

INSTITUTO FEDERAL DO ESPÍRITO SANTO
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

WDNEI RIBEIRO DA PAIXÃO

**APLICAÇÃO DE TÉCNICAS DE RECONHECIMENTO DE
TEXTURAS COLORIDAS NO PROCESSO DE
CLASSIFICAÇÃO DE ESPÉCIES DE TARTARUGAS
MARINHAS**

SERRA

2015

WDNEI RIBEIRO DA PAIXÃO

**APLICAÇÃO DE TÉCNICAS DE RECONHECIMENTO DE
TEXTURAS COLORIDAS NO PROCESSO DE
CLASSIFICAÇÃO DE ESPÉCIES DE TARTARUGAS
MARINHAS**

Trabalho de Conclusão de Curso apresentado
à Coordenadoria do Curso de Bacharelado
em Sistemas de Informação do Instituto
Federal do Espírito Santo, como requisito
parcial para obtenção do título de Bacharel
em Sistemas de Informação.

Orientadora:

Prof^a Dr^a Karin Satie Komati

SERRA

2015

Wdnei Ribeiro da Paixão

Aplicação de Técnicas de Reconhecimento de Texturas Coloridas no Processo de Classificação de Espécies de Tartarugas Marinhas/ Wdnei Ribeiro da Paixão. – Serra, 2015-

38 p. : il. (algumas color.) ; 30 cm.

Orientadora: Karin Satie Komati

Monografia (Graduação) – Instituto Federal do Espírito Santo , 2015.

1. Software- Desenvolvimento. 2. Extração de características - Visão Computacional.
2. Classificação de espécies. I. Komati, Karin Satie. II. Instituto Federal do Espírito Santo. III. Título.

FOLHA DE APROVAÇÃO

Trabalho de Conclusão de Curso sob o título “Aplicação de Técnicas de Reconhecimento de Texturas Coloridas no Processo de Classificação de Espécies de Tartarugas Marinhas”, defendida por Wdnei Ribeiro da Paixão e aprovada em XX de XXXXXX de 2015, em Serra, Estado do Espírito Santo, pela banca examinadora constituída pelos professores:

Karin Satie Komati
Orientadora

Professor
Convidado 1

Professor
Convidado 2

Serra
2015

DECLARAÇÃO DO AUTOR

Declaro, para fins de pesquisa acadêmica, didática e técnico-científica, que o presente Trabalho de Conclusão de Curso pode ser parcial ou totalmente utilizado desde que se faça referência à fonte e aos autores.

Wdnei Ribeiro da Paixão

Serra, de Mês de 2015

*Este trabalho é dedicado às crianças adultas que,
quando pequenas, sonharam em se tornar cientistas.*

*“Não vos amoldeis às estruturas deste mundo,
mas transformai-vos pela renovação da mente,
a fim de distinguir qual é a vontade de Deus:
o que é bom, o que Lhe é agradável, o que é perfeito.
(Bíblia Sagrada, Romanos 12, 2)*

Resumo

Tartarugas marinhas possuem um ciclo de vida longo e complexo. Projetos de conservação marinha como o Tamar-ICMBio, monitoram estas espécies através de informações geradas por um sistema de informação, permitindo identificar a tendência populacional de cada espécie. Entretanto, cada espécie de tartaruga marinha possui características relativamente complexas, como padrão de cor ou textura em seu casco, que requerem um especialista para classificá-las. Neste trabalho, comparamos duas técnicas de extração de características para classificação de espécies de tartarugas marinhas, visando identificar qual técnica possui maior taxa de acerto. Implementamos as técnicas de extração de características Histograma de cor e Momentos de cromaticidade, aplicadas a um banco de imagens de texturas coloridas de tartarugas previamente classificadas, e através do método k-vizinhos mais próximos para classificar, avaliamos cada técnica quanto a taxa de acertos. Os resultados mostraram que o descritor usando histogramas coloridos apresentou melhores resultados (taxa de acerto?) nas situações X e Y, enquanto o descritor usando momentos de cromaticidade foi melhor na situação W. Portanto, o uso de técnicas de classificação podem auxiliar o trabalho do especialista.

Palavras-chave: Classificação de Tartarugas Marinhas, Extração de Características, Reconhecimento de Padrões.

Abstract

This is the english abstract.

Keywords: sample. key. words.

Lista de ilustrações

Figura 1.1 – Captura incidental de tartarugas marinhas	11
Figura 1.2 – Etapas básicas de um sistema de reconhecimento de padrões	13
Figura 1.3 – Cascos das tartarugas marinhas encontradas no Brasil	15
Figura 2.1 – Imagem representada por matriz de valores.	18
Figura 2.2 – Conceito de 4-vizinhança, vizinhança diagonal e 8-vizinhança	19
Figura 2.3 – Esquematização do espaço de cor CIE-RGB.	20
Figura 2.4 – Diagrama xy de cromaticidade CIE.	20
Figura 2.5 – Exemplo de histograma de uma imagem.	22
Figura 2.6 – Inconsistência em histogramas	23
Figura 2.7 – Exemplo do diagrama e histograma de cromaticidade	25
Figura 2.8 – Exemplo de classificação usando KNN	26
Figura 3.1 – Exemplo de quadros 80x80 <i>pixels</i> extraídos de cada imagem	29
Figura 3.2 – Estrutura básica do OpenCV	30

Lista de tabelas

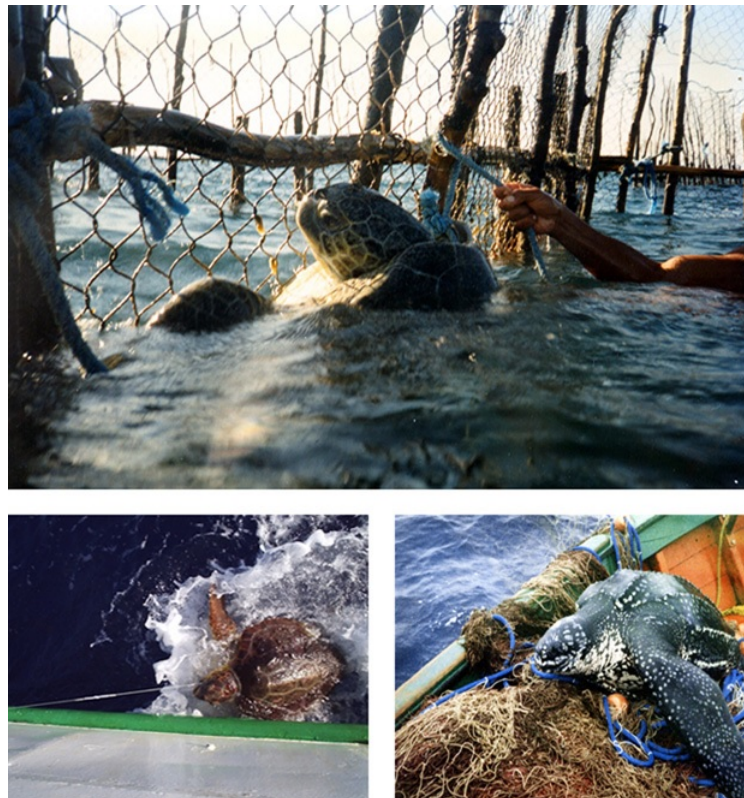
Sumário

1	INTRODUÇÃO	11
1.1	Justificativa	14
1.2	Objetivos	15
1.2.1	Objetivo Geral	15
1.2.2	Objetivos Específicos	16
1.3	Estrutura do Trabalho	16
2	REFERENCIAL TEÓRICO	17
2.1	Introdução a Visão computacional	17
2.2	Imagem Digital e Propriedades	18
2.2.1	Padrões de Cor	19
2.3	Extração de Características	21
2.3.1	Histogramas coloridos	22
2.3.2	Momentos de cromaticidade	24
2.4	Método de Classificação	25
2.4.1	K-vizinhos mais próximos	26
3	DESENVOLVIMENTO	28
3.1	Aquisição das Amostras	28
3.2	OpenCV	29
3.3	Padrões de Cor	31
3.4	Extração de Características	32
3.4.1	Histogramas Coloridos	33
3.4.2	Momentos de Cromaticidade	33
3.5	Classificação	33
3.5.1	Extratégia de Classificação adotada	33
3.5.2	Detalhes do uso dos classificadores nos experimentos	33
3.6	Comparação prévia dos métodos	33
4	RESULTADOS E DISCUSSÕES	34
5	CONSIDERAÇÕES FINAIS	35
	Referências	36

1 Introdução

O homem tem acelerado o processo de extinção de tartarugas marinhas através da exploração de recursos naturais, da destruição de habitats, e principalmente, da caça predatória e do roubo de ovos (ROSTAN; ANDRADE; DAMATO, 1997). Além disso, a mortalidade pela captura incidental de tartarugas marinhas por redes de pesca é ainda o maior problema (SALES et al., 2003). Uma vez capturada a tartaruga pode ser usada como contrabando ou mesmo se devolvida ao mar pode não conseguir sobreviver por causa de ferimentos sofridos pela captura. Como resultado, estas ações têm provocado um decaimento abrupto das espécies. Contudo, dentre sete espécies de tartarugas marinhas existentes no mundo, 5 são encontradas no Brasil e todas as espécies estão listadas como ameaçadas ou vulneráveis pela IUCN (União Mundial para a Conservação da Natureza – The World Conservation Union) (The IUCN Red List of Threatened Species, 2015). Na Figura 1.1 são mostrados alguns exemplos de captura incidental de tartarugas marinhas.

