UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ

***CAMPUS* LUIZ MENEGHEL**

RAFAEL FERNANDO RODRIGUES

DESENVOLVIMENTO DE UMA FERRAMENTA PROPOSTA DE UM ALGORITMO DE PROCESSAMENTO DE IMAGENS PARA AUXILIAR O RECONHECIMENTO DE ESPÉCIES DE BORBOLETAS

Bandeirantes
2016

RAFAEL FERNANDO RODRIGUES

DESENVOLVIMENTO DE UMA FERRAMENTA PROPOSTA DE UM ALGORITMO DE PROCESSAMENTO DE IMAGENS PARA AUXILIAR O RECONHECIMENTO DE ESPÉCIES DE BORBOLETAS

Trabalho de Conclusão de Cursosubmetido à Universidade Estadual do Norte do Paraná, como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. Me. Thiago Adriano Coleti

Bandeirantes

2016

RAFAEL FERNANDO RODRIGUES

DESENVOLVIMENTO DE UMA FERRAMENTA PROPOSTA DE UM ALGORITMO DE PROCESSAMENTO DE IMAGENS PARA AUXILIAR O RECONHECIMENTO DE ESPÉCIES DE BORBOLETAS

Trabalho de Conclusão de Curso submetido à Universidade Estadual do Norte do Paraná, como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

COMISSÃO EXAMINADORA

_____
Prof. Me. Thiago Adriano Coleti
UENP – *Campus* Luiz Meneghel

_____
Prof. Dr. Ederson Marcos Sgarbi
UENP – *Campus* Luiz Meneghel

_____
Prof. Sc. Wellington Della Mura
UENP – *Campus* Luiz Meneghel

Bandeirantes, 11 de julho de 2016.

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter me dado força e saúde para superar as dificuldades e terminar esse trabalho.

A minha família pelo apoio e compreensão.

Ao meu orientador e co-orientador, pelo suporte que me foi dado com suas correções e ensinamentos.

RESUMO

O processamento de imagem é uma área bastante explorada em todas as plataformas, com o intuito de identificar, mapear e reconhecer, esta técnica vem ganhando mais espaço em softwares que possuem este objetivo. Desenvolvido um sistema em linguagem Java, o processamento de imagem pode se tornar ainda mais efetivo devido a agilidade da linguagem aliado ao poder de cálculo de similaridade e processamento da biblioteca JAI, que também será utilizada por este trabalho. Para a comparação de similaridade de imagens, serão utilizados conceitos de distância Euclidiana e reconhecimento baseado em conteúdo. Assim, este trabalho propõe o desenvolvimento de uma ferramenta que seja capaz de verificar e reconhecer espécies de borboletas, facilitando pesquisas e utilização do sistema em todos os tipos de locais. Como resultado, pretende-se aperfeiçoar de forma confiável o tempo do processo de reconhecimento das espécies, visto que este trabalho se dá de forma manual atualmente.

Palavras-chaves: Espécie de Borboletas1, Processamento de Imagens 2, Linguagem Java *3*.

ABSTRACT

Image processing is a quite area exploited across all platforms, in order to identify, map and recognize this technique is gaining more space in software that has this goal. Developed a system in Java language, image processing can become even more effective because the agility of language combined with the power of similarity and processing JAI library calculation, which will also be used for this work. For the comparison of image similarity are used Euclidean distance concepts and recognition based on content. This work proposes the development of a tool that is able to verify and recognize butterfly species, facilitating research and use of the system in all types of locations. As a result, we intend to optimize reliably time the species recognition process, since this work is given to currently manually.

Keywords: Butterfly Species 1, Image Processing 2, 3 Java language.

LISTA DE FIGURAS

SUMÁRIO

1. Introdução

1.1 Contextualização

Tudo que ocorre ao nosso redor é completamente monitorado, e com a natureza não é diferente. Um impacto ambiental por menor que seja, em alguns casos, pode devastar uma região bem extensa. Existem algumas formas para verificar qual o tipo do impacto, a quem ele pode interferir e o porquê de estar ocorrendo, uma delas é através do monitoramento ambiental.

O monitoramento ambiental é uma atividade recorrente, contínua e periódica que envolve fatores qualitativos e quantitativos com o objetivo de verificar os impactos e alterações provocadas no meio ambiente. As observações e pesquisas geram relatórios da situação real que determinada região se encontra, facilitando para que as instituições de pesquisa ambiental tomem medidas preventivas (MAGNUSSON, 2013).

Há dois tipos de monitoramento ambiental: (1) em micro escala: Como o próprio nome diz em micro escala é direcionada para espaços menores, e as funções também são mais limitadas, tornando uma forma de monitoramento mais genérica, sendo usado para verificação de emissão de poluentes; e (2) macro escala: completamente o oposto ao anterior, sendo direcionado para grandes áreas geográficas. Pesquisas espacialmente orientada resultam em muitos tipos de dados, tais como planos de pesquisa, mapas, medições, publicações científicas, fotografias e informações sobre coleções biológicas, cada um com requisitos de armazenamento e de acessos diferentes (MAGNUSSON apud. BILLICK, 2010).

O monitoramento das borboletas é uma fonte de dados sobre o habitat do grupo, indicando também a qualidade ambiental e do estado de conservação do local, a espécie apresenta uma grande facilidade no manuseio e na captação, tornando isso uma vantagem para facilitar o monitoramento (ARAUJO, 2004).

A revolução da tecnologia da informação tem levado muitos a “descobrir” o potencial de armazenamento e recuperação de dados científicos (MAGNUSSON, 2013).

A necessidade de praticidade é requisito básico para que o tempo seja otimizado em uma pesquisa. Desenvolver um sistema que tenha esta característica pode ser a solução deste problema, uma vez que com um aplicativo criado, todos os

dados e informações relevantes a uma pesquisa podem ser armazenados nele e interpretado pelo mesmo. Existem diversos tipos de recursos que podem ser utilizados que tornam a pesquisa dinâmica, como por exemplo, a identificação de sons, gravação de vídeos, a captura e reconhecimento de imagens e assim por diante.

A área de processamento de imagens é bastante abrangente. Muitos sistemas utilizam este recurso visando soluções rápidas. Softwares de reconhecimento facial, por exemplo, que se utilizam desta mesma técnica, captam pontos específicos em uma foto para distinguir, reconhecer e classificar de forma única cada fisionomia humana (GONZALEZ e WOODS, 2000). Outro exemplo de sistemas baseados em processamento de imagens são softwares meteorológicos, que através de mapas de calor e umidade conseguem prever o clima. Acompanhando a velocidade das tecnologias, *smartphones, tablets* e outros dispositivos portáteis vêm ganhando cada vez mais espaço, pelo simples fato da praticidade.

Este trabalho abordará o reconhecimento e processamento de imagem, que é uma técnica que consiste na entrada e saída, são imagens tais como fotografias já manipuladas ou vídeos. Através de algoritmos de reconhecimento aliados ao processamento de imagem verificarão a cor, tamanho e formato de cada espécie, tornando mais prática uma análise ou catalogação, uma vez que o tempo para estes procedimentos poderá ser reduzido drasticamente.

1.2 Objetivos

Esta seção apresenta os objetivos gerais e específicos deste trabalho.

1.2.1 Objetivo Geral

Este projeto tem como principal objetivo desenvolver um algoritmo de processamento de imagens para reconhecimento de espécies de borboletas. Pretende-se com esse algoritmo otimizar o tempo de processamento e espécies no processo de monitoramento da biodiversidade.

1.2.2 Objetivos Específicos

Para alcançar o objetivo geral, os seguintes objetivos específicos deverão ser atingidos:

- Determinar a efetividade do funcionamento de algoritmos de processamento de imagens na linguagem Java;
- Compreender o processo de identificação de espécies de borboletas;
- Determinar qual o melhor algoritmo para reconhecimento de espécie de borboleta;
- Verificar a eficiência do algoritmo proposto para o reconhecimento de espécie de borboletas;

1.3Justificativa

O objetivo deste projeto é desenvolver um algoritmo que seja capaz de reconhecer e identificar espécies de borboletas dentro de uma plataforma *Desktop*. Este tema foi escolhido a partir da necessidade do desenvolvimento de um sistema para auxiliar pesquisadores que atuam na área de biologia no processo de monitoramento e controle de borboletas, visto que este procedimento ainda é feito de forma manual e demorada.

Para desenvolver um trabalho efetivo, se faz necessário entender os conceitos sobre processamento de imagens, visto que esta tecnologia será o ponto fundamental para o desenvolvimento da pesquisa. Segundo MARQUES FILHO (1999), o processamento de imagens se dá em cinco passos, sendo eles: a (1) aquisição, onde a imagem capturada é convertida em uma base numérica. O (2) armazenamento que pode ser um desafio dependendo do tamanho e qualidade da foto tirada. O (3) processamento que envolve os procedimentos padrões na execução dos algoritmos que realizam o trabalho que analisam as imagens. A (4) transmissão que seria o envio à distância da imagem já processada, utilizando protocolos já existentes. E por fim, a (5) exibição que consiste na apresentação das informações obtidas.

Muitos softwares utilizam esta técnica para fazer análise de imagens. O objetivo de se usar processamento digital de imagens é melhorar o aspecto visual de certas feições estruturais para o analista humano e fornecer outros subsídios para a

sua interpretação, inclusive gerando produtos que possam ser posteriormente submetidos a outros processamentos (SPRING, 1996).

2. Fundamentação Teórica

Este capítulo apresenta a fundamentação teórica dividido em três subtópicos: Linguagem Java, Processamento de Imagem e Monitoramento da Biodiversidade.

2.1 Linguagem Java

Segundo MENDES (2011), no ano de 1991, a tecnologia Java foi criada inicialmente como uma ferramenta de programação de um projeto que tinha como principal objetivo criar uma nova plataforma para a computação interativa, ou seja, a linguagem de programação não era o principal objetivo do projeto. Posteriormente, no ano de 1992 foi gerada a primeira demonstração do projeto, que representou um sistema executando em um hardware similar ao *Palmtop* com capacidade de controle remoto que ainda oferecia uma interface sensível ao toque interativo. Este sistema inicialmente criado foi construído com o objetivo de puxar ligações telefônicas entre membros da mesma equipe de trabalho, uma vez que a linguagem Java dentro do sistema atendia a vários dispositivos de uso doméstico e apresentava uma interface com animação.

