



DESENVOLVIMENTO DE PROTÓTIPO EXPERIMENTAL PARA ANÁLISE DE INSPEÇÃO DIMENSIONAL EM PERFIS DE ALUMÍNIO

Patryck Civiero Elias¹

Fernando Guessi Placido²

Resumo: O presente trabalho busca propor a utilização dos sistemas de visão computacional, aplicados no meio industrial para fins de inspeção dimensional em perfis de alumínio extrudados. O mercado de perfis de alumínio vem ganhando espaço nos últimos anos, e com isso surge a necessidade de controles de qualidade mais precisos e confiáveis, afim agregar ainda mais valor na sua comercialização. O objeto do trabalho é o desenvolvimento de um protótipo experimental, integrando hardware e software afim realizar medições qualitativas em perfis extrudados. Os resultados dos estudos indicam que os sistemas de visão podem sim ser utilizados para esta aplicação, entretanto é preciso que os componentes periféricos de captação de dados sejam dimensionados de acordo com o nível de precisão que se almeja alcançar.

Palavras-Chave: Alumínio. Inspeção. Perfis. Sistemas. Visão Computacional.

1 INTRODUÇÃO

O mercado do alumínio vem apresentando crescimento contínuo nos últimos anos, pesquisas mostram que o consumo de alumínio vem aumentando, e aliado a isso surge a necessidade maiores capacidades produtivas por parte das empresas do seguimento.

Segundo ABAL no Anuário estatístico Alumínio 2019, o crescimento do produto interno bruto cresceu cerca de 1,1% no ano de 2019, tendo em vista as projeções de fato foi um crescimento baixo, porém mesmo assim o consumo de alumínio aumentou cerca de 8,2 % no mesmo ano, atingindo o número de 1485 mil toneladas, ficando próximos de números recordes do seguimento nos últimos anos ABAL (2020).

¹ Graduando em Engenharia Mecatrônica. Ano 2021-2. E-mail: patryck96373320@gmail.com

² Professor do Centro Universitário UniSATC. E-mail: fernando.placido@satc.edu.br



Dentre os vários seguimentos as quais os produtos de alumínio se aplicam, um apresentou destaque pelo aumento significativo em seu potencial de produção “a capacidade instalada de produtos extrudados apresentou maior crescimento, passando de 469 mil toneladas em 2011 para 489 mil toneladas ao fim de 2012 – volume adicional de 20 mil toneladas, com 174 prensas em operação “(Ministério da Indústria, Comércio Exterior e Serviços, 2016).

Um mercado aquecido como o de perfis de alumínio torna-se uma opção de investimento interessante, já que, além de poder ser comercializado como produto acabado, ele serve como matéria prima para uma infinidade de outras mercadorias. Desta forma é possível observar que os derivados do alumínio estão presentes em diversas cadeias do processo produtivo, podendo ser aplicado em objetos comuns como latas de refrigerante ou comida, mas também está presente em soluções de alta criticidade, como fuselagem de aviões e estruturas de embarcações.

As características citadas acima possivelmente são fatores que propiciaram o aumento do número de prensas de extrusão instaladas no país, mas em paralelo a isso surge a necessidade de criação de novos métodos de controle da qualidade, que por sua vez, tem como objetivos garantir uma boa eficiência e assertividade na produção deste material.

Diante dos dados apresentados, os quais demonstram a alta competitividade que existe neste seguimento, a fábrica de produtos extrudados do GRUPO IBRAP busca otimizar seus custos operacionais para se tornar mais competitiva frente ao mercado.

Uma das alternativas para aumentar a competitividade de negócios, está relacionada a redução de custos a partir do controle assíduo da qualidade em seus produtos. Este processo se torna possível com a redução e controle de defeitos nas fases iniciais do processo, o que resulta em queda de custos no beneficiamento de materiais que serão reprovados ao final do processo, e possivelmente colabora para redução da incidência das reclamações e devoluções de materiais já entregues aos seus clientes.

Uma possível proposta para tal seria o uso de conceitos de programação e automação, a fim de desenvolver um protótipo integrando visão computacional e hardware adaptável, capaz de diferenciar amostras aprovadas e reprovadas dentro dos padrões de qualidade definidos pelo Grupo IBRAP.



2 REVISÃO BIBLIOGRÁFICA

Nesta sessão serão apresentados os embasamentos teóricos utilizados para desenvolvimento do projeto.

2.1 SISTEMAS DE VISÃO COMPUTACIONAL

Existem opiniões distintas acerca da importância dos sentidos humanos, mas segundo Antonello, “os cinco sentidos dos seres humanos são: olfato, tato, audição, paladar e visão. De todos eles, temos que concordar que a visão é o mais, ou pelo menos, um dos mais importantes. (ANTONELLO; RICARDO, 2016, p. 03).

A visão tem papel fundamental na percepção dos seres humanos quanto ao ambiente a sua volta, contudo o “privilegio” da visão não se restringe apenas aos seres humanos, ou ao grupo seleto dos seres vivos. Com os avanços tecnológicos das últimas décadas as máquinas estão adquirindo de certa forma esta capacidade, sendo isso possível através dos Sistemas de Visão Computacional.

2.1.1 Definição / Contexto Histórico

Durante a década de oitenta no ano de 1982, os escritores Ballard e Brown definiram que os estudos responsáveis por dar a uma máquina a capacidade de perceber características do ambiente ao seu redor definiria segundo suas opiniões o conceito de Visão Computacional, esta capacidade seria obtida através da utilização de dispositivos de captura e análise de imagens (BALLARD, BROWN, 1982, p. 12).

Como descrito acima durante os anos 80 já existiam estudos referentes a visão computacional, como a obra *Computer Vision*, entretanto, “as primeiras menções sobre Visão Computacional ocorreram na década de 50. Já os primeiros trabalhos foram iniciados por volta de 20 anos depois” (Barelli; Felipe, 2018, p. 03).

Os sistemas de visão como muitas das tecnologias que foram desenvolvidas ao longo dos anos, surgiram com intuito de resolver problemas do cotidiano ou do meio industrial, trazendo maior comodidade para a sociedade, dentre

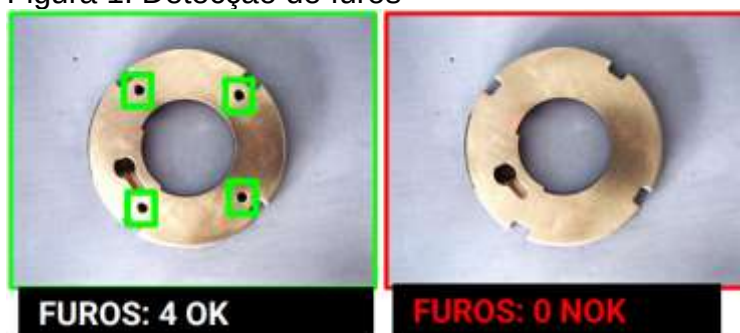
as atribuições dadas a este advento temos que segundo Antonelo (2018, p.06.), na obra Introdução a Visão computacional com Python e OpenCV,

Neste ponto é importante frisar que, além de fazer máquinas “enxergarem”, reconhecerem objetos, paisagens, gestos, faces e padrões, os mesmos algoritmos podem ser utilizados para reconhecimento de padrões em grandes bases de dados, não necessariamente feitos de imagens.

