



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

GEOVANNY MAGALHÃES DE OLIVEIRA FILHO

**CLASSIFICAÇÃO DO NÚMERO DE PESSOAS NAS MOTOS EM
IMAGENS DE TRÂNSITO CORRETAMENTE SEGMENTADAS**

**QUIXADÁ
2013**

GEOVANNY MAGALHÃES DE OLIVEIRA FILHO

**CLASSIFICAÇÃO DO NÚMERO DE PESSOAS NAS MOTOS EM
IMAGENS DE TRÂNSITO CORRETAMENTE SEGMENTADAS**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso de Graduação em Sistemas de Informação da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Bacharel.

Área de concentração: computação

Orientador Prof. Dr. Críston Pereira de Souza

**QUIXADÁ
2013**

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca do Campus de Quixadá

O48c Oliveira Filho, Geovanny Magalhães de
Classificação do número de pessoas nas motos em imagens de trânsito corretamente
segmentadas / Geovanny Magalhães de Oliveira Filho. – 2013.
53 f. : il. color., enc. ; 30 cm.

Monografia (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de
Sistemas de Informação, Quixadá, 2013.
Orientação: Prof. Dr. Críston Souza Pereira

Título. 1. Redes neurais. 2 Visão computacional. 3. Imagem – visão por computador I. Título. 3. I.

CDD 006.37

GEOVANNY MAGALHÃES DE OLIVEIRA FILHO

**CLASSIFICAÇÃO DO NÚMERO DE PESSOAS NAS MOTOS EM
IMAGENS DE TRÂNSITO CORRETAMENTE SEGMENTADAS**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso de Graduação em Sistemas de Informação da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Bacharel.

Área de concentração: computação

Aprovado em: _____ / fevereiro / 2013.

BANCA EXAMINADORA

Prof. DSc. Críston Pereira de Souza
Universidade Federal do Ceará-UFC

Prof. DSc. Davi Romero de Vasconcelos
Universidade Federal do Ceará-UFC

Prof. MSc. Ricardo Reis Pereira
Universidade Federal do Ceará-UFC

DEDICATÓRIA

Aos meus pais, a minha avó, a minha noiva
e aos meus amigos ...

AGRADECIMENTOS

Queria agradecer primeiramente a minha avó Vanda, que foi mais do que uma avó para mim durante a minha vida, ela sempre me ajudou em praticamente tudo.

Aos meus pais Geovanni e Arlene por terem me dado toda educação necessária para ter ingressado na Universidade Federal do Ceará. E por sempre terem acreditado em mim.

Aos meus tios Cristina e Chico que me sempre me ajudaram, mesmo antes de entrar na faculdade.

A minha noiva Katarina Holanda por todo amor e apoio dado mesmo com a minha ausência durante a semana.

A minha filha Yasmine por todos os momentos bons que me proporcionou com a notícia de sua chegada.

Ao Prof. Dr. Críston Souza Pereira, meu orientador, pela amizade. Pela receptividade e paciência dada ao longo do desenvolvimento desse trabalho e por todo apoio, incentivo e a ótima formação dado ao longo da minha vida acadêmica.

Ao Prof. MSc. Wladimir Araújo Tavares pela ótima formação que foi dada durante as reuniões do Grupo de Estudos para Maratona de Programação. Pela amizade, apoio e incentivo.

Ao meu grande amigo Marcelo Arraes, pelos bons momentos de companheirismo, pelas ótimas conversas e discussões e por toda ajuda que foi dada durante esses quatro anos.

A todos os professores que de alguma forma contribuíram para a minha formação acadêmica.

A todos os meus amigos da faculdade pela amizade e boa convivência.

"Procure ser uma pessoa de valor, em vez
de procurar ser uma pessoa de sucesso.
O sucesso é consequência."
(Albert Einstein)

RESUMO

Uma infração de trânsito comum é o excesso de passageiros em motos. Para permitir a detecção automática deste tipo de infração, é importante realizar uma contagem de passageiros nas motos com baixa taxa de erro. Uma maneira de realizar esta contagem de passageiros é aplicar um algoritmo de classificação utilizando como entrada as características da região da imagem contendo a moto e os passageiros (região segmentada). Neste trabalho verificamos se esta é uma boa abordagem, dado que a imagem tenha sido corretamente segmentada. Concluímos que é possível obter área ROC acima de 95% empregando redes neurais em um dos modelos propostos neste trabalho para representação da região segmentada.

Palavras chave: Visão Computacional. Mobilidade Urbana. Classificação. Redes Neurais. Árvores de Decisão.

ABSTRACT

A common traffic violation is the excess passengers on motorcycles. In order to detect this kind of violation, it is important to conduct a low error rate counting of passengers on motorcycles. One way to accomplish this is to count passengers using a classifier, having as input the characteristics of the image region containing the motorcycle and passengers (segmented region). In this work we verify if this is a good approach, given that the image is correctly segmented. We conclude that it is possible to obtain ROC area above 95% by employing a neural network and one of the models proposed in this work to represent the segmented region.

Keywords: Computer Vision. Urban Mobility. Classification. Neural Networks. Decision Trees.

LISTA DE ILUSTRAÇÕES

| | |
|--|----|
| Figura 1 - Etapas de um típico processo de reconhecimento de imagens, retirado de Araújo (2009). | 15 |
| Figura 2 - Imagens e suas matrizes respectivas: (a) imagem binária, (b) imagem em escala de cinza, (c) imagem colorida no formato RGB, retirado de Hamilton e Ribeiro (2008). | 16 |
| Figura 3 - À esquerda a imagem do motociclista na escala de cinza, ao centro, o histograma ilustrando o ponto de corte do <i>Threshold</i> , e à direita, a imagem de saída do algoritmo. | 18 |
| Figura 4 - Imagem do motociclista no padrão RGB(esquerda) e ao lado a imagem processada pelo algoritmo <i>Watershed</i> (direita). | 19 |
| Figura 5 - Ilustração do funcionamento do algoritmo para segmentação do fundo, retirado de Ferreira (2012). | 20 |
| Figura 6 - Esquema do neurônio de McCulloch e Pitts, retirado de Rodrigues (2002). | 22 |
| Figura 7 - Arquitetura de um MLP | 23 |
| Figura 8 - Exemplo de classificador utilizando árvore de decisão, retirado de Garcia (2000). | 24 |
| Figura 9 - Classificação de uma instância pela árvore de decisão, retirado de Garcia (2000). | 25 |
| Figura 10 - Matriz de confusão ou tabela de contingência, retirado de Araújo (2009). | 26 |
| Figura 11 - Gráfico ROC, retirado de Da Silva (2006). | 27 |
| Figura 12 - Curva ROC para classificadores contínuos, retirado de Braga (2000). | 28 |
| Figura 13 - Algoritmo Threshold aplicado em duas imagens com a mesma calibração. | 30 |
| Figura 14 - Imagem de fundo gerada pelo algoritmo Background Subtractor. | 31 |
| Figura 15 - Segmentação feita pelo o algoritmo <i>Background Subtractor</i> | 31 |
| Figura 16 - Frames de um vídeo sendo processado pelo o algoritmo <i>Background Subtractor</i> | 32 |
| Figura 17 - Imagem com as linhas que representam os grupos de sementes do algoritmo <i>Watershed</i> | 32 |
| Figura 18 - À esquerda a imagem foi dividida em três regiões pelo <i>Watershed</i> , duas que representam o asfalto e uma que representa o motociclista. À direita a imagem segmentada com o motociclista. | 33 |
| Figura 19 - Imagem com a menor região retangular que delimita a área segmentada. | 34 |
| Figura 20 - O quadrado amarelo representa a posição de corte na imagem. | 34 |
| Figura 21 - À esquerda, imagem contendo um passageiro. À direita, imagem contendo dois passageiros. | 37 |
| Figura 22 - Exemplo de uma imagem removida. É notória a dificuldade de se identificar o número de passageiro somente com o contorno. Na imagem à esquerda é possível observar que o motociclista está incompleto. | 37 |
| Figura 23 - Sumário com os resultados do classificador no modelo A. | 39 |
| Figura 24 - Gráfico ROC da classe UM. | 40 |
| Figura 25 - Gráfico ROC da classe DOIS. | 40 |
| Figura 26 - Sumário com os resultados do classificador no modelo B. | 41 |
| Figura 27 - Gráfico ROC da classe UM. | 42 |
| Figura 28 - Gráfico ROC da classe DOIS. | 42 |
| Figura 29 - Sumário com os resultados do classificador no modelo A. | 43 |
| Figura 30 - Gráfico ROC da classe UM. | 44 |
| Figura 31 - Gráfico ROC da classe DOIS. | 45 |
| Figura 32 - Sumário com os resultados do classificador no modelo B. | 46 |

| | |
|--|----|
| Figura 33 – Detalhes da classificação em um ponto demarcado..... | 47 |
| Figura 34 – Gráfico ROC da classe UM..... | 47 |
| Figura 35 – Gráfico ROC da classe DOIS..... | 48 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 – Tabela ilustrativa da descrição da imagem no arquivo de entrada csv..... | 35 |
| Tabela 2 – Tabela com resultados dos diferentes parâmetros para a classe UM. Em negrito, os melhores resultados para cada métrica. | 49 |
| Tabela 3 - Tabela com resultados dos diferentes parâmetros para a classe DOIS. Em negrito, os melhores resultados para cada métrica..... | 49 |

SUMÁRIO

| | | |
|-------|--|----|
| 1 | INTRODUÇÃO..... | 13 |
| 2 | REVISÃO BIBLIOGRÁFICA..... | 14 |
| 2.1 | Conceitos das áreas de Computação Gráfica e Visão Computacional..... | 14 |
| 2.1.1 | Imagem Digital..... | 15 |
| 2.1.2 | Aquisição..... | 16 |
| 2.1.3 | Segmentação..... | 17 |
| 2.2 | Algoritmos de Classificação..... | 21 |
| 2.2.1 | Redes Neurais Artificiais..... | 21 |
| 2.2.2 | Árvores de Decisão..... | 24 |
| 2.3 | Métricas para Avaliação de Classificadores..... | 25 |
| 2.3.1 | Análise ROC..... | 27 |
| 3 | DESENVOLVIMENTO..... | 29 |
| 3.1 | Segmentação..... | 29 |
| 3.2 | Descrição..... | 33 |
| 3.3 | Classificação..... | 36 |
| 4 | RESULTADOS EXPERIMENTAIS..... | 36 |
| 4.1 | Base de dados..... | 36 |
| 4.2 | Resultados Experimentais para a Árvore de Decisão J48..... | 38 |
| 4.2.1 | Experimentos com o Modelo A..... | 38 |
| 4.2.2 | Experimentos com o Modelo B..... | 41 |
| 4.3 | Resultados Experimentais para a Rede Neural Multilayer Perceptron..... | 43 |
| 4.3.1 | Experimentos com o Modelo A..... | 43 |
| 4.3.2 | Experimento com o Modelo B..... | 45 |
| 4.4 | Mudança nos Parâmetros da Rede Neural para o Modelo B..... | 48 |
| 5 | CONSIDERAÇÕES FINAIS..... | 50 |
| | REFERÊNCIAS..... | 51 |

1 INTRODUÇÃO

“A habilidade do ser humano em reconhecer e classificar objetos sempre impressionou e continua a impressionar os cientistas... Desde os primórdios da computação, a tarefa de implementar algoritmos emulando essa capacidade humana, tem-se apresentado como a mais intrigante e desafiadora!” (Marques de Sá, 2004).

É natural termos a percepção visual de tudo que nos rodeia. Identificamos e classificamos objetos constantemente. Porém, mesmo com o avanço da tecnologia, identificar e classificar objetos em imagens digitais, utilizando somente o computador, é algo bastante desafiador.

A *visão computacional* é uma área da *ciência da computação* que busca simular a percepção visual humana através de sistemas computacionais. Ainda imatura e diversa, começou a ser aprofundada após o final da década de 70 (DE MILANO; HONORATO, 2010). Podemos pensar que alguns dos motivos para o seu crescimento recente foram: a facilidade de se obter imagens digitais a um baixo custo, e a possibilidade de processá-las e armazená-las em grandes quantidades. O grande objetivo dessa área é produzir sistemas computacionais que sejam capazes de obter informações a partir de imagens. Informação é tudo aquilo que possibilita a tomada de decisões.

Este trabalho está inserido em um projeto da Fotosensores Tecnologia Eletrônica Ltda, que consiste no desenvolvimento de um sistema computacional para a detecção de infrações por excesso de passageiros em motos.

O principal objetivo desse trabalho é produzir um estudo preliminar que seja capaz de responder a hipótese a seguir:

Hipótese: É possível determinar o número de passageiros em motos, com taxa aceitável de erro (área ROC acima de 90%), utilizando apenas o contorno da moto e de seus passageiros.

Para se obter um contorno aceitável, é necessário realizar corretamente a tarefa de *segmentação* da moto/passageiros em relação ao restante da cena. Os algoritmos de segmentação empregados neste trabalho são descritos na Seção 0. Um critério subjetivo de qualidade na segmentação é um ser humano ser capaz de identificar a cabeça e ombros de cada passageiro, utilizando apenas o contorno segmentado. Essa premissa precisa ser

atendida, já que a base da nossa hipótese diz que o contorno fornece informação suficiente para uma boa classificação. As imagens que não atendem este critério foram removidas da base de dados.

Após os experimentos iniciais, obtivemos boas segmentações automáticas apenas em vídeos de trânsito (utilizando o algoritmo Background Subtractor, descrito na Seção 2.1.3c). Infelizmente, a base de dados de teste consiste apenas de imagens, embora seja simples coletar vídeos de trânsito com as características desejadas (frontais, e com resolução apropriada). Portanto, para verificar a hipótese, realizamos a segmentação semi-automática descrita na Seção 3.1

Neste trabalho a biblioteca OpenCV foi utilizada para auxiliar na construção dos algoritmos de processamento de imagens. E a ferramenta Weka (“Waikato Environment for Knowledge Analysis”) foi utilizada para auxiliar na classificação das imagens (WEKA, 2013).

Na Seção 2 é realizado um levantamento bibliográfico dos algoritmos, técnicas e ferramentas empregados neste trabalho. Na Seção 3 descrevemos os experimentos, e na Seção 4 apresentamos uma discussão sobre a base de dados e sobre os resultados.

2 REVISÃO BIBLIOGRÁFICA

Nesta seção apresentamos os principais conceitos utilizados durante a execução deste trabalho. Na Seção 2.1 descrevemos brevemente os conceitos das áreas de computação gráfica e visão computacional. Os algoritmos de classificação utilizados neste trabalho são apresentados na Seção 2.2, e as métricas utilizadas para avaliar estes algoritmos são descritas na Seção 2.3.

2.1 Conceitos das áreas de Computação Gráfica e Visão Computacional

Podemos definir a visão computacional como sendo a sub-área do processamento de imagens que estuda o desenvolvimento de métodos e técnicas que possibilitam a interpretação de uma imagem em um sistema computacional (CONCIN; AZEVEDO; LETA, 2008). Um dos grandes desafios na área de visão computacional é a criação de sistemas computacionais que sejam capazes de simular as capacidades humanas na realização de funções ligadas à análise de imagens (HUTCHINSON; HAGER; CORKE, 1996). Um sistema de visão

computacional deve ser dotado de algumas características como: habilidade para extrair informações relevantes a partir de uma cena que na maioria das vezes está repleta de informações não relevantes, habilidades para fazer inferência a partir de informações incompletas, e habilidades para reconhecer objetos com maior independência possível, de sua posição, tamanho, orientação e mudança no ponto de vista (ARAÚJO, 2009).

