

UNIVERSIDADE FEDERAL DE OURO PRETO ESCOLA DE MINAS COLEGIADO DO CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO - CECAU



IRÃ DIAS DE MAGALHÃES

PROTÓTIPO EMBARCADO PARA A IDENTIFICAÇÃO DE DEFEITOS SUPERFICIAIS EM ESTEIRAS DE TRANSPORTADORES DE CORREIA

MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Ouro Preto, 2021

IRÃ DIAS DE MAGALHÃES

PROTÓTIPO EMBARCADO PARA A IDENTIFICAÇÃO DE DEFEITOS SUPERFICIAIS EM ESTEIRAS DE TRANSPORTADORES DE CORREIA

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro de Controle e Automação.

Orientador: Prof. Andrea Gomes Campos Bianchi, Ph.D. Coorientador: Prof. Bruno Nazário Coelho, Dr.Sc.

> Ouro Preto Escola de Minas – UFOP 2021

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

M188p Magalhaes, Ira Dias De. Protótipo para a identificação de defeitos superficiais em esteiras de correias transportadoras. [manuscrito] / Ira Dias De Magalhaes 2022. 47 f.: il.: color., gráf., tab + Diagramas.	
Orientadora: Profa. Dra. Andrea Bianchi. Coorientador: Prof. Dr. Bruno Coelho. Monografia (Bacharelado). Universidade Federal de Ouro Preto. Escola de Minas. Graduação em Engenharia de Controle e Automação .	
1. Visão Computacional. 2. Correias Transportadoras. 3. Visão computacional - OpenCV. 4. Microcomputadores - Raspberry Pi. I. Bianchi, Andrea. II. Coelho, Bruno. III. Universidade Federal de Ouro Preto. IV. Título.	
CDU 681.5	

Bibliotecário(a) Responsável: Maristela Sanches Lima Mesquita - CRB-1716



MINISTÉRIO DA EDUCAÇÃO UNIVERSIDADE FEDERAL DE OURO PRETO REITORIA ESCOLA DE MINAS DEPARTAMENTO DE ENGENHARIA CONTROLE E AUTOMACAO



FOLHA DE APROVAÇÃO

Irã Dias de Magalhães

Protótipo embarcado para a identificação de defeitos superficiais em esteiras de transportadores de correia

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheiro de Controle e Automação

Aprovada em 13 de Janeiro de 2022

Membros da banca

Doutora – Andrea Gomes Campos Bianchi - Orientador – Universidade Federal de Ouro Preto Doutor – Bruno Nazário Coelho - Coorientador – Universidade Federal de Ouro Preto Doutor – Saul Emanuel Delabrida - Universidade Federal de Ouro Preto Doutor – Pedro Henrique Lopes Silva - Universidade Federal de Ouro Preto

Bruno Nazário Coelho, coorientador do trabalho, e Andrea Gomes Campos Bianchi, orientadora do trabalho, aprovaram a versão final e autorizaram seu depósito na
Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 20/01/2022



Documento assinado eletronicamente por Bruno Nazário Coelho, PROFESSOR DE MAGISTERIO SUPERIOR, em 30/05/2022, às 21:06, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.



A autenticidade deste documento pode ser conferida no site <u>http://sei.ufop.br/sei/controlador_externo.php?</u> <u>acao=documento_conferir&id_orgao_acesso_externo=0</u>, informando o código verificador **0268063** e o código CRC **2B2A7681**.

Referência: Caso responda este documento, indicar expressamente o Processo nº 23109.000484/2022-06

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35400-000 Telefone: 3135591533 - www.ufop.br SEI nº 0268063

AGRADECIMENTOS

Gostaria de agradecer à minha orientadora Andrea Bianchi e ao meu coorientador Bruno Nazário por me guiarem nesta jornada de pesquisa e aprendizado. Gostaria de agradecer também, ao laboratório XR4GOOD por meu fornecer os materiais para realizar este trabalho.

RESUMO

Transportadores de correias são equipamentos de extrema importância para uma planta de mineração, pois possibilitam o transporte contínuo de materiais de um ponto a outro, viabilizando a alimentação e descarga contínua dos equipamentos presentes na planta.

Como todo equipamento, é necessário que sejam feitas inspeções periódicas, a fim de verificar a integridade de seus componentes. Uma dessas inspeções é voltada para a procura de rasgos na esteira do transportador de correia, sendo feita, normalmente, por um funcionário da planta. Como esse procedimento é realizado por um humano, o resultado da detecção de defeitos se torna um produto de uma interpretação subjetiva daquilo que é considerado como um defeito, sendo, portanto, passível de erros.

O presente trabalho aborda a implementação de um protótipo para detecção de defeitos em correias transportadoras, por meio de um algoritmo que identifica falhas com base na curvatura de um feixe de laser incidido na esteira do transportador. Será mostrado que, para plataformas como o Raspberry Pi 3, não é possível obter resultados satisfatórios de detecção. Para isso, serão apresentados a estrutura que aloja os componentes do equipamento, o dimensionamento realizado para garantir a autonomia do protótipo e os resultados dos testes realizados.

Palavras-chaves: Correias Transportadoras. Visão Computacional. OpenCV. Raspberry Pi. Inspeção.

ABSTRACT

Conveyor belts are of extreme importance to a mining plant for allowing the non-stop transportation of materials from one point to another, making possible the continuous feeding of the equipment of the plant.

Like every machine, it's necessary to perform regular inspections to verify the integrity of it's components. One of them is about searching for tearings on the belt, which is done, usually, by a worker of the plant. Since this procedure is performed by a human being, the result of the detection is a product of a subjective interpretation of what is considered to be a defect, being, therefore, error prone.

This paper presents the development of a prototype that detects surface defects on a conveyor belt, through the use of an algorithm that can detect tearings based on the curvature of a laser beam focused on the belt. It will be shown that, for platforms like Raspberry Pi 3, it's not possible to obtain satisfactory detection results. For that, it will be presented the structure that contains the equipment components, the energy consumption sizing made to assure the prototype autonomy and the results of the tests that were made.

Key-words: Conveyor Belt. Computer Vision. OpenCV. Raspberry Pi. Inspection.

