Universidade Federal de Pernambuco
Centro de Informática

Mestrado em Ciência da Computação

Segmentação e Classificação de Padrões Visuais Baseadas em Campos Receptivos e Inibitórios

Bruno José Torres Fernandes

Dissertação de Mestrado

Recife
20 de janeiro de 2009

Universidade Federal de Pernambuco
Centro de Informática

Bruno José Torres Fernandes

Segmentação e Classificação de Padrões Visuais Baseadas em Campos Receptivos e Inibitórios

Trabalho apresentado ao Programa de Mestrado em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: *Prof. Dr. George Darmiton da Cunha Cavalcanti*

Recife
20 de janeiro de 2009

Fernandes, Bruno José Torres
Segmentação e classificação de padrões visuais baseadas em campos receptivos e inibitórios / Bruno José Torres Fernandes. – Recife : O Autor, 2009.
xiii, 75 folhas ; il., quadros., fig., tab., gráf.

Dissertação (mestrado) – Universidade Federal de Pernambuco. CIn. Ciência da Computação, 2009.

Inclui bibliografia e apêndice.

1. Inteligência artificial. 2. Processamento de imagens. 3. Redes neurais (computação). I. Título.

006.3 CDD (22.ed.) MEI 2009-003

Dissertação de Mestrado apresentada por **Bruno José Torres Fernandes** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título **"Segmentação e Classificação de Padrões Visuais Baseados em Campos Receptivos e Inibitórios"**, orientada pelo **Prof. George Darmiton da Cunha Cavalcanti** e aprovada pela Banca Examinadora formada pelos professores:

Prof. Tsang Ing Ren
Centro de Informática / UFPE

Prof. Herman Martins Gomes
Departamento de Sistemas e Computação / UFCG

Prof. George Darmiton da Cunha Cavalcanti
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 20 de janeiro de 2009.

p/ **Prof. FRANCISCO DE ASSIS TENÓRIO DE CARVALHO**
Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

Prof. Nelson Souto Rosa
Vice-Coordenador da Pós-Graduação em
Ciência da Computação/UFPE

Eu dedico este trabalho a todos que em mim acreditaram.

Agradecimentos

Agradeço, primeiramente, a Deus e a Nossa Senhora por estarem sempre iluminando o meu caminho.

Agradeço a minha família (meus pais, Sérgio e Thelma, meus irmãos, Serginho e Mariana, minha tia Tânia e todos os agregados e demais membros da família) pelo apoio, conselhos e, acima de tudo, pelo suporte que me deram nas fases mais difíceis da minha vida e do meu mestrado.

Agradeço a minha namorada, Dani, por todo o amor e paciência que teve comigo e por estar sempre me alegrando nos momentos em que precisava. Também agradeço a Dani por ter me ajudado na correção deste trabalho.

Agradeço também a meu orientador, George, e ao professor Tsang por me darem toda a ajuda necessária para que eu construísse o meu mestrado da melhor forma possível.

Por fim, agradeço a todos os amigos da Provider Sistemas e do Centro de Informática, que direta ou indiretamente também contribuíram para a realização deste trabalho.

We can only see a short distance ahead, but we can see plenty there that needs to be done.

—ALAN TURING (Pai da computação moderna, 1912-1954)

Resumo

O sistema visual humano é um dos mecanismos mais fascinantes da natureza. É através dele que o ser humano é capaz de realizar as suas tarefas mais básicas, como assistir televisão, até as mais complexas, como realizar análises através de microscópios em laboratórios. Por conseguinte, neste trabalho são propostos dois modelos baseados no comportamento do sistema visual humano. O primeiro é um modelo de segmentação supervisionada baseado nos conceitos de campos receptivos, chamado *Segmentation and Classification Based on Receptive Fields* (SCRF). O outro é uma nova rede neural, chamada I-PyraNet. A I-PyraNet é uma implementação híbrida da PyraNet e dos conceitos de campos inibitórios.

Então, no intuito de validar os modelos aqui propostos, nesta dissertação é apresentada uma revisão do estado-da-arte, descrevendo-se desde o funcionamento do sistema visual humano até as várias etapas existentes numa tarefa de processamento de imagens.

Por fim, os modelos propostos foram aplicados em duas tarefas de reconhecimento. O modelo SCRF e a I-PyraNet foram aplicados juntos num problema de detecção de floresta em imagens de satélite. Enquanto a I-PyraNet foi aplicada sobre um problema de detecção de facos. Ambos alcançaram bons resultados quando comparados aos outros modelos aqui apresentados.

Palavras-chave: Segmentação de imagens, classificação de imagens, detecção de faces, campos receptivos e inibitórios, redes neurais.

Abstract

The human visual system is one of the most fascinating mechanisms from nature. It is through the visual system that humans are able to accomplish their most basic tasks, like watching TV, until the most complexes ones, like making microscopic analysis in laboratories. Then, two models based on the behavior of the human visual system are proposed in this work. The first one is a supervised segmentation model based on the concepts of receptive fields, called *Segmentation and Classification Based on Receptive Fields* (SCRF). The other one is a new neural network, called I-PyraNet. The I-PyraNet is a hybrid implementation between the PyraNet and the concepts of inhibitory fields.

Therefore, aiming at the validation of the models here proposed, in this dissertation a wide revision of the state-of-the-art is presented, describing aspects that vary from the way that human visual system works to the many steps existent in a image processing task.

Finally, the models proposed in this work were applied to recognition tasks and obtained excellent results when compared to the other models presented in this work.

Keywords: Image segmentation, image classification, face detection, receptive and inhibitory fields, neural network.

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

Introdução

1.1 Motivação

O olho constitui um dos órgãos mais importantes do corpo humano. De maneira simplória, pode-se dizer que o olho é o órgão dos animais que permite detectar a luz e transformar essa percepção em impulsos elétricos. Contudo, é a partir de tal percepção que somos capazes de distinguir elementos, encontrar padrões, reconhecê-los, interpretá-los e, principalmente, adquirir rapidamente informações do mundo externo.

A área de processamento e análise de imagens (Gonzalez e Woods 2007) vem tentando, com cada vez mais eficácia, reproduzir em um computador os resultados advindos do sistema visual humano. Nesta busca, é facilmente perceptível os ganhos obtidos pela utilização de redes neurais na simulação do cérebro humano. Não obstante, destaca-se o maior aprendizado obtido sobre o funcionamento do córtex visual do cérebro humano, o qual permite a criação de modelos capazes de replicar em parte seu funcionamento. Aplicações como reconhecimento de caracteres manuscritos, detecção e reconhecimento de faces, dentre outras tarefas fáceis de serem realizadas pelo Homem, ainda são alvos constantes de pesquisas na área. Contudo, apesar do homem ser capaz de realizar tais tarefas, ele não consegue fazê-las de uma forma exaustiva, o que não é um entrave para a máquina que é capaz de realizar a mesma tarefa indefinidamente.

Como exemplo de um importante conceito presente no sistema visual humano, foi apontada a presença de neurônios conectados a regiões específicas, denominadas campos receptivos (Hubel 1963). Tal conceito foi aplicado nas mais diversas técnicas, motivando, por exemplo, o filtro 2-D de Gabor (Jones e Palmer 1987) e a PyraNet (Phung e Bouzerdoum 2007), sendo esta uma rede neural com excelente capacidade de reconhecimento.

Uma outra tarefa muito importante realizada pelo sistema visual humano é a de segmentação de imagens que consiste no problema de particionar um conjunto de dados em um certo número de segmentos ou grupos. Seu objetivo básico é separar as partes mais relavantes dentro de uma imagem. A maioria dos trabalhos desenvolvidos nessa área utilizam uma abordagem não-supervisionada para realizar a segmentação, mas se a intenção é reconhecer objetos den-

tro de uma figura, esse tipo de segmentação implica necessariamente uma posterior análise da imagem para identificar cada grupo. O ser humano pode conduzir essa tarefa sem mais complicações devido à rica e eficaz análise de textura que ele realiza. A análise de textura é uma tarefa de suma importância para o homem, uma vez que a mesma permite mais facilmente que seja realizada a distinção entre os diferentes padrões. Inúmeros são os trabalhos destinados a análise de textura, contudo pode-se dizer que nenhum conseguiu simular o mecanismo biológico, ainda.

Desse modo, esta dissertação foca-se no desenvolvimento e na apresentação de técnicas que sejam biologicamente motivadas para realização de tarefas de segmentação e classificação de imagens. Tais técnicas devem ser aplicadas em diferentes estudos de caso, de modo a demonstrar os ganhos e as desvantagens obtidas por cada uma em comparação a outras técnicas já consolidadas.

1.2 Objetivos

Os objetivos desta dissertação são:

- Descrever de forma suscinta o sistema visual humano;
- Analisar o estado-da-arte de processamento de imagens, com suas diferentes técnicas e aplicações, no que concerne aos modelos propostos neste trabalho e aos experimentos realizados;
- Apresentar um novo modelo de segmentação de imagens baseado nos conceitos de campos receptivos;
- Apresentar uma nova rede neural baseada nos conceitos de campos receptivos e inibitórios;
- Realizar experimentos que demonstrem as vantagens e desvantagens de cada uma das técnicas estudadas.

1.3 Estrutura da Dissertação

Esta dissertação apresenta dois novos modelos para realização de tarefas de segmentação e classifição de imagens. Ambos modelos são biologicamente motivados e fazem uso dos conceitos de campos receptivos.

No Capítulo 2 é apresentado o sistema visual humano. Nele são descritas a estrutura do olho humano e o modelo do sistema visual. Por fim, também é apresentado o conceito de campos receptivos e inibitórios.

No Capítulo 3 é introduzido o estado-da-arte de processamento de imagens. Nele são apresentadas as tarefas de pré-processamento de imagens, análise de textura, segmentação de imagens, classificação de imagens e pós-processamento de imagens. Técnicas importantes de cada tópico são revisadas e utilizadas nos experimentos realizados no Capítulo 5.

No Capítulo 4 são apresentados os modelos propostos neste trabalho. Primeiro, é descrito um novo modelo de segmentação baseado no conceito de campos receptivos, denominado *Segmentation and Classification Based on Receptive Fields* (SCRF). Em seguida é apresentada uma nova rede neural, a I-PyraNet, que é uma combinação da PyraNet com os conceitos de campos inibitórios.

No Capítulo 5 são apresentados os experimentos com os modelos propostos em comparação com outras técnicas de processamento de imagens. Tais experimentos foram realizados sobre as tarefas de detecção de floresta em imagens de satélite e de detecção de faces. Nesse capítulo, quatro bancos de dados são apresentados e os experimentos sobre cada um deles são descritos.

Finalmente, no Capítulo 6 são apresentadas as conclusões obtidas acerca desta dissertação.

CAPÍTULO 2

O Sistema Visual Humano

2.1 Estrutura do Olho Humano

O sistema visual do ser humano é composto, principalmente, por dois olhos e pelo córtex visual. Um olho humano mede em torno de 20 milímetros de diâmetro e é o responsável por captar a luz refletida pelos objetos a sua volta e transimitir tal informação ao córtex visual no cérebro (Gonzalez e Woods 2007, Lim 1990).

Na Figura 2.1 é apresentado um desenho que representa um corte horizontal do olho humano. À frente do olho está a córnea, um tecido resistente e transparente que tem por função principal refratar a luz. Ela atua como a lente de uma câmera.

Por trás da córnea, dentro da câmara anterior, encontra-se o humor aquoso, que se trata de uma substância semi-líquida e transparente. A pressão externa que o humor aquoso exerce faz com que a córnea se torne protuberante. Atravessando o humor aquoso, encontra-se a íris que possui uma pequena abertura circular no centro conhecida por pupila. Essa abertura central contrai-se ou expande-se de forma a controlar a quantidade de luz que entra no olho.

Atrás da íris está o cristalino que é formado por camadas concêntricas de células fibrosas encapsuladas numa membrana transparente e elástica. Sua principal função é focar com precisão a luz entrante numa tela na parte traseira do olho chamada de retina. De modo a conseguir focar objetos perto e longe em diferentes momentos, a forma do cristalino muda, sendo tal processo conhecido por acomodação. A acomodação é controlada pelo corpo ciliar, um grupo de músculos que cercam o cristalino.

Por trás do cristalino encontra-se o humor vítreo que é similar ao humor aquoso. Sua principal função é sustentar a forma do olho. Por trás dele está a retina, que é a tela onde a luz entrante é focada e as células fotossensoras convertem a luz em sinais neurais.

Existem dois tipos de células fotossensoras na retina, elas são chamadas de cones e bastonetes devido às suas formas. Os cones não são muito sensíveis a luminosidade e são responsáveis pela visão colorida. Os bastonetes são mais sensíveis a luz e proporcionam a visão noturna. Na fóvea, uma depressão na retina, concetram-se a maioria dos cones. Nesta pequena região não há a presença bastonetes.

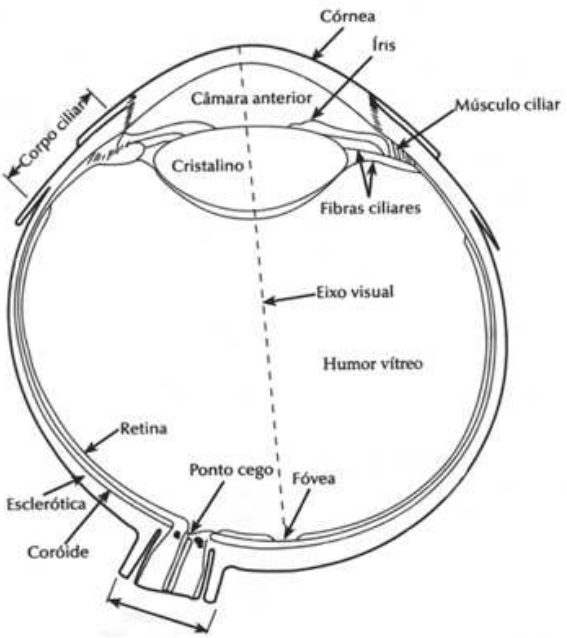


Figura 2.1 Corte horizontal do olho humano.

Quando a luz atinge os cones e os bastonetes uma complexa reação eletro-química ocorre e a luz é convertida em impulsos neurais que são transmitidos para o cérebro por nervos ópticos encontrados no ponto cego do olho.

2.2 Modelo do Sistema Visual

O sistema visual humano pode ser visto como dois sistemas ligados em cascata. O primeiro, representando o nível periférico, é o responsável por converter a luz em sinais neurais. O segundo, representando o nível central, processa o sinal neural para extrair informação.

O nível periférico é bem entendido e um modelo simples de imagens monocromáticas que pode descrevê-lo foi proposto por Stockham (Stockham 1972). Nele, a intensidade da imagem é modificada por uma função não-linear, como a operação logarítmica. O resultado dessa operação é então filtrado por um sistema linear invariante ao deslocamento, motivado pelo tamanho finito da pupila e pelo processo de inibição lateral. Então, a resposta da fibra neural, conectada

ao olho e responsável por conduzir a informação visual ao cérebro, é uma combinação entre os sinais dos cones e dos bastonetes, uns contribuindo positivamente e outros negativamente. É importante notar que apesar de existerem 130 milhões de células fotossensoras (cones e bastonetes), existem apenas 1 milhão de fibras nervosas, sendo que algumas fibras nervosas servem às vezes a apenas um cone na fóvea, enquanto outras servem vários bastonetes de uma só vez.

