

Pedro Henrique Quintino Garcia

**Sistema embarcado de baixo custo projetado
para dimensionar ovos de galinha através de
visão computacional**

UBERLÂNDIA

2021

Pedro Henrique Quintino Garcia

**Sistema embarcado de baixo custo projetado para
dimensionar ovos de galinha através de visão
computacional**

Trabalho de Conclusão de Curso da Engenharia de Controle e Automação da Universidade Federal de Uberlândia - UFU - Câmpus Santa Mônica, como requisito para a obtenção do título de Graduação em Engenharia de Controle e Automação

Universidade Federal de Uberlândia – UFU

Faculdade de Engenharia Elétrica

Orientador: Prof. Dr. Mauricio Cunha Escarpinati

Coorientador: MSc. Jocival Dias Júnior

UBERLÂNDIA

2021

A todos familiares, sobretudo ao meu avô Eurípedes Barsanulfo (in memoriam), que me guiou aos bons princípios através de sua religião.

Agradecimentos

Sozinho não se chega a lugar algum!

É com este pensamento que agradeço a todos aqueles que, de uma maneira ou de outra, colaboraram para a conclusão deste trabalho.

Em primeiro lugar a Deus, pela vida, saúde e pelas pessoas que estão ao meu lado.

Aos meus orientadores, professor Dr. Maurício Cunha Escarpinati e Jocival Dias Júnior, pelo apoio, paciência e ensinamentos, que me ajudaram a concluir este trabalho. Agradeço também aos meus professores que durante muito tempo me ensinaram e me mostraram o quanto estudar é recompensador.

À minha família, por sua capacidade de acreditar e investir em mim. A Camila Gonçalves que há anos me incentiva e me apoia a buscar o progresso. Obrigado pelo carinho, a paciência e por sua capacidade de me trazer paz na correria de cada dia.

Aos meus amigos e colegas, que me ajudaram a superar momentos difíceis e acreditaram nessa conquista.

A todos aqueles que de alguma forma estiveram próximo de mim, fazendo esta vida valer cada vez mais a pena.

*“Vitória nem sempre é sobre vencer,
mas sobre melhorar.”,
(Caio Bottura)*

Resumo

A utilização de novas tecnologias para automação de processos no setor agroindustrial brasileiro vem sendo amplamente abordada nos últimos anos, em especial a visão computacional. Na avicultura de postura, a automação aliada ao processamento digital de imagens produz técnicas eficazes para aumentar a qualidade na produção de ovos de galinha, garantir agilidade e reduzir custos. Neste trabalho, um sistema embarcado de baixo custo foi desenvolvido para dimensionamento de ovos de galinha utilizando técnicas de visão computacional. Um protótipo capaz de fornecer um ambiente ideal - livre de ruídos externos - foi projetado para captura das fotografias utilizadas no processamento digital de imagens, com autonomia para 4 horas de trabalho sem interrupção de energia. O algoritmo de visão computacional foi elaborado através da linguagem de programação Python e a biblioteca de código aberto OpenCV. O usuário possui a opção de interagir diretamente com o sistema embarcado através de um *display touchscreen* ou realizar o acesso de forma remota. Da mesma forma, os resultados podem ser coletados localmente na aplicação ou enviados para outra plataforma através do sistema de comunicação *Bluetooth*, desenvolvido como requisito funcional para construção do sistema. O índice de acerto para as medições obtidas através do sistema foi de 91% para o eixo transversal e 89% para o eixo longitudinal do ovo.

Palavras-chave: Sistema embarcado. Visão Computacional. Ovo de galinha. Python. Raspberry Pi. Bluetooth.

Abstract

The use of new technologies for process automation in the Brazilian agro-industrial sector has been widely discussed in recent years, especially computer vision. In laying poultry, automation combined with digital image processing produces effective techniques to increase the quality of chicken egg production, ensure agility and reduce costs. In this work, a low-cost embedded system was developed for sizing chicken eggs using computer vision techniques. A prototype capable of providing an ideal environment - free from external noise - was designed to capture the photographs used in digital image processing, with autonomy for 4 hours of work without power interruption. The computer vision algorithm was developed using the Python programming language and the OpenCV open source library. The user has the option of interacting directly with the embedded system through a touchscreen display or accessing it remotely. Likewise, the results can be collected locally in the application or sent to another platform through the Bluetooth communication system, developed as a functional requirement for the construction of the system. The success rate for the measurements obtained through the system was 91% for the transverse axis and 89% for the longitudinal axis of the egg.

Keywords: Embedded System. Computer Vision. Chicken Egg. Python. Raspberry Pi. Bluetooth.

Lista de ilustrações

Figura 1 – Exemplo das medidas obtidas da casca do ovo.	14
Figura 2 – Ovo quebrado e reconhecimento de área quebrada	20
Figura 3 – Regras Fuzzy no sistema de classificação de ovos	20
Figura 4 – Exemplo de saída do sistema projetado	21
Figura 5 – FIS desenvolvido na ferramenta MATLAB Simulink para classificação de ovos	21
Figura 6 – Dispositivo construído para determinação dos parâmetros externos do ovo	22
Figura 7 – Passo a passo do fluxo de processamento de imagem	23
Figura 8 – Estimativa do raio de curvatura das pontas dos ovos	23
Figura 9 – Exemplo da classificação de ovos baseado nos valores do índice da forma e do índice do tamanho	24
Figura 10 – Tela inicial do aplicativo	25
Figura 11 – Resultado do processamento	26
Figura 12 – Representação 3D do protótipo	27
Figura 13 – Perspectiva isométrica do protótipo	28
Figura 14 – Protótipo sob o ponto de vista do operador	29
Figura 15 – Raspberry PI 4 Model B	31
Figura 16 – Indicador de carga remanescente da UPS	32
Figura 17 – Integração entre Raspberry Pi 4 e UPS	33
Figura 18 – Display Raspberry Pi Touchscreen 7”	34
Figura 19 – Câmera Raspberry Pi v2 8MP	35
Figura 20 – Linha de LED RGB x10 WS2813 Grove	36
Figura 21 – Representação genérica do sistema embarcado desenvolvido	38
Figura 22 – Parâmetros armazenados no banco de dados	39
Figura 23 – Fluxo de informações entre os sistemas e o banco de dados	40
Figura 24 – Fluxo de trabalho da aplicação de visão computacional	40
Figura 25 – Exemplo de processamento	42
Figura 26 – Página inicial da aplicação	43
Figura 27 – Página responsável pela função de captura das imagens dos ovos	43
Figura 28 – Exemplo de resultados alocados no banco de dados	44
Figura 29 – Configurações do sistema embarcado	44
Figura 30 – Página com a função de processamento em lote	45
Figura 31 – Página responsável pela exportação dos resultados, imagens e limpeza do banco de dados	46
Figura 32 – Exemplo de dados enviados via <i>Bluetooth</i> após processamento	47
Figura 33 – Exemplo de ovo marrom anormal processado pelo sistema	48

Lista de tabelas

Tabela 1 – Comparação estatística entre os dois métodos de medição em milímetros 49

Lista de abreviaturas e siglas

A	<i>Ampere - Unidade de medida da corrente elétrica</i>
API	<i>Application Programming Interface</i>
ARM	<i>Advanced RISC Machine</i>
BLE	<i>Bluetooth Low Energy</i>
cm	<i>Centímetro - Unidade de medida de comprimento</i>
CSI	<i>Camera Serial Interface</i>
DC	<i>Direct Current</i>
DDR4	<i>Double Data Rate Fourth Generation</i>
DSI	<i>Display Serial Interface</i>
fps	<i>frames per second</i>
g	<i>grama - Unidade de medida de massa</i>
GB	<i>gigabyte</i>
GHz	<i>GigaHertz - Unidade de medida de frequência</i>
GPIO	<i>General Purpose Input/Output</i>
HDMI	<i>High-Definition Multimedia Interface</i>
I2C	<i>Inter-Integrated Circuit</i>
IHM	<i>Interface Homem-Máquina</i>
LAN	<i>Local Area Network</i>
LCD	<i>Display de Cristal Líquido</i>
LED	<i>Light-Emitting Diode</i>
mAh	<i>miliAmpere-hora - Unidade de medida da capacidade de baterias</i>
MDF	<i>Medium Density Fiberboard</i>
mm	<i>Milímetro - Unidade de medida de distância</i>

MP	<i>Megapixels</i>
OSH	<i>OpenSource Hardware</i>
PDI	<i>Processamento Digital de Imagens</i>
PoE	<i>Power over Ethernet</i>
RAM	<i>Random Access Memory</i>
RGB	<i>Red Green Blue</i>
SBC	<i>Single-Board Computer</i>
SMD	<i>Surface Mounted Device</i>
TFT	<i>Thin Film Transistor</i>
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
UPS	<i>Uninterruptible Power Supply</i>
USB	<i>Universal Serial Bus</i>
V	<i>Volts - Unidade de medida de tensão elétrica</i>
W	<i>Watts - Unidade de medida de potência</i>
°C	<i>grau Celsius - Unidade de medida de temperatura</i>

Sumário

1	INTRODUÇÃO	13
1.1	Justificativa	13
1.2	Objetivo	14
1.3	Organização do Trabalho	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Processamento Digital de Imagens	16
2.2	Algoritmo de Visão Computacional	17
2.2.1	Python & OpenCV	17
2.3	Interface Homem-Máquina	18
3	TRABALHOS CORRELATOS	19
3.1	An expert egg grading system based on machine vision and artificial intelligence techniques	19
3.2	Assessing exterior egg quality indicators using machine vision	22
3.3	Egg Evaluation: Ovoscofia via Smartphone	24
3.4	Comentários	25
4	MATERIAIS E MÉTODOS	27
4.1	Construção do protótipo	27
4.1.1	Raspberry Pi 4 Model B	29
4.1.2	Fonte de alimentação ininterrupta para Raspberry Pi 4B	31
4.1.3	Display Raspberry Pi Touchscreen 7"	33
4.1.4	Câmera Raspberry Pi v2 8MP	34
4.1.5	Linha de LED RGB x10 WS2813 Grove	36
4.1.6	Desenvolvimento do sistema embarcado	37
4.2	Desenvolvimento do Software	38
4.2.1	Banco de dados	39
4.2.2	Desenvolvimento da API	39
4.2.3	Algoritmo de Visão Computacional	39
4.2.4	Desenvolvimento da Aplicação	42
4.2.5	Sistema de comunicação <i>Bluetooth</i>	46
5	RESULTADOS E DISCUSSÕES	48
6	CONCLUSÃO E TRABALHOS FUTUROS	50

REFERÊNCIAS	51
--------------------	-----------

1 Introdução

A indústria avícola nas últimas décadas vem se aprimorando e passa a se destacar no setor agroindustrial brasileiro, sofrendo modificações voltadas para a necessidade do consumidor e para a qualidade do mercado. O aumento da produção avícola de postura, ramo da avicultura voltado para a produção de ovos, requer uma redução das perdas que ocorrem durante o processo de produção. Além disso, o aumento das exigências impostas pelo mercado nacional e internacional, exige um maior controle de qualidade dos produtos, a fim de evitar problemas como contaminação e fornecimento de alimentos de baixa qualidade.

Devido a esses fatores, grandes investimentos têm sido realizados pelas empresas produtoras de ovos em pesquisas para desenvolver métodos de produção com o menor custo possível. Uma das alternativas para a redução dos custos de produção é a automatização dos processos e o investimento em melhoramento genético das poedeiras, visando reduzir a quantidade de produtos com qualquer tipo de problema e maximizar a capacidade de postura das aves.

De acordo com Lima (2018), a qualidade do ovo pode estar relacionada à sua parte interna ou externa. A qualidade interna do ovo se relaciona, em geral, com a qualidade do albúmen. Já a qualidade exterior está relacionada à casca do ovo e os principais pontos observados são a cor, a espessura, a porosidade, a resistência e a sujeira da casca. Esses aspectos externos da qualidade dos ovos estão relacionados com a função principal da casca: a proteção do interior do ovo contra danos físicos e invasões bacterianas. A avaliação da qualidade do ovo comercial passa pela análise da qualidade da casca do ovo, pois produtos com casca intacta podem permanecer por mais tempo nas prateleiras de estabelecimentos sem deixar de atender as exigências sanitárias e do consumidor final. Por causa das dificuldades de conferir os ovos de forma manual no processamento de triagem, a automação produz técnicas eficazes, fornecendo agilidade, precisão e aumentando a qualidade do ovo.

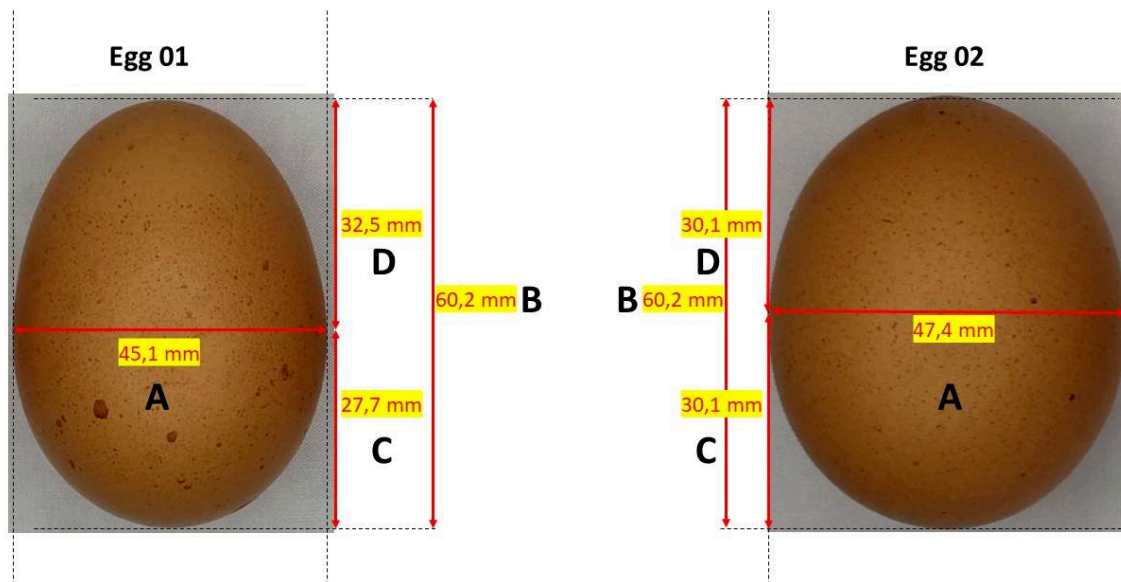
1.1 Justificativa

Na indústria avícola de postura o formato do ovo de galinha tem bastante relevância para o desenvolvimento genético de novas espécies, ovos com a base mais larga são preferidos em detrimento de ovos mais alongados para auxiliar no rompimento da casca pelos filhotes de aves, como pode ser visto na Figura 1.