LISTA DE ILUSTRAÇÕES

Figura 1 –	Laser incidido em uma dobra	19
Figura 2 –	Conversão da imagem para um sinal unidimensional.	19
Figura 3 –	Matriz resultante.	20
Figura 4 –	Anteparo de papelão utilizado para os testes.	24
Figura 5 –	Média Mensal de Incidência Solar em Belém.	28
Figura 6 –	Número de dias chuvosos x mínimo mensal de incidência solar	28
Figura 7 –	Incidência em dias chuvosos x incidência em dias ensolarados x maior mí-	
	nimo mensal de incidência em Belém.	30
Figura 8 –	Diagrama de regime de funcionamento do equipamento	32
Figura 9 –	Diagrama de conexão do protótipo	32
Figura 10 –	Vista isométrica do protótipo desmontado - a) laser MXD12; b) Raspberry	
	Pi 3 b+; c) compartimento do Raspberry; d) trava do compartimento da	
	câmera; e) câmera XIMEA MQ022CG-CM; f) tampa do compartimento do	
	Raspberry; g) ventoinha 30mm x 30mm x 8mm 5V	33
Figura 11 –	Vista isométrica do protótipo montado.	33
Figura 12 –	Vista isométrica do case do Raspberry Pi - a) saída de ventilação; b) entrada	
	para o conector microUSB; c) entrada para o conector HDMI; d) entrada para	
	o conector de 3,5mm de áudio e vídeo; e) entrada para o conector Ethernet;	
	f) e g) entradas para os conectores USB; h) saída para jumpers	34
Figura 13 –	Vista isométrica da tampa do case - a) saída de ventilação; b) furação para os	
	parafusos da tampa; c) entrada para pino de I/O isolado 20V da câmera; d)	
	entrada para os conectores USB da câmera; e) local de encaixe da ventoinha;	
	f) e g) furo e compartimento do laser, respectivamente; h) furação para os	
	parafusos da ventoinha.	34
Figura 14 –	Vista superior do compartimento da ventoinha	35
Figura 15 –	Dimensões relevantes da tampa do compartimento - a) altura máxima da	
	tampa; b) comprimento máximo da tampa; c) comprimento da base da tampa;	
	d) altura do suporte do laser até à base da tampa	35
Figura 16 –	Distância máxima de funcionamento - a) compartimento da câmera; b) dis-	
	tância ideal entre o protótipo e a esteira; c) esteira	36
Figura 17 –	Vista isométrica do protótipo desmontado - a) laser MXD12; b) suporte do	
	laser; c) câmera (modelo); d) tripé da câmera.	36
Figura 18 –	Vista isométrica do protótipo montado	37
Figura 19 –	Visão Frontal do Suporte do Laser - a) largura da área de encaixe da câmera;	
	b) altura da área de encaixe da câmera; c) altura do braço de suporte do laser.	37

3
3
9
9
9
)
)
)
1
2
3
5
5
5

LISTA DE TABELAS

Tabela 1	-	Consumo de Corrente do Raspberry Pi	26
Tabela 2	_	Testes de performance para diferentes ROI	42

LISTA DE ABREVIATURAS E SIGLAS

ROI	Region of Interest
RPA	Remotely Piloted Aircraft
VANT	Veículo Aéreo Não Tripulado
ACF	Canais de Características Agregados
TC	Transportadores de Correia

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Estado da arte	14
1.2	Justificativa do trabalho	15
1.3	Objetivos	15
1.3.1	Objetivo geral	15
1.3.2	Objetivos específicos	16
2	REVISÃO DA LITERATURA	17
3	DESENVOLVIMENTO	19
3.1	Metodologia	19
3.2	Protótipo Para Testes em Ambiente Real	21
3.2.1	Estrutura do protótipo	21
3.2.2	Dimensionamento energético	22
3.2.3	Algoritmo embarcado	22
3.2.4	Impressão 3D	23
3.3	Protótipo Para Testes em Laboratório	23
3.3.1	Estrutura do protótipo	23
3.3.2	Algoritmo embarcado	23
3.3.3	Testes de detecção	23
4	RESULTADOS	26
4.1	Protótipo Para Testes em Ambiente Real	26
4.1.1	Dimensionamento energético	26
4.1.2	Diagrama de conexão	32
4.1.3	Estrutura do protótipo	33
4.2	Protótipo Para Testes em Laboratório	36
4.2.1	Estrutura do protótipo	36
4.2.2	Testes de detecção	38
4.2.2.1	Testes de luminosidade	38
4.2.2.2	Teste de mapeamento de defeitos	40
4.2.2.3	Teste de performance para diferentes ROI	41
5	DISCUSSÃO	44
6	CONCLUSÃO	47

REFERÊNCIAS		48
-------------	--	----

1 INTRODUÇÃO

1.1 Estado da arte

Em um ambiente de mineração, no caso de esteiras de transportadores de correias (TCs), é normal o surgimento de rasgos, bolhas, desgastes e danos causados por intemperismos. Assim, para evitar possíveis prejuízos provenientes da perda do material transportado ou, até mesmo, danos mais graves ao próprio equipamento utilizado, é necessário um sistema para a detecção de falhas no mesmo.

Devido ao elevado comprimento das esteiras utilizadas nos TCs, a verificação da existência de defeitos é uma tarefa muito trabalhosa para ser realizada por humanos. Além da tarefa possuir uma natureza monótona - que pode contribuir para que a pessoa realizando a inspeção deixe alguma falha passar desapercebida -, a necessidade de se estar próximo do TC para realizar a inspeção pode colocar a vida do trabalhador em risco. Dessa maneira, a implementação de um sistema automatizado para o reconhecimento de falhas aumentaria, de forma considerável, a precisão da inspeção do TC e eliminaria os riscos impostos ao funcionário responsável pela inspeção. Para isso, portanto, seria necessário um algoritmo capaz de identificar os diferentes formatos que os defeitos poderiam apresentar.

Técnicas de visão computacional têm se mostrado muito eficientes na solução de problemas envolvendo detecção de contornos. Em Wolfson et al. (1988), por exemplo, foi desenvolvido um algoritmo que montava quebra-cabeças de 104 peças utilizando visão computacional para detectar o formato das bordas de cada peça.

É interessante notar, entretanto, que não basta apenas desenvolver um algoritmo de visão computacional, mas deve-se, também, assegurar que as condições ideais para que ele funcione estejam presentes, sendo elas uma combinação de iluminação, câmera, placa de captura de imagem, hardware e software (BROSNAN; SUN, 2004). Assim, para satisfazer as condições citadas, muitas vezes, recorre-se ao desenvolvimento de protótipos.

Um elemento importante na criação de um protótipo que contenha um hardware que realize alguma detecção, é a estrutura onde o mesmo estará acoplado. A impressão em 3D seria extremamente apropriada para o desenvolvimento de uma estrutura experimental, pois por décadas ela têm sido usada, principalmente, para a produção de protótipos. Além disso, existe uma enorme variedade de arquivos .stl - terminação que os modelos de impressão possuem - para usos comerciais (VENTOLA, 2014).

Outro fator decisivo é a plataforma de embarque a ser utilizada. Se o protótipo for instalado em um ambiente de fábrica, por exemplo, não haveria como manter um desktop ou notebook conectado ao sistema a todo momento. Dessa forma, seria necessário embarcar o

algoritmo em um computador de placa única - como o Raspberry Pi, por exemplo.

É importante notar, porém, que esses computadores possuem menor capacidade de processamento que um computador convencional. Como algoritmos de visão computacional tendem a consumir grandes quantidades de processamento, seria necessário otimizar o código utilizado para detecção de falhas.

Além disso, dificilmente haverá um sistema que consiga realizar todas as tarefas relacionadas à inspeção de um equipamento. No contexto de um transportador de correia, provavelmente haveria um equipamento destinado para cada tipo de teste a ser realizado. Assim, um protótipo que fosse desevolvido para detecção de rasgos em uma esteira, precisaria ser prático, robusto e econômico, sendo, portanto, otimizado para a tarefa de inspeção de defeitos.

1.2 Justificativa do trabalho

Devido à natureza do processo de mineração, TCs possuem, normalmente, alguns quilômetros de extensão e elevada velocidade de funcionamento. Nesses equipamentos, a verificação da presença de defeitos é feita por um funcionário que se posiciona em um ponto onde é possível averiguar a esteira em movimento e analisa - pelo período necessário para que ela complete uma volta - a presença de rasgos, furos e outros tipos de danos.

