**Uma imagem contendo Linha do tempo

Descrição gerada automaticamente**

**Aplicação de Processamento de Imagens para Detecção e Contagem de ovos do Camarão Macrobrachium amazonicum**

**Rhuã Yuri Nascimento Sardinha**

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO MARANHÃO**

**PRÓ-REITORIA DE PESQUISA, PÓS-GRADUAÇÃO E INOVAÇÃO**

**PROGRAMA INSTITUCIONAL DE BOLSAS DE INICIAÇÃO CIENTÍFICA**

**TÍTULO**: Aplicação de Processamento de Imagens para Detecção e Contagem de ovos do Camarão Macrobrachium amazonicum

**ALUNO**: Rhuã Yuri Nascimento Sardinha

**Programa**: Programa Institucional de Bolsas de Iniciação Científica – PIBIC EM

**CURSO**: TÉCNICO EM INFORMÁTICA

**CAMPUS**: ITAPECURU – MIRIM

**ORIENTADOR**: Prof. Ms.Dejailson Nascimento Pinheiro

**Itapecuru – Mirim, 23 de março de 2021**

**SUMÁRIO**

[***RESUMO 4***](#_2et92p0)

[***1.***](#_tyjcwt) ***INTRODUÇÃO 5***

[**1.1.**](#_3dy6vkm) **Objetivo 6**

[**1.1.1. Objetivos Específicos**](#_1t3h5sf) **6**

[**1.2. Processamento de imagem digital**](#_z79x6fq8794m) **7**

[***2.***](#_4d34og8) ***METODOLOGIA* 8**

[**2.1.**](#_huzwlgvdb805)**Ferramentas do Processamento de Imagem 10**

[**2.1.1.**](#_5bmhfhh0ssi9) **OpenCV 10**

[**2.1.1.1.**](#_dp40m5n6t0tq) [**Filtragem da Imagem**](#_yyon74avf9u7) **11**

[**2.1.1.2.**](#_p7dp3nhx4bbz)  [**Processamento de Histograma**](#_vu5nzthdutnf) **12**

[**2.1.1.3.**](#_yyon74avf9u7) [**Método cvCvtColor**](#_p7dp3nhx4bbz) **12**

[**2.1.1.4.**](#_qpmu9esxopxi) [**Limiarização de Otsu**](#_qpmu9esxopxi) **13**

[**2.1.1.5.**](#_eds4kcu4vgyj)  [**Método de Canny**](#_eds4kcu4vgyj) **14**

[**2.1.1.6.**](#_3gio8q51xzki) [**Detecção de círculos**](#_3gio8q51xzki) **15**

[**2.1.2.**](http://scikit-image)[**Scikit-image**](#_egto9zwo7p4q) **16**

[**2.1.2.1.**](#_is9j4ydpqtr2) [**Método regionprops**](#_is9j4ydpqtr2) **16**

[**2.2.**](#_zbsh5z1a0ggl)**Metrologia da imagem digital 17**

[**2.3.**](#_ajph57g1xv97)**Etapas para a obtenção dos dados 18**

[***3.***](#_3rdcrjn) ***RESULTADOS* 18**

[***4.***](#_26in1rg) ***DISCUSSÃO 2*1**

***5***[***.***](#_42345kdp7i4e)***CONCLUSÃO 22***

[***REFERÊNCIAS***](#_lnxbz9) **23**

# 

# 

# RESUMO

O processo de contagem de objetos para a extração de informações, é uma tarefa muito comum no nosso dia-a-dia, da mesma forma, muitos trabalhos de pesquisa utilizam deste princípio para alcançar seus objetivos, como é o exemplo do projeto Fecundidade do camarão branco. O projeto visa buscar informações como a taxa de fecundidade das fêmeas, comprimento e peso de cada espécime, além do volume das massas dos ovos e a quantidade de ovos que uma fêmea possui, para que essas informações sejam adquiridas os pesquisadores precisam realizar a contagem dos ovos de camarão, essa atividade de contagem de objetos pode provocar o cansaço visual que ocorre naturalmente, na execução das tarefas manuais repetitivas, além de exigir uma grande quantidade de tempo para concluí-la, acarretando em erros nos dados brutos obtidos, neste sentido o seguinte projeto de pesquisa visa reduzir o tempo de análise das amostras, assim como a eliminação do cansaço visual durante o processo de contagem de objetos e a mensuração dos dados das amostras de ovos de camarão coletados, com o desenvolvimento de uma aplicação que utiliza os conceitos de visão computacional capaz de identificar objetos que estão contidos em imagens digitais. O projeto tem como objetivo desenvolver um software que consiga extrair informações como: a quantidade de ovos de camarão branco contida na imagem, assim como suas dimensões (diâmetro maior e menor) para que assim seja feito o cálculo para sabermos os dados como a média de fecundidade, média da massa dos ovos. O desenvolvimento do programa será realizado com a linguagem de programação Python e a utilização das bibliotecas de código aberto OpenCV, Scikit-image, que ficaram encarregadas de realizarem o processamento e extração dos dados contidos nas imagens e a biblioteca gráfica Tkinter (nativa da própria linguagem python) para a interação do sistema com o usuário.

**Palavras-chave**: Visão computacional, Detecção de objetos, Processamento de imagem de digital.

# INTRODUÇÃO

As tarefas manuais são atividades recorrentes no nosso dia-a-dia, e dependendo do nível de dificuldade ou a quantidade de vezes que elas são feitas, isso pode ser desgastante para qualquer pessoa. No ramo da larvicultura, muitas atividades são feitas de forma manual e repetitivas, tanto o excesso de repetição, quanto o tempo podem cansar visualmente o pesquisador e atrapalhar no levantamento dos dados brutos e consequentemente nos resultados da pesquisa.

Com o uso automação de tarefas repetitivas intermediadas por computador podemos eliminar o cansaço visual que ocorre nesta etapa fazendo com que os dados catalogados sejam mais confiáveis, além disso sistemas computacionais que executam tarefas rotineiras tendem a se tornarem mais eficientes durante sua execução.

Esse tipo de ferramenta vem ganhando mais espaço no mercado, a ponto de vermos sistemas de segurança por reconhecimento facial ou em campos mais específicos como na ciência dos materiais (ALBUQUERQUE et al, 2007) na medicina (VERONEZI et al, 2001) e avicultura (CATANEO, 2017), áreas onde faz-se necessário o uso do sentido da visão para extrair dados de imagens digitais. Este trabalho propõe uma abordagem que emprega métodos e/ou técnicas de processamento de imagem e detecção de objetos para a realização da coleta dos dados.

