**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, xx de setembro de 2021**

**SUMÁRIO**

[***RESUMO 4***](#_z337ya)

[***1.*** ***INTRODUÇÃO 5***](https://docs.google.com/document/d/106vLAdWPjiiDEf7CGKpuk7nTAaIXti6b1MM7q9leAH4/edit#heading=h.tyjcwt)

[**1.1.** **Objetivo 6**](https://docs.google.com/document/d/106vLAdWPjiiDEf7CGKpuk7nTAaIXti6b1MM7q9leAH4/edit#heading=h.3dy6vkm)

[**1.1.1.** **Objetivos Específicos 6**](https://docs.google.com/document/d/106vLAdWPjiiDEf7CGKpuk7nTAaIXti6b1MM7q9leAH4/edit#heading=h.1t3h5sf)

[***2.*** ***METODOLOGIA 8***](https://docs.google.com/document/d/106vLAdWPjiiDEf7CGKpuk7nTAaIXti6b1MM7q9leAH4/edit#heading=h.4d34og8)

[***3.*** ***ETAPAS REALIZADAS 10***](https://docs.google.com/document/d/106vLAdWPjiiDEf7CGKpuk7nTAaIXti6b1MM7q9leAH4/edit#heading=h.2s8eyo1)

[***4.*** ***ETAPAS A SEREM REALIZADAS 13***](https://docs.google.com/document/d/106vLAdWPjiiDEf7CGKpuk7nTAaIXti6b1MM7q9leAH4/edit#heading=h.17dp8vu)

[***5.*** ***RESULTADOS 16***](https://docs.google.com/document/d/106vLAdWPjiiDEf7CGKpuk7nTAaIXti6b1MM7q9leAH4/edit#heading=h.3rdcrjn)

[***6.*** ***DISCUSSÃO 16***](https://docs.google.com/document/d/106vLAdWPjiiDEf7CGKpuk7nTAaIXti6b1MM7q9leAH4/edit#heading=h.26in1rg)

[***REFERÊNCIAS 17***](https://docs.google.com/document/d/106vLAdWPjiiDEf7CGKpuk7nTAaIXti6b1MM7q9leAH4/edit#heading=h.lnxbz9)

[***ANEXO I: Acompanhamento do Bolsista 19***](https://docs.google.com/document/d/106vLAdWPjiiDEf7CGKpuk7nTAaIXti6b1MM7q9leAH4/edit#heading=h.35nkun2)