Geralmente um sistema de visão computacional utiliza um típico processo para o reconhecimento de padrões em imagens. Este processo é composto por cinco etapas que são: a aquisição da imagem, o pré-processamento, a segmentação, a representação e descrição, e o reconhecimento e interpretação (ARAÚJO, 2009). Este processo pode ser visto na Figura 1.

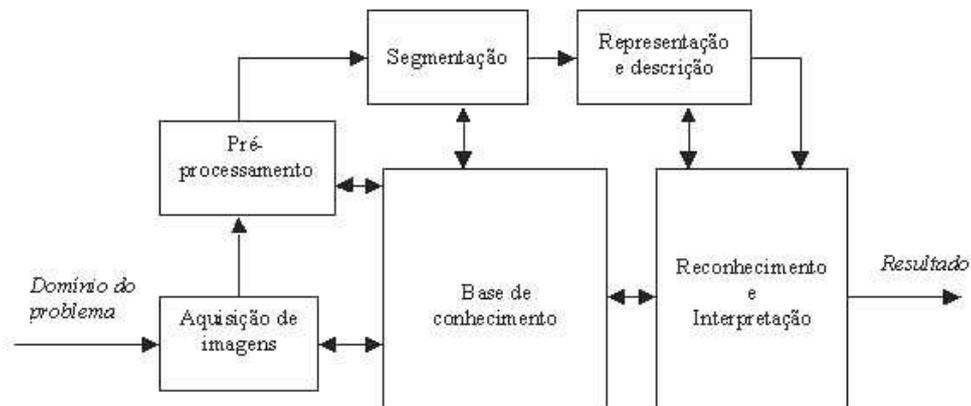


Figura 1 - Etapas de um típico processo de reconhecimento de imagens, retirado de Araújo (2009).

2.1.1 Imagem Digital

Uma imagem pode ser definida como a representação visual de um objeto. Matematicamente ela é definida como uma função bidimensional $f(x,y)$ com $x, y \in \mathbb{R}^2$, onde x e y representam as coordenadas espaciais que identificam um ponto, e o valor da função f representa o nível da intensidade em qualquer ponto (x,y) da imagem (ARAÚJO, 2009). Quando (x,y) e a amplitude de f fazem parte de um conjunto de valores finitos, ou discretos, a imagem é chamada de imagem digital (GONZALEZ; WOODS, 2007).

Computacionalmente, não podemos representar funções contínuas. Podemos somente simulá-las (ESCAÑO, 1999). Uma imagem digital é uma imagem real discretizada em suas coordenadas espaciais e em seu nível de intensidade. Uma imagem digital pode ser representada através de uma matriz de N linhas e M colunas, onde cada elemento da matriz corresponde ao nível de intensidade em um determinado ponto da imagem. Estes elementos são conhecidos como pixels. Nas imagens binárias os pixels podem assumir os valores 0 e 1,

que correspondem as cores preto e branco, respectivamente. Nas imagens de tonalidade cinza os valores dos pixels podem variar entre 0 a 255, demonstrando assim 256 tonalidades diferentes. Diferente da outras, a imagem colorida possui três canais, ou seja, cada pixel é composto por três valores que variam de 0 a 255 no formato RGB, totalizando assim 16.777.216 cores diferentes (HAMILTON; RIBEIRO, 2008). À esquerda da Figura 2 é apresentada uma imagem de nove pixels em suas formas binária, em escala de cinza e colorida. À sua direita é apresentada a matriz com os valores respectivos.

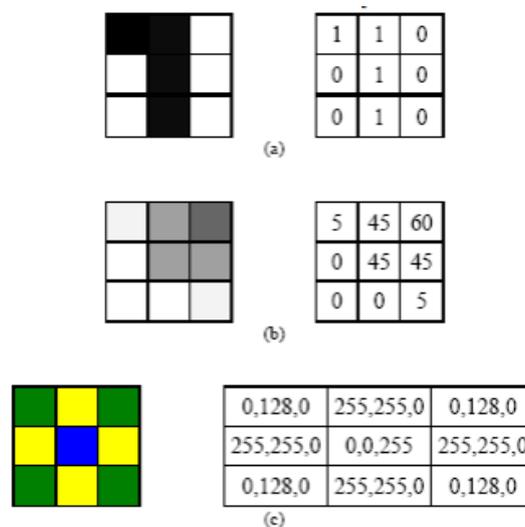


Figura 2 - Imagens e suas matrizes respectivas: (a) imagem binária, (b) imagem em escala de cinza, (c) imagem colorida no formato RGB, retirado de Hamilton e Ribeiro (2008).

2.1.2 Aquisição

A Fotosensores Tecnologia Eletrônica Ltda. utiliza um Sistema de Monitoramento de Trânsito com Câmera Digital (SMTCD), que é um equipamento foto sensor. Ele é composto basicamente de um sensor e uma câmera digital. Esse sistema de aquisição é capaz de detectar os veículos passando pela via. Para cada veículo detectado, é disparada uma chamada à câmera digital para que ela se encarregue de digitalizar a imagem naquele momento. O equipamento é alocado ao lado das vias e é capaz de monitorar até quatro faixas de trânsito (FOTOSENSORES, 2012).

2.1.3 Segmentação

A segmentação é a divisão de uma imagem em um conjunto de regiões homogêneas em relação aos atributos ou propriedades que são definidos, tais como níveis de cinza, contraste, ou propriedades de textura (PAIVA, 2010). Suponha que R represente a região inteira de uma imagem e P uma propriedade qualquer, o processo de segmentação consiste em identificar o conjunto das sub-regiões $R_1, R_2, R_3, \dots, R_N$, de modo que:

1. $R = R_1 \cup R_2 \cup \dots \cup R_N$.
2. $R_i \cap R_j = \emptyset, \forall i \neq j$.
3. $P(R_i) = \text{verdadeiro}, \forall i$.
4. $P(R_i \cup R_j) = \text{falso}, i \neq j$.

Na primeira condição, é indicado que R é formado pela união das sub-regiões; a segunda condição fala que as regiões devem ser distintas, ou seja, não existe uma sub-região que contenha algum elemento de outra sub-região; a terceira condição está relacionada às propriedades que cada região segmentada contém e a última condição exclama que as regiões R_i e R_j possuem diferentes propriedades.

Existem diversas técnicas que podem ser utilizadas para a segmentação, essas técnicas são classificadas em quatro grupos:

- Segmentação por detecção de borda.
- Segmentação por corte.
- Segmentação por crescimento de regiões.
- Segmentação por subtração de fundo

a) Segmentação por Corte (*Threshold*)

As técnicas de segmentação por cortes em níveis de cinza são métodos de baixo custo computacional e simples de serem implementados, eles são utilizados para particionar uma imagem em regiões mutuamente exclusivas. A idéia central deste tipo de operador é separar objetos com base na variação de intensidade entre os próprios pixels do objeto e os pixels do fundo. Para diferenciar os pixels de interesse do resto dos pixels (que eventualmente serão

rejeitados), é realizada uma comparação do valor da intensidade de cada pixel em relação a um limiar (determinado de acordo com o problema a resolver).

Uma vez que os pixels importantes estejam separados adequadamente, podemos defini-los com um valor determinado para identificá-los (ou seja, podemos atribuir-lhes um valor de 0 (preto), 255 (branco) ou qualquer valor que se adapte às necessidades). Na Figura 3, é possível ver uma ilustração da segmentação por este algoritmo. À esquerda, a imagem de um motociclista em escala de cinza, ao centro, o seu histograma com o local da partição, e à direita, a imagem resultante do processamento do algoritmo. Esse método é conhecido como *Threshold*.

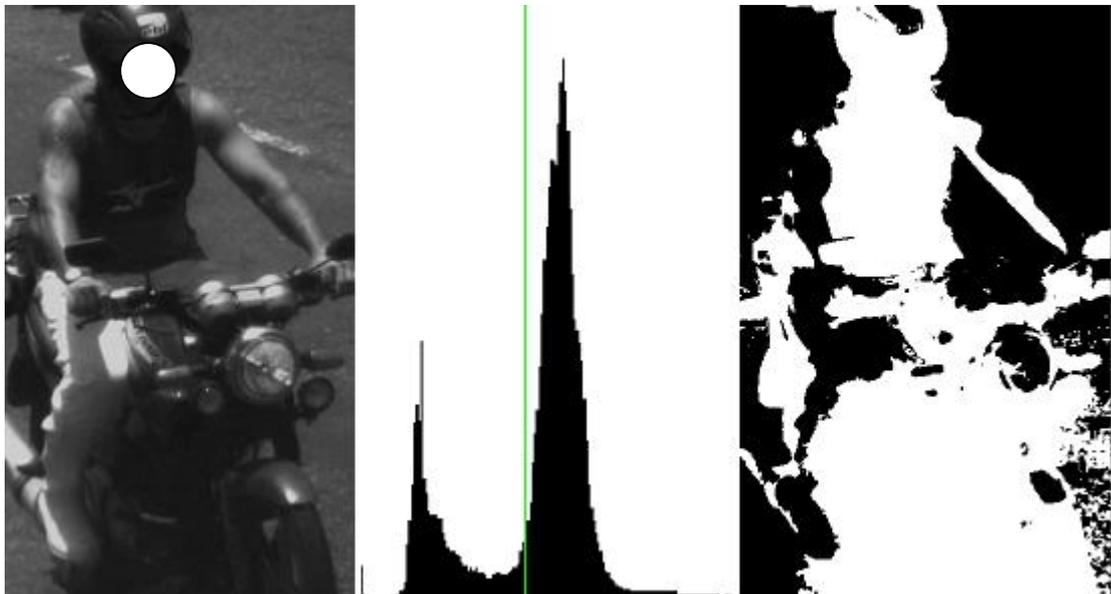


Figura 3 - À esquerda a imagem do motociclista na escala de cinza, ao centro, o histograma ilustrando o ponto de corte do *Threshold*, e à direita, a imagem de saída do algoritmo.

b) Segmentação por crescimento de regiões

Segundo Marengoni e Stringhini (2009) este tipo de segmentação encontra regiões diretamente na imagem agrupando pixels ou sub-regiões em regiões maiores baseado em critérios de crescimento pré-definidos. O procedimento parte de um conjunto de pontos, chamados de sementes, e a partir destes pontos são agrupados outros pontos utilizando uma vizinhança de influência, formando assim, as regiões. Nesta vizinhança são analisadas algumas propriedades e são medidas similaridades para determinar se o pixel faz parte ou não da região a ser analisada. As propriedades normalmente consideradas são: cor, intensidade de nível de cinza, textura, momentos, etc.

Algoritmo Watershed

O algoritmo *Watershed* foi proposto primeiramente por Digabel e Lantuéjoul (1978) e posteriormente foi melhorado por Beucher e Lantuéjoul (1979). Ele é um dos principais métodos dentro do grupo de segmentação por crescimento de regiões. Esse método é baseado no princípio de *inundação de relevos topográficos* que foi mostrada em Russ (1998). Nesta abordagem é possível abstrair a imagem em nível de cinza como sendo um relevo topográfico. Os vales do relevo representam os pixels de regiões mais escuras da imagem, enquanto que as montanhas são representadas por regiões mais claras da imagem. É possível imaginar a inundação de duas maneiras distintas: a água caindo por cima, como se fosse chuva, e se aglomerando nas regiões dos vales; ou vinda de baixo, como se o relevo estivesse perfurado nas regiões de baixa altitude e fosse submerso em um lago. À medida que as bacias vão sendo inundadas, as águas de diferentes sub-regiões se encontram. Esses encontros são representados pelas linhas divisórias de água, as chamadas *Watershed*. Quando a água atinge o maior pico no relevo, o processo termina. E o resultado é o relevo particionado em regiões definidas pela *Watershed*. (PECCINI; ORNELLAS, 2004).

Na Figura 4, é possível observar a imagem do motociclista no padrão *Red-Green-Blue* (RGB), e ao lado, a imagem dividida em quatro regiões que foram encontradas pelo algoritmo *Watershed*. Vale ressaltar que os grupos de sementes que deram origem as regiões foram definidos manualmente.



Figura 4 - Imagem do motociclista no padrão RGB(esquerda) e ao lado a imagem processada pelo algoritmo *Watershed* (direita).

c) Background Subtractor

O *Background Subtractor* é um método amplamente utilizado para detectar objetos em vídeos. Segundo Oliveira (2010) para a utilização desta abordagem é necessário cumprir alguns requisitos, como a necessidade de utilizar uma câmara estática, os objetos que vão ser detectados têm que estar em movimento e o fundo da cena tem de variar pouco ou de uma forma previsível. Se algum desses requisitos não for cumprido então a utilização dessa abordagem não poderá ser aplicada.

O algoritmo funciona com a “subtração” do quadro atual do vídeo com um modelo de fundo aprendido. Essa subtração varia desde operações da matemática elementar até complexos cálculos probabilísticos. Ao mesmo tempo em que o algoritmo subtrai o quadro atual, ele pode reatualizar o modelo de fundo com o mesmo quadro. Para a construção do modelo, existem diversas abordagens. Uma das mais simples é calcular a média de cada pixel em um conjunto de amostras.

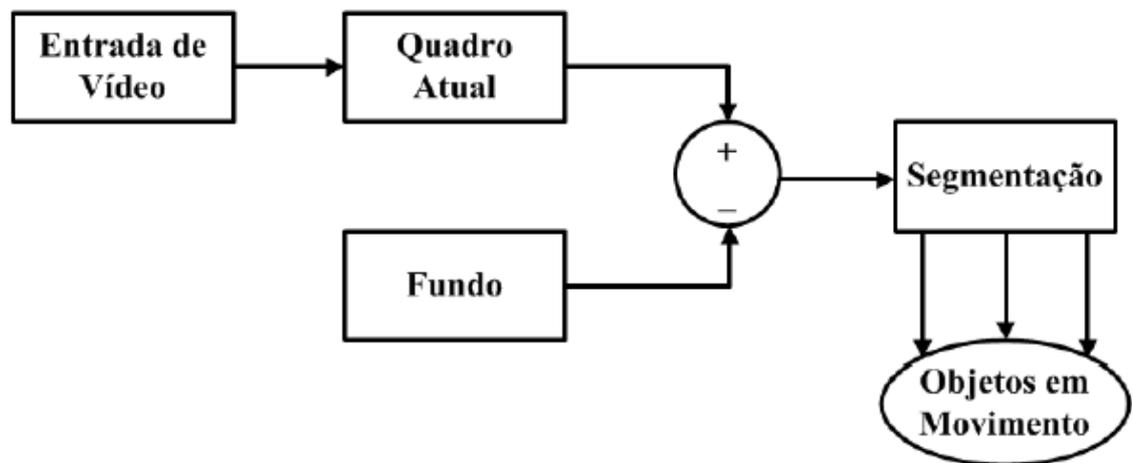


Figura 5 – Ilustração do funcionamento do algoritmo para segmentação do fundo, retirado de Ferreira (2012).

Matematicamente o algoritmo pode ser definido pela seguinte equação,

$$d(x, y, t) = \begin{cases} 1, & \text{se } |f(x, y, t) - B(x, y)| > T_d \\ 0, & \text{caso contrário} \end{cases}$$

onde T_d , é um limiar pré determinado, $f(x, y, t)$ é uma imagem tomada no tempo t , e $B(x, y)$ representa o modelo de fundo. Na análise final do cálculo, para todo $d(x, y, t)$ com o valor igual a “1” é considerado como pertencente a objeto em movimento na cena (FERREIRA, 2012).

2.2 Algoritmos de Classificação

Segundo Teixeira (2011), a classificação consiste na associação de um determinado conjunto de características à classe em que se enquadra. A classificação pode acontecer de duas maneiras: classificação supervisionada e classificação não supervisionada. Os classificadores supervisionados requerem treinamento antes da classificação, já os classificadores não supervisionados trabalham diretamente nos dados, assim dispensando a etapa de treinamento.