A linguagem de programação Java representa uma linguagem simples, orientada a objetos, multithread, interpretada, neutra de arquitetura, portátil, robusta, segura e que oferece alto desempenho. É importante observar que a tecnologia Java é composta de uma linguagem de programação e de uma plataforma (API e a máquina virtual) (MENDES, 2011).

A plataforma Java apresenta uma série de características que justifica o fato desta linguagem ser uma das linguagens mais populares que temos atualmente segundo DEITEL (2010), como por exemplo: simplicidade, a orientação a objetos, que torna esta linguagem mais dinâmica, aplicação do conceito de *Multithread*, que seria o desenvolvimento do sistema em concorrência possibilitando escalar este sistema horizontalmente sem perder a eficiência, a independência de arquitetura é outra característica positiva da linguagem, pois, permite que um sistema seja desenvolvido para Windows, Linux ou até mesmo em MAC, desde que tenha uma JVM (*Java Virtual Machine*) instalada e configurada corretamente para o uso. Um

ponto destacado por MENDES (2011) é a portabilidade da plataforma que permite desenvolver para desktops, dispositivos móveis e para sistemas Web também, oferecendo ao desenvolver aplicar seu sistema em plataformas distintas. Também são reconhecidas, como ponto positivo, as características de alto desempenho da linguagem, sendo uma das mais utilizadas em sistemas distribuídos por conta deste fator como destaca DEITEL (2010), e também a segurança que o Java proporciona.

Embora o desenvolvimento para dispositivos móveis se torne cada vez mais uma tendência, como aponta LECHETA (2010), há sistemas em que aplicações móveis ainda não são indicadas devido a uma série de fatores.

O sistema *Android* (que utiliza a linguagem Java) possui um ambiente de execução que possibilita rodar programas complexos com memória limitada, bateria limitada e processamento limitado (LECHETA, 2010).

A arquitetura Java pode ser apresentada em três camadas distintas: Camada de Apresentação, de Negócio e de Persistência. (1) A Camada de Apresentação é a camada que trata a parte de interface gráfica. Quando se fala em interface gráfica, entende-se por botões, janelas, formulários, caixas de texto (*TextBox*), textos escritos na tela (*Label*), e muitos outros componentes que o Java permite que sejam usados para se referir à parte visual do programa que é apresentada ao usuário final, com o objetivo de fazer interação com o sistema.

(2) Camada de Negócios é camada responsável pela lógica do sistema. Não é uma camada visível para o usuário final, ao contrário da Camada de Apresentação, que apresenta interface om o usuário. Esta é a camada que contém a lógica de como o sistema trabalha, e as regras de negócio transcorrem.

E por fim, a (3) Camada de Persistência é camada responsável por salvar o objeto na base de dados. Todo sistema requer que as informações colhidas e calculadas sejam gravadas em uma estrutura que seja possível recuperar depois para efeito de comparação ou de apresentação. Esta camada tem este objetivo, de criar a interface com o banco fazendo ambas aplicações se comunicarem. Esta camada também não é visível ao usuário final, a menos que esta seja exibida na tela.

Figura 1: Exemplo da arquitetura Java em 3 camadas

Existe, hoje, uma grande quantidade de softwares de processamento de imagens com os mais variados métodos e utilizados pelos mais diversos tipos de usuários (SANTOS, 2010).

A linguagem Java, como dito anteriormente, utiliza o conceito de Orientação a Objetos, e dentro dele podemos citar que esta plataforma possibilita o desenvolvimento de sistemas com a utilização de bibliotecas. O conceito de bibliotecas funciona como um pacote de classes, que possui diversas funcionalidades e métodos prontos para o uso, que devem ser importadas para dentro do código. Para este trabalho, destaca-se a biblioteca JAI (Java *AdvancedImaging)*, que se trata de um poderoso conjunto de classes voltadas para a manipulação de processamento digital de imagens.

A biblioteca JAI possui uma classe básica para representação de uma imagem e seus pixels denominada *PlannarImage*. Esta classe permite arepresentação de imagens no Java com maior flexibilidade do que a classe *BufferedImage* (classe básica do Java para representar imagens) (SANTOS, 2010).

A estrutura de uma *PlannarImage* é relativamente simples. O componente *Raster* é uma instancia do *PlannarImage* em que são armazenados os pixels da imagem. *DataBuffer* e *SampleModel* são subclasses de empacotamento de imagem que estão contidas dentro do *Raster.* Outro componente desta classe é o ColorModel que possui uma única subclasse associada a ela, a *ColorSpace*que, segundo SANTOS (2010) determina com um valor de um pixel pode ser traduzido em valores de cor. A figura abaixo mostra a estrutura da *PlannarImage.*

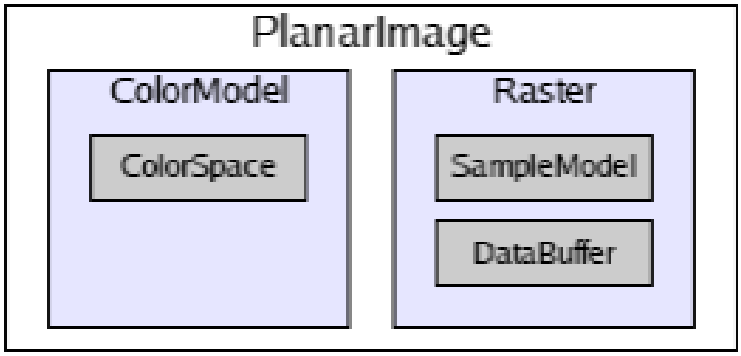

Figura 2: Estrutura de uma *PlannarImage*

Outra estrutura muito usada pela biblioteca JAI é a *TiledImage,* referente ao processo de deixar a imagem quadriculada em “trilhos” e analisar de forma individual cada um deles, como na figura abaixo.

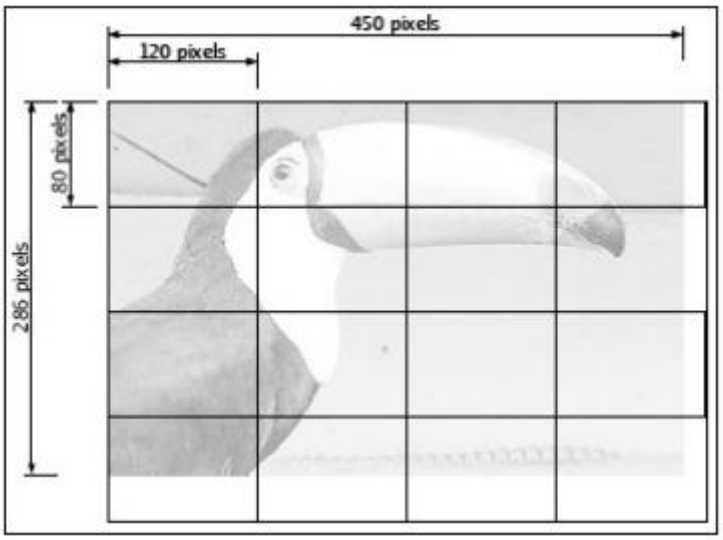

Figura 3: Estrutura de uma *TiledImage*

Segundo SANTOS (2010), as imagens na JAI podem ser multidimensionais (com vários valores diferentes associados a um pixel) e ter pixels representados por inteiros ou ponto flutuante.

2.2 Processamento de Imagens

Esta seção abordará as definições, conceitos e métodos de extração e análises de processamento de imagens.

Processamento de imagem é um vasto conjunto de operações que podem ser aplicadas em uma representação matemática de imagem com objetivo de resolver um determinado problema (GONZALEZ e WOODS, 2000). Existe outra definição feita por AFFONSO (2002) que diz que por processamento digital de imagens entende-se a manipulação de uma imagem via computador, de modo que a entrada e saída do processo sejam imagens, com o objetivo de melhorar o aspecto visual de certas feições estruturais, proporcionando maior facilidade na extração de informações.

Dessa forma o processamento de imagens, ao longo do tempo, veio alinhando-se à computação gráfica, a fim de otimizar e aperfeiçoar software de diversas áreas, como ambiental, médica, mecânica, entre outras, facilitando através da extração e análise das imagens, o monitoramento de atividades frequentes (GONZALEZ e WOODS, 2000).

O primeiro passo relativo ao processamento de imagens é a aquisição de imagens, que funciona como uma atividade fundamental, uma vez que é a partir dela que as imagens, fotos ou vídeos são captados para serem processados e interpretados, para isso são necessários dispositivos de vídeo. Quanto maior à qualidade do dispositivo, mais preciso e eficiente será o processamento e a análise.

De acordo com FLORENZANO (2008) as técnicas de processamento de imagens, podem ser divididas em três etapas, sendo elas: pré-processamento de imagens, realce de imagens e analise de imagens.

Dentro da etapa de pré-processamento de imagens, ocorrem funções preliminares, como tratamento prévio dos dados obtidos, correção de erros ou possíveis distorções geométricas e a suavização de ruídos, que são causados por erros na transmissão de dados.

Segundo NUNES (2011), a técnica de suavização é outro fator elementar na hora dos dados serem processados, principalmente se a imagem captada não for de boa qualidade ou muito antiga, para resolver este problema, filtros de suavização são muito usuais. Dentro da técnica de suavização de imagem há dois tipos de filtros que são destacados por NUNES (2011), que são Filtros de Média e Filtros de Mediana.

Filtro de Média, assim como Filtro de Mediana, é analisado com bases em matrizes, 3x3 ou 5x5, por exemplo, (lembrando que quanto maior for a janela, ou matriz de analise, maior será o efeito sobre o pixel a ser suavizado) cada pixel

assume um valor de sua respectiva cor, em uma janela 3x3, obtemos 9 valores de pixel onde todos esses valores são somados e dividido por 9, obtendo um valor médio, desta forma o valor médio substitui, o pixel a ser suavizado que anteriormente estava discrepante. Abaixo segue um exemplo de suavização por Média Vizinhança retirado do trabalho de NUNES (2011).