2.1.2 Aplicações de Sistemas de Visão Computacional

Existem vários exemplos em que são aplicados os sistemas de visão computacional, é possível encontrar este tipo de tecnologia no meio industrial para controle de qualidade dos produtos, como no caso mostrado na Fig. 1, onde se faz uso da ferramenta de visão para detecção de detalhes específicos de uma peça fabrica em serie.

Figura 1: Detecção de furos



Fonte: Adaptado de Panfleto de Aplicações MVISIA

Também podem ser observada este tipo de tecnologia no trânsito através dos veículos autônomos e semiautônomos, reguladores de tráfego e câmeras de monitoramento de trânsito. Na Fig. 2 podemos observar um sistema de visão integrado a veículo autônomo, sendo o mesmo capaz de diferenciar diversos objetos, pessoa e outros veículos.

Figura 2: Detecção de objetos na rua



Fonte: Daumas (2020)

2.1.3 Linguagem de Programação Python

O Python é uma linguagem de programação muito difundida atualmente, sua criação ocorreu no final da década de oitenta mais precisamente no ano de 1989 e seu criador foi o Programador Guido Van Rossum, segundo o próprio Rossum, "tudo começou com o ABC, uma linguagem de ensino maravilhosa que ajudei a criar no início dos anos oitenta. Era uma linguagem incrivelmente elegante e poderosa, destinada a programadores não profissionais" (Lutz; Mark, 1996).

Ao longo dos anos vários softwares foram criados utilizando como linguagem base o Python, segundo Rossum (1996), isso se deve pelo fato de o Python apresentar em sua composição soluções para pontos considerados negativos encontrados em outras formas de programação, por isso ele teve uma difusão tão boa entre os programadores. Na Fig. 3 podemos observar um breve exemplo de programação escrita em Python, nele o programador executa a importação de biblioteca e faz referência a um arquivo de vídeo que seu programa utilizara para realização de leitura dos dados.



Figura 3: Programa escrito em Python

```
import cv2

captura = cv2.VideoCapture("video.mp4")

while True:
    ret, frame = captura.read()
    cv2.imshow("Imagem", frame)

    if cv2.waitKey(1) & 0xFF == ord("q"):
        break

captura.release()
cv2.destroyAllWindows()
```

Fonte: Barelli (2018)

Para a que os computadores sejam capazes de interpretar programas escritos em Python, é necessário realizar a instalação do interpretador do mesmo, em alguns casos os sistemas operacionais como *macOS* e *Linux* que já tem esta funcionalidade instalada de fábrica, enquanto os equipamentos com *Windows* nem em sua versão mais recente dispõem desta ferramenta de forma nativa (Barelli; Felipe, 2018).

Nos casos em que o interpretador do Python ainda não estiver instalado ao computador, o programa de instalação dele pode ser encontrado no site oficial da plataforma (<https://www.python.org/>).

2.2 BIBLIOTECA OPENCV

Existem várias formas de definir o que são Bibliotecas de Programação, para Barelli: “as bibliotecas são pacotes com funções que não são nativas da linguagem” (Barelli; Felipe, 2018, p. 15).



Existe uma infinidade de Bibliotecas de Programação à disposição dos programadores, visto que muito do material que às compõem provém do compartilhamento de programas criados pelos próprios usuários, mas existem também de empresas especializadas no desenvolvimento deste tipo de material.

A “OpenCV” foi desenvolvida com propósito específico para aplicação em sistemas de visão, segundo o site oficial da OpenCV.

OpenCV (Open Source Computer Vision Library) é uma biblioteca de software de visão computacional e aprendizado de máquina de código aberto. O OpenCV foi construído para fornecer uma infraestrutura comum para aplicativos de visão computacional e para acelerar o uso da percepção da máquina em produtos comerciais. Por ser um produto licenciado por BSD, o OpenCV torna mais fácil para as empresas utilizar e modificar o código.

A biblioteca OpenCV de acordo com Barelli (2018),” Foi originalmente desenvolvida pela Intel em 2000, sendo totalmente livre para o uso acadêmico e comercial” (Barelli; Felipe, 2018, p. 16).

Na figura abaixo pode-se observar a escrita de programação em Python para que seja importada a biblioteca OpenCV, a partir do momento que a biblioteca é importada, todos os recursos inclusos a mesma, passam a estar à disposição do programador para criação do programa.

Pelo fato de OpenCV ser uma biblioteca dedicada a aplicações voltadas a sistemas de visão, ela foi escolhida como biblioteca base para a programação do protótipo

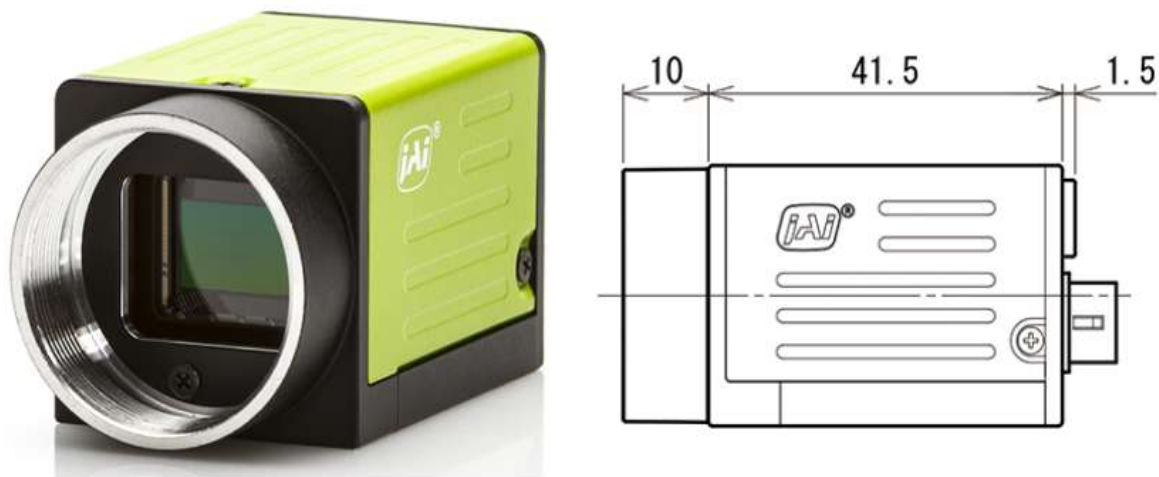
2.3 DISPOSITIVOS DE CAPTURA DE IMAGEM

Nessa sessão do trabalho serão apresentadas duas opções de câmeras que podem ser utilizadas para implementação do projeto, estas foram escolhidas levando em consideração aplicabilidade ao protótipo, sem que fossem considerados, neste momento, características comerciais como valor do produto e lote mínimo de venda.

2.3.1 Câmera GO-5100MP-USB Polarizada (IMX250MZR)

Segundo o site oficial em que a câmera GO-5100MP é comercializada, sua configuração e ideia para aplicação de visão industrial, vigilância ou tráfego. Ela é composta por um sensor CMOS Polarizado da marca Sony que propicia uma resolução de 5,1 MP e uma taxa de quadros de 74 FPS (JAI, 2020).