Sobre o nível central do sistema visual ainda não existe um modelo completo que o explique. Nele se destaca o córtex visual que é responsável pelo processamento de imagens visuais (Machado 1993). A primeira parte do córtex visual que recebe a informação advinda do olho passando pelo NGL (Núcleo Geniculado Lateral, região responsável pelas noções de profundidade e movimento, dentre outras) é chamada de córtex visual primário (ou V1). A informação, então, flui através de uma hierarquia no córtex, V2, V3, V4 e MT. Neurônios nas áreas V1 e V2 respondem seletivamente a barras em orientações específicas, ou combinação de barras. Acredita-se que essas áreas são as responsáveis pela detecção de bordas. Nelas foram descobertas os conceitos de campos receptivos e inibitórios apresentados na próxima seção.

2.3 Campos Receptivos e Inibitórios

No começo da década de 1960, foi descoberta a presença de neurônios no córtex visual atuando como detectores de bordas em imagens (Hubel 1963). Estes neurônios respondem fortemente quando confrontados com uma borda ou linha numa dada orientação e posição no campo visual (Grigorescu *et al.* 2003*b*). Mais tarde, outras diversas funcionalidades foram reveladas na atuação destes neurônios, sendo que a única condição assumida para que um neurônio deste tipo pudesse enviar uma forte resposta seria a presença de um estímulo apropriado em uma região específica do campo visual. Tal região é chamada de campo receptivo clássico (*classical receptive field*, CRF).

Um campo receptivo é definido como uma área na qual a estimulação leva a resposta de um particular neurônio sensitivo (Levine e Shefner 2000). Em poucas palavras, o campo receptivo de um neurônio é a região de afetação que o mesmo cobre. Esse tipo de neurônio existe em muitas partes do cérebro humano, já tendo sido identificado no sistema auditório, somatosensório e visual (Levine e Shefner 2000).

Contudo, foi demonstrado que simultaneamente ao estímulo do campo receptivo, outro estímulo fora dele pode, também, ter um efeito sobre o neurônio (Rizzolatti e Camarda 1975). Na maior parte do tempo, esse estímulo é inibitório e é referido como o não-clássico campo receptivo (*non*-CRF) inibitório. Este tipo de estímulo foi motivado biologicamente pelo córtex

visual de alguns macacos e tem tido sua influência demonstrada nos seres humanos também. O campo inibitório aparenta ser uma propriedade comum dos detectores de bordas biológicos, sendo aplicado com sucesso na detecção de contorno (Grigorescu *et al.* 2003*a*). O efeito do campo inibitório também pode ser constatado no trabalho realizado por Blakemore e Tobin (Blakemore e Tobin 1972) no qual é medido o estímulo dos campos inibitórios sobre a imagem de uma barra branca.

CAPÍTULO 3

Processamento de Imagens

3.1 Introdução

Processamento de imagens (Gonzalez e Woods 2007) é um ramo da inteligência artificial que estuda a interpretação e a análise de imagens. A aplicação de técnicas de processamento de imagens pode resultar em novas imagens ou em conjuntos de características ou parâmetros relacionados a imagem.

As aplicações de processamento de imagens são várias, indo desde a compressão até o reconhecimento da imagem. Em geral, uma atividade de processamento de imagens envolve algumas das etapas apresentadas na Figura 3.1. Primeiro, uma imagem é adquirida. Em seguida, a mesma pode ser pré-processada para normalizar ou adequar seus dados ou atenuar ruídos. Então, a imagem pode ter sua textura analisada de forma a melhorar a análise realizada sobre a mesma. Ddependendo da tarefa a ser realizada, a imagem pode ser segmentada para ser, posteriormente, classificada. Tal segmentação pode sofrer um pós-processamento antes da classificação. O resultado de uma tarefa de classificação também pode ser pós-processado de forma a melhorar o resultado obtido. A seguir, são descritas as etapas desde o pré-processamento até o pós-processamento de imagens.

É importante notar que na Figura 3.1 é feita uma descrição bem suscinta das tarefas existentes em processamento de imagens. A descrição é resumida as etapas relacionados aos modelos propostos neste trabalho e aos experimentos realizados. Uma melhor visualização das etapas de uma tarefa em processamento de imagens pode ser encontrada no livro de Gonzalez e Woods (Gonzalez e Woods 2007).

3.2 Pré-Processamento de Imagens

O pré-processamento de imagens, caso ocorra, deve ser uma das primeiras etapas em qualquer tarefa de reconhecimento de padrões visuais. Alguns dos objetivos da atividade de pré-processamento são eliminar ruídos da imagem, normalizar as imagens de um dado conjunto

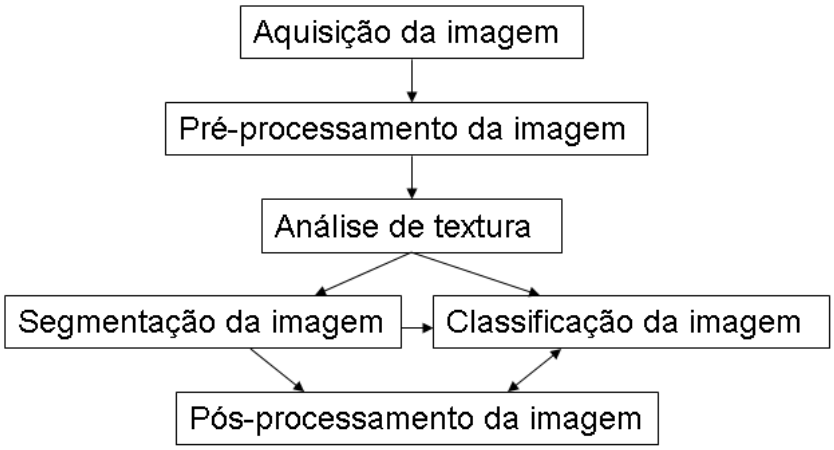


Figura 3.1 Etapas realizadas numa tarefa de processamento de imagens.

e extrair características de uma dada imagem. É importante notar que as tarefas de pré-processamento se cruzam em vários pontos, possuindo métodos e objetivos em comum.

A tarefa de pré-processamento é de suma importância quando os dados utilizados são coletados de ambientes não controlados (sob diferentes condições de iluminação, por exemplo). Nesse caso, é importante realizar uma filtragem da imagem para eliminar os ruídos e fazer uma normalização das imagens da base, de forma que elas passem a possuir o aspecto de terem sido coletadas sob as mesmas condições.

O pré-processamento de imagens é também bastante útil quando se deseja converter uma imagem para um formato adequado a um modelo que se pretende trabalhar. Nesse caso, as imagens devem ter seus dados convertidos para algum formato através de um método específico. Durante esta etapa pode ser realizada a extração de características, no intuito de comprimir a imagem ou tornar seus dados mais adequados para o processamento. Métodos de análise de textura também podem ser considerados na realização de uma tarefa de extração de características. Os mesmos são descritos na próxima seção.

A seguir são descritos duas abordagens para extração de características, a Análise de Componentes Principais (*Principal Component Analysis*, PCA) e o Discriminante Linear de Fisher (*Linear Discriminant Analysis*, LDA). Por fim, é descrito o método de Equalização de Histograma para normalização da iluminação.

3.2.1 PCA

Principal Component Analysis (PCA) é um método clássico da estatística. É uma transformação linear amplamente utilizada na análise e compressão de dados e é baseado numa representação estatística de uma variável aleatória.

Tecnicamente, PCA é uma transformação linear ortogonal que transforma um conjunto de dados para um novo sistema de coordenadas de forma que a maior variância por qualquer projeção dos dados apareça na primeira coordenada, a segunda maior na segunda coordenada e assim por diante. PCA pode ser usado para redução de dimensões num conjunto de dados, enquanto mantém as características que mais contribuem para sua variância.

O método funciona da seguinte forma:

- Primeiro deve-se adquirir as imagens a serem analisadas de tamanho $m \times n$;
- O conjunto de pixels de cada imagem deve ser representado por um vetor e todas as imagens devem ser colocadas juntas numa matriz ($k \times mn$, onde k é o número total de imagens na base);
- Deve-se então subtrair a média de cada dimensão de dados. Isso produz um conjunto de dados com média é zero;
- Então, é calculada a matriz de covariância, que aponta o grau e o tipo de dependência entre as dimensões;
- Devem ser calculados, então, os autovetores e autovalores da matriz de covariância;
- Tais autovalores servirão para representar as características mais importantes que distinguem as imagens;
- A partir da ordenação por importância dos autovalores é possível determinar quais dimensões devem ser eliminadas, obtendo assim a matriz de transformação;
- Por fim, deve-se multiplicar a imagem original pela matriz de transformação, reduzindo assim o número de características das imagens.

Assim, PCA deve realizar uma transformação das coordenadas para um conjunto de imagens. Suas aplicações são utéis nas áreas de compressão de imagens e reconhecimento de objetos.

3.2.2 LDA

Linear Discriminant Analysis (LDA) (Duda *et al.* 2000), também conhecido por discriminante linear de Fisher, é um método estatístico usado para encontrar a combinação linear entre as características que melhor separam duas ou mais classes de objetos (imagens). Ao contrário do PCA, o LDA leva em consideração o rótulo que classifica um elemento no momento de extrair suas características.

A proposta de LDA é separar amostras de grupos distintos através da transformação de seu espaço para um outro que maximize a separação inter-classes, enquanto minimiza as variâncias intra-classes. Então, o objetivo de LDA é realizar a redução de dimensionalidade ao mesmo tempo em que preserva o máximo possível de informação discriminatória das classes. Na Figura 3.2 são apresentadas as projeções geradas pelos métodos de LDA e PCA durante o processo de extração de características. A linha vermelha é o componente principal encontrado por PCA, enquanto a linha azul é o vetor encontrado por LDA. Como pode ser visto a projeção realizada pelo discriminante de Fisher possibilita a maior separação entre as classes.

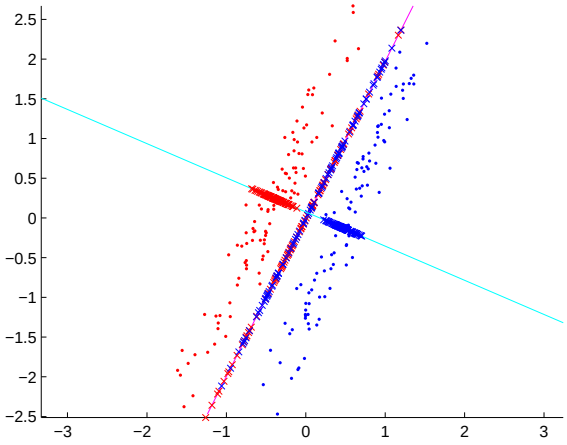


Figura 3.2 Projeção de LDA contra a de PCA. Linha azul é a projeção de LDA e a linha vermelha a projeção de PCA.

LDA tem aplicação nas mais diversas áreas que envolvem discriminação de dados. Indo desde a aplicação em processamento de imagens, para detecção de faces por exemplo, até a análise de mercado, para distinção de diferentes tipos de compradores e produtos com base em dados previamente coletados.

3.2.3 Equalização de Histograma

O método de equalização do histograma (Gonzalez e Woods 2007) tem por objetivo mapear os pixels de entrada de uma imagem para uma distribuição uniforme de intensidade. Tal método, geralmente, aumenta o constrate global das imagens, ou seja, aumenta o tamanho do intervalo de cinza onde se encontram distribuídos os pixels dentro de uma imagem em tons de cinza. A equalização de histograma mostra-se bastante útil em imagens onde os objetos e o fundo encontram-se ambos na mesma intensidade de luz (claros ou escuros).

A equalização de histograma funciona da seguinte forma: considere uma dada imagem $n = M \times N$ pixels, com cada pixel assumindo valores discretos para os níveis de cinza $k = 0, 1, ..., L-1$, onde L é o número total de níveis de cinza que podem estar presentes na imagem. Então, a função de distribuição de probabilidade g_k que pode ser utilizada para equalizar o histograma é dada por,

$$g_k = \sum_{i=0}^{k} \frac{n_i}{n}, \tag{3.1}$$

sabendo que n_i é o número de ocorrências do nível de cinza i. Na Figura 3.3 é apresentado um exemplo da equalização do histograma de uma imagem.

3.3 Análise de Textura

Análise de textura pode ser resumida na tarefa de definir um conjunto de características capazes de descrever de maneira efetiva cada região contida numa imagem (Predini e Schwartz 2008). O olho humano realiza muito bem essa tarefa, contudo é difícil definir um modelo para fazê-lo. Tal dificuldade é representada pela imensa quantidade de diferentes trabalhos desenvolvidos pelos pesquisadores da área.

Em outro trabalho desenvolvido por Tamura et al. (Tamura *et al.* 1978), textura é definida como algo que constitui uma região macroscópica com uma estrutura atribuída simplesmente aos padrões nos quais os elementos ou primitivas são organizados de acordo com uma regra de localização. Já no trabalho de Zucker e Kant (Zucker e Kant 1981), textura é dita como uma noção paradoxal. De um lado, ela é comumente usada no pré-processamento de informação visual, especialmente quando o objetivo é classificar uma imagem. Por outro lado, ninguém obteve sucesso em produzir uma definição comumente aceita para textura. A resposta a esse paradoxo, de acordo com Zucker e Kant, irá depender de um modelo mais rico e melhor desenvolvido sobre o pré-processamento de informação visual, no qual o aspecto central serão os sistemas de representação em vários níveis de abstração.