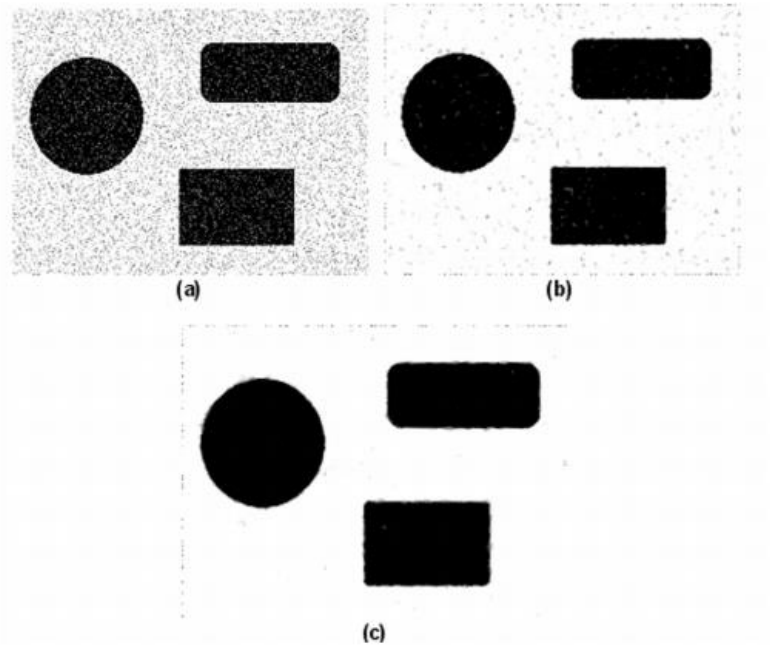

Figura 4:

(a) Imagem original, (b) suavizada por uma matriz 3x3, e (c) suavizada por uma matriz 5x5, adaptado de NUNES (2011).

O Filtro de Mediana funciona de forma similar, com a diferença, que o valor do pixel a ser substituído não é o valor da média obtida, mas, como o próprio nome diz a mediana, que consiste em alinhar todos os valores da matriz em ordem crescente e tomar como resultado o valor central, ou seja, em uma matriz de nove elementos, o quinto valor é o valor mediano. A desvantagem deste filtro em uma vizinhança retangular é o dano causado nas linhas finais e curvas agudas (WATT, 1998). E sua principal vantagem, é ser um dos filtros que mais preservam o contorno na hora da suavização.

Outra técnica utilizada é a Limiarização, que segundo NUNES (2011), consiste, basicamente, em alterar os valores dos pixels de uma imagem deixando-a com uma quantidade menor de níveis com o objetivo de separar estruturas de

interesse do fundo da imagem. Como vantagem, esta técnica tem a simplicidade de implementação.

A Limiarização funciona como uma binarização da imagem obtida, nesta técnica é utilizada uma escala para medir os níveis de cinza, indo de zero que simboliza a cor branca, até o valor máximo da escala, que simboliza o preto, esta escala recebe o nome de taxa de limiaridade. Uma vez que este trabalho tem como foco a plataforma móvel, essa técnica se mostra uma boa opção, pela simplicidade e agilidade de processamento. A figura abaixo representa a Limiarização, onde em (a) é a imagem real a ser processada, o passo seguinte apresenta dois passos: em (b) a taxa de limiaridade é de 128, e em (c) a taxa é de 100. Em (b) o realce entre o preto e o branco é maior do que em (c).

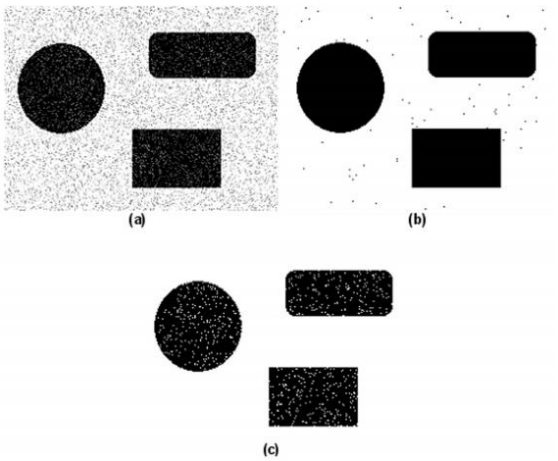

Figura 5:

(a) Imagem real, (b) imagem binarizada com a taxa limiar = 128, e (c) imagem binarizada com a taxa limiar = 100, adaptado de NUNES (2011).

Splitting é outra técnica utilizada para complementar a Limiarização, que segundo NUNES (2011) visa a aumentar o contraste de uma imagem com base no seu histograma. Esta operação divide os pixels em dois grupos distintos de níveis de cinza. Desta forma, os pixels com valores menores da imagem (cores escuras) são diminuídos ainda mais e aqueles com valores altos (pixels claros) tornam-se ainda mais claros. Uma forma simples de implementar este algoritmo é somar ou subtrair uma constante em cada pixel da imagem:

```
defina limiar
      defina       constante_splitting
            para linha = 1 até quantidade_linhas
           para coluna = 1 até quantidade_colunas
                se pixel_linha,coluna < limiar
                    pixel_linha,coluna ← pixel_linha,coluna - constante_splitting
                   senão
                    pixel_linha,coluna ← pixel_linha,coluna + constante_splitting
                fim se
          fim para
      fim para
```

Figura 6: Algoritmo que representa a Limiarização e o Splitting

Na fase seguinte ao processo de suavização, temos a segmentação que para GONZALEZ e WOODS (2000) é uma das principais etapas do processamento de imagens, pois é nela que a imagem a ser processada é dividida em partes menores e individualmente, obtendo dessa forma um melhor resultado. Esta técnica pode ser utilizada de diversas formas, com a intenção de captar detalhes minúsculos de uma foto ou vídeos, como por exemplo, em Ultrassom ou exames de Raio-X. Neste trabalho está técnica será aplicada para captar detalhes e desenhos das asas das borboletas, com o objetivo de otimizar o processo de identificação.

Outra questão que merece destaque é a utilização de recursos computacionais que proporcionem alto desempenho, uma vez que as imagens médicas apresentam grande volume de dados para processamento e este processamento envolve, geralmente, milhares de operações em tempo real (NUNES 2011).

Dando sequência ao processo, a próxima fase, é a etapa de Representação, esta fase tem por objetivo determinar de que forma ocorrerá o processamento, uma vez que a imagem já foi segmentada. Há duas opções: por fronteira ou por região. A representação por fronteira analisa o contorno, os cantos e os limites da imagem, propicia para este trabalho, uma vez que a maioria das borboletas pode ser diferenciada pelo formato das asas. Já a representação por região a imagem é analisada num contexto mais profundo, pois esta técnica é indicada quando se quer buscar maiores detalhes de dentro da imagem.

Junto com a fase de Representação pode-se destacar, a etapa de Descrição, onde os dados extraídos e obtidos são analisados, quantificados e preparados para a última etapa do processo.

Por fim, vem a etapa de Reconhecimento e Interpretação, nesta etapa as informações obtidas são nomeadas e agrupadas de acordo com o grau de similaridade entre elas. Nesta fase, o algoritmo de reconhecimento determinará a espécie da imagem obtida.

2.3 Monitoramentos da Biodiversidade

Neste tópico será abordado a relação das borboletas com o ambiente em que vive, além das diferentes formas que são monitoradas.

Segundo BITAR& ORTEGA (1998), Monitoramento Ambiental consiste na realização de medições e/ou observações específicas, dirigidas a alguns poucos indicadores e parâmetros, com a finalidade de verificar se determinados impactos ambientais estão ocorrendo, podendo ser dimensionada sua magnitude e avaliada a eficiência de eventuais medidas preventivas adotadas. O processo de monitoramento ambiental tem como principais objetivos verificar os impactos em um determinado ambiente, dimensionar seu tamanho, avaliar e rever medidas eficientes a serem tomadas.

Para MAGNUSSON (2013), métodos padronizados de monitoramento são tão importantes para a avaliação dos serviços ecossistêmicos como para outros aspectos relacionados à biodiversidade.

Figura 7: Imagem da realização do monitoramento ambiental

O monitoramento e observação das espécies se dão através da coleta, onde podem ser usadas duas técnicas distintas: rede de coleta de insetos, ou armadilhas nas árvores, onde as borboletas são atraídas por iscas de frutas. Usando as armadilhas é possível diminuir o problema de falta de experiência ou prática em coletar insetos (VIEIRA, 2010), dessa forma garantindo que qualquer membro de uma equipe de pesquisa possa realizar esta tarefa.

As borboletas integram um sistema maior denominado Meio Ambiente, garantindo que dessa forma ela tenha uma função dentro do processo. As borboletas que visitam flores podem carregar grãos de pólen, ajudando na reprodução de plantas (VIEIRA, 2010).

Uma outra caracteristica das borboletas é que elas funcionam como bioindicadores, refletindo o que acontece em um ecossitstema. Quantas espécies aparecem, se são abundantes, se ocorrem durante todo o ano ou só em alguns meses, onde estão ocorrendo, se estão aparecendo espécies que não eram vistas anteriormente. Estas observações podem indicar o quanto uma área esta sendo alterada, naturalmente ou pela ação do homem. (VIEIRA,2010).

A quantidade de espécies e suas variedades modificam conforme o ambiente habitado por elas, isso se explica pelo fato da comunidade de borboletas ser afetada diretamente pelas condições climáticas ou pela ação do homem sobre o meio. Com as mudanças que ocorrem quando se derruba a mata, espécies comuns podem se tornar raras e outras que não eram vistas, podem aparecer aos montes (VIEIRA, 2010).

As diferenças entre macho e fêmea variam de espécie para espécie, sendo que em alguns casos elas possuem as mesmas cores e só podem ser diferenciadas pelas diferenças de tamanho, onde as fêmeas são maiores, e em outros casos podem não ter características semelhantes entre os sexos, que são classificadas como duas espécies.