Para a realização deste trabalho tivemos o como parceria a equipe de pesquisa do curso de Meio Ambiente do Instituto Federal de Educação do Maranhão, campus Itapecuru-Mirim, essa equipe tem como nome de sua pesquisa; “Fecundidade do camarão *Macrobrachium amazonicum*”. O estudo destes pesquisadores tinha como objetivos estudar o camarão de água doce decapoda, que pertence a família *Palaemonidae* (PASCHOAL, 2017), conhecido no nordeste brasileiro como “camarão branco”. Os pesquisadores buscavam obter informações como a taxa de fecundidade destes camarões, assim como o comprimento, peso de cada espécime, volume das massas dos ovos e a quantidade de ovos que cada fêmea possui.

A obtenção dos dados citados é realizada com técnicas manuais e com o auxílio de um microscópio e um paquímetro, a metodologia utilizada pela equipe se dava em remover os ovos das fêmeas e em seguida armazená-los em recipientes plásticos junto com álcool 70% para mantê-los preservados e para que os mesmos se desafixarem um dos outros. Para a contagem dos ovos é utilizado um recipiente de vidro com linhas tracejadas formando pequenos quadrados, os ovos são despejados nesse recipiente e organizados nos quadrados, e após isso é feito a contagem das amostras. A metodologia utilizada pelos pesquisadores necessita de uma grande quantidade de tempo, pois são analisadas cerca de 30 fêmeas ovígeras mensalmente, e para cada fêmea de camarão branco é analisada a taxa de fecundidade, a massa dos ovos dentre outros aspectos. A execução destas tarefas acarreta uma fadiga visual para os pesquisadores que necessitam repetir a atividade inúmeras vezes, neste sentido buscamos realizar o desenvolvimento do software para realizar a contagem dos ovos existentes em cada imagem digital aplicando os métodos de processamento de imagem para a obtenção dos dados relacionados aos ovos de camarão.

## Objetivo

Esta proposta visa o desenvolvimento de um software que utilize métodos de processamento de imagem digital e detecção de objetos para fazer análises em amostras de ovos de camarões da espécie camarão branco.

## Objetivos Específicos

* Realizar um levantamento bibliográfico das técnicas de processamento de imagens digitais, e assim, determinar a melhor combinação para realizar o tratamento de imagens que possibilitem a obtenção de uma imagem resultante adequada para análise final do processamento.
* Selecionar as técnicas de processamento de imagens digitais para a determinação da que melhor se aplica para o tratamento de imagens.
* Aplicar o conjunto de funcionalidades disponíveis na API (*Application Programming Interface*) da biblioteca *open source* OpenCV[[1]](#footnote-1) e Scikit-Image[[2]](#footnote-2), para realizar o tratamento e processamento das imagens.
* Desenvolver uma aplicação de baixo custo para realizar a contagem e o cálculo volumétrico da massa dos ovos presentes nas imagens digitais.
* Realizar uma análise comparativa dos resultados obtidos pela solução de software com os resultados do processo de contagem pelo método manual de ovos do camarão branco.

## Processamento de imagem digital

Ao falarmos de processamento de imagem, primeiro é preciso entender o que é uma imagem, segundo Gonzalez e Woods (2008), uma imagem pode ser definida como “uma função bidimensional, f (x, y), em que x e y são coordenadas espaciais (plano), e a amplitude de f em qualquer par de coordenadas (x, y) é chamada de intensidade ou nível de cinza da imagem nesse ponto. Quando os valores de x, y e f são finitos, temos uma imagem digital”.

Já o processamento digital de imagens se refere ao processamento de imagens digitais por um computador digital (Gonzalez; Woods, 2008), embora pareça redundante, o processamento de imagem se trata de qualquer tratamento efetuado em uma imagem, seja para melhorar sua qualidade ou na aplicação de métodos que façam com que ela seja enviada da melhor forma possível por exemplo. A origem do tratamento de imagem se dá no início da década de 1920, com um sistema de cabos oceânicos que enviavam imagem entre Nova York e Londres, esse sistema foi revolucionário, pois fazia com que fotografias que antes demoraria mais de uma semana para cruzar o atlântico fossem enviadas em menos de 3 horas, nos dias atuais o podemos ver os métodos de processamento de imagem no reconhecimento facial, na identificação de placas de veículos, no sistema de locomoção de carros automatizados, dentre várias outras utilidades (Gonzalez; Woods, 2008).