Figura 4: Ilustração / Desenho técnico da Câmera GO-5100MP-USB



Fonte: Adaptado do fabricante (JAI, 2020)

Dentre as características apresentadas pelo equipamento, e que podem ser observadas em parte na Fig. 4 foram elencadas abaixo as 3 que tiveram maior relevância quanto a seleção desta como uma opção para ser aplicada ao projeto;

- Câmera desenvolvida para meio industrial;
- Suporte à conexão USB;
- Cabos de alimentação e dados intercambiáveis;

2.3.2 Câmera Raspberry Pi High Quality

Sendo uma opção para substituir o modulo de Câmera V2, cuja configuração é menos robusta, a câmera Raspberry Pi High Quality é recomendada pelo fabricante para aplicações em ambiente industrial, além de apresentar uma

resolução de 12.3 megapixels ela dispõe da opção de acoplamento com lentes focais do tipo C (RASPBerry,2020).

Na Fig. 5 abaixo é possível observar as características físicas do equipamento, em destaque sua lente Telefoto Raspberry Pi 16 mm com suporte C (RASPBerry,2020).

Figura 5: Ilustração da Raspberry PiHigh Quality Câmera



Fonte: Raspberry Pi High Quality Câmera - product brief (RASPBerry,2020).

Dentre as características apresentadas pelo equipamento, foram citadas abaixo as 3 que tiveram maior relevância quanto a seleção desta como uma opção para ser aplicada ao projeto;

- Suporte através de cabo flete;
- Capacidade de ser integrada a lentes de alto desempenho;
- Compatibilidade com placas Raspberry;

2.4 DISPOSITIVOS DE ILUMINAÇÃO

Existem diversas opções de periféricos de iluminação, desta forma nesta sessão será apresentada uma opção de equipamento que tem fácil acesso no

mercado e que por suas funcionalidades e características específicas foram selecionados para possível utilização no projeto.

2.4.1 Iluminador Ring Light

Este dispositivo de iluminação pode ser adquirido de em vários tamanhos e normalmente vem acompanhado por tripes para posicionamento do mesmo, segundo um dos sites vendedores deste produto, "É possível controlar as diferentes temperaturas de cores através do controle de ajuste que vai de 3500k a 5500k. Na frente dos leds tem um filtro difusor para suavizar a luz "espalhando" a luz com maior uniformidade (AMAZON,2021).

Abaixo seguem as características que o dispositivo apresenta e que o tornam uma opção a ser considerada para implementação no protótipo;

- Alimentação via cabo USB;
- Opção de mudança de temperatura e intensidade de cor;
- Geometria circular para aplicação próxima da lente da câmera;

Na Figura 6 podemos observar a forma construtiva do disposto, sua geometria circular possibilita aplicação próxima a lente da câmera, sem causar marcas de sombra ou contaminação visual as imagens capturadas.

Figura 6: Iluminador Ring Light de 16 centímetros



Fonte: Adaptado anúncio de venda anel de luz (AMAZON,2021).



3 PROCEDIMENTOS EXPERIMENTAIS

Para análise da eficácia do protótipo proposto, este foi submetido a testes que simulam uma aplicação real de inspeção dimensional, esta por sua vez é tipicamente realizada no processo de controle de qualidade dos perfis de alumínio extrudados.

Nesta sessão serão expostos os passos seguidos para elaboração do hardware e software do equipamento experimental em questão, explicando o motivo de cada material utilizado, tendo como base os resultados que estes propiciaram ao projeto. Quanto ao software desenvolvido, serão descritas as etapas lógicas utilizadas para sua composição.

3.1 HARDWARE DE CAPTURA DE IMAGENS

O objetivo principal do hardware de captura de imagens é propiciar um ambiente extremamente controlado, para que os dados obtidos que não sofressem a influência de condições adversas que o meio pode proporcionar, desta forma pode-se garantir uma maior constância e confiabilidade nos dados que foram adquiridos e que posteriormente foram processados pelo software.

3.1.1 Câmera

A câmera utilizada foi Câmera Fisheye 5Mp OV5647, ela foi escolhida por apresentar características semelhantes a Raspberry Pi High Quality, porém com custo menor e fácil acesso a compra.

Este modelo de câmera em específico acompanha um conjunto de sensores infravermelhos, desta forma foi possível realizar testes mais amplos quando a captura de imagens, sendo as mesmas com alta ou baixa incidência de luminosidade.

Figura 7: Câmera Fisheye 5Mp OV5647



Fonte: Anúncio de venda câmera (MERCADO LIVRE,2021).

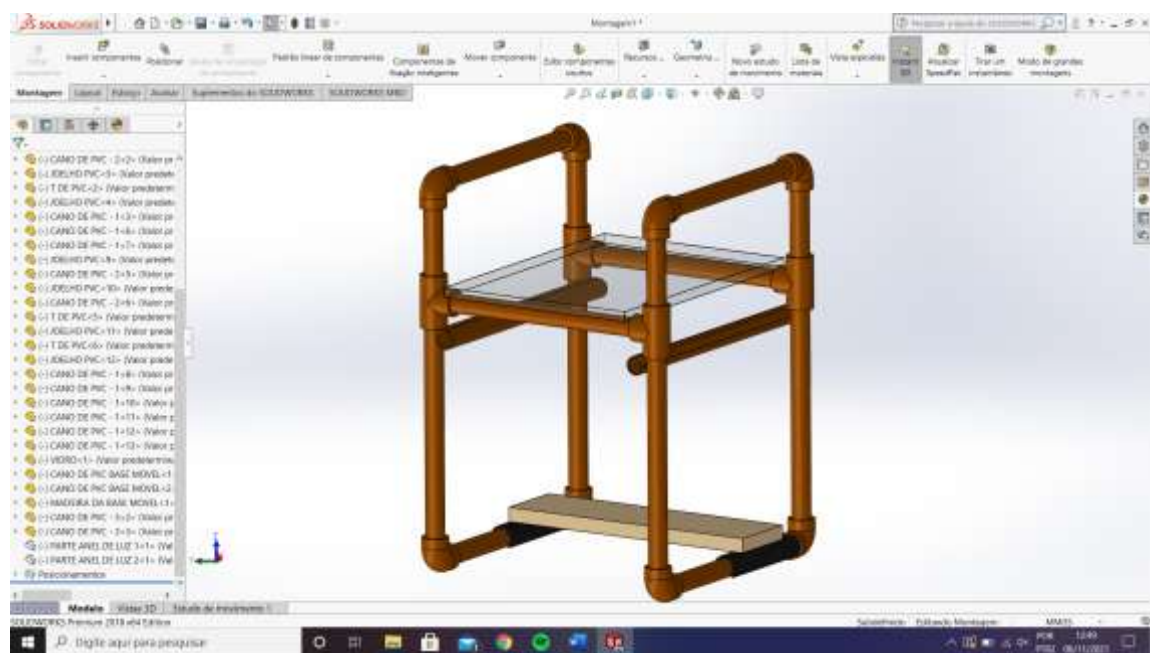
3.1.2 Projeto da Estrutura

Também foi necessária a fabricação de uma estrutura capaz de servir como apoio para os corpos de prova e proteção contra a influência de iluminação externa. Como pré-requisito para compor o projeto desta estrutura foi levado em consideração a utilização de produtos de baixo custo e fácil aquisição, além de atender a necessidade estrutural proposta.