Porém, por ser uma atividade monótona, é normal que funcionários desviem sua atenção, mesmo que por meros instantes, da tarefa de procurar por defeitos na esteira. Como os TCs costumam funcionar à uma velocidade elevada, não prestar atenção por alguns segundos resultaria na passagem de algumas dezenas de metros de esteira, o que pode fazer com que o funcionário deixe de enxergar algum defeito presente. Além disso, o tempo necessário para essa atividade pode facilmente superar 30 minutos, dado o comprimento dos TCs. Isso representa um período de tempo que, além de ser muito longo para que alguém consiga se manter concentrado sem interrupções, poderia estar sendo direcionado para atividades mais produtivas.

Torna-se, portanto, interessante o desenvolvimento de um equipamento que consiga realizar a tarefa de detectar falhas nas esteiras de TCs, de forma automatizada. Por meio de um aparelho autônomo, seria possível rodar um algoritmo de detecção de falhas durante uma volta completa da esteira por várias vezes ao dia, possibilitando não só a averiguação dos defeitos sem riscos de deixar algo passar, como também um estudo detalhado da evolução dos mesmos ao longo do tempo.

1.3 Objetivos

1.3.1 Objetivo geral

O presente trabalho tem por objetivo construir um protótipo que possa ser usado em campo e que permita detectar defeitos em esteiras de TCs usando o algoritmo de visão computa-

cional préviamente desenvolvido por Netto (2019).

- 1.3.2 Objetivos específicos
 - Projetar a estrutura de um protótipo para detecção de defeitos em esteiras de TCs.
 - Realizar um dimensionamento energético que permita a autonomia do protótipo em campo.
 - Verificar a viabilidade da implementação de um sistema de identificação de defeitos em um ambiente real.

2 REVISÃO DA LITERATURA

A ideia de desenvolver algum método de buscar defeitos em esteiras de TC já foi anteriormente abordada por outros autores. Junior (2019) propõe a utilização de um drone com uma câmera térimca embarcada para detectar defeitos nos rolamentos dos transportadores. A assinatura térmica dos rolamentos defeituosos seria identificado utilizando o algoritmo de Canais de Características Agregados (ACF) de Piotr Dollar.

Para que o método funcione, entretanto, são necessários quatro pilares: A identificação das regiões de interesse, o processamento morfológico, dados radiométricos e a plataforma de deslocamento via RPA (*Remotely Piloted Aircraft*). Caso um deles esteja ausente, o método se torna inviável.

Assim, se por um lado o método agiliza o processo de detecção de defeitos em rolamentos, por outro, a implementação pode ser dita como muito trabalhosa, uma vez considerados os prérequisitos já citados.

Nascimento (2018) propõe um arquitetura para construção de plataformas de inspeção que auxiliam no monitoramento das condições dos rolos dos TCs. Assim, são testados protótipos que se baseiam em Veículos Aéreos Não Tripulados (VANTs), que foram projetados para monitorar os rolos por meio de uma câmera, e é feita a construção de um ambiente virutal de um porto para aplicações baseadas em VANTs em um simulador.

É mostrado que os VANTs, além de poderem ser controlados manualmente via joysticks, podem também receber comandos para executar tarefas específicas de forma autônoma, como decolar ou aterrisar, por exemplo. A adição de um ambiente virtual também auxilia no treinamento dos operadores, uma vez que há o risco do VANT colidir durante o vôo, caso o piloto não seja experiente.

Porém, é relatado que houveram problemas de performance com a transmissão de vídeos com resolução maior que 1024x768 tanto no simulador - pois o computador precisa renderizar tanto o ambiente simulado quanto a imagem da câmera da simulação -, quanto nos testes práticos, por causa do protocolo de transmissão usado para realizar o *streaming* das imagens filmadas pela câmera do protótipo.

Por último, Netto (2019), em sua dissertação, desenvolve um algoritmo de detecção de defeitos superficiais, que utiliza a incidência de um laser para determinar os contornos das falhas presentes na esteira e os projeta em uma matriz de pontos. Sendo, portanto, possível mapear o formato do defeito em duas dimensões. O algoritmo apresenta uma taxa elevada de detecção de defeitos, sendo mais eficiente que seres humanos - apresentando uma precisão de 96% do algoritmo contra 70% dos humanos. Porém, todos os testes foram feitos em ambientes controlados, não, portanto, havendo nenhum experimento que tenha sido feito em campo para

comprovar a real viabilidade do método.

Assim, esse trabalho visa a criação de um protótipo que utiliza a metodologia descrita por Netto (2019) para detecção de defeitos em esteiras, almejando, portanto, testar a viabilidade do algoritmo quando embarcado em um computador de placa única.

3 DESENVOLVIMENTO

3.1 Metodologia

O presente trabalho visa montar um protótipo de um equipamento de detecção de falhas em esteiras de TCs, que utiliza o algoritmo proposto por Netto (2019). Para isso, foi necessário projetar uma estrutura de proteção para o hardware, que também fosse capaz de fixar os periféricos utilizados (como o laser e a câmera) em suas devidas posições. Além disso, foi necessário fazer um dimensionamento do consumo de energia do aparelho, uma vez que será utilizado um sistema de alimentação por paineis solares.

O funcionamento do algoritmo se dá da seguinte maneira: Um feixe de laser é incidido no defeito que se deseja mapear. A imagem do feixe é, então, transformada em um sinal unidimensional - uma vez que o processamento de uma imagem, isto é, um sinal bidimensional, é muito mais custoso -, sendo, então, mapeados pontos de alta curvatura presentes no sinal que são, então, passados para uma matriz de mapeamento, descrevendo o formato do defeito detectado. As Figuras 1, 2 e 3, retiradas da dissertação, ilustram o processo descrito.



Figura 1 – Laser incidido em uma dobra.



Figura 2 – Conversão da imagem para um sinal unidimensional.

Porém, para que o algortimo funcione de maneira ideal, é necessário que certas condições de posicionamento da câmera e do laser sejam cumpridas. São elas:

- Que o plano da câmera esteja paralelo ao objeto da detecção
- Que a incidência do laser esteja a 30° do ponto central do plano da câmera
- Que o laser incidida paralelo ao eixo X do plano da câmera



Figura 3 – Matriz resultante.

Além disso, caso o protótipo desenvolvido seja usado em um ambiente real, ele precisa ser autônomo. Para isso, é necessário que ele possua uma fonte que forneça energia de forma constante e, consequentemente, que o equipamento não consuma mais do que é gerado. Assim, faz-se necessário um dimensionamento energético do equipamento.

A fim de cumprir esses requisitos, foram feitos dois protótipos: Um para ser testado em uma planta de mineração - a ser referido como **Protótipo Para Testes em Ambiente Real** - e o outro para realização de testes em um ambiente controlado - a ser referido como **Protótipo Para Testes em Laboratório** -, visando testar a viabilidade do algoritmo quando embarcado em um computador de placa única, da estrutura projetada e do dimensionamento energético realizado.

Porém, como consequência da pandemia que aconteceu em 2020, a realização de testes em uma planta se tornou inviável. Dessa forma, foram apenas realizados os testes com o protótipo para testes em laboratório. Ainda sim, este trabalho apresenta o dimensionamento, estrutura e funcionamento de ambos os protótipos.