A organização multinacional Lohmann Tierzucht GmbH, líder no ramo de melho-

ramento genético de ovos de galinha está sempre buscando melhorias para seu processo de estudo genético, sendo assim surgiu a necessidade de criar um dispositivo automatizado capaz de extrair as medidas da casca de um ovo através do processamento digital de imagens, com o intuito de minimizar erros e incertezas de medições manuais.

Figura 1 – Exemplo das medidas obtidas da casca do ovo.



1.2 Objetivo

Este trabalho tem o objetivo de desenvolver um sistema embarcado de baixo custo para medir automaticamente as características dimensionais de uma casca de ovo, incluindo a maior distância transversal e longitudinal, juntamente com as medidas secundárias de uma casca de ovo em seu estado natural. A solução proposta baseia-se na obtenção de uma foto do ovo e na utilização de algoritmos de computação visual para produzir as principais medidas da casca do ovo.

O dispositivo em questão deverá incluir:

- Um mecanismo de correção da posição do ovo para adquirir as imagens. Este mecanismo consiste em uma base para fixar o ovo em uma posição pré-estabelecida, suporte para a câmera e as portas para conectar o equipamento ao computador;
- Uma pequena câmera de alta resolução responsável por adquirir a imagem com a resolução necessária para realizar o processamento;
- Um pequeno computador para embarcar o sistema desenvolvido;

- Um sistema de computação visual desenvolvido para controlar todo o processo, incluindo a aquisição de imagens e processamento de informações, fornecendo automaticamente os resultados.

1.3 Organização do Trabalho

No capítulo 2 é apresentado a fundamentação teórica responsável por demonstrar os conceitos sobre visão computacional e interface homem-máquina. No capítulo 3 são descritos alguns dos principais trabalhos correlatos à classificação de ovos de galinha e os métodos usados pelos autores. No capítulo 4 são encontrados os materiais e métodos utilizados neste trabalho, que demonstra a integração entre hardware e software desenvolvidos para a construção de um sistema embarcado e como os mesmos foram usados para obtenção dos resultados. No capítulo 5 são apresentados os resultados e discussões, salienta-se os dados estatísticos observados durante o trabalho e também a discussão sobre os resultados obtidos. No capítulo 6 as conclusões com relação ao sistema embarcado e levantamento de pontos de melhoria para trabalhos futuros são apresentadas.

2 Fundamentação teórica

Este capítulo será organizado em três seções. Na seção 2.1 serão embasadas teorias principais cujo conhecimento são necessários para uma melhor compreensão do desenvolvimento realizado. A princípio será apresentado o conceito de Processamento Digital de Imagens (PDI) e das suas principais técnicas de filtragem. Na seção 2.2 definir-se-á o algoritmo de computação visual utilizado, suas regras e particularidades. Por fim, a seção 2.3 detalha o conceito de interface homem-máquina, sua função e características aplicadas no dispositivo.

2.1 Processamento Digital de Imagens

Processamento Digital de Imagens - PDI consiste na manipulação de imagens de maneira que tanto a entrada quanto a saída do processo seja uma imagem. A ideia em processar uma imagem é que ela seja representada com mais efetividade no contexto computacional, uma vez que ela é convertida em dados. Os dados capturados após o processamento são submetidos a técnicas afim de obter variações favoráveis da imagem para que essas variações sejam utilizadas em outros processamentos (CÂMARA et al., 1996).

De acordo com Gonzalez, Woods e Eddins (2012) o espectro que vai do PDI até a visão computacional, passa por 3 níveis, sendo eles baixo-nível, nível-médio e alto-nível. O processo de baixo-nível consiste em operações primitivas como redução de ruídos ou melhorias no contraste da imagem, nível-médio que são processos como segmentação ou classificação, já os de alto-nível são relacionados a tarefas de cognição associadas a visão humana. São tarefas relativas ao baixo-nível:

1. **Escala de Cinza:** Escala de cinza é uma escala em tons de cinza em que os valores *pixels* da imagem variam de acordo com sua intensidade (luminosidade), numa escala entre preto e branco. Pode ser obtida a partir da conversão RGB para YIQ que só normaliza os níveis de brilho da imagem (FILHO; NETO, 1999).
2. **Histograma:** O histograma de uma imagem é um conjunto de números indicando o percentual de *pixels* naquela imagem que apresentam um determinado nível de cinza. Estes valores são normalmente representados por um gráfico de barras que fornece para cada nível de cinza o número (ou o percentual) de *pixels* correspondentes na imagem. Através da visualização do histograma de uma imagem obtemos uma indicação de sua qualidade quanto ao nível de contraste e quanto ao seu brilho médio (se a imagem é predominantemente clara ou escura) (FILHO; NETO, 1999).

3. **Binarização:** O algoritmo de binarização é normalmente utilizado quando é preciso fazer a separação entre o fundo da imagem com os objetos que representam os caracteres. A binarização ou limiarização (do inglês, *thresholding*) é o método mais simples para segmentação de imagens que consiste em separar regiões de não interesse através da escolha de um ponto limiar. Em alguns casos não é possível dividir a imagem em apenas um limiar que resulte em resultados satisfatórios, nesses casos são definidos mais de um ponto de corte (*threshold*) da imagem (FILHO; NETO, 1999).
4. **Filtros:** As técnicas de filtragem são transformações da imagem pixel a pixel, que não dependem apenas do nível de cinza de um determinado pixel, mas também do valor dos níveis de cinza dos *pixels* vizinhos. O processo de filtragem é feito utilizando matrizes denominadas máscaras, as quais são aplicadas sobre a imagem (FILHO; NETO, 1999).

2.2 Algoritmo de Visão Computacional

A visão computacional procura modelar e replicar a visão humana usando software e hardware. O processamento de imagens tem uma imagem como parâmetro de entrada e um conjunto de valores numéricos como saída (objetos complexos, vetores, matrizes, etc). Adiante, tem-se uma breve introdução da linguagem de programação utilizada neste projeto e sua principal biblioteca associada, capaz de facilitar o processamento das imagens com suas funções.

2.2.1 Python & OpenCV

De acordo com PyScience-Brasil (2016), Python é uma linguagem de programação criada por Guido van Rossum em 1991. Os objetivos do projeto da linguagem eram: produtividade e legibilidade. Em outras palavras, Python é uma linguagem que foi criada para produzir código bom e fácil de manter de maneira rápida. Além disso, Python suporta múltiplos paradigmas de programação. A programação procedimental pode ser usada para programas simples e rápidos, mas estruturas de dados complexas, como tuplas, listas e dicionários, estão disponíveis para facilitar o desenvolvimento de algoritmos complexos. Grandes projetos podem ser feitos usando técnicas de orientação a objetos, que é completamente suportada em Python (inclusive sobrecarga de operadores e herança múltipla). Um suporte modesto para programação funcional existe, o que torna a linguagem extremamente expressiva: é fácil fazer muita coisa com poucas linhas de comando. E também possui inúmeras capacidades de meta-programação: técnicas simples para alterar o comportamento de comportamentos da linguagem, permitindo a criação de linguagens de domínio específico. Por fim, e não menos importante, Python é uma linguagem livre e multiplataforma. Isso

significa que os programas escritos em uma plataforma serão executados sem nenhum problema na maioria das plataformas existentes sem nenhuma modificação. E, caso a plataforma objetivo não tenha uma versão de Python, desenvolvedores têm a liberdade de estudar e modificar o código da linguagem para fazer com que ela execute onde quer que seja.

OpenCV (*Open Source Computer Vision*) é uma biblioteca de programação, de código aberto e inicialmente desenvolvida pela Intel com o objetivo de tornar a visão computacional mais acessível a desenvolvedores e hobistas. Atualmente possui mais de 500 funções, pode ser utilizada em diversas linguagens de programação (C++, Python, Ruby e Java) e é usada para diversos tipos de análise em imagens e vídeos, como detecção, *tracking* e reconhecimento facial, edição de fotos e vídeos, detecção e análise de textos, entre outros (CEDROTECH, 2018).

2.3 Interface Homem-Máquina

De acordo com Flateck (2017), a IHM tem grande importância na área industrial pois transforma a linguagem de baixo nível em imagens intuitivas para o operador. A ISO 9241-110 define uma interface de usuário como todas as partes de um sistema interativo (de software ou hardware) que fornecem informações e controle necessários para que o usuário realize uma determinada tarefa com o sistema interativo (COPADATA, 2017).

Portanto, sempre que existir uma tela onde a navegação aconteça por meio de ícones, janelas ou qualquer outra maneira simples de ser compreendida e de fácil execução, essa é então uma IHM (SABER ELÉTRICA, 2017).

3 Trabalhos Correlatos

Neste tópico serão listados três trabalhos correlatos ao proposto. O trabalho apresentado na seção 3.1 apresenta um estudo sobre o desenvolvimento de um sistema de rede neural baseado em lógica fuzzy e visão computacional capaz de classificar ovos de galinha (OMID et al., 2013). Na seção 3.2 é demonstrado um dispositivo capaz de calcular indicadores de qualidade de ovos de galinha, dentre eles, dimensões geométricas, textura e espessura da casca através de processamento digital de imagens (VASILEVA et al., 2018). Na seção 3.3 é abordado a construção de um aplicativo *mobile* disponível para a plataforma Android capaz de extrair e analisar as características morfológicas das imperfeições externas existentes em ovos de galinha através de fotografias presentes no próprio *smartphone* (ROSA, 2017).

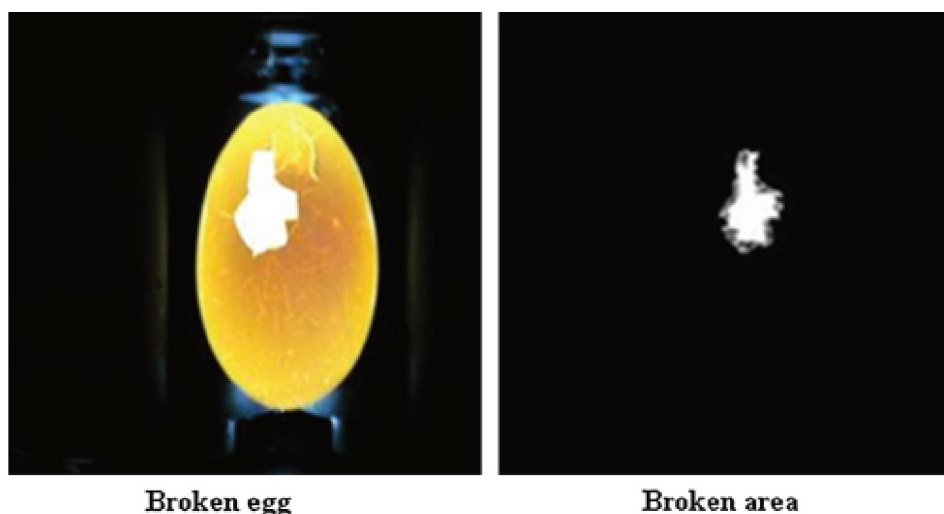
3.1 An expert egg grading system based on machine vision and artificial intelligence techniques

O trabalho de Omid et al. (2013) apresenta um sistema inteligente baseado em técnicas combinadas de lógica fuzzy e visão computacional para classificação de ovos usando parâmetros como defeitos e tamanho dos ovos. Os defeitos detectados foram manchas de sangue internas, rachaduras e quebras na casca de ovo. A taxa de classificação correta (CCR) foi de 95% para a detecção do tamanho, 94,5% para detecção de rachaduras e 98% para detecção de quebra.

De acordo com Omid et al. (2013) para ter uma imagem adequada de um ovo, a criação de uma luz uniforme em torno da amostra é importante e deve ser cuidadosamente projetada. Nesse sentido, em seu experimento, uma caixa cúbica de 0,40 m³ foi fabricada com iluminação na base e no topo da caixa. Para captação das imagens, uma câmera SAMSUNGTM CCD (modelo: SDN-550) foi usada e fixada aproximadamente 200 mm acima da amostra de ovo.

Através da segmentação da imagem capturada pela câmera e algoritmos de processamento de imagem, Omid et al. (2013) determinaram o tamanho do ovo, presença de manchas de sangue, trincas e rupturas, a Figura 2 traz um exemplo de detecção de ruptura na casca do ovo de galinha.

Figura 2 – Ovo quebrado e reconhecimento de área quebrada



Fonte: *Omid et al. (2013)*

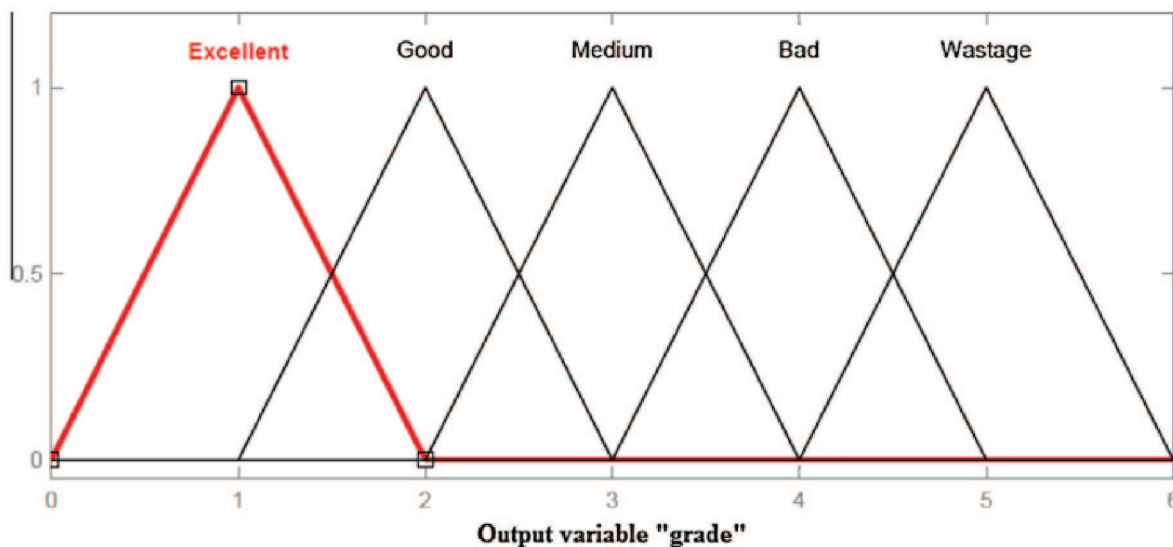
O Sistema de Inferência Fuzzy (FIS) foi projetado para classificar os ovos com base em quatro variáveis de entrada (trinca, manchas de sangue, ruptura e tamanho). A saída do sistema projetado é a classificação do ovo baseado nas variáveis pré-estabelecidas respeitando as regras fuzzy aplicadas no FIS, conforme mostra a Figura 3. Dessa forma, o ovo é classificado em cinco tipos: Excelente (A), Bom (B), Médio (C), Ruim (D) e Desperdício (E), como pode ser visto na Figura 4. A Figura 5 demonstra o modelo do Sistema de Inferência Fuzzy desenvolvido no MATLAB Simulink utilizado neste trabalho.