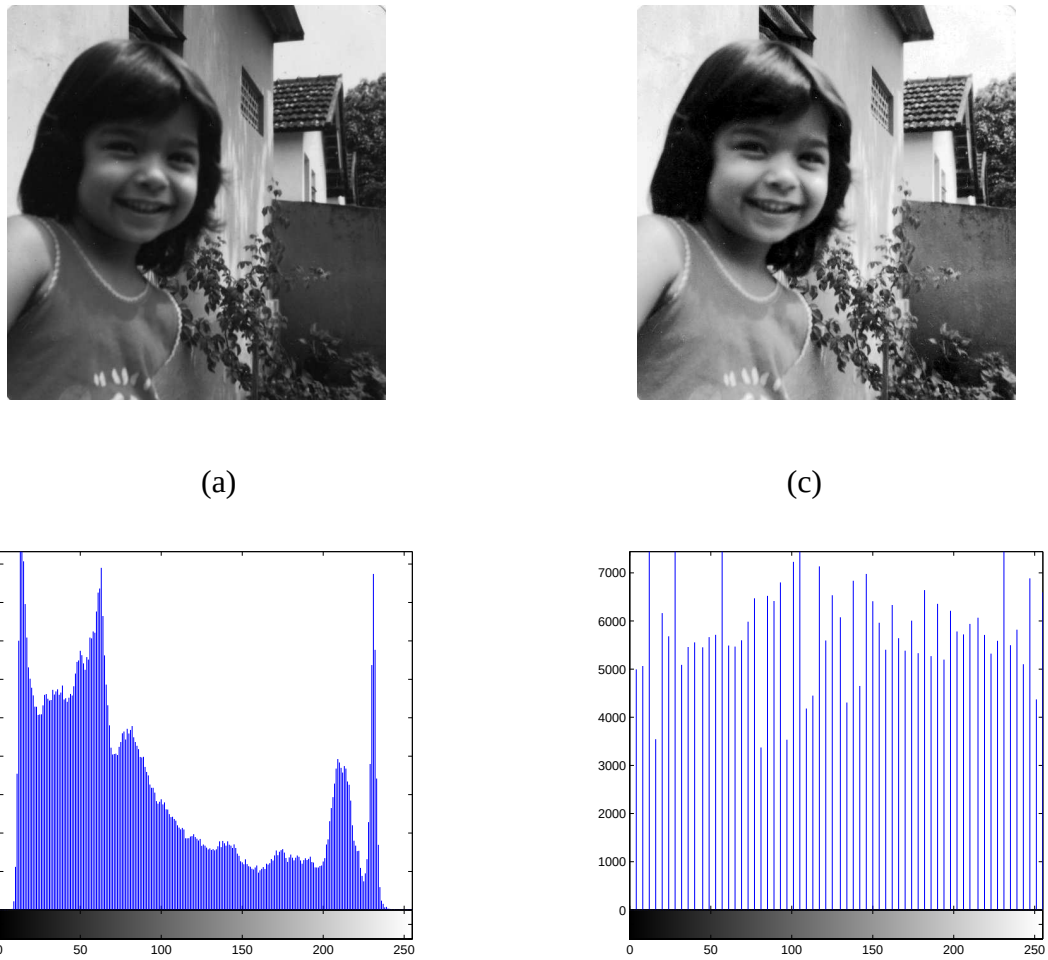


Figura 3.3 Exemplo de equalização de histograma. (a) Imagem original e (b) seu histograma. (c) Imagem equalizada e (d) seu histograma.

De um modo geral, pode-se dizer que a análise de textura é uma área útil e importante no estudo em processamento de imagens. Um bom sistema visual deve ser capaz de lidar com as texturas presentes nas mais naturais superfícies.

Do ponto de vista humano, a análise de textura é uma tarefa essencial a sua sobrevivência. Por exemplo, o sucesso em detectar um tigre no meio de folhagens pode resultar na sobrevivência de uma pessoa. A tarefa da análise de textura pelo ser humano é também bastante útil para avaliar os modelos desenvolvidos para análise de textura, comparando os resultados de um algoritmo com os obtidos pelo ser humano.

Do ponto de vista da máquina, os métodos de análise de textura têm sido usados nas mais

diversas aplicações. Dentre elas podemos citar (Chen *et al.* 1998):

- Inspeção: na qual a análise de textura é utilizada em processos industriais para encontrar defeitos a partir de certas imagens;
- Análise de imagens médicas: na qual, no geral, as aplicações envolvem a extração automática de características da imagem que serão então usadas para uma ampla gama de tarefas de classificação, como, distinção entre um tecido normal e um anormal. Dependendo do objetivo da tarefa de classificação, as características extraídas podem capturar propriedades morfológicas, propriedades de cores ou algumas outras propriedades da textura da imagem;
- Processamento de documentos: no qual as aplicações vão desde reconhecimento de endereço postal em cartas até interpretação de mapas. Nesse caso, a análise de textura pode ser útil para separar as regiões de informações importantes do fundo da imagem;
- Sensoriamento remoto: no qual a análise de textura tem sido aplicada extensivamente para classificar imagens detectadas por satélite. A classificação de segmentos terrestres, nos quais regiões homogêneas de terreno (tais como plantações de trigo, corpos de água, zonas urbanas, áreas florestadas, etc.) devem ser identificadas, tem sido um importante alvo de aplicação.

O restante dessa seção é focada em dois pontos. O primeiro é a descrição do filtro de Gabor, bastante utilizado na área de processamento de imagens. O segundo é um modelo proposto para geração automática de texturas representando áreas extraídas de imagens de satélite.

3.3.1 Filtro de Gabor

O filtro de Gabor foi originalmente introduzido por Gabor (Gabor 1946). O filtro unidimensional de Gabor é definido como a multiplicação de uma curva cossenóide/sinusóide por uma janela gaussiana, como se segue,

$$g_c(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}} cos(2\pi\omega_0 x), \tag{3.2}$$

$$g_s(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}} sin(2\pi\omega_0 x), \tag{3.3}$$

sabendo que ω_0 define a frequência central (ou seja, a frequência na qual o filtro possui máxima resposta) e σ é a dispersão da janela Gaussiana.

Os filtros de Gabor podem servir como excelentes filtro de passa-banda para sinais unidimensionais. Um filtro complexo de Gabor é definido como a soma entre a curva cossenóide e o produto da janela gaussiana pela curva senóide dado por,

$$g(x) = g_c(x) \times ig_s(x) = \frac{1}{2\pi\sigma} e^{-\frac{x^2}{2\sigma^2}} e^{i(2\pi\omega_0 x)}, \tag{3.4}$$

o qual atinge o ótimo (limite inferior) compromisso entre a localização no espaço e nos domínios da freqüência.

A função de Gabor tem sido reconhecida como uma ferramenta bastante útil na área de processamento de imagens, especialmente na análise de textura, devido as suas propriedades ótimas de localização, tanto na frequência espacial quanto na do domínio (Yang *et al.* 2003). No trabalho realizado por Daugman (Daugman 1980) foi proposto o filtro de Gabor bidimensional, desenvolvido no intuito de realizar a modelagem dos campos receptivos de alguns mamíferos. O filtro complexo 2-D de Gabor sobre o domínio da imagem (x,y) é definido por

$$G(x,y) = exp\left(-\frac{(x-x_0)^2}{2\sigma_x^2} - \frac{(y-y_0)^2}{2\sigma_y^2}\right) \times exp(-2\pi i(u_0(x-x_0) + v_0(y-y_0))), \tag{3.5}$$

sabendo que (x_0,y_0) especifica a localização na imagem, com (u_0,v_0) especificando a modulação que tem frequência espacial $\omega_0 = \sqrt{u_0^2+v_0^2}$ e orientação $\theta_0 = \arctan(v_0/u_0)$, e σ_x e σ_y são as dispersões da janela gaussiana ao longo dos eixos x e y, respectivamente. Derivando essa equação, o componente real do filtro 2-D de Gabor original pode ser obtido, utilizado em outros trabalhos (Jain e Farrokhnia 1991, Hong *et al.* 1998):

$$g(x,y;T,\theta) = exp\left(-\frac{1}{2}\left[\frac{x_\theta^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2}\right]\right)\cos\left(\frac{2\pi x_\theta}{T}\right), \tag{3.6}$$

$$x_\theta = x\cos\theta + y\sin\theta, \tag{3.7}$$

$$y_\theta = -x\sin\theta + y\cos\theta, \tag{3.8}$$

sabendo que θ é orientação do filtro de Gabor e T é o período da onda senoidal.

Na Figura 3.4, extraída do artigo de Grigorescu et al. (Grigorescu *et al.* 2003*a*), é apresentado o mapa de intensidade de um filtro 2-D de Gabor, o qual modela o perfil do campo receptivo de uma célula. Nela, os tons de cinza mais claros e mais escuros que o fundo da imagem indicam as zonas nas quais a função de Gabor assume valores positivos e negativos, respectivamente. A elipse brilhante especifica o limite do campo receptivo no qual a função de

Gabor assume valores extremamente pequenos.

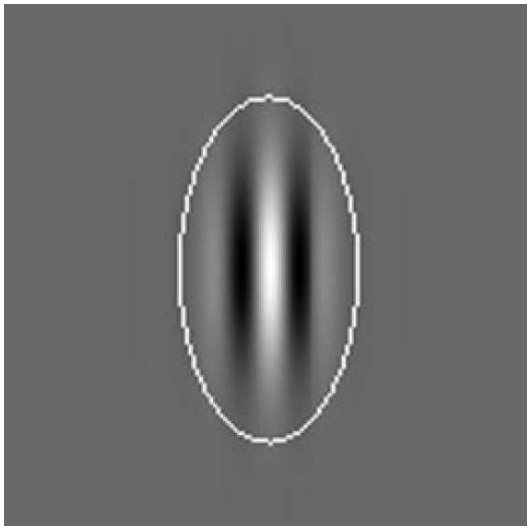


Figura 3.4 Mapa de intensidade do filtro 2-D de Gabor

A representação de Gabor de uma imagem é computada através da convolução entre a imagem com o filtro de Gabor (Zhou e Wei 2006). Suponha que $f(x,y)$ seja a intensidade da imagem em tons de cinza na coordenada (x,y), sua convolução com o filtro de Gabor $g(x,y;T,\theta)$ é definida por

$$\psi(x,y;T,\theta) = f(x,y) \star g(x,y;T,\theta), \tag{3.9}$$

sabendo que $\star$ é o operador de convolução.

Na literatura tem sido apresentados diversos parâmetros na configuração de um filtro de Gabor 2-D. Contudo, não existe um consenso sobre alguma heurística para estimar os parâmetros de Gabor.

3.3.2 Simulação de Textura de Imagens de Satélite

Simulação estocástica consiste no processo de gerar amostras de variáveis aleatórias num ambiente computacional e, então, realizar uso de tais amostras para obtenção de um determinado resultado (Triola 2005). Uma das maiores vantagens de realizar simulação de dados para processamento de imagens reside no fato de que os testes dos algoritmos são realizados em ambientes controlados. Tendo em vista esses fatos, Frery et al. (Frery *et al.* 1997) demonstrou algumas funções capazes de representar diferentes áreas de imagens SAR (*Synthetic Aperture Radar*) com base no grau de homogeneidade dos seus níveis de cinza.

O sistema SAR provém da necessidade atual de obtenção de imagens para monitoramento ambiental, mapeamento de recursos terrestres, sistemas militares, dentre outros que necessitam de imagens em alta resolução em ambientes não tão propícios para a obtenção da mesma, por exemplo, ambientes com pouca luz ou sob diferentes condições climáticas.

O sistema SAR funciona tipicamente da seguinte forma. Um avião acoplado a um radar segue capturando as imagens que estão perpendiculares a sua velocidade. Tipicamente, são produzidas imagens 2-D, onde uma dimensão é conhecida por intervalo e é uma medida da linha de visão do radar para o alvo, ela é calculada medindo o tempo de envio de um pulso e a demora para receber o eco do alvo. A outra dimensão é chamada de azimute, que é a direção medida em graus em que se encontra um astro ao redor de um observador, e é perpendicular ao intervalo. Para obtenção do azimute, uma antena é utilizada para focar a energia transmitida e recebida de um forte feixe de luz, a intensidade do feixe define a resolução de tal dimensão. O formato de processamento dos dados SAR depende de alguns valores, como número de visadas (*looks*), formato desejado para a imagem e espaço que a mesma ocupa quando armazenada. O objetivo do número de visadas é reduzir o ruído produzido na obtenção das imagens SAR.

Os dados SAR, obtidos a partir de radiação coerente, possuem sua explicação muitas vezes descrita em modelos multiplicativos, ou seja, seus dados formam uma variável resultante da multiplicação de duas outras. Assim sendo, Frery et al. (Frery *et al.* 1997) associaram funções as imagens SAR obtidas a partir de áreas florestadas e urbanas, com base na homogeneidade dos níveis de cinza dos mesmos. Na Tabela 3.1 são apresentadas tais distribuições.

Tabela 3.1 Distribuições multiplicativas para diferentes regiões

Região	Distribuição	Fórmula
Floresta	$K_a(\alpha,\lambda,n)$	$K_a(\alpha,\lambda,n) = \Gamma^{1/2}(\alpha,\lambda) \times \Gamma^{1/2}(n,n)$
Urbana	$G_a^0(\alpha,\gamma,n)$	$G_a^0(\alpha,\gamma,n) = \frac{\Gamma^{1/2}(n,n)}{\Gamma^{1/2}(\alpha,\gamma)}$

sabendo que Γ representa a função incompleta de gama.

Então, para gerar uma imagem SAR simulada, a partir de parâmetros previamente definidos (α, λ, γ e n), basta aplicar uma das funções descritas nos pixels da imagem.

3.4 Segmentação de Imagens

Segmentação de imagens refere-se à tarefa de decomposição de uma cena em vários componentes. Esta tarefa é importante para auxiliar a atividade de interpretação de dados, particio-

nando a imagem de entrada em grupos com conteúdo semântico relevante para a aplicação que está sendo desenvolvida.

O problema de segmentação é crucial na área de processamento de imagens. É a partir da segmentação que um modelo é capaz de encontrar objetos dentro de uma imagem. Além de possibilitar a detecção de objetos, o processo de segmentação é importante para uma melhor visualização da imagem, para uma quantização dos elementos dentro da imagem (área, perímetro, volume, etc.), dentre outros.

A área de segmentação de imagens envolve diversas aplicações de seus algoritmos que podem ser supervisionados ou não-supervisionados. No primeiro modo, uma imagem é segmentada com base num conhecimento *a priori* que se tem sobre as classes a serem detectadas. Enquanto que no segundo, os segmentos são formados com base nos valores de outras regiões da mesma imagem, ou seja, o algoritmo não-supervisionado busca regiões homogêneas dentro da imagem.

A seguir são descritos em mais detalhes os problemas da segmentação supervisionada e não-supervisionada. Por fim, será descrito o *Probabilistic Rand Index*, que é um método para avaliar a segmentação realizada.

3.4.1 Segmentação de Imagens Supervisionada

Segmentação supervisionada de imagens é aquela na qual a divisão da imagem em regiões se dará com base num conhecimento prévio das regiões a serem detectadas numa imagem, por essa razão ela é supervisionada. Esse tipo de abordagem, em geral, engloba um processo de classificação dentro de si, pois a segmentação é realizada de forma a indicar a qual classe cada segmento pertence.

Um exemplo de segmentação supervisionada é apresentado no trabalho realizado por Phung et al. (Phung *et al.* 2005), no qual é feito um estudo comparativo sobre as técnicas de detecção de pele em diferentes representações de cores. Nesse contexto, diferentes classificadores supervisionados são aplicados sobre cada pixel da imagem, de forma a decidir a qual classe o mesmo pertence. Algoritmos de classificação gaussiano (Yang e Ahuja 1999), bayesiano com técnica do histograma (Jones e Rehg 2002) e *Multilayer Perceptron* (Haykin 1999) foram aplicados, sendo que o último obteve o melhor resultado. É importante notar que esse tipo de segmentação ocorre pixel-a-pixel, onde o processo depende da classificação supervisionada dos pixels da imagem para um posterior agrupamento.

O algoritmo a seguir descreve o processo de segmentação pixel-a-pixel.

Como pode ser visto nesse algoritmo, cada pixel da imagem original deve ser classificado

```
   entrada: Imagem original IO de tamanho w × h
   saída  : Imagem segmentada IS
1  for i ← 1 to w do
2      for j ← 1 to h do
3          IS[i, j] ← Classifica(IO[i, j]);
4      end
5  end
```

Algoritmo de segmentação pixel-a-pixel

individualmente e o resultado de tal classificação irá indicar o valor do pixel na mesma posição na imagem segmentada.