3.2 Protótipo Para Testes em Ambiente Real

O desenvolvimento desse protótipo foi voltado para a utilização em uma planta de mineração, possuindo todo o seu dimensionamento voltado para isso. A sua viabilidade foi determinada por testes indiretos, uma vez que, como já citado, por causa da pandemia que começou em 2020, não foi possível obter acesso à nenhuma planta de mineração.

O protótipo seria desenvolvido utilizando os seguintes componentes:

- Computador de placa única Raspberry Pi 3 B+
- Laser MXD1230 6mm 650nm 5mW Red Line Laser Module, 20mA
- Painel solar de 30W, 655mm x 350mm
- Bateria 12V, 5 Ah, No Break APC SMS
- Módulo mppt 30A USB 12/24V PWM LCD
- Câmera XIMEA MQ022CG-CM
- Cooler fan de 30mm x 30mm x 8mm, 5V, 150mA

A escolha do Raspberry foi feita com base no custo benefício, familiaridade e documentação do mesmo - outras opções ou possuiam preço muito elevado, ou, por serem muito novas, apresentavam pouca documentação, o que dificultaria a realização do projeto. Os demais componentes foram resultado do dimensionamento do consumo de energia aliado a ideia de tornar o projeto o mais barato possível sem comprometer o funcionamento do mesmo.

3.2.1 Estrutura do protótipo

A ideia por trás do planejamento da estrutura era fazer um compartimento que protegesse o Raspberry Pi, fornecesse ventilação suficiente e pudesse alocar os periféricos necessários. A estrutura do protótipo foi projetada. também, levando em conta as caracterítcas de funcionamento do algoritmo proposto por Netto (2019). Assim, ela precisaria comportar o mesmo modelo de câmera utilizado durante os testes descritos em seu trabalho e garantir que a incidência do laser fosse de 30 graus em relação ao plano da esteira, uma vez que é mostrado que nesse ângulo são obtidos os melhores resultados de reconhecimento dos defeitos.

O projeto da estrutura foi feito utilizando a ferramenta de CAD, SolidWorks 2020¹. O CAD do protótipo final foi feito com componentes cuja modelagem é resultado do presente trabalho, salvo os modelos do Raspberry Pi e da ventoinha de 30mm x 30mm, que, por questões de conveniência, foram obtidos no site GrabCAD².

¹ https://www.solidworks.com/media/solidworks-2020-improved-performance

² https://grabcad.com/

3.2.2 Dimensionamento energético

O dimensionamento energético foi feito com o objetivo de garantir a maior autonomia possível do protótipo. Foi, portanto, proposto um regime de funcionamento para o equipamento, de forma que, por dia, fosse feito um número aceitável de análises do transportador, ao mesmo tempo em que a energia fornecida pela bateria e placa fotovoltaica fosse suficiente para manter o aparelho funcionando em épocas de chuva, onde as incidências solares são abaixo do ideal.

Para determinar esse regime, foram levadas em conta as seguintes variáveis:

- Potência fornecida pela placa fotovoltaica ($P_{fotovoltaica}$)
- Potência fornecida pela bateria (P_{bateria})
- Potência consumida pelo Raspberry Pi com o algoritmo rodando e todos os periféricos ligados (P_{ligado})
- Potência consumida pelo Raspberry Pi com o algoritmo em *standby* e todos os periféricos desligados ($P_{desligado}$)
- Incidência solar média em dias ensolarados em Belém, Pará (I_{sol})
- Incidência solar média em dias chuvosos/nublados em Belém, Pará (I_{chuva})

As potências citadas foram obtidas através da multiplicação da corrente consumida pelo Raspberry Pi informada no site do fabricante ³, equanto as incidências solares foram obtidas dos sites WeatherSpark ⁴, LG Energy ⁵ e da dissertação por Ribamar (2017).

A região escolhida para coletar os dados de incidência foi a cidade de Belém, no Pará. O motivo da escolha é o fato de lá estar presente o complexo de mineração da Vale, o S11D, que seria utilizado para a realização dos testes com o protótipo desenvolvido.

3.2.3 Algoritmo embarcado

O algoritmo foi embarcado no Raspberry Pi por meio da linguagem Python. Ela foi escolhida por sua facilidade de implementação, grande variedade de aplicações, ampla documentação e porque o algoritmo em que este trabalho se baseia foi, convenientemente, escrito em Python. O código de detecção de falhas é executado em loop, alternando entre um estado onde é feita a detecção de defeitos nas esteiras e um estado onde o algoritmo é pausado e os periféricos conectados ao Raspberry são desligados, a fim de reduzir o consumo de energia, resultando, portanto, em uma maior autonomia do equipamento no período noturno.

³ https://www.raspberrypi.org/

⁴ https://weatherspark.com/

⁵ https://www.lgenergy.com.au/

3.2.4 Impressão 3D

A impressora 3D Creality, modelo Ender 3⁶, foi escolhida para realizar a impressão das peças da estrutura do protótipo por ser uma opção popular em termos de custo-benefício, além de ser o modelo presente no laboratório que seria usado para realização de testes.

3.3 Protótipo Para Testes em Laboratório

Esse protótipo foi desenvolvido para testes voltados especificamente para o funcionamento do algoritmo embarcado.

Ele foi construído utilizando os seguintes componentes:

- Computador de placa única Raspberry Pi 3 B+
- Laser MXD1230 6mm 650nm 5mW Red Line Laser Module, 20mA
- Câmera LS-F36-1080P(XM)

A ausência de um painel fotovoltáico e bateria se justificam pelo fato do raspberry estar conectado ao notebook, não havendo, portanto, necessidade da realização de um dimensionamento energético.

3.3.1 Estrutura do protótipo

O projeto da estrutura foi feito utilizando a ferramenta de CAD, SolidWorks 2020. Para as suas dimensões, foi usada como base a câmera Câmera XIMEA MQ022CG-CM, que estava disponível no momento. A estrutura foi feita com o objetivo de ser o mais simples possível e garantir a realização dos testes de detecção do algoritmo, obedecendo às condições necessárias estabelecidas por Netto (2019) em sua dissertação.

3.3.2 Algoritmo embarcado

O algoritmo do protótipo em escala reduzida é mais simples que o do protótipo de chão de fábrica, uma vez que, como citado anteriormente, não há necessidade de dimensionamento energético e nem racionamento de energia, já que o raspberry Pi se encontrava conectado à uma fonte de energia constante. Seu funcionamento se resume em executar o algoritmo de detecção em loop, enquanto mostra em uma janela separada a matriz de detecção do algoritmo.

3.3.3 Testes de detecção

Como não havia nenhuma esteira de transportador de correia acessível, foi feita uma simulação de um defeito usando um anteparo de papelão, conforme mostra a Figura 4. O papelão

⁶ https://www.crealitystore.com.br/impressora-3d-ender-3-fdm-creality-3d

foi escolhido por ser barato, fácil de encontrar e ser relativamente rígido em comparação a outros materiais sendo, portanto, fácil de manusear.



Figura 4 – Anteparo de papelão utilizado para os testes.

Os testes realizados no algoritmo de detecção foram:

- Testes de luminosidade
- Teste de mapeamneto de defeitos
- Teste de performance para diferentes Regiões de Interesse (ROI)

O teste de luminosidade consiste em três etapas. A primeira é avaliar de que modo a variação da luz do ambiente interfere na detecção de defeitos. O protótipo é apontado para uma parede onde incida luz e, então, é analisado como os pontos de detecção gerados pelo algoritmo se comportam.