Figura 3 – Regras Fuzzy no sistema de classificação de ovos

1	IF Egg is (not Bloody) AND (not Broken) AND (No Crack) AND (Large) THEN (Grade is A)
2	IF Egg is (not Bloody) AND (Breakage is not Broken) AND (Crack is No Crack) AND (Size is Medium) THEN (Grade is B)
3	IF Egg is (not Bloody) AND (Breakage is not Broken) AND (Crack is No Crack) AND (Size is Small) THEN (Grade is B)
4	IF Egg is (not Bloody) AND (Breakage is not Broken) AND (Crack is Low Crack) AND (Size is Large) THEN (Grade is B)
5	IF Egg is (not Bloody) AND (Breakage is not Broken) AND (Crack is Low Crack) AND (Size is Medium) THEN (Grade is C)
6	IF Egg is (not Bloody) AND (Breakage is not Broken) AND (Crack is Low Crack) AND (Size is Small) THEN (Grade is C)
7	IF Egg is (not Bloody) AND (Breakage is not Broken) AND (Crack is Medium Crack) AND (Size is Large) THEN (Grade is B)
8	IF Egg is (not Bloody) AND (Breakage is not Broken) AND (Crack is Medium Crack) AND (Size is Medium) THEN (Grade is C)
9	IF Egg is (not Bloody) AND (Breakage is not Broken) AND (Crack is Medium crack) AND (Size is Small) THEN (Grade is D)
10	IF Egg is (not Bloody) AND (Breakage is not Broken) AND (Crack is High crack) AND (Size is Large) THEN (Grade is C)
11	IF Egg is (not Bloody) AND (Breakage is not Broken) AND (Crack is High Crack) AND (Size is Medium) THEN (Grade is D)
12	IF Egg is (not Bloody) AND (Breakage is not Broken) AND (Crack is HighCrack) AND (Size is Small) THEN (Grade is D)
13	IF Egg is (Bloody) THEN (Grade is E)
14	IF (Breakage is Broken) THEN (Grade is E)

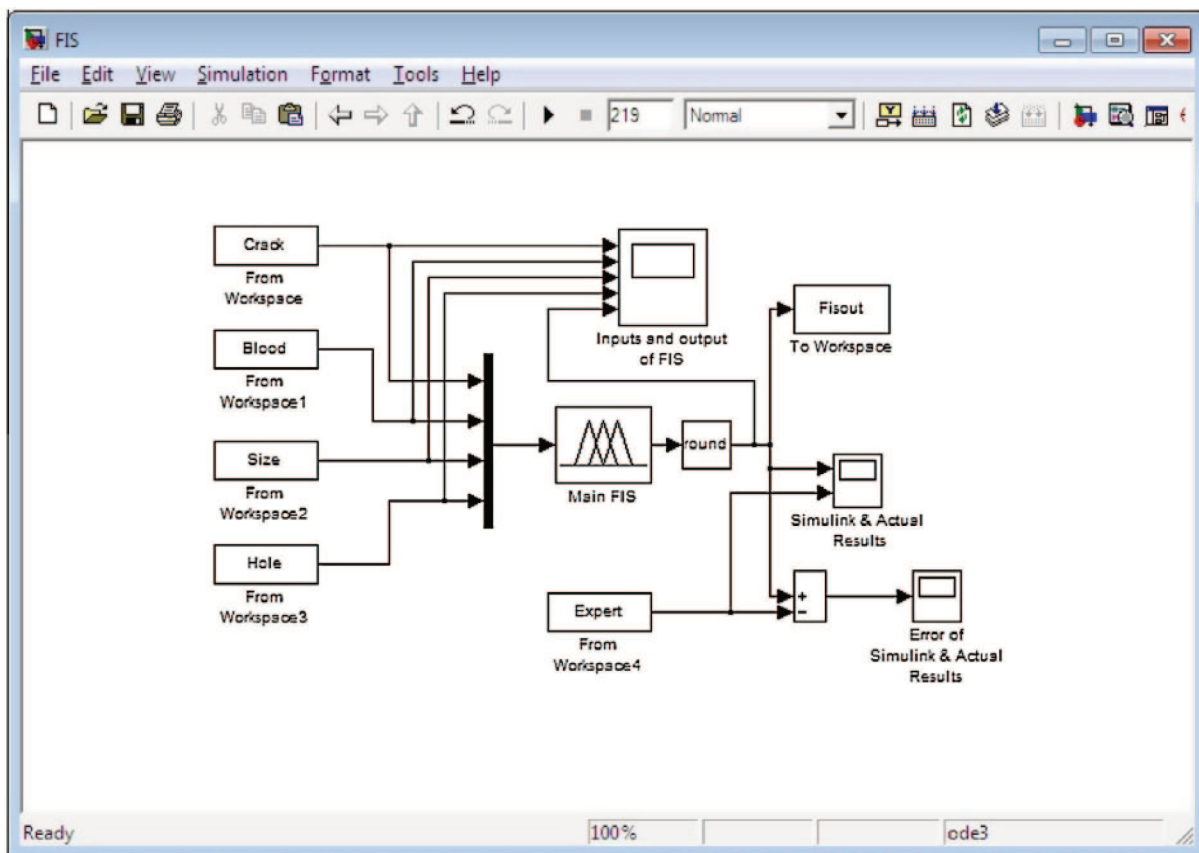
Fonte: *Omid et al. (2013)*

Figura 4 – Exemplo de saída do sistema projetado



Fonte: Omid et al. (2013)

Figura 5 – FIS desenvolvido na ferramenta MATLAB Simulink para classificação de ovos



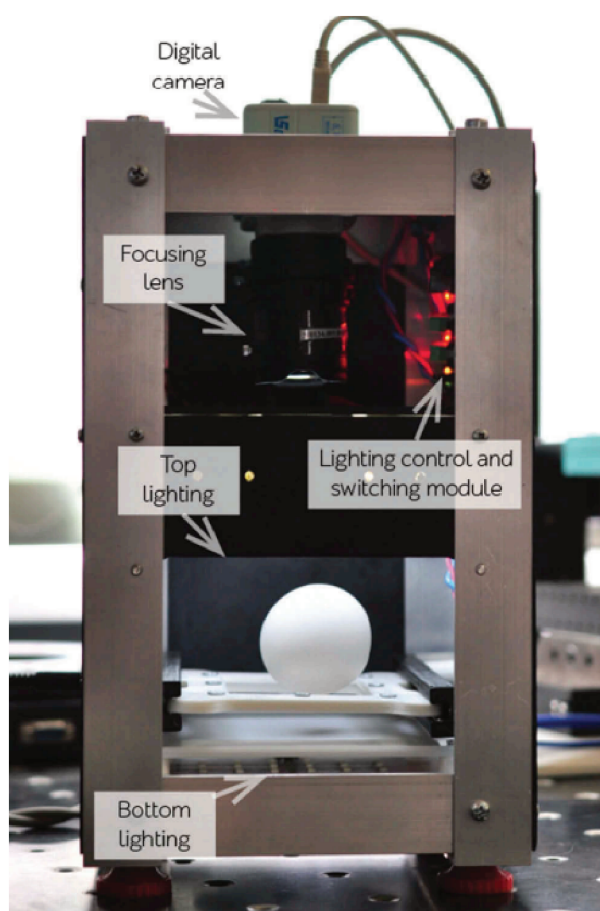
Fonte: Omid et al. (2013)

3.2 Assessing exterior egg quality indicators using machine vision

O trabalho de Vasileva et al. (2018) teve por objetivo desenvolver um método de visão computacional capaz de analisar parâmetros externos de ovos de galinha para automatizar a etapa de classificação primária. O algoritmo desenvolvido baseia-se em limiares predeterminados capaz de abstrair indicadores de qualidade do ovo, incluindo dimensões geométricas, textura e espessura da casca. Um total de 400 amostras de ovos foram analisadas, comparando os resultados do algoritmo com os obtidos pelo método manual tradicional, o valor médio da diferença dos valores dos raios foi de $0,095 \pm 0,058$ mm para a ponta alongada e $0,080 \pm 0,047$ mm para a ponta achatada do ovo, com desvios padrão de 0,58 mm e 0,49 mm, respectivamente.

O dispositivo desenvolvido neste trabalho se assemelha bastante ao desenvolvido por Vasileva et al. (2018), conforme ilustrado na Figura 6.

Figura 6 – Dispositivo construído para determinação dos parâmetros externos do ovo

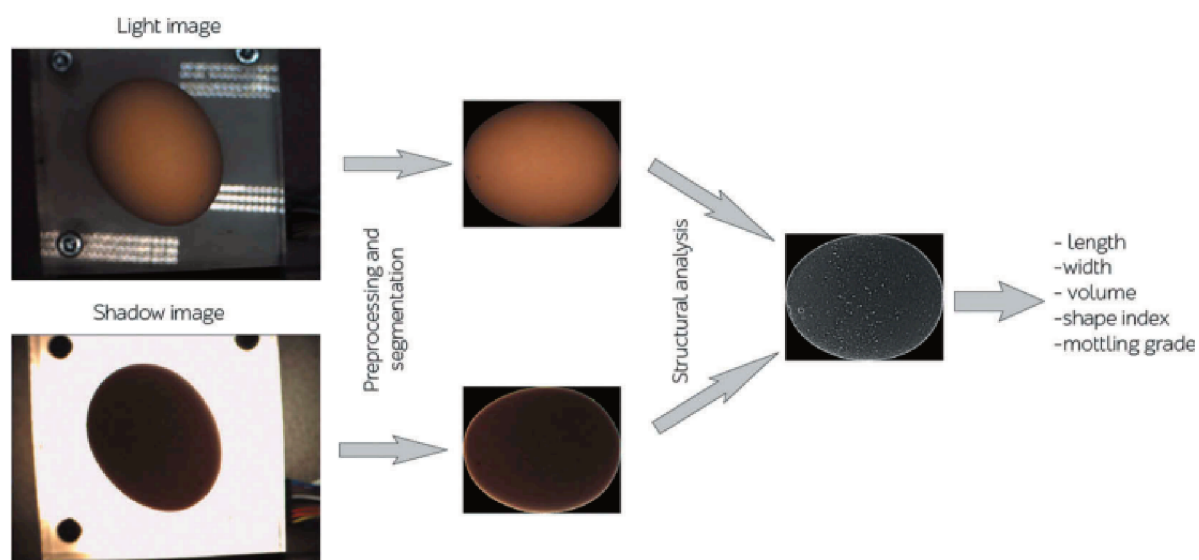


Fonte: Vasileva et al. (2018)

Os autores ainda descreveram em seu trabalho o processo de segmentação das imagens retiradas pela câmera, o contraste entre a imagem iluminada e a imagem da

sombra do ovo gerou a imagem utilizada para obtenção dos parâmetros dimensionais, formato do ovo, espessura da casca e presença de manchas no exterior, conforme ilustrado na Figura 7.

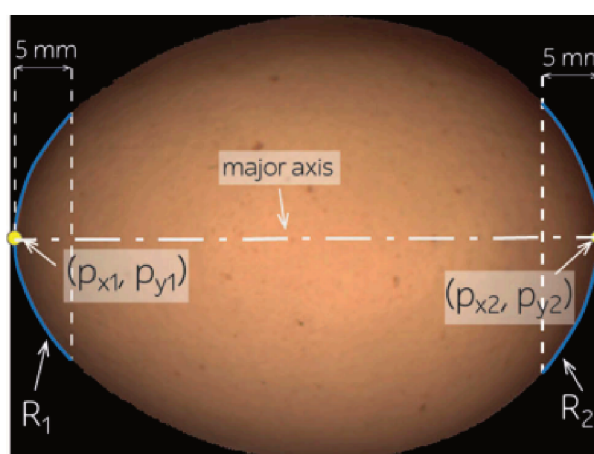
Figura 7 – Passo a passo do fluxo de processamento de imagem



Fonte: Vasileva et al. (2018)

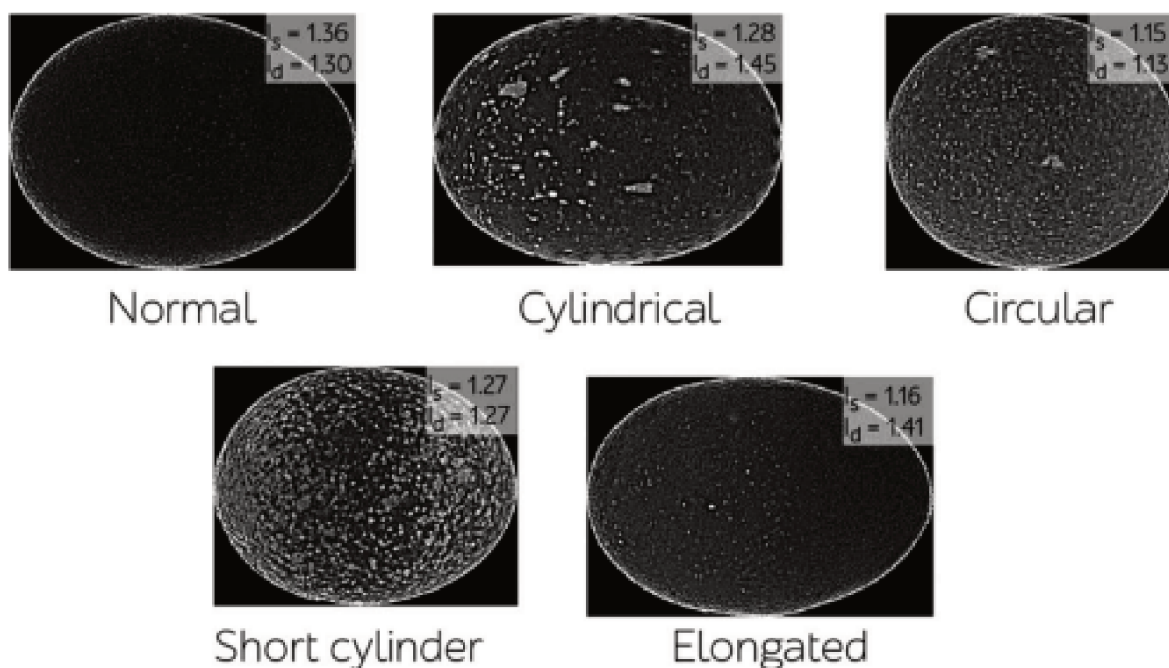
O algoritmo de processamento de imagens desenvolvido por Vasileva et al. (2018) calcula a maior distância entre dois pontos da superfície externa do ovo nomeado como *major axis*, e através desta medida estima-se o raio de curvatura das pontas dos ovos, ilustrados como R_1 e R_2 na Figura 8. Com os valores dos raios de curvatura obtidos, Vasileva et al. (2018) classificam os ovos de acordo com seu formato e comprimento, de acordo com a Figura 9.