# RESUMO

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

# INTRODUÇÃO

## Objetivo

## Objetivos Específicos

## 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

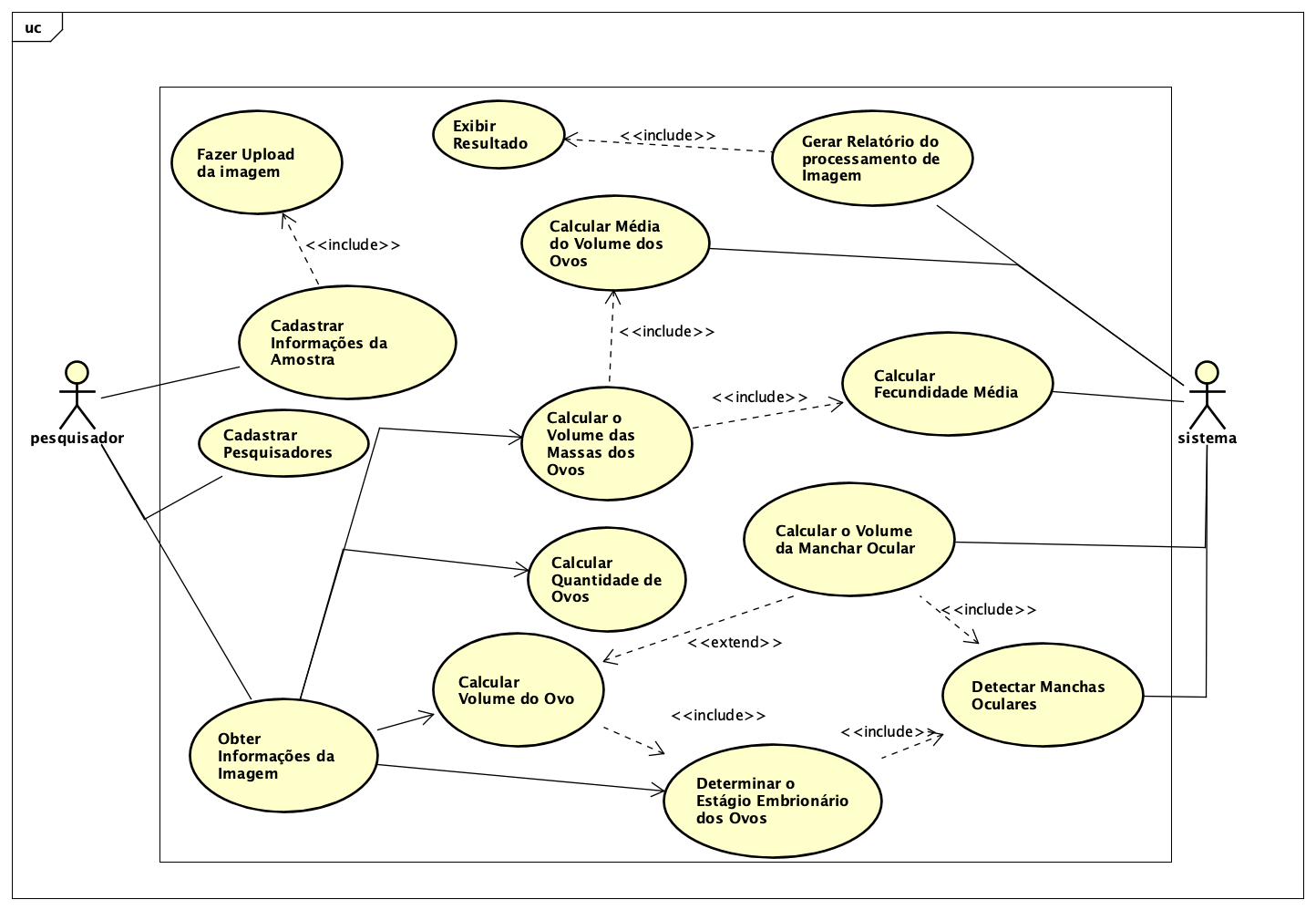
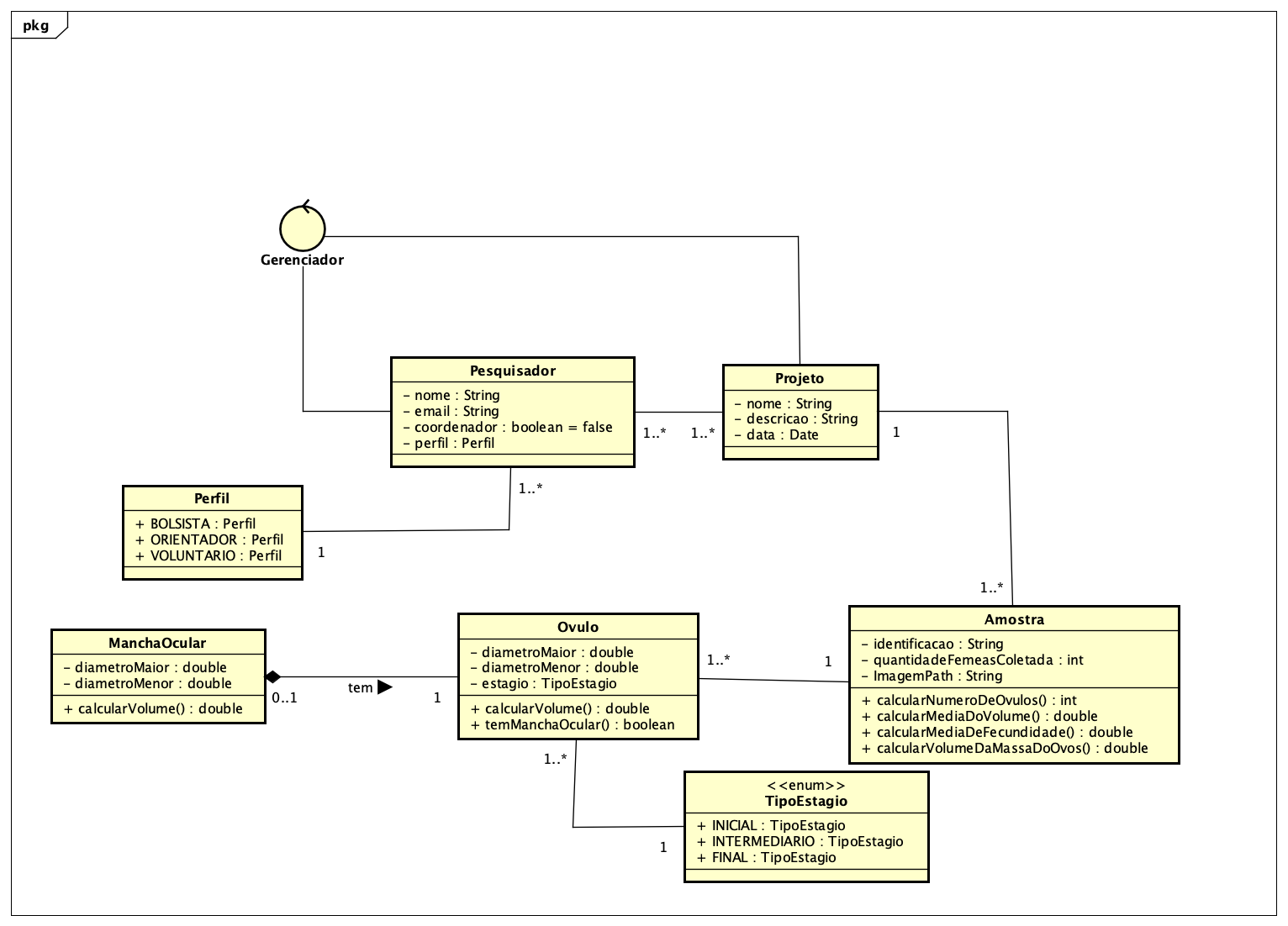
O objetivo do projeto como já mencionado é o desenvolvimento de um software de visão computacional capaz de detectar os ovos de camarão e aplicar os devidos cálculos com os dados captados, para que assim análises que muitas vezes poderiam demorar e cansar o pesquisador pudessem ser feitas de forma mais 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 melhor 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 da 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[[1]](#footnote-0)), 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 o desenvolvimento a prototipação 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[[2]](#footnote-1), 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.