# METODOLOGIA

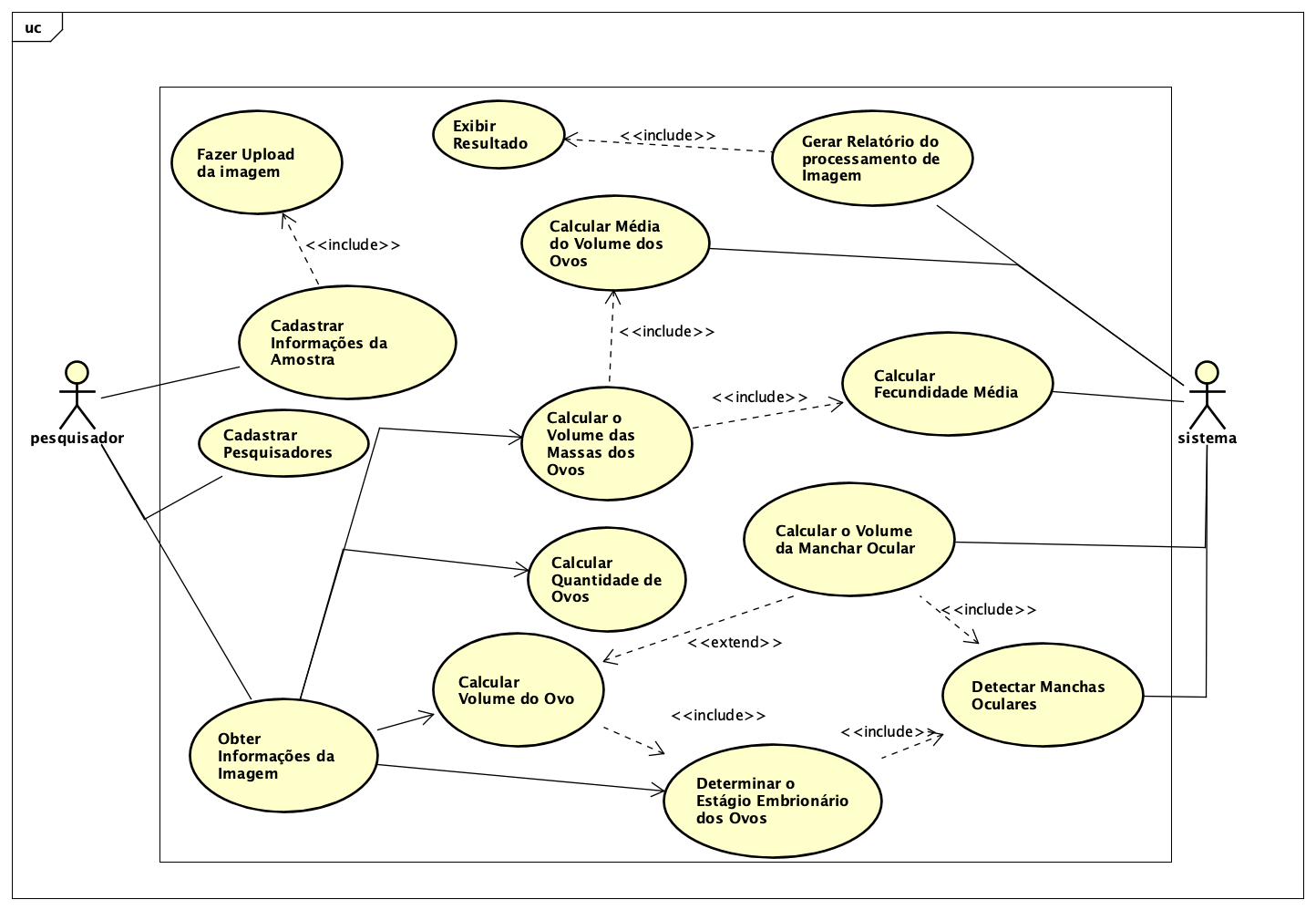
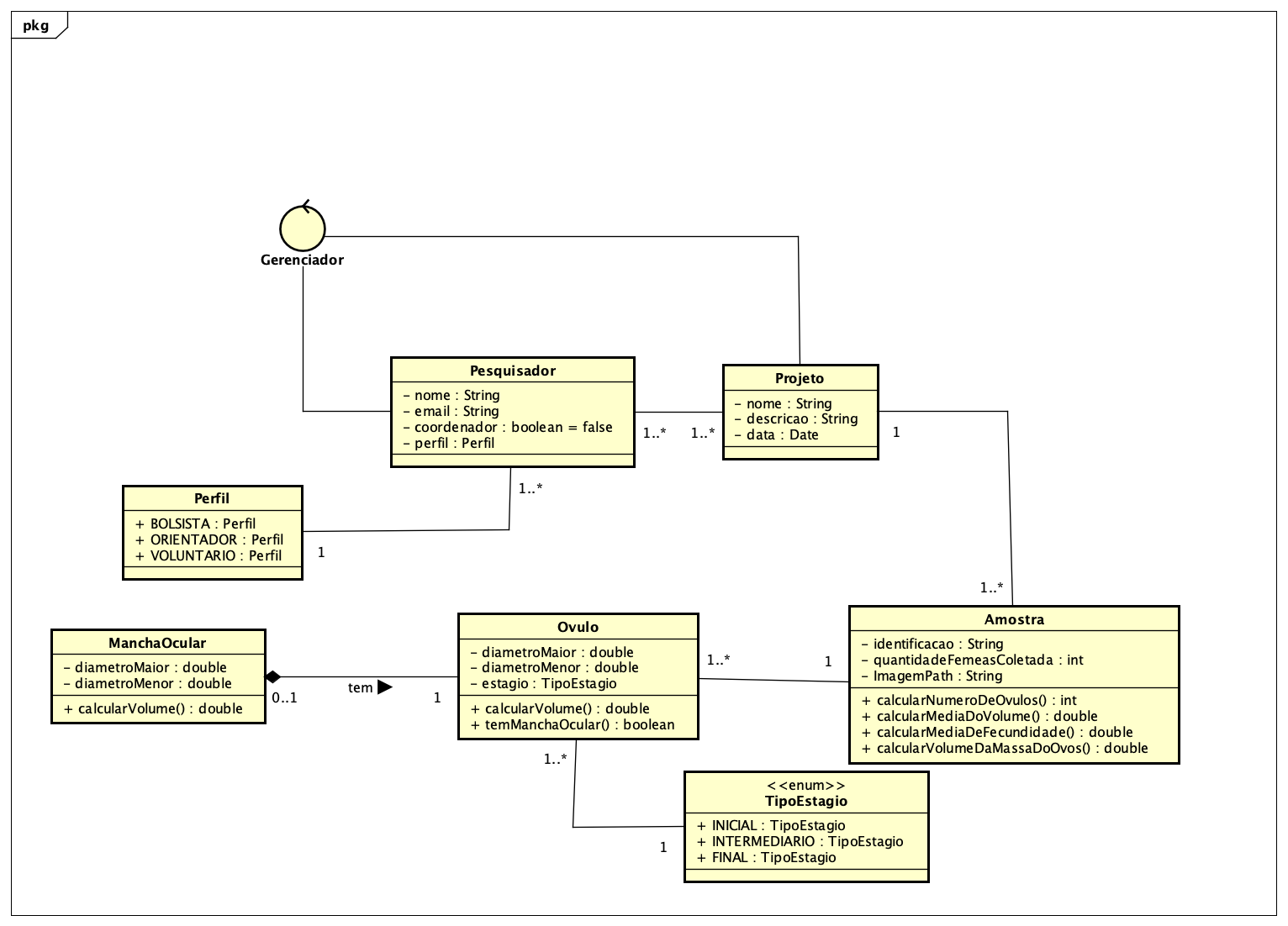
O objetivo do projeto como já mencionado é o desenvolvimento de um software de visão computacional capaz de detectar os ovos de camarão e aplicar os devidos cálculos com os dados captados, para que assim análises que muitas vezes poderiam demorar e cansar o pesquisador pudessem ser feitas de forma mais prática e eficaz, para que esse objetivo fosse concretizado foi realizado um extenso trabalho de pesquisa, que vai da fundamentação do assunto para entendermos quais informações são relevantes para a construção do programa e quais não são.