Figura 8 – Estimativa do raio de curvatura das pontas dos ovos



Fonte: Vasileva et al. (2018)

Figura 9 – Exemplo da classificação de ovos baseado nos valores do índice da forma e do índice do tamanho



Fonte: Vasileva et al. (2018)

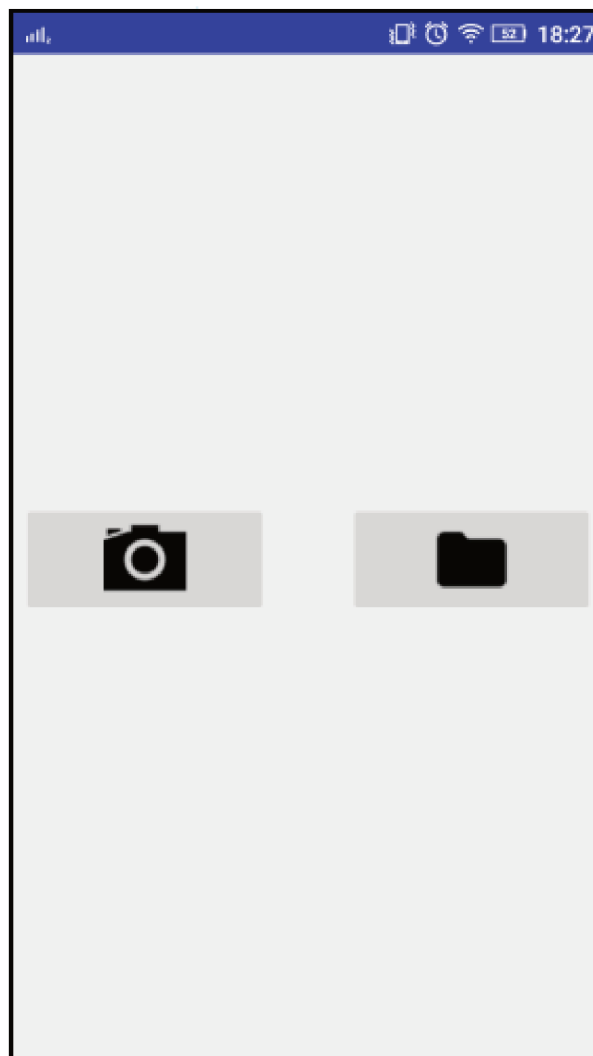
3.3 Egg Evaluation: Ovoscopia via Smartphone

O trabalho de ROSA (2017) apresenta o desenvolvimento de um aplicativo para a plataforma Android que realiza processamento de imagens em ovos de galinha da cor branca buscando extrair características como a sujeira, rachaduras e a qualidade da casca. As aplicações das técnicas de processamento de imagem foram realizadas através da biblioteca JavaCV, que é uma adaptação da biblioteca OpenCV para Java. O aplicativo faz o uso de imagens digitais capturadas através de dispositivos móveis, tais como: fotos e fotos da galeria, como pode ser visto na Figura 10 referente a tela inicial do aplicativo.

Os resultados apresentados por ROSA (2017) dizem a respeito das características encontradas no ovo. Estas características são respectivamente: se o ovo está quebrado, presença de sujeira e a qualidade da casca. As rachaduras do ovo são representadas da cor azul. A sujeira do ovo, como fezes ou gema, são representadas da cor amarela. E por último a porosidade da casca, que é representada na cor verde, um exemplo do resultado da classificação de um ovo de galinha pode ser observado na Figura 11

Os resultados deste trabalho foram gerados a partir da análise de 30 ovos distintos, com o intuito de avaliar as seguintes características: presença de sujeira na casca, presença de rachadura na casca e porosidade da casca. Conforme os resultados apresentados pelo autor, o diagnóstico da presença de sujeiras obteve sucesso, embora não foi possível destacar com alta precisão todas as sujeiras percebidas a olho nu. Também foi percebido uma

Figura 10 – Tela inicial do aplicativo



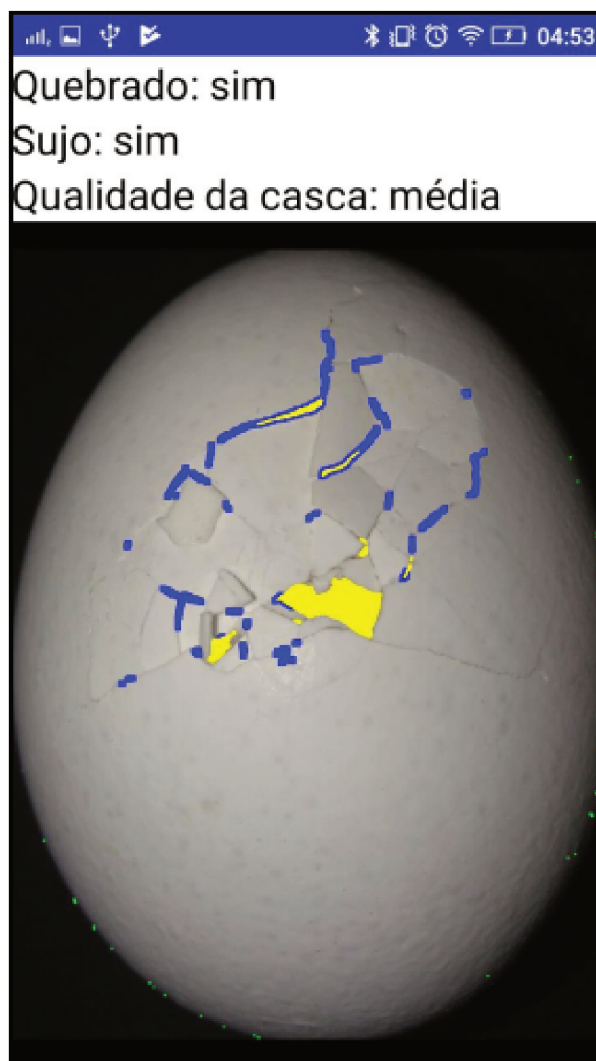
Fonte: ROSA (2017)

dificuldade no diagnóstico da presença de rachaduras, porém em todos os casos onde havia efetivamente rachaduras, elas foram encontradas. A dificuldade na distinção dá-se em virtude da variedade de cores na sujeira e também devido ao sombreamento sobre as sujeiras. A presença de pontos de sangue também confundiu a classificação. O diagnóstico da qualidade da casca obteve um bom resultado, porém em alguns casos ele deu uma nota acima do esperado, em outros, abaixo, afirma ROSA (2017).

3.4 Comentários

Os trabalhos apresentados neste capítulo são capazes de extrair as características morfológicas de ovos de galinha de forma automática através do processamento digital de imagens, cada qual com seu custo e benefício. Notou-se que os sistemas de processamento

Figura 11 – Resultado do processamento



Fonte: ROSA (2017)

de imagem apresentaram resultados bastante encorajadores (acima de 94% de índice de acerto), com baixa taxa de falso positivo.

Todos os trabalhos desenvolveram um sistema para captação das imagens através de uma câmera de alta resolução, porém apenas o sistema embarcado desenvolvido neste trabalho se preocupou em desenvolver uma interface homem-máquina para interação do usuário e ainda é capaz de exportar os dados. Por outro lado, os trabalhos apresentados realizam a detecção de múltiplos indicadores de qualidade e este trabalho foi desenvolvido para extrair apenas as características dimensionais de uma casca de ovo.

4 Materiais e métodos

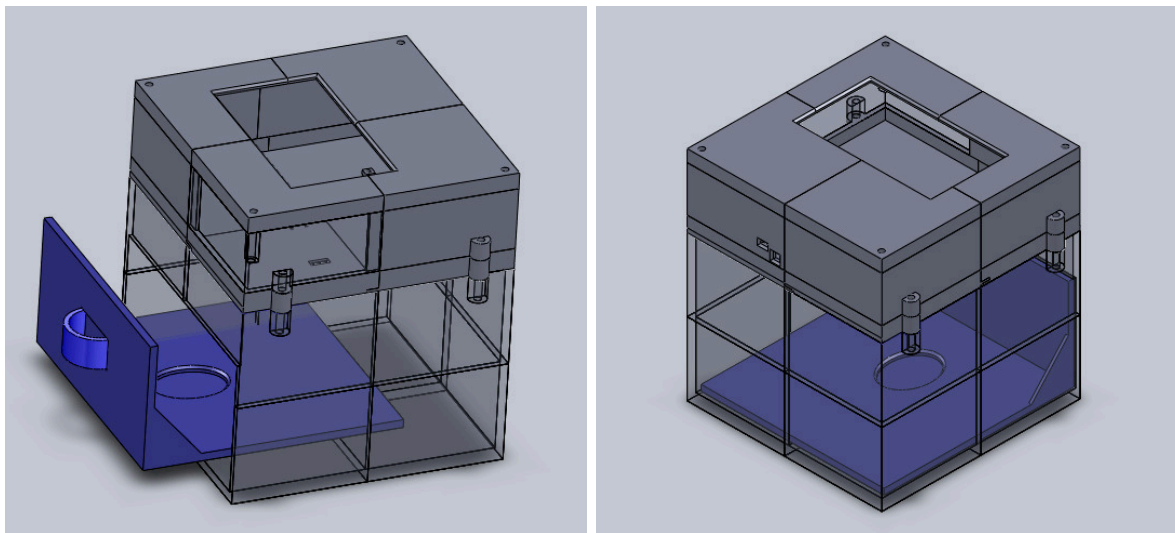
O presente capítulo apresenta as principais etapas realizadas para o desenvolvimento deste trabalho e, em cada uma delas, são explicitados os procedimentos teóricos, numéricos e experimentais empreendidos.

4.1 Construção do protótipo

A escolha dos equipamentos utilizados em um projeto representam a parte primordial do desenvolvimento de um produto e todas as futuras decisões dependem desta. O desenho do dispositivo precisa ser bem dimensionado e atender às especificações da aplicação para não comprometer o desenvolvimento das etapas subsequentes.

O protótipo foi projetado para acomodar o conjunto de hardware e fornecer ao usuário uma fácil troca de amostras para novas medições. A concepção do projeto também permite que o dispositivo seja tratado como uma aplicação multifuncional, onde o usuário pode transportá-lo e realizar medições em campo, como pode ser visto na Figura 12.

Figura 12 – Representação 3D do protótipo



(a) Vista Lateral

(b) Vista Posterior

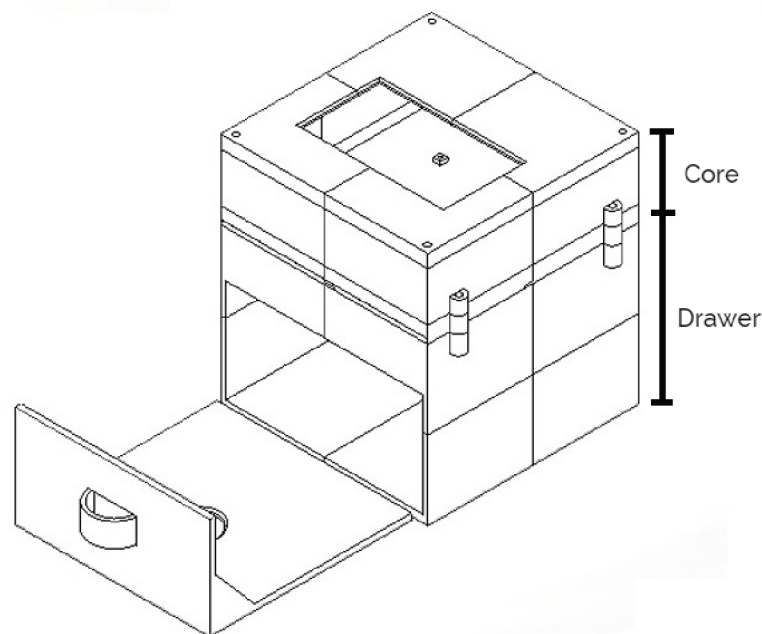
O protótipo desenvolvido teve sua estrutura segmentada em duas seções bem definidas, a seguir irei detalhar cada uma delas:

1. **Core:** Esta área, localizada no topo do dispositivo foi construída para acomodar as interconexões entre todos os equipamentos utilizados, uma vez que nela foram

instalados o conjunto de hardware responsável pelo processamento dos dados e interface homem-máquina (IHM). Além disto, esta parte do dispositivo é responsável pela interação do usuário com o protótipo, onde todas as informações e resultados serão exibidos através de um *display touchscreen* de 7 polegadas.

2. **Drawer:** Compartimento para acomodação do ovo em análise. Consiste em uma câmara de gaveta, com controle de iluminação, para o posicionamento e aquisição das imagens dos ovos.

Figura 13 – Perspectiva isométrica do protótipo



O material escolhido para construção da estrutura física do protótipo foi placa de fibra de média densidade, também conhecido como MDF (*Medium Density Fiberboard*), a Figura 14 abaixo representa o ponto de vista do usuário com o compartimento de acomodação dos ovos de galinha aberto e IHM desligada.

Figura 14 – Protótipo sob o ponto de vista do operador



A seguir, serão descritos os principais equipamentos escolhidos para construção do protótipo e suas respectivas especificações técnicas:

4.1.1 Raspberry Pi 4 Model B

O Raspberry Pi consiste em um computador com o tamanho reduzido, com baixo custo e baixo consumo de energia, desenvolvido em 2011, pela Fundação Raspberry Pi (RASPBERRY PI FOUNDATION, 2017). De acordo com MONK (2013), o computador foi criado com o intuito de promover o ensino de programação para crianças e adultos. Hoje ele é utilizado de diversas formas, por exemplo, para programar, como vídeo game, mini servidor, automação residencial, robótica entre outros projetos.