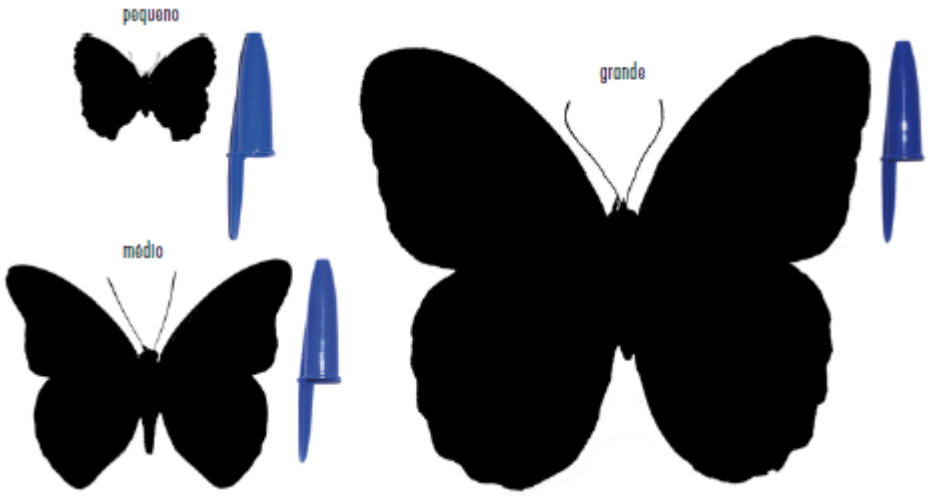

Figura 8: Tamanho médio de uma borboleta.
Fonte: Guia de Identificação ICMBio

A coleta de dados no campo é cara, difícil e as vezes perigosa. Ela não só exige uma estreita interação com os organismos, mas também uma estreita colaboração com muitas pessoas diferentes (MAGNUSSON, 2013). Ao serem colhidas, as espécies são encaminhadas ao centro de pesquisa para realizar a classificação e catalogação. Após serem coletadas, todas as borboletas recebem um número de registro e etiqueta com o nome científico e outros dados da coleta (VIEIRA, 2010). Pesquisas espacialmente orientada resultam em muitos tipos de dados, tais como planos de pesquisa, mapas, medições, publicações científicas, fotografias e informações sobre coleções biológicas, cada um com requisitos de armazenamento e de acessos diferentes (BILLICK, 2010). O manuseio e a classificação são processos que exigem muita atenção e cuidados, inclusive no local em que serão armazenadas, pois, as espécies coletadas serão estudadas por outros pesquisadores posteriormente. A segurança do local em que as espécies capturadas serão armazenadas é necessária tomar precauções com a incidência de luz, uma vez que as cores das borboletas podem ser alteradas de acordo com a claridade.

Figura 9: Coleta de borboletas com a rede específica para a atividade.

A atividade de monitorar uma determinada espécie pode ser considerada um experimento social, uma vez que, pessoas nativas de uma região e voluntários serão sempre necessárias para o auxílio das pesquisas. O acesso ao local de pesquisa é um atributo importante (MAGNUSSON apud. LINDENMAYER & LIKENS 2010). A maior parte da biodiversidade do mundo fica longe de grandes centros e logísticas complexas podem ser necessárias para se chegar aos lugares das pesquisas (MAGNUSSON 2013).

3. Trabalhos Relacionados

Foi realizada uma pesquisa bibliográfica com o objetivo de verificar a existência de trabalhos relacionados a este, baseada nos métodos para atingir os objetivos traçados previamente e nos resultados obtidos. Foram analisados de forma geral quatro trabalhos nas áreas de processamento de imagens e desenvolvimento para a plataforma que utilizaram a linguagem Java em seu desenvolvimento. Detalhes sobre os trabalhos serão apresentados nas próximas seções.

3.1- *Desenvolvimento de Aplicação Android para Reconhecimento Óptico de Caracteres em Brinco de Identificação Animal*

Este trabalho tem como objetivo inicial substituir o uso de um bastão leitor, que se conecta o *smartphone* à balança para realizar a verificação do número, pela própria câmera do celular. O autor optou por usar o *OpenCV*, que é uma biblioteca para desenvolvimento de aplicativos na área de visão computacional, que oferece suporte ao *Android*, e também utilizou o *Tesseract* que se trata de um software de reconhecimento óptico de caracteres, que dessa forma diminui a variação de erro de reconhecimento por parte do usuário.

Na área de processamento de imagens, o autor utilizou a técnica de binarização ou Limiarização, como forma de captar e distinguir letras do espaço de fundo. Utilizou também a segmentação, onde foi possível “desmontar” a imagem e utilizar somente a parte que continha a escrita que era o objetivo de sua pesquisa, essa forma ganhando tempo de processamento, analisando somente a parte necessária da imagem e não ela como um todo.

Segundo o próprio autor, o resultado foi satisfatório, pois, o software desenvolvido obteve uma média de 82% de acerto, levando em conta amostragens pequenas, e foi satisfatório também no quesito processamento de informação também, uma vez que os resultados retornavam a 2800 ms.

Figura 10: Imagem comparativa do resultado da ferramenta

Para trabalhos futuros, o autor sugere melhorias no algoritmo de processamento de imagens a fim de ficar menos vulnerável à mudanças de luminosidade, rotação e melhoria no reconhecimento de imagens tremidas. Sugere também a integração do seu sistema com outro software, o Sisgado, que se trata de um sistema de identificação animal que é compatível com o sistema *Android*.

3.2 - *Aplicativo de Reconhecimento de Imagens em Dispositivos Móveis para Ambientes Previamente Mapeados*

O objetivo inicial do trabalho foi avaliar de maneira criteriosa o comportamento da ferramenta que foi tema central do trabalho, revendo as técnicas de geoprocessamento e geoposicionamento. O SIROAM (Sistemas Integrado de Reconhecimento de Objetos em Ambientes Mapeados) funciona como um inventário online, onde os objetos cadastrados podem ser especificados e detalhados de acordo com o lugar previamente referenciado. O sistema conta com um acesso via Web para cadastro do local onde os objetos serão monitorados, enquanto a plataforma móvel atua como meio de captação de imagens para a identificação do sistema.

O SIROAM ainda está em fase de desenvolvimento, a ferramenta é recente, tendo sido apresentada em 2014 e roda na grande maioria dos dispositivos móveis que possuem o *Android*, GPS e câmera.

Para efetivar e validar o sistema foram testados ambientes e objetos captados e 2D e 3D, sob estes aspectos, foram feitos dois testes, o primeiro que visava a qualidade e confiabilidade, verificando como o software se comportaria diante de algumas variáveis do local, como diferença de luminosidade, fotos captadas de diferentes ângulos e imagens captadas com o flash do dispositivo ligado e também desligado.

Figura 11: Tela de apresentação da ferramenta

O segundo teste visou avaliar a eficiência do algoritmo, levando em consideração o tempo de processamento. Os resultados obtidos mostraram que é possível identificar um ambiente utilizando técnicas de processamento de imagens em ambientes previamente mapeados. O SIROAM apresentou resultados satisfatórios nos testes de luminosidade e processamento.

Para projetos futuros, o autor sugere melhorias no sistema, como na interface gráfica do ambiente da plataforma móvel, e também melhorias no problema de escalabilidade.

3.3 - *SmartCoompras: Desenvolvimento de um aplicativo para celulares Smartphone*

Este trabalho apresenta como temas principais o processamento de imagens e mineração de dados. O objetivo principal foi desenvolver uma ferramenta de compras mobile que utilizasse o *Android* como sistema operacional, a proposta do software é realizar a captura da imagem do código de barras tirada a partir da câmera do dispositivo móvel, e aplicar conceitos de mineração de dados com o objetivo de mostrar ao usuário possíveis produtos que ele gostaria de adquirir.

Analisando somente a área de processamento de imagens, dentro do sistema, a imagem é transformada em uma versão digital, a cor desta imagem é tratada, restaurada, segmentada e representada e reconhecimento de padrões.

A imagem central do processamento do sistema é do código de barras, onde dentro da fase segmentação ocorre a Limiarização, que conforme dita anteriormente, também será útil neste trabalho, uma vez que para uma foto de código de barras a “binarização” da imagem auxilia bastante no processamento da mesma por se tratar de uma imagem monocolor. Técnicas como Métodos Baseados em Região ou Media Vizinhança e Mediana Vizinhança também foram utilizadas para verificar a similaridade de pixels entres os vizinhos e agrupá-los em regiões próximas, e a partir daí fazer o Reconhecimento de Padrões e finalizar o processamento e identificação da imagem.

Figura 12: Tela de Apresentação do Sistema

Os autores destacam a importância da conversão da imagem em escala de cinza antes de iniciar o procedimento de processamento de imagem, e desta forma eliminando muitos ruídos que a imagem colorida possivelmente teria. Outro ponto a se destacar é que a foto tirada inicialmente pode não mostrar os traços retos do código de barras, precisando passar por um tratamento prévio em que o contorno é realçado, e a partir daí se faz o histograma, para o que software leia da maneira correta.

Este processamento digital tem diversas funções segundo os autores, como: reconhecimento de placa de veículos, reconhecimento de caracteres, análise de cromossomos, entre outras aplicações.

Figura 13: Tela de Apresentação dos resultados do Sistema

O aplicativo criado mostrou-se mais eficiente na parte de mineração de dados do que no processamento de imagens em si. Pois, foi detectado um problema por parte dos desenvolvedores quanto ao modo de tirar a foto, com o risco de o programa não fazer a leitura se não for feita da forma correta. Quanto a trabalhos futuros na área de processamento digital, o grupo salientou que o algoritmo de reconhecimento de imagem pode ser melhorado, assim como as bibliotecas utilizadas para o desenvolvimento.

4. Metodologia

O presente trabalho caracteriza-se como uma pesquisa experimental, que segundo GIL (2007), consiste em determinar um algoritmo capaz de reconhecer espécies de borboletas com o auxílio de um algoritmo que verifique suas principais características a fim de otimizar o tempo de catalogação de espécies, visto que atualmente este procedimento se dá de forma manual.