Um classificador tem como objetivo identificar um objeto a partir de suas características. Estas características são as entradas do classificador que são definidas de acordo com cada aplicação.

2.2.1 Redes Neurais Artificiais

Redes neurais artificiais são técnicas computacionais que propõem um modelo matemático baseado na estrutura neural de seres inteligentes, mais especificamente o cérebro humano (TAFNER; XEREZ; FILHO, 1995). Uma das grandes capacidades das redes neurais artificiais é aprender a partir de exemplos e, assim, classificar novos padrões (RODRIGUES, 2002).

A rede neural artificial é composta por neurônios que são agrupados por camadas. O neurônio artificial é o elemento básico que forma a rede neural artificial, ele simula o funcionamento de um neurônio biológico. Um neurônio possui um ou mais sinais de entrada. São esses sinais que serão processados e de acordo com o resultado, o neurônio artificial é ativado ou não. O processamento ocorre da seguinte maneira:

- Os sinais de entrada (X_1, X_2, \dots, X_p) são apresentados ao neurônio.
- Cada sinal é multiplicado por um número, que é conhecido como peso sináptico, esse número mostra a influência da entrada na saída do neurônio (W_1, W_2, W_p).
- Acontece uma soma ponderada dos sinais já multiplicados pelos pesos sinápticos, o resultado ilustra a atividade do neurônio (Σ).
- Uma função de ativação verifica o nível de atividade do neurônio, se este exceder a um limiar o neurônio é ativado, caso contrário ele permanece inativo.

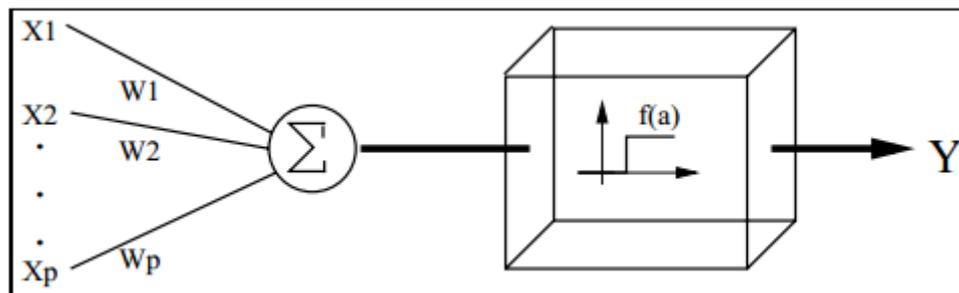


Figura 6 - Esquema do neurônio de McCulloch e Pitts, retirado de Rodrigues (2002).

Assim como ocorre no neurônio biológico, o neurônio artificial processa todos os sinais de forma simultânea, ou seja, no mesmo processamento. Uma rede neural se assemelha ao cérebro de duas maneiras (HAYKIN, 1999):

- O conhecimento é adquirido na rede a partir de um processo de aprendizagem.
- Forças de conexão entre os neurônios, conhecidos como pesos sinápticos, são utilizados para armazenar o conhecimento adquirido no processo de aprendizagem

Apesar das redes neurais serem utilizadas em diversos campos, os seus objetivos são quase sempre os mesmos. Elas são utilizadas para reconhecer e classificar padrões além de generalizar informações.

Uma rede neural possui dois momentos distintos no processamento, um é momento de treinamento, e o outro é o momento de utilização. Já em um projeto de rede neural podemos contar com três momentos distintos, que são: momento de definição, momento de aprendizagem, e momento de utilização (LOESCH; SARI, 1996).

a) **Multilayer Perceptron**

O *Multilayer Perceptron* (MLP) foi concebido para resolver problemas mais complexos, os quais não poderiam ser resolvidos pelo modelo de neurônio básico (TATIBANA; KAETSU). Essas redes neurais de múltiplas camadas diferem das redes neurais simples pelo o número de camadas ocultas. As camadas ocultas ou camadas intermediárias são as camadas que ficam entre os neurônios da camada de entrada e os neurônios da camada de saída (AFFONSO; SILVA; SILVA; RODRIGUES, 2010). A função das camadas ocultas é intervir entre a camada de entrada e a camada de saída da rede de uma forma útil. Ao

adicionar uma ou mais camadas ocultas, a rede se habilita a extrair estatísticas de ordem superior (KISI, 2005). Para maiores informações sobre MLP, ver Haykin (1994).

Um MLP pode ter mais de uma camada oculta, no entanto, trabalhos teóricos têm mostrado que uma única camada oculta é suficiente para uma rede neural artificial se aproximar de qualquer complexo de função não linear (CYBENCO, 1989; HORNIK ET AL, 1989).

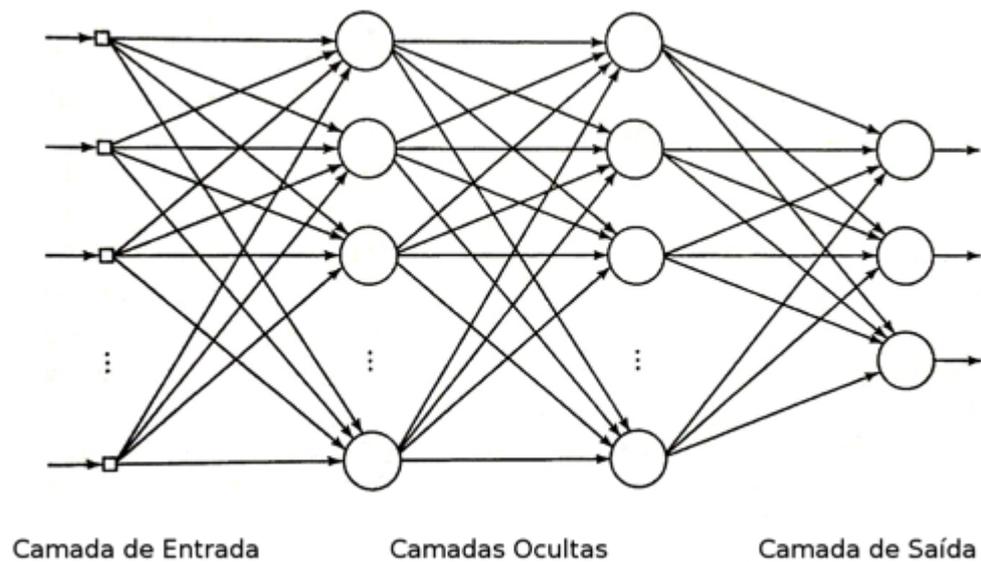


Figura 7 – Arquitetura de um MLP

- **Camada de entrada:** é nessa camada onde a informação de entrada fica gravada e é repassada para as outras camadas, nela não há nenhum tipo de processamento.
- **Camadas ocultas:** elas são responsáveis por transmitir as informações de entrada para a saída, mais antes de transmitir essas informações, elas sofrem um processamento. As conexões guardam os pesos, e esses pesos serão multiplicados pelos os valores de entrada pra depois passar pra camada seguinte até chegar à camada de saída.
- **Camada de saída:** é nessa camada onde o resultado é apresentado.

Um dos métodos mais utilizados para treinamento de modelos MLP é o *Backpropagation* que se baseia na aprendizagem por minimização dos erros. O *Backpropagation* é um algoritmo de aprendizado supervisionado. A idéia geral é que quando o valor da saída do algoritmo difere da saída esperada, o erro é calculado e seus valores retropropagados para a entrada. Os pesos são ajustados e o valor de saída calculado novamente (AFFONSO; SILVA; SILVA;

RODRIGUES, 2010). De acordo com Ferrari et al. (2006) o algoritmo funciona da seguinte maneira:

- Primeiro passo é apresentar o padrão à camada de entrada da rede.
- Logo em seguida o padrão é processado camada por camada até que a camada de saída forneça a resposta processada.
- A resposta de saída é comparada com a resposta esperada, caso a resposta esteja errada, o erro é calculado.
- Então o erro é retropropagado da camada de saída para a camada de entrada e conforme isso acontece, os pesos das ligações entre as camadas são ajustados e o processamento feito novamente enquanto não houver uma resposta desejada.

2.2.2 Árvores de Decisão

Árvore de decisão é um método poderoso para a classificação e previsão de resultados que tem o objetivo de facilitar a tomada de decisão em problemas de decisão sequenciais. Ela utiliza o modelo de aprendizado supervisionado. E são similares a regras if-else.

Uma árvore de decisão utiliza a estratégia de *divisão e conquista*, ou seja, um problema complexo é dividido em subproblemas, onde esses subproblemas são também divididos em outros subproblemas de forma recursiva (CREPALDI ET AL, 2010). Ela tem uma estrutura similar a um fluxograma, onde cada nó interno denota uma condição a determinado atributo, cada galho representa um resultado de teste, e cada nó folha guarda um rótulo de classe que se quer classificar. O nó superior em uma árvore é chamado de nó raiz (SANTOS; NETO, 2010). Na Figura 8, é ilustrado a estrutura de um classificador baseado em árvore de decisão.

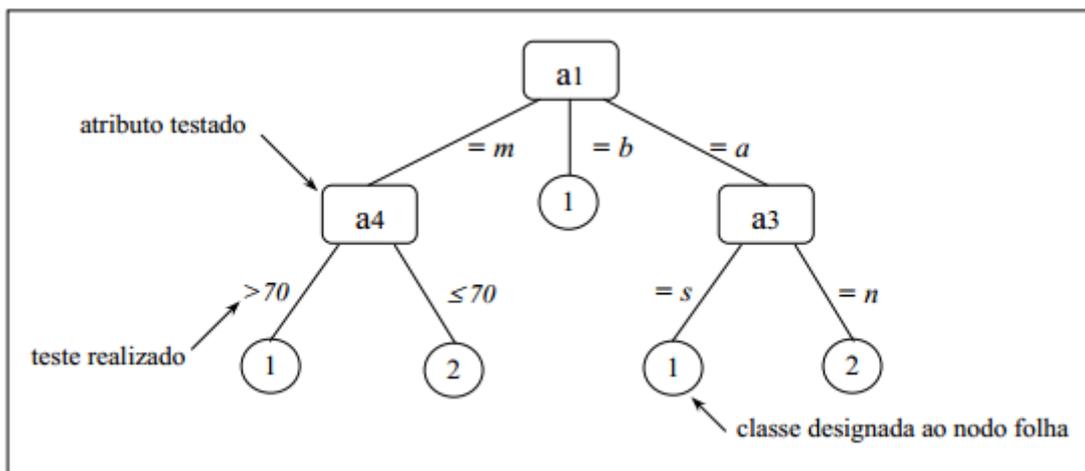


Figura 8 – Exemplo de classificador utilizando árvore de decisão, retirado de Garcia (2000).

Existem dois tipos de atributo em uma árvore de decisão, o tipo categórico que é representado por uma igualdade, e o tipo quantitativo que é representado por uma faixa de valores. No exemplo acima é possível ver os tipos de atributos nos galhos da árvore, por exemplo: $=m$ (tipo categórico) e >70 (tipo quantitativo).

A raiz de uma árvore de decisão é representada pelo atributo mais importante, enquanto que os nós menos importantes são representados pelos nós subseqüentes. A sua vantagem é a tomada de decisão levando em consideração os atributos mais relevantes (CREPALDI ET AL, 2010). A classificação de uma instância na árvore de decisão ocorre ao se percorrer o caminho do nó raiz até um nó folha. Na Figura 9 é possível ver a classificação de uma instância que tinha os seguintes valores de atributos: $a1 = m$; $a3 = s$; $a4 = 80$.

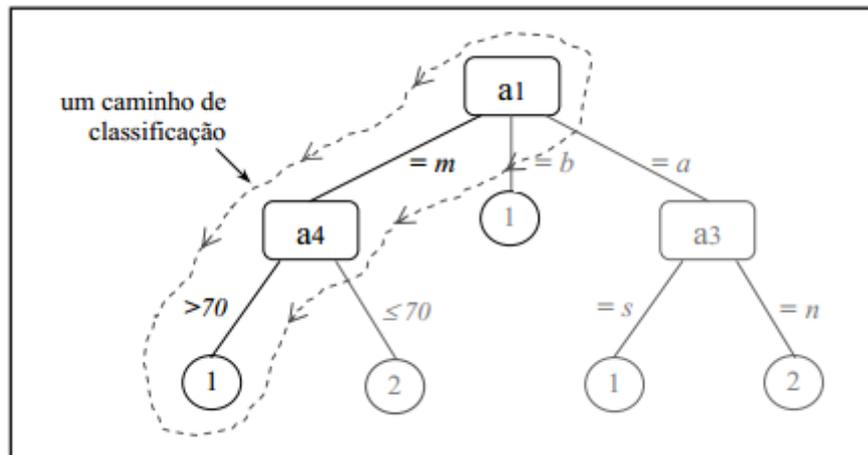


Figura 9 – Classificação de uma instância pela árvore de decisão, retirado de Garcia (2000).

2.3 Métricas para Avaliação de Classificadores

Algumas métricas podem ser usadas para a avaliação de um classificador. Um exemplo é a precisão, que representa o percentual dos registros classificados corretamente de uma classe, dentre o número de registros classificados como pertencentes a classe. Outro exemplo é a acurácia, que fornece o percentual de acerto de todas as classes do classificador dentre as instâncias totais. Essas métricas podem indicar o quão o modelo de classificação é confiável (DALLAGASSA, 2011). Essas e outras métricas são calculadas a partir da matriz de confusão ou tabela de contingência mostrada na Figura 10.

| | | Existe uma instância do objeto na imagem | |
|--|----------------|--|---|
| | | Positivo (SIM) | Negativo (Não) |
| O algoritmo detectou uma instância do objeto na imagem | Positivo (SIM) | Verdadeiro Positivo (VP) Detecção correta | Falso Positivo (FP) Erro na detecção |
| | Negativo (Não) | Falso Negativo (FN) Erro na detecção | Verdadeiro Negativo (VN) |

Figura 10 – Matriz de confusão ou tabela de contingência, retirado de Araújo (2009).

Definição das categorias Verdadeira Positivo, Falso Positivo, Falso Negativo e Verdadeiro Negativo (ARAÚJO, 2009).

- Verdadeiro Positivo (VP): ocorre quando o objeto que se procura está na imagem e ele é detectado pelo algoritmo.
- Falso Positivo (FP): ocorre quando o objeto que se procura não está contido na imagem e é detectado pelo algoritmo.
- Falso Negativo (FN): ocorre quando o objeto que se procura está na imagem e não é detectado pelo algoritmo.
- Verdadeiro Negativo (VN): ocorre quando objeto que se procura não está na imagem e ele não é detectado pelo algoritmo

Abaixo são apresentas algumas métricas que podem ser obtidas a partir da matriz de confusão ou tabela de contingência.

$$\text{Taxa de verdadeiros positivos (TVP ou TP Rate ou Recall)} = \frac{VP}{VP + FN}$$

$$\text{Taxa de falsos positivos (TFP ou FP Rate)} = \frac{FP}{FP + VN}$$

$$\text{Precisão (Precision)} = \frac{VP}{VP + FP}$$

$$\text{Acurácia} = \frac{VP + VN}{VP + FP + VN + FN}$$

$$\text{F-Measure} = \frac{2 * \textit{Precision} * \textit{Recall}}{\textit{Precision} + \textit{Recall}}$$

2.3.1 Análise ROC

A análise ROC (Receiver Operating Characteristic) é um método gráfico utilizado na área de aprendizado de máquina para avaliação de classificadores. Teve início na teoria de decisão estatística onde foi utilizada entre as décadas de 50 e 60 para avaliar a qualidade de transmissão de um sinal em um canal com ruídos na detecção de sinais (BRAGA, 2000).