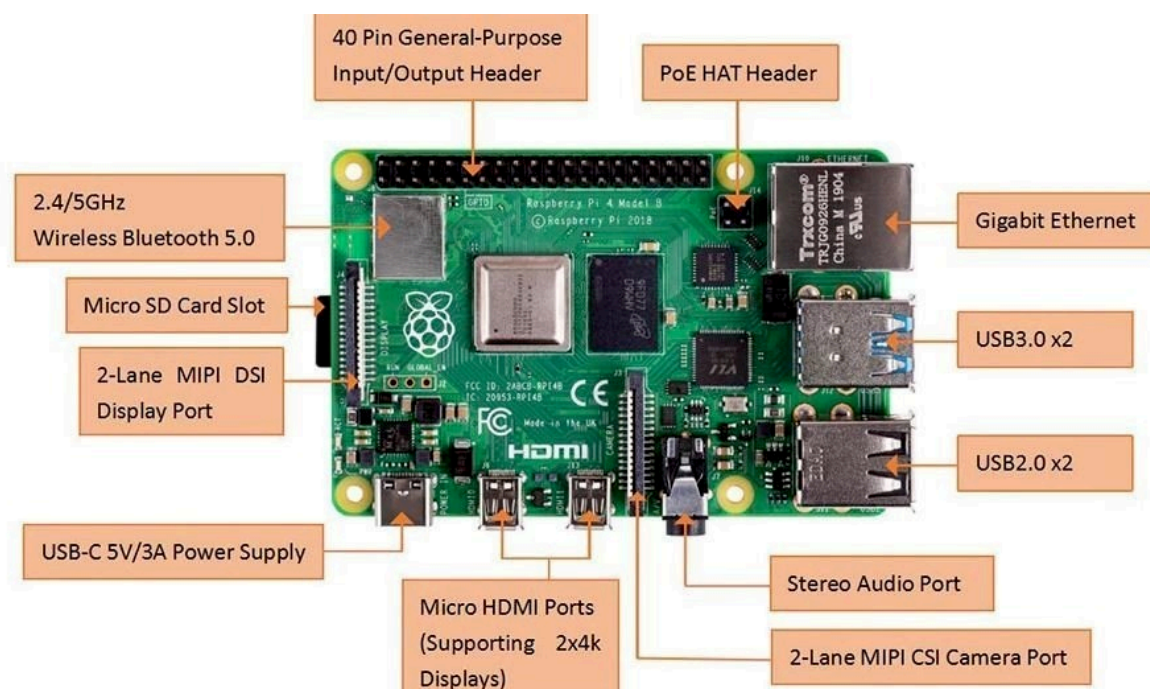
Este dispositivo é classificado como SBC (*Single-Board Computer*), pois integram em uma única placa, processador, memória e interfaces de entrada e saída. O Raspberry Pi se enquadra na categoria de equipamento denominada *OpenSource Hardware* (OSH). Os OSHs podem ser definidos como dispositivos elétricos ou mecânicos cujas informações

sobre o projeto são disponibilizadas para todos, de forma que o público possa fazer uso, modificar, produzir ou distribuir este material.

Existem vários modelos de Raspberry Pi, alguns são apropriados para projetos mais robustos e outros para projetos mais simples. Nesta pesquisa, utilizou-se o Raspberry Pi 4 modelo B, vide Figura 15. O modelo Raspberry Pi 4 B possui a seguinte especificação:

- Processador Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5 GHz;
- Memória RAM de 4GB DDR4;
- Adaptador Wifi 2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN integrado;
- Bluetooth 5.0 BLE integrado;
- 2 portas micro HDMI com suporte a vídeos 4K e 60fps;
- 2 portas USB 3.0 e 2 portas USB 2.0;
- Alimentação através de fonte DC chaveada 5.1V 3A com conector USB-C ou *Power over Ethernet* (PoE) habilitado;
- Interface para câmera (CSI);
- Interface para *display* (DSI);
- *Slot* para cartão microSD;
- Conector de áudio e vídeo;
- GPIO de 40 pinos;
- Dimensões: 85 x 56 x 17 mm;
- Temperatura de operação 0–50 °C.

Figura 15 – Raspberry Pi 4 Model B

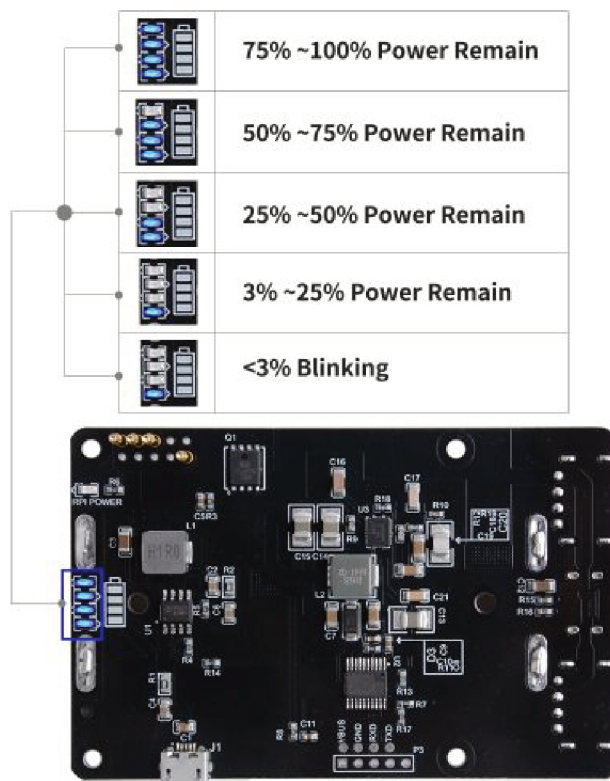


Fonte: *Raspberry Pi Foundation (2017)*

4.1.2 Fonte de alimentação ininterrupta para Raspberry Pi 4B

A fonte de alimentação utilizada neste trabalho consiste em uma UPS (*Uninterruptible Power Supply*) capaz de fornecer energia suficiente para que o Raspberry Pi funcione em movimento, garantindo mais autonomia ao processo de trabalho. De acordo com 52Pi Wiki contributors (2020), este *no-break* ainda propicia a facilidade na operação de substituir a bateria por conta própria, suporta a maioria das baterias padrão 18650 e está totalmente em conformidade com as características da bateria em termos de vida útil da bateria. Além disso, o indicador de energia fornecido na placa de circuito impresso também é muito amigável. O LED iluminado pode mostrar rapidamente a energia restante e apoiar a descarga durante o carregamento, conforme pode ser visto na Figura 16. É possível ainda, se conectar diretamente à fonte de alimentação externa para carregar, sem que o Raspberry Pi seja desligado.

Figura 16 – Indicador de carga remanescente da UPS



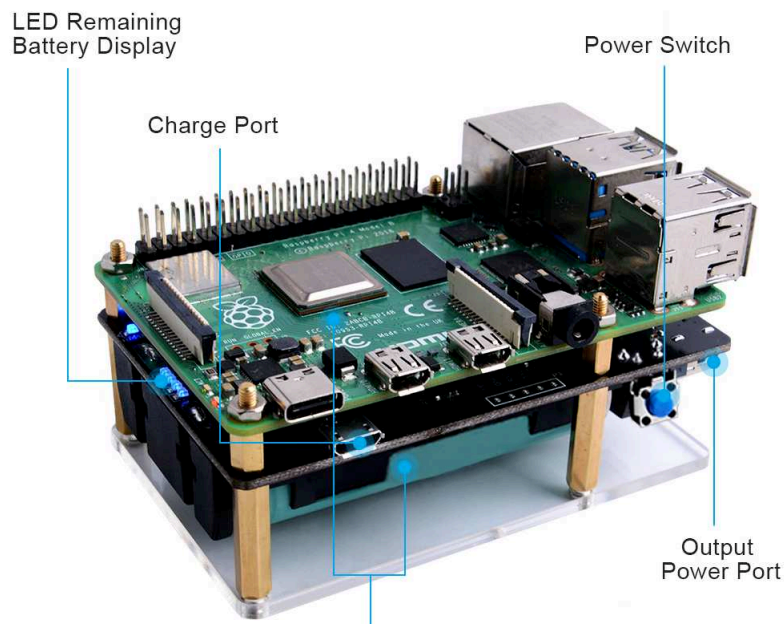
Fonte: *52Pi Wiki contributors (2020)*

Neste trabalho foram utilizadas duas baterias de Li-Ion 18650 2.200mAh, logo abaixo detalharei a fonte de alimentação ininterrupta para Raspberry Pi 4B em conjunto com as duas baterias supracitadas:

- LED indicador de bateria;
- Compatibilidade com Raspberry Pi 4B / 3B+ / 3B;
- Potência nominal: 15 W;
- Saída de energia USB-A;
- Saída de energia USB-C;
- Proteção contra sobrecorrente;
- Dimensões da UPS: 87,2 x 57 mm.
- Bateria cilíndrica
- Tipo: Li-íon

- Tensão Nominal: 3,7 V
- Capacidade nominal: 2200 mAh
- Dimensões da bateria: 65 mm de altura, 18 mm de diâmetro
- Peso: 45 g aproximadamente

Figura 17 – Integração entre Raspberry Pi 4 e UPS



Fonte: Adaptada de *52Pi Wiki contributors (2020)*

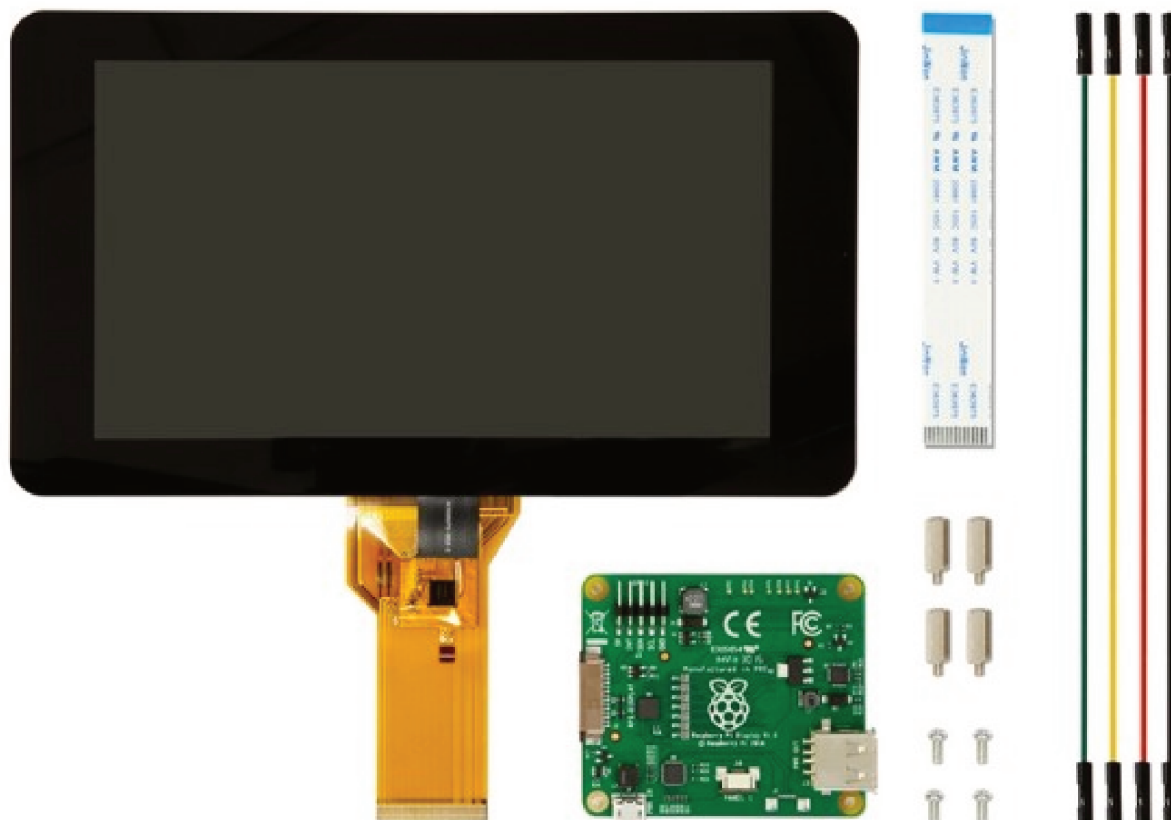
4.1.3 Display Raspberry Pi Touchscreen 7"

O *Display Raspberry Pi Touchscreen 7"* consiste em um display TFT de 7 polegadas com tela sensível ao toque e resolução de 800×480 pixels. No escopo deste trabalho, a IHM é representada pelo *display* integrado ao sistema supervisorio da aplicação desenvolvida, estabelecendo nosso principal meio de comunicação com o usuário.

Como explica FAVARI (2021), o TFT (*Thin Film Transistor* ou Transistor de Película Fina) na verdade é um tipo de LCD (*Display* de Cristal Líquido) cujos pixels ou pontos são controlados (cor, brilho e nitidez) individualmente. No TFT, cada ponto é composto por três células (vermelho, verde e azul) e elas contam com iluminação traseira por lâmpadas especiais (ou LED, nas mais tradicionais), que controlam a intensidade da passagem da luz individualmente, para formar a cor ou o brilho desejados em cada pixel. Em relação às telas tradicionais, a TFT tem melhor qualidade de imagem e também não deixa os rastros e sombras que ocorrem em mudanças muito rápidas, porém apresenta um custo mais alto em relação as demais tecnologias.

O *display* é conectado à uma placa adaptadora, que se comunica com o Raspberry Pi utilizando o conector DSI (*Display Serial Interface*), a alimentação 5V pode ser feita pela própria GPIO do Raspberry, utilizando dois pinos (FILIFEFLOP, 2019).

Figura 18 – Display Raspberry Pi Touchscreen 7”



Fonte: *FilipeFlop (2019)*

O *Display Raspberry Pi Touchscreen* possui as seguintes especificações:

- Display TFT Touchscreen 7” Raspberry Pi;
- Resolução de 800×480 pixels;
- Touch Capacitivo;
- Dimensões: 19,4 x 11 x 2 cm (incluindo espaçadores);
- Área visível da tela: 15,5 x 8,6 cm.

4.1.4 Câmera Raspberry Pi v2 8MP

A escolha da câmera ideal representou um ponto de bastante atenção no desenvolvimento deste projeto, seria necessário um dispositivo capaz de entregar alta resolução e

qualidade nas capturas e ao mesmo tempo compatível com o Raspberry Pi e as proporções do protótipo construído.

Desta forma, o dispositivo escolhido foi a Câmera Raspberry Pi v2 com 8MP e sensor Sony IMX219, com resolução de 8 megapixels, capacidade de captar vídeos de 1080p30, 720p60, 640x480p90 e imagens estáticas de até 3280×2464 pixels. A integração com o Raspberry Pi se dá através do conector CSI (*Camera Serial Interface*), especialmente dedicado para este tipo de conexão e troca de dados (FILIPEFLOP, 2021).

Especificações:

- Sensor: Sony IMX219;
- Resolução: 8MP;
- Imagem: 2592×1944 pixels;
- Vídeo: 1080p30, 720p60 e 640x480p60/90;
- Comunicação via CSI;
- Suporta vídeos 1080p;
- Dimensões: 25 x 20 x 9 mm;
- Peso: 3g.

Figura 19 – Câmera Raspberry Pi v2 8MP



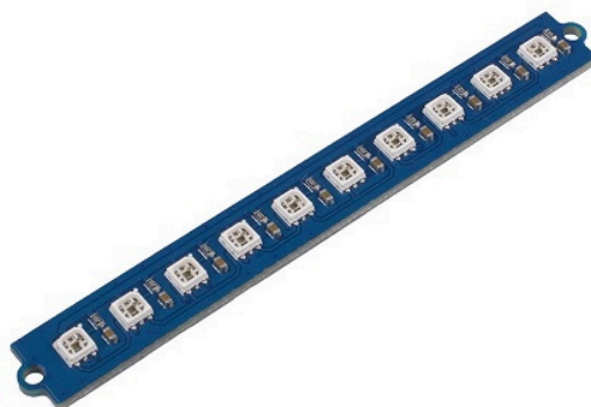
Fonte: *FilipeFlop (2021)*

4.1.5 Linha de LED RGB x10 WS2813 Grove

Neste trabalho foram utilizadas duas linhas de LED RGB x10 WS2813 responsáveis por controlar a iluminação do compartimento onde as imagens são capturadas. O dispositivo consiste em 10 LEDs em uma única placa, sendo que cada LED pode ser acionado individualmente através do chip de *driver* WS2813 mini. Além disso, utilizando o conector Grove, todos os LEDs são controlados por apenas um pino de sinal GPIO do Raspberry Pi.