Será desenvolvida uma ferramenta para dispositivos móveis que utilizará a plataforma *Java Desktop* e após o desenvolvimento serão realizados testes por especialistas de processamento de imagens, biólogos e pesquisadores utilizando a ferramenta desenvolvida. Para o desenvolvimento deste trabalho serão necessários os seguintes passos metodológicos:

a) Revisão bibliográfica;
b) Estudo de livros e artigos da área de processamento de imagens, plataforma e desenvolvimento Java, monitoramento e biodiversidade;
c) Estudo sobre as variáveis que influenciam o processamento digital de imagem e seus métodos;
d) Estudo da biblioteca *JAI (*Java *Advanced Imaging);*
e) Selecionar as possíveis interfaces da ferramenta a ser desenvolvida;
f) Teste da ferramenta desenvolvida, que será realizado por especialistas de Processamento de imagem, pesquisadores e biólogos.

5. Desenvolvimento

5.1 – A ferramenta construída

Foi desenvolvida uma ferramenta na linguagem Java com o objetivo de realizar testes e validar o algoritmo de reconhecimento de imagem proposto por este trabalho utilizando a IDE de desenvolvimento NetBeans 8.1. Foi escolhida esta linguagem devido à facilidade de interação desta, em relação à linguagem Java.

Para a implementação das telas foi utilizado o Java Swing simples com alguns botões e uma tabela que funciona como lista armazenando e apresentando o conteúdo gravado. Para a implementação de reconhecimento e processamento e imagem dentro da aplicação foi utilizada a biblioteca JAI (Java *Advanced Imaging*) levando em conta que esta biblioteca possui uma grande quantidade de recursos de processamento de imagem já implementados. Outra característica importante desta API de manipulação de imagens, é que, as imagens carregadas dentro da biblioteca são convertidas para um padrão de formato único, podendo aceitar vários tipos de formato de imagem.

Foram desenvolvidos alguns extratores de imagem, com o objetivo de quantificar as características desejadas. Os extratores de imagem desenvolvidos foram:

Conjunto dos extratores implementados		
Cor	Média do Histograma em Nível de Cinza	Calcula através do histograma, a média da frequência dos pixels. Objetivo: obter uma informação relativa ao tom da imagem (mais clara ou mais escura)
	Média do Histograma em RGB	Calcula através do histograma, a média da frequência dos pixels. Objetivo: obter uma informação relativa ao tom da imagem (mais clara ou mais escura)
	Desvio Padrão do Histograma em Nível de Cinza	Calcula através do histograma, o desvio padrão da frequência dos pixels. Objetivo: obter a informação da variação de cores em uma imagem (se houve muita ou pouca variação)

	Desvio Padrão do Histograma em RGB	Calcula através do histograma o desvio padrão da frequência dos pixels. Objetivo: obter a informação da variação de cores em uma imagem (se houve muita ou pouca variação)
	Média dos Pixels em Nível de Cinza	Calcula através da varredura individual dos pixels, a média de suas cores. Objetivo: obter uma informação relativa ao tom da imagem (mais clara ou mais escura)
	Média dos Pixels em RGB	Calcula através da varredura individual dos pixels, a média de suas cores. Objetivo: obter uma informação relativa ao tom da imagem (mais clara ou mais escura)
	Desvio Padrão dos Pixels em Nível de Cinza	Calcula através da varredura individual dos pixels, o desvio padrão de suas cores. Objetivo: obter a informação da variação de cores em uma imagem (se houve muita ou pouca variação)
	Desvio Padrão dos Pixels em RGB	Calcula através da varredura individual dos pixels, o desvio padrão de suas cores. Objetivo: obter a informação da variação de cores em uma imagem (se houve muita ou pouca variação)
Textura	Contraste	Por meio da matriz de co-ocorrência, usando os parâmetros de análise dos pixel imediatamente vizinhos e à esquerda, para a verificação da probabilidade de ocorrência de determinado padrão de cor. Com essa probabilidade é calculado o contraste. Objetivo: Determinar a variação dos pixels.

Tabela adaptada de BERGAMASCO (2010)

```
public class CalcHistograma {
        public double media, desvio, incl;
        public double mediaR, mediaG, mediaB ;
        public double desvioR, desvioG, desvioB;
        public double inclR, inclG, inclB;


        public double[] mediaDesvioCinza(String path){
                double sum = 0;
                double mediaIn =0;
                media = 0;
                desvio = 0;
                incl=0;
                double sumDesvio = 0;
                double sumIncl = 0;

                /* Transforma em cinza*
                 * Equação de Luminância:
                 * Relação entre XYZ e RGB+1. Saída = uma banda, por isso uma linha*/
                double[][] bandCombine = {{0.5f,0.71f,0.25f,0.0f}};
                double[] vetorCaracteristicas = new double [2];
                PlanarImage image = JAI.create("fileload",path);

                /*Transformando em níveis de cinza*/
                PlanarImage imageBand = JAI.create("BandCombine",image, bandCombine);
                ParameterBlock pb = new ParameterBlock();
            pb.addSource(imageBand);
            pb.add(null); // The ROI.
```

Figura 14: Trecho de código da implementação de um dos extratores. Adaptado de BERGAMASCO (2010).

Todos estes extratores acima citados realizam cálculos separadamente dentro do programa, retornam valores e são gravados em um banco de dados PostgreSQL. Foi escolhida a versão PostgreSQL 9.5, por ser um banco de dados relacional de fácil implementação e implantação já que o foco do sistema não está na performance do banco. O banco de dados da ferramenta possui duas tabelas sendo que em uma é armazenada a imagem e em outra sua característica conforme a figura abaixo:

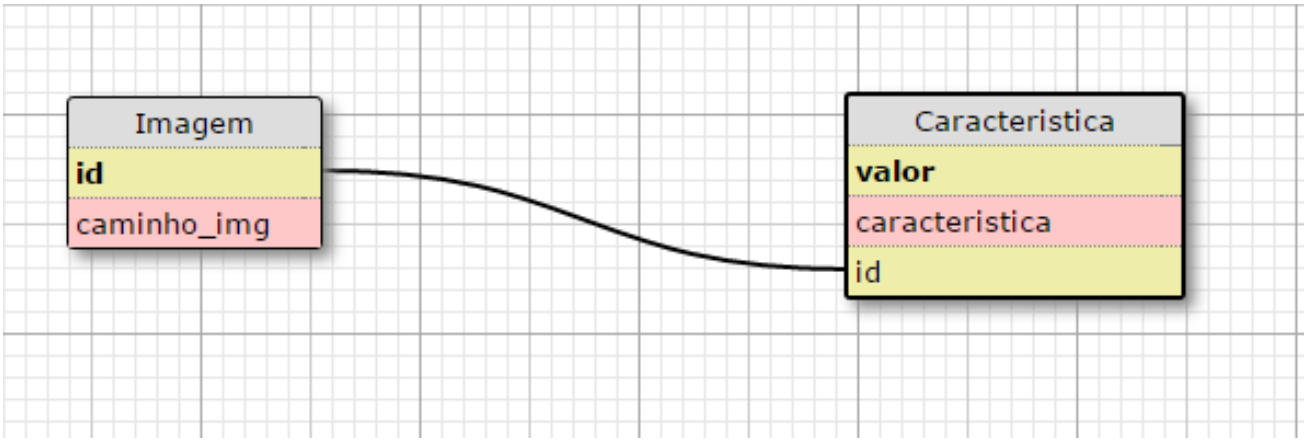

Figura 15: Tabelas do banco de dados da aplicação

A tabela “Imagem” armazena o ID da imagem padrão, seu caminho e seu nome, e está conectada por uma chave estrangeira a outra tabela denominada “Característica”, que é composta por três atributos: ID, o nome da característica que foi extraída e seu valor.

Só serão gravadas no banco da aplicação somente as imagens padrão, as imagens para efeito de teste e comparação não serão gravadas, pois, isto poderia gerar redundância de resultado caso duas imagens muito similares fossem carregadas retornando um veredicto muito aproximado entre ambas e consequentemente um resultado ilusório.

Para realizar o cálculo da comparação de imagens, após estas estarem gravadas no banco, foram analisadas duas formas de calcular a distância de pixels dentro da matriz da imagem: Distância Euclidiana e Distância *Manhattan*. A distância *Manhattan*, é obtida através da soma das diferenças absolutas entre as suas coordenadas, enquanto a distância Euclidiana, é a raiz da somatória das diferenças absolutas entre suas coordenadas ao quadrado (BERGAMASCO, 2010).

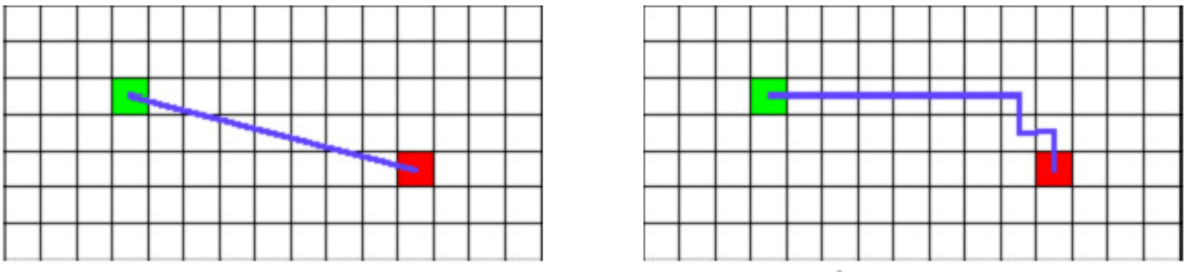

Figura 16: Distância Euclidiana e Distância *Manhattan* – adaptado de BERGAMASCO (2010)

$$\sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}.$$

Figura 17: Fórmula da Distância Euclidiana

Escolheu-se a Distância Euclidiana, pois, ela realiza o cálculo de similaridade de forma mais otimizada em relação a Distancia *Manhattan*, conforme a figura acima.

```
public double [] calculaDistancia(double[][] matriz, int id){
                double[] diferenca = new double [24];
                id = id-1;
                for(int i=0;i <24;i++)
                {
                        double sum = 0;
                        double raiz = 0;
                        for(int j=0;j<16;j++)
                        {
                                sum = sum + (Math.pow(((matriz[id][j]) - (matriz[i][j])),2));

                        }

                        raiz = Math.sqrt(sum);
                        diferenca[i] = raiz;
                }
                System.out.println("Diferença!");
                for(int k=0;k <24;k++){
                        System.out.println(diferenca[k]);
                }
                return diferenca;
}
}
```

Figura 18: Código usado para o cálculo da distância entre pontos utilizando o conceito de Distância Euclidiana. Adaptado de BERGAMASCO (2010)

Referente à escolha do formato das imagens, foi escolhido o formato Bitmap, pois, os testes iniciais feitos em PNG apresentaram alguns problemas, apresentando erros em algumas partes do sistema. As imagens padrão gravadas no sistema foram digitalizadas com fundo branco, e desta forma as imagens que serão carregadas com o intuito classifica-las também foram digitalizadas com o fundo branco para não apresentar nenhuma alteração grave de média de cores. Caso o banco de imagem fosse digitalizado e o carregamento de “imagens-teste” fossem fotos capturadas com fundo branco, os resultados obtidos pelo programa seriam fictícios e imprecisos, uma vez que, o sistema faz o cálculo baseado nas cores em toda a área da foto, sendo assim, fatores como a cor de fundo, sombras, luminosidade e ângulo em que a foto foi retirada seriam variáveis que contribuiriam para a inconsistência do resultado. Isto justifica o uso de somente imagens digitalizadas nos testes que serão apresentados posteriormente.