A análise é feita através de um gráfico bidimensional formado pela taxa de falsos positivos (TFP) no eixo das abscissas, e pela taxa de verdadeiros positivos (TVP) no eixo da ordenadas.

Os classificadores discretos e contínuos são representados de forma diferente no gráfico ROC. Um *classificador discreto* classifica um objeto somente com a informação se ele é ou não pertencente à classe. A sua representação no gráfico ROC é feita por pontos. Um ponto é obtido através do cálculo da TFP e da TVP do resultado final de uma classificação em uma amostragem de dados. Um *classificador contínuo* classifica um objeto com um valor de *score* que representa a probabilidade da instância pertencer à classe que está sendo analisada. A sua representação no gráfico ROC é feito através de uma curva. Essa curva é montada variando um limiar de acordo com *score* de classificação.

No gráfico ROC existem algumas regiões que merecem destaque. Veja na Figura 11 a ilustração de quatro regiões importantes e uma linha na diagonal que representa os classificadores aleatórios (DA SILVA, 2006).

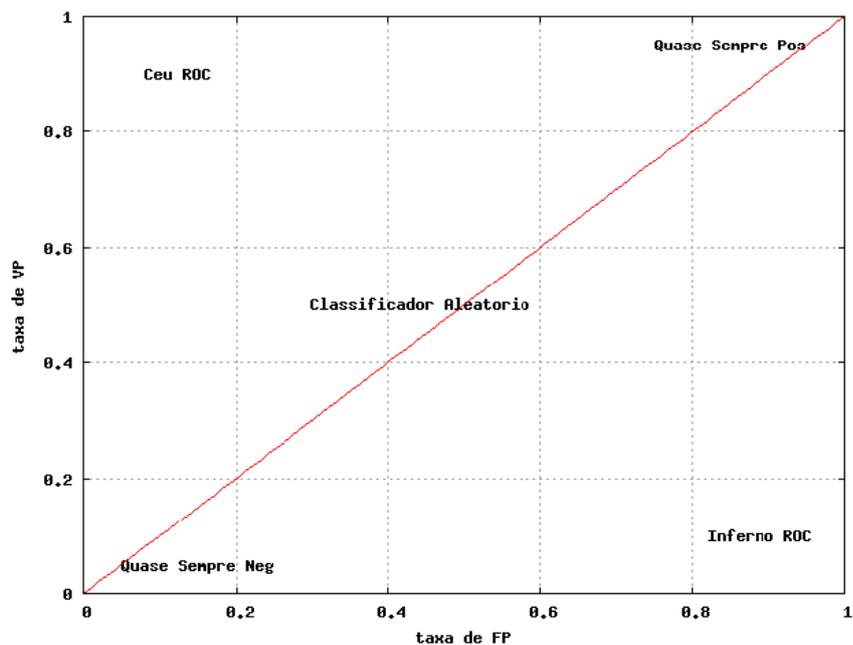


Figura 11 – Gráfico ROC, retirado de Da Silva (2006).

Em classificadores discretos, os pontos que caem na região *Céu ROC*, são os que estão mais perto de uma classificação perfeita, ou seja, uma baixa taxa de falsos positivos e uma alta taxa de verdadeiros positivos. Já os pontos que caem no *Inferno ROC*, são os pontos que chegam a ser pior do que os classificadores aleatórios, mas isso não é ruim, eles possuem informações com capacidade de distinguir as classes, porém de maneira errada. Portanto o resultado pode ser facilmente revertido, basta que seja trocada a resposta do classificador por sua resposta inversa, em outras palavras, o ponto que era (x,y) passa a ser (y,x) . Os pontos que caem na região *Quase Sempre Neg* rotulam quase sempre resultados negativos, e eles são decorrentes de classificadores conservadores, aqueles que não admitem produzir falsos positivos, porém são modelados de uma maneira que não conseguem rotular os verdadeiros positivos. Já na região *Quase Sempre Pos* acontece o contrario, os classificadores rotulam quase sempre resultados positivos, e eles são modelados de uma maneira que acabam produzindo muitos falsos positivos. Os pontos que caírem próximos a linha vermelha são considerados ruins, pois a linha vermelha representa os classificadores aleatórios. Estes tipos de classificadores não sabem nada sobre a classe, eles sorteiam os resultados aleatoriamente e por isso não agregam valor (DA SILVA, 2006).

Em um classificador contínuo, é possível ajustar seu resultado, a fim de atender a um critério escolhido. Um critério pode ser *estrito*, *moderado* ou *brando*. Na Figura 12 é mostrada uma curva formada por um classificador contínuo e as possíveis regiões onde os critérios se localizam.

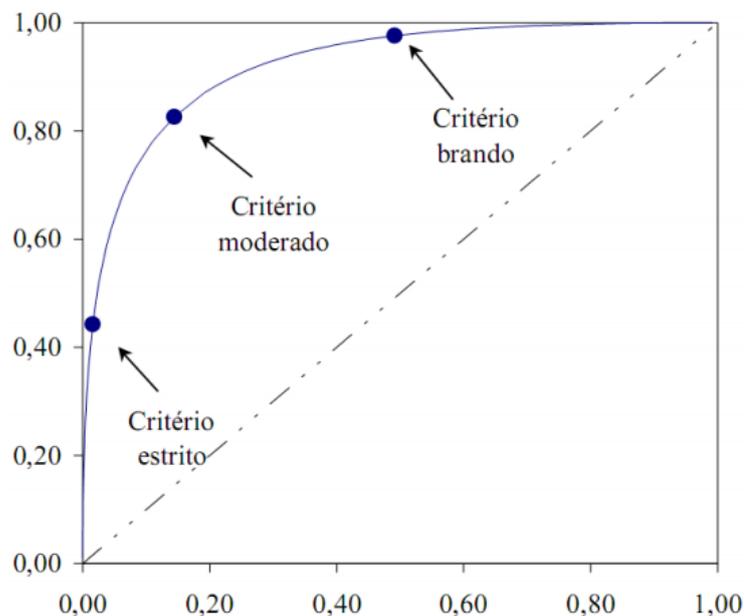


Figura 12 – Curva ROC para classificadores contínuos, retirado de Braga (2000).

O critério estrito é utilizado quando se admite somente pequenas frações de falsos positivos na classificação, por exemplo, em um classificador de spam uma instância é declarada positiva quando se tem uma evidência muito forte para isso. O critério moderado é utilizado quando se quer chegar ao máximo próximo de boa classificação, ou seja, baixa taxa de falsos positivos e alta taxa de verdadeiros positivos representado no gráfico pelo ponto (0,1). O critério brando é utilizado quando se quer uma alta taxa de verdadeiros positivos, e a taxa de falsos positivos não é tão prejudicial.

3 DESENVOLVIMENTO

Esse trabalho foi dividido em três etapas: (i) a etapa de segmentação, (ii) a etapa de descrição, que tem como objetivo transformar a imagem segmentada em um vetor de características, e (iii) a etapa de classificação, que tem como objetivo classificar cada imagem usando seu vetor de características como entrada. Descrevemos a seguir cada uma destas etapas.

3.1 Segmentação

Nesta etapa, o passo inicial é separar o motociclista do resto da imagem. Como citado anteriormente foi feito um estudo inicial com o objetivo de tornar a segmentação automática. Porém o objetivo não foi alcançado, pois diversos problemas foram encontrados. Os algoritmos testados durante o estudo foram o *Threshold* da luminância e o *Background Subtractor* (KAEWTRAKULPONG; BOWDEN, 2001). Os problemas encontrados são discutidos a seguir.

O *Threshold* da luminância mostrou que durante a sua segmentação, algumas partes específicas da região do motociclista eram descartadas junto com o asfalto. O problema ocorreu pois, nessas partes, as taxas de intensidade são muito próximas das taxas de intensidade do asfalto. Na Figura 3, é possível visualizar que os braços do motociclista ficaram incompletos após o processamento. Outro problema dessa abordagem é a calibração do limiar de corte em cada imagem que difere de uma pra outra. A ajuda humana seria necessária na calibração do algoritmo para este tipo de problema. A Figura 13 ilustra o problema encontrado. A imagem (a) foi processada pelo algoritmo onde foi definido um limiar de corte manualmente, o resultado pode ser visto na imagem (b). Depois o algoritmo com o mesmo limiar foi usado na imagem (c) que por sua vez não teve um bom resultado, o

resultado pode ser visto na imagem (d). Ou seja, não foi possível encontrar uma única configuração que produza bons resultados para mais de 90% das imagens. Uma alternativa é obter uma calibração dinâmica, com base em características da imagem, o que sugerimos como trabalho futuro.

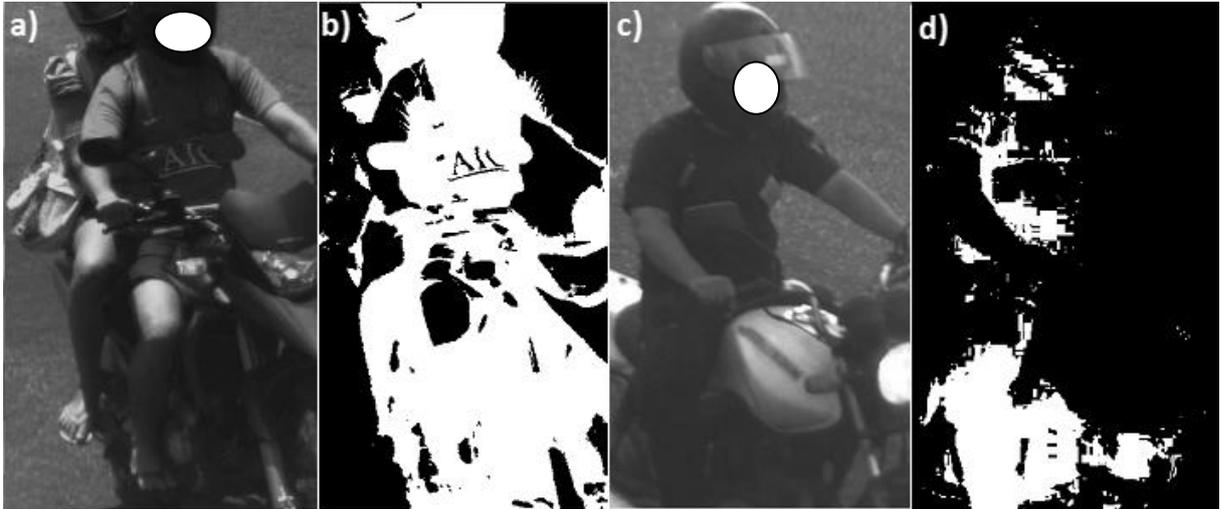


Figura 13 – Algoritmo Threshold aplicado em duas imagens com a mesma calibração.

O *Background Subtractor*, foi configurado para processar todas as imagens, simulando a reprodução de um vídeo, para que a partir delas pudesse ser gerado um modelo de fundo para a segmentação. O problema dessa abordagem é que as imagens da base de dados foram capturadas em diversos horários do dia, ou seja, a iluminação particular de cada horário influenciou na formação das diferentes cores de asfalto nas imagens. Como a aprendizagem do algoritmo não ocorreu de forma natural (através de um vídeo), essa mudança repentina na cor do asfalto de uma imagem para outra, atrapalhou a saída do algoritmo. A explicação para isso é que quando uma imagem é processada, se ela está com uma cor de asfalto diferente (mais claro ou mais escuro) do asfalto do modelo de fundo, essa diferença de cor é interpretada como um novo objeto na cena. Na Figura 14 é possível ver o modelo da imagem de fundo aprendida pelo algoritmo. E na Figura 15 é possível ver a saída do algoritmo, onde uma parte do asfalto foi interpretada como frente e não como fundo.



Figura 14 – Imagem de fundo gerada pelo algoritmo Background Subtractor.



Figura 15 – Segmentação feita pelo o algoritmo *Background Subtractor*

Mesmo com a má segmentação produzida pelo *Background Subtractor* nas imagens, foi visto que existem evidências claras de que ele produz bons resultados em vídeos de trânsito. Uma evidência é que todos os requisitos para um bom funcionamento do algoritmo são atendidos no processo de aquisição de imagens, como por exemplo: a posição da câmera é estática, os motociclistas que trafegam na via estão em movimento, e o fundo da cena (asfalto) muda de forma gradual com o passar do dia. A outra evidência surgiu após a aplicação do algoritmo em um vídeo de trânsito disponibilizado pela empresa Fotosensores para a base de dados de outro projeto. O vídeo se encontra em uma posição diferente, o motociclista é visto de costas na cena, porém o objetivo de processá-lo é mostrar que o algoritmo segmenta corretamente para esse tipo de problema uma vez que os requisitos são atendidos. Um critério subjetivo de qualidade na segmentação é um ser humano ser capaz de identificar a cabeça e ombros de cada passageiro, utilizando apenas o contorno segmentado. Na Figura 16 é visto que esse critério subjetivo é atendido. Concluímos então que a premissa de boa segmentação

pode ser obtida com procedimentos totalmente automáticos, desde que a base de dados consista de vídeos de trânsito com as características de resolução, zoom e inclinação das imagens que usamos em nossos experimentos.



Figura 16 – Frames de um vídeo sendo processado pelo o algoritmo *Background Subtractor*.

O objetivo desse trabalho é mostrar que, com um bom recorte, é possível classificar o número de ocupantes em uma moto, com uma boa avaliação. Como a abordagem proposta para resolver esse problema exige uma boa segmentação, então foi optado por fazer uma segmentação semi-automática utilizando o algoritmo *Watershed* (MEYER, 1992). Primeiramente foram definidos grupos de sementes na região do motociclista e na região do asfalto. Abaixo, na Figura 17, é possível ver os grupos de sementes representadas pelas linhas coloridas. A linha vermelha e a linha azul representam o grupo de sementes da região do asfalto, enquanto que a linha verde representa o grupo de sementes da região do motociclista.



Figura 17 – Imagem com as linhas que representam os grupos de sementes do algoritmo *Watershed*.

Depois da definição dos grupos de sementes o algoritmo foi executado a fim de separar as regiões desejadas. A saída do algoritmo pode ser vista na Figura 18. À esquerda a imagem dividida em três regiões pelo *Watershed*, e à direita a imagem segmentada somente com a região do motociclista.

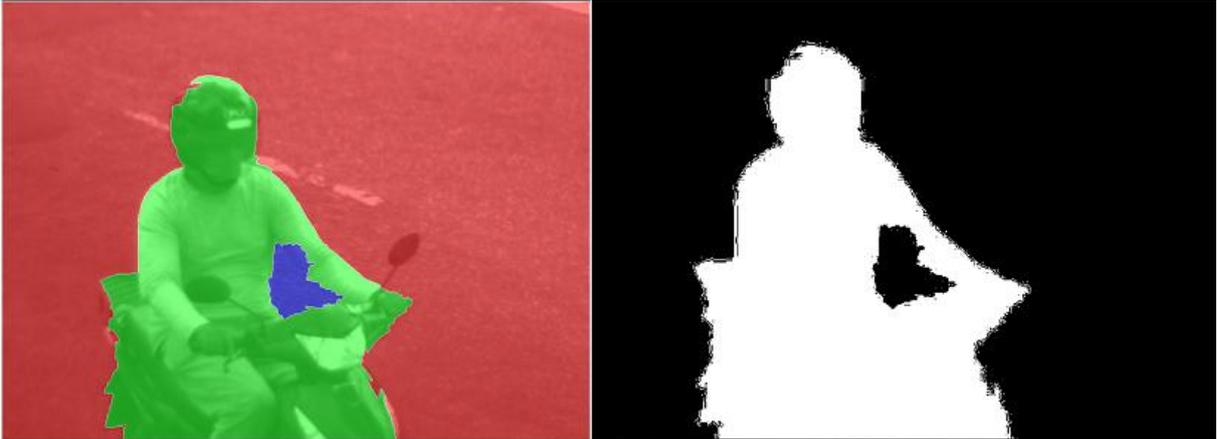


Figura 18 – À esquerda a imagem foi dividida em três regiões pelo *Watershed*, duas que representam o asfalto e uma que representa o motociclista. À direita a imagem segmentada com o motociclista.