O sistema Grove é um sistema que utiliza cabos padrão Grove e conectores de 4 pinos, simplificando o esquema de conexão e possibilitando a comunicação entre dispositivos com interfaces como I2C, UART e outras (FILIPEFLOP, 2020).

Figura 20 – Linha de LED RGB x10 WS2813 Grove



Fonte: *FilipeFlop (2020)*

Especificações:

- Modelo: WS2813 mini;
- LED: 3535 RGB SMD;
- Tensão de operação: 5V;
- Potência: 0,3W;

- Número de LEDs: 10;
- Tipo de LED: RGB alto brilho;
- Dimensões: 80 x 10 x 3 mm.

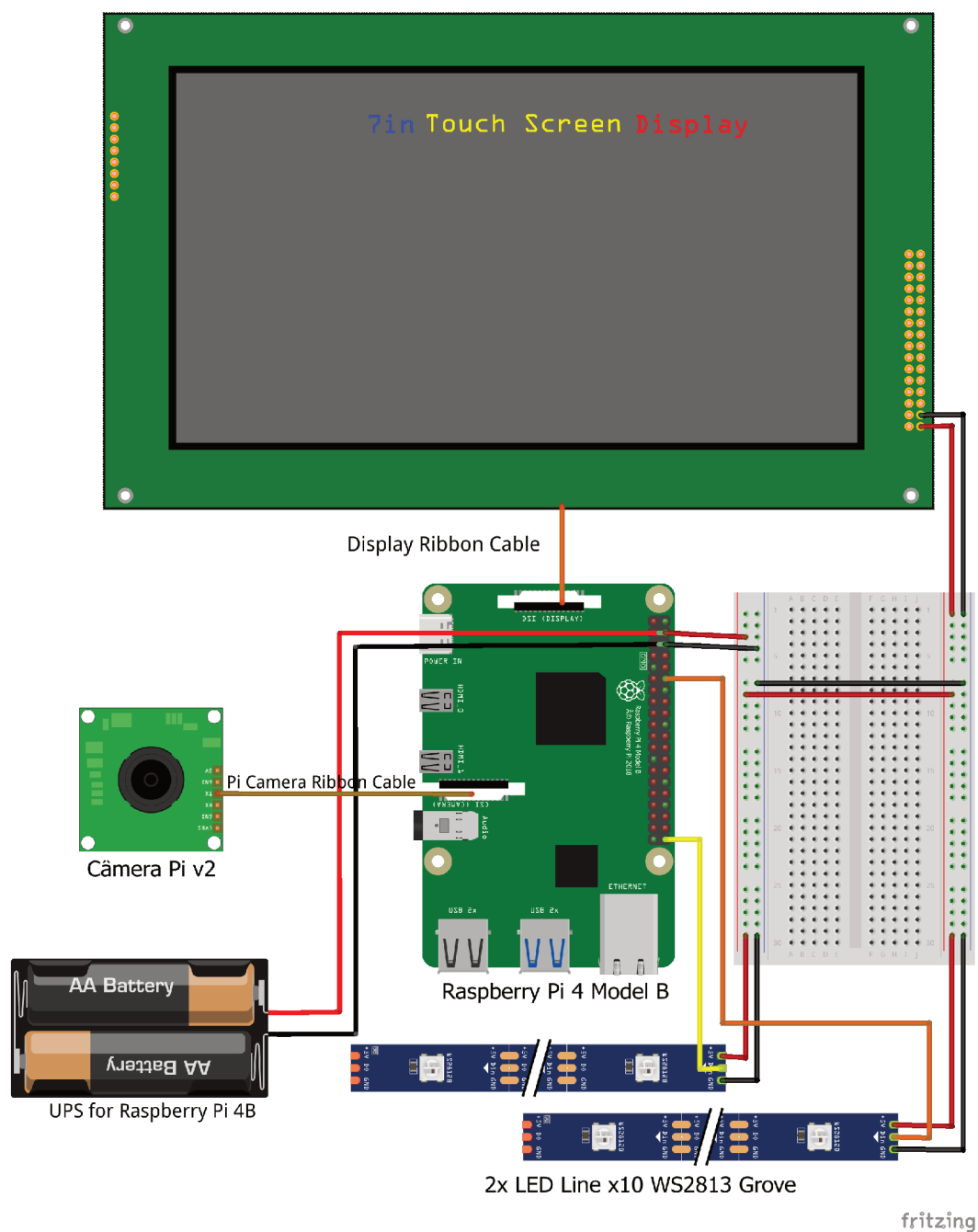
4.1.6 Desenvolvimento do sistema embarcado

Neste tópico, será detalhado a construção do hardware do sistema embarcado através da junção dos dispositivos supracitados, que integrados ao software responsável pelo processamento das imagens capturadas, resultaram na construção de um sistema embarcado de baixo custo capaz de satisfazer os objetivos deste trabalho.

A definição para sistema embarcado consiste na capacidade de colocar recurso computacional dentro de um circuito integrado, equipamento ou sistema. Esta é uma definição para o que é um sistema embarcado. É um sistema completo e independente, mas preparado para realizar apenas uma determinada tarefa. O usuário final não terá acesso ao programa que foi embutido no dispositivo, mas poderá interagir com o equipamento através de interfaces como teclados, *displays*, etc. desde que o sistema tenha sido projetado para tanto (CUNHA, 2007).

A Figura 21 a seguir, ilustra de maneira genérica a interconexão entre os dispositivos utilizados para a construção do protótipo, a fonte de alimentação ininterrupta foi representada de maneira genérica por não possuir biblioteca na ferramenta de diagramação utilizada.

Figura 21 – Representação genérica do sistema embarcado desenvolvido



4.2 Desenvolvimento do Software

O desenvolvimento do software foi dividido em duas etapas principais:

1. Desenvolvimento de uma API (Interface de Programação de Aplicações) para fornecer uma forma eficaz de exportar as medições obtidas e facilitar a integração com os sistemas existentes da empresa que fomentou o desenvolvimento deste projeto.

2. Desenvolvimento de um aplicativo de visão computacional para a obtenção das dimensões superficiais da casca de ovos de galinha.

4.2.1 Banco de dados

Para armazenar as informações de medição, foi utilizado o banco de dados relacional SQLite 3.32.3. O SQLite foi escolhido por ser uma biblioteca em linguagem C que implementa um banco de dados SQL embutido, garantindo acesso a banco de dados SQL sem executar um processo Sistema de Gerenciamento de Banco de Dados (SGBD) separado. O banco de dados consiste em apenas uma tabela responsável por armazenar as informações relacionadas às medições realizadas. A estrutura desta tabela é vista na Figura 22.

Figura 22 – Parâmetros armazenados no banco de dados

Measurements	
id	INTEGER PRIMARY KEY AUTO_INCREMENT
width	REAL
height	REAL
highest_height	REAL
lowest_height	REAL
img_path	STRING(255)
date	DATETIME

O fluxo de informações entre os sistemas e o banco de dados é mostrado na figura 23.

4.2.2 Desenvolvimento da API

A tecnologia escolhida para o desenvolvimento da API foi a linguagem de programação Python em sua versão estável 3.8.3. O principal objetivo da API é fornecer uma forma simplificada de permitir a integração entre o produto a ser entregue e qualquer outro software existente.

4.2.3 Algoritmo de Visão Computacional

Para manter um padrão de desenvolvimento, utilizar-se-á a linguagem Python para desenvolver a solução de visão computacional em conjunto com a biblioteca OpenCV versão 4.3.0. A Figura 24 mostra o esquema que será seguido pelo algoritmo de processamento de imagens. As etapas do fluxo de trabalho são adequadamente descritas abaixo:

- **Captura da imagem RGB:** Nesta etapa, o ovo será colocado na gaveta onde será posicionado a 15 cm de distância da câmera para a captura. A gaveta será

Figura 23 – Fluxo de informações entre os sistemas e o banco de dados

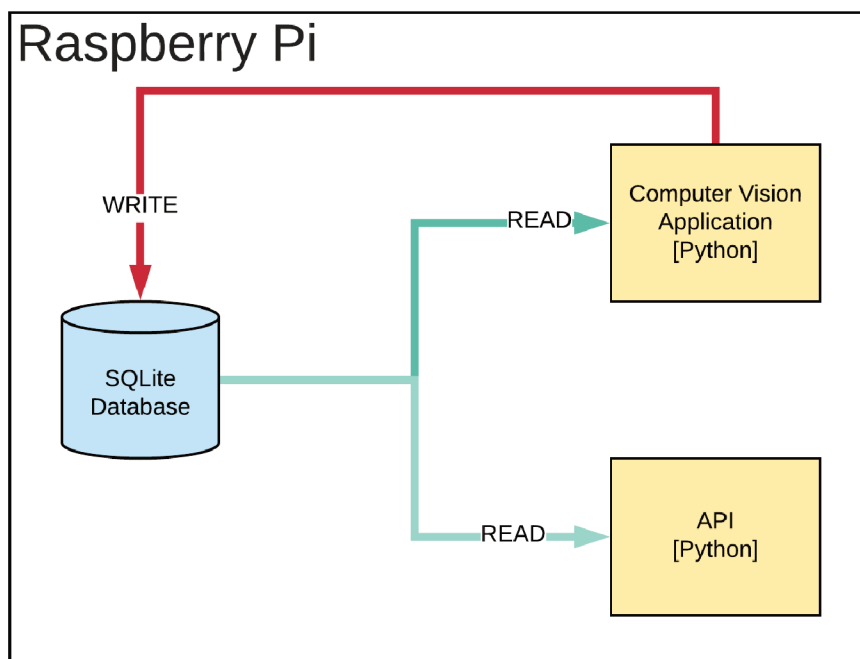
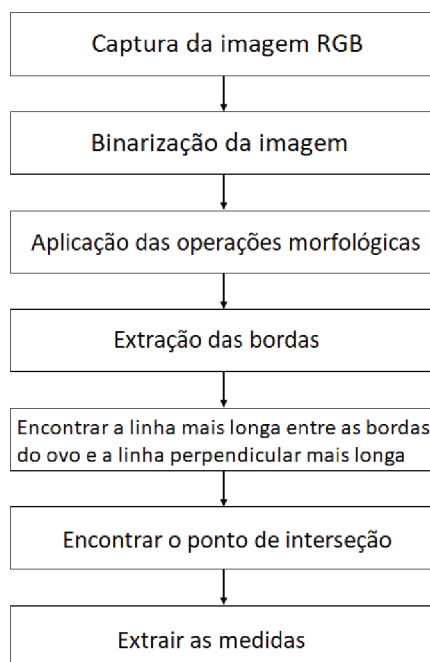


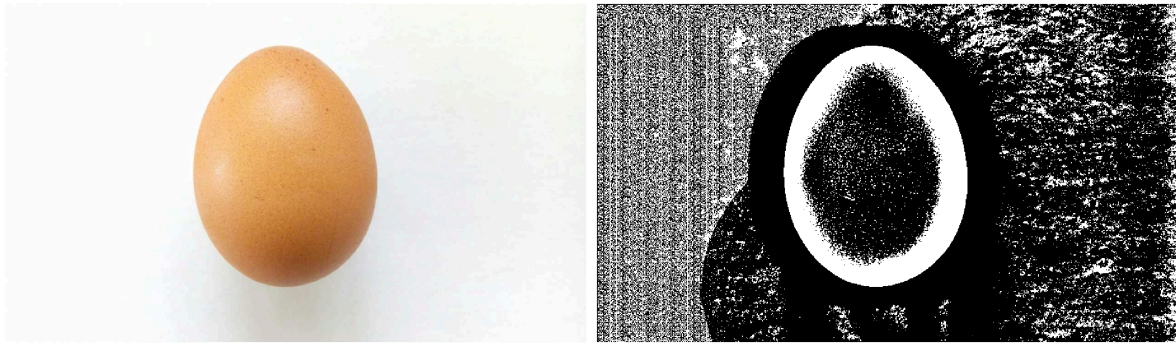
Figura 24 – Fluxo de trabalho da aplicação de visão computacional



iluminada com LEDs para evitar qualquer presença de sombra que afete a qualidade da imagem. A base para posicionamento do ovo terá uma cor contrastante para facilitar os processos de segmentação e evitar a presença de ruídos nas imagens.

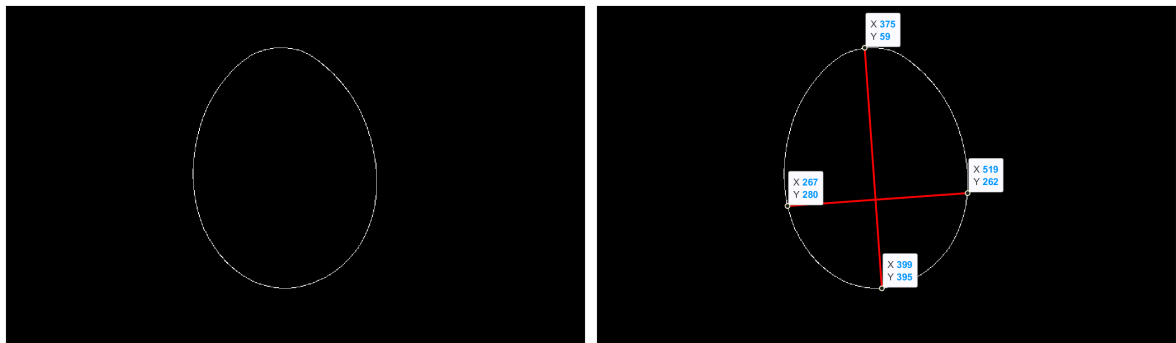
- **Binarização da Imagem:** Após a captura, a imagem será transformada em tons de cinza com base na média entre os três canais (Vermelho, Verde e Azul). Posteriormente, esta imagem será binarizada usando o método de limiar adaptativo. Esse método foi escolhido por garantir uma qualidade superior na identificação das bordas da imagem (quando comparado aos métodos de limiar global). Na Figura 25b, um exemplo de binarização de imagem é fornecido.
- **Aplicação das operações morfológicas:** Na terceira etapa, uma série de operações morfológicas (erosão, dilatação, entre outras) serão aplicadas à imagem.
- **Extração das bordas:** Na etapa de extração das bordas, é aplicado o método de Sobel para obter o contorno do ovo (Figura 25c).
- **Encontrar a linha mais longa entre as bordas do ovo e a linha perpendicular mais longa:** Com as bordas do ovo definidas, uma busca é realizada entre todos os pontos na borda para aqueles que têm a maior distância entre si. Depois de encontrar este ponto, todas as linhas perpendiculares a ele são traçadas e a linha mais longa é escolhida (Figura 25d). Essas duas linhas serão utilizadas para extrair as medidas desejadas.
- **Encontrar o ponto de interseção e extrair as medidas:** Neste estágio, o ponto de interseção entre as duas linhas é encontrado e as medidas (como visto na imagem 25d) são extraídos.