O software de projeto utilizado para a execução dos modelos 3D foi o SOLIDWORKS (Dassault Systèmes, 2018). Através do programa foi possível mensurar o tamanho necessário para o dispositivo, partindo da referência inicial dos corpos de prova, perfis de alumínio tubulares em formato retangular de 50,8 mm de largura, por x 25,4 mm de altura.

Na Figura 8 podemos observar a utilização do SOLIDWORKS para a montagem do projeto 3D, o material principal de sua composição foi o cano de PVC, visto que ele atende aos pré-requisitos necessários.

Figura 8: Projeto da estrutura mecânica.



Fonte: Do autor (2021)

3.1.3 Dispositivo de Iluminação

Após a montagem da estrutura, foram adicionadas a mesma os dispositivos de iluminação Ring Light. Estes equipamentos têm a mesma configuração de intensidade de iluminação e temperatura de luz, contendo três níveis de potência e três diferentes temperaturas da cor da iluminação, desta forma foi possível realizar testes com diferentes condições de incidência de luz para aquisição de dados.

De forma adicional também foi necessária a inclusão de um suporte para posicionamento da câmera, uma régua USB e de um fundo verde na parte superior da estrutura, localizado acima dos corpos de prova, este tem como função destacar os perfis a serem avaliados, evitando a aquisição de dados desnecessários.

Figura 9: Protótipo montado com adição do sistema de iluminação.



Fonte: Do autor (2021)

Na Tabela 1 estão listados os materiais utilizados, descrevendo sua quantidade e especificação.

Tabela 1 – Lista de materiais utilizados.

Especificação	Unidade de Medida	Quantidade
Cano de PVC de diâmetro 20 mm	Metro Linear	3
Conexão em formato T de diâmetro 20 mm	Peça	4
Conexão de 45° de diâmetro 20 mm	Peça	8
Madeira 285x75x15 mm	Peça	1
Lâmina de vidro 300x220x4 mm	Peça	1
Ring Light de 90 mm	Peça	2
Ring Light de 160 mm	Peça	2
Ring Light de 200 mm	Peça	1
Reguá USB 7 portas	Peça	1
Papelão 300x220x2 mm	Peça	1
Cartolina Verde 300x220 mm	Peça	1

Fonte: Do autor (2021)

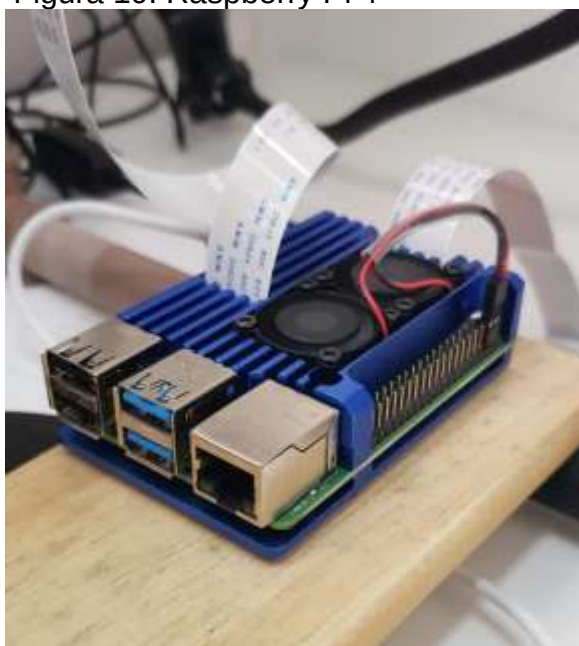
3.2 HARDWARE DE PROCESSAMENTO DE DADOS

Diferente dos componentes estruturais, a parte eletrônica não foi definida tendo como base o objeto que será analisado, e sim através dos tipos de dados que foram obtidos.

Na situação em questão o dado que precisara ser processado serão imagens, desta forma foi preciso selecionar um dispositivo que além de possuir boa capacidade de processamento, tivesse compatibilidade com dispositivos de imagens.

Tendo como base os dados acima foi definido que o equipamento responsável pelo processamento dos dados seria o Raspberry Pi 4. Este dispositivo já foi utilizado em diversas aplicações que compõem sistemas de visão, além de ser uma plataforma preparada para suportar periféricos como câmeras e conexão via WIFI.

Figura 10: Raspberry Pi 4



Fonte: Do autor (2021)

3.3 SOFTWARE DE DESENVOLVIMENTO E PROGRAMAÇÃO

O software de programação escolhido para desenvolvimento do programa de análise das imagens foi o PyCharm (JetBrains,2020), este programa utiliza de



forma nativa a linguagem de programação Python, e serve como uma ferramenta de desenvolvimento integrado a computadores.

O principal motivo pelo qual o programa em questão foi escolhido, se deve ao fato de ao longo da graduação o software ter sido utilizado como ferramenta de ensino dos métodos de manipulação de imagens. Além de ser multiplataforma, podendo ser utilizado em sistemas operacionais Windows e Linux.

Foram desenvolvidos dois programas distintos para execução dos testes, o primeiro deles tem como função principal encontrar a melhor configuração do filtro para separação da imagem de interesse, e foi denominado como PROG01, já o segundo tem como objetivo realizar a análise dos corpos de prova e demonstrar as medições realizadas, ele foi identificado como PRO02.

3.3.1 Desenvolvimento PROG01

Nas imagens abaixo serão expostas em partes a composição do desenvolvimento do PROG01.

Figura 11: Parte 1 do PROG01.

```
1  # Importando bibliotecas:
2  import cv2
3  import numpy as np
4
5  def nothing():
6      pass
7
8  # Criando / denominando janelas de demonstração de imagens:
9  cv2.namedWindow('result', cv2.WINDOW_NORMAL)
10 cv2.namedWindow('original', cv2.WINDOW_NORMAL)
11 cv2.namedWindow('hav', cv2.WINDOW_NORMAL)
12
13 # Definindo variáveis:
14 h_s_v = 100, 100, 100
15 h1_s1_v1 = 255, 255, 255
16
17 # Criando painel configurador:
18 cv2.createTrackbar('h', 'result', 0, 255, nothing)
19 cv2.createTrackbar('s', 'result', 0, 255, nothing)
20 cv2.createTrackbar('v', 'result', 0, 255, nothing)
21 cv2.createTrackbar('h1', 'result', 255, 255, nothing)
22 cv2.createTrackbar('s1', 'result', 255, 255, nothing)
23 cv2.createTrackbar('v1', 'result', 255, 255, nothing)
24 cap = cv2.VideoCapture(0)
25
```

Fonte: Do autor (2021)



Nesta primeira etapa ilustrada na Figura 11 são importadas as bibliotecas necessárias, além da criação das telas de visualização dos resultados e a definição das variáveis e seus valores iniciais. Por fim foram escritos os comandos responsáveis pela criação do painel ajustes do filtro h,s,v.