Figura 1.1: Captura incidental de tartarugas marinhas



Fonte: retirado de PROJETO TAMAR-ICMBIO (2011c)

Em adição, o despejo de resíduos antropogênicos em áreas costeiras também tem prejudicado o desenvolvimento de tartarugas marinhas, pois estes animais possuem hábitos alimentares irregulares. Como resultado, pedaços de redes ou descartáveis plásticos

deixados em áreas costeiras são confundidos por alimento, levando as tartarugas a morte por sufocamento ou obstrução do trato digestório (MACEDO *et al.*, 2011). Apesar disso, tartarugas marinhas possuem um ciclo de vida longo e complexo, no qual, a vida adulta e sexual destes animais se inicia após os 25 anos, dificultando o desenvolvimento da espécie (PROJETO TAMAR-ICMBIO, 2011d). Entretanto, graças a projetos de proteção a animais, a extinção de tartarugas marinhas tem sido combatida, através do mapeamento do ciclo de vida destes animais.

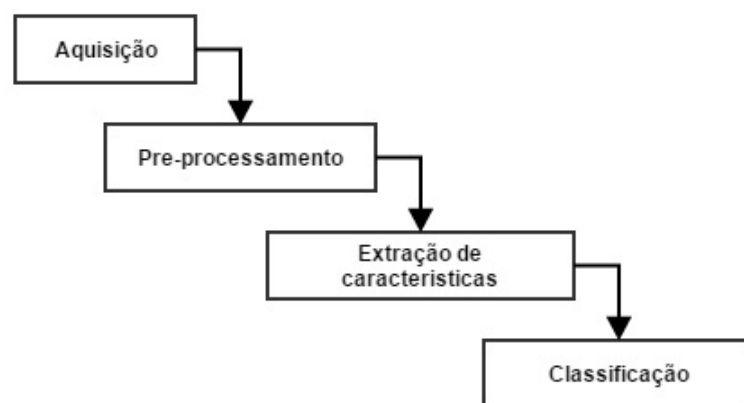
O levantamento sobre informação de tartarugas marinhas no Brasil começou no início dos anos 80, identificando as áreas de reprodução no litoral e confirmando a ocorrência das cinco espécies. Ainda, pesquisadores levantaram que a participação das comunidades costeiras seria a peça fundamental para qualquer plano de conservação das tartarugas marinhas, pois foram eles que ensinaram o que sabiam aos biólogos. Consequentemente, as primeiras bases de pesquisa foram criadas, nas principais áreas de reprodução (MARCOVALDI; SANTOS; SALES, 2011). Por outro lado, sendo o ciclo de vida das tartarugas marinhas muito longo, Chaloupka *et al.* (2008) diz que para identificar a tendência populacional das tartarugas, necessita-se de uma base de dados histórica de no mínimo 20 anos. Como resultado, os sistemas de informação se tornaram uma ferramenta essencial no desenvolvimento das pesquisas de conservação.

Na busca pela conservação da biodiversidade marinha, existem várias ações sendo realizadas pelo projeto Tamar-ICMBio (PROJETO TAMAR-ICMBIO, 2011a) — Programa Nacional de Conservação de Tartarugas Marinhas vinculado à Diretoria de Biodiversidade do Instituto Chico Mendes da Biodiversidade (ICMBio) — tal como descrito no “Plano De Ação Nacional Para Conservação Das Tartarugas Marinhas”, mas vale ressaltar que planos de conservação não visam somente estratégia para diminuir o risco de extinção, mas também produz uma grande malha de informações, tais como, taxonomia, longevidade e tipos de uso (MARCOVALDI; SANTOS; SALES, 2011). Para compreender cada vez melhor o comportamento da tartarugas marinhas, o Tamar-ICMBio iniciou uma coleta de dados sistematizada, através da marcação das fêmeas e parâmetros de incubação dos ninhos, e os estudos de comportamento na fase de desova, por exemplo. O rastreamento por satélite revelou suas rotas e comportamento no mar; estudos genéticos demonstraram as diferenças entre as populações; um novo sistema de informação integrado foi criado e desenvolvido, compondo assim o SITAMAR (Sistema de Informação Sobre Tartarugas Marinhas) (PROJETO TAMAR-ICMBIO, 2011b). Este sistema permite o armazenamento e disponibilização dos dados, a realização de consultas, análises comparativas e emissão de relatórios e mapas. Consequentemente, dando suporte no avanço de pesquisas contribuindo para maior aperfeiçoamento das técnicas de conservação das tartarugas marinhas. Portanto, um sistema de informação das tartarugas marinhas tem sido a melhor ferramenta para monitorar o desenvolvimento destas espécies.

Por outro lado, o SITAMAR é um sistema de acesso limitado e a inserção dos dados é restrita a pesquisadores do projeto Tamar. Neste seguimento, a expansão deste sistema para a população em geral poderia abranger áreas em que o Tamar-ICMBio não possui uma base de pesquisa ou conservação, ou seja, tendo um sistema de fácil acesso, por exemplo, um aplicativo *mobile*, qualquer pessoa poderia colaborar com a captação de informações de atividades que envolvam tartarugas marinhas. Desta forma, espera-se uso intenso do sistema *mobile*, e logo, um grande crescimento da massa de dados do SITAMAR. Consequentemente, com um sistema em grande escala, o trabalho manual de classificação feito por um especialista humano torna-se impraticável. Como resultado, torna-se necessário um sistema automático e para auxiliar o trabalho dos especialistas quanto a classificação dos dados. Assim sendo, os especialistas poderiam fazer a validação dos resultados do sistema automático e as classificações, apenas dos resultados ambíguos do sistema automático.

Em virtude da classificação automática, [Hernández-Serna e Jiménez-Segura \(2014\)](#) e [Spampinato et al. \(2010\)](#) mostraram que é possível utilizar técnicas de processamento digital de imagens (PDI) e reconhecimento de padrões para classificação das espécies de animais que possuem, em sua formação corpórea, um padrão de cor e textura distintos. Neste contexto, um sistema para a classificação automática de espécies de tartarugas marinhas, deve seguir algumas etapas básicas ([Figura 1.2](#)), para que a imagem de uma tartaruga seja classificada através de um sistema de reconhecimento de padrões.

Figura 1.2: Etapas básicas de um sistema de reconhecimento de padrões



Fonte: Elaborada pelo autor.

A primeira etapa é a aquisição, no qual, a imagem física é transformada em uma imagem digital através de um sensor, por exemplo, uma câmera fotográfica. Depois, acontece o pré-processamento, onde processa-se a imagem para assegurar que ela satisfaça algumas condições, tais como redução de ruídos, aumento de contraste e seleção de regiões

de interesse específicas. A próxima etapa é a extração de características, que consiste em extrair características matemáticas das imagens em vários níveis de complexidade. Finalmente a última etapa é a classificação, onde o reconhecimento de padrões é a tarefa que identificará, por exemplo, qual é a espécie da tartaruga, baseada nas características extraídas ([ALBUQUERQUE et al., 2011](#)).

Contudo, este trabalho se concentra em duas etapas: a extração de características e a classificação. Na extração de características, se analisa duas técnicas de reconhecimento de texturas coloridas: histograma de cor e momentos de cromaticidade. Na classificação, é usado o método de classificação K-vizinhos mais proximos (KNN). Levando a uma avaliação comparativa que indicará se alguma abordagem clássica de reconhecimento de padrões possui resultados satisfatórios perante resultados de humanos. O processo de reconhecimento de padrões recebe imagens já segmentadas disponibilizadas por uma base de dados própria e também definida neste trabalho.

1.1 Justificativa

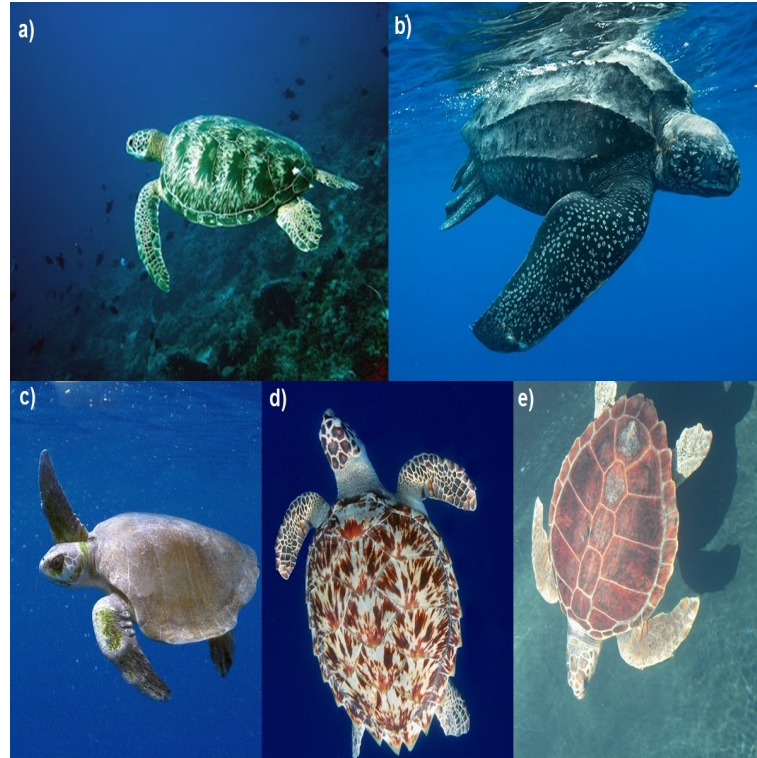
Comunidades pesqueiras ou frequentadores de praias, por exemplo, surfistas e mergulhadores, são as pessoas que possuem maior contato com tartarugas marinhas, mas que não possuem um meio de compartilharem tal informação. Por exemplo, captura incidental, maus tratos, despejo de resíduos em áreas costeiras e principalmente a foto-identificação destas ocorrências ([SITAMAR, s.d](#)), são informações importantes que qualquer pessoa poderia enviar para o sistema dos projetos de conservação. Como resultado, expandir o processo de captura de informações do SITAMAR à qualquer pessoa, levaria a uma maior abrangência das pesquisas de conservação marinha.