Figura 25 – Exemplo de processamento



(a) Etapa 01 - Captura de imagem RGB

(b) Etapa 02 - Binarização de imagem



(c) Etapa 04 - Extração de bordas

(d) Etapa 06 - Localização da maior dimensão

4.2.4 Desenvolvimento da Aplicação

Para facilitar a interação do usuário com o sistema embarcado foi desenvolvido uma aplicação *web*, outrora denominada IHM, hospedada diretamente no sistema embarcado, onde é possível iniciar uma nova captura de imagem, analisar os resultados obtidos através do próprio dispositivo e alterar as configurações do sistema, como pode ser observado respectivamente nas Figuras 26 - 28.

Figura 26 – Página inicial da aplicação

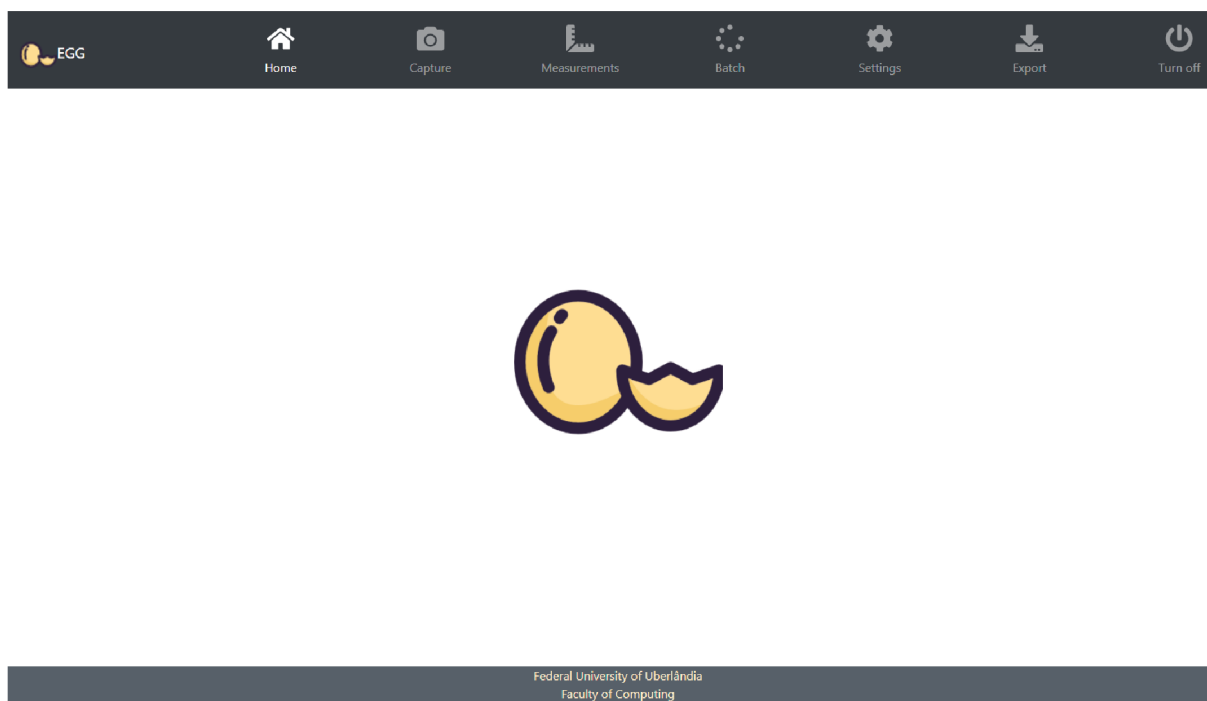


Figura 27 – Página responsável pela função de captura das imagens dos ovos

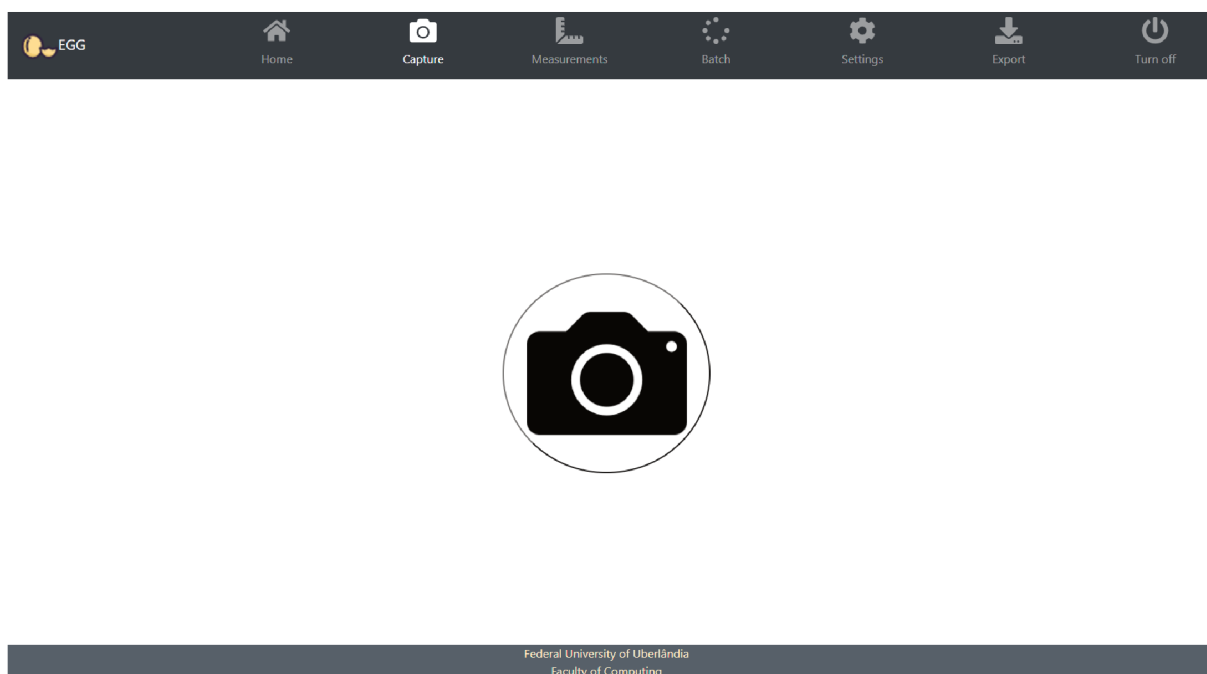


Figura 28 – Exemplo de resultados alocados no banco de dados

Identifier	A mm	B mm	C mm	D mm	Measurement date	Image
209 Delete	65.424	49.758	36.15	29.274	2021-09-19 13:38:52	
208 Delete	65.424	49.758	36.15	29.274	2021-09-19 13:38:31	

Além disso, a aplicação possui dois modos de operação, captura individual ou captura em lote, representados por *real-time* e *batch* nas configurações do sistema, como pode ser visto na Figura 29.

Figura 29 – Configurações do sistema embarcado

Settings

Type of Processing:

Metric:

Timezone:

Prefix:
Available variables: [date] and [id]. Example: egg_[date]_[id]

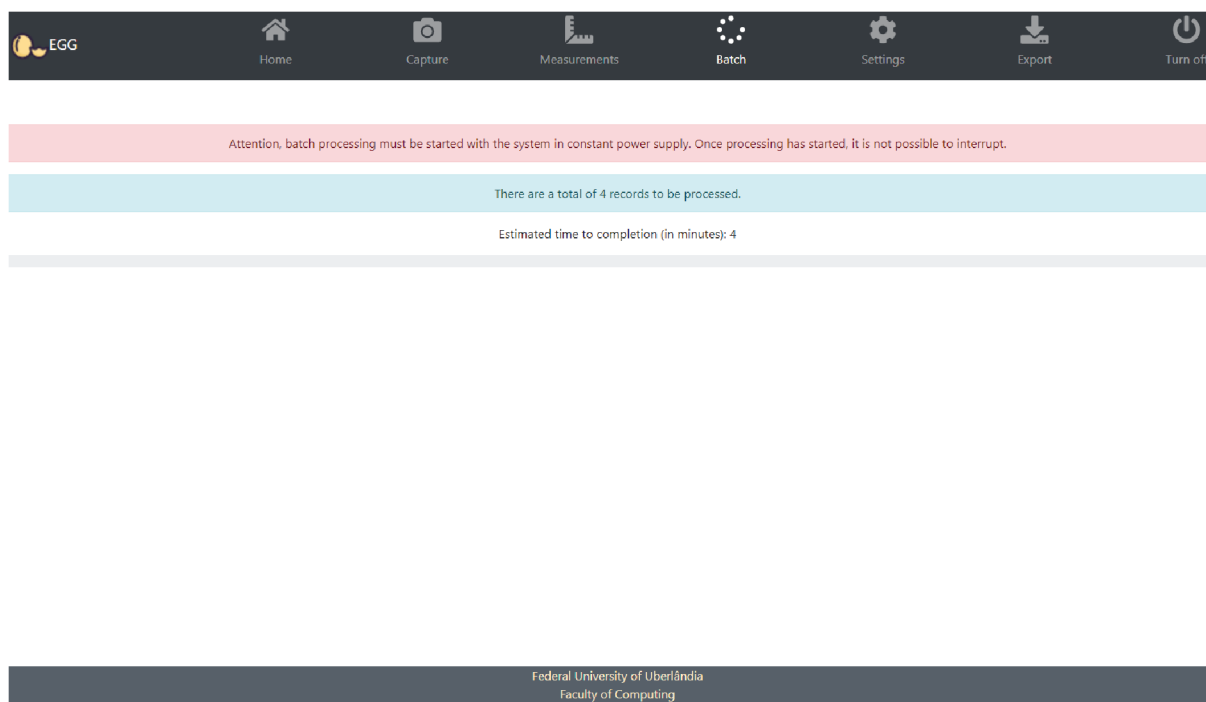
Precision in floating point values:

Approximation Factor:

O modo de operação em tempo real, captura uma imagem, realiza o processamento digital da imagem a fim de obter as dimensões do ovo e exibe o resultado individualmente.

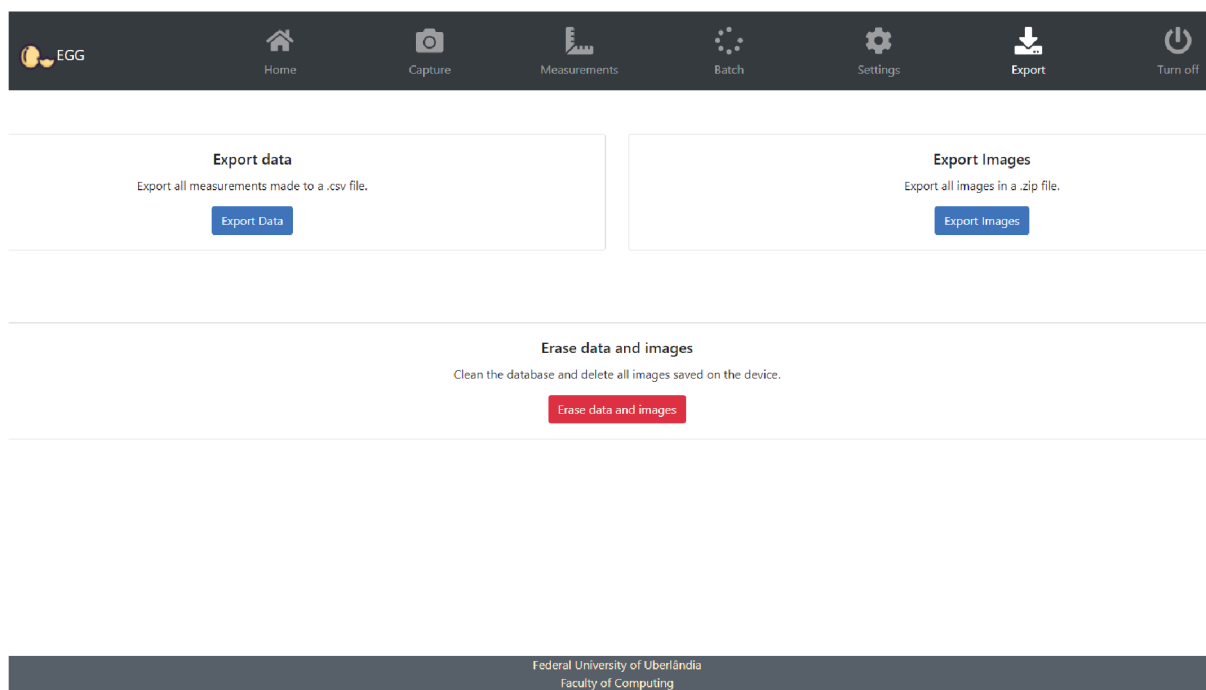
Por outro lado, o modo de operação em lote consiste na captura de múltiplas imagens, quantas forem necessárias para o usuário, para posterior processamento de todo o lote. A Figura 30 exemplifica a página responsável por realizar o processamento em *batch*.