É importante ressaltar que tal algoritmo pixel-a-pixel pode ser aplicado também sobre grupos de pixels. Dessa forma, a unidade a ser classificada deixa de ser um único pixel e passa a ser uma sub-imagem da imagem original. Nesse caso, a segmentação funciona através de uma janela deslizante sobre a imagem.

3.4.2 Segmentação Não-Supervisionada

A tarefa segmentação não-supervisionada de imagens pode ser comparada ao problema de agrupamento, no qual a intenção é particionar um certo conjunto de dados (imagem) numa dada quantidade de diferentes *clusters* (segmentos da imagem). Os tipos de algoritmos que realizam essas tarefas lidam com dados não rotulados, ou seja, não se sabe a qual classe eles pertencem.

Então, o processo de segmentação não-supervisionada de imagens busca montar grupos de pixels que são similares entre si e dissimilares dos demais pixels pertencentes a outros grupos.

Os elementos básicos da tarefa de segmentação não-supervisionada, em geral, envolvem:

- Seleção de atributos, de forma a codificar a maior quantidade possível de informações relacionadas à tarefa de interesse;
- Medida de proximidade, que indicará quão similar ou dissimilar dois elementos são;
- Critério de agrupamento, que define se um dado padrão deve ou não pertencer a um grupo;
- Algoritmo de agrupamento, responsável por revelar com base na medida de proximidade e no critério de agrupamento a estrutura agrupada do conjunto de dados;

- Verificação dos resultados, pois uma vez obtidos os resultados do algoritmo de agrupamento, deve ser verificada a sua correção. Isto geralmente é feito através de testes apropriados;
- Interpretação dos resultados.

A seguir são descritos alguns algoritmos que podem ser aplicados ao problema de segmentação não-supervisionada de imagem.

3.4.2.1 *k-Means*

O algoritmo de *k-Means* é aquele em que um conjunto de dados com n elementos é dividido em k grupos com $k < n$. Esse algoritmo assume que os atributos de cada padrão formam um vetor no espaço. Assim, seu objetivo é minimizar a variância dentro de cada grupo de dados (ou segmento de imagem). A equação a seguir descreve o valor a ser minimizado,

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} \left(x_j - \mu_i\right), \tag{3.10}$$

sabendo que existem k grupos representados por $S_i, i = 1, 2, ..., k$ e μ_i é o centróide de todos os pontos $x_j \in S_i$.

O algoritmo *k-Means* funciona, em geral, da seguinte forma. Primeiro, k centros devem ser escolhidos aleatoriamente, então para cada elemento da base deve atribuir ele a um grupo com base na sua distância para os centros. Após todos os elementos serem atribuídos, os centros de cada grupo são recalculados e o algoritmo começa outra vez. Uma vez que de uma iteração para outra nenhum elemento troque de grupo o algoritmo deve ser encerrado.

3.4.2.2 Otsu

Otsu (Otsu 1979) é um método de limiarização global que procura escolher o melhor limiar de separação entre as classes. Ele realiza o cálculo automático do limiar t objetivando separar os níveis de cinza de uma imagem em duas classes $C_0 = 0, 1, ..., t$ e $C_0 = t + 1, ..., n$, onde n é o nível máximo de cinza. Dessa forma, o método de Otsu busca separar o nível de cinza do objeto do nível de cinza do fundo da imagem. Seja p_i é a frequência de pixels na imagem com intensidade i em tons de cinza. O limiar ótimo é, então, calculado minimizando a variância σ^2 entre as classes C_0 e C_1 que é dada por,

$$\sigma^2 = \omega_0 \omega_1 \left(\mu_1 \mu_0\right), \tag{3.11}$$

$$\omega_0 = \sum_{i=0}^{t} p_i, \tag{3.12}$$

$$\omega_1 = 1 - \omega_0, \tag{3.13}$$

$$\mu_T = \sum_{i=0}^{n} i p_i, \tag{3.14}$$

$$\mu_t = \sum_{i=0}^{t} i p_i, \tag{3.15}$$

$$\mu_1 = \frac{\mu_T - \mu_t}{\omega_1} e \tag{3.16}$$

$$\mu_0 = \frac{\mu_t}{\omega_0}. \tag{3.17}$$

Então, pode-se dizer que o objetivo do método de Otsu é reduzir a quantidade de níveis de cinza de uma dada imagem. Nele, limiares são calculados para maximizar o critério de separabilidade das classes resultantes dos níveis de cinza.

3.4.2.3 *Fuzzy C-Means*

O *Fuzzy C-Means* (FCM) (Dunn 1973) é um método que, assim como o *k-Means*, tem por objetivo encontrar as centróides dos C grupos que minimizam uma função de dissimilaridade. A maior diferença reside no fato que o algoritmo do FCM não associa um elemento a um grupo diretamente, mas associa um grau de pertinência variando de 0 a 1 de cada elemento para cada grupo. Seu objetivo é minimizar a seguinte equação

$$J(U,M) = \sum_{i=1}^{c} \sum_{j=1}^{n} (u_{i,j}^{m}) D_{i,j}, \tag{3.18}$$

sabendo que:

- n é a quantidade total de elementos a serem particionados;
- c é a quantidade de partições a serem criadas, com $c < n$;
- $u_{i,j}$ é o grau de pertinência do dado j a partição i;
- m é um número real positivo maior do que 1 utilizado como parâmetro de fuzificação;
- $D_{i,j}$ é a medida de distância entre o vetor que representa o elemento j ao vetor protótipo representante da partição i;

- U é uma matriz de partições fuzzy, de dimensões $c \times n$. Tal matriz é iniciada aleatoriamente e atualizada durante o processo de minimização. Ela é representada pela matriz,

$$U = \begin{pmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,n} \\ u_{2,1} & u_{2,2} & \cdots & u_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{c,1} & u_{c,2} & \cdots & u_{c,n} \end{pmatrix}$$

  Sua atualização no instante t é dada pela seguinte equação,

$$u_{i,j}^{t} = \frac{1}{\sum_{l=1}^{c} \left(\frac{D_{l,j}}{D_{i,j}} \right)^{\frac{1}{1-m}}} \tag{3.19}$$

- M é uma matriz de protótipos de partições, de dimensões $c \times n$. Assim, como U ela é iniciada aleatoriamente durante o processo de minimização. Sua representação é dada por,

$$M = \begin{pmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,n} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{c,1} & m_{c,2} & \cdots & m_{c,n} \end{pmatrix}$$

  Sua atualização é dada por,

$$m_{i,j}^{t+1} = \frac{\sum_{j=1}^{n} \left(u_{i,j^{t+1}} \right) x_j}{\sum_{j=1}^{n} \left(u_{i,j^{t+1}} \right)}, \tag{3.20}$$

  onde x_j é o elemento representante da posição j.

3.4.3 *Probabilistic Rand Index*

Um outro ponto importante acerca dos algoritmos de segmentação é a avaliação dos mesmos na realização de sua tarefa. No trabalho desenvolvido por Unnikrishnan e Herbet (Unnikrishnan e Herbet 2005) é proposto um algoritmo para medição de similaridade entre imagens segmentadas por uma dada técnica e outras segmentadas manualmente, chamado *Probabilistic Rand Index* (PR). O PR deriva do *Rand Index* (Rand 1971) que foi proposto como uma função de similaridade que converteu o problema de comparação entre duas partições com possibilidade de diferenças na quantidades de classes segmentadas no problema de computação de represen-

tação de pares.

O método PR funciona da seguinte forma: considere um conjunto de imagens manualmente segmentadas $S_1, S_2, ..., S_K$ correspondendo a imagem $X = x_1, x_2, ..., x_i, ..., x_N$, onde o índice indica um dos N pixels presentes na imagem (considerando que N seja o total de pixels na imagem). Seja S a segmentação obtida por algum algoritmo na imagem X e que deve ser comparada com o conjunto de imagens manualmente segmentadas. Assuma, então, que o ponto x_i da imagem é correspondido pelo ponto l_i na segmentação S e por $l_i^{(k)}$ na imagem segmentada manualmente S_k. É considerada então a existência de um conjunto que possui os "verdadeiros rótulos", o qual é denotado por $\hat{l}_i$ para o pixel x_i . Embora não existe apenas um, mas vários conjuntos corretos de *labels*, a medida proposta considera a distribuição da relação de pares entre os pixels e não pelos valores definidos por um conjunto de dados. O objetivo do PR é, então, comparar uma imagem candidata a segmentação S com esse conjunto e obter uma medida adequada da consistência de S para com as K imagens segmentadas manualmente.

Então, dada as imagens segmentadas, a probabilidade empírica da relação entre os pares de pixels x_i e x_j pode ser calculada da seguinte forma,

$$\hat{P}(\hat{l}_i = \hat{l}_j) = \frac{1}{K} \sum_{k=1}^{K} I\left(l_i^{(k)} = l_j^{(k)}\right) \tag{3.21}$$

e

$$\begin{aligned} \hat{P}(\hat{l}_i \neq \hat{l}_j) &= \tfrac{1}{K} \textstyle\sum_{k=1}^{K} I\left(l_i^{(k)} \neq l_j^{(k)}\right) \\ &= 1 - \hat{P}(\hat{l}_i = \hat{l}_j) \end{aligned} \tag{3.22}$$

Assim, o PR da imagem será calculado através da seguinte equação,

$$PR(S, S_{1...K}) = \frac{1}{\binom{N}{2}} \sum_{\substack{i,j \\ i \neq j}} \left[I\left(l_i = l_j\right) P\left(l_i = l_j\right) + \left(l_i \neq l_j\right) P\left(l_i \neq l_j\right) \right] \tag{3.23}$$

Dessa forma, o PR calcula valores no intervalo $[0, 1]$, onde 0 representa que S não possui nenhuma similaridade com o conjunto $S_1, S_2, ..., S_K$ e 1 representa que todas as segmentações são idênticas.

3.5 Classificação de Imagens

A classificação de imagens pode ser vista como uma especialização do problema de classificação de padrões, onde os padrões são representados por imagens. A classificação de padrões

tem por objetivo rotular elementos de um conjunto de dados com base nas propriedades extraídas dos mesmos. Elementos classificados com o mesmo rótulo são ditos pertencentes a mesma classe, ou seja, compartilham propriedades em comum. A tarefa de classificação é de suma importância pois é a partir dela que é possível realizar o reconhecimento dos padrões.

A classificação de imagens pode ser usada para classificar pixels dentro de uma imagem ou para classificar a própria imagem por inteiro. No primeiro caso, a aplicação se destina a encontrar regiões dentro de uma imagem (nesse trabalho tal tarefa foi descrita como segmentação pixel-a-pixel). Enquanto que no segundo caso, a intenção é realizar a identificação da imagem como um todo, indicando a qual classe a mesma pertence.

Assim como na segmentação de imagens, o processo de classificação pode utilizar algoritmos supervisionados ou não-supervisionados. No caso de um classificador supervisionado, são consideradas classes previamente definidas por padrões e parâmetros. Tais classes participam de uma etapa anterior a da classificação chamada de treinamento. Nessa etapa, os parâmetros que caracterizam cada classe são definidos, sendo que esses parâmetros são utilizados para classificar cada novo padrão que chega ao classificador.

Já na classificação não-supervisionada, não existem parâmetros ou informações *a priori* sobre as classes existentes. Assim, o aprendizado de informações acontece na classificação de cada novo padrão. As amostras que compartilham propriedades semelhantes são ditas como pertencentes a mesma classe.

A tarefa de classificação de imagens tem aplicação em vários problemas. Pode-se citar detecção e reconhecimento de faces, reconhecimento de escrita, detecção de tumores, controle de qualidade em linhas de produção, dentre outros. Vários foram os métodos desenvolvidos para trabalhar os problemas que envolvem classificação de padrões e que podem ser aplicados nas tarefas de reconhecimento de imagens.

Algumas das famosas técnicas de classificação são a *MultiLayer Perceptron* (MLP) (Haykin 1999) e o k-NN (Duda *et al.* 2000). A seguir, é descrita uma outra técnica que vem apresentando excelentes resultados na área de processamento de imagens, a Máquina de Vetor de Suporte. Por fim, também é apresentada a PyraNet (Phung e Bouzerdoum 2007), que é uma rede neural modelada para classificar padrões em duas dimensões.

3.5.1 Máquina de Vetor de Suporte

As Máquinas de Vetores de Suporte (SVM, *Support Vector Machine*) são baseadas na Teoria da Aprendizagem Estatística (Vapnik 1995). Tal teoria busca encontrar condições matemáticas para escolher uma função capaz de separar dados aprendidos em problemas de categorização.

Separação essa que deve minimizar o erro de treinamento enquanto maximiza a capacidade de generalização do classificador.

Em outras palavras, pode-se dizer que uma SVM é um classificador que mapeia seus vetores de entrada para um espaço dimensional maior, onde é possível ser construído um hiperplano de separação entre duas classes. A SVM realiza essa tarefa minimizando o erro de treino, chamado erro empírico, enquanto maximiza a margem geométrica entre as diferentes classes.

A função de kernel de uma SVM é que é responsável por aumentar seu espaço dimensional na busca pela separabilidade linear. Algumas das funções de kernel mais utilizadas são:

- Gaussiana,
$$K(x,y) = \exp\left(\frac{-\|x-y\|^2}{c}\right) \tag{3.24}$$
- Polinomial,
$$K(x,y) = ((xy)+\theta)^d \tag{3.25}$$
- Multiquadrática Inversa,
$$K(x,y) = \frac{1}{\sqrt{\|x-y\|^2+c^2}} \tag{3.26}$$

É importante notar que esses exemplos de funções de kernel possuem parâmetros associados aos mesmos, c, θ, e d. Tais parâmetros devem ser pré-definidos para aplicação do kernel.

A SVM tem uma alta capacidade de generalização com robustez para categorização de dados em altas dimensões. Além de possuir sua teoria bem estabelecida nas áreas da matemática e da estatística. Suas principais aplicações encontram-se no campo do reconhecimento de caracteres manuscritos, detecção de faces, detecção de pele em fotografias, categorização de textos, regressão linear e bioinformática.

3.5.2 PyraNet

A PyraNet é uma rede neural artificial (Phung e Bouzerdoum 2007) desenvolvida para o reconhecimento de padrões visuais. Ela é motivada pelo conceito de imagens piramidais que tem sido usado com sucesso em várias tarefas de processamento de imagens (por exemplo, decomposição de imagem, segmentação de imagem e compressão de imagem (Gonzalez e Woods 2007)). Contudo, a PyraNet difere das imagens piramidais no ponto de que o processamento não-linear ocorrido nos estágios piramidais podem ser sintonizados, através do aprendizado, para problemas específicos de reconhecimento. A PyraNet também possui várias vantagens das redes neurais bidimensionais, como a integração da extração de características e

da etapa de classificação em uma única estrutura, e do uso dos conceitos de campos receptivos para reter a topologia espacial 2-D dos padrões de imagem. Além disso, a PyraNet possui um esquema de conexão que simplifica a tarefa de projetar a rede neural e permite a concepção de algoritmos de treinamento genéricos.