Entretanto, permitir que qualquer pessoa envie informações de tartarugas marinhas para o SITAMAR, pode ser um grande empecilho quanto a consistência destes dados, pois as informações devem ser validadas por um especialista antes de ser realmente inserida na base de informações. Especificamente, torna-se necessária a classificação dos dados do sistema de acordo com as espécies, pois cada espécie de tartaruga marinha possui um comportamento migratório e reprodutivo distinto ([MARCOVALDI; SANTOS; SALES, 2011](#)). Por outro lado, mesmo que não haja a expansão do sistema, [Hernández-Serna e Jiménez-Segura \(2014\)](#) dizem que a grande variedade de espécies existentes e o limitado número de taxonomistas ou especialistas em classificação de espécies, apontam grandes desafios no futuro de estudos biológicos e de conservação, ou seja, o volume de dados que serão inseridos pode inviabilizar a validação realizada de forma manual. Sendo, necessário o uso de métodos para validar os dados automaticamente.

Surpreendentemente, [Márquez \(1990\)](#) fez uma definição de cada espécie de tartaruga marinha tornando fácil a classificação de cada animal, como pode ser observado na

Figura 1.3, cada espécie de tartaruga marinha possui em seu casco um padrão de textura colorida.

Figura 1.3: Cascos das tartarugas marinhas encontradas no Brasil



(a)Tartaruga Verde ou Aruanã (*Chelonia mydas*), (b)Tartaruga Gigante ou de Couro (*Dermochelys coriacea*), (c)Tartaruga Oliva ou Pequena (*Lepidochelys olivacea*), (d)Tartaruga de Pente (*Eretmochelys imbricata*), (e)Tartaruga Cabeçuda (*Caretta caretta*). Fonte: adaptado de [BANCO DE IMAGENS TAMAR \(2015\)](#).

Neste contexto, o presente trabalho propõe comparar a acurácia de técnicas de extração de características de texturas coloridas aplicadas a classificação de espécies das Tartarugas Marinhas que são encontradas no litoral brasileiro. Visando uma maneira de agilizar os processos de validação de dados do sistema de informação para conservação de tartarugas marinhas. É importante ressaltar que, além dos benefícios diretos, a classificação automática viabiliza uma maior aquisição de informações a respeito das tendências populacionais e migratórias de cada espécie de tartaruga marinha.

1.2 Objetivos

1.2.1 Objetivo Geral

Comparar a acurácia de técnicas de extração de características de texturas coloridas aplicadas a classificação de espécies das Tartarugas Marinhas que são encontradas no litoral brasileiro.

1.2.2 Objetivos Específicos

- Definir uma base de imagens de tartarugas marinhas segmentadas e classificadas.
- Criar protótipo do sistema de classificação de tartarugas marinhas.
- Comparar as técnicas de extração de características avaliando sua precisão, qualidade e limitações.

1.3 Estrutura do Trabalho

O presente trabalho está dividido em cinco capítulos. Este capítulo 1 traz uma introdução ao assunto abordado, a justificativa para a sua realização, os objetivos pretendidos e a forma como o trabalho foi organizado.

No capítulo 2 são levantados os conceitos para a compreensão deste trabalho. A definição se inicia com conceitos de imagem, destaca brevemente sobre processos de extração de características de texturas coloridas e classificação de imagens, descrevendo os métodos utilizados neste trabalho.

Em seguida, o capítulo 3 traz o desenvolvimento do sistema que fará a extração de características usando dois métodos diferentes e um método para classificação. Também será apresentado o ambiente de desenvolvimento, bem como a estratégia de classificação usada.

No capítulo 4 é apresentada a base de dados que foi criada com as imagens de entrada para o sistema e os resultados obtidos dos testes realizados para cada um dos métodos de extração de características utilizados, bem como a precisão, qualidade e limitações destes métodos para o trabalho proposto.

Por fim, o capítulo 5 apresenta as considerações finais obtidas através da análise dos resultados e os projetos futuros que podem ser desenvolvidos a partir do presente trabalho.

2 Referencial Teórico

Neste capítulo são abordados conceitos fundamentais de processamento de imagem e reconhecimento de padrões de imagens necessários para o entendimento do problema. Deste modo, este capítulo busca: (1) fazer uma introdução de processamento de imagens e visão computacional. (2) Definir imagens e suas propriedades. (3) Apresentar conceitos e técnicas básicas para extrair características de imagens coloridas. (4) Apresentar um método básico de classificação para avaliar a capacidade de diferenciar tartarugas usando vetores de características baseados em cor.

2.1 Introdução a Visão computacional

Processamento digital de imagens (PDI) tem sido aplicado em varias áreas, trazendo benefícios econômicos e sociais. Visão computacional é uma área que aplica diretamente métodos de PDI, expandindo estes método com o uso de inteligência artificial (IA) e técnicas de reconhecimento de padrões. A primeira aplicação de imagem digital foi na industria de jornais, onde foi usado um cabo submarino para transferir imagens digitais entre Londres e Nova York. Introduzido no inicio dos anos 20, o sistema Bartlane, reduziu o tempo de transmissão de imagens de uma semana para menos de 3 horas. No entanto, apesar do termo imagem digital ser diretamente ligado a computadores digitais, estes não estão ligados a criação de imagens digitais, pois atividades usando estas imagens não resultavam em um contexto de PDI. Embora, os avanços de técnicas de PDI estejam diretamente relacionados com o desenvolvimento de computadores digitais a partir década de 60. Assim, neste período os computadores digitais já apresentavam poder de processamento, armazenamento e transmissão, suficiente para executar tarefas de PDI ([GONZALEZ; WOODS, 2002](#)).

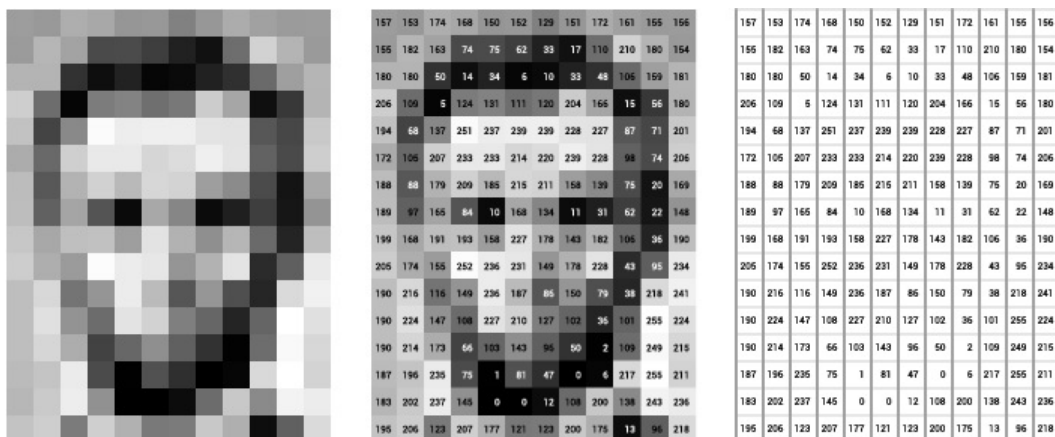
Visão computacional foi a grande área que surgiu com o desenvolvimento da imagem digital e o avanço nas técnicas de PDI. A primeira aplicação de visão computacional surgiu em 1966, em um projeto de verão de um aluno do MIT (Massachusetts Institute of Technology), que conectou uma câmera a um computador, fazendo com que o computador descrevesse o que era mostrado. Embora, seja dito que visão computacional surgiu através dos avanços de processamento digital de imagens, no qual, houve a vontade de se reproduzir a estrutura tridimensional do mundo real através de imagens. Em adição, tendo um melhor entendimento da visão humana, tornou-se possível reproduzir-la através de uma maquina com o uso de inteligência artificial (IA). No entanto, a partir dos anos 90, com o uso de modelos matemáticos, a visão computacional tomou diversas direções, como por exemplo em aplicações gráficas, robotes e segurança.

Atualmente, a visão computacional tem a tendência de ser usada em técnicas de aprendizagem de máquina aplicadas no reconhecimento de objetos sem a supervisão humana. (SZELISKI, 2010). Entretanto, sistemas de visão computacional dependem do objetivo e do domínio no qual serão aplicados, pois cada problema pode apresentar diferentes imagens e cada imagem pode expor diferentes características a serem identificadas. Deste modo, ainda não há um método que se aplique a todos os tipos de problemas que envolvam a análise de imagens (ZÚÑIGA, 2012). Portanto, neste trabalho são abordados conceitos aplicados à classificação automática de espécies de tartarugas marinhas, que talvez não poderiam ser aplicados a outra espécie animal.