A saída desta etapa é uma imagem binária, onde o motociclista segmentado é representado pela cor branca, e o resto da imagem é representado pela cor preta.

3.2 Descrição

Depois da etapa de segmentação, é dado início a etapa de descrição. O objetivo dessa etapa é transformar a imagem segmentada em um vetor de características para servir de entrada para os classificadores. Os passos realizados nesta etapa são descritos a seguir.

Primeiramente a resolução da imagem foi diminuída para a resolução de 38x50 pixels. O objetivo desse passo é simplificar ao máximo a informação sem afetar a capacidade de classificação. Esta nova resolução foi definida através de um processo exaustivo, onde foi feita uma diminuição gradual da imagem até chegar a um ponto limite, ou seja, a partir desse ponto ficaria difícil até para um ser humano classificar a imagem segmentada.

Outro passo foi executado para extrair a menor região retangular que delimita a área segmentada. Para isso foi produzido um método que percorre a matriz que representa a imagem, e exclui as linhas e colunas que são formadas somente por pixels pretos. O objetivo

desse passo é descartar as informações irrelevantes na imagem. A Figura 19 ilustra o menor retângulo que delimita a área segmentada.



Figura 19 – Imagem com a menor região retangular que delimita a área segmentada.

O contorno da moto é outra informação irrelevante na imagem e também precisou ser excluída, para isso foi definido um recorte na forma de um quadrado para subtrair da imagem a parte superior do motociclista, ou seja, cabeça, ombro e tronco. O tamanho do quadrado foi definido empiricamente, resultando em 20 pixels para cada lado. Esta imagem quadricular é base para a criação do vetor de características, que serve de entrada para o classificador. Na Figura 20, o quadrado amarelo representa a posição do corte na imagem em relação ao retângulo definido no passo anterior.



Figura 20 – O quadrado amarelo representa a posição de corte na imagem.

Neste trabalho são definidos dois modelos de entrada para os classificadores (o modelo A e o modelo B). Estes modelos são representados por um arquivo do tipo csv (“*comma separated values*”). A primeira linha do arquivo contém os nomes dos atributos de entrada [A1, A2,..., AN]. Cada linha subsequente corresponde a uma instância (imagem descrita) de entrada. A última coluna de cada linha contém o valor da classe pertencente à própria instância, [UM] para um passageiro e [DOIS] para dois passageiros. O número total de atributos no modelo depende da forma como ele foi criado. O número de instâncias

depende do número de imagens que serão classificadas. Na Tabela 1 é possível ver como fica a estrutura da descrição da imagem.

| | A1 | A2 | ... | AN | Classe |
|---------------|----|----|-----|----|--------|
| Instância 1 | 0 | 1 | ... | 0 | UM |
| Instância 2 | 1 | 1 | ... | 1 | DOIS |
| Instância ... | 1 | 0 | ... | 1 | UM |
| Instância N | 1 | 1 | ... | 1 | DOIS |

Tabela 1 – Tabela ilustrativa da descrição da imagem no arquivo de entrada csv.

No modelo A, cada instância é composta por quatrocentos (400) atributos que são adquiridos a partir da concatenação das linhas da matriz que representa a imagem. Na **Equação 1** é possível ver a matriz que representa a imagem, e logo em seguida na **Equação 3** é possível observar como ficou o vetor com a concatenação das linhas da matriz.

$$\text{Imagem} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,20} \\ a_{2,1} & a_{2,2} & a_{2,3} & \dots & a_{2,20} \\ a_{3,1} & a_{3,2} & a_{3,3} & \dots & a_{3,20} \\ \dots & \dots & \dots & \dots & \dots \\ a_{20,1} & a_{20,2} & a_{20,3} & \dots & a_{20,20} \end{bmatrix}$$

Equação 1

$$\forall a_{i,j} = \begin{cases} 1, & \text{se o pixel } (i,j) \text{ é branco} \\ 0, & \text{caso contrário} \end{cases}$$

Equação 2

$$\text{Instância A} = \left[a_{1,1} \quad \dots \quad a_{1,20} \quad a_{2,1} \quad \dots \quad a_{2,20} \quad \dots \quad a_{20,1} \quad \dots \quad a_{20,20} \right]$$

Equação 3

No modelo B cada instância é composto por vinte (20) atributos que são adquiridos a partir da soma dos pixels brancos em cada linha. Na **Equação 4** é possível ver como ficou o vetor das somas dos pixels brancos de cada linha.

$$\text{Instância B} = \left[\sum_{j=1}^{20} a_{1,j} \quad \sum_{j=1}^{20} a_{2,j} \quad \sum_{j=1}^{20} a_{3,j} \quad \dots \quad \sum_{j=1}^{20} a_{20,j} \right]$$

Equação 4

3.3 Classificação

A classificação é a etapa final do nosso experimento, nela foram testados dois classificadores: a árvore de decisão *J48* e a rede neural *Multilayer Perceptron*. A ferramenta Weka (WEKA, 2013) foi utilizada, pois fornece a implementação destes dois classificadores, além de fornecer a técnica de *cross-validation* para a classificação do modelo. O *cross-validation* é importante por ter poucos dados. No *cross-validation* os dados são divididos aleatoriamente em n partes (estas partes são chamadas de *folds*). O algoritmo classificador é executado n vezes, em cada execução um *fold* é classificado, enquanto que os $n-1$ *folds* são usados para treinar o classificador. No final deste processo todos os *folds* são classificados uma vez, e usados para treinar o classificador $n-1$ vezes. Neste trabalho o número de *folds* foi fixado em dez. Os resultados da classificação são mostrados na próxima Seção.

4 RESULTADOS EXPERIMENTAIS

Nesta Seção são mostrados os resultados dos dois classificadores atuando em cada modelo proposto. Estes modelos são definidos na Seção 3.2. A rede neural utilizando o modelo B foi o único cenário que forneceu resultados satisfatórios (curva ROC acima de 90%). A seguir são mostrados os detalhes da base de dados e os resultados experimentais de cada classificador utilizado os dois modelos propostos, e os resultados da variação dos parâmetros da rede neural para o modelo B. Para os outros classificadores, os parâmetros padrões do Weka foram mantidos.

4.1 Base de dados

A empresa Fotosensores disponibilizou as imagens para a realização dos experimentos. A base de dados fornecida é composta por cento e noventa e cinco fotos, onde em vinte e oito delas o motociclista aparece incompleto na imagem, isto ocorreu pelo fato de algumas motos

passarem muito acima ou muito abaixo da velocidade prevista. Todas as imagens possuem uma moto com um ou dois passageiros, conforme ilustrado na Figura 21.

Algumas das imagens precisaram ser removidas, pois até para um ser humano é difícil classificá-las corretamente usando somente o contorno. Após esta filtragem, a base de dados ficou composta de cento e sessenta e sete fotos, das quais, dezenove possuem dois passageiros, e cento e quarenta e oito, possuem somente um passageiro (piloto). Na Figura 22 é apresentado um exemplo de foto que precisou ser removida.



Figura 21 – À esquerda, imagem contendo um passageiro. À direita, imagem contendo dois passageiros.



Figura 22 – Exemplo de uma imagem removida. É notória a dificuldade de se identificar o número de passageiro somente com o contorno. Na imagem à esquerda é possível observar que o motociclista está incompleto.

A resolução das imagens é 800x600 pixels. E a intensidade de luminosidade em cada uma delas varia muito, já que foram capturadas em diferentes horários do dia. Embora a resolução seja baixa, a configuração de zoom da câmera permite um nível adequado de detalhes para a extração de contornos.

4.2 Resultados Experimentais para a Árvore de Decisão J48

O classificador *J48* forneceu resultados ruins nos dois modelos testados. O modelo A apresentou uma área ROC resultante de 63,1%. No modelo B a área ROC resultante foi de 71,2%. Neste trabalho é esperada uma área ROC resultante acima de 90% para ser considerada uma boa classificação.

4.2.1 Experimentos com o Modelo A

Os resultados da classificação do modelo A ficaram próximos aos resultados de classificadores aleatórios. Um classificador “aleatório” sorteia a classe de cada instância, onde a probabilidade de sortear uma classe é igual à proporção observada de instâncias desta classe. Portanto, classificadores “aleatórios” possuem área ROC próxima de 50%, e o gráfico ROC se assemelha a uma reta. No caso da árvore de decisão *J48* com modelo A, a área ROC foi de 63,1%, o que consideramos próximo à área do classificador “aleatório”.

A Figura 23 apresenta a saída do Weka para este classificado. Nela é possível ver a matriz de confusão e algumas métricas como: a taxa de verdadeiros positivos (TP Rate), a taxa de falsos positivos (FP Rate), a precisão (Precision), a cobertura (Recall) que é similar a taxa de verdadeiros positivos, a média harmônica da precisão e da cobertura (identificada pelo nome de F-Measure), a área ROC (ROC area), dentre outras. Estas métricas são calculadas a partir do limiar de *score*, que é fixado pelo Weka em 50%. O conceito “limiar de *score*” é apresentado na Seção 2.3.1. Com este resultado podemos observar que houve alta taxa de verdadeiros positivos (TP Rate = 0.959) e uma alta taxa de falsos positivos (FP Rate = 0.737), na classe UM. Isso significa que ele acertou 95,9% das instâncias de sua classe, porém classificou 73,7% de instâncias da outra classe como sua. Na classe DOIS, apesar do classificador ter errado apenas 4,1% das instâncias, ele acertou somente 26,3% de suas instâncias corretamente. Em outras palavras, este classificador não consegue identificar bem instâncias da classe DOIS.

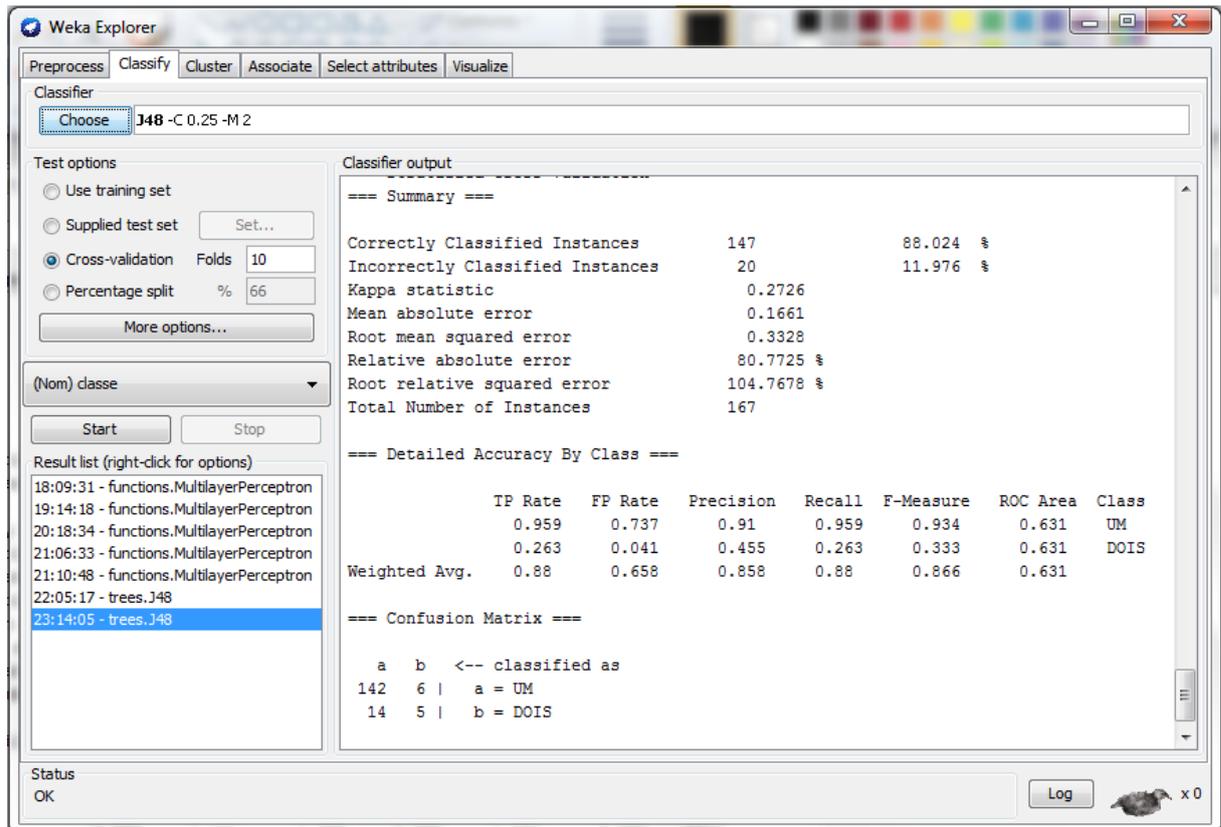


Figura 23 – Sumário com os resultados do classificador no modelo A.

A Figura 24 e a Figura 25 ilustram o gráfico ROC das classes UM e DOIS, respectivamente. Nelas é possível observar que as curvas dos gráficos se aproximam da linha vermelha que representa classificadores aleatórios, e é considerado o pior resultado para um classificador.

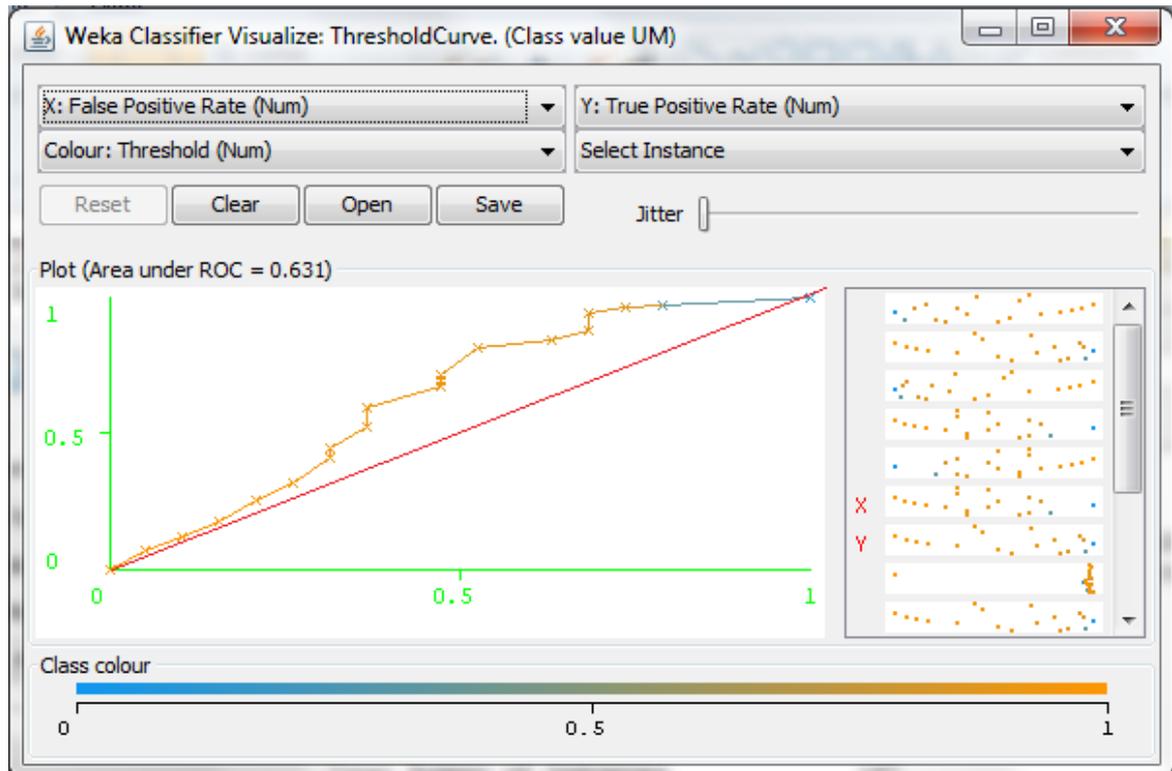


Figura 24 – Gráfico ROC da classe UM.