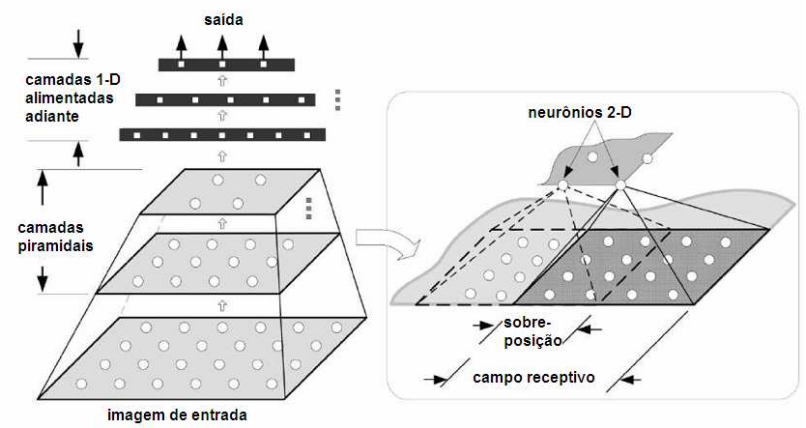


Figura 3.5 Arquitetura da PyraNet.

A arquitetura da PyraNet pode ser vista na Figura 3.5 (Phung e Bouzerdoum 2007). Ela é disposta em dois tipos de camadas. Camadas 2-D, que realizam a extração de características e a redução de dados, com os neurônios arranjados em uma matriz. Camadas 1-D, que realizam a classificação do padrão. Toda a rede é conectada em cascata, a saída da última camada 2-D é rearrumada num vetor e serve como entrada para a primeira camada 1-D.

Cada neurônio nas camadas 2-D são conectados a campos receptivos nas camadas anteriores. Tais campos receptivos podem ser sobrepostos entre neurônios vizinhos. O uso do fator de sobreposição é justificado tanto pela acuidade sobre as regiões das imagens, quanto pela tolerância a falhas devido a redundância da informação. É importante notar que os neurônios na primeira camada 2-D são conectados a campos receptivos sobre a imagem de entrada. Assim, a saída de um neurônio na posição (u,v) numa camada 2-D é dada por,

$$y_{u,v} = f\left(\underbrace{\sum_{i,j\in R_{u,v}} w_{i,j}\times y_{i,j}}_{Campo\ Receptivo} + \underbrace{b_{u,v}}_{Bias}\right), \tag{3.27}$$

sabendo que $y_{i,j}$ é a saída de um neurônio presente no campo receptivo do neurônio na posição (u,v), $w_{i,j}$ é um peso associado a tal neurônio e f é uma função de ativação não-linear.

Já as camadas 1-D assumem o papel de funcionar como uma MLP. As entradas para a primeira camada 1-D consistem numa rearrumação na forma de vetor da saída da última camada 2-D. Assim, os pesos sinápticos de toda a PyraNet são treinados por algoritmos de retropropagação do erro.

No trabalho realizado por Phung e Bouzerdoum (Phung e Bouzerdoum 2007), a PyraNet é aplicada na realização de tarefas de detecção de faces e reconhecimento de gênero (homem ou mulher), e obtém ótimos resultados na taxa de classificação, no tempo computacional gasto para classificar um novo padrão e na memória consumida durante a realização da sua tarefa.

3.6 Pós-Processamento de Imagens

O pós-processamento de imagens é aplicado, normalmente, sobre os resultados obtidos por um processo de segmentação. Seu objetivo é justamente melhorar o resultado obtido sobre uma imagem segmentada.

Geralmente, o procedimento de pós-processamento é aplicado através de operações lógicas e de morfologia matemática. Dois operadores de morfologia matemática que podem ser aplicados no pós-processamento de imagens são a dilatação e a erosão. O processo de dilatação entre uma imagem A e um elemento estruturante B corresponde a aplicar a translação de B sobre todas as posições de A e, em cada posição transladada, os valores de B são somados aos valores dos pixels de A, tomando-se o valor máximo.

De maneira análoga à dilatação, o processo de erosão consiste na translação de B sobre todas as posições de A e, em cada posição transladada, os valores de B são subtraídos dos valores dos pixels de A, tomando-se o valor mínimo.

Existem, também, os operadores de abertura ou fechamento que são resultantes da aplicação das técnicas de dilatação e erosão simultaneamente. O processo de abertura resulta da aplicação da dilatação sobre uma imagem que sofreu o processo de erosão. Enquanto o processo de fechamento é a aplicação da erosão sobre uma imagem que sofreu o processo de dilatação.

Também são consideradas no ramo do pós-processamento as operações de realce de contraste. Tais operações funcionam através da aplicação dos operadores de dilatação e erosão. Os operadores também podem ser aplicados na atenuação de ruído (Predini e Schwartz 2008).

Importante notar que as tarefas de pós-processamento nem sempre implicam em ganhos nas taxas de classificação sobre imagens segmentadas. Contudo, sua utilidade vai além disso, podendo ser aplicada como uma ferramenta para melhorar a visualização do resultado obtido sobre várias tarefas de processamento de imagens.

CAPÍTULO 4

Modelos Propostos

4.1 Introdução

Este capítulo descreve os modelos que foram desenvolvidos neste trabalho. Primeiro, é apresentado o Modelo de Segmentação e Classificação Baseado em Campos Receptivos, que é um modelo utilizado para segmentação supervisionada que tem por base os conceitos de campos receptivos. Em seguida, é apresentada uma nova rede neural, a I-PyraNet, baseada na combinação da PyraNet com os conceitos de campos inibitórios. É importante notar que os modelos propostos neste trabalho tem um forte embasamento biológico.

4.2 Modelo de Segmentação e Classificação Baseado em Campos Receptivos

O modelo SCRF (*Segmentation and Classification Based on Receptive Fields*) (Fernandes *et al.* 2008) tem como proposta a realização de tarefas de segmentação supervisionada de imagens. Ou seja, sua segmentação é realizada de forma a indicar a qual classe cada segmento pertence. Assim, o funcionamento de tal modelo deve levar em consideração imagens em seu conjunto de treino, no intuito de realizar a segmentação de uma nova imagem.

Para realizar sua tarefa o modelo SCRF deve ser empregado em conjunto com um classificador supervisionado que indique as probabilidades de uma dada imagem pertencer a cada uma das classes conhecidas. Dessa forma, o modelo SCRF pode ser considerado um modelo determinístico, se o classificador utilizado também o for. Ou seja, se para cada entrada o classificador produz uma mesma saída, o modelo SCRF assim também o fará.

O objetivo do modelo SCRF é dividir uma imagem em várias sub-imagens, de forma que a classificação de cada uma delas possa indicar a classificação de cada pixel na imagem. Isto é feito baseado nos conceitos de campos receptivos sobre uma imagem (Hubel 1963). O modelo consiste em gerar sub-imagens compartilhando alguns pixels sobrepostos, levando a vantagem de que a classificação do pixel não vai depender apenas dele mesmo, mas vai também depender

da classificação de todas as sub-imagens que o contém. A utilização do conceito de campos receptivos no modelo implica que a vizinhança de um pixel vai afetar diretamente a sua classificação. Na Figura 4.1 é apresentado o modelo proposto.

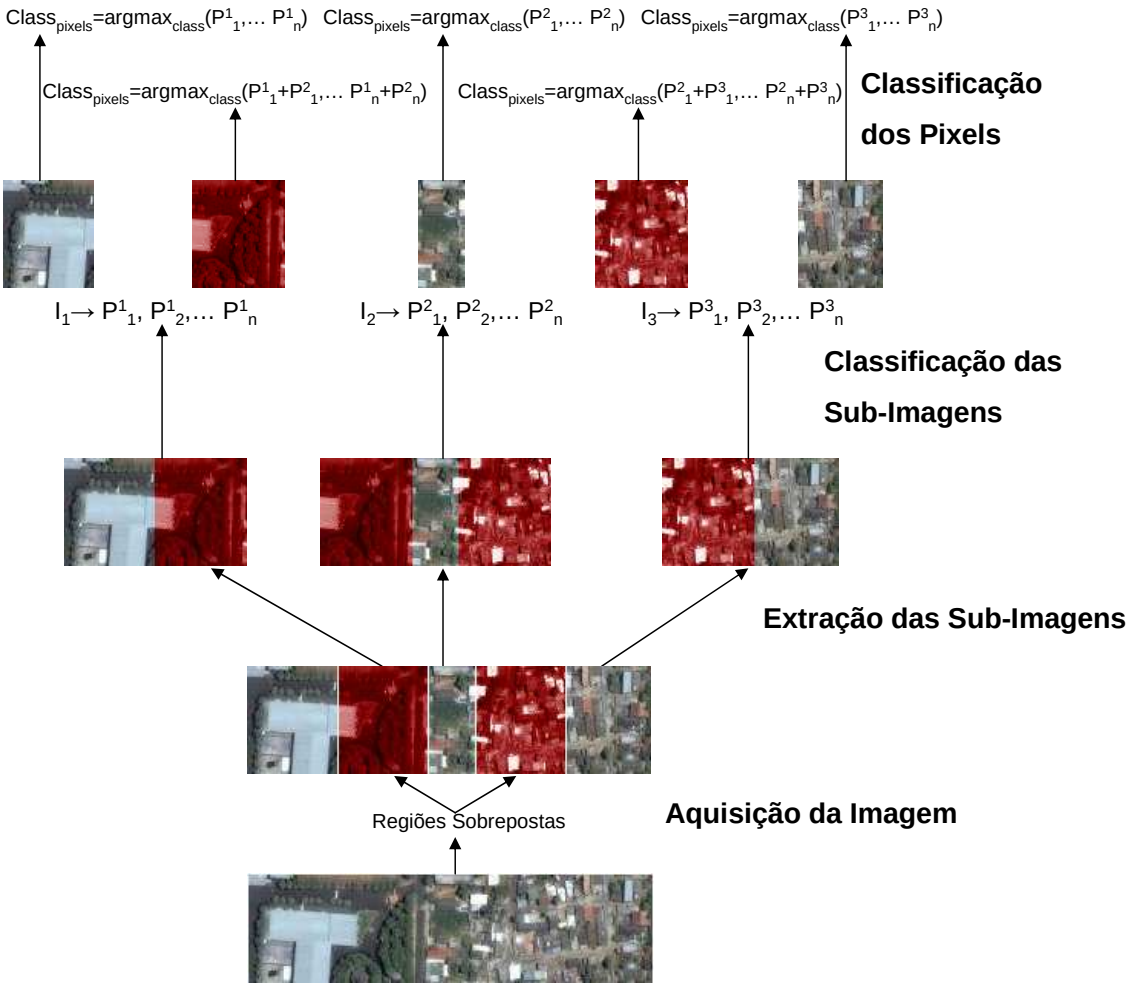


Figura 4.1 Modelo SCRF.

O modelo SCRF funciona da seguinte forma:

- Primeiro, uma imagem em duas dimensões é adquirida, (Aquisição de imagem, Figura 4.1). Tal imagem pode estar em qualquer nível de cor desde que o classificador utilizado esteja apto a trabalhar com tal nível. Neste trabalho são utilizadas apenas imagens em tons de cinza.

- Em seguida, a imagem de entrada deve ser dividida em sub-imagens (Extração de Sub-Imagens, Figura 4.1). Tais sub-imagens devem ter um tamanho definido $r \times r$ e compar-

tilharão alguns pixels sobrepostos com as sub-imagens adjacentes. A divisão em sub-imagens representa a aplicação dos conceitos de campos receptivos no modelo.

- Então, deve ser calculada a probabilidade de cada sub-imagem pertencer a cada uma das classes conhecidas (Classificação das Sub-Imagens, Figura 4.1). Isso é feito através da utilização de algum classificador supervisionado.

- Finalmente, no intuito de classificar cada pixel da imagem, o modelo define a classificação de um pixel como pertencente a classe que apresenta a maior soma de probabilidades dentre as sub-imagens que o contém (Classificação dos Pixels, Figura 4.1). Contudo, se o pixel não está numa área de sobreposição, o que significa que apenas uma sub-imagem o possui, será gerada somente uma probabilidade de pertinência a cada uma das classes e o pixel será atribuído a classe com a qual possuir o maior grau de pertinência.

A equação a seguir apresenta como cada pixel é classificado dentro do modelo SCRF:

$$C_{x_{i,j}} = \underset{class\ c}{\operatorname{argmax}} \left(\sum_{SI | x_{i,j} \in SI} P(c, SI) \right), \tag{4.1}$$

sabendo que $x_{i,j}$ é um pixel na posição (i, j) da imagem, $C_{x_{i,j}}$ é o resultado da classificação do pixel, c significa uma das classes possíveis, SI representa uma sub-imagem e $P(c, SI)$ representa a probabilidade *a posteriori* da sub-imagem SI pertencer a classe c. Apesar de neste trabalho ser utilizado o somatório das probabilidades para definir a classe do pixel, qualquer outra métrica usando as probabilidades calculadas pode ser aplicada.

4.3 I-PyraNet

A I-PyraNet (Fernandes *et al.* 2008, Fernandes e Cavalcanti 2008) é uma rede neural artificial desenvolvida para classificação de padrões visuais baseada na PyraNet (Phung e Bouzerdoum 2007). É importante notar que a PyraNet foi motivada pelos bons resultados obtidos pelas redes neurais convolucionais (Lecun *et al.* 1989) e pelos conceitos de campos receptivos. Contudo, a PyraNet considerou apenas os efeitos excitatórios dos neurônios dentro de um campo receptivo. Então, é proposto aqui a I-PyraNet, modelo no qual um neurônio pode ser capaz de enviar tanto estímulos excitatórios quanto estímulos inibitórios para os neurônios nas camadas posteriores. A idéia por trás do uso do estímulo inibitório é baseada no conceito de campos inibitórios que cercam um dado campo receptivo (Grigorescu *et al.* 2003*a*). A presença de um neurônio em um

campo inibitório irá afetar apenas a sua saída, fazendo com que a mesma inverta o seu sinal. É importante notar que a PyraNet é simplesmente um caso especial da I-PyraNet com o tamanho dos campos inibitórios igual a zero (ou seja, sem a utilização de campos inibitórios).

O principal benefício do uso dos conceitos inibitórios é de que um mesmo neurônio pode ser capaz de produzir dois tipos diferentes de saída de acordo com sua posição espacial. Na PyraNet a saída de um neurônio de uma camada 2-D irá sempre produzir a mesma entrada para os neurônios na camada posterior. Isso se deve ao fato do peso estar associado ao neurônio presente no campo receptivo e não à conexão entre dois neurônios. Enquanto que na I-PyraNet, o emprego de campos inibitórios irá possibilitar que a saída de um neurônio se transforme em duas entradas diferentes para os neurônios na camada posterior dependendo do fato do mesmo estar presente num campo receptivo ou inibitório. Sendo importante notar que tal possibilidade de inversão do sinal da saída de um neurônio é um fenômeno inspirado biologicamente.

A seguir é apresentada a arquitetura da I-PyraNet e a formulação utilizada para o treinamento da mesma.