2.2 Imagem Digital e Propriedades

Uma imagem digital é uma amostragem espacial e a quantização da imagem de um objeto ou conceito. Esta pode ser definida por uma função discretizada $f(x, y)$, onde x e y são coordenadas espaciais, que representam o local do ponto de amostragem e f , o valor de cada ponto, identificando o brilho ou intensidade luminosa naquele ponto. Em outras palavras, a melhor definição de uma imagem é uma matriz, onde cada elemento, também chamado de *pixel*, representa um ponto na imagem e seu valor representa a cor deste ponto. No entanto, a cor de uma imagem pode ser unidimensional (tons de cinza ou monocromática) ou multidimensional (imagem colorida). Desta forma, em casos de imagens coloridas, a função f é aplicada a cada dimensão de cor (GONZALEZ; WOODS, 2002). A Figura 2.1 mostra uma imagem representada por uma matriz de valores em tons de cinza em que $0 < f(x, y) < 255$.

Figura 2.1: Imagem representada por matriz de valores.

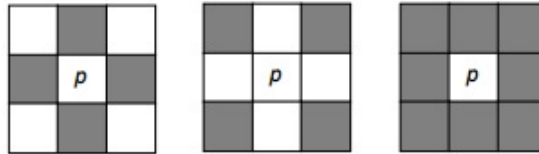


Fonte: retirado de Levin (2015)

Cada *pixel* de uma imagem possui propriedades relevantes quanto aos 8 *pixels* adjacentes, conceito também conhecido como vizinhança de pixel. Esta definição é necessária quanto ao modo em que cada pixel se relaciona com seus adjacentes sendo

conhecido como *4-neighbors* (4-vizinhança), relação apenas com vizinhos horizontais e verticais, e *8-neighbors* (8-vizinhança), relação de vizinhos horizontais, verticais e diagonais (GONZALEZ; WOODS, 2002), como mostrado na Figura 2.2. Portanto, seja um pixel representado por $p(x, y)$, então seu 4 vizinhos horizontais e verticais são definidos por: $p(x + 1, y), p(x - 1, y), p(x, y + 1), p(x, y - 1)$ e seus quatro vizinhos diagonais são: $p(x + 1, y + 1), p(x - 1, y + 1), p(x + 1, y - 1), p(x - 1, y - 1)$.

Figura 2.2: Conceito de 4-vizinhança, vizinhança diagonal e 8-vizinhança



Fonte: retirado de MARQUES FILHO e VIEIRA NETO (1999)

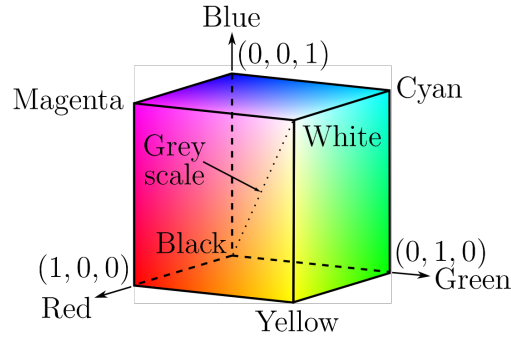
A cor é outro atributo muito importante em uma imagem, pois representa uma característica que melhor diferencia uma imagem de outra (DEOLE; LONGADGE, 2014). A representação de cor em uma imagem surgiu usando as percepções tricromáticas do olho humano, no qual, os receptores do olho respondem a combinação de 3 diferentes espectros de cores, por exemplo, o vermelho(R-*red*), o verde(G-*green*) e o azul(B-*blue*). Possibilitando a representação de somente três cores primárias para gerar muitas outras. Embora, para quantificar uma cor é necessário o uso de representações discretas e padronizadas, também de chamados espaço de cor ou padrões de cor.

Neste trabalho a cor será muito relevante para identificar os padrões de textura de uma imagem, sendo possível identificar propriedades distintivas usando uma representação matemática ou discreta, também conhecido como vetor de característica.

2.2.1 Padrões de Cor

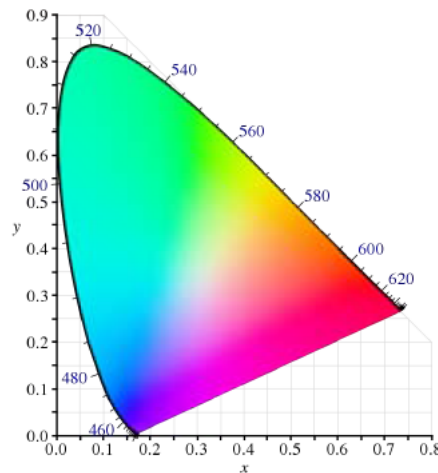
Em 1931, através da CIE (*Commission Internationale d'Eclairage* - Comissão Internacional em Iluminação), foi criado o primeiro padrão de cor, o espaço de cor CIE-RGB (*Red, Green, Blue*), que representa a combinação das cores vermelho, verde e azul. Este padrão, surgiu por meio de experimentos aplicados a comprimentos de onda de luz. O padrão CIE-RGB, é representado pelo sistema de coordenadas cartesianas, definidas por um cubo em que cada cor é definida por um ponto dentro ou em volta deste cubo. Por ser mais simples, CIE-RGB ainda é usado como principal padrão em diferentes dispositivos digitais, por exemplo, câmeras digitais e monitores de vídeo (GONZALEZ; WOODS, 2002). Além disso, cada cor em CIE-RGB é frequentemente representada em modelos computacionais em valores de 24 bits, sendo 8 bits para cada canal vermelho, verde e azul. Neste modelo cada cor primária varia de valores entre 0 e 255, sendo possível identificar 16777216 cores quando combinadas, como mostrado na Figura 2.3.

Figura 2.3: Esquematisação do espaço de cor CIE-RGB.



Fonte: retirado de Cattin (2015)

Entretanto o padrão CIE-RGB, apresenta algumas inconsistências, tais como, incidência de valores negativos em seus espectros para representar cores visíveis. Logo, com a desvantagem deste modelo houve a necessidade de reconstruir um novo espaço de cor, o CIE-XYZ, em que, foi feita a adaptação dos valores do CIE-RGB para que existisse representações positivas, formando assim um espectro de cor mais puro, podendo representar cores visíveis e invisíveis, facilitando o uso mais preciso em métodos matemáticos, pois os eixos X e Z identificam a cromaticidade e o eixo Y a luminescência. Além disso, a CIE definiu um diagrama xy de cromaticidade (Figura 2.4), onde as coordenadas de cromaticidade (x,y) descartam a intensidade absoluta de uma determinada cor, representado somente a cor em sua forma pura (SZELISKI, 2010).

Figura 2.4: Diagrama xy de cromaticidade CIE.

As cores e seus correspondentes valores (x,y) . Fonte: Szeliski (2010)

Neste âmbito, a Equação 2.1 mostra a transformação do modelo CIE-XYZ através

do CIE-RGB (WYSZECKI; STILES, 2000).

$$\begin{aligned} X &= 0,607R + 0,174G + 0,200B, \\ Y &= 0,299R + 0,587G + 0,114B, \\ Z &= 0,066G + 1,111B \end{aligned} \quad (2.1)$$

Contudo, com os valores de X,Y e Z é possível definir as coordenadas de cromaticidade (x,y) no diagrama de cromaticidade, como mostra a Equação 2.2.

$$\begin{aligned} x &= \frac{X}{X + Y + Z}, \\ y &= \frac{Y}{X + Y + Z}, \\ z &= 1 - x - y \end{aligned} \quad (2.2)$$

2.3 Extração de Características

Todas as informações de uma imagem digital é formada por seu conjunto de *pixels*, entretanto uma imagem inteira é muito grande para ser usada diretamente em sistemas de visão computacional, pois possui muito dado irrelevante a descrição de seu conteúdo. Assim, as imagens necessitam passa por um processo conhecido como extração de características, no qual, as imagens são reduzidas usando um conjunto de descritores. Estes descritores usam vetores escalares, também chamados de vetores de características, para agrupar numericamente as características do conteúdo de uma imagem (ZÚÑIGA, 2012).

A extração de característica é o processo mais sensível em sistemas de visão computacional ou de reconhecimento de padrões, pois para cada problema o objeto em uma imagem pode ser representado de acordo com sua cor, forma ou textura. Tornando a escolha do descritor dependente do problema a ser resolvido, ou até mesmo a combinação de vários descritores. Consequentemente, a classificação de imagens usando a textura ainda é uma tarefa difícil para a visão computacional e o reconhecimento de padrões, pois imagens da mesma classe podem apresentar objetos com variações em iluminação, escala e ângulos (KHAN et al., 2013)(KHAN et al., 2014). Tipicamente descritores de textura trabalham em escala cinza ignorando a informação de cor. Esta exclusão de cor é geralmente ocasionada pela grande variação de compressão, sombras e pelo aumento multidimensional dos descritores (KHAN et al., 2012), como resultado, levando a um custo computacional muito alto.