A segunda etapa avalia se o algoritmo consegue detectar ou não as bordas do rasgo feito no papelão dependendo da luminosidade no ambiente. Nela, o papelão é posicionado com um rasgo alinhado com o campo de visão da câmera, de forma que o laser do protótipo incida na falha. Já a terceira etapa avalia se a adição de uma região de interesse afeta a detecção das bordas em ambientes claros e escuros.

O teste de mapeamento de defeitos avalia o mapeamento feito pela matriz de defeitos daquilo que foi detectado pelo algoritmo, servindo para averiguar se o formato obtido dos

defeitos é fiel com ao formato real. O teste de performance para diferentes ROI consiste em apontar o protótipo para uma parede e contar o número de iterações do algoritmo por segundo para cada ROI testada. Esse teste é feito usando uma ROI como base e, a partir das dimensões dela, variando ora apenas a altura e ora apenas a largura da mesma.

4 RESULTADOS

A princípio, foi iniciada a implementação do protótipo para testes em ambiente real. Foi desenvolvida a estrutura e o dimensionamento energético do mesmo. Porém, com o evento da pandemia de 2020, não foi possível ter acesso a uma parte dos periféricos necessários para a montagem do mesmo e, também, não seria possível acessar nenhuma planta de mineração por questões de segurança sanitária. Assim, o foco da realização de testes foi o protótipo para testes em laboratório.

Dessa forma, essa seção se divide em duas subseções: A seção **4.1 Protótipo Para Testes em Ambiente Real** descreve tanto a estrutura desenvolvida quando o regime de funcionamento obtido para o primeiro protótipo. Já a segunda seção, **4.2 Protótipo Para Testes em Laboratório** descreve a estrutura e os testes realizados com o segundo protótipo.

4.1 Protótipo Para Testes em Ambiente Real

4.1.1 Dimensionamento energético

Para que fosse possível determinar um regime de funcionamento do algortimo que maximizasse a autonomia do protótipo, foi necessário mapear, inicialmente, o consumo de cada periférico quando ligado ao Raspberry Pi. A Tabela 1 mostra o consumo de corrente do Raspberry Pi 3 B+ em função dos periféricos ligados.

Estado Normal	CPU sob uso intenso	HDMI	USB	Wi-fi	Ventoinha	Laser	Consumo
							780mA
							605mA
							585mA
							365mA
							312mA
							162mA
							142mA
		-					
Periféricos	Consumo individual			L	egenda		
HDMI	20mA						
USB	220mA				Ativado		
Wi-fi	53mA						
Ventoinha	150mA				Desativado		
Laser	20mA						

Tabela 1 – Consumo de Corrente do Raspberry Pi.

Considerando os dados da tabela e como a aplicação do protótipo não necessita de conexão com a internet, a potência necessária P_{ligado} com o algoritmo de detecção funcionando

e todos os periféricos, exceto o Wi-fi, ligados, seria, portanto:

$$P_{ligado} = 5V \times 727mA = 3,635W$$

Caso as funções do openCV estivessem em pausa e os periféricos desligados, a potência seria:

$$Pd = 5V \times 142mA = 0,710W$$

Considerando um valor médio de velocidade obtido de esteira em (CAETANO, 2019) e o comprimento de TC citado na revista da vale que descreve o S11D¹, é possível afirmar que o tempo médio necessário para a passagem completa de um esteira ($T_{esteira}$) é:

$$T_{esteira} = \frac{9500m}{\frac{3m}{s}} \approx 3167s$$

sendo, assim, $T_{esteira} \approx 53$ minutos.

A bateria inicial foi dimensionada como sendo uma de 12V e 5Ah. Ela forneceria:

$$P_{bateria} = 12V \times 5Ah = 60Wh$$

Levando em conta que o ácido da bateria não seria drenado até menos de 50% (para evitar problemas com a vida útil e fornecimento de energia da mesma), a potência fornecida cai para:

$$P_{bateria} = 30Wh$$

Se considerar uma eficiência de 80%:

1

$$Pb = 30Wh \times 0, 8 = 24Wh$$

O painel solar inicialmente escolhido foi um de 30W, com área (A) de:

$$A = 655mm \times 350mm = 229250mm^2 = 0,22925m^2$$

Supondo uma eficiência de 80% do painel, em dias ensolarados, sob condições ideais:

$$P_{fotovoltaica} = 30Wh \times 0, 8 = 24Wh$$

http://www.vale.com/PT/initiatives/innovation/s11d/Documents/s11d-2016/assets/doc/revista-vale-s11d.pdf

Nem sempre haverá garantia de que o painel irá fornecer a potência máxima, pois isso depende dos horários em que o sol estará visível e os níveis de incidência solar ao longo do dia. Para garantir que o dimensionamento da alimentação do equipamento fosse adequado, foi utilizada a dissertação de Ribamar (2017), a fim de saber os níveis de incidência da região onde o protótipo seria inicialmente testado.

O gráfico da Figura 5 mostra a média mensal de incidência solar com base na inclinação da placa fotovoltáica descrita por Ribamar (2017). Como espera-se montar um sistema com maior confiabilidade possível, foi escolhida a inclinação que fornece o maior mínimo mensal de incidência.



Figura 5 – Média Mensal de Incidência Solar em Belém.

O site WeatherSpark² fornece o número de dias chuvosos por ano em Belém. A fim de validar os dados obtidos dessa fonte, foi feita uma comparação entre eles e os dados obtidos da dissertação de Ribamar (2017), como mostrado no gráfico da Figura 6.



Figura 6 – Número de dias chuvosos x mínimo mensal de incidência solar.

² https://pt.weatherspark.com/y/30136/Clima-caracter%C3%ADstico-em-Bel%C3%A9m-Brasil-durante-o-ano

É possível perceber que há, de fato, uma relação direta entre o aumento de dias chuvosos e a diminuição de incidência solar - como deveria ser. Dessa forma, fica evidente que os dados retirados do site WeatherSpark correspondem com a realidade.

A fim de se obter um dimensionamento que permita o sistema funcionar no maior número de situações possível, foi necessário descobrir qual a incidência em dias chuvosos e dias ensolarados. Segundo o site da empresa LG, que fabrica equipamentos fotovoltáicos, a incidência em dias nublados pode ficar entre 25% a 45% do que seria possível obter sob condições ideais. Em dias chuvosos, a incidência pode cair para até 20%.