Figura 12: Parte 2 do PROG01.

```
27 # Loop
28 while(1):
29
30     ret, img = cap.read()
31     cv2.imshow('original', img)
32
33     # Convertendo para HSV:
34     hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
35     cv2.imshow('hsv', hsv)
36
37     # Aplicando valores setados a mascara:
38     h = cv2.getTrackbarPos('h', 'result')
39     s = cv2.getTrackbarPos('s', 'result')
40     v = cv2.getTrackbarPos('v', 'result')
41     h1 = cv2.getTrackbarPos('h1', 'result')
42     s1 = cv2.getTrackbarPos('s1', 'result')
43     v1 = cv2.getTrackbarPos('v1', 'result')
44
45     # Criando as mascaras com valores definidos:
46     lower_blue = np.array([h,s,v])
47     upper_blue = np.array([h1,s1,v1])
48     mask = cv2.inRange(hsv, lower_blue, upper_blue)
49     result = cv2.bitwise_and(img, img, mask=mask)
50     cv2.imshow('result', result)
51
52     k = cv2.waitKey(5) & 0xFF
53     if k == 27:
54         break
55 cap.release()
56 cv2.destroyAllWindows()
```

Fonte: Do autor (2021)

Na segunda e última parte foram escritos os comandos do loop de execução do programa, onde inicialmente é introduzida a imagem original capturada e em seguida a mesma é submetida aos filtros de acordo com os valores configurados. Ao rodarmos o programa são apresentadas três imagens, a original sem ajustes, uma imagem intermediária apenas previamente tratada, e a imagem final já com a máscara aplicada.

3.3.2 Desenvolvimento PROG02

Nas imagens abaixo serão expostas em partes a composição do desenvolvimento do PROG02.



Conforme indicado na Figura 13 que consta abaixo a primeira parte do PROG02 também inicia com os comandos de importação das bibliotecas, em seguida é adicionado o comando que chama a imagem que está sendo capturada, por fim é criado o loop de execução do programa e dentro dele são criadas as máscaras aplicadas a imagem original para destacar o objeto de interesse.

Figura 13: Parte 1 do PROG02.

```
1 # Importando bibliotecas:
2 import cv2
3 import time
4 import numpy as np
5
6 fonte = cv2.FONT_HERSHEY_DUPLEX
7
8
9 cap = cv2.VideoCapture(0)
10 time.sleep(0.1)
11
12 # Loop:
13 while (cap.isOpened()):
14
15     ret, frame = cap.read()
16     (canalAzul, canalVerde, canalVermelho) = cv2.split(frame)
17     zeros = np.zeros(frame.shape[:2], dtype="uint8")
18
19     hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
20
21     # Criando as máscaras:
22     lower_blue = np.array([0, 0, 0])
23     upper_blue = np.array([172, 255, 255])
24
25     mask = cv2.inRange(hsv, lower_blue, upper_blue)
26
27     result = cv2.bitwise_and(frame, frame, mask=mask)
28
29     imggray = cv2.cvtColor(result, cv2.COLOR_BGR2GRAY)
30     ret, thresh = cv2.threshold(imggray, 128, 255, cv2.THRESH_BINARY)
```

Fonte: Do autor (2021)

Na Figura 14 estão descritos os comandos responsáveis pela limpeza interna e externa do objeto de interesse, este passo auxilia no processamento dos dados, pois a partir daqui o software passa a analisar apenas os dados necessários.

Figura 14: Parte 2 do PROG02.

```
31 # Faz a limpeza externa do objeto de interesse:
32 kernel = np.ones((3,3), np.uint8)
33 thresh = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, kernel)
34
35 # Faz a limpeza interna do objeto de interesse:
36 kernel = np.ones((3,3), np.uint8)
37 thresh = cv2.morphologyEx(thresh, cv2.MORPH_CLOSE, kernel)
```

Fonte: Do autor (2021)

Abaixo temos a terceira e última parte do PROG02, nesta etapa foram escritos os comandos condicionais para funcionamento do software, nele também foram definidas algumas variáveis importantes para calibração de proporção de pixels, que está diretamente ligada a capacidade de medição dos corpos de prova.

Por fim foram criados os comandos que escrevem na tela as medições realizadas, só então é gerada a tela final, sendo esta uma combinação da imagem original, somada aos filtros criados e as medidas realizadas.

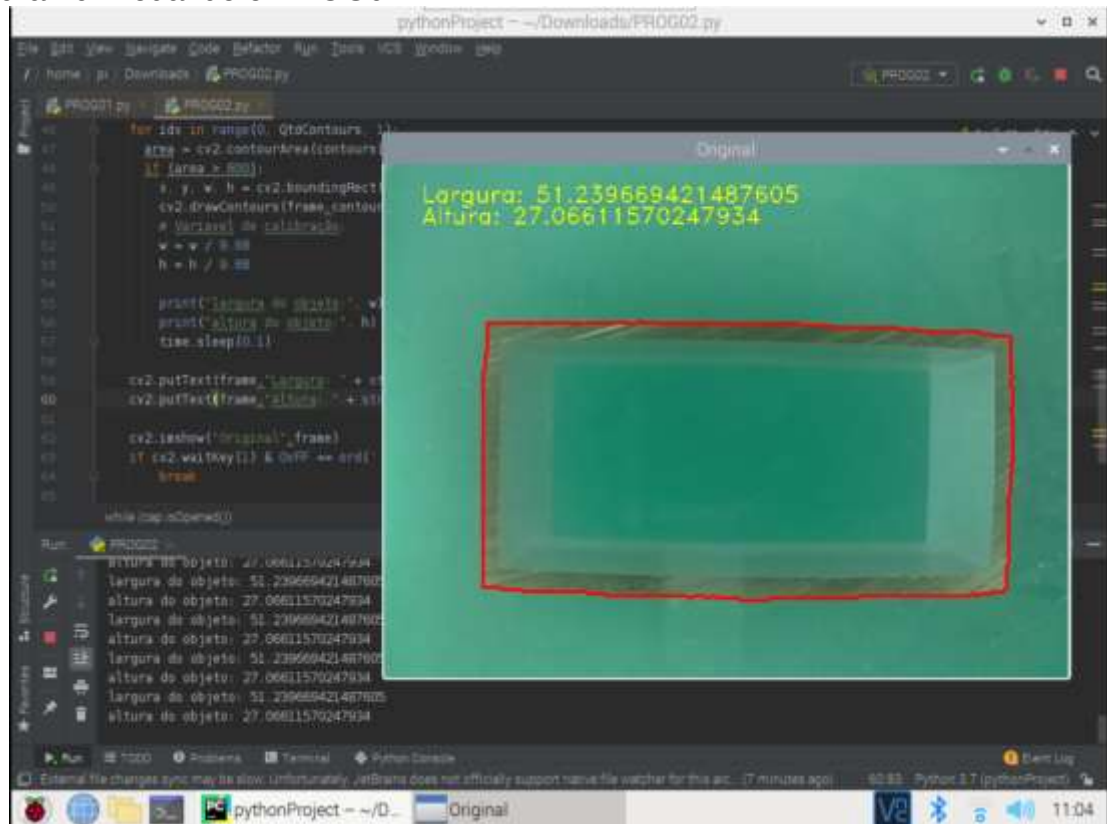
Figura 15: Parte 3 do PROG02.

```
44 # Condições:
45 for idx in range(0, QtdContours, 1):
46     area = cv2.contourArea(contours[idx])
47     if (area > 800):
48         x, y, w, h = cv2.boundingRect(contours[idx])
49         cv2.drawContours(frame, contours, [idx], -50, (0, 0, 255), 2)
50         # Variáveis de calibração:
51         w = w / 9.68
52         h = h / 9.68
53
54         print("largura do objeto:", w)
55         print("altura do objeto:", h)
56         time.sleep(0.1)
57
58         cv2.putText(frame, "largura: " + str(w), (35, 35), fonte, 0.7, (0, 255, 255), 1)
59         cv2.putText(frame, "altura: " + str(h), (35, 55), fonte, 0.7, (0, 255, 255), 1)
60
61         cv2.imshow("Original", frame)
62         if cv2.waitKey(1) & 0xFF == ord(' '):
63             break
64
65 cap.release()
66 cv2.destroyAllWindows()
```

Fonte: Do autor (2021)

Abaixo imagem do programa rodando via Raspberry Pi 4, foi possível observar que ele foi capaz de identificar o objeto de interesse, além de realizar a medição das suas medidas de altura e largura.