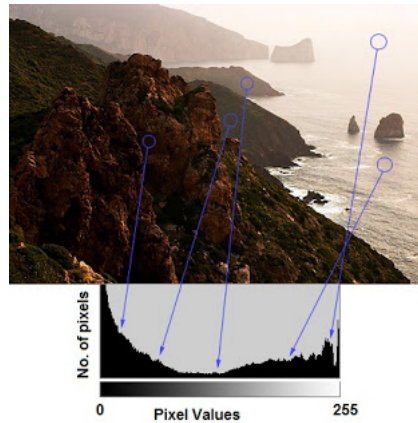
Por outro lado, mesmo que se tenha um custo de processamento alto, há estudos que mostram uma melhora significativa, quanto a performance e acurácia, na representação de imagens realizada por descritores que combinam textura e cor (KHAN et al., 2013)(KHAN et al., 2014)(WANG; ZHANG; YANG, 2014). Neste contexto, este trabalho considera a

combinação de textura e cor, o elemento principal a ser abordado e analisado usando dois descritores diferentes: Histogramas coloridos e Momentos de cromaticidade, no qual serão analisados separadamente para determinar a importância de cada um na classificação de espécies de tartarugas marinhas.

2.3.1 Histogramas coloridos

Histograma são coleções da quantidade de dados organizados em conjuntos de valores chamados células. Por serem simples, este tipo de descritores de características são invariantes a rotação, transição e escalabilidade de objetos em uma imagem (SWAIN; BALLARD, 1991). Sendo cada *pixel* associado a uma cor ou um valor de intensidade de luz, estes podem ser descritos de acordo com a quantidade de *pixels* da mesma cor, denominando histogramas coloridos. Histogramas coloridos são comumente usados como descritores em sistema de recuperação de imagens baseados em conteúdo (SINGHA; HEMACHANDRAN, 2012).

Figura 2.5: Exemplo de histograma de uma imagem.



Histograma representado pelo gráfico sendo o eixo y o numero de *pixels* e o eixo x o valor discretizado do canal de cor. Fonte: retirado de Mordvintsev e Abid (2014)

Segundo Casanova (2008) e Zúñiga (2012), sendo as dimensões de uma imagem M e N e um *pixel* de uma imagem definido por $f(x, y)$, com $x = 0, 1, \dots, N - 1$ e $y = 0, 1, \dots, M - 1$ e possuindo valores discretos de $i = 0, 1, \dots, G - 1$, onde G é o valor possível de intensidade de cada *pixel* em determinado espaço de cor. Assim, a construção de um histograma é dada pela equação:

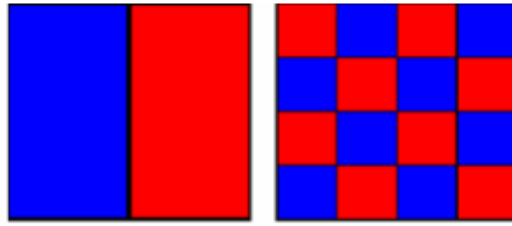
$$h(i) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \delta(f(x, y), i) \quad (2.3)$$

Onde:

$$\delta(i, j) = \begin{cases} 1 & \text{se } j = i \\ 0 & \text{se } j \neq i \end{cases} \quad (2.4)$$

Adotando o espaço de cor RGB estende-se que a função $h(i)$ será aplicada a cada canal, ou seja $h(r, g, b)$, levando a uma abordagem multi-dimensional. Além disso, os valores gerados por $h(i)$ podem ser usados diretamente como vetores de características. Entretanto, por representarem a proporção de cor em uma imagem, há alguns problemas associados ao uso direto de histogramas como vetor de características. Segundo [Sharma, Rawat e Singh \(2011\)](#) duas imagens aparentemente diferentes podem apresentar o mesmo histograma de cor, tornando as vezes a descrição das características das imagens inconsistentes no momento de comparar a similaridade das imagens como mostrado na [Figura 2.6](#).

Figura 2.6: Inconsistência em histogramas



Duas imagens aparentemente diferentes, mas com a mesma proporção de cor. Fonte: retirado de [Sharma, Rawat e Singh \(2011\)](#)

Como consequência, é necessário analisar os casos em que imagens de classes diferentes apresentem o mesmo histograma de cor, para que não se tenha incongruência ao se fazer o reconhecimento da classe na qual esta imagem pertence. Embora, propriedades mais descritivas possam ser extraídas de histogramas usando métodos estatísticos de primeira ordem ([CASANOVA, 2008](#)). Por conseguinte, tendo como base a função de densidade de probabilidade $p(i)$, dada por:

$$p(i) = \frac{h(i)}{NM}, i = 0, 1, \dots, G - 1 \quad (2.5)$$

Logo, dado um histograma de uma imagem, suas características podem ser expressas estatisticamente pelas equações:

Média:

$$\mu = \sum_{i=0}^{G-1} ip(i) \quad (2.6)$$

Variância:

$$\sigma = \sum_{i=0}^{G-1} (i - \mu)^2 p(i) \quad (2.7)$$

Curtose:

$$\mu_4 = \sigma^{-4} \sum_{i=0}^{G-1} (i - \mu)^4 p(i) - 3 \quad (2.8)$$

Energia:

$$E = \sum_{i=0}^{G-1} [p(i)]^2 \quad (2.9)$$

Entropia:

$$H = - \sum_{i=0}^{G-1} p(i) \log_2 [p(i)] \quad (2.10)$$

A Média determina o nível de intensidade da imagem, a Variância identifica a variação ao longo da Média, a Curtose é uma medida do achatamento do histograma, a Energia avalia a uniformidade textural em uma imagem e a Entropia mede a desordem dos dados no histograma.

2.3.2 Momentos de cromaticidade

Métodos que trabalham com cor exigem uma maior carga de processamento por trabalharem com vetores multidimensionais. No entanto, [Paschos \(2000\)](#) determinou uma solução mais simples para o reconhecimento de texturas coloridas, chamada momentos de cromaticidade (do inglês *Chromaticity moments*). Usando o espaço de cor CIE-XYZ (apresentado na seção 2.2.1), o trabalho é baseado no conceito de cromaticidade, onde cada *pixel* reflete a um ponto (x, y) determinado pelo diagrama de cromaticidade, como definido pela [Equação 2.2](#).

Uma imagem pode apresentar varios *pixels* com o mesmo valor de cromaticidade, levando assim a uma distribuição bidimensional, ou seja, um histograma sobre o espaço de cromaticidade. Como resultado, uma imagem pode ser caracterizada por dois atributos: a formação bidimensional das cores(i.e diagrama de cromaticidade) e a distribuição da quantidade pixels com estas cores(i.e histograma). Posteriormente, para que se tenha uma melhor leitura das informações destes dados, é usado o conceito de **momentos**, que são utilizados para descrever a distribuição espacial dos pontos contidos na imagem, também chamados de momentos estatísticos ([ALBUQUERQUE et al., 2011](#)). Assim, uma imagem I com dimensões L_x, L_y defini-se a partir do *diagrama de cromaticidade* como sendo:

$$T(x, y) = \begin{cases} 1 & \text{se } \exists(i, j) : I(i, j) \text{ produzindo } (x, y), \\ 0 & \text{caso contrário} \end{cases} \quad (2.11)$$

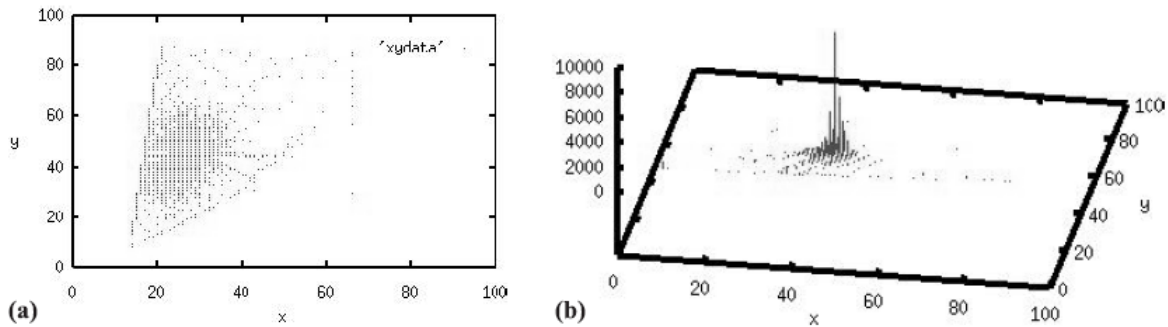
$$0 \leq i \leq L_x - 1, 0 \leq j \leq L_y - 1$$

E associado com o *diagrama de cromaticidade* há uma distribuição bidimensional (i.e histograma) definida por:

$$D(x, y) = k, \text{ sendo } k = \text{número de pixels produzindo } (x, y) \quad (2.12)$$

Deste modo, D representa uma projeção bidimensional de T , como mostrado na Figura 2.7.

Figura 2.7: Exemplo do diagrama e histograma de cromaticidade



(a) imagem representada pelo diagrama de cromaticidade (T), (b) distribuição bidimensional da quantidade de pixels representados no diagrama de cromaticidade (D). Fonte: retirado de (PASCHOS, 2000)

Consequentemente, as funções $T(x, y)$ e $D(x, y)$ podem ser aproximadamente caracterizadas por um conjunto de momentos (m, l) , definidos por:

$$M_T(x, y) = \sum_{x=0}^{X_S-1} \sum_{y=0}^{Y_S-1} x^m y^l T(x, y) \quad (2.13)$$

$$M_D(x, y) = \sum_{x=0}^{X_S-1} \sum_{y=0}^{Y_S-1} x^m y^l D(x, y) \quad (2.14)$$

Onde, $m, l = 0, 1, 2, \dots$ e X_S, Y_S são dimensões discretas do espaço x - y . Isto é, verificando que $x, y \in [0, 1]$, temos que ao reescalar e discretizar, i.e, $x = [x \times 100]$ e $y = [y \times 100]$, assim $X_S = Y_S = 100$. Entretanto, vale destacar que em questões de processamento, este método apresenta complexidade de ordem $O(n)$, ou seja, para calcular um conjunto de momentos o crescimento computacional é linear.

