**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***](#_2et92p0)

[***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

# METODOLOGIA

A metodologia utilizada para o desenvolvimento deste trabalho, foi dividida em partes, para que assim, fosse possível analisar e reunir os melhores materiais e métodos para a criação do software de visão computacional e assim poder concluir a pesquisa.

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

Após a concluímos o levantamento de requisitos e estudarmos o um pouco 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 as pesquisa 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, 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 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.

## 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.

### 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 atlânticos 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.

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 explicado 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 cv.cvtColor e técnica threshold

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 cv.cvtColor, 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.

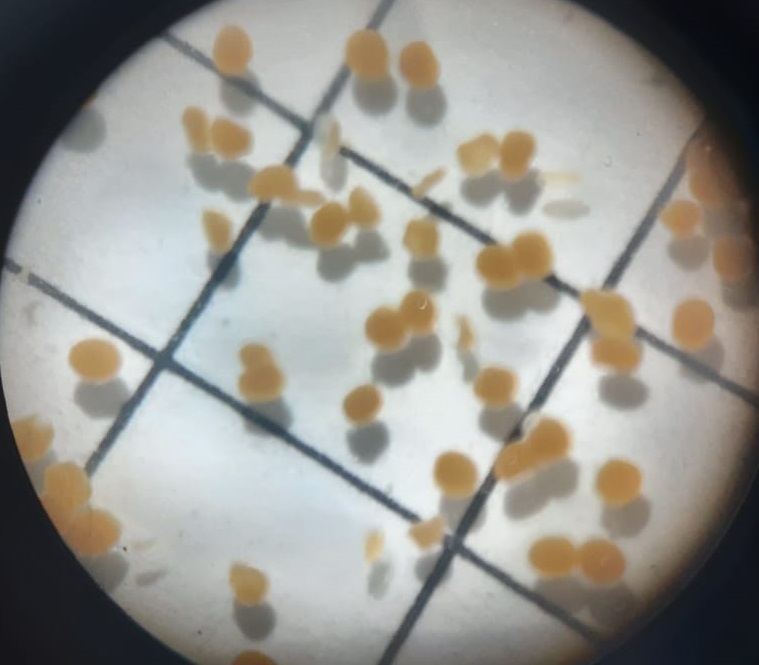


Figura 1: original

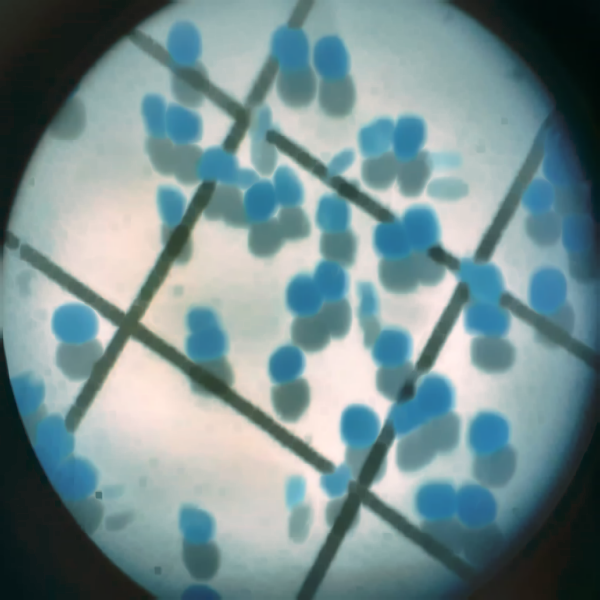


Figura 2: Imagem final

Com a obtenção da imagem com os ovos azulados, aplica-se um medo de erosão na imagem, isso serve para que as bordas dos óvulos fiquem mais em destaque, utilizamos a função cv.cvtColor novamente, desta vez para transformar a imagem para tons de cinza que varia de 0 (preto) e 255 (branco), assim, ao aplicarmos o método de threshold, que procura o nível de intensidade dos pixel em um intervalo - que 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.

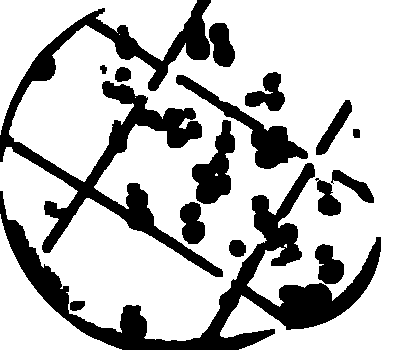


Figura 3.

### 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 cv2.HoughCircles(), 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

Inicialmente o software seria desenvolvido apenas com a biblioteca openCV para o tratamento de imagens, mas infelizmentes alguns métodos para a extração de características não eram adequados ou suficientes, retornando muitos valores falsos, por isso foi necessário a utilização desta biblioteca, que recebe a imagem já tratada por operações openCV, com alguns círculos detectados, após isso, com o método de procura por regiões, onde se define um centróide para a detecção de contornos, a biblioteca buscas as coordenadas *(xcentro, ycentro)* e das bordas, tendo esses valores, é aplicado o teorema de pitágoras para encontrar o raio da elipse, assim obtermos o diâmetro do ovo, essa etapa marca o fim do processo na imagem.

# ETAPAS REALIZADAS

# RESULTADOS

# DISCUSSÃO

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