4.3.1 Arquitetura da I-PyraNet

A arquitetura da I-PyraNet é bastante similar a da PyraNet original, porém com a diferença de que na I-PyraNet um novo parâmetro é atribuído: a quantidade horizontal ou vertical dos neurônios inibitórios que cercam qualquer campo receptivo de uma dada camada, denotado por h. Os neurônios presentes nesse espaço definido por h é que irão contribuir negativamente para o neurônio que os possui cercando seu campo receptivo. Na Figura 4.2 é apresentada a arquitetura da I-PyraNet.

Assim como sua predecessora, a I-PyraNet tem sua arquitetura formada por uma rede de multicamadas com dois diferentes tipos de camadas de processamento:

- Camadas 2-D que realizam a extração de características e redução de dados e estão localizadas na base da rede, os neurônios dela são arranjados numa matriz;
- Camadas 1-D que realizam o estágio de classificação e estão localizadas no topo da rede.

Toda a rede é conectada em cascata, ou seja, a saída de uma camada funciona como entrada para a próxima. A entrada para a primeira camada 2-D é a imagem a ser classificada. Já a entrada para a primeira camada 1-D consiste numa rearrumação em forma de linha da última camada 2-D. Cada neurônio da camada 2-D é conectado a um campo receptivo cercado por um campo inibitório sobre a camada anterior. A saída de cada neurônio 2-D consiste numa função de ativação não-linear aplicada sobre a soma ponderada das saídas dos neurônios contidos

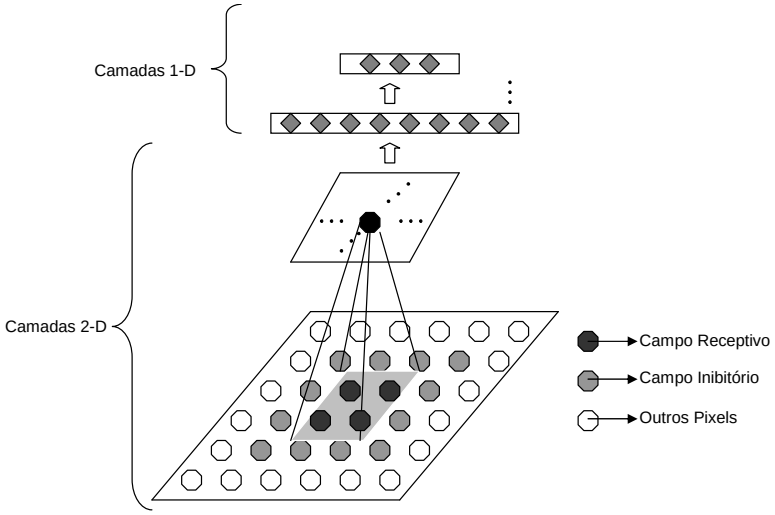


Figura 4.2 Arquitetura da I-PyraNet

em seu campo receptivo subtraída da soma ponderada das saídas dos neurônios contidos em seu campo inibitório. Neurônios numa mesma camada 2-D podem compartilhar um ou mais neurônios em seu campo receptivo ou inibitório. É importante notar que na camada 2-D os pesos não estão associados com a conexão entre neurônios, mas sim com o neurônio contido no campo receptivo ou inibitório. Dessa forma, não é compartilhada apenas a saída do neurônio como também o peso associado a ele.

Então, suponha que $r \times r$ seja o tamanho do campo receptivo de uma camada e o seja o tamanho da sobreposição horizontal ou vertical em pixels entre dois campos receptivos adjacentes (ou seja, a quantidade de pixels compartilhados). O espaço g entre campos receptivos adjacentes é dados por $g = r - o$. Então, a altura H e a largura W de duas camadas 2-D adjacentes são dados por

$$H_l = \lfloor ((H_{l-1} - o_l)/g_l) \rfloor \tag{4.2}$$

e

$$W_l = \lfloor ((W_{l-1} - o_l)/g_l) \rfloor \tag{4.3}$$

sabendo que l indica o índice da camada na rede e $l = 0$ é a imagem de entrada.

Para calcular a saída de um neurônio na I-PyraNet algumas reformulações devem ser feitas com relação a equação que calcula a saída de um neurônio na PyraNet. Então, sendo (u,v) a posição do neurônio na camada piramidal l, (i,j) a posição do neurônio na camada anterior

$l-1$ e $b_{u,v}$ o bias do neurônio (u,v), a saída do neurônio $y_{u,v}$ é dada por

$$y_{u,v} = f\left(\underbrace{\sum_{i,j\in R_{u,v}} w_{i,j}\times y_{i,j}}_{Campo\ Receptivo} - \underbrace{\sum_{i,j\in I_{u,v}} w_{i,j}\times y_{i,j}}_{Campo\ Inibitorio} + \underbrace{b_{u,v}}_{Bias}\right), \tag{4.4}$$

sabendo que $w_{i,j}$ representa o peso associado com a posição de entrada (i,j) para a camada 2-D l, $y_{i,j}$ a saída do neurônio (i,j) na camada anterior $l-1$, $R_{u,v}$ o campo receptivo do neurônio (u,v) e $I_{u,v}$ o campo inibitório do neurônio (u,v) com seu tamanho dado por h.

Assim como na PyraNet, as camadas 1-D da I-PyraNet funcionam como simples MLPs (*Multilayer Perceptron*), onde a saída de um neurônio é dada pela função de ativição não-linear aplicada sobre a soma ponderada dos neurônios conectados a ele, sendo que nesse caso o peso é associado a conexão de neurônio para neurônio. Então, a saída do neurônio y na posição n na camada 1-D l, representada por y_n^l é calculada através da seguinte equação,

$$y_n^l = f\left(\sum_{m=1}^{N_{l-1}} w_{m,n}\times y_m^{l-1} + b_n^l\right), \tag{4.5}$$

sabendo que N_{l-1} representa a quantidade de neurônios na camada anterior $l-1$, $w_{m,n}$ é o peso sináptico do neurônio m na camada $l-1$ para o neurônio n na camada l, y_m^{l-1} é a saída do neurônio m na camada $l-1$ e b_n^l é o bias do neurônio n na camada l. A saída da última camada 1-D funciona como a saída da rede.

4.3.2 Treinamento da I-PyraNet

De modo a ficar apta para realizar alguma tarefa de reconhecimento de padrões visuais, a I-PyraNet deve ser primeiro treinada. Assim como a PyraNet, a I-PyraNet é uma rede treinada por retropropagação do erro (Rumelhart *et al.* 1986), na qual o objetivo é reduzir o erro obtido entre a saída desejada e a saída da rede ajustando os pesos sinápticos na I-PyraNet. A abordagem aqui utilizada para realizar essa tarefa é a função de entropia-cruzada (CE, *cross-entropy*) (Bishop 2007), na qual as saídas da rede servem para estimar a probabilidade *a posteriori* do padrão visual pertencer a cada uma das classes conhecidas. Para esse propósito uma função *soft-max* (Bridle 1990) deve ser aplicada sobre a saída da rede. Então, sendo y_n^L a saída do neurônio n na última camada da rede L para uma imagem k, a probabilidade *a posteriori* p_n é dada por

$$p_n^k = exp\left(y_n^{L,k}\right) / \sum_{i=1}^{N_L} exp\left(y_i^{L,k}\right), \tag{4.6}$$

sabendo que N_L é a quantidade de neurônios na camada L. Então, de modo a ajustar os pesos da I-PyraNet, o gradiente do erro dos pesos deve ser calculado através da sensibilidade de cada neurônio ao erro obtido.

A sensibilidade ao erro δ para cada neurônio n na camada de saída L_{1D}, para uma imagem k é dada por

$$\delta_n^{L_{1D},k} = e_n^k f'\left(s_n^{L_{1D},k}\right), \tag{4.7}$$

sabendo que e_n^k é a saída y_n^k produzida pelo neurônio n na última camada L_{1D} menos a saída desejada d_n^k, $e_n^k = y_n^k - d_n^k$, e $s_n^{L_{1D},k}$ é a soma ponderada pelos pesos de entrada para o neurônio n na camada L_{1D} e f' é a derivada da função de ativação f. Então, para as outras camadas $l_{1D} < L_{1D}$ a sensibilidade ao erro dos neurônios é dada por

$$\delta_n^{l_{1D},k} = f'\left(s_n^{l_{1D},k}\right) \times \sum_{m=1}^{N_{l_{1D}+1}} \delta_m^{l_{1D}+1,k} \times w_{n,m}, \tag{4.8}$$

sabendo que $N_{l_{1D}+1}$ representa a quantidade de neurônios na próxima camada $l_{1D}+1$, $w_{n,m}$ é o peso sináptico entre o neurônio n na camada l_{1D} para o neurônio m na camada $l_{1D}+1$ e $\delta_m^{l_{1D}+1,k}$ é a sensibilidade ao erro do neurônio m na camada $l_{1D}+1$.

A sensibilidade ao erro para os neurônios da última camada 2-D é feita usando a equação anterior, só que rearrumada numa grade 2-D. Nas demais camadas 2-D, a sensibilidade ao erro do neurônio na posição (u,v) é dada por

$$\delta_{u,v}^{l_{2D},k} = f'\left(s_{u,v}^{l_{2D},k}\right) \times w_{u,v} \times \sum_{i=i_l^{max}}^{i_h^{max}} \sum_{j=j_l^{max}}^{j_h^{max}} \gamma_{i,j}^{l_{2D}+1,k}, \tag{4.9}$$

sabendo que $s_{u,v}^{l_{2D},k}$ é o somatório ponderado de entrada para o neurônio (u,v), $w_{u,v}$ é o peso associado ao neurônio (u,v) na camada l_{2D} e $\gamma_{i,j}^{l_{2D}+1,k}$ dado por

$$\gamma_{i,j}^{l_{2D}+1,k} = \begin{cases} \delta_{i,j}^{l_{2D}+1,k} & i_l \leq i \leq i_h, j_l \leq j \leq j_h \\ -\delta_{i,j}^{l_{2D}+1,k} & \textit{caso contrario} \end{cases}, \tag{4.10}$$

sendo $\delta_{i,j}^{l_{2D}+1,k}$ a sensibilidade ao erro do neurônio na posição (i,j) na camada seguinte, e i_l^{max}, i_h^{max}, j_l^{max}, j_h^{max}, i_l, i_h, j_l e j_h são calculados por

$$i_l^{max} = \left\lceil \frac{u - (r_{l+1} + h_{l+1})}{g_{l+1} - h_{l+1}} \right\rceil + 1, i_l = \left\lceil \frac{u - r_{l+1}}{g_{l+1}} \right\rceil + 1 \tag{4.11}$$

$$i_h^{max} = \left\lfloor \frac{u-1}{g_{l+1}-h_{l+1}} \right\rfloor + 1, i_h = \left\lfloor \frac{u-1}{g_{l+1}} \right\rfloor + 1 \tag{4.12}$$

$$j_l^{max} = \left\lceil \frac{v-(r_{l+1}+h_{l+1})}{g_{l+1}-h_{l+1}} \right\rceil + 1, j_l = \left\lceil \frac{v-r_{l+1}}{g_{l+1}} \right\rceil + 1 \tag{4.13}$$

$$j_h^{max} = \left\lfloor \frac{v-1}{g_{l+1}-h_{l+1}} \right\rfloor + 1, j_h = \left\lfloor \frac{v-1}{g_{l+1}} \right\rfloor + 1 \tag{4.14}$$

Então, os gradientes dos erros para os pesos e os bias podem ser obtidos através da equações que se seguem.

- Pesos 1-D: o gradiente de erro para os pesos sinápticos $w_{m,n}$ do neurônio m na camada $l_{1D}-1$ para o neurônio n na camada l_{1D} para todas as imagens de entrada K, é dado por

$$\frac{\partial E}{\partial w_{m,n}} = \sum_{k=1}^{K} \delta_n^k y_m^{l_{1D}-1,k}. \tag{4.15}$$

- Pesos 2-D: O peso sináptico $w_{u,v}$ associado ao neurônio (u,v) na camada l_{2D} para a camada $l_{2D}+1$ é calculado por

$$\frac{\partial E}{\partial w_{u,v}} = \sum_{k=1}^{K} \left\{ y_{u,v}^{l_{2D},k} \times \sum_{i=i_l^{max}}^{i_h^{max}} \sum_{j=j_l^{max}}^{j_h^{max}} \gamma_{u,v}^{l_{2D}+1,k} \right\}, \tag{4.16}$$

- Bias 1-D: o gradiente de erro para o bias b_n dos neurônio n na camada l_{1D} é dado por

$$\frac{\partial E}{\partial b_n} = \sum_{k=1}^{K} \delta_n^k \tag{4.17}$$

- Bias 2-D: o gradiente de erro para o bias $b_{u,v}$ do neurônio (u,v) na camada l_{2D} é dado por

$$\frac{\partial E}{\partial b_{u,v}} = \sum_{k=1}^{K} \delta_{u,v}^k \tag{4.18}$$

Finalmente, após calcular todos os gradientes, os pesos da rede podem ser ajustados pelo uso do Gradiente Descendente (Rumelhart *et al.* 1986). Isto completa a fase de treinamento da I-PyraNet.

CAPÍTULO 5

Experimentos

5.1 Introdução

Este capítulo apresenta diversos experimentos realizados com o objetivo de apresentar a capacidade do modelo SCRF e do classificador I-PyraNet na realização de suas tarefas. Diferentes métodos de processamento de imagens foram testados para servir como base de comparação com o modelo proposto neste trabalho.

Os métodos foram testados em duas diferentes tarefas de reconhecimento. A primeira é uma tarefa de detecção de floresta em imagens de satélite. Enquanto na segunda, o objetivo é realizar a tarefa de detecção de faces. Assim, quatro bancos de dados foram utilizados no decorrer de todos os testes. Dois deles montados com imagens destinadas ao problema de detecção de floresta e os outros dois destinados ao problema de detecção de faces.

Todos os experimentos deste trabalho foram realizados sobre um Pentium Dual Core de 1.73GHz de CPU e com 2-GB RAM. Eles foram testados sobre a plataforma Java.

O capítulo é organizado como se segue. Primeiro, são apresentados os bancos de dados utilizados nos experimentos. Em seguida, são apresentados os dois experimento realizados na tarefa de detecção de floresta. Então, são apresentados os dois experimentos referentes ao problema da detecção de faces. Por fim, são apresentadas algumas considerações finais.

5.2 Bancos de Dados

Para realização deste trabalho foram montados três bancos de dados. Uma trata de imagens de satélite e contém fotos de diferentes cidades do Brasil. A segunda é composta de imagens SAR simuladas a partir das imagens de satélite obtidas. A terceira trata do problema de detecção de faces e possui várias imagens representando padrões de face e de não-face. Por fim, foi analisado um quarto banco de dados, já consolidada no estado-da-arte, que trata do problema de detecção de faces. Os quatro bancos são apresentadas a seguir.

5.2.1 Banco de Imagens Reais de Satélite do Google Maps™

Esse banco foi apresentado no trabalho desenvolvido por Fernandes et al. (Fernandes *et al.* 2008), para uma aplicação de detecção de florestas em imagens de satélite. Ele foi montado com imagens extraídas do Google Maps™das cidades de Jundiaí, Manaus e Recife. Todas as imagens estão aproximadamente na mesma escala e foram coletadas sob a luz do dia.