2.4 Método de Classificação

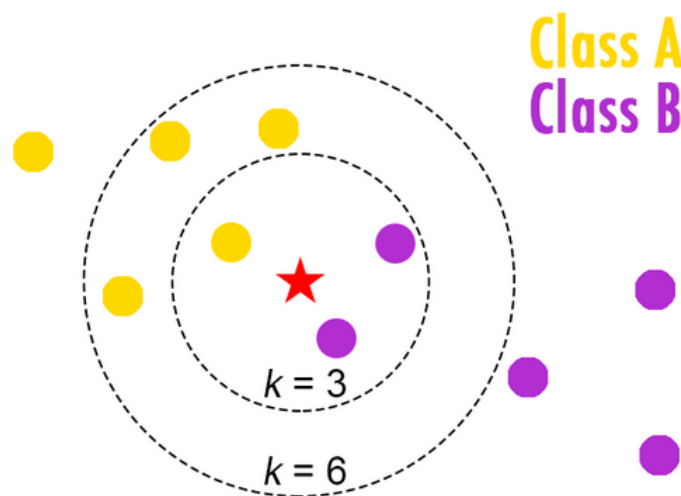
Em um sistema de reconhecimento de padrões, para que a similaridade entre classes seja definida é necessário o uso de sistemas que possam reconhecer a característica de um objeto e vinculá-lo a uma classe. Sistemas estes também conhecidos como classificadores.

2.4.1 K-vizinhos mais próximos

K-vizinhos mais próximos (KNN, do inglês *K-nearest neighbor*) é uma técnica de classificação simples fundamentada na predição de classes a partir de um sistema de votação entre os objetos mais próximos. Esta técnica foi formulada por [Cover e Hart \(1967\)](#), sendo uma técnica não parametrizada ou não linear, pois não assume uma forma funcional.

KNN parte do princípio de distância entre dois objetos, onde a distância euclidiana é frequentemente usada, podendo outra distância ser usada dependendo do domínio do problema. Assim, dado um objeto de classe desconhecida P a ser testado, e um conjunto de objetos treinados de classes conhecidas Q , é calculada a distância de cada objeto de Q em relação a P . Em seguida, os k objetos mais próximos (vizinhos) são reiterados há um sistema de votação onde a classe em maior número é a ganhadora, definindo assim a classe do objeto P , sendo $1 \leq k \leq N$ e $N = \text{número de objetos em } Q$. Um exemplo é ilustrado na [Figura 2.8](#).

Figura 2.8: Exemplo de classificação usando KNN



Após o cálculo das distâncias, o objeto de teste (estrela) deve ser classificado sendo da classe **A** (objetos amarelos) ou **B** (objetos roxos). Assim, sendo $k=3$, o objeto de teste é associado a classe B, pois há 2 objetos da classe B e 1 objeto A. Por outro lado, sendo $k=6$, será associada a classe A (4 objetos de A vs. 2 objetos de B). Fonte: adaptado de [DeWilde \(2012\)](#)

Segundo [Zúñiga \(2012\)](#) a seleção do valor de k e o método de busca dos objetos, podem ser dois fatores a serem tomados com cautela. Assim, sendo o valor de k muito pequeno ou muito grande, o classificador pode ficar muito sensível a ruídos. Além do mais, sendo o método de busca inadequado, este pode afetar no tempo de classificação, por realizar as comparações entre objetos a serem classificados e o conjunto de objetos já treinados.

Por outro lado, a qualidade da classificação é outro fator a ser empregado. Tendo um conjunto de objetos com classes corretamente definidas R , a efetividade do KNN pode ser calculada com a divisão de R em dois subconjuntos: R_D - objetos de teste e R_T - objetos treinados. Usando estes dois conjuntos como dados de entrada para o classificador KNN, podemos ter a taxa de efetividade do classificador através da [Equação 2.15](#).

$$Efetividade = 100 * \frac{N_{\text{acertos}}}{N_{\text{testes}}} \quad (2.15)$$

Onde N_{acertos} é o número de objetos do conjunto R_T classificados corretamente e N_{testes} é o número de objetos contidos em R_D .

3 Desenvolvimento

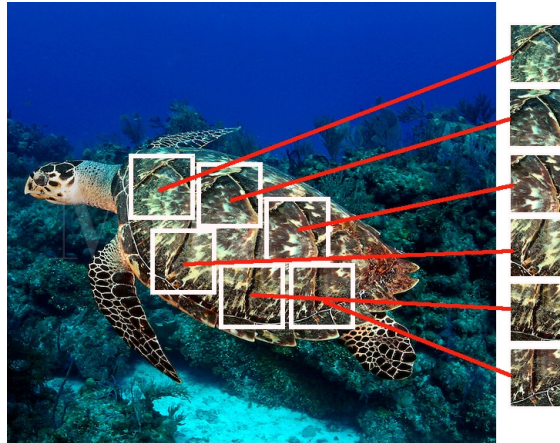
O sistema proposto neste trabalho tem como objetivo, a partir de quadros de imagens de segmentadas de cascos de tartarugas, classificar ou pelo menos encontrar uma similaridade entre texturas. Para tanto são utilizadas duas técnicas para criação de vetores de características, Histograma de Cor e Momentos de Cromaticidade, a fim de comparar seus resultados. Em adição, é usado uma técnica simples para a classificação supervisionada de imagens, K-vizinhos mais próximos, também conhecido do inglês como KNN (*K nearest neighbors*), a fim de testar os vetores de características.

O primeiro método de vetores de características foi o método Histograma de Cor. Esse método faz uma representação uniforme das cores existentes em cada imagem seguindo como critérios, o numero de *pixels* com mesma cor ou tonalidade. O segundo método utilizado Momentos de cromaticidade, os *pixels* também são validados de acordo com sua cor, mas diferentemente do método de Histograma de cor, esse método utiliza o conceito de momentos no avalia a proporção que as cores estão distribuídas na imagem.

Esses métodos devem gerar vetores de características como saída, no qual estes vetores são usados como entrada para o classificador KNN, ficando este a cargo de classificar uma imagem de acordo com sua classe, sendo a classe a espécie de tartarugas, para este trabalho . Neste capítulo descreve-se os detalhes de implementação dos sistemas desenvolvidos. O desenvolvimento ocorreu na linguagem Python, utilizando o *framework* OpenCV.

3.1 Aquisição das Amostras

Para realizar a aquisição de amostras foi feito uma varredura em repositórios de imagens online, no qual foram recuperadas 25 imagens sendo 5 imagens para cada espécie de tartaruga, tendo estas imagens uma média de dimensão de 800x600 *pixels*. Entretanto, para que tivesse uma análise mais aprofundada nas texturas dos cascos, foi feita a extração de seis quadros de 80x80 *pixels*, como mostra a [Figura 3.1](#). No entanto, não foi realizado nenhum tratamento de brilho ou contraste, para que a imagem seja o mais próximo da realidade de um sistema de armazenamento de imagens, no qual recebe qualquer imagens com ou sem edição.

Figura 3.1: Exemplo de quadros 80x80 *pixels* extraídos de cada imagem

Fonte: imagem retirada de [Mergulhadores \(2014\)](#) e editada pelo autor

3.2 OpenCV

Neste trabalho foi utilizada a versão 2.4.11 do *framework* OpenCV. Todas as versões podem ser obtidas através da instalação do pacote completo disponível em seu site ([<http://opencv.org/downloads.html>](http://opencv.org/downloads.html)). OpenCV é uma biblioteca de visão computacional sob a licença *opensource*, criada e desenvolvida pela Intel em C e C++. Sendo multi-plataforma pode ser utilizada em Windows, Linux e Mac OS X. Ela foi desenvolvida para aumentar a capacidade computacional com foco em aplicações de tempo-real e seu principal papel é prover uma infraestrutura para criar aplicações de forma simples e rápida. Com mais de 500 funções OpenCV que se divide em módulo, que se expandem em várias áreas em visão computacional, principalmente em funções para serem usadas em problemas que envolvam aprendizado de máquina ([BRADSKI, 2008](#)).

Entretanto, neste trabalho foi usada um versão mais simplificada do OpenCV na linguagem Python (OpenCV-Python) ¹, já que o foco é criar uma aplicação que entenda as texturas. Embora, estando na linguagem Python a legibilidade do código seja muito maior comparada com a biblioteca original em C ou C++. Em adição ao OpenCV foram também usadas outras bibliotecas como Numpy ² e Matplotlib³. O Numpy é uma biblioteca que otimiza as operações com números potencializando a leitura e armazenamento dos vetores de características e além de facilitar o uso das bibliotecas do Matplotlib. O Matplotlib é uma biblioteca para geração de gráficos a partir de dados armazenados em listas ou *arrays*, no qual foi usada para se ter uma noção visual dos resultados gerados.