Figura 19: Exemplo de imagem padrão utilizada pelo sistema

Após o carregamento das imagens determinadas como “padrão”, o sistema extrai todas as características da mesma e armazena-as no banco de dados, juntamente com o nome da espécie de borboleta a qual a imagem se refere. Feito isto, o usuário tem a opção de escolher a imagem a ser testada clicando no botão “Testar Imagem” (Figura 20), dessa forma, a imagem testada é comparada individualmente, através do algoritmo da Distância Euclidiana, com todas as outras carregadas e gravadas previamente. Após o termino da comparação, uma tela é retornada ao usuário onde aparece o nome e a imagem que foi realizado o teste, e as duas imagens, com seus respectivos nomes, mais similares de acordo com o algoritmo proposto.

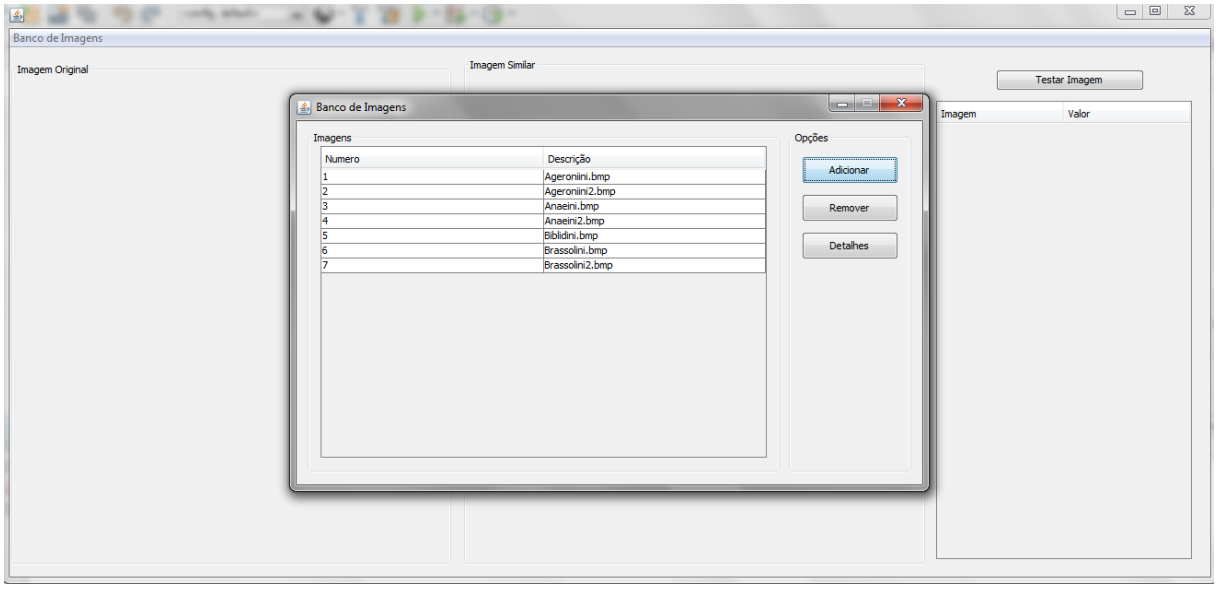

Figura 20: Tela para carregamento de imagem no banco de dados

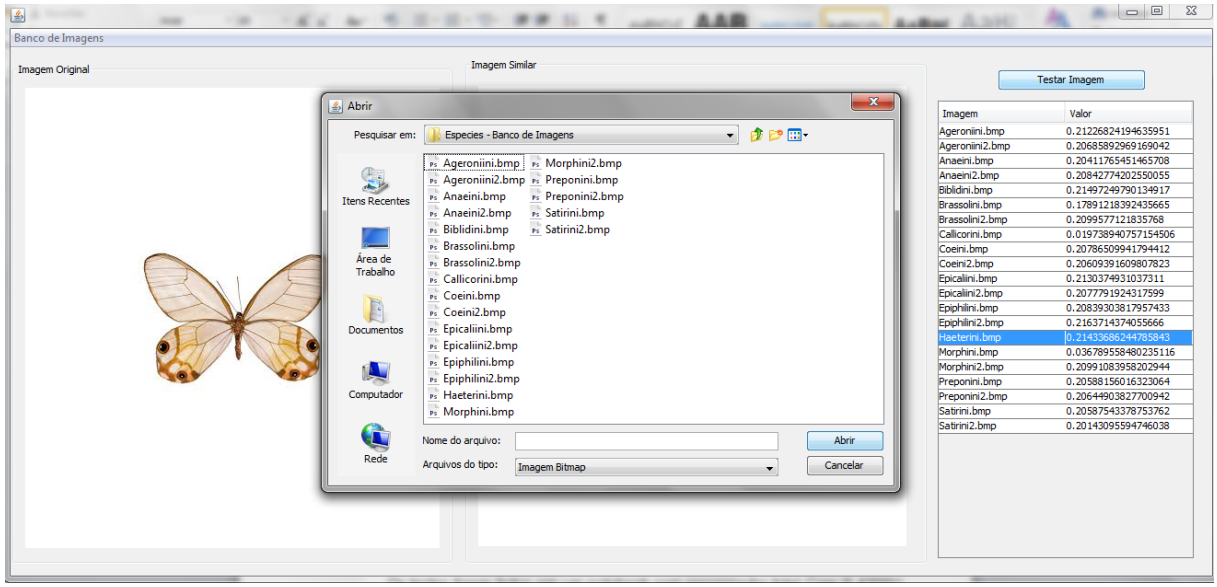

Figura 21: Tela para escolha da imagem para upload no sistema ao clicar em “Testar Imagem”

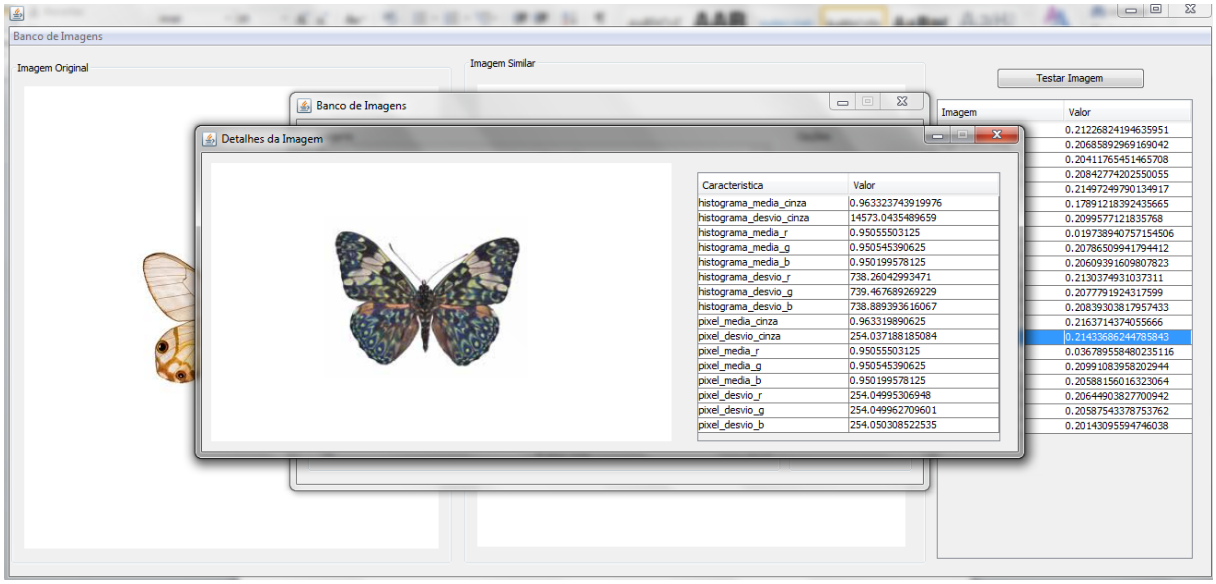

Figura 22: Tela de exibição dos valores extraídos pelo programa, na aba “Detalhes”.

5.2 – Testes

5.2.1 – Especificações Iniciais

Os testes foram feitos em um notebook com processador Intel Core i5-4200U CPU @ 1.60GHz 2.30GHz com 6GB de memória RAM e Windows 7Ultimate de 64 Bits.

Todas as imagens, tanto para o teste I quanto para o teste II, foram digitalizadas e salvas em arquivo em formato Bitmap (.bmp), pois, este formato foi o que melhor se comportou dentro da aplicação, uma vez que o formato PNG (*Portable Network Graphics*) apresentou erros de recuperação das imagens no algoritmo de escala e média de cinza, já o formato JPEG (*Joint Photographic Experts Group*) apresentou inconsistência em alguns testes por razões desconhecidas, deste modo o Bitmap foi adotado, para este trabalho como formato padrão em todos os casos de teste.