Por conveniência de cálculo, dias nublados e chuvosos foram considerados como sendo iguais e, portanto, a incidência resultante foi representada por uma média entre o valor dos dias chuvosos e o valor máximo dos dias nublados:

$$\frac{20\% + 45\%}{2} = 32,5\%$$

É possível encontrar a incidência mensal utilizando a seguinte fórmula, que corresponde à uma média ponderada da incidência solar, utilizando o número de dias chuvosos e ensolarados como peso:

$$\frac{I_{chuva} \times D_{chuva} + I_{sol} \times D_{sol}}{D_{mes}} = I_{mensal}$$

Onde:

- *I*_{sol} = Incidência em dias ensolarados
- D_{chuva} = Número de dias chuvosos
- D_{sol} = Número de dias ensolarados
- D_{mes} = Número de dias no mês
- I_{mensal} = Incidência mensal

Considerando que dias ensolarados possuem uma incidência ideal (I_{sol}) e que a incidência em dias chuvosos (I_{chuva}) resulta em 32,5% dessa incidência, tem-se que:

$$I_{chuva} = I_{sol} \times 0,325$$

Tem-se, portanto:

$$\frac{I_{chuva} \times D_{chuva} + \frac{I_{chuva}}{0.325} \times D_{sol}}{D_{mes}} = I_{mensal}$$

O que implica em:

$$I_{chuva} = \frac{D_{mes} \times I_{mensal}}{D_{chuva} + \frac{D_{sol}}{0.325}}$$

O gráfico da Figura 7 mostra uma comparação entre as incidências médias em dias chuvosos, ensolarados e o maior mínimo mensal de incidência, considerando uma área de incidência igual à da placa fotovoltaica escolhida ($655mm \times 350mm$)



Figura 7 – Incidência em dias chuvosos x incidência em dias ensolarados x maior mínimo mensal de incidência em Belém.

Com base na potência fornecida pela bateria, é possível descobrir quantas horas o protótipo poderia funcionar com o algoritmo rodando, sem necessitar de carga:

$$T_{ligado} = \frac{P_{bateria}}{P_{ligado}} = \frac{24Wh}{3,635W} = 6,6h$$

Enquanto o tempo de funcionamento com os periféricos desligados seria:

$$T_{desligado} = \frac{P_{bateria}}{P_{desligado}} = \frac{24Wh}{0,710W} = 33,8h$$

Considerando que, segundo o site WeatherSpark, o tempo médio de sol ao longo do ano em Belém (T_{sol}) é de aproximadamente 12 horas, então o tempo em que o sol não está presente é, em média 12 horas. Portanto, seria possível deixar o código rodando durante quase todo o período de sol sem carga. Porém, com a presença de um painel solar conforme o dimensionado, a potência gerada ($P_{fotovoltaica}$) seria mais que suficiente para recuperar a carga consumida da bateria(P_{ligado}). Essa afirmação é válida mesmo em dias nublados/chuvosos pois, como foi mostrado no gráfico da Figura 7, a incidência por hora em dias de chuva (a menor de todas as incidências consideradas) é maior que 50W. No intervalo de tempo onde não há sol, por não haver possibilidade de uso das placas fotovoltaicas, seria necessário distribuir os 24W da bateria ao longo de 12 horas. Durante esse período, é possível manter o protótipo em *standby*, com todos os periféricos desligados, e ainda sobraria potência o suficiente para rodar o algoritmo, sem nenhuma pausa, por pouco mais que 4 horas:

$$(T_{desligado} - T_{noite}) \times P_{desligado} = (33, 47 - 12)h \times 0,710W = 15,48Wh$$

 $\frac{15,48Wh}{3,635W} = 4,25h \approx 255min$

O que seria suficiente para rodar o algoritmo aproximadamente 5 vezes:

$$Te = 50minutos \Longrightarrow \frac{255min}{50min} = 5,11 \ voltas$$

Considerando $5,08 \approx 5$, tem-se que:

$$\frac{5 \ voltas}{12h} = \frac{1 \ volta}{2, 4h} = \frac{1 \ volta}{144min}$$

Considerando que a cada 144 minutos, 50 minutos seriam do algoritmo executando, sobrariam:

$$(144 - 50)min = 94min$$

Como 94 minutos seriam do equipamento em *standby*, o consumo final após 12 horas seria:

$$5 \times (\frac{94min \times 0,710W + 50min \times 3,635W}{60min}) = 20,70W$$

O que implica que, durante as 12 horas sem sol, seria completamente viável que de cada 144 minutos, 50 fossem do algoritmo funcionando.

A fim de haverem intervalos iguais entre cada execução do algoritmo, seriam necessárias paradas de:

$$\frac{(144-50)min}{2} = 47minutos$$

O que implica que, o regime de operação seria, portanto de 12 horas de funcionamento sem pausa, seguido de 12 horas compostas por 50 minutos de funcionamento intercalados com pausas iniciais e finais de 47 minutos e pausas intermediárias de 144 minutos, como mostra a figura 8.



Figura 8 – Diagrama de regime de funcionamento do equipamento.

4.1.2 Diagrama de conexão

A Figura 9 mostra um diagrama de como os periféricos, bateria, módulo mppt e painel fotovoltáico estariam conectados ao Raspberry Pi. É interessante notar que a vantagem do modelo do módulo mppt utilizado é que ele possui portas USB, o que facilita a alimentação do Raspberry Pi, eliminando a necessidade de um conversor buck para converter a tensão de saída do módulo para um valor compatível com a entrada do Raspberry Pi.



Figura 9 – Diagrama de conexão do protótipo.

4.1.3 Estrutura do protótipo

A Figura 10 mostra a estrutura em vista isométrica, realçando todas as peças que compõe o protótipo. É possível observar os locais onde a ventoinha, laser, câmera e raspberry serão fixados, além das furações e os respectivos parafusos. Já a Figura 11 mostra um vista isométrica do protótipo montado.



Figura 10 – Vista isométrica do protótipo desmontado - a) laser MXD12; b) Raspberry Pi 3 b+;
c) compartimento do Raspberry; d) trava do compartimento da câmera; e) câmera XIMEA MQ022CG-CM; f) tampa do compartimento do Raspberry; g) ventoinha 30mm x 30mm x 8mm 5V.



Figura 11 – Vista isométrica do protótipo montado.

O compartimento do Raspberry foi desenvolvido com o intuito de reduzir a exposição do equipamento a possíveis intempéries. Como mostra a Figura 12, ele possui entradas para todos os conectores princiapais do Raspberry, além de uma saída para os jumpers que serão conectados. Além disso, ele possui uma saída de ar, a fim de tornar mais eficaz a ventilação realizada pela ventoinha.



Figura 12 – Vista isométrica do case do Raspberry Pi - a) saída de ventilação; b) entrada para o conector microUSB; c) entrada para o conector HDMI; d) entrada para o conector de 3,5mm de áudio e vídeo; e) entrada para o conector Ethernet; f) e g) entradas para os conectores USB; h) saída para jumpers.

A tampa do compartimento é onde estão alojados o laser e a câmera do protótipo. Como é possível ver na Figura 13, o compartimento da câmera possui saídas de ar. Isso se deve ao fato de que a câmera possui uma temperatura operacional de 50 graus sendo, portanto, desejável que ela não esteja em um lugar sem ventilação.



Figura 13 – Vista isométrica da tampa do case - a) saída de ventilação; b) furação para os parafusos da tampa; c) entrada para pino de I/O isolado 20V da câmera; d) entrada para os conectores USB da câmera; e) local de encaixe da ventoinha; f) e g) furo e compartimento do laser, respectivamente; h) furação para os parafusos da ventoinha.

O posicionamento do compartimento da ventoinha foi escolhido com base na localização do processador do Raspberry, uma vez que é sabido que este componente é o que gera mais calor. Como é possível ver na Figura 14, o compartimento se encontra logo acima do processador.



Figura 14 – Vista superior do compartimento da ventoinha.