Realizamos estudos que vão do campo do processamento de imagem ao campo da larvicultura, isso serviu de base para que o trabalho deste projeto fosse realizado da melhor forma possível, dividimos assim as etapas de estudos as quais serão melhores explicadas nos demais parágrafos.

Com base na divisão das etapas, iniciamos um levantamento sobre os requisitos funcionais e não funcionais do sistema, onde utilizamos técnica de levantamento de requisitos (SOMMERVILLE, 2011), além disso, fizemos um estudo sobre larvicultura, para que assim fosse possível ter um entendimento melhor sobre assunto de vital importância para a conclusão desta pesquisa

Com base nos estudos que estavam sendo feitos sobre levantamento de requisitos e sobre o campo da larvicultura, iniciamos uma pesquisa sobre técnicas de desenvolvimento de software, metodologia orientada ao objeto (para isso houve uma revisão da API do Python[[3]](#footnote-3)), além de estudar sobre as técnicas de prototipação de interface gráfica de usuário (PRESSMAN, 2011).

Em paralelo às pesquisas citadas anteriormente, foi feito a construção do modelo da arquitetura do projeto do software, onde com base nesta arquitetura foi elaborado o diagrama de Caso de Uso e o diagrama de Classe e em seguida iniciamos o desenvolvimento dos prototipos da interface gráfica, esta que fica encarregada de receber os dados do usuário e repassar as informações obtidas na análise das imagens.



Com o desenvolvimento da interface gráfica completa fizemos uma pesquisa bibliográfica sobre técnicas de processamento de imagem digital e detecção de objetos, onde escolhemos a biblioteca OpenCV, pela extensa variedade de funções e métodos que ela disponibiliza para o tratamento de imagens digitais, assim sendo possível a análise dos dados brutos contidos nas imagens, e junto com a biblioteca OpenCV utilizamos também a biblioteca Scikit-image*: image processing* in Python, onde as duas trabalham em conjunto para obter as informações contidas nas imagens.

Ao fim desse processo, fizemos a integração das bibliotecas opencv e scikit-image com a interface gráfica do software, com a conclusão dessa etapa o programa desenvolvido, já conseguia aplicar as técnicas de pré-processamento de imagem, segmentação, além do reconhecimento de padrões. Para que o software pudesse fazer estas análises utilizamos técnicas como de redimensionamento de imagem, aplicação de máscaras, filtros, métodos de reconhecimento de padrão dentre várias outras funções que serão melhor detalhadas mais adiante.

## Ferramentas do Processamento de Imagem

Como já mencionado o sistema utiliza como principais bibliotecas para o tratamento da imagem e extração de dados as bibliotecas OpenCV e a Scikit-image, essas duas ferramentas trazem consigo uma extensa variedade de métodos e funções para o processamento de imagem, veremos agora quais desses métodos foram utilizados para o desenvolvimento da aplicação

### OpenCV

Como mencionado anteriormente, para o desenvolvimento do software, foi utilizado a biblioteca opencv que foi desenvolvida pela Intel e a escolha desta biblioteca se deu pelo fato da mesma possuir mais de 500 funções que auxiliam no tratamento de imagens, análise estrutural, análise de movimento e rastreamento de objetos, reconhecimento de padrões, calibração de câmera e reconstrução 3D.

Para o projeto utilizamos alguns passos para o processamento de imagem, sendo eles a etapa de aquisição, onde obtemos a imagem com um diálogo entre a interface gráfica e o usuário que faz com que o software busque o caminho da imagem no sistema de arquivos do computador, logo após utilizamos os métodos de pré-processamento de imagem, como a aplicação de filtros e máscaras para que seja possível a remoção de ruídos e a transformação da imagem do padrão BGR (blue, green, red) para BGRA (blue, green, red, alfa). Os métodos para a filtragem dos ruídos e a transformação dos canais de cor serão melhor explicados mais adiante. Após a aplicação dos filtros e a transformação da imagem, obtemos uma imagem com os ovos mais azulados, assim podemos aplicar a técnica de equalização de histograma para remover ruídos que permaneceram mesmo após o primeiro tratamento e para destacar os ovos em relação ao fundo.

#### Filtragem da Imagem

O pré-processamento da imagem tem como objetivo remover os ruídos e consequentemente melhorar a qualidade da imagem para as demais etapas do processamento de imagem, para que isso seja possível utilizamos métodos de suavização de imagem com o filtro gaussiano, esse filtro percorre a imagem agrupando os pixels em uma matriz 5 por 5, onde é obtido a média de intensidade dos pixels, assim os valores de cada ponto da matriz 5x5 são alterados para a média dos pontos.

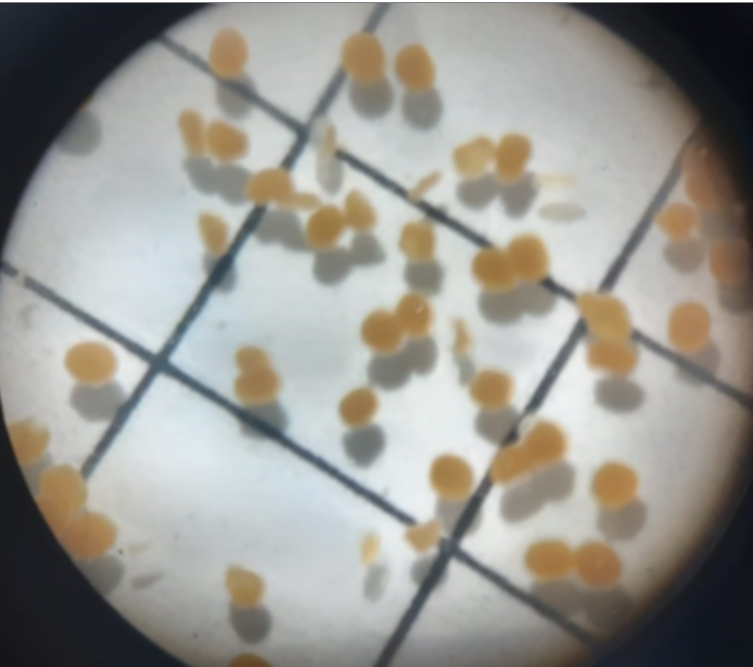


Figura 1.

Como pode ser visto na figura 1, obtemos com esse processo uma imagem com os objetos meios borrados, isto serve para que métodos como canny e de limiarização que serão melhor explicados mais à frente possam distinguir melhor as bordas dos objetos.