5.2.2 – Teste Genérico

Foram realizados dois tipos de testes: O primeiro feito com 4 imagens, sendo 3 imagens-padrão e 1 imagem de teste, na qual as imagens padrão sofreram pequenas alterações de cor em relação a imagem de teste. Este teste tinha objetivo de verificar a funcionalidade e eficiência do algoritmo de distância

Euclidiana em uma base simples. As quatro imagens que foram utilizadas para este teste possuíam o mesmo tamanho.

5.2.3 – Teste Específico

No segundo teste a dinâmica foi um pouco diferente, porém, seguindo a mesma lógica, todas as imagens deste segundo teste foram retiradas do Guia de Identificação de Tribos de Borboletas Frugívoras da ICMBio (Instituto Chico Mendes de Conservação da Biodiversidade), recortadas e colocadas em um arquivo padrão, no tamanho de 500 pixels de altura por 500 pixels de largura, esta medida foi necessária, pois em um pré-teste, observou-se que o tamanho da imagem influenciou no resultado final. Uma das características da imagem extraída pelo sistema é a quantidade de pixels contida nela, mesmo ambas as imagens contendo a mesma espécie de borboleta, e dessa forma obteve-se resultados mais consistentes.

Este teste tinha o objetivo de verificar a funcionalidade e eficiência do algoritmo de distância Euclidiana em uma base mais complexa e com maior diversidade de espécies.

A figura a seguir mostra todas as imagens que foram gravadas no banco, consideradas como padrão, todas elas tiveram suas características extraídas e foram gravadas individualmente com seu nome identificando-as.

Figura 23: Imagens Padrão gravadas no banco para realização do teste Específico

Ao total no banco de imagens foram carregadas 21 imagens padrão, contendo algumas espécies com dois exemplares de famílias diferentes, porém pertencendo à mesma tribo, como o guia de identificação se refere.

Na figura abaixo estão todas as imagens de teste que compõe o teste específico, o sistema permite carregar uma imagem de teste por vez, onde a comparação é feita de forma individual, e assim como nas imagens gravadas no banco, estas, também passam pelo mesmo processo de extração de características e são comparadas de acordo com a similaridade com todas as outras, mas não são guardadas no banco de dados.

Figura 24: Imagens de Teste gravadas no banco para realização do teste específico.
Fonte: Guia de identificação da ICMBio

Ao total foram carregadas 9 imagens de teste, contendo algumas espécies com dois exemplares de famílias diferentes, porem pertencendo à mesma tribo, como o guia de identificação se refere.

5.3 – Resultados

Esta seção tem como objetivo apresentar os resultados obtidos pelos testes feitos na seção anterior.

5.3.1 – Resultados Teste Genérico

Figura 25: Imagem Teste para o teste genérico

Figura 26: Imagem Padrão 1 para o teste genérico

Figura 27: Imagem Padrão 2 para o teste genérico

Figura 28: Imagem Padrão 3 para o teste genérico

Como neste teste não se tinha o objetivo de identificar a espécie de borboleta e sim a funcionalidade do algoritmo, foi utilizada uma foto genérica e suas

variações. Para facilitar o entendimento dos resultados foi tomada como base a nomenclatura que segue:

Resultado Teste Genérico			
	Imagem Padrão 1	Imagem Padrão 2	Imagem Padrão 3
Imagem Teste	97,90%	95,40%	83,70%

Tabela de apresentação dos resultados do Teste Genérico

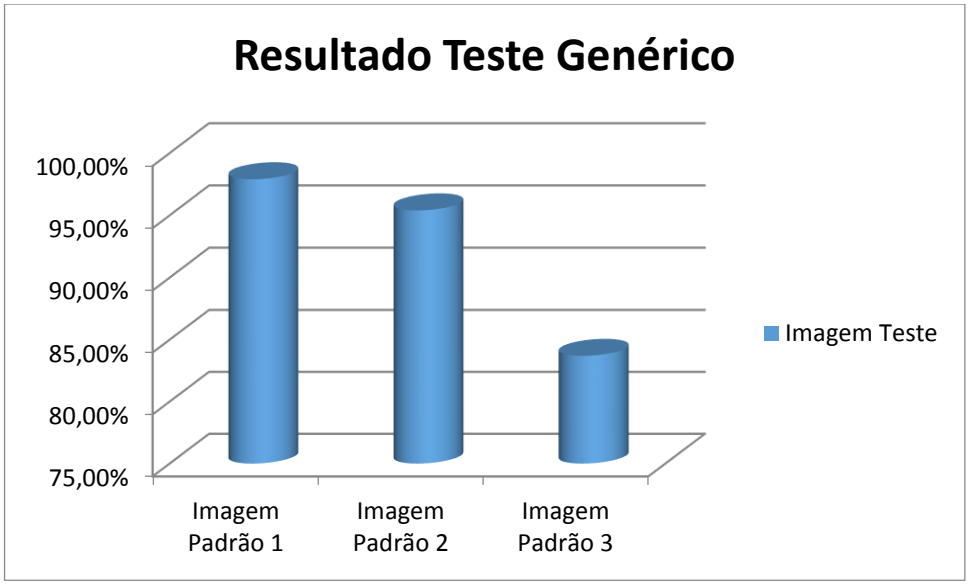

Figura 29: Imagem do gráfico dos resultados obtidos no Teste Genérico

Os resultados obtidos deste primeiro teste foram bem satisfatórios, visto que a imagem que sofreu a menor alteração (Imagem Padrão 1) apresentou 97,9% de similaridade com a imagem teste e deste mesmo modo o algoritmo conseguiu retornar resultados consistentes para a Imagem Padrão 2, obtendo 95,4% de similaridade com a imagem teste e pôr fim a Imagem Padrão 3 que obteve 83,7% de similaridade com a imagem teste. Neste teste foi possível obter a porcentagem, pois a base não estava tão carregada e os resultados obtidos foram multiplicados por 100 para obter a porcentagem, diferente do Teste Específico.

5.3.2 – Resultados Teste Específico

Conforme dito anteriormente, foram utilizados neste segundo teste uma base de dados mais complexa contendo ao menos uma imagem de cada “tribo” de

espécies de borboletas que estão caracterizadas do Guia de Identificação de Espécies de Borboletas da ICMBio, e os resultados a seguir serão apresentados em forma de 9 subtestes, com o objetivo de verificar a semelhança por espécie. Devido ao fato de usar o algoritmo de distância Euclidiana, usou-se como medida o maior número retornado como mais similar, portanto retornaram-se números entre 0 e 1, e quanto mais próximo do 1, mais similar é a foto que está sendo testada.

Os resultados apresentados serão em forma de tabela com as 5espécies que tiveram os maiores valores retornados pelo programa, adotou-se este método para facilitar o entendimento dos resultados obtidos.

5.3.2.1 – Teste A (Biblidini)

Neste teste o resultado apareceu com coerência à pesquisa realizada retornando ao usuário a espécie Biblidini como a mais similar entre as outras imagens carregadas no banco de dados.

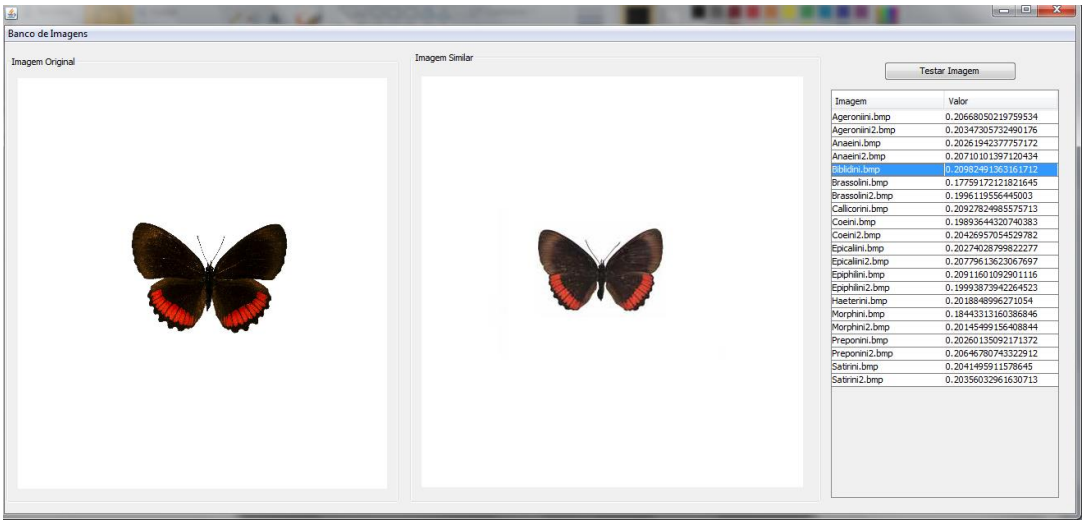

Figura 30: Tela de apresentação dos resultados da aplicação

5.3.2.2 – Teste B (Brassolini)

Figura 31: Imagem da espécie Brassolini

Neste caso de teste a espécie Brassolini foi testada e o resultado não foi coerente à pesquisa realizada retornando ao usuário outras espécies como mais similares entre as outras imagens carregadas no banco de dados. A espécie Brassolini não aparece nem entre as 5 mais parecidas, estando em último lugar, sendo a mais diferente.

5.3.2.3 – Teste C (Coeini)

Figura 32: Imagem da espécie Coeini

Neste caso de teste a espécie Coeini foi testada e o resultado não foi coerente à pesquisa realizada porem a espécie Coeini apareceu entre as 5 mais similares. A espécie Coeini aparece como a terceira mais similar estando atrás apenas da espécie Haeterini e Epicaliini.

5.3.2.4 – Teste D (Epicaliini)

Figura 33: Imagem da espécie Epicaliini

Neste caso de teste a espécie Epicaliini foi testada e o resultado não foi coerente à pesquisa realizada retornando ao usuário outras espécies como mais similares entre as outras imagens carregadas no banco de dados. A espécie Epicaliini não aparece nem entre as 5 mais parecidas.

5.3.2.5 – Teste E (Haeterini)

Figura 34: Imagem da espécie Haeterini

Neste caso de teste a espécie Haeterini foi testada e o resultado apareceu com coerência à pesquisa realizada retornando ao usuário a espécie Haeterini como a mais similar entre as outras imagens carregadas no banco de dados.