A figura 15 mostra algumas dimensões da tampa do compartimento. O comprimento máximo da tampa é de 201,17mm, enquanto a altura máxima é de 41mm. Isso se deve ao fato de que o protótipo foi projetado para ser impresso em uma impressora 3D Creality, modelo Ender 3, cuja área de impressão é de 220mm x 220mm x 250mm, sendo portanto, um fator limitante para as dimensões do protótipo.



Figura 15 – Dimensões relevantes da tampa do compartimento - a) altura máxima da tampa; b) comprimento máximo da tampa; c) comprimento da base da tampa; d) altura do suporte do laser até à base da tampa.

O comprimento máximo da peça afeta, ainda, a distância de funcionamento do protótipo, uma vez que, por ser necessária uma angulação de 30 graus entre o plano da câmera e o laser, a distância ideal da esteira, para essa estrutura, fica limitada à 279,13mm, como mostra a Figura 16.



Figura 16 – Distância máxima de funcionamento - a) compartimento da câmera; b) distância ideal entre o protótipo e a esteira; c) esteira.

Esse problema poderia ser contornado mudando a direção da impressão, porém isso tornaria a estrutura frágil, uma vez que as camadas impressas estariam perpendiculares ao comprimento do suporte do laser e paralelas às furações dos parafusos, o que faria a peça se partir facilmente.

4.2 Protótipo Para Testes em Laboratório

4.2.1 Estrutura do protótipo

A Figura 17 mostra a estrutura do protótipo em escala reduzida em vista isométrica, realçando todas as peças que compõe o protótipo. Nela é possível ver como se dá o encaixe da câmera no suporte do laser.



Figura 17 – Vista isométrica do protótipo desmontado - a) laser MXD12; b) suporte do laser; c) câmera (modelo); d) tripé da câmera.



Já a Figura 18 mostra um vista isométrica do protótipo montado.

Figura 18 – Vista isométrica do protótipo montado.

A Figura 19 mostra algumas dimensões do protótipo em escala reduzida. Nota-se que ele é menor do que o protótipo de chão de fábrica e mais simples, também.



Figura 19 – Visão Frontal do Suporte do Laser - a) largura da área de encaixe da câmera; b) altura da área de encaixe da câmera; c) altura do braço de suporte do laser.

Nota-se, também, que o alcance deste protótipo é menor (141,24mm), como mostra a Figura 20



Figura 20 – Vista Superior do Suporte do Laser - a) suporte do laser; b) distância ideal entre o protótipo e a esteira; c) esteira.

4.2.2 Testes de detecção

4.2.2.1 Testes de luminosidade

Neste primeiro teste, fizemos a aquisição da imagem do laser em uma parede branca com luminosidade ambiente. As Figuras 21 e 22 mostram o resultado do teste de luminosidade. É possível perceber que em um ambiente claro, os pontos de detecção ficam deslocados do laser, enquanto em um ambiente escuro, esses mesmos pontos se algomeram em cima do laser, como esperado.



Figura 21 – Teste de luminosidade - ambiente claro.



Figura 22 - Teste de luminosidade - ambiente escuro.

As Figuras 23 e 24 mostram o resultado do teste de detecção de bordas. Em um ambiente claro, sem a presença de uma ROI, possivelmente por causa de ruídos gerados pela iluminação, o algoritmo falha em detectar as bordas do rasgo no papelão. Por outro lado, em um ambiente escuro, a detecção acontece como esperado.



Figura 23 – Teste de detecção de bordas - ambiente claro.



Figura 24 - Teste de detecção de bordas - ambiente escuro.

As Figuras 25 e 26 mostram o resultado do teste de detecção de bordas com ROI. Dessa vez, em ambos os casos, ocorre a detecção da borda como esperado.



Figura 25 – Teste de detecção de bordas com ROI - ambiente claro.



Figura 26 – Teste de detecção de bordas com ROI - ambiente escuro.

4.2.2.2 Teste de mapeamento de defeitos

A Figura 27 mostra a matriz de defeitos resultante do teste de detecção de defeitos.



Figura 27 – Matriz de defeitos resultante do teste.

Em seguida, foi realizado um teste para entender como o aumento da velocidade da passagem do defeito pelo laser pode afetar a detecção do mesmo e, consequentemente, o seu mapeamento. Este teste consistiu em deslocar o anteparo à velocidades gradualmente maiores, sendo realizado um movimento oscilatório - para cima e para baixo - com o anteparo. Dessa forma, a matriz de detecção iria traçar vários defeitos subsequentes, cada um resultado de uma velocidade diferente. A Figura 28 apresenta o resultado da detecção dos pontos de alta curvatura obtidos pelo algoritmo.



Figura 28 – Matriz de defeito resultante do anteparo ser deslocado com velocidades gradualmente maiores.

4.2.2.3 Teste de performance para diferentes ROI

Este teste consistiu na realização de um *benchmark* para saber a performance do algoritmo com base em diferentes tamanhos de ROI. A ideia era descobrir qual fator deve ser priorizado na hora de determinar uma ROI ideal, se é o formato ou a quantidade total de pixels da mesma.

Assim, partindo de um valor considerado ótimo - chamado valor base - , obtido por tentativa e erro, foram realizadas duas etapas: Incialmente variando a altura da ROI, enquanto se mantinha a largura fixa e, posteriormente, variando a largura enquanto se mantinha a altura fixa. Os resultados do teste de performance são mostrados na Tabela 2.

	Largura	Altura	Quantidade de pixels	Número de iterações/minuto	Número de iterações/segundo
Valor base	375	25	9375	689	11.48333333
	375	27	10125	688	11.46666667
	375	31	11625	684	11.4
Variando a altura da imagem	375	35	13125	650	10.83333333
	375	39	14625	628	10.46666667
	375	43	16125	605	10.08333333
	400	23	9200	626	10.43333333
	425	23	9775	599	9.98333333
Variando a largura da imagem	450	23	10350	567	9.45
	475	23	10925	529	8.816666667
	500	23	11500	471	7.85

Tabela 2 – Testes de performance para diferentes ROI.

O gráfico da Figura 29 mostra que, embora aumentar apenas a altura da ROI resulte em um maior número de pixels por imagem, o número de iterações por segundo não varia de forma significativa, indo de 11,48 para 10 iterações por segundo.



Figura 29 – Gráfico Iterações/segundo em função da variação da altura da imagem.

Por outro lado, de acordo com o gráfico da Figura 30, o aumento da largura da ROI, embora resulte em um menor número de pixels que o outro teste, resulta em uma queda significativa no número de iterações, caindo de 11,48 para 7,85 iterações por segundo.



Figura 30 – Gráfico Iterações/segundo em função da variação da largura da imagem.

5 DISCUSSÃO

Nesta monografia, a proposta era construir um protótipo e avaliar o seu desempenho em uma planta real de mineração. Entretanto, a pandemia impediu que fossem realizados testes em um ambiente real. Porém, os testes realizados com o protótipo de laboratório forneceram resultados muito pertinentes quanto à implementação do algoritmo em uma plataforma portátil.

Após a realização dos testes, foi possível perceber que, se por um lado, o dimensionamento energético mostrou um resultado positivo, quando analisado o número de vezes que seria possível executar a varredura diariamente, por outro lado, embora seja possível fazer o algoritmo funcionar em uma plataforma como o Raspberry Pi, existem certas limitações voltadas para a questão de taxa de processamento e luminosidade.