#### Processamento de Histograma

O histograma de uma imagem são os diferentes níveis de intensidade no intervalo [0, L – 1] é uma função discreta h(rk) = nk, onde rk é o k-ésimo valor de intensidade e nk. Imagens com muitos ruídos tendem a ter muitas divergências no nível de intensidade de seus pixels, essas diferenças podem atrapalhar na obtenção dos dados, para que isso não ocorra utilizamos o método de equalização de histograma, que tem como objetivo normalizar essas diferenças, para que isso seja possível precisamos dividir rk pelo o produto de MN que são as dimensões das linhas e colunas.

p(rk ) = rk /MN para k = 0, 1, 2, … , L – 1 (Gonzalez; Woods, 2008).

Por fim chegamos ao resultado de p(rk) que é uma estimativa da probabilidade de ocorrência do nível de intensidade rk em uma imagem. Com a biblioteca OpenCV podemos fazer a equalização de uma imagem utilizando o método *cv.equalizeHist*, que recebe como parâmetro a imagem a ser processada.

#### Método cvCvtColor

Após a imagem receber um tratamento de filtragem e a aplicação da máscara para remover a grande região preta deixada pela lente do microscópio, aplica-se o método de transformação de imagem com o cvCvtColor, onde o mesmo recebe como parâmetros:

* **src:** imagem de entrada
* **dst:** imagem de saída do mesmo tamanho e profundidade que src
* **code:** código de conversão do espaço de cores
* **dstCn:** número de canais na imagem de destino e é um atributo opcional.

Com a aplicação deste método podemos transformar a imagem que antes estava no padrão BGR para BGRA, assim os ovos ficam com uma tonalidade mais azul, como pode ser visto na figura 3.

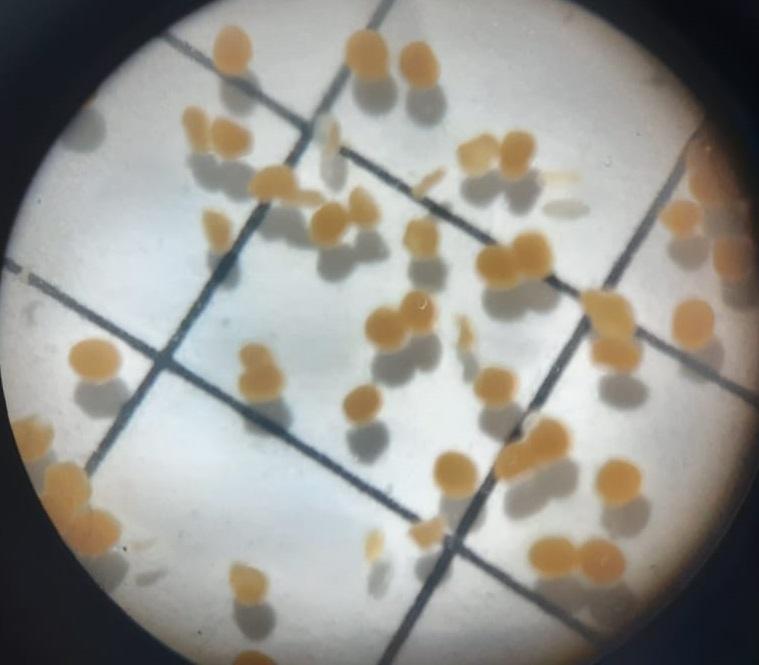


Figura 2: original

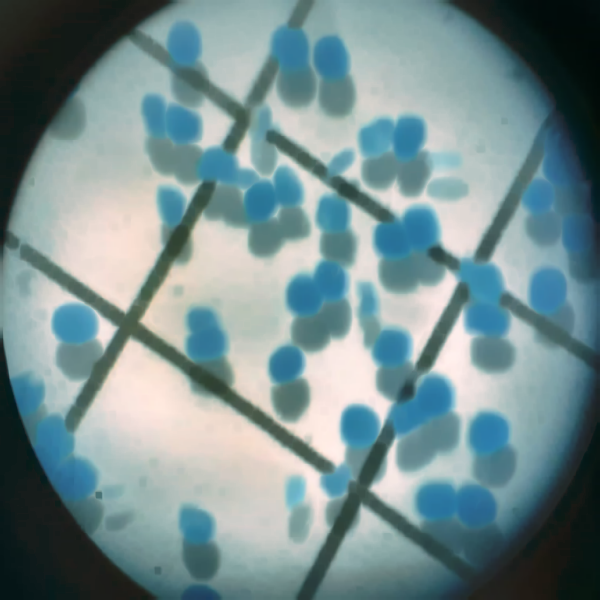


Figura 3: Imagem final

#### 

#### Limiarização de Otsu

O método de limiarização de otsu, é utilizado na parte da segmentação da imagem, onde essa etapa tem a função de realçar as bordas dos objetos contidos na imagem. Neste sentido o limiar de otsu serve para que seja possível transformar uma imagem em tons de cinza em uma imagem binário, assim destacando os ovos do fundo, o processo de transformação utiliza da constante T (nível de intensidade do pixel, que no programa foi dado no intervalo (127, 255) para determinar os valores que uma coordenada Ii,j, terá, onde se Ii,j estiver com os nível de intensidade contidos no intervalo da constante T, o pixel nesta posição receberá o valor 0 (preto) e caso seja maior ou menor, receberá 255 (branco), desta forma obtemos uma imagem binário.

A utilização do método do limiar de otsu é utilizado pelo software após a aplicação do pré-processamento, onde foram aplicados os filtros e máscaras na imagem, além de transformá-la para o padrão RGBA, logo após isso, transformamos a imagem para tons de cinza para a utilização da limiarização de otsu, dada pela função cvThreshold na biblioteca opencv e como esperados obtemos uma imagem binário, onde temos os ovos de camarão com a cor preta e o fundo em branco

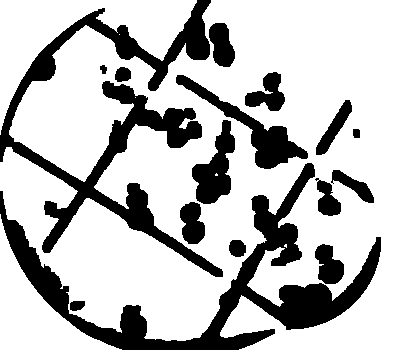


Figura 4

#### Método de Canny

O aplicação do método de Canny é muito importante na segmentação de uma imagem, pois permite a detecção das bordas, pois bordas bem definidas otimizam o processo de detecção de objeto, o método de canny é um exemplo de função que tem como objetivo destacar as bordas do objeto, para isso utiliza da variação de intensidade entre as coordenadas Ij1,i1 e Ij2,i2, para identificar um ponto qualquer, assim o método interpreta como boda os pontos que de fato pertencem a uma borda. Com a utilização do método cvCanny, obtemos uma imagem com as bordas em branco e as demais características da imagem em preto.