Assim como mostrados na [Figura 3.2](#), os principais módulos do *framework* OpenCV são ([BRADSKI, 2008](#)):

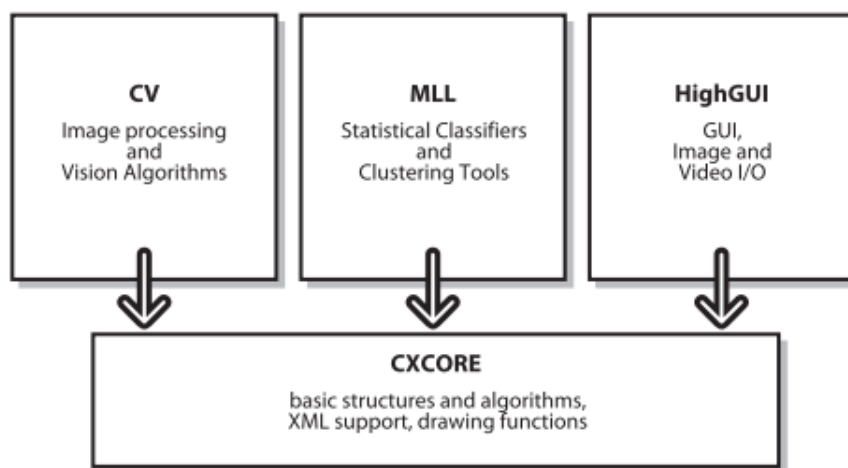
¹ [<http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_tutorials.html>](http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_tutorials.html)

² [<http://www.numpy.org/>](http://www.numpy.org/)

³ [<http://matplotlib.org/>](http://matplotlib.org/)

- CXCORE - contém estrutura e tratamento de dados básico, como matrizes, objetos de persistência, gerenciamento de memória, ferramentas de desenho, texto e matemática básica.
- CV - contém ferramentas para processamento de imagens, análise de estruturas, *motion* e *tracking*, reconhecimento de padrões e calibração de câmeras.
- Machine Learning (ML) - contém ferramentas para clusterização, classificação e funções para análise de dados
- HighGUI - contém ferramentas para interface com o usuário

Figura 3.2: Estrutura básica do OpenCV



Fonte: imagem retirada de [Bradski \(2008\)](#)

No entanto, na versão OpenCV-Python, esta divisão de módulos é menos transparente, para simplificar a criação de aplicações. Logo, para que se tenha uma melhor orientação, no desenvolvimento deste trabalho, utilizou-se somente os módulos **CXCORE** e **HighGUI** por serem os módulos básicos e necessários para o desenvolvimento dos métodos propostos. OpenCV-Python prover as funções `imread()`(leitura) e `imwrite()`(escrita) que suportam vários formatos de imagens não vetoriais, como por exemplo, BMP, PNG, JPEG, TIFF, etc. Por padrão a função `imread()` retorna uma imagem no padrão de cor BGR(azul-verde-vermelho/*blue-green-red*). Entretanto, BGR representa o espaço de cor RGB(vermelho-verde-azul/*red-green-blue*), mas com os *bytes* em ordem inversa. Esta informação de como é o retorno da função de leitura de imagens é muito importante, pois neste trabalho a conversão de BGR para RGB é muito usada e deve ser entendida. Em exemplo o [Código Fonte 3.1](#), mostra os passos iniciais para trabalhar com o *framework* OpenCV-Python.

```

1  # Importando as principais bibliotecas a serem usadas
2  import cv2 # biblioteca OpenCV-Python
3  import numpy as np #biblioteca do Numpy
4  from matplotlib import pyplot as plt #biblioteca do Matplotlib para desenho
   grafico
5
6  def ler_imagem(filename):
7      """Retorna imagem RGB usando a biblioteca opencv"""
8      # Ler uma imagem usando OpenCV
9      imagem = cv2.imread(filename)#imagem recuperada no padrao BGR
10     return imagem

```

Na linha 9 é recuperada uma imagem no padrão BGR e esta imagem é representada por uma matriz de *pixels* ou valores, no qual cada elemento desta matriz contem todos os canais do espaço de cor. O [Código Fonte 3.2](#) mostra como caminhar em uma imagem(matriz) e recuperar o valor de cada canal em cada pixel.

Código Fonte 3.2: Recuperando valores em *pixels*

```

1  # Ler uma imagem usando OpenCV
2  imagem = cv2.imread(filename)#imagem recuperada no padrao BGR
3  #recuperando infomacoes da imagem
4  numeroLinhas , numeroColunas , numeroCanais=imagem.shape
5  for i in range(numeroLinhas):
6      for j in range(numeroColunas):
7          canalB , canalG , canalR = imagem[i , j] # recuperando o
              valor de cada canal em cada pixel

```

3.3 Padrões de Cor

Neste trabalho são usados dois espaços de cor, RGB(vermelho, verde, azul) e XYZ(cromaticidade e luminescência). Mesmo sendo dois padrões que podem representar varias cores a partir de 3 vetores, estes são diferentes entre si. RGB representando três diferentes cores primarias que juntas formam outras cores, mas que não demonstra de forma clara a intensidade da cor. Por, outro lado, XYZ representando cores em sua forma pura, que facilita o uso mais preciso em método matemáticos.

Entretanto, mesmo o RGB podendo ser convertido para XYZ usando formulas matemáticas, foi usada a conversão nativa e otimizada que é oferecida pelo OpenCV-Python, onde uma imagem no padrão BGR pode ser convertida para vários padrões de cor incluindo o padrão RGB e XYZ, como mostrado no [Código Fonte 3.3](#).

Código Fonte 3.3: Convertendo para os espaços de cor RGB e XYZ

```

1  # Ler uma imagem usando OpenCV (leitura natina no padrao BGR)
2  imagemBGR = cv2.imread(filename)

```

```
3
4     # Converter a imagem do padrao BGR para RGB
5     imagemRGB= cv2.cvtColor(imagemBGR, cv2.COLOR_BGR2RGB)
6
7     # Converter a imagem do padrao BGR para XYZ
8     imagemXYZ= cv2.cvtColor(imagem2, cv2.COLOR_BGR2XYZ)
```

3.4 Extração de Características

O processo de extração de características se inicia pela leitura e conversão da imagem como citados no [Código Fonte 3.1](#) e [Código Fonte 3.3](#). Entretanto, para que seja gerado um vetor de característica com os métodos propostos é necessário gerar um histograma com os *pixels* ou dados extraídos de cada imagem. Por este motivo é necessário ter entendimento sobre as entradas e saídas do método de geração de histograma oferecido pelo OpenCV-Python.

Código Fonte 3.4: Criando um histograma

```
1
2     numBins=256 #numero de subdivisooes do histograma
3     rangeValores=[0, 256] #valores em que variam o vetor
4     # Criando um histograma
5     imagem=ler_imagemRGB(filename)
6     #Gerar histograma do canal R
7     hist_r=cv2.calcHist([imagem], [0], None,[ numBins], rangeValores)
8     #Gerar histograma do canal G
9     hist_g=cv2.calcHist([imagem], [1], None,[ numBins], rangeValores)
10    #Gerar histograma do canal B
11    hist_b=cv2.calcHist([imagem], [2], None,[ numBins], rangeValores)
```

Como já explicado, um histograma é um vetor em que cada elemento indica o número de incidências em uma determinada faixa de valores ou *bins*, em um dado vetor ou matriz de valores. Logo o OpenCv-Python oferece a função **calcHist(...)** para gerar os histogramas a partir de vetores ou matrizes de valores. Como mostrado no [Código Fonte 3.4](#) na linha 5, é gerado um histograma a partir do canal R(vermelho) ou "[0]"da imagem. Assim, não sendo a função **calcHist(...)** restrita somente ao formato da imagem, mas a qualquer vetor ou *array* gerado pela biblioteca *Numpy*, sendo esta funcionalidade muito importante, pois os histogramas gerados neste trabalho não ficam limitados a geração a partir de uma matriz de uma imagem.

3.4.1 Histogramas Coloridos

3.4.2 Momentos de Cromaticidade

3.5 Classificação

3.5.1 Estratégia de Classificação adotada

3.5.2 Detalhes do uso dos classificadores nos experimentos

3.6 Comparação prévia dos métodos

CRONOGRAMA PARA PD2(SEÇÃO PARA FINS DA DISCIPLINA DE PD1)

Em PD2, pretende-se escrever o capítulo de desenvolvimento seguindo as seções citadas acima. O desenvolvimento do código ainda está em fase de codificação, pois mesmo que seja implementada todos os algoritmos, ainda é necessário implementar uma forma que teste a acurácia destes algoritmos. Logo, para PD2 está planejado:

1. Montar o banco de imagens - Pretende-se ir ao posto de conservação marinha do projeto tamar em Regência, para capturar fotos de tartarugas marinhas a serem segmentadas e classificadas.
2. Finalizar a codificação do método de classificação para histogramas coloridos
3. Definir o algoritmo de teste usando Validação cruzada (conceito a ser adicionado no capítulo de revisão (Cap. 2)
4. Analisar os Resultados com a base de dados e os algoritmos implementados e escrever o capítulo de resultados e discussões (Cap. 4)
5. Escrever considerações finais

O codificação desenvolvida para este trabalho pode ser encontrada em <https://github.com/wdnei/identificarTartarugas>