Como mostrado nos testes da seção 4.2.2.1, a luminosidade pode afetar de maneira negativa a detecção de defeitos. Ainda que a adição de ROIs resolva essa questão, isso deve ser feito de forma que toda a seção transversal do objeto esteja em foco. Esse fator, por sua vez, é afetado pela distância do protótipo em relação ao objeto, que varia de acordo com o comprimento do suporte do laser, o que é uma questão estrutural, que não pode ser facilmente alterada.

A taxa de atualização do algoritmo é de extrema importância para a detecção adequada dos defeitos de uma esteira, uma vez que um baixo número de iterações não apenas afeta a resolução do defeito, como também muda o seu formato. Isso pode ser notado nos resultados do teste de mapeamento de defeitos na seção 4.2.2.2.

Como foi visto na seção 4.2.2.3, a dimensão da ROI utilizada está diretamente relacionada com a performance do algoritmo. ROIs com maior largura demandam mais poder de processamento do que ROIs com maior altura. Como é explicado em Netto 2019, quando o laser é detectado pelo algoritmo, ele é convertido para um sinal, isto é, a imagem do laser, que é um sinal bidimensional, é convertida para um sinal unidimensional. Durante essa transformação, a altura da imagem é convertida para a amplitude do sinal, enquanto a largura é convertida para a resolução do mesmo. Assim, quanto maior a largura da ROI, maior a resolução do sinal resultante e, quanto maior a resolução de um sinal, maior a quantidade de informações que ele carrega.

O diagrama da Figura 31 exemplifica, por meio de uma situação hipotética, como seria a detecção caso o algoritmo funcionasse com o maior número de iterações por segundo obtido durante os testes de ROI. A Figura 32 exemplifica como seria a detecção, na mesma situação hipotética, à uma taxa de atualização equiavalente a de uma câmera de baixo orçamento - 24fps. E a Figura 33 exemplifica como seria a detecção à uma taxa de atualização equiavalente a da câmera sugerida por (NETTO, 2019) em sua dissertação - 170fps.

Embora seja possível, como foi demonstrado, mudar o tamanho da imagem capturada por meio de ROIs, o número de iterações continua abaixo do ideal no melhor dos casos. Portanto,



Figura 31 – Diagrama exemplificando o resultado de uma detecção feita a 11,4 fps.



Figura 32 – Diagrama exemplificando o resultado de uma detecção feita a 24 fps.

não seria possível implementar o algoritmo utilizando um Raspberry Pi, versão 3 B+, de forma satisfatória.

Assim, a solução mais viável, seria, provavelmente otimizar o algoritmo para que ele não consuma tanto poder de processamento - talvez migrando para uma linguagem mais leve que o Python, por exemplo.

Outra questão importante é a distância ótima de funcionamento do protótipo. Por uma



Figura 33 – Diagrama exemplificando o resultado de uma detecção feita a 170 fps.

questão trigonométrica, o ângulo de incidência que Netto (2019) sugere em sua dissertação como sendo o melhor para detecção, não pode ser obtido a menos que o laser se encontre à uma distância exata da esteira. E como seu trabalho mostra, a variação do ângulo acarreta numa diminuição na precisão da detecção de defeitos.

Portanto, a menos que se garanta que não haverá necessidade de mudar o protótipo de posição durante o seu uso, a detecção de defeitos será prejudicada.

Além disso, é importante ressaltar que faz-se necessária uma calibração antes da utilização do protótipo, a fim de alinhar o laser com o centro da imagem da câmera, para garantir que o laser esteja incidindo na angulação correta.

Outro ponto importante é o fator do dimensionamento energético precisar ser refeito toda vez que o local de implementação for alterado. Como os valores obtidos são referentes às incidências solares em Belém, não há garantia de que eles sejam válidos para outras regiões do país.

Por último, é necessário considerar o fator do aquecimento do protótipo. Embora tenha sido utilizada uma ventoinha própria para o Rapsberry, por boa prática, a constante utilização de um algoritmo de processamento de imagem poderia gerar muito mais calor do que a ventoinha consegue aguentar, causando o desligamento preventivo da plataforma, a fim de prevenir danos, o que iria interromper a detecção de defeitos e tornaria necessário que o código fosse reiniciado.

6 CONCLUSÃO E TRABALHOS FUTUROS

A utilização de algoritmos para detecção de padrões é um campo que tem recebido grande atenção nos útlimos anos, graças à evolução do poder de processamento gráfico dos computadores. Por meio deste trabalho, é possível perceber que a utilização de técnicas de visão computacional para detecção de defeitos é um método com grande potêncial.

Porém, existem ainda ressalvas quanto à praticidade da implementação de um sistema como esse em ambientes reais. Seja na questão estrutural do protótipo ou na performance do algoritmo, é inegável que existem vários desafios que precisam ser enfrentados antes que seja possível desenvolver um produto final que seja portátil, autônomo e funcional.

Dessa maneira, algumas sugestões para trabalhos futuros seriam: otimizar o algoritmo proposto por Netto (2019) a fim de tornar sua utilização viável em plataformas como o Raspberry Pi, abordar questões de ruídos em ambientes luminosos, implementar uma estrutura física que não possua as limitações de posicionamento descritas, estudar a questão do aquecimento do protótipo durante longos períodos de utilização, e finalmente, testar o protótipo para chão de fábrica em um transportador de correia utilizando alguma plataforma de embarque cuja arquitetura seja voltada para o processamento de imagens, possivelmente melhorando a performance do protótipo.

REFERÊNCIAS

BROSNAN, T.; SUN, D.-W. Improving quality inspection of food products by computer vision—a review. *Journal of Food Engineering*, 2004. v. 61, p. 3–16, 01 2004. Citado na página 14.

CAETANO, M. *Cálculo e Selecção de Correias Transportadoras*. 2019. Disponível em: <https://www.ctborracha.com/borracha-sintese-historica/aplicacoes/correias-transportadoras-/calculo-e-seleccao-de-correias-transportadoras/>. Citado na página 27.

FILHO, J. R. M. Análise de viabilidade econômica do mercado de microgeração fotovoltaica on grid no estado do maranhão. 2017. 2017. Citado 2 vezes nas páginas 22 e 28.

JUNIOR, J. R. d. C. Processamento digital de imagens para a identificação automática de falhas em rolos dos transportadores de correias. 2019. 2019. Citado na página 17.

NASCIMENTO, R. S. d. Inspeção de transportadores de correia: Arquitetura integrada para uma plataforma de inspeção com uso de vants. 2018. 2018. Citado na página 17.

NETTO, G. G. Método de visão computacional baseado em laser para monitoramento de defeitos em correias transportadoras. 2019. 2019. Citado 9 vezes nas páginas 16, 17, 18, 19, 21, 23, 44, 46 e 47.

VENTOLA, C. L. Medical applications for 3d printing: current and projected uses. *Pharmacy and Therapeutics*, 2014. MediMedia, USA, v. 39, n. 10, p. 704, 2014. Citado na página 14.

WOLFSON, H. et al. Solving jigsaw puzzles by computer. Annals of Operations Research, 1988. v. 12, n. 1, p. 51–64, Dec 1988. ISSN 1572-9338. Disponível em: https://doi.org/10.1007/BF02186360. Citado na página 14.