Figura 16: Rodando o PROG02.



Fonte: Do autor (2021)

3.4 PREPARAÇÃO DOS CORPOS DE PROVA / TESTES

Para fins de verificação, conforme proposta do trabalho a inspeção dimensional seguiu os parâmetros de amostragem utilizadas pela empresa GRUPO IBRAP, onde a cada 500 kg de material produzido, uma inspeção dimensional é realizada.

O lote em analisado em específico tem 3850 kg, desta forma foram preparadas 8 amostras submetidas a inspeção dimensional. Estas peças foram identificadas por números, conforme exposto na Figura 17.

Figura 17: Amostras de perfis.



Fonte: Do autor (2021)

Devidamente identificas as amostras foram inspecionadas com paquímetro, ele apresenta capacidade máxima de medição de 150 mm, com uma resolução de 0,01 mm, sua exatidão é de 0,03 mm. Nas Tabelas 2 e 3 podemos verificar as medidas obtidas.

Tabela 2 – Medições de largura com paquímetro.

Identificação	Largura (mm)	
	Medida Nominal	Medição Paquímetro
1	50,8	50,87
2		50,97
3		51,04
4		50,94
5		50,87
6		50,87



7		50,87
8		50,87

Fonte: Do autor (2021)

Tabela 3 – Medições de largura com paquímetro.

Identificação	Altura (mm)	
	Medida Nominal	Medição Paquímetro
1	25,4	25,47
2		25,68
3		25,68
4		25,67
5		25,46
6		25,46
7		25,46
8		25,46

Fonte: Do autor (2021)

Após o processo de inspeção manual com o paquímetro as amostras foram submetidas a testes práticos no protótipo, elas foram inspecionadas em ordem crescente de acordo com suas identificações. Foram realizados 4 processos de inspeção idênticos, a fim de medir a assertividade dos dados obtidos no teste.

Antes da realização do primeiro ensaio dimensional, foi realizado o processo de calibração do protótipo através da variável de proporção de pixels, o ajuste foi feito tendo como base as medidas de altura e largura do corpo de prova 1.

4 RESULTADOS E DISCUSSÕES

Neste capítulo serão apresentados os resultados gerados apartir do procedimento experimental descrito neste estudo.

4.1 ANÁLISE CRÍTICA HARDWARE DE IMAGENS E PROCESSAMENTO

Os processos de desenvolvimento do prototipo seguiram iniciariam com o projeto da estrutura mecanica e a aquisição dos equipamentos eletronicos para composição do mesmo.

A câmera que compos o prototipo tem resolução e foco suficientes para realização dos testes, seu conjunto adicional de sensores infra-vermelhos possibilitou



a realização de alguns testes em ambientes de baixa luminosidade. Contudo nas configurações selecionadas não foi possível alcançar um resultado desejado, os níveis de variação das medidas realizadas pelo prototipo não supriram a necessidade exigida pela aplicação.

A estrutura foi montada a partir de canos de PVC e se mostrou eficaz para a aplicação proposta, as conexões escolhidas forneceram rigidez ao conjunto, sem elevar seu peso. Entretanto como se tratam encaixes cíclicos, notou-se certa dificuldade para alinhar paralelamente suas bases, o que eventualmente gera um leve desalinhamento da estrutura.

A solução proposta de apoiar as amostras sobre uma lâmina de vidro, para manter o mesmo distanciamento entre a lente da câmera e a amostra foi positivamente efetiva, através deste método é possível realizar a calibração do prototipo via software, uma vez que a quantidade de pixels na imagem não se altera, devido ao fato da distância ser fixa.

Uma dificuldade encontrada que não foi prevista no projeto, foi a fixação dos Ring Lights à estrutura, inicialmente a ideia seria trabalhar apenas com um destes dispositivos, que seria acoplado ao redor da lente da câmera, entretanto nos testes iniciais, notou-se que isso não seria suficiente, desta forma foi necessário adicionar mais dispositivos de iluminação semelhantes ao redor da estrutura.

Outra melhoria implementada ao longo do desenvolvimento do prototipo, foi a adição do fundo verde na parte superior da estrutura, servindo como contraste para melhorar a capacidade de identificação das amostras dentro da programação. A escolha da cor verde foi feita tendo como base processos realizados dentro da computação gráfica, onde são utilizados os “Chroma Key”.

4.2 ANÁLISE CRÍTICA DESENVOLVIMENTO DO SOFTWARE

A linguagem de programação Python atendeu as expectativas, apresentando soluções práticas para a aplicação proposta, através da Open CV foi possível fazer os diversos tratamentos nas imagens obtidas, conseguindo assim atingir o objetivo de mensurar medidas a partir das imagens.

O software PyCharm rodando diretamente no Raspberry foi capaz de suprir a necessidade de processamento de dados, embora exista certa limitação quanto ao



processamento gráfico, gerando lentidão e pequenos travamentos em algumas etapas do processo.

Quanto ao desenvolvimento dos softwares em si não houve dificuldade, aja visto que eles apresentam estrutura simples, e as mesmas bases com as quais trabalhamos em sala de aula na matéria de Sistemas de Visão Computacional. Uma observação importante a ressaltar é que não existem muitas aplicações usuais deste tipo de inspeção dimensional através dos sistemas de visão, pois ela é uma tecnologia recente, desta forma se torna difícil a criação de programas mais complexos, visto que as bases de comparação para isso ainda não são tão difundidas.

4.4 TESTE PRÁTICO DO PROTÓTIPO

Na Tabela 4 podemos observar os resultados de cada uma das 4 medições de largura das 8 amostras, ficou evidente que existe uma variação na medição, mesmo em condições idênticas.

Tabela 4 – Medições de largura com protótipo.

Medições de Largura Realizadas pelo Protótipo (mm)					
Identificação	Medição 01	Medição 02	Medição 03	Medição 04	Média
1	50,86	50,86	50,86	51,07	50,91
2	51,07	50,86	51,07	51,07	51,02
3	50,86	50,86	51,07	50,66	50,86
4	50,86	50,86	51,07	50,45	50,81
5	50,86	50,86	51,07	51,07	50,97
6	51,48	51,07	51,27	51,07	51,22
7	50,66	50,45	50,86	50,86	50,71
8	50,86	51,07	50,66	51,07	50,92

Fonte: Do autor (2021)

Ainda analisando a Tabela 4, pode-se observar que a amostra 6 na medição 01 apresentou a maior medição, ficando cerca de 22 décimos de milímetro acima da média de medições da amostra. Já a menor medição registrada foi de 50,45 mm, ela pode ser vista na amostra 7 durante a medição 02 e na amostra 4 durante a medição 04.