5.3.2.6 – Teste F (Haeterini 2)

Figura 35: Imagem da espécie Haeterini para o segundo teste da espécie

Neste segundo teste da espécie Haeterini verificou-se a necessidade devido ao fato de esta espécie ter a coloração predominantemente clara, justifica-se o segundo teste da mesma espécie com imagem diferente para verificar o comportamento da ferramenta diante deste cenário. O resultado obtido foi semelhante ao teste anterior, sendo a espécie Haeterini reconhecida pelo aplicativo como a mais similar entre as outras carregadas e gravadas no banco de dados.

5.3.2.7 – Teste G (Morphini)

Figura 36: Imagem da espécie Morphini

Neste caso de teste a espécie Morphini foi testada e o resultado não foi tão coerente à pesquisa realizada, retornando ao usuário outras espécies como mais similares entre as outras imagens carregadas no banco de dados. A espécie Morphininão aparece nem entre as 5 mais similares

5.3.2.8 – Teste H (Preporini)

Figura 37: Imagem da espécie Preporini

Neste caso de teste a espécie Preporini foi testada e o resultado apareceu de forma mais concreta, com a espécie Preporini em segundo lugar entre as mais similares retornadas pelo sistema, este resultado pode ser considerado como satisfatório levando em conta que há muitas outras espécies.

5.3.2.9 – Teste I (Satirini)

Figura 38: Imagem da espécie Satirini

Neste caso de teste a espécie Satirini foi testada e o resultado não foi tão coerente à pesquisa realizada, retornando ao usuário outras espécies como mais similares entre as outras imagens carregadas no banco de dados. A espécie Satirinia parece em quinto lugar como a mais similar.

5.3.2.10 – Apresentação dos Resultados

Todos os dados obtidos dos testes anteriormente apresentados foram colhidos e apresentados na tabela que segue abaixo:

Resultados do Teste Específico							
Teste A		Teste B		Teste C		Teste D	
Biblidini	0,2098	Haeterini	0,2062	Haeterini	0,2073	Haeterini	0,1921
Calicorini	0,2092	Epicaliini	0,2013	Epicaliini	0,2023	Epiphiliini	0,1876
Epiphiliini	0,2091	Preponini	0,1995	Coeini	0,2019	Brassolini	0,1874
Epicaliini	0,2077	Epiphiliini	0,1992	Epiphiliini	0,2018	Anaeini	0,1873
Anaeini	0,2071	Coeini	0,1975	Ageroniini	0,2008	Morphiini	0,1867

Resultados do Teste Específico							
Teste E		Teste F		Teste G		Teste H	
Haeterini	0,2123	Haeterini	0,2115	Haeterini	0,208	Epiphiliini	0,2098
Anaini	0,2122	Epicaliini	0,2101	Epiphiliini	0,2047	Preporini	0,2074
Preporini	0,2096	Anaeini	0,2092	Calicorini	0,2029	Ageroniini	0,2052
Epiphiliini	0,2091	Coeini	0,2082	Ageroniini	0,2019	Coeini	0,2047
Brassolini	0,209	Preponini	0,2066	Biblidini	0,2011	Epicaliini	0,2045

Resultados do Teste Específico	
Teste I	
Haeterini	0.2115
Epiphiliini	0,2106
Biblidini	0,2086
Calicorini	0,2085
Satirini	0,2076

Esta tabela apresenta os resultados obtidos de forma direta pelo sistema, onde constam as informações com a identificação do teste processado, o nome da espécie, e seu resultado obtido a partir da formula da distância Euclidiana. No ranking aparecem somente as 5 primeiras colocadas (mais similares com a imagem testada), ainda que o sistema retorna o valor de todas as imagens inseridas no banco independente de sua posição no resultado final.

6. Considerações Finais

Esta seção apresentará as considerações finais deste trabalho, dividido em dois tópicos: Conclusão e Trabalhos Futuros.

6.1 - Conclusão

Buscando inicialmente reconhecer ao menos uma espécie de borboleta, conclui-se que este objetivo foi atingido, pois como mostrado na seção anterior, foram reconhecidas duas espécie com precisão (Biblidini e Haeterini), e em um terceiro teste com a espécie Preporini, o sistema retornou a imagem como a segunda imagem mais similar, visto que o objetivo do sistema seria auxiliar um profissional desta área, apresentar uma espécie como segunda mais similar é considerado um resultado satisfatório, visto que reduz de 21 possibilidades para 2 possibilidades (neste caso de teste).

Dos 9 testes realizados 3 obtiveram o melhor resultado retornando a espécie correta na primeira posição, analisando este segundo teste, o sistema teve um aproveitamento de 33% de acerto.

O fato de ter usado um extrator para média de RGB, cálculo de RGB e outro para cada um destes componentes separados pode ter influenciado no resultado da comparação, visto que uma imagem escura com partes brancas pode ter a mesma média de uma imagem somente cinza. Utilizar estes extratores não são as melhores formas de se calcular a similaridade caso espera-se resultado com maior precisão, sendo necessário outros cálculos complementares para o auxílio da medição da Distância Euclidiana.

O algoritmo utilizado respondeu de forma muito satisfatória no primeiro teste onde foram usadas uma imagem de teste e suas variações a partir da imagem original.

A área de catalogação e identificação de borboletas, atualmente, é um processo que se dá de forma manual, sendo assim, este trabalho propôs uma nova abordagem e uma ferramenta como validação dos resultados visando otimizar este processo.

6.2 – Trabalhos Futuros

Como trabalhos futuros sugere-se o aprimoramento do código deste trabalho visando melhorar o tempo de resposta da aplicação e a precisão dos resultados utilizando outros cálculos complementares ao de média de RGB, média de cinza e outros cálculos que possam camuflar algum tipo de resultado.

Sugere-se também o desenvolvimento desta aplicação em outros tipos de plataforma como Web e Mobile, visando a praticidade do usuário final a plataforma *Android* é uma boa opção já que trabalha com código aberto, deve ser levado em consideração esta opção de funcionamento em modo off-line, mesmo que os recursos para um periférico deste porte sejam limitados, e para o desenvolvimento Web pode ser usado o JSF (Java Server Faces) que trata-se de uma especificação Java para a construção de interfaces de usuário baseadas em componentes para aplicações Web.

Referências

ARAUJO, Jaime Franklin Vidal. Vocabulário Básico de Recursos Naturais e Meio Ambiente**.** 2. ed. Rio de Janeiro: Instituto Brasileiro de Geografia e Estatística - Ibge, 2004.

BERGAMASCO, Leila Cristina Carneiro. Recuperação de imagens por conteúdo utilizando Lógica Fuzzy - um estudo de caso sobre imagens faciais. São Paulo:2010.

BILLICK, I. *Managing place-based data.The Ecology of Place.University of Chicago Press, Chicago, IL, USA,* 2010.

BITAR, O.Y & ORTEGA, R.D. Gestão Ambiental. In: OLIVEIRA, A.M.S. & BRITO, S.N.A. (Eds.). Geologia de Engenharia. São Paulo: Associação Brasileira de Geologia de Engenharia (ABGE), 1998. cap. 32, p.499-508

DEITEL, Paul; DEITEL, Harvey. Java Como Programar. 8. ed. Porto Alegre: Pearson Education, 2010.

FLORENZANO, T. G. Introdução à Geomorfologia. In: Florenzano T. G. (Org.) Geomorfologia: Conceitos e Tecnologias Atuais. 1ª ed. São Paulo: Oficina de Textos, v.1, 2008.

GIL, Antonio Carlos. Métodos e técnicas de pesquisa social. 5.ed. São Paulo: Atlas, 2007.

GONZALEZ, R.; WOODS, R.E. Processamento de Imagens Digitais. Editora Edgard Blucher, São Paulo, 2000.

HIRAMA, Edgar Tamio. Desenvolvimento de Aplicação Android para Reconhecimento Óptico de Caracteres em Brinco de Identificação Animal. São Carlos, 2014.

JOHNSON, Thiene M.. Java para Dispositivos Movéis: Desenvolvendo Aplicações com J2ME. São Paulo: Novatec, 2007.

LECHETA, Ricardo R. (Org.). Google *Android*: Aprenda a criar aplicações para dispositivos móveis com o *Android* SDK. 3. ed. São Paulo: Novatec, 2013.

MAGNUSSON, Willian; BRAGA NETO, Ricardo; PEZZINI, Flavia (Org.). Biodiversidade e Monitoramento Ambiental Integral. Manaus: Áttema Editorial, 2013.

MENDES, Douglas Rocha. Programação Java com Ênfase em Orientação a Objetos. 2. ed. São Paulo: Novatec, 2011.

NUNES, F. de Lourdes dos S. Processamento Gráfico para aplicações em saúde. Técnicas, requisitos, ferramentas, desafios e oportunidades. Universidade de São Paulo, Agosto 2011.

OLIVEIRA, Andressa Pereira de; FRAGOMENI, Bruno Noronha; NOGUEIRA, MildredhHarue. SMARTCOMPRAS: DESENVOLVIMENTO DE UM APLICATIVO PARA CELULARES SMARTPHONE. São Paulo, 2011.

SPRING: *IntegratingremotesensingandGISbyobject-orienteddatamodelling*. *Camara* G, Souza RCM, Freitas UM, Garrido J .*Computers&Graphics*, 1996.

SANTOS, Java AdvancedImaging API: A Tutorial - Tutorial desenvolvido por Rafael Santos, apresentado no SIBGRAPI04. <https://jaistuff.dev.java.net/docs/jaitutorial.pdf> Acessado em 09/07/2016.

TOMASEL, Tiago Souza. APLICATIVO DE RECONHECIMENTO DE IMAGENS EM DISPOSITIVOS MÓVEIS PARA AMBIENTES PREVIAMENTE MAPEADOS. Canoas, 2014.

VIEIRA, Eliane; RIBEIRO, Deise Helena Baggio, - São Paulo: Centro de P&D de Proteção Ambiental,2010

WATT, A. H., F. Policarpo - The Computer Image , Addison-Wesley Pub Co (Net); 1998