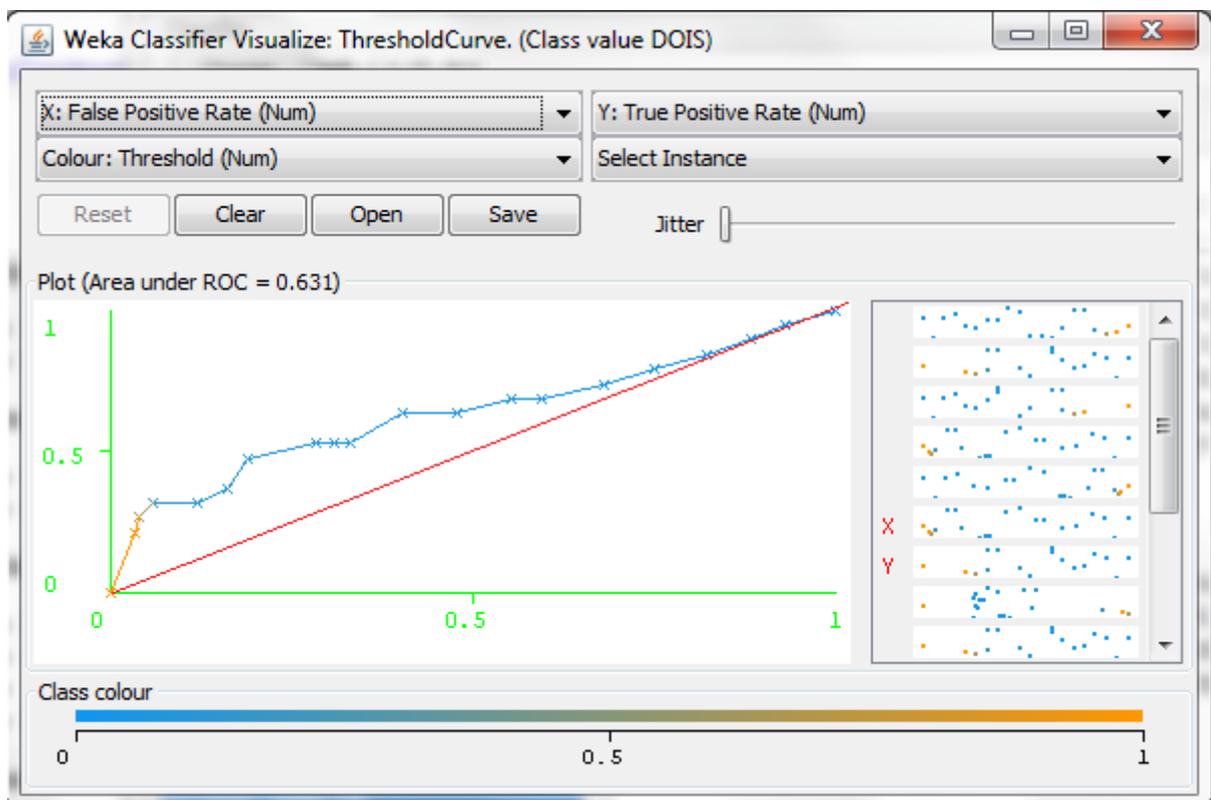


Figura 25 – Gráfico ROC da classe DOIS.

4.2.2 Experimentos com o Modelo B

O modelo B apresentou resultados melhores do que o modelo A, porém ainda está longe de uma boa classificação. A Figura 26 apresenta a saída do classificador no modelo B com um limiar de *score* de 50%. Na classe UM, a taxa de verdadeiros positivos (TP Rate = 0.953) continuou basicamente a mesma, enquanto que a taxa de falsos positivos (FP Rate = 0.474) diminuiu. Já na classe DOIS a taxa de verdadeiros positivos (TP Rate = 0.526) dobrou, e a taxa de falsos positivos (FP Rate = 0.047) continuou também praticamente a mesma. O resultado da área ROC para este modelo foi de 71,2%, ou seja, 18,8% abaixo do esperado.

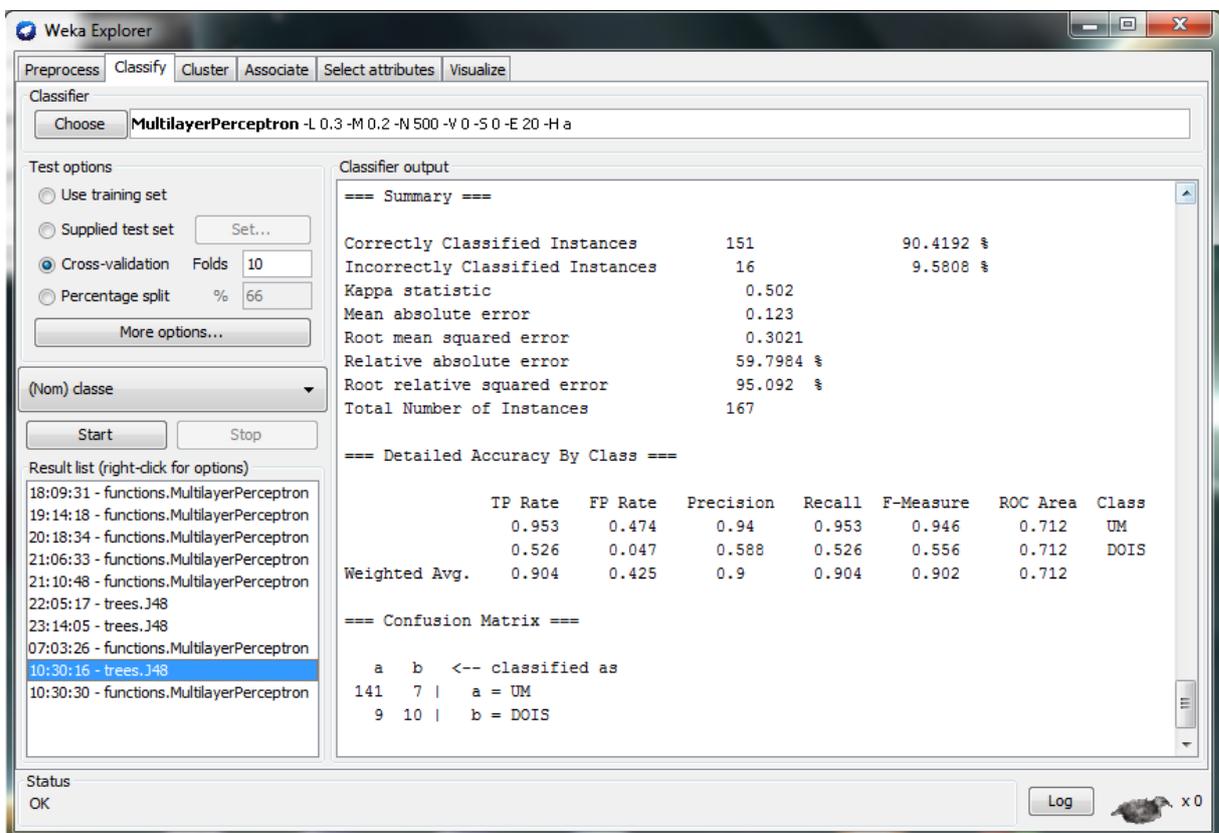


Figura 26 – Sumário com os resultados do classificador no modelo B.

Na Figura 27 e na Figura 28 é apresentado o gráfico ROC das classes UM e DOIS, respectivamente. É fácil perceber que a curva dos gráficos está um pouco mais distante dos classificadores aleatórios do que no modelo A, ainda sim, estão longe de representar uma boa classificação.

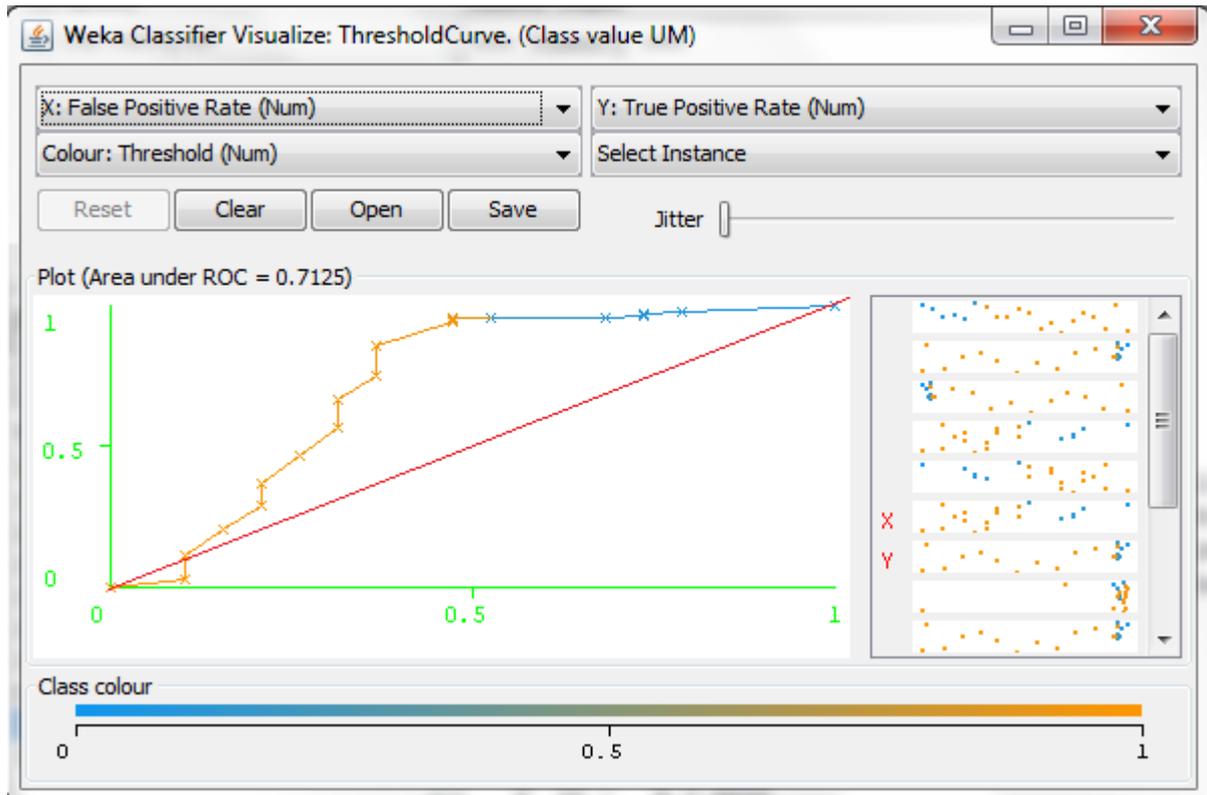


Figura 27 – Gráfico ROC da classe UM.

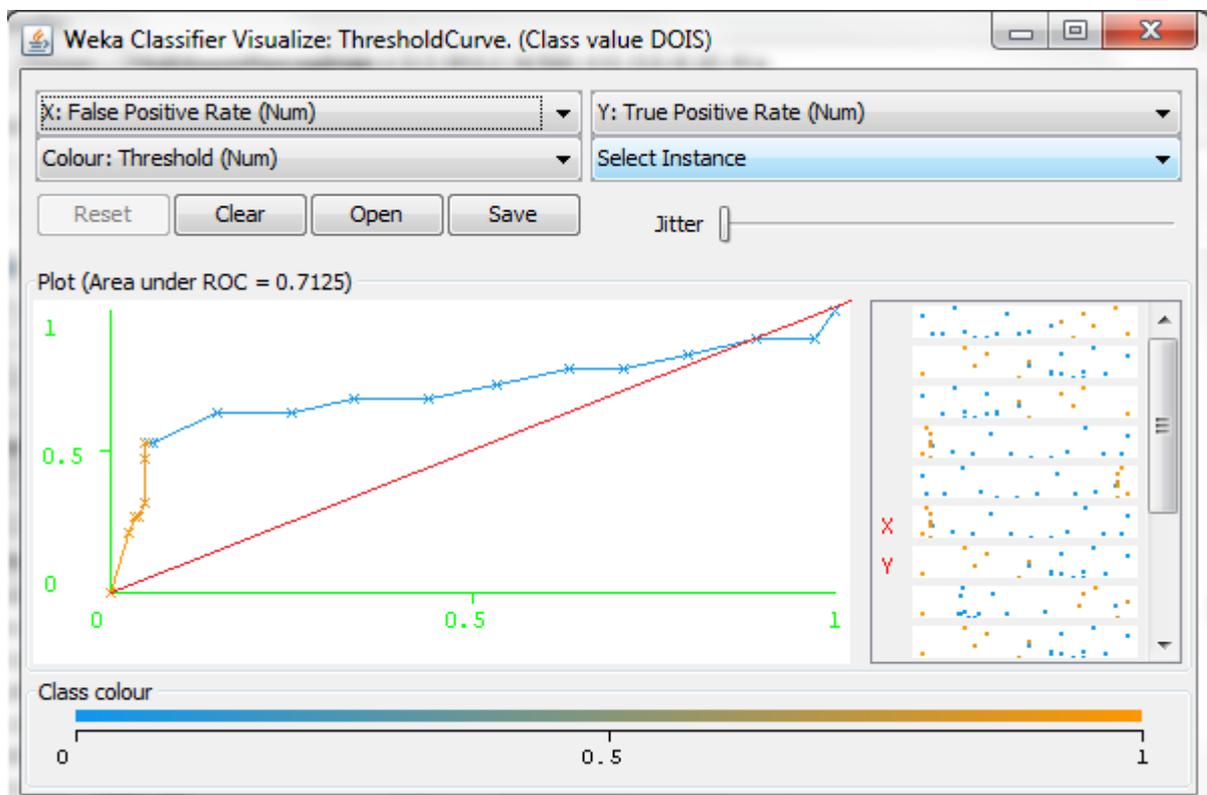


Figura 28 – Gráfico ROC da classe DOIS.

4.3 Resultados Experimentais para a Rede Neural Multilayer Perceptron

Este classificador alcançou área ROC considerada satisfatório quando a região segmentada é descrita através do modelo B, mesmo utilizando os parâmetros *default* do Weka. O mesmo não ocorreu com o modelo A. O resultado da área ROC no modelo B foi de 95%, ou seja, 5% acima do desejado. No modelo A, o resultado da área ROC foi de 61,1%, portanto próxima de classificadores aleatórios.

4.3.1 Experimentos com o Modelo A

A Figura 29 apresenta a saída do classificador, onde é possível observar os resultados deste modelo. Na classe UM, a taxa de falsos positivos (FP Rate = 0.842) foi altíssima, chegando a quase 85% de erro. A taxa de verdadeiros positivos (TP Rate = 0.946) também foi alta. Os resultados mostraram que quase todas as instâncias foram classificadas como pertencentes à classe UM, enquanto que poucas instâncias foram classificadas como da classe DOIS.

