digital de Imagens**. [S.l: s.n.], 1999. v. 21/03. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-261X1998000100035&lng=pt&nrm=iso&tlng=pt>.

GALLON, Lucas. **Sistema de visão computacional para classificação de pedras naturais através de vídeo em tempo real**. 2014. Disponível em: <<https://www.univates.br/bdu/handle/10737/501>>.

GOMES CAROLINE. **O que é OCR e quais os benefícios para a sua empresa**. Disponível em: <<https://blog.simply.com.br/o-que-e-ocr-e-beneficios/>>. Acesso em: 20 ago 2019.

GONÇALVES, Gabriel Resende. **License Plate Recognition Based on**. 2016. 75 f.

Universidade Federal de Minas Gerais, 2016.

GONÇALVES, Julio Cesar e CENTENO, Tania Mezzadri. **Utilização de Técnicas de Processamento de Imagens e Classificação de Padrões no Reconhecimento de Dígitos em Imagens de Medidores de Consumo de Gás Natural**. *Abakós*, v. 5, n. 2, p. 59, 2017.

GONZALEZ, Rafael C. **Digital_Image_Processing_2ndEd.pdf**. . [S.l: s.n.]. , 2002

GUIMARÃES, Heitor Rodrigues. **RECUPERAÇÃO DE INFORMAÇÕES MUSICAIS: UMA ABORDAGEM UTILIZANDO DEEP LEARNING** Heitor. 2018. 121 f. Escola Politécnica da Universidade Federal do Rio de Janeiro, 2018.

GUINGO, Bruno Clemente e CARLOS, Antonio e THOMÉ, Gay. **de Características e Reconhecimento de Caracteres de Placas de Veículos**. [S.d.].

INMETRO - INSTITUTO NACIONAL DE METROLOGIA QUALIDADE E TECNOLOGIA. **Vocabulário Internacional de Termos de Metrologia Legal**. [S.l: s.n.], 2012.

INTRUSUL, Intrumentos de medição. **O que é e qual a melhor trena a laser? - Blog Instrusul**. Disponível em: <<http://blog.instrusul.com.br/o-que-e-e-qual-melhor-trena-laser/>>. Acesso em: 31 jul 2019.

KOTU, Vijay e DESHPANDE, Bala. **Deep Learning**. *Data Science*, p. 307–342, 2016.

MARENGONI, Maurício e STRINGHINI, Denise. **Tutorial: Introdução à Visão Computacional usando OpenCV**. *Revista de Informática Teórica e Aplicada*, v. 16, n. 1, p. 125–160, 2009. Disponível em: <http://seer.ufrgs.br/rita/article/download/rita_v16_n1_p125/7289>.

MATTSSON, Michael. **Evolution and composition of object-oriented frameworks**. [S.l: s.n.], 2000. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.96.2067&rep=rep1&type=pdf>>.

OpenALPR Documentation — openalpr 2.6.101 documentation. Disponível em: <<http://doc.openalpr.com/>>. Acesso em: 21 ago 2019.

PATEL, Chirag e PATEL, Atul e PATEL, Dharmendra. **Optical Character Recognition by Open source OCR Tool Tesseract: A Case Study**. *International Journal of Computer Applications*, v. 55, n. 10, p. 50–56, 2012.

PONTI, Moacir Antonelli e DA COSTA, Gabriel B. Paranhos. **Como funciona o Deep Learning**. [S.l: s.n.], 2018. Disponível em: <<http://arxiv.org/abs/1806.07908>>.

RESENDE, Rodrigo Ribeiro e colab. **Princípios fundamentais dos lasers e suas**

aplicações. Biotecnologia Aplicada à Agro&Indústria - Vol. 4, p. 815–894, 2017.

SANTOS, CN Dos. **Aprendizado de máquina na identificação de sintagmas nominais: o caso do português brasileiro.** ... thesis, Instituto Militar de Engenharia, Rio ..., 2006. Disponível em:
<<http://pandeiro.learn.fplf.org.br/AM/images/9/90/DissertacaoCicero.pdf>>.

SCHOENKNECHT, Eduardo. **Projeto de diplomação medição de distância com o uso de laser.** 2010.

SILVA, Eugênio. **Inteligência Computacional - Reconhecimento de Caracteres.** Disponível em: <<http://www.nce.ufrj.br/conceito/artigos/2006/016p1-3.htm>>. Acesso em: 15 ago 2019.

BRANCO, Pato e colab. **Laser Scanner Terrestre: teoria, aplicações e prática.** p. 99–109, 2016.

CAMILO, Santelmo; **Trenas a laser oferecem precisão para medir grandes distâncias | AECweb.** Disponível em: <https://www.aecweb.com.br/cont/m/rev/trenas-a-laser-oferecem-precisao-para-medir-grandes-distancias_17140_10_22>. Acesso em: 10 jun 2019.

FARUOLO, Luciano Bruno. **A importância do ensino de metrologia, com foco na incerteza de medição, na formação de engenheiros.** n. August, 2015.

FERRAMENTAS GERAIS COMÉRCIO E IMPORTAÇÃO DE FERRAMENTAS E MÁQUINAS LTDA. **Trenas a Laser - Confira 5 benefícios para medição em obras.** Disponível em: <<http://conectafg.com.br/trenas-a-laser/>>. Acesso em: 10 maio 2019.

GALLON, Lucas. **Sistema de visão computacional para classificação de pedras naturais através de vídeo em tempo real.** 2014. Disponível em:
<<https://www.univates.br/bdu/handle/10737/501>>.

GONÇALVES, Julio Cesar e CENTENO, Tania Mezzadri. **Utilização de Técnicas de Processamento de Imagens e Classificação de Padrões no Reconhecimento de Dígitos em Imagens de Medidores de Consumo de Gás Natural.** Abakós, v. 5, n. 2, p. 59, 2017.

INMETRO - INSTITUTO NACIONAL DE METROLOGIA QUALIDADE E TECNOLOGIA. **Vocabulário Internacional de Termos de Metrologia Legal.** [S.l: s.n.], 2012.

MARENGONI, Maurício e STRINGHINI, Denise. **Tutorial: Introdução à Visão Computacional usando OpenCV.** Revista de Informática Teórica e Aplicada, v. 16, n.

1, p. 125–160, 2009. Disponível em:

<http://seer.ufrgs.br/rita/article/download/rita_v16_n1_p125/7289>.

RESENDE, Rodrigo Ribeiro e colab. **Princípios fundamentais dos lasers e suas aplicações**. Biotecnologia Aplicada à Agro&Indústria - Vol. 4, p. 815–894, 2017.

SANTOS, CN Dos. **Aprendizado de máquina na identificação de sintagmas nominais: o caso do português brasileiro**. ... thesis, Instituto Militar de Engenharia, Rio ..., 2006. Disponível em:

<<http://pandeiro.learn.fplf.org.br/AM/images/9/90/DissertacaoCicero.pdf>>.