### Método de equalização 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, para que seja possível normalizamos o histograma de uma imagem 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 e cvThreshold

Após a imagem receber um tratamento com os filtro gaussiano 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 óvulos ficam com uma tonalidade mais azul, como pode ser visto na figura 2.

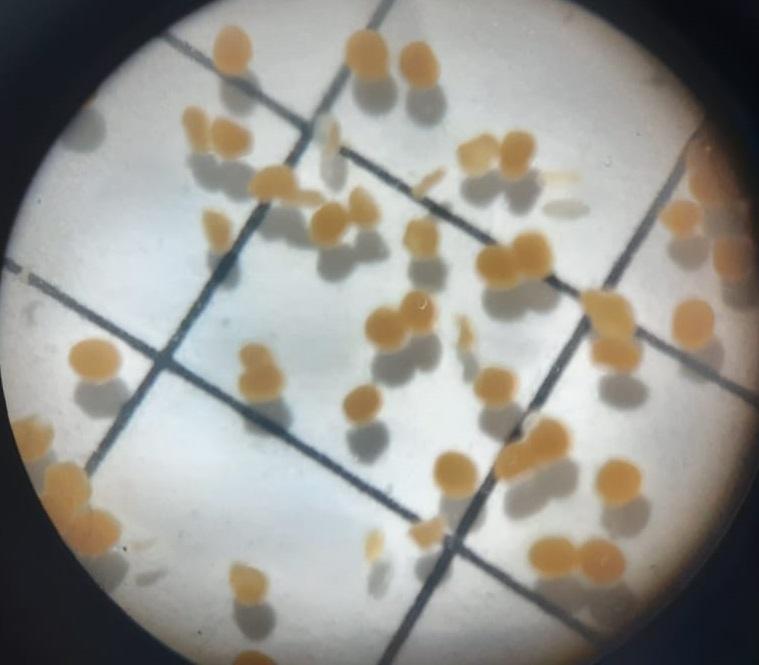


Figura 1: original

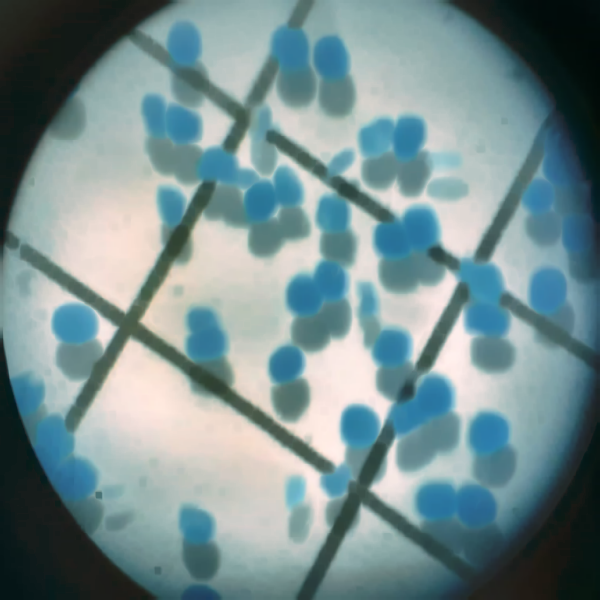


Figura 2: Imagem final

Com a obtenção da imagem com os ovos azulados, aplica-se um método de erosão na imagem, isso serve para que as bordas dos óvulos fiquem mais em destaque, utilizamos a função cvCvtColor novamente, desta vez para transformar a imagem para tons de cinza que varia de 0 (preto) e 255 (branco), assim, ao aplicarmos o método cvThreshold, que procura o nível de intensidade dos pixel em um intervalo - para o software foi entre 127 e 255 - e ao fim deste processo obtemos uma imagem com os ovos de camarão em preto e o fundo em branco.

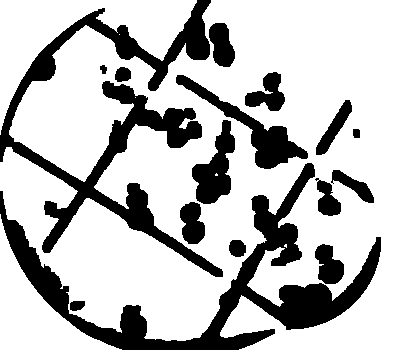


Figura 3.

### 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 pixel 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.

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 mesmo, além média de fecundidade e média da massa dos ovos.

## 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, por isso é feito uma transformação de pixel para mm³, onde para que isso seja possível, o software considera o objeto mais a cima e a esquerda como objeto padrão, esse objeto terá as dimensões dada pelo pesquisador, e com os dados do objeto padrão adquiridos podemos realizar uma operação matemática simples, onde temos:

Tamanho em pixel -> tamanho real (objeto padrão)

Tamanho em pixel -> x (ovos encontrado pelo programa)

Com isso obtemos uma imagem com os ovos destacados em vermelho por círculos desenhados em cor de vermelho e os dados obtidos pelos cálculos da média da massa, do volume e fecundidade.

# RESULTADO

# 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 varia entre 30 a 40 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 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 pixel 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

1. Python disponível em < <https://www.python.org>> [↑](#footnote-ref-0)
2. Scikit-image disponível em <https://scikit-image.org> [↑](#footnote-ref-1)