O banco de imagens reais de satélite possui duas imagens de treino de 900×450 pixels, uma representando a área florestada e outra representando a área não florestada que podem ser vistas na Figura 5.1. Ele também possui 9 imagens de teste de 900×450 pixels e 9 segmentações realizadas manualmente de cada imagem de teste. As imagens de teste receberam a seguinte nomenclatura: Jundiai-1, Jundiai-2, Jundiai-3, Manaus-1, Manaus-2, Manaus-3, Manaus-4, Recife-1 e Recife-2. A imagem Jundiai-1 recebeu algumas segmentações manuais extras para propósitos a serem apresentados nos experimentos. Na Figura 5.2 são apresentadas as imagens de Jundiai-1 e Manaus-2 com suas respectivas segmentações manuais.

(a) (b)

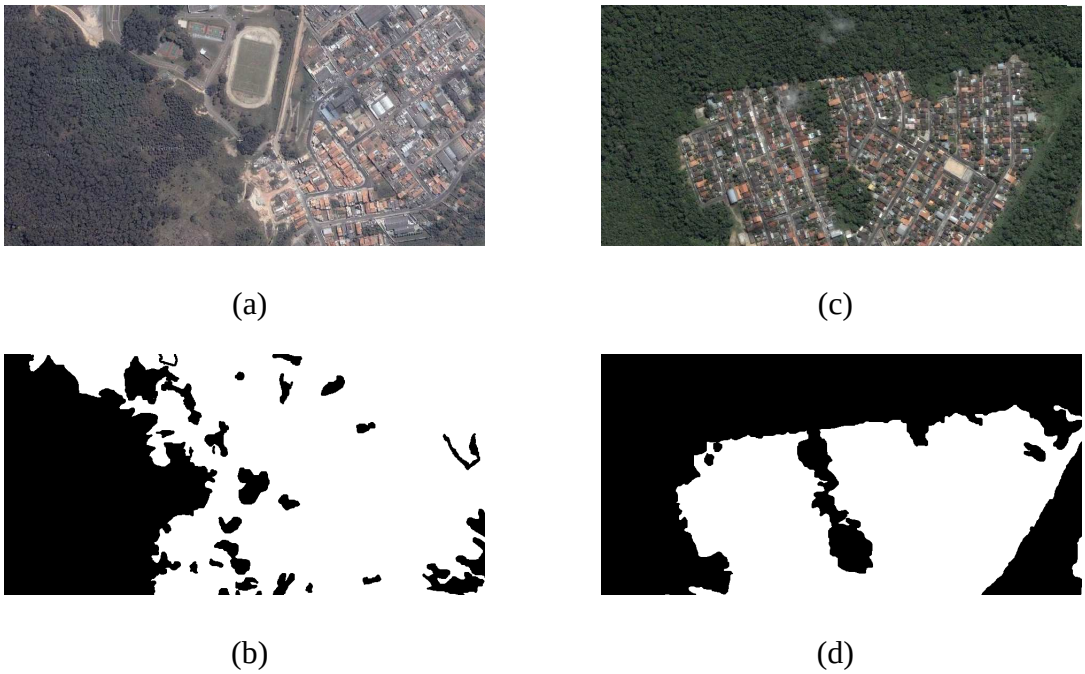
Figura 5.1 Imagens de treinamento: (a) área florestada, (b) área não-florestada.

Todas as imagens que compõem o banco de imagens de satélite extraídas do Google Maps™ podem ser encontradas em `http://cin.ufpe.br/~bjtf/SCRF`.

5.2.2 Banco de Imagens SAR Simuladas

O banco de imagens SAR simuladas foi gerado sinteticamente pelas funções descritas por Frery et al. (Frery *et al.* 1997) a partir das imagens utilizadas para validação do banco de imagens reais de satélite, apresentado anteriormente. As imagens foram simuladas de acordo com os parâmetros apresentados em (Souza 1999). A seguir, são descritos os parâmetros utilizados para as funções K_a e G_a^0.

- $K_a(\alpha, \lambda, n)$ onde $\alpha = 2$, $\lambda = 0,00023$ e $n = 3$



(a) (c)

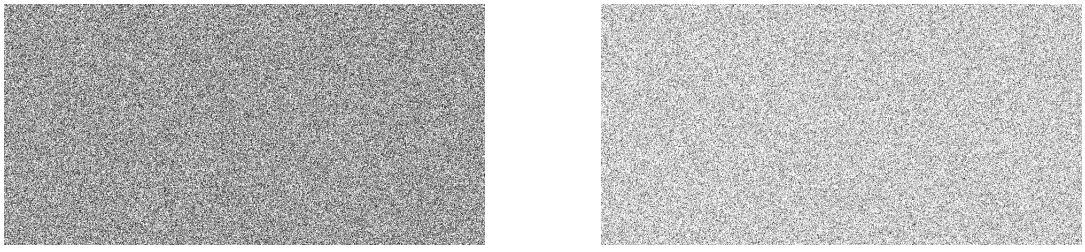
(b) (d)

Figura 5.2 Exemplos de imagens de teste. (a) Jundiai-1 e (b) sua segmentação manual, (b) Manaus-2 e (d) sua segmentação manual

- $G_a^0(\alpha,\gamma,n)$ onde $\alpha=-5$, $\gamma=203.987$ e $n=3$

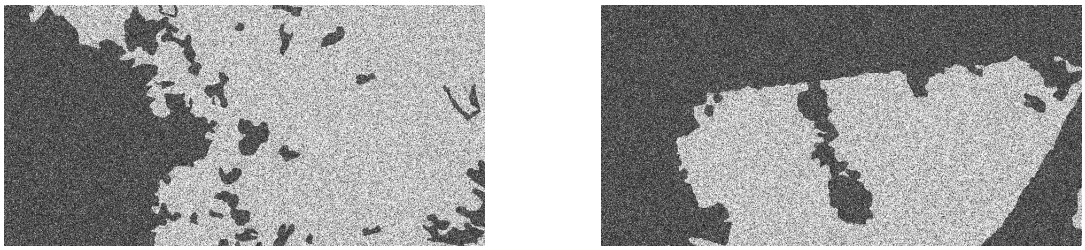
Tais parâmetros foram usadas para todas as simulações realizadas neste trabalho.

Na Figura 5.3 são apresentadas as imagens de treinamento geradas com base nas funções descritas com tamanho de 900×450 pixels. Ao passo que na Figura 5.4 são apresentadas as imagens de teste simuladas a partir das imagens de validação das cidades de Jundiai-1 e Manaus-2.



(a) (b)

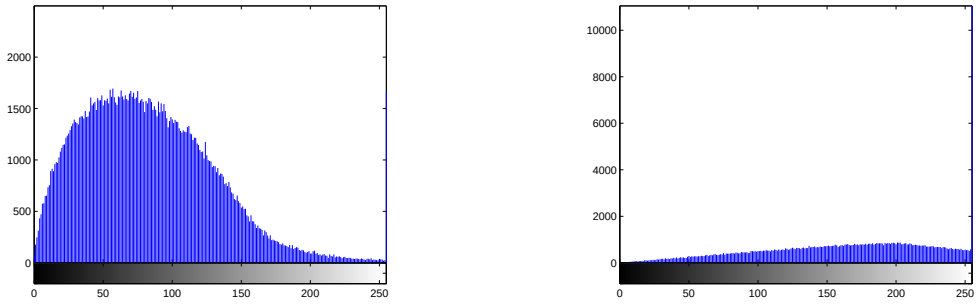
Figura 5.3 Imagens de treinamento simuladas: (a) área florestada, (b) área não-florestada.



(a) (b)

Figura 5.4 Exemplos de imagens de teste simuladas. Simulação de (a)Jundiai-1 e (b)Manaus-2.

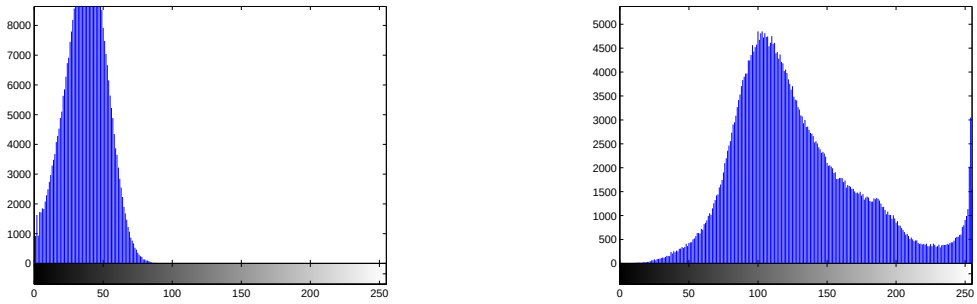
Também foram gerados os histogramas das áreas florestadas e não florestadas das imagens de treinamento simuladas. Eles são apresentados na Figura 5.5. Nela pode-se ver claramente que enquanto a simulação da floresta contém grande concentração de pixels mais escuros, a simulação de área urbana possui pixels bastante heterogêneos variando por todos os níveis de cinza.



(a) (b)

Figura 5.5 Histogramas das imagens simuladas, (a) área florestada e (b) área não-florestada.

Para efeitos de comparação, também foram geradas os histogramas das imagens de treinamento reais. Tais distribuições podem ser vistas na Figura 5.6. Nela, mais uma vez, pode-se perceber um maior espalhamento dos níveis de cinza das áreas não-florestadas, enquanto as regiões de floresta concentram-se na parte mais escura do histograma, lembrando um pouco as mesmas características encontradas nas imagens SAR simuladas.

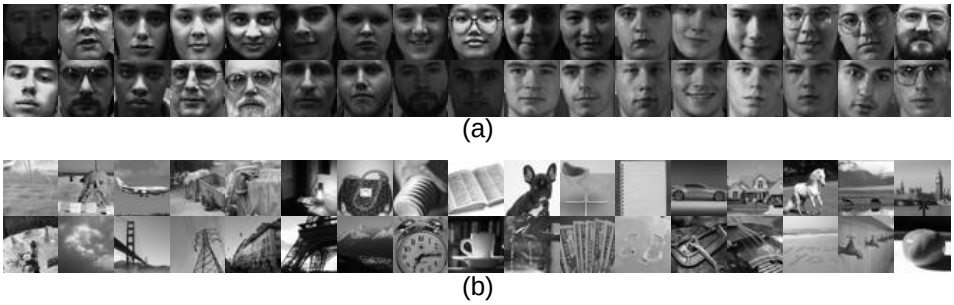


(a) (b)

Figura 5.6 Histogramas das imagens reais, (a) área florestada e (b) área não-florestada.

5.2.3 Banco de Detecção de Faces

Outro banco aqui desenvolvido pode ser aplicado no problema de detecção de faces. Os dados usados para formá-lo foram coletados de diferentes bases. Os que compõem os padrões de face foram retirados do *Essex Database*[1] e são compostos por faces de diferentes homens e mulheres. Os demais, pertencentes aos padrões de não-face, foram coletados de páginas gratuitas na Web, sendo randomicamente extraídos de fotos de cenários. Um total de 271 imagens foram criadas para cada classe, totalizando 542 imagens. Tal banco foi apresentado no trabalho desenvolvido por Fernandes e Cavalcanti (Fernandes e Cavalcanti 2008).



(a)

(b)

Figura 5.7 Exemplos de imagens: (a) faces, (b) não-faces

Na Figura 5.7 são apresentados alguns exemplos das imagens que compõem esse banco. É importante notar que todas as imagens do banco de dados montado para ser utilizado neste trabalho estão em tons de cinza e possuem um tamanho de 32×32 pixels.

[1] http://cswww.essex.ac.uk/mv/allfaces/index.html

5.2.4 Banco Usado Para Detecção de Faces MIT CBCL

O banco *MIT CBCL Face Database #1* sob domínio do *MIT Center For Biological and Computation Learning*[2] é um banco usado para detecção de faces. Tal banco contém em separado imagens de faces de pessoas e imagens de cenas aleatórias que não possuem faces. As faces presentes nesse banco são expostas sob diferentes condições de iluminação, rotação, inclinação, expressão e com detalhes (como óculos ou barba) em algumas delas. A base de treinamento possui um total de 2.429 padrões de face e 4.548 padrões de não-face, enquanto a base de teste possui 472 faces e 23.573 não-faces.

O banco MIT CBCL foi gerado a partir de diversos trabalhos. Do trabalho desenvolvido em (Heisele *et al.* 2000) foram obtidas as faces do conjunto de treino. Já o conjunto de treinamento dos padrões não-face foi obtido do trabalho realizado por Sung (Sung 1996). Finalmente, o conjunto de teste faz parte de um subconjunto do CMU Test Set 1 (Rowley *et al.* 1998).

Todas as imagens do banco MIT CBCL foram convertidas para o formato JPEG e possuem um tamanho de 19×19 pixels. Exemplos de imagens desse banco são apresentadas na Figura 5.8.

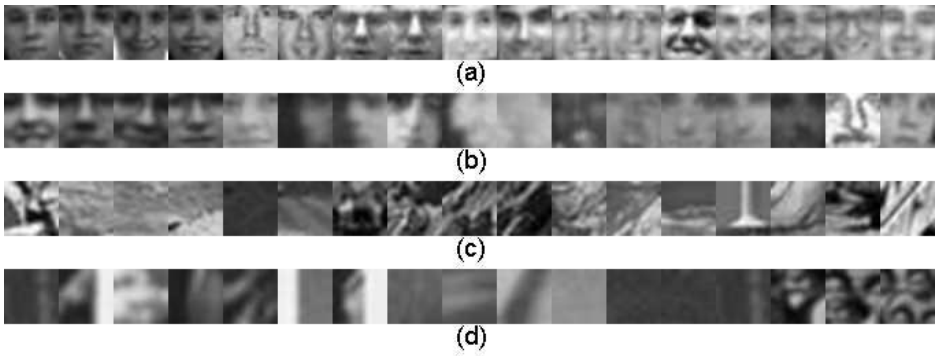


Figura 5.8 Exemplos de imagens do banco MIT CBCL. (a) Faces da base de treinamento, (b) faces da base de teste, (c) não-faces da base de treinamento e (d) não-faces da base de teste.

[2]http://www.ai.mit.edu/projects/cbcl

5.3 Detecção de Floresta em Imagens de Satélite

Atualmente, existe uma preocupação crescente com a quantidade de área de floresta existente no planeta. Projetos como o Deter (Shimabukuro *et al.* 2005), utilizado pelo Governo Federal Brasileiro, tentam determinar áreas de devastação vegetal na floresta Amazônica fazendo uso de técnicas não-supervisionadas de segmentação, na qual um especialista define após a tarefa de segmentação a qual classe cada segmento pertence. Essa é a diferença crucial entre os métodos de segmentação não-supervisionado e supervisionado; o segundo não implica necessariamente em uma posterior classificação. O objetivo desse experimento é testar a capacidade de segmentação supervisionada do modelo SCRF em combinação com o classificador I-PyraNet.