Na Tabela 5 estão expostos os resultados das medições de altura, seguindo as mesmas métricas de exibição utilizadas para a Tabela 4.



Tabela 5 – Medições de altura com protótipo.

Medições de Altura Realizadas pelo Protótipo (mm)					
Identificação	Medição 01	Medição 02	Medição 02	Medição 04	Média
1	25,49	25,29	25,49	25,68	25,49
2	25,88	25,88	25,88	26,07	25,93
3	25,88	25,88	25,88	25,68	25,83
4	25,88	25,88	25,68	25,68	25,78
5	25,09	25,29	25,49	25,88	25,44
6	26,27	25,68	26,27	25,88	26,03
7	25,49	25,49	25,49	25,49	25,49
8	25,49	25,49	25,49	25,68	25,54

Fonte: Do autor (2021)

Nas medições de altura também é possível verificar que existe uma variação de resultados, o valor mais alto encontrado nesta sessão foi 26,27 mm, esta variação foi constada na amostra 6 nas medições 01 e 02. Quanto ao menos valor encontrado temos 25,09 mm, ele ocorreu na amostra 5 durante a primeira medição e representa uma variação de 0,35 mm abaixo da média de medições realizada na amostra 5.

Após uma avaliação dos resultados de encontrados nas quatro medições feitas em cada amostra, ficou evidente que existe um erro de medição entre elas, os testes foram realizados seguindo os mesmos parâmetros para as 4 tiragens, desta forma o resultado leva a crer que existe uma incerteza na medição realizada pelo protótipo.

Como possíveis causas para esta incerteza podemos citar as variações do ângulo de corte das amostras, o posicionamento das mesmas em relação a câmera, visto que não existe um gabarito e a diferença de tamanho entre as amostras, como é possível verificar na Figura 17.

4.4.1 Comparação de Resultados

Neta sessão serão realizadas comparação entre os resultados obtidos pelas medições do protótipo, em detrimento as medidas obtidas manualmente com o paquímetro.



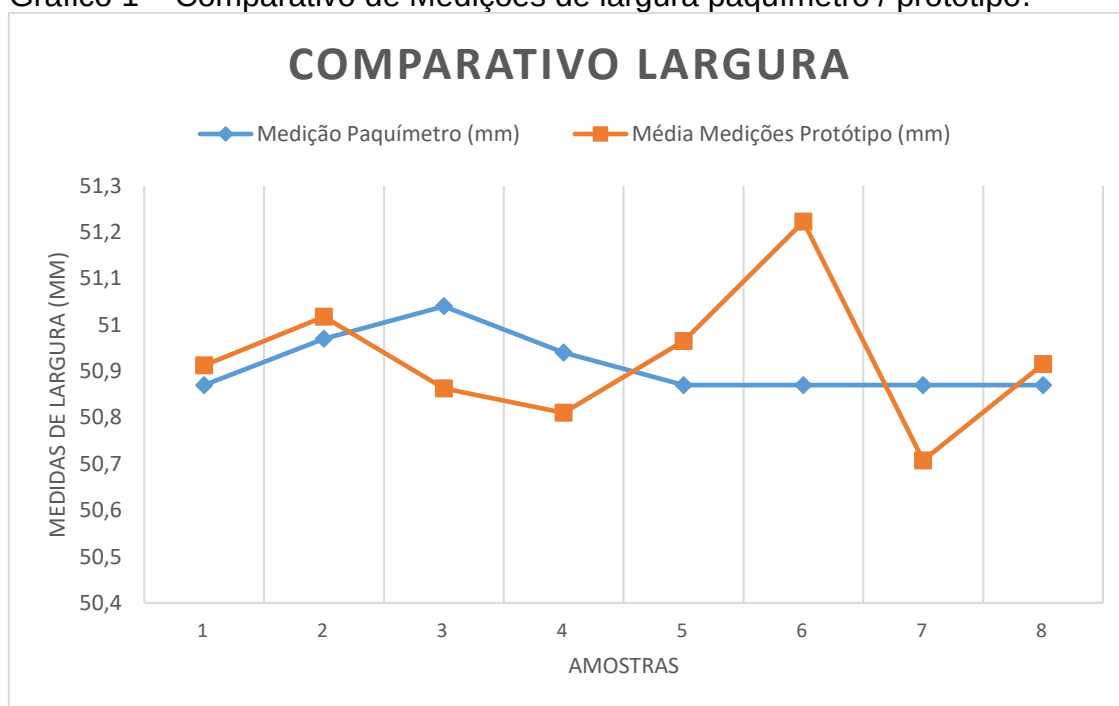
A Tabela 6 faz uma comparação específica entre os resultados de medição apenas da largura, foram utilizados os dados médios das quatro medições do protótipo. A média de erro de medição em milímetros é de 0,13.

Tabela 6 – Comparativo de Medições de largura paquímetro / protótipo.

Identificação	Medição Paquímetro (mm)	Média Medições Protótipo (mm)	Erro de Medição (mm)	Erro de Medição (%)
1	50,87	50,91	0,04	0,08%
2	50,97	51,02	0,05	0,09%
3	51,04	50,86	0,18	0,35%
4	50,94	50,81	0,13	0,26%
5	50,87	50,97	0,09	0,19%
6	50,87	51,22	0,35	0,69%
7	50,87	50,71	0,16	0,32%
8	50,87	50,92	0,05	0,09%

Fonte: Do autor (2021)

Gráfico 1 – Comparativo de Medições de largura paquímetro / protótipo.



Fonte: Do autor (2021)

O Gráfico 1 evidencia uma grande variação de medidas, principalmente na amostra 06, ela apresenta variação 0,69 % em detrimento a medição realizada com paquímetro. As amostras com maior similaridade de resultados são 1,2 e 8.