The screenshot shows the Weka Explorer interface with the following data:

Classifier output

```

=== Summary ===
Correctly Classified Instances      143      85.6287 %
Incorrectly Classified Instances    24       14.3713 %
Kappa statistic                    0.1272
Mean absolute error                 0.1512
Root mean squared error             0.3654
Relative absolute error             73.5451 %
Root relative squared error        115.0161 %
Total Number of Instances          167
  
```

Detailed Accuracy By Class

| | TP Rate | FP Rate | Precision | Recall | F-Measure | ROC Area | Class |
|---------------|---------|---------|-----------|--------|-----------|----------|-------|
| Weighted Avg. | 0.856 | 0.752 | 0.826 | 0.856 | 0.839 | 0.611 | |
| | 0.946 | 0.842 | 0.897 | 0.946 | 0.921 | 0.611 | UM |
| | 0.158 | 0.054 | 0.273 | 0.158 | 0.2 | 0.611 | DOIS |

Confusion Matrix

```

=== Confusion Matrix ===
 a  b  <-- classified as
140  8 | a = UM
 16  3 | b = DOIS
  
```

Figura 29 – Sumário com os resultados do classificador no modelo A.

A Figura 30 e a Figura 31 ilustram o gráfico ROC para as classes UM e DOIS. É possível observar que a curva dos gráficos chega bem próxima a linha vermelha que, como discutido anteriormente, representa classificadores aleatórios.

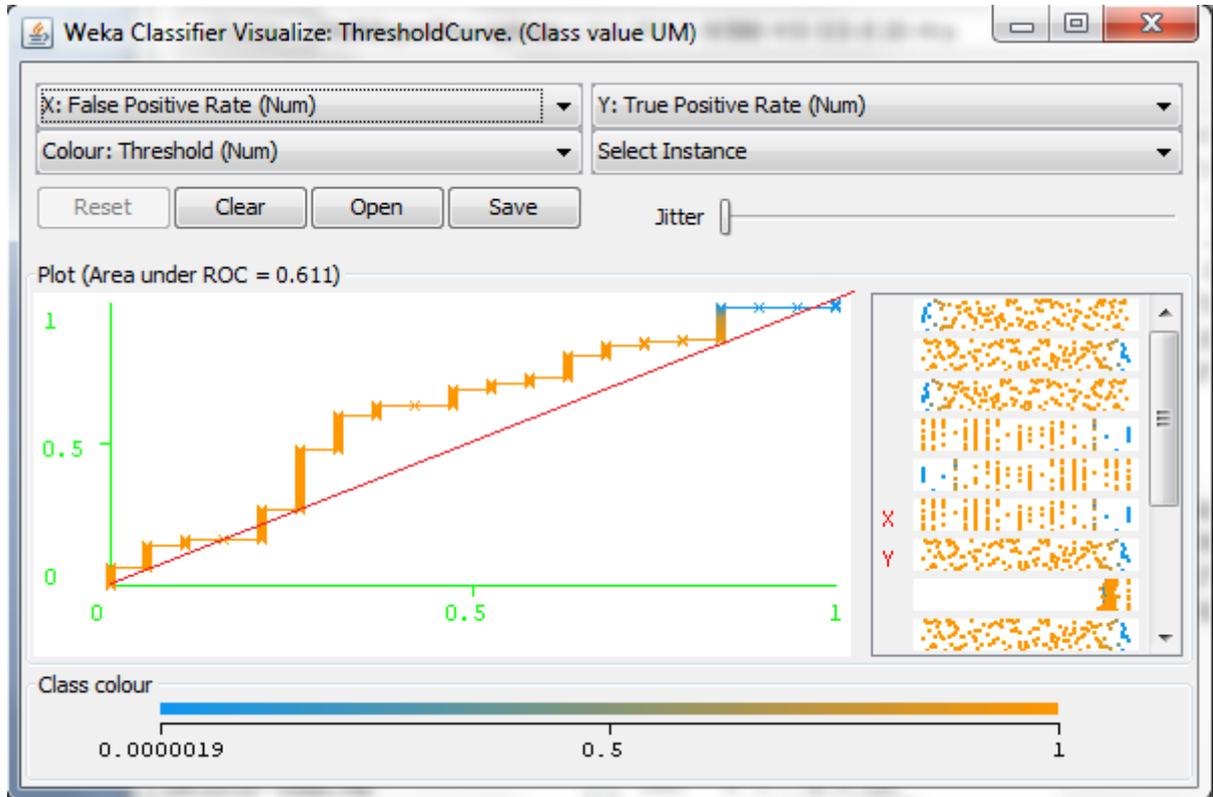


Figura 30 – Gráfico ROC da classe UM.

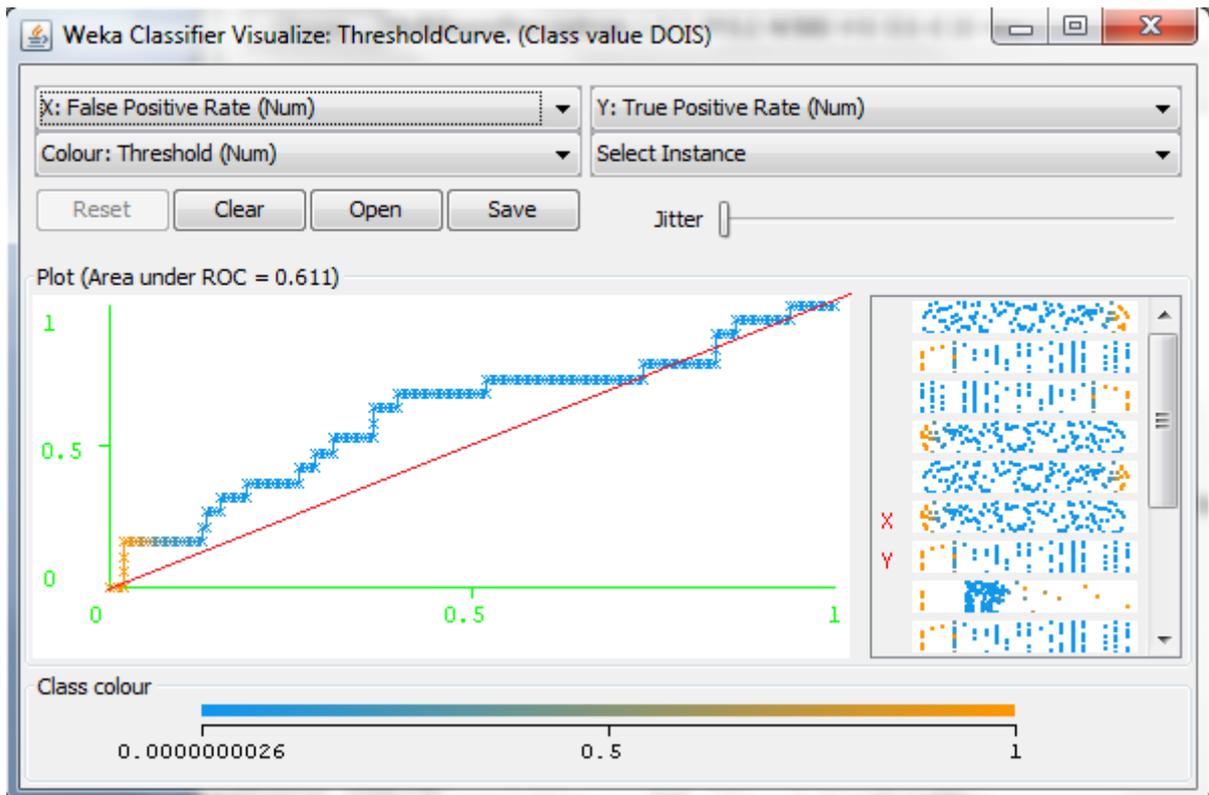


Figura 31 – Gráfico ROC da classe DOIS.

4.3.2 Experimento com o Modelo B

A Figura 32 apresenta a saída do classificador, onde é possível observar os resultados deste modelo. A classe UM teve uma alta taxa de verdadeiros positivos (TP Rate = 0.953) e uma baixa taxa de falsos positivos (FP Rate = 0.211), esse resultado é o que se busca dentro da classificação. A classe DOIS também teve alta taxa de verdadeiros positivos (TP Rate = 0.789) e uma baixíssima taxa de falsos positivos (FP Rate 0.047). A área ROC resultante deste modelo foi de 95%, ou seja, dentro da região desejada.

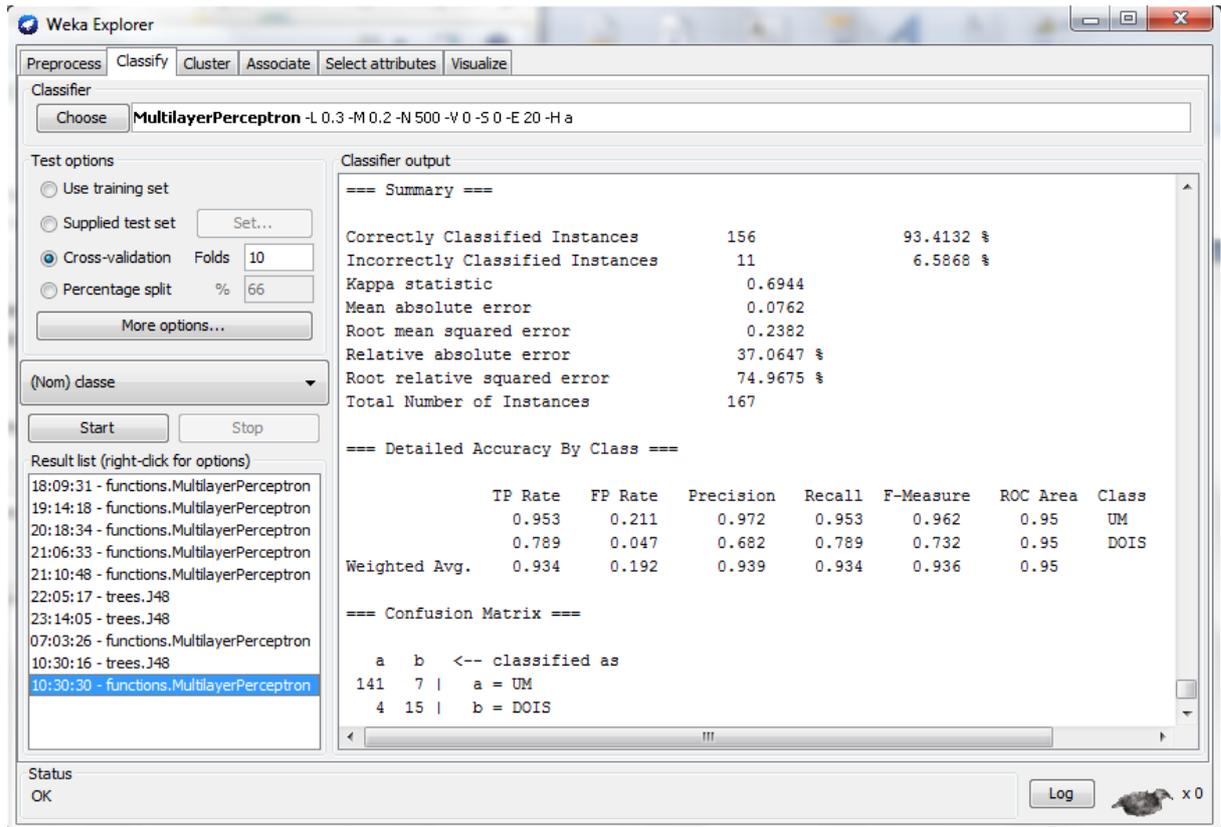


Figura 32 – Sumário com os resultados do classificador no modelo B.

A Figura 34 e a Figura 35 apresentam o gráfico ROC para as classes UM e DOIS. É possível observar que a curva de cada gráfico fica bem distante da linha vermelha. A área ROC é 95%.

Nem sempre o limiar de 50% é a melhor escolha para uma classificação final. Os critérios de escolha para uma classificação final vão depender dos requisitos de cada aplicação. Por exemplo, se o problema deste trabalho tivesse o seguinte requisito: alta taxa de acerto da classe A e erro zero da classe B. O limiar que teria o melhor resultado e atenderia a este requisito é 99.91%, ele está marcado de preto na Figura 34. Como é possível observar na Figura 33, a taxa de acerto nesse ponto é de aproximadamente 72,3% a custo de zero erros. O que podemos considerar que é um ótimo resultado.

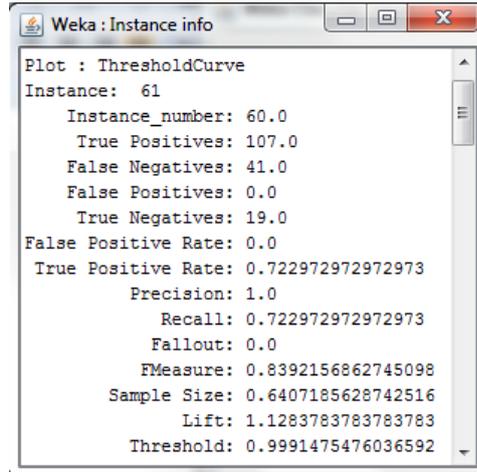


Figura 33 – Detalhes da classificação em um ponto demarcado.

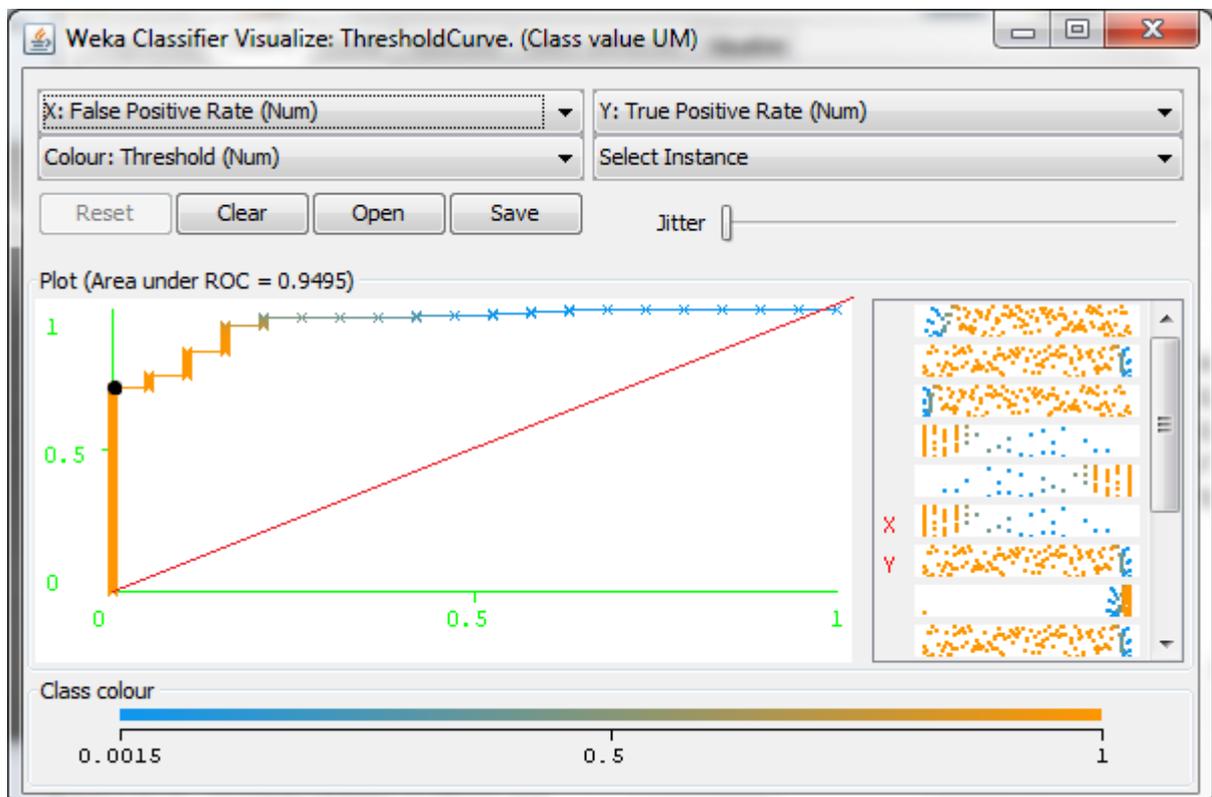


Figura 34 – Gráfico ROC da classe UM.