Trabalhos recentes desenvolvidos para realizar a tarefa de reconhecimento de imagens de satélite possuem a necessidade de utilizar diferentes espectros de uma mesma imagem de forma a estarem aptos a realizar uma boa classificação (Venkatesh e Raja 2002, Venkatalakshmi *et al.* 2006). Contudo, a tarefa realizada aqui pretende detectar áreas florestadas em imagens de satélite em tons de cinza, as quais são mais fácies e baratas de serem adquiridas. Vários métodos de segmentação não-supervisionados aplicados em outras situações foram usados para comparar seus resultados com os obtidos pela combinação entre o modelo SCRF e o classificador I-PyraNet. Esses métodos são o *k-Means*, o Otsu e o *Fuzzy C-Means* (FCM). Além destes, duas técnicas supervisionadas também foram utilizadas. Uma delas é o k-NN pixel-a-pixel aplicado sobre pixels em tom de cinza; a outra é a MLP classificando também pixel-a-pixel, sendo este o único classificador a atuar sobre pixels coloridos no esquema RGB. O modelo SCRF também foi aplicado usando um classificador k-NN com medidas estatísticas dos pixels dentro das sub-imagens (média, desvio-padrão, curtose e assimetria).

A seguir são apresentados os experimentos realizados com imagens reais de satélite e imagens simuladas.

5.3.1 Experimentos com Imagens Reais de Satélite

O banco de imagens aqui utilizado é o banco construído especificamente para essa tarefa com imagens colhidas do Google Maps™. A taxa de classificação é calculada por uma comparação pixel-a-pixel entre a imagem gerada e a imagem segmentada manualmente. Então, a taxa de erro para uma dada imagem é resultante da divisão do número de pixels classificado erroneamente pelo total de pixels presentes na imagem. Cada imagem do banco foi testada 10 vezes; a taxa de erro apresentada é a média obtida entre os 10 testes realizados.

O modelo SCRF usado nessa tarefa gera sub-imagens de tamanho 18×18 com uma so-

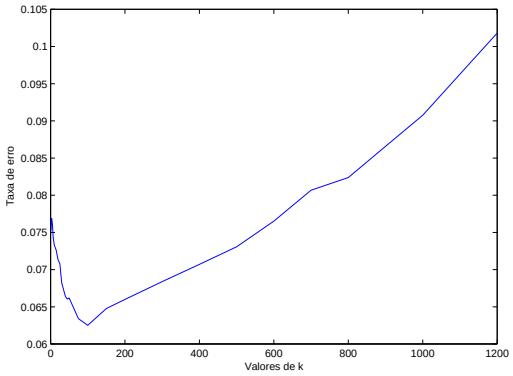


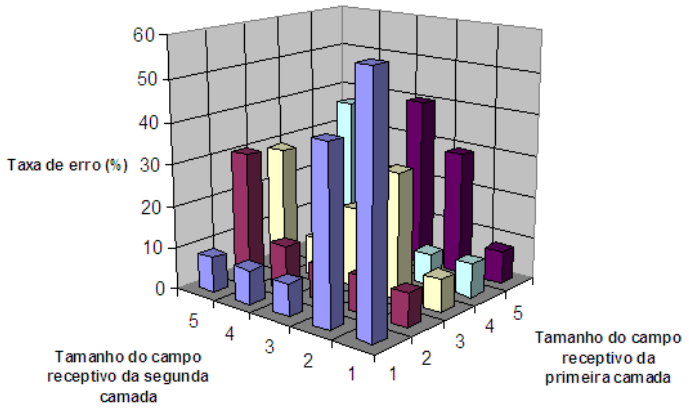
Figura 5.9 Taxa de erro, eixo vertical, para diferentes valores de *k*, eixo horizontal, na combinação entre o modelo SCRF e o classificador k-NN aplicado sobre a imagem Manaus-3.

breposição de 6 pixels, resultando num total de 1250 sub-imagens por imagem. Primeiramente, são analisados os resultados obtidos pela combinação do modelo SCRF com o classificador k-NN para diferentes valores de *k*, o resultado é apresentado na Figura 5.9. A menor taxa de erro foi obtida com $k = 100$, logo, esse será o valor utilizado para *k* nos demais experimentos.

A I-PyraNet utilizada nesse trabalho tem duas camadas 2-D e uma camada 1-D de saída com dois neurônios, com cada um deles estimando a probabilidade *a posteriori* da imagem de entrada pertencer a uma área florestada ou não. Diversas configurações para as duas camadas 2-D foram testadas para realizar a tarefa de detecção de floresta. Nas Figuras 5.10, 5.11 e 5.12 são apresentadas as taxas médias de erros obtidas sobre todas as imagens para diferentes configurações de campos receptivos usando fatores de sobreposição de tamanhos 0, 1 e 2, respectivamente, para ambas as camadas. A melhor configuração com uma taxa de erro de 7,68% foi obtida com um campo receptivo de tamanho 3×3 na primeira camada 2-D e 2×2 na segunda, tendo uma sobreposição de 1 pixel para ambas as camadas (Figura 5.11).

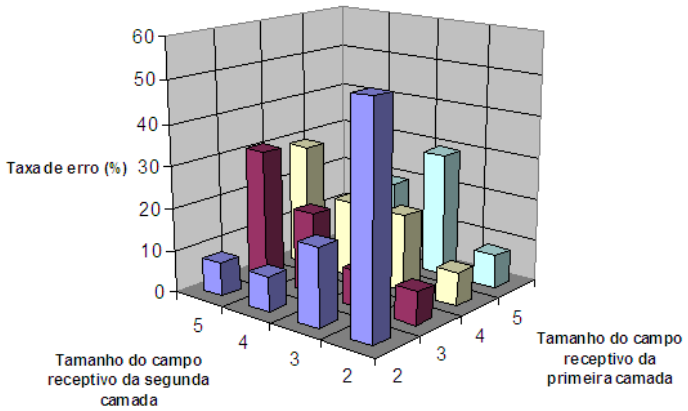
Na Figura 5.13 são apresentadas as taxas de erro com diferentes configurações para os campos inibitórios, utilizando a melhor configuração de campos receptivos e fatores de sobreposição. A figura encontra-se numa escala diferente das demais por possuir uma taxa de erro máxima muito inferior com relação a das outras figuras. É fácil perceber que o campo inibitório de tamanho 1 na primeira camada 2-D e 2 na segunda camada obtiveram a menor taxa de erro de 7,18%. Os resultados a seguir foram calculados considerando a melhor configuração para campos receptivos, campos inibitórios e fator de sobreposição.

Finalmente, nas Tabelas 5.1 e 5.2 são apresentadas as médias de erro de classificação para todos os algoritmos com o melhor resultado para cada imagem em negrito. Cada método de segmentação recebeu uma nomenclatura, como mostra a lista a seguir:



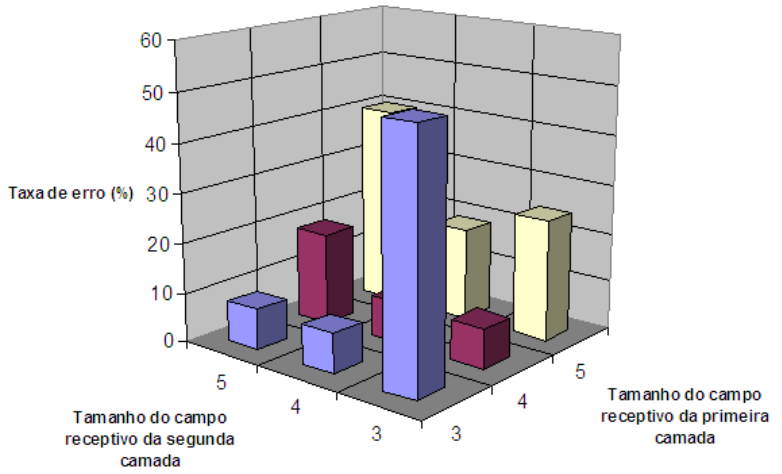
	5	4	3	2	1
1	8.49	8.15	7.87	41.96	59.43
2	30.73	10.65	8.46	8.97	7.99
3	29.37	9.31	19.18	30.04	7.87
4		40.57	7.97	7.84	8.04
5			40.57	29.58	8.01

Figura 5.10 Taxa de erro usando uma sobreposição de tamanho 0 para ambas camadas 2-D para diferentes configurações de campos receptivos.



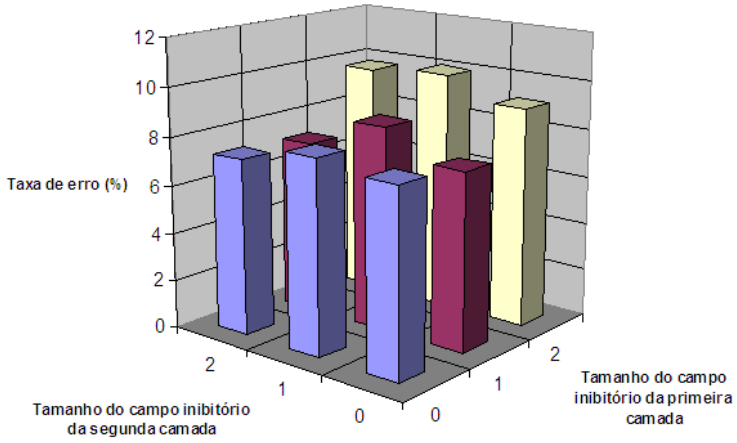
	5	4	3	2
2	7.97	8.19	18.49	53.14
3	31.27	19.21	8.58	7.68
4	29.44	18.47	18.18	7.89
5		19.82	29.51	7.95

Figura 5.11 Taxa de erro usando uma sobreposição de tamanho 1 para ambas camadas 2-D para diferentes configurações de campos receptivos.



	5	4	3
3	8.09	7.99	50.83
4	18.55	8.41	8.06
5	40.57	18.76	24.64

Figura 5.12 Taxa de erro usando uma sobreposição de tamanho 2 para ambas camadas 2-D para diferentes configurações de campos receptivos.



	2	1	0
0	7.31	8.01	7.68
1	7.18	8.51	7.35
2	9.74	10.02	9.13

Figura 5.13 Taxa de erro usando a melhor configuração para campos receptivos e fator de sobreposição para diferentes tamanhos de campos inibitórios.

Tabela 5.1 Taxa de erro em % para detecção de floresta com métodos supervisionados

Classificador	SCRF-IPN	SCRF-PN	SCRF-NN	PN	100-NN	MLP
Jundiai-1	**12**,**83**	14,27	28,17	17,11	37,81	16,87
Jundiai-2	**11**,**13**	11,91	19,80	13,86	35,77	28,96
Jundiai-3	**8**,**18**	8,46	13,61	10,91	35,77	30,15
Manaus-1	**5**,**51**	5,73	5,67	7,04	55,73	23,75
Manaus-2	**5**,**36**	5,57	5,49	6,67	53,63	27,3
Manaus-3	7,37	7,63	**6**,**48**	8,98	16,03	8,98
Manaus-4	**9**,**03**	9,51	14,79	11,50	31,75	34,52
Recife-1	**2**,**49**	2,72	3,65	3,34	48,26	23,98
Recife-2	**2**,**73**	3,31	4,91	4,20	47,18	23,18
$\bar{x}$	**7**,**18**	7,68	11,40	9,29	40,21	24,19

- *SCRF-IPN*: É a aplicação do modelo SCRF com o classificador I-PyraNet;
- *SCRF-PN*: É a aplicação do modelo SCRF com o classificador PyraNet;
- *SCRF-NN*: É a aplicação do modelo SCRF com o classificador k-NN;
- PN: É a aplicação do classificador PyraNet sozinho, o que implica que ele é aplicado numa janela deslizante por toda a imagem;
- 100-NN: É a aplicação do classificador k-NN pixel-a-pixel com $k = 100$;
- MLP: É a aplicação do classificador MLP recebendo como entrada pixels coloridos (RGB). A justificativa para sua aplicação sobre pixels coloridos vem do trabalho desenvolvido por Phung et al. (Phung *et al.* 2005), onde uma MLP é aplicada na detecção de pele em imagens coloridas;
- k-Means, Otsu e FCM: É a aplicação dos algoritmos de segmentação que levam o mesmo nome.

Primeiramente, é fácil de notar que o k-NN pixel-a-pixel e os algoritmos não-supervisionados tiveram a maior taxa de erro entre todos os métodos. O classificador MLP obteve uma pequena melhora em relação a eles, alcançando um percentual de 24,19% na taxa de erro. Por outro lado, a aplicação do modelo SCRF com a PyraNet reduziu a taxa de erro médio em $1,61$ ponto percentual em comparação com o classificador que usa apenas a PyraNet. Já a aplicação da I-PyraNet com o SCRF reduziu a taxa de erro médio ainda mais, alcançando a menor taxa entre

Tabela 5.2 Taxa de erro em % para detecção de floresta com métodos não-supervisionados

Classificador	k-Means	Otsu	FCM
Jundiai-1	23,64	23,41	23,41
Jundiai-2	35,16	36,01	33,07
Jundiai-3	22,53	21,99	21,14
Manaus-1	13,97	13,79	13,17
Manaus-2	22,97	23,27	21,48
Manaus-3	45,52	45,69	46,90
Manaus-4	36,67	36,67	34,98
Recife-1	15,98	16,14	15,35
Recife-2	16,95	17,38	16,66
$\overline{x}$	25,93	26,04	25,13

todos os métodos testados, qual seja 7,18%. É importante perceber que a menor taxa de erro médio para todas as imagens ocorreram sempre na presença do modelo SCRF.

Na Tabela 5.3 são apresentadas as taxas de falsos positivos e falsos negativos para todas as imagens usando o modelo SCRF e o classificador I-PyraNet. Os falsos positivos são os pixels da classe não-floresta classificados como floresta. Enquanto que os falsos negativos são os pixels da classe floresta classificados como não-floresta. Nessa tabela pode ser visto que o erro é bem distribuído entre os dois tipos de erro, comprovando que o modelo SCRF com a I-PyraNet é uma boa abordagem para separar regiões de floresta e de não-floresta numa imagem de satélite. Na Figura 5.14 é apresentada a segmentação realizada sobre a imagem de Recife-2 com o modelo SCRF aplicado em conjunto com a I-PyraNet, PyraNet e o k-NN.

Para realizar uma análise dos resultados sob outro ponto de vista, a medida *Probabilistic Rand Index* (Unnikrishnan e Herbet 2005) é utilizada para medir a similaridade entre as imagens geradas pelos métodos aqui testados e imagens segmentadas manualmente. Essa técnica é uma função de similaridade bastante útil para determinar consistência entre partições e prover a agregação sensata de resultados sobre diversas imagens para avaliar diferentes algoritmos. Na Tabela 5.4 é apresentada a taxa de classificação para todos os classificadores em comparação com duas imagens segmentadas manualmente da cidade de Jundiai-1. Nessa tabela pode ser visto mais uma vez que a aplicação do modelo SCRF com o classificador I-PyraNet alcançou a melhor taxa de classificação.

No intuito de realizar uma comparação mais detalhada entre os diferentes algoritmos utilizados, na Tabela 5.5 é apresentado o tempo computacional requerido para classificar uma imagem de 900×450 pixels em segundos. O tempo gasto pelos métodos que utilizam o clas-

Tabela 5.3 Taxas de falso positivo e