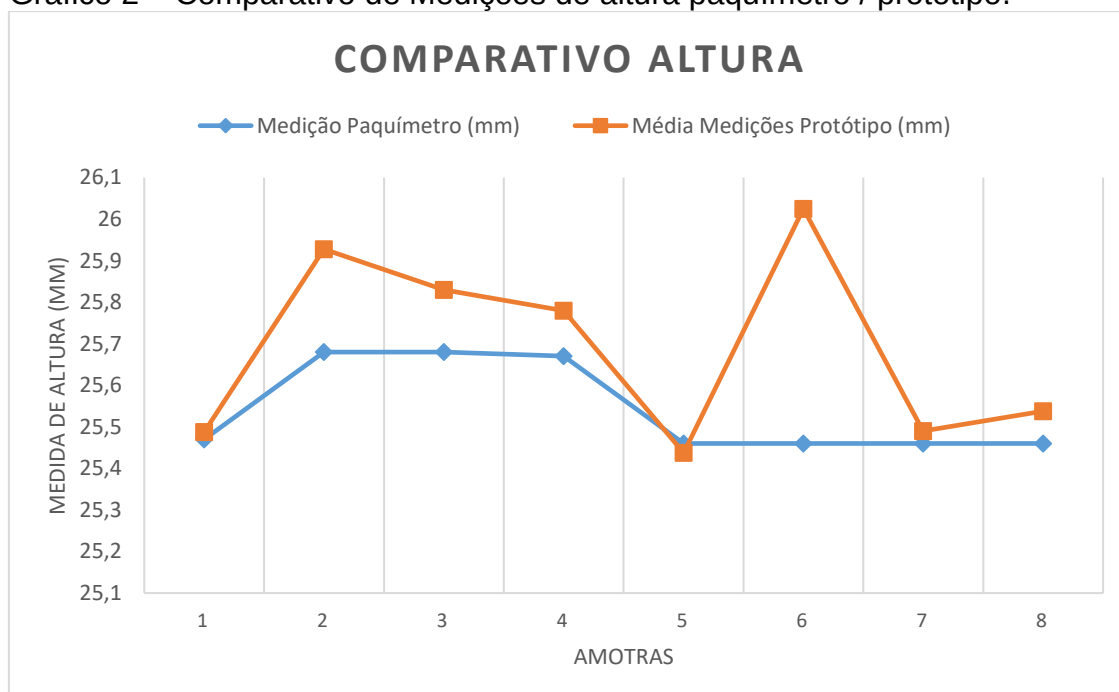
A Tabela 7 trata da comparação entre as medições de altura do paquímetro e protótipo. Nela é possível observar que existem amostras com maior % de variação, chegando aos 2,22 % na 6, em contraponto também podemos observar a medição mais precisa até então, a amostra 1 apresentou erro de medição de apenas 0,02 mm.

Tabela 7 – Comparativo de Medições de altura paquímetro / protótipo.

Identificação	Medição Paquímetro (mm)	Média Medições Protótipo (mm)	Erro de Medição (mm)	Erro de Medição (%)
1	25,47	25,49	0,02	0,07%
2	25,68	25,93	0,25	0,96%
3	25,68	25,83	0,15	0,58%
4	25,67	25,78	0,11	0,43%
5	25,46	25,44	0,02	0,09%
6	25,46	26,03	0,56	2,22%
7	25,46	25,49	0,03	0,12%
8	25,46	25,54	0,08	0,30%

Fonte: Do autor (2021)

Gráfico 2 – Comparativo de Medições de altura paquímetro / protótipo.



Fonte: Do autor (2021)

No Gráfico 2 é possível novamente observar uma variação maior concentrada na amostra 6, são 0,56 mm a mais que a medida realizada pelo



paquímetro, está também é a maior variação encontrada. As medidas mais similares são as das amostras 1,5 e 7.

5 CONCLUSÕES

O meio insdustrial vem a cada ano se tornando mais competitivo, as aplicações automatizadas deixaram de ser um desnumbre do futuro, e passaram a incorporar o dia a dia das fabricas, as novas necessidades de redução de custos e aumento da qualidade dos produtos faz frente a necessidade da introdução de novas tecnologias dentro dos processos já consolidados.

A aplicação de sistemas de visão para inspeção dimensional de perfis de aluminio se mostrou viavel sim. Seguindo a premissa de que precisamos tornar nossos processos mais acertivos, a proposta de eliminar o fator humano realizando a inspeção manual dos perfis extrudados se torna interessante, além da possibilidade redução de custos.

De acordo com os resultados encontrados na elaboração deste trabalho, é preciso dimencionar de forma precisa os equipamentos perifericos que realizarão a captação das imagens e preparação do ambiente de inspeção, visto que as configurações destes dispositivos estarão diretamente ligadas ao nivel de precisão de resultados que poderão serão obtidos.

Outro fator importante é o software, por se atratar uma tecnologia realtivamente recente, ainda não existem grandes acervos de aplicações funcionais para aplicação de inspeção dimensional atravéz de sistema de visão, oque dificulta delimitar com precisão quais são as limitações a nivel de software, aja visto que ainda tem muito a ser desenvolvido dentro deste nicho de aplicação.

Contudo considerando a montagem de um prototipo industrial, com o investimento adequado tanto em equipamentos, quanto no desenvolvimento do softaware, entendo que esta seria uma excelente solução para aplicação dentro do GRUPO IBRAP para realização da inspeção dimensional proposta.



REFERÊNCIAS

ALUMÍNIO.**mdic.gov.br**, 2016. Disponível em: <<http://www.mdic.gov.br/licitacoes-e-contratos/9-assuntos/categ-comercio-exterior/479-metallurgia-e-siderurgia-3>>. Acesso em: 23, novembro de 2020.

ANTONELLO, Ricardo. **Introdução a Visão Computacional com Python e OpenCV**. Santa Catarina: Ricardo Antonello, 2016. 03-06 p.

ASSOCIAÇÃO BRASILEIRA DO ALUMÍNIO – ABAL (Brasil). **ANUÁRIO ESTATÍSTICO ALUMÍNIO 2019**. São Paulo: Associação Brasileira do Alumínio – Abal, 2020. 06 p.

BALLARD, Dana; BROWN, Christopher. **COMPUTER VISION**. Nova York: Prentice Hall, 1982. 12 p.

BARELLI, Felipe. **Introdução à Visão Computacional: Uma abordagem prática com Python e OpenCV**. Brasil: Casa do Código, 2018. 01-252 p.

Completamente atualizado, reprojeto Mais rápido, mais poderoso.**raspberrypi.org**, 2020. Disponível em: <<https://www.raspberrypi.org/products/raspberry-pi-4-model-b/?resellerType=home>>. Acesso em: 23, novembro de 2020.

Luminária Ring Light. **Amazon.com**, 2021. Disponível em: <https://www.amazon.com.br/Iluminador-Light-Misto-3500K-5500K/dp/B07WFC5JY8/ref=asc_df_B07WFC5JY8/?tag=googleshopp00-20&linkCode=df0&hvadid=379726667452&hvpos=&hvnetw=g&hvrnd=16683751540192439513&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=1031576&hvtargid=pla-830850148205&psc=1>

LUTZ, Mark. **Programming Python**. Nova York: O'Reilly Media, 1996. 01-100 p.

Open CV. **opencv.org**, 2020. Disponível em: <<https://opencv.org/about/>>. Acesso em: 23, novembro de 2020.

Panfleto De Aplicações. **mvisia.com**, 2020. Disponível em: <<https://mvisia.com.br/?gclid=Cj0KCQjw5eX7BRDQARIsAMhYLP-SVtlkOOFDYjhbY-ABHTfDYr3JfeRivfcDBbF8b11cLkJIUEXO7o8aAog->>. Acesso em: 23, novembro de 2020.

Série Go: câmeras de varredura de área pequena e versátil. **jai.com**, 2020. Disponível em: <<https://www.jai.com/products/product-lines/go-series-single-sensor-area-scan-small-size>>. Acesso em: 23, novembro de 2020.



Série 38 - câmeras industriais monocromáticas USB 3.1. **theimagingsource.com**, 2020. Disponível em: <<https://www.theimagingsource.com/products/industrial-cameras/usb-3.1-monochrome/>>. Acesso em: 23, novembro de 2020.

[PT-EN] Visão Computacional e Carros Autônomos. **pt.ufrjnautilus.com**, 2020. Disponível em: < <https://pt.ufrjnautilus.com/post/vis%C3%A3o-computacional-e-carros-aut%C3%B4nomos>>. Acesso em: 23, novembro de 2020.