Figura 30 – Página com a função de processamento em lote



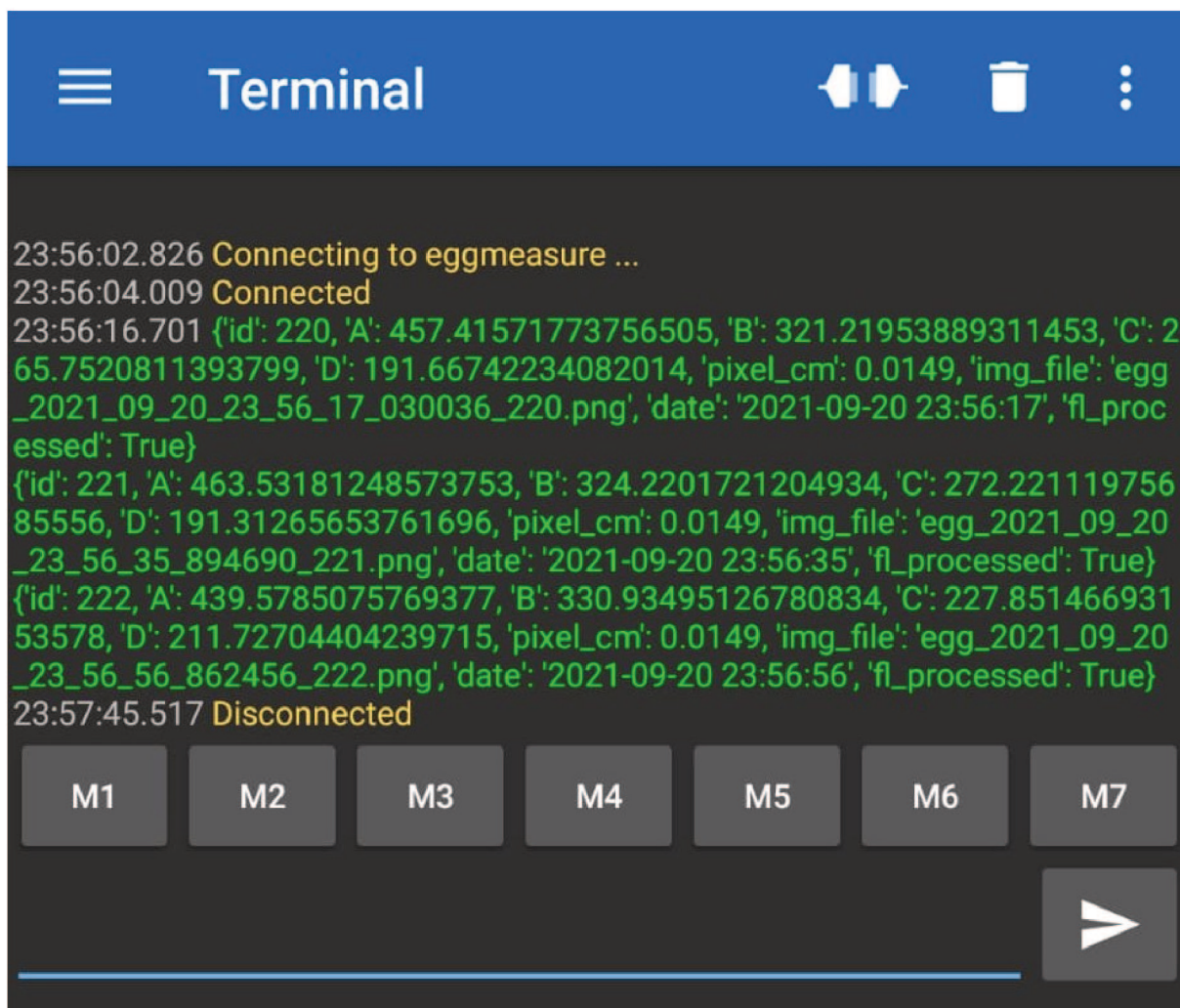
A aplicação ainda oferece a possibilidade de exportar os resultados processados através de um arquivo .csv, exportar todas imagens capturadas por meio de um .zip e limpar todo o banco de dados, como ilustra a Figura 31.

Figura 31 – Página responsável pela exportação dos resultados, imagens e limpeza do banco de dados



4.2.5 Sistema de comunicação *Bluetooth*

Além da possibilidade de extrair os resultados processados pelo sistema através de uma planilha no formato .csv, foi desenvolvido a funcionalidade de enviá-los através do protocolo de comunicação *Bluetooth* a fim de facilitar a importação dos dados para outras plataformas, a Figura 32 ilustra como os dados são exportados via *Bluetooth*.

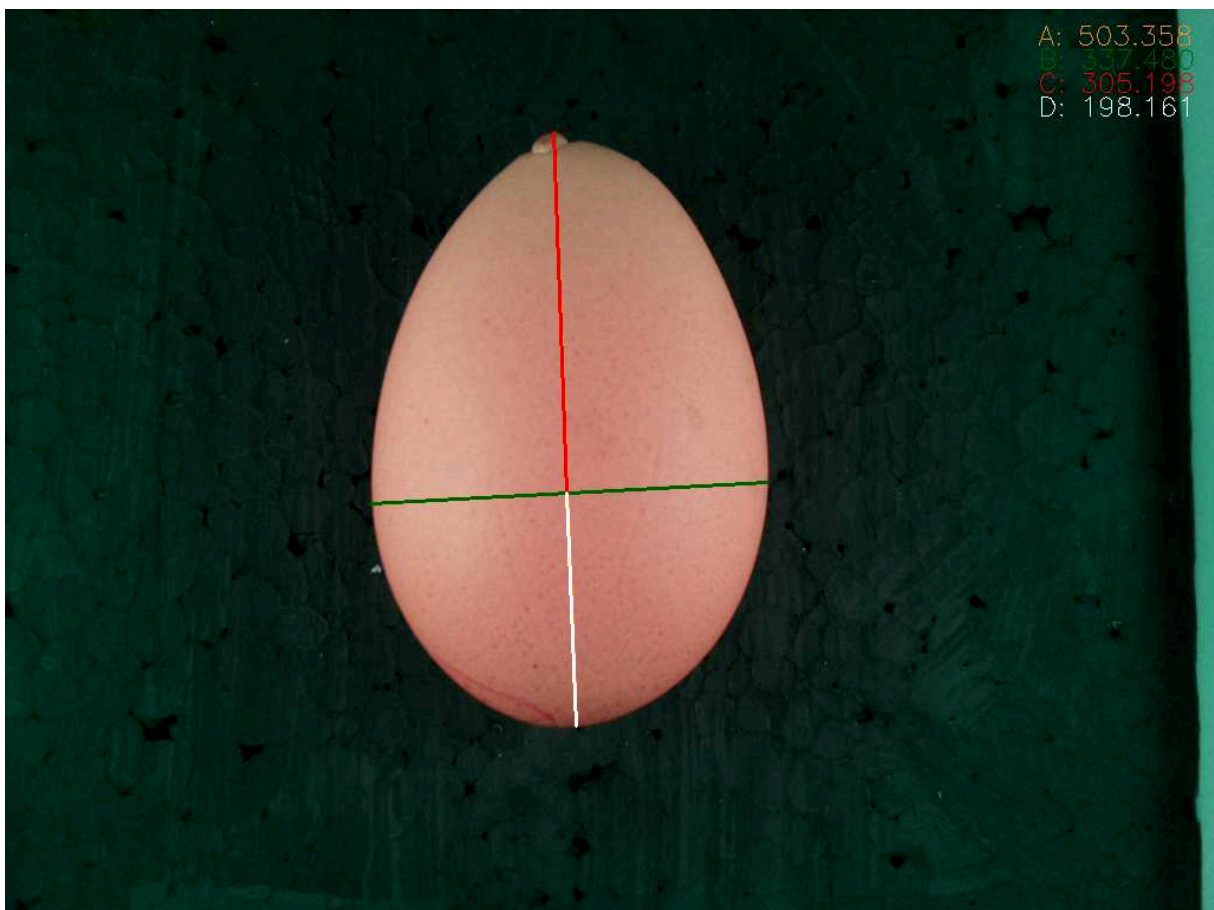
Figura 32 – Exemplo de dados enviados via *Bluetooth* após processamento

5 Resultados e Discussões

Durante os testes o objetivo foi minimizar ruídos externos que poderiam prejudicar a qualidade das imagens capturadas, maximizar a resolução de captura da câmera e o desempenho do dispositivo para evitar gargalos na produção, e refinar o algoritmo de visão computacional para garantir assertividade nos resultados apresentados pelo sistema embarcado.

Além disso, foi possível validar a robustez do algoritmo desenvolvido através de variações nas características dos ovos analisados. A Figura 33 exibe um caso de teste, onde a maior distância longitudinal foi traçada até uma protuberância presente na superfície da casca do ovo.

Figura 33 – Exemplo de ovo marrom anormal processado pelo sistema



A fim de garantir a exatidão do método de visão computacional foi analisado uma amostra de 50 ovos de galinha distintos, com variações de tamanho e tonalidades. Comparando os resultados do algoritmo com os obtidos pelo método manual foi possível

determinar o erro médio entre os dois métodos e o desvio padrão da amostra, conforme Tabela 1.

Tabela 1 – Comparação estatística entre os dois métodos de medição em milímetros

	Eixo Transversal	Eixo Longitudinal
Erro médio	5,4800 ± 0,5406	7,1800 ± 0,6079

Os resultados apresentados na Tabela 1 foram avaliados pelos especialistas que consideraram o erro aceitável comparado com o erro humano da medição. Cabe destacar que, devido a ajustes na posição da câmera e modificações na base para fixação do ovo, a distância entre a câmera e o ovo pode se alterar, o que faz-se necessário uma nova calibração do aparelho. Hoje, essa calibração é realizada através de um parâmetro de configuração no software do equipamento, responsável por definir a relação entre pixel por centímetro utilizada no dimensionamento da imagem processada. Sendo assim, a ausência da calibração prévia antes dos testes, pode ter impactado negativamente os resultados.

Além disso, os dispositivos escolhidos para construção do sistema embarcado desempenharam suas funções com excelência, a capacidade de processamento não impactou a operação, não foram gerados gargalos no decorrer do experimento. A câmera utilizada foi capaz de capturar imagens de alta resolução com o auxílio do sistema de iluminação construído, responsável por eliminar possíveis fontes de ruído na fotografia. O Raspberry Pi 4B não apresentou nenhuma falha de comunicação com os outros dispositivos durante a execução do algoritmo e manteve a qualidade dos resultados com o passar do tempo.

6 Conclusão e Trabalhos Futuros

O desenvolvimento do presente estudo proporcionou a análise de como dispositivos de baixo custo podem ser eficientes na automação de processos. Através deste trabalho garante-se ainda a motivação de indústrias avícolas a investirem em inovações tecnológicas com o intuito de aprimorar seus processos através dos seus centros de pesquisa.

O mecanismo desenvolvido para correção da posição do ovo a fim de adquirir as imagens teve sua proposta validada através do experimento realizado. A câmera utilizada responsável por capturar imagens de alta resolução se provou eficaz durante a operação, justificando a sua escolha e satisfazendo o objetivo proposto, assim como o computador para embarcar o sistema desenvolvido.

Tendo em vista os resultados apresentados pode-se afirmar que o sistema embarcado de baixo custo desenvolvido neste estudo provou-se efetivo na detecção e dimensionamento das medidas transversais e longitudinais da casca de um ovo de galinha através de visão computacional. Ademais, a interface homem máquina proporcionou a interação do usuário com o sistema de forma eficaz através da aplicação *web*, facilitando a interpretação e manipulação dos dados.

Dada a importância do assunto, sugere-se em trabalhos futuros configurar uma rotina de calibração prévia à operação do sistema e ampliação do dispositivo para inspeção simultânea de múltiplos ovos.

Nesse sentido, a utilização de dispositivos de baixo custo permitem acompanhar os avanços tecnológicos com investimento inicial menor e sem gastos adicionais em software e atualizações. Desta forma se torna rentável para o empresário e simples para o programador, motivando a ambos.

Referências

- 52PI WIKI CONTRIBUTORS. *UPS for Raspberry Pi 4B/3B+/3B SKU: EP-0114*. [S.l.], 2020. Disponível em: <https://wiki.52pi.com/index.php/UPS_for_RaspberryPi4B/3B+/3B_SKU:_E_P_-_0114>. Citado 3 vezes nas páginas 31, 32 e 33.
- CÂMARA, G. et al. Spring: Integrating remote sensing and gis by object-oriented data modelling. *Computers & graphics*, Elsevier, v. 20, n. 3, p. 395–403, 1996. Citado na página 16.
- CEDROTECH. *OpenCV: Uma breve introdução à visão computacional com python*. [S.l.], 2018. Disponível em: <<https://blog.cedrotech.com/opencv-uma-breve-introducao-visao-computacional-com-python/>>. Citado na página 18.
- COPADATA. *Interface homem-máquina (HMI)*. [S.l.], 2017. Disponível em: <<https://www.copadata.com/pt/solucoes-hmi-scada/interface-homem-maquina-hmi/>>. Citado na página 18.
- CUNHA, A. F. O que são sistemas embarcados. *Saber Eletrônica*, v. 43, n. 414, p. 1–6, 2007. Citado na página 37.
- FAVARI, F. *Qual a diferença entre uma tela LCD e uma TFT?* [S.l.], 2021. Disponível em: <<https://quatorrodas.abril.com.br/auto-servico/o-que-e-uma-tela-tft-e-quais-as-suas-vantagens/>>. Citado na página 33.
- FILHO, O. M.; NETO, H. V. *Processamento digital de imagens*. [S.l.]: Brasport, 1999. Citado 2 vezes nas páginas 16 e 17.
- FILIPEFLOP. *Display Raspberry Pi Touchscreen 7*. [S.l.], 2019. Disponível em: <<https://www.filipeflop.com/produto/display-raspberry-pi-touchscreen-7/>>. Citado na página 34.
- FILIPEFLOP. *Linha de LED RGB x10 WS2813 Grove*. [S.l.], 2020. Disponível em: <<https://www.filipeflop.com/produto/linha-de-led-rgb-x10-ws2813-grove/>>. Citado na página 36.
- FILIPEFLOP. *Câmera Raspberry Pi v2 8MP*. [S.l.], 2021. Disponível em: <<https://www.filipeflop.com/produto/camera-raspberry-pi-v2-8mp/>>. Citado na página 35.
- FLATECK. *IHM - INTERFACE HOMEM MÁQUINA*. [S.l.], 2017. Disponível em: <<http://flateck.com.br/servicos/detalhe/ihm-interface-homem-maquina/>>. Citado na página 18.
- GONZALEZ, R.; WOODS, R.; EDDINS, S. Image processing place. *Diambil kembali dari Image Database: http://www.imageprocessingplace.com/downloads_V3/root_downloads/image_databases/standard_test_images.zip*, 2012. Citado na página 16.

LIMA, A. d. S. Desenvolvimento de um sistema acústico para detecção de trincas em ovos. 2018. Citado na página 13.

MONK, S. Programando o raspberry pi: primeiros passos com python. *São Paulo: Novatec*, 2013. Citado na página 29.

OMID, M. et al. An expert egg grading system based on machine vision and artificial intelligence techniques. *Journal of Food Engineering*, v. 118, n. 1, p. 70–77, 2013. ISSN 0260-8774. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0260877413001404>>. Citado 3 vezes nas páginas 19, 20 e 21.

PYSCIENCE-BRASIL. *Python: O que é? Por que usar?* [S.l.], 2016. Disponível em: <<http://pyscience-brasil.wikidot.com/python:python-oq-e-pq>>. Citado na página 17.

RASPBERRY PI FOUNDATION. *RASPBERRY PI PRODUCTS*. [S.l.], 2017. Disponível em: <<https://www.raspberrypi.org/products/>>. Citado 2 vezes nas páginas 29 e 31.

ROSA, G. M. D. Egg evaluation: ovoscopia via smartphone. 2017. Citado 4 vezes nas páginas 19, 24, 25 e 26.

SABER ELÉTRICA. *IHM - APLICAÇÃO COM CLP EM PROJETOS DE AUTOMAÇÃO*. [S.l.], 2017. Disponível em: <<https://www.sabereletrica.com.br/o-que-e-ihm/>>. Citado na página 18.

VASILEVA, A. V. et al. Assessing exterior egg quality indicators using machine vision. *British Poultry Science*, Taylor & Francis, v. 59, n. 6, p. 636–645, 2018. PMID: 30221543. Disponível em: <<https://doi.org/10.1080/00071668.2018.1523535>>. Citado 4 vezes nas páginas 19, 22, 23 e 24.