Figura 5.

Como é possível ver na imagem acima, ao utilizarmos o método de canny, é possível identificar e destacar as bordas dos objetos encontrados.

#### Detecção de círculos

Com a finalização das operações de pré-processamento e segmentação, são realizadas operações de reconhecimento de padrões para encontrarmos os ovos na imagem, para o melhor entendimento dessa etapa precisamos entender, o que é um círculo? Um círculo pode ser representado matematicamente por *(x−xcentro)² + (y−ycentro)² = r²* (equação da circunferência), onde *(xcentro, ycentro*) são o centro e *r²* é o raio do círculo, para essa operação utilizamos a função da OpenCV cvHoughCircles, que é uma variação do método Transformada Hough que encontra padrões geométricos em imagem, através “uso da equação da circunferência procura-se em toda a imagem contornos cujos pontos que o definem pertencem à zona interna definida pela equação” (NETO et al,) . O método da OpenCV recebe como parâmetros a:

* **imagem:** pré-processada e tratada.
* **método de detecção de círculo**: Atualmente a biblioteca openCV só disponibiliza o método HOUGH GRADIANT.
* **dp:** razão inversa da resolução do acumulador em relação a imagem, esse pareamento dita a sensibilidade do método de encontrar círculos
* **minDist:** Representa a distância mínima entre dois círculos.
* **param1:** Limite máxima para a detecção de bordas.
* **param2:** Limite mínimo para a detecção de bordas
* **minRadius e maxRadius:** Parâmetros opcionais que servem basicamente para termos uma média do tamanho dos círculos, ou seja, o tamanho mínimo e o tamanho máximo que eles pudessem ter.

Com a aplicação desse método, é possível que o software destaque os ovos de camarão na imagem, pois a função retorna as coordenadas no plano cartesiano, sendo x, y e o raio.

### Scikit-image

A biblioteca scikit-image, assim como a OpenCV, é uma ferramenta de código aberto que trás consigo uma extensa variedade de funções que viabilizam o processamento de imagem, classificação, clusterização e recuperação de imagens baseada em conteúdo (ROMUERE, 2017).

Inicialmente o software seria desenvolvido apenas com a biblioteca openCV para o tratamento de imagens, entretanto, com o andar da pesquisa, acabou sendo necessário a utilização desta biblioteca pois alguns métodos utilizados pela openCV para a extração de características não eram adequados ou suficientes, retornando muitos valores falsos.

#### Método regionprops

O método utilizado pela biblioteca sckit-image é o agrupamento por região, onde a função agrupa os pixels com mesma intensidade de cor, em relação a um centróide, a função *regionprops*, retorna uma lista com as propriedades da região, sendo assim possível determinar os pontos cartesianos (x, y) que ligam o centro a extremidade do círculo, com isso é utilizado a fórmula de bhaskara

h² = Co² + Ca²

Sendo:

h = Raio

Co = x

Ca = y

Para encontrar os valores do raio correspondente ao diâmetro maior e menor do ovo, após isso é feito uma verificação do tamanho do raio para que regiões que embora sejam captadas pela função regionprops, mas que não são efetivamente um ovo de camarão, essa verificação é dada pelo tamanho em pixel dos objetos, em nossos testes os ovos de camarão variam seus raios entre 3 e 35 pixels, valores maiores ou menores que esses são desconsiderados.

Com a conclusão desta etapa é retornado para o usuário uma imagem com os ovos circulados em vermelho e informações como número de ovos, assim como a média do volume dos ovos, além média de fecundidade e média da massa dos ovos.

### Metrologia da imagem digital

Após a obtenção dos raios dos ovos, para que seja possível a realização dos devidos cálculos das amostras, é preciso primeiro fazer a conversão de pixel para milímetro. A operação que utilizamos é bem parecida com as Máquinas Medição por Coordenada (MMC), descritas em Leta et al (2005), que embora seja de certa forma crítica as MMCs de poder haver erros na conversão dos valores, ainda assim são equipamentos que otimizam o processo de medição dos objetos. Assim como as MMCs, utilizamos coordenadas cartesianas para fazer a devida conversão dos valores, tendo como parâmetro o objeto no canto superior esquerdo da imagem (que suas dimensões em milímetro são dadas pelo usuário), com isso é possível termos as devidas dimensões dos demais objetos contidos na imagem.

### Etapas para a obtenção dos dados

Ao decorrer das sessões passadas foi explicado os métodos de processamento de imagem, que foram realizados para a obtenção dos dados, agora falaremos o passo-a-passo de todo o processo da obtenção da imagem, até a tela final, onde teremos todos os dados para a visualização do pesquisador.

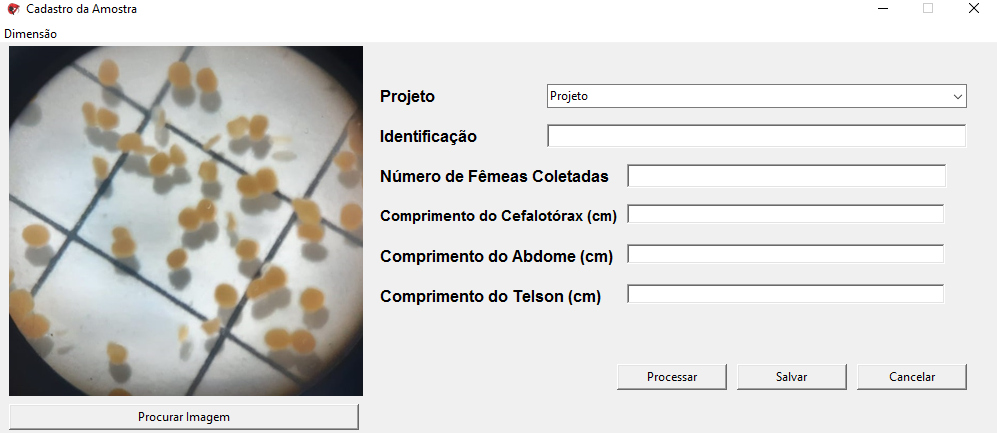
A primeira etapa o software adquire a imagem através de uma tela de diálogo, assim o sistema obtém a imagem. Com a imagem carregada no programa, e o formulário preenchido com as informações da amostra, são aplicados os métodos de processamento de imagem, que vai da melhora de qualidade, até a detecção dos primeiros círculos com a OpenCV, após isso, a imagem é passada para a biblioteca scikit-image que fica encarregada de obter os raios dos ovos, lembrando que, os raios obtidos estão em pixel, então é realizado o processo de conversão dos valores em pixel para milímetro, para que assim possa ser realizados os cálculos matemáticos.