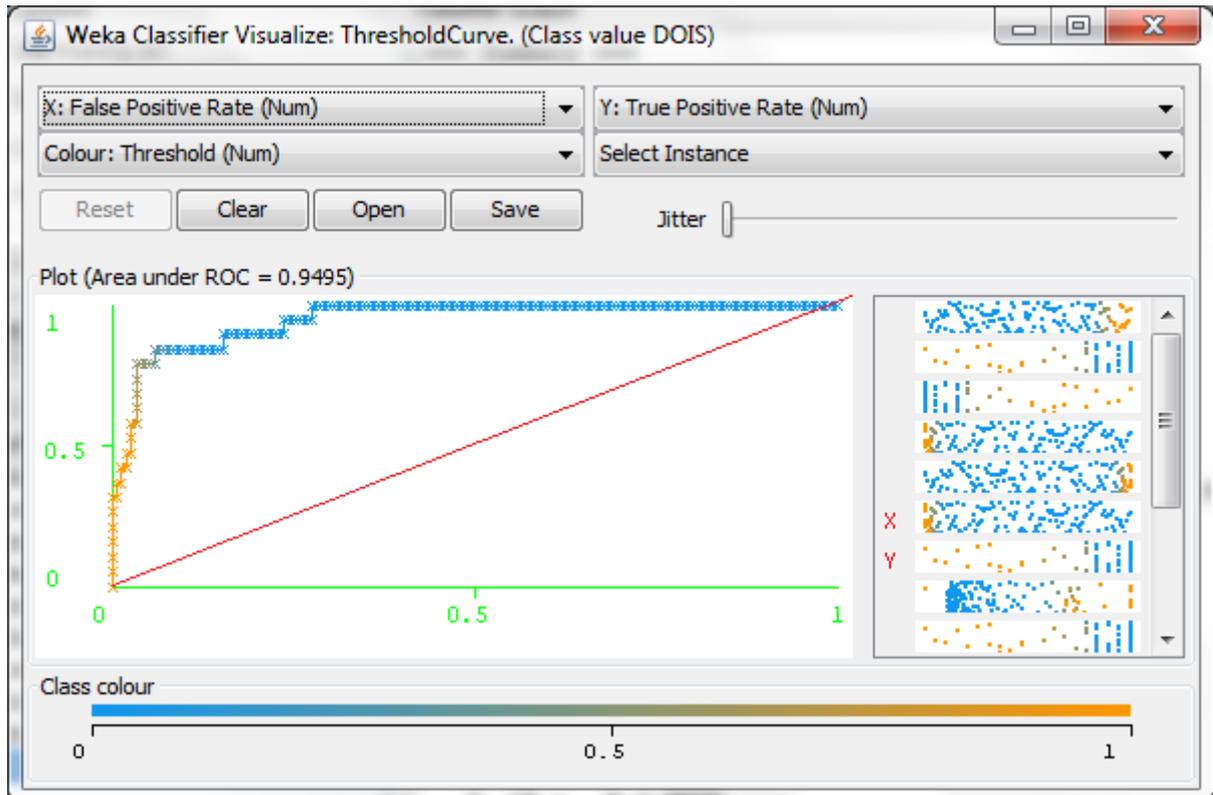


Figura 35 – Gráfico ROC da classe DOIS.

4.4 Mudança nos Parâmetros da Rede Neural para o Modelo B

Para tentar obter resultados ainda melhores para a Rede Neural no modelo B, variamos um dos parâmetros do Weka buscando aquele que oferece melhores resultados. O número de camadas ocultas na rede neural foi o parâmetro modificado neste teste. Foram testadas redes neurais compostas de uma a doze camadas ocultas, e foi concluído que todas essas validam a hipótese levantada neste trabalho. A rede neural de nove camadas ocultas foi a que apresentou o melhor resultado dentre elas, com área ROC resultante de 95,4%, ou seja, 4,6% abaixo do resultado ótimo.

A Tabela 2 apresenta em cada linha o resultado da classificação, variando o número de camadas ocultas presente na rede neural. Os resultados desta tabela são referentes à classe UM.

Classe UM

| Nº Camadas | TP Rate | FP Rate | Precision | Recall | F-Measure | ROC Area |
|------------|---------|---------|--------------|--------------|--------------|--------------|
| 1 | 0.953 | 0.263 | 0.966 | 0.953 | 0.959 | 0.91 |
| 2 | 0.959 | 0.316 | 0.959 | 0.959 | 0.959 | 0.931 |
| 3 | 0.939 | 0.263 | 0.965 | 0.939 | 0.952 | 0.939 |
| 4 | 0.966 | 0.368 | 0.953 | 0.966 | 0.96 | 0.938 |
| 5 | 0.953 | 0.211 | 0.972 | 0.953 | 0.962 | 0.942 |
| 6 | 0.966 | 0.263 | 0.966 | 0.966 | 0.966 | 0.94 |
| 7 | 0.959 | 0.263 | 0.966 | 0.959 | 0.963 | 0.941 |
| 8 | 0.959 | 0.211 | 0.973 | 0.959 | 0.966 | 0.949 |
| 9 | 0.959 | 0.211 | 0.973 | 0.959 | 0.966 | 0.954 |
| 10 | 0.966 | 0.211 | 0.973 | 0.966 | 0.969 | 0.949 |
| 11 | 0.953 | 0.211 | 0.972 | 0.953 | 0.962 | 0.95 |
| 12 | 0.959 | 0.211 | 0.973 | 0.959 | 0.966 | 0.95 |

Tabela 2 – Tabela com resultados dos diferentes parâmetros para a classe UM. Em negrito, os melhores resultados para cada métrica.

A Tabela 3 apresenta os mesmo resultados, porém para a classe DOIS.

Classe DOIS

| Nº Camadas | TP Rate | FP Rate | Precision | Recall | F-Measure | ROC Area |
|------------|---------|---------|-------------|--------------|--------------|--------------|
| 1 | 0.737 | 0.047 | 0.667 | 0.737 | 0.7 | 0.91 |
| 2 | 0.684 | 0.041 | 0.684 | 0.684 | 0.684 | 0.931 |
| 3 | 0.737 | 0.061 | 0.609 | 0.737 | 0.667 | 0.939 |
| 4 | 0.632 | 0.034 | 0.706 | 0.632 | 0.667 | 0.938 |
| 5 | 0.789 | 0.047 | 0.682 | 0.789 | 0.732 | 0.942 |
| 6 | 0.737 | 0.034 | 0.737 | 0.737 | 0.737 | 0.94 |
| 7 | 0.737 | 0.041 | 0.7 | 0.737 | 0.718 | 0.941 |
| 8 | 0.789 | 0.041 | 0.714 | 0.789 | 0.75 | 0.949 |
| 9 | 0.789 | 0.041 | 0.714 | 0.789 | 0.75 | 0.954 |
| 10 | 0.789 | 0.034 | 0.75 | 0.789 | 0.769 | 0.949 |
| 11 | 0.789 | 0.047 | 0.682 | 0.789 | 0.732 | 0.95 |
| 12 | 0.789 | 0.041 | 0.714 | 0.789 | 0.75 | 0.95 |

Tabela 3 - Tabela com resultados dos diferentes parâmetros para a classe DOIS. Em negrito, os melhores resultados para cada métrica.

5 CONSIDERAÇÕES FINAIS

O objetivo desse trabalho era produzir um estudo preliminar que fosse capaz de responder a uma hipótese levantada “é possível determinar o número de passageiros em motos, com uma taxa de erro aceitável (área ROC acima de 90%), utilizando apenas informação de contorno da moto e dos passageiros”. Ao final deste trabalho foi visto que o objetivo foi atingido. Foram testados quatro cenários, e somente em um deles, o resultado validou a hipótese. O cenário ideal para este problema é utilizar uma rede neural *Multilayer Perceptron* e classificar os dados utilizando o modelo B que foi definido na Seção 3.2

A ferramenta Weka e a biblioteca OpenCV, ajudaram bastante na realização deste trabalho. Se fosse implementado tudo desde início, não haveria tempo de escrever o experimento.

Em um estudo inicial realizado antes deste trabalho, foi tentado trabalhar com um descritor bastante conhecido no estado da arte, o HOG (*Histogram of Oriented Gradients*). Ele é um descritor totalmente novo e está sendo usado com muita frequência para a detecção de pessoas. O objetivo de usar esse descritor era conseguir encontrar nas imagens de trânsito o número de motociclistas sem a necessidade de uma etapa de segmentação automática, já que foi mostrado que não era possível fazer isto por conta da base de dados ser composta somente por fotos. A grande dificuldade encontrada nessa abordagem foi conseguir executar as ferramentas presentes na literatura. Como o assunto é novo, a documentação é bastante precária. Esta abordagem ficou como um trabalho futuro.

Outra idéia para um trabalho futuro é criar novos modelos de descrição. O modelo B é composto pela soma das linhas da matriz que representa a imagem. Um modelo C poderia ser estendido para coletar além da soma das linhas, a soma das colunas também.

REFERÊNCIAS

AFFONSO, E. T. F.; SILVA, A. M.; SILVA, M. P.; RODRIGUES, T. M. D.; MOITA, G. F. Uso Redes Neurais Multilayer Perceptron (MLP) em Sistema de Bloqueio de Websites Baseado em Conteúdo, 2010.

ARAÚJO, S. A. Casamento de Padrões em Imagens Digitais Livre de Segmentação e Invariante sob Transformações de Similaridade, 2009.

BEUCHER, S.; LANTUÉJOUL, C. Use of Watersheds in Contour Detection. In international Workshop on Image Processing, p. 2.1–2.12, Rennes. CCETT/IRISA, 1979.

BRAGA, A. C. S. Curvas Roc: Aspectos Funcionais e Aplicações, 2000.

CYBENCO, G. Approximation by superposition of a sigmoidal function. Mathematics of control. Signals and Systems 2, 303–314, 1989.

CONCIN, A.; AZEVEDO, E.; LETA, F. R. Computação Gráfica. 1. ed. Rio de Janeiro: Elsevier, 2008.

CREPALDI, P. G.; AVILA, R. N. P.; OLIVEIRA, J. P. N.; RODRIGUES, P. R.; MARTINS R. L. Um Estudo Sobre a Árvore de Decisão e sua Importância na Habilidade de Aprendizado, 2010.

DA SILVA, F. C. Análise ROC, 2006.

DALLAGASSA M. R.; DA SILVA, S. H.; CARVALHO D. R. Avaliação de Modelos Para a Classificação De Beneficiários Com Indicativos para o Diabetes Mellitus Tipo 2, 2011.

DE MILANO, D.; HONORATO, L. B. Visão Computacional, 2010.

DIGABEL, H.; LANTUÉJOUL, C. Iterative Algorithms. In Chermant, J.-L., editor, Quantitative Analysis of Microstructures in Materials Sciences, biology and medicine, pages 85–99, Stuttgart. Dr. Riederer-Verlag GmbH, 1978.

ESCAÑO, A. Fundamentos da Imagem digital, 1999.

FERREIRA, C. S. Implementação do Algoritmo de Subtração de Fundo para Detecção de Objetos em Movimento, Usando Sistemas Reconfiguráveis, 2012.

FOTOSENSORES TECNOLOGIA ELETRÔNICA LTDA.. Sistema de Monitoramento de Trânsito com Câmera Digital (SMTCD). Disponível em: www.fotosensores.com/SMTCD. Acessado em: 26/05/2012.

GARCIA, S. C. O uso de árvores de decisão na descoberta de conhecimento na área da saúde, 2000.

GONZALEZ, R.; WOODS, R. Digital Image Processing, 3rd ed. Prentice Hall, 2007.

HAMILTON, S.; RIBEIRO, R. Processamento de Imagens, 2008.

HAYKIN, S. Neural Networks: A Comprehensive Foundation. IEEE Press, New York, NY, USA, 1994.

HAYKIN, S. Neural Networks. A Comprehensive Foundation. Prentice Hall. 2nd edition, 1999.

HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. Neural Networks 2, 359–366, 1989.

HUTCHINSON, S.; HAGER, G. D.; CORKE, P. I. A Tutorial on Visual Servo Control, 1996.

KAEWTRAKULPONG P.; BOWDEN R. An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection, 2001.

KISI, O. Multi-layer perceptrons with Levenberg Marquardt training algorithm for suspended sediment concentration prediction and estimation, 2005.

LOESCH, C.; SARI, S. T. Redes neurais artificiais: fundamentos e modelos. Editora da FURB, Blumenau, 1996.

MARENGONI, M.; STRINGHINI, D. Tutorial: Introdução à Visão Computacional Usando OpenCV, 2009.

MARGOTTO P. R. Curva roc: Como fazer e interpretar no SPSS, 2010.

MEYER, F. Color Image Segmentation, ICIP92, 1992

NEVES S. R. Algoritmos Para Segmentação De Imagens Infravermelhas, 2003

OLIVEIRA, A. C. F. Sistema de Monitorização de Espaços Urbanos baseado em Visão por Computador, 2010.

PAIVA, C. Introdução a técnicas de Segmentação, 2010.

PALOMINO N. L. A.; CONCEPCIÓN L. P. Watershed: Un Algoritmo Eficiente y Flexible para Segmentación de Imágenes de Geles 2-DE, 2010.

PECCINI G.; ORNELLAS M. C. Segmentação de Imagens por Watersheds: Uma Implementação Utilizando a Linguagem Java, 2004.

PIZARRO, P.; COMUNELLO, E.; MANTELLI, S. Segmentações por Crescimento de Regiões.

PRATI, R. C.; BATISTA, G. E. A. P. A.; MONARD, M. C. Curvas ROC para avaliação de classificadores, 2008.

RODRIGUES, F. A. Localização e Reconhecimento de Placas de Sinalização Utilizando um Mecanismo de Atenção Visual e Redes Neurais Artificiais, 2002.

RUSS, J. C. The Image Processing Handbook. CRC Press, Boca Raton, 1998.

SANTOS, M. S.; NETO, J. C. C. Árvore de Decisão para Classificação de Ocorrências de Dengue nos Municípios de Ilhéus e Itabuna, 2010.

TAFNER, M. A.; XEREZ, M.; FILHO, I. W. R. Redes Neurais Artificiais: Introdução e Princípios de Neurocomputação. Blumenau: EKO: Editora da FURB, 11º edição, 1995.

TATIBANA, C. Y.; KAETSU, D. Y. Uma Introdução às Redes Neurais. Disponível em: <<http://www.din.uem.br/ia/neurais/#links>>. Acessado em: 31/01/2013.

TEIXEIRA, A. S. B. Desenvolvimento De Uma Interface Gráfica Para Classificadores De Imagem, 2011.

WEKA. Machine Learning. McGraw-Hill, 1997. Disponível em: <<http://www.cs.waikato.ac.nz/~ml/weka/>>. Acesso em: 15/02/2013.