4 Resultados e Discussões

5 Considerações Finais

Referências

- ALBUQUERQUE, M. P. et al. *Aplicação da técnica de momentos invariantes no reconhecimento de padrões em imagens digitais*. Centro Brasileiro de Pesquisas Físicas. Rio de Janeiro, 2011. Citado 2 vezes nas páginas 14 e 24.
- BANCO DE IMAGENS TAMAR. *Banco de Imagens - Projeto Tamar*. 2015. Disponível em: <<http://imagens.tamar.org.br/>>. Acesso em: 14 maio 2015. Citado na página 15.
- BRADSKI, A. K. G. *Learning OpenCV*. [S.l.]: O'Reilly, 2008. ISBN 0596516134, 978-0-596-51613-0. Citado 2 vezes nas páginas 29 e 30.
- CASANOVA, D. *Identificação de espécies vegetais por meio da análise de textura foliar*. Dissertação (Mestrado - Programa de Pós-graduação em Ciência da Computação e Matemática Computacional) — Universidade de São Paulo, São Carlos, 2008. Citado 2 vezes nas páginas 22 e 23.
- CATTIN, P. *Digital Image Fundamentals: Image processing and computer vision*. 2015. Slides. Disponível em: <<https://miac.unibas.ch/SIP/02-Fundamentals.html>>. Acesso em: 14 maio 2015. Citado na página 20.
- CHALOUPKA, M. et al. Encouraging outlook for recovery of a once severely exploited marine megaherbivore. *Global Ecology and Biogeography*, v. 17, p. 297–304, 2008. Citado na página 12.
- COVER, T. M.; HART, P. E. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, IEEE, v. 13, n. 1, p. 21–27, 1967. Citado na página 26.
- DEOLE, M. P. A.; LONGADGE, R. Content based image retrieval using color feature extraction with knn classification. *IJCSMC*, v. 3, n. 5, p. 1274—1280, maio 2014. Citado na página 19.
- DEWILDE, B. *Classification of Hand-written Digits*. 2012. Disponível em: <<http://bdewilde.github.io/blog/blogger/2012/10/26/classification-of-hand-written-digits-3/>>. Acesso em: 19 julho 2015. Citado na página 26.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 2. ed. Upper Saddle River, NJ: Prentice-Hall, 2002. ISBN 0-201-18075-8. Citado 3 vezes nas páginas 17, 18 e 19.
- HERNÁNDEZ-SERNA, A.; JIMÉNEZ-SEGURA, L. F. Automatic identification of species with neural networks. *PeerJ*, PeerJ Inc., v. 2, p. e563, 2014. Citado 2 vezes nas páginas 13 e 14.
- KHAN, F. S. et al. Color attributes for object detection. In: IEEE. *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. [S.l.], 2012. p. 3306–3313. Citado na página 21.
- KHAN, F. S. et al. Compact color–texture description for texture classification. *Pattern Recognition Letters*, Elsevier, v. 51, p. 16–22, 2014. Citado na página 21.

- KHAN, F. S. et al. Evaluating the impact of color on texture recognition. In: SPRINGER. *Computer Analysis of Images and Patterns*. [S.l.], 2013. p. 154–162. Citado na página 21.
- LEVIN, G. *Image Processing and Computer Vision*. Openframeworks, 2015. Disponível em: <http://openframeworks.cc/ofBook/chapters/image_processing_computer_vision.html>. Acesso em: 14 maio 2015. Citado na página 18.
- MACEDO, G. R. et al. Ingestão de resíduos antropogênicos por tartarugas marinhas no litoral norte do estado da bahia, brasil. *Ciência Rural*, SciELO Brasil, v. 41, n. 11, p. 1938–1943, 2011. Citado na página 12.
- MARCOVALDI, M. Â.; SANTOS, A.; SALES, G. Plano de ação nacional para conservação das tartarugas marinhas. *Brasília: ICMBio*, 2011. Citado 2 vezes nas páginas 12 e 14.
- MARQUES FILHO, O.; VIEIRA NETO, H. *Processamento digital de imagens*. Rio de Janeiro: Brasport, 1999. ISBN 8574520098. Disponível em: <<http://www.ppgia.pucpr.br/~facon/ComputerVisionBooks/1999LivroProcessamentoImagensDigital.pdf>>. Acesso em: 14 maio 2015. Citado na página 19.
- MÁRQUEZ, R. Fao species catalogue. sea turtles of the world: An annotated and illustrated catalogue of sea turtle species known to date. *FAO fisheries Synopsis*, Rome, v. 11, n. 125, p. 81, 7 1990. Citado na página 14.
- Mergulhadores. *Tartaruga de Pente*. 2014. Disponível em: <http://www.mergulhadores.com.br/portal/especies.html/__/tartaruga-de-pente-r4>. Acesso em: 01 novembro 2015. Citado na página 29.
- MORDVINTSEV, A.; ABID, K. . *OpenCV-Python Tutorials: Histograms - 1 : Find, plot, analyze !!!* 2014. Disponível em: <http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_imgproc/py_histograms/py_histogram_begins/py_histogram_begins.html>. Acesso em: 14 maio 2015. Citado na página 22.
- PASCHOS, G. Fast color texture recognition using chromaticity moments. *Pattern Recogn. Lett.*, Elsevier Science Inc., New York, NY, USA, v. 21, n. 8, p. 837–841, jul. 2000. ISSN 0167-8655. Disponível em: <<http://dl.acm.org/citation.cfm?id=351795.351806>>. Citado 2 vezes nas páginas 24 e 25.
- PROJETO TAMAR-ICMBIO. *Programa Nacional de Conservação de Tartarugas Marinhas*. 2011a. Disponível em: <<http://www.tamar.org.br/>>. Acesso em: 14 maio 2015. Citado na página 12.
- PROJETO TAMAR-ICMBIO. *Informações gerais Projeto Tamar/ICMBio*. 2011b. Disponível em: <<http://tamar.org.br/releases/release-projeto-tamar-infos-gerais.pdf>>. Acesso em: 14 maio 2015. Citado na página 12.
- PROJETO TAMAR-ICMBIO. *Interação com a pesca*. 2011c. Disponível em: <<http://www.tamar.org.br/interna.php?cod=73>>. Acesso em: 14 maio 2015. Citado na página 11.
- PROJETO TAMAR-ICMBIO. *Resultados*. 2011d. Disponível em: <<http://www.tamar.org.br/interna.php?cod=399>>. Acesso em: 14 maio 2015. Citado na página 12.

ROSTAN, G.; ANDRADE, I. M. F.; DAMATO, A. F. Atividades de educação ambiental desenvolvidas pelo projeto tamar-ibama no litoral norte da bahia. In: CONGRESSO NORDESTINO DE ECOLOGIA, 7., Ilheus. *Anais...* Bahia: EDITUS, 1997. p. 299. Citado na página 11.

SALES, G. et al. Captura incidental de tartarugas marinhas pela frota de rede de emalhe de deriva sediada em ubatuba, são paulo-brasil. In: II JORNADAS DE CONSERVACIÓN Y USO SUSTENTABLE DE LA FAUNA MARINA, Montevideo, Uruguay. *Resumo...* São Paulo, 2003. v. 1, p. 1–3. Citado na página 11.

SHARMA, N. S.; RAWAT, P. S.; SINGH, J. S. Efficient cbir using color histogram processing. *Signal & Image Processing*, Academy & Industry Research Collaboration Center(AIRCC), v. 2, n. 1, 2011. Citado na página 23.

SINGHA, M.; HEMACHANDRAN, K. Content based image retrieval using color and texture. *Signal & Image Processing : An International Journal (SIPIJ)*, Academy & Industry Research Collaboration Center (AIRCC), v. 3, n. 1, p. 39–57, 2012. Citado na página 22.

SITAMAR. *Sistema de Informação Sobre Tartarugas Marinhas*. s.d. Disponível em: <http://www.mma.gov.br/estruturas/sbf_chm_rbbio/_arquivos/tamar_icmbio_72.pdf>. Acesso em: 14 maio 2015. Citado na página 14.

SPAMPINATO, C. et al. Automatic fish classification for underwater species behavior understanding. In: ACM. *Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams*. NY,USA: ACM, 2010. p. 45–50. Citado na página 13.

SWAIN, M. J.; BALLARD, D. H. Color indexing. *International journal of computer vision*, Springer, v. 7, n. 1, p. 11–32, 1991. Citado na página 22.

SZELISKI, R. *Computer vision: algorithms and applications*. [S.l.]: Springer Science & Business Media, 2010. Citado 2 vezes nas páginas 18 e 20.

The IUCN Red List of Threatened Species. *The IUCN Red List of Threatened Species*. 2015. Disponível em: <<http://www.iucnredlist.org/>>. Acesso em: 14 maio 2015. Citado na página 11.

WANG, X.-Y.; ZHANG, B.-B.; YANG, H.-Y. Content-based image retrieval by integrating color and texture features. *Multimedia tools and applications*, Springer, v. 68, n. 3, p. 545–569, 2014. Citado na página 21.

WYSZECKI, G.; STILES, W. Color science: Concepts and methods, quantitative data and formulae. Wiley, New York, 2000. Citado na página 21.

ZÚÑIGA, A. M. G. *Sistema de visão artificial para identificação do estado nutricional de plantas*. Dissertação (Mestrado - Programa de Pós-graduação em Ciência da Computação e Matemática Computacional) — Universidade de São Paulo, São Carlos, 2012. Citado 4 vezes nas páginas 18, 21, 22 e 26.