# RESULTADOS

Para testarmos a eficácias do programa, utilizamos imagens de amostras que obtivemos com o auxílio do grupo de pesquisa Fecundidade do camarão *Macrobrachium amazonicum*, já mencionado anteriormente, além disso este mesmo grupo se disponibilizou para fazer a análise dos ovos para que assim tivéssemos uma média do tempo necessário que levaria para fazer a contagem de forma manual.

Na análise manual, os pesquisadores demoram em média 3 a 5 minutos para contar 200 ovos, levando mais tempo caso fosse necessário a pegar informações como o diâmetro maior e menor. Para a contagem de 1000 ovos os pesquisadores levam em média 20 minutos a 30 minutos, dependendo de fatores como cansaço visual e repetições de vezes que eles contam uma mesma amostra para terem a certeza dos dados catalogados.

Já análise com o software desenvolvido trás resultados satisfatórios, pois com o uso do programa em um computador com um processador i5 com memória RAM de 4GB, nós conseguimos obter as informações como média de fecundidade, média da massa dos ovos e a média do volume dos ovos em um tempo médio de 59 milissegundos a 1 segundo, esse tempo varia de acordo com a quantidade de objetos contidos na imagem, além das configurações do computador. Em nossa pesquisa, trabalhamos com um número em média de 40 ovos por imagem, levando em conta esses números o tempo médio para fazer a análise de 1000 ovos, seria de apenas 25 segundos (levando em conta que cada análise demoraria 1 segundo).

Tela de cadastro da amostra (Figura 6).

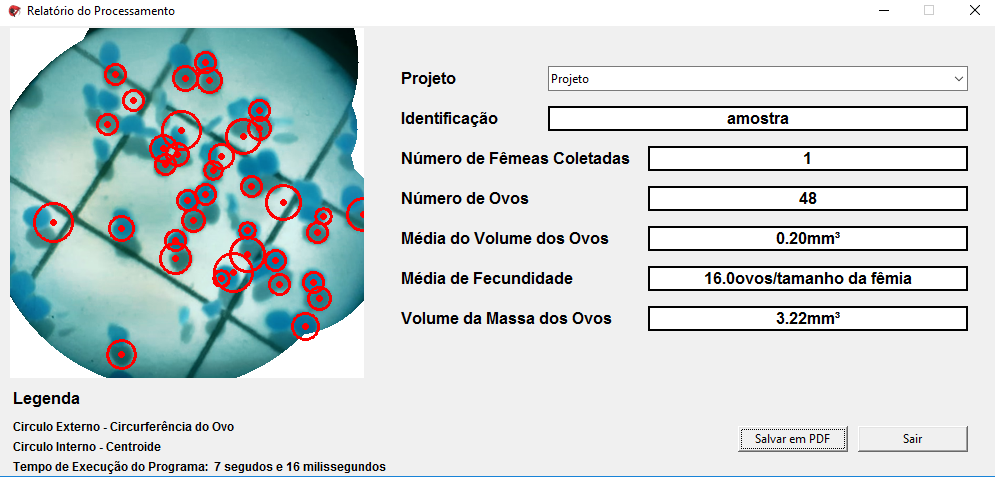


Imagem Processada com os dados (Figura 7).

Tabela com os diâmetros dos ovos analisados da Figura 7.

|  |  |
| --- | --- |
| Diâmetro Maior (mm³) | Diâmetro Menor (mm³) |
| 1.08 | 0.80 |
| 1.01 | 0.54 |
| 0.39 | 0.32 |
| 0.38 | 0.38 |
| 0.38 | 0.38 |
| 0.85 | 0.85 |
| 0.38 | 0.38 |
| 0.57 | 0.57 |
| 0.26 | 0.71 |

OBS: Apenas os raios dos 9 primeiros ovos.

Nas imagens acima podemos ver as informações que antecedem o processamento da imagem e os resultados obtidos após o mesmo, é importante ressaltar que nem todos os círculos em vermelho são efetivamente ovos de camarão e que mesmo que eles estejam visíveis para os usuários, eles não são contados pelo software, isso se dá pelo fato que a biblioteca OpenCV considerar muito falsos negativos, por esse motivos utilizamos a biblioteca scikit-image, com ela esses falsos positivos são dispensados, mas mesmo que a scikit-image tenha uma função que consiga detectar os ovos com mais confiabilidade caso a imagem esteja com uma qualidade baixa assim mesmo falsos positivos podem ser aceitos.

# DISCUSSÃO

O software desenvolvido, como já mencionado, tem como objetivo facilitar a análise dos ovos de camarão, já que se trata de um trabalho que demanda de uma série de repetições, o que pode ocasionar um cansaço visual ao pesquisador que realiza essa tarefa. A utilização do software facilita muito na coleta de dados, já que, após obter as fotos das amostras o processo para obter informações como a quantidade de ovos, massa dos ovos e média de fecundidade não demora 1 minuto, isso faz com que uma tarefa que pode variar entre 20 e 30 minutos, dependendo da quantidade de ovos de cada fêmea de camarão ou até mesmo a quantidade de fêmeas a ser analisadas, ou seja a utilização do software é benéfica e eficaz.

Em nossas pesquisas sobre o assunto foram encontradas algumas dificuldades por outros pesquisadores como em Ramos (2014), que ao desenvolver um algoritmo de reconhecimento de objetos, teve dificuldade ao obter o objeto de análise, pois problemas como o fundo da imagem e ruídos o atrapalhavam de obter os resultados desejados, estes problemas também ocorrem no desenvolvimento deste software, onde o programa tem dificuldade de distinguir ovos muito próximos, ou seja uma imagem que esteja repleta de ovos muito aglomerados pode trazer muitos falsos positivos ou o contrário, além disso a presença dos ruídos, pode fazer com que em alguns casos os ovos não possam ser identificados ou que o fundo da imagem possa ser reconhecido como parte da circunferência de um ovo, esses problemas podem ocorrer, ao aplicarmos o método cvThreshold, onde ovos que estejam muito próximos de pontos muito luminosos (ruídos que mesmo após aplicação dos métodos de pré-processamento e segmentação não foram possíveis ser removidos) acabam sumindo, ou então a intensidade dos pixels dos ovos esteja muito parecida com a dos ovos, fazendo com que parte do fundo seja reconhecido como um ovo.

Em suma a aplicação do software ainda se torna mais prática do que análise manual, pois o trabalho maior é em obter uma imagem com um certo nível de qualidade e um certo cuidado em posicionar os ovos, pois assim, erros no processamento da imagem são reduzidos e consequentemente a análise trará resultados mais satisfatórios ao pesquisador, tornando um trabalho de meia hora em uma tarefa de 1 minuto.

# CONCLUSÃO

Com os dados obtidos podemos notar a importância da visão computacional no nosso dia-a-dia, pois com ela podemos automatizar tarefas muitas vezes cansativas e demoradas, com o software que foi desenvolvido a análise dos ovos de camarão pode ser feita de forma prática, sem ser preciso uma grande quantidade de equipamentos.

Os testes feitos apontaram um grande desempenho do sistema ao analisar as amostras, além disso o processamento das imagens demora cerca de 5 segundos, em comparação a contagem feita pelo o grupo de pesquisadores que nos auxiliou no andar do nosso projeto, onde a contagem de 200 ovos pode variar de 3 a 5 minutos sem pegar as informações dos ovos, ou seja o uso deste equipamento potencializar as pesquisas nessa área da biologia ao modo que facilita na obtenção de dados quantitativos dos estudos.

# REFERÊNCIAS

ALBUQUERQUE, V. et al. **Sistema de segmentação de imagens para quantificação de microestruturas em metais utilizando redes neurais artificiais**. Matéria (Rio J.), Rio de Janeiro, v. 12, n. 2, 2007. Disponível em <http://www.scielo.br/scielo.php? script=sci\_arttext&pid=S151770762007000200018 &lng=pt&nrm=iso>. Acesso em 20 fev. 2021.

ANGER, K. A.; MOREIRA, G.S.M. **Morphometric and Reproductive Traits of Tropical Caridean Shrimp**, Journal of Crustacean Biology, Volume 18, Issue 4, 1 October 1998, p. 823–838.

CATANEO, L. G. **Contador Eletrônico de Ovos**. 2017. 92f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica) Faculdade de Engenharia e Arquitetura da Universidade Passo Fundo, Rio Grande do Sul, 2017.

COLLART, O. O.; RABELO, H.R. **Variation in Egg Size of the Fresh-Water Prawn Macrobrachium Amazonicum (Decapoda: Palaemonidae)**. Journal of Crustacean Biology, Volume 16, Issue 4, 1 October 1996, p. 684–688. Disponível em: <<https://academic.oup.com/jcb/article/16/4/684/2418820>>. Acesso em: 08 de fev. de 2021.

E. M. Nazari, M. S. Simões-Costa, Y. M R. Müller, D. Ammar, M. Dias. **Comparisons of Fecundity, Egg Size, and Egg Mass Volume of the Freshwater Prawns Macrobrachium Potiuna and Macrobrachium Olfersi (Decapoda, Palaemonidae)**. Journal of Crustacean Biology, Volume 23, Issue 4, 1 December 2003, p. 862–868. Disponível em: <<https://academic.oup.com/jcb/article/23/4/862/2679849>>. Acesso em: 07 de fev. de 2021.

GARCÍ-DÁVILA, C.R.G.D.; ALCANTÁRA B., F.A.B.; VASQUEZ R., E.V.R.; CHUJANDAMA S., M.C.S. **Biologia reprodutiva do Camarão Macrobrachium brasiliense (Heller, 1862) (Crustacea: Decapoda: Palaemonidae) em Igarapés de terra firme da Amazônia Peruana**,Acta Amaz. vol.30 no.4 Manaus Dec. 2000.

LETA, F. R.; FELICIANO, F. F.; SOUZA, I. L. **VISÃO COMPUTACIONAL APLICACADA À METROLOGIA DIMENSIONAL AUTOMATIZADA: CONSIDERAÇÕES SOBRE SUA EXATIDÃO**. 2005. Disponível em: <<https://doi.org/10.22409/engevista.v7i2.164>>. Acesso em: 30 de set. de 2021.

MÜLLER, V.M.R.M.; NAZARI, E.M.N.; AMMAR, D.A.; FERREIRA, E.C.F.; BELTRAME, I.T.B.; PACHECO, C.P. **Biologiados Palaemonidae (Crustacea, Decapoda) da bacia hidrográfica de Ratones, Florianópolis, Santa Catarina, Brasi**l, Rev. Bras. Zool. vol.16 no.3 Curitiba 1999.

PASCHOAL, L. R. P. **História natural de Macrobrachium amazonicum (Heller, 1862) (Decapoda: Palaemonidae) e sua importância em reservatórios neotropicais do sudeste brasileiro**. 2017. Disponível em: <<https://repositorio.unesp.br/handle/11449/151993>>. Acesso em: 20 de fev. 2021

PRESSMAN, R. S. **Engenharia de Software: uma abordagem profissional**. 7 ed. McGraw Hill, 2011.

ROMA, J. V. M. **Método Computacional para Medição Automática do Diâmetro Limbar Computational Method for Automatic Measurement of the Limbus Diameter**. 2020. Disponível em: <<https://monografias.ufma.br/jspui/handle/123456789/4252?mode=full>>. Acesso em: 30 de set. de 20221.

SOMMERVILLE, I. **Engenharia de Software**. 9ed. São Paulo: Pearson Prentice Hall, 2011. SZELISKI, R. **Computer Vision: algorithms and applications**. 1 Ed. Berlin: Springer-Verlag, 2010.

VERONEZI, C. C. D. et al. **Análise computacional para auxílio ao diagnóstico de osteoartrite de coluna lombar baseado em redes neurais artificiais**. Revista Brasileira de Ortopedia, Paulo, v. 46, n. 2, 2011.

1. OpenCV disponível em <<https://opencv.org>> [↑](#footnote-ref-1)
2. Scikit-image disponível em <https://scikit-image.org> [↑](#footnote-ref-2)
3. Python disponível em < <https://www.python.org>> [↑](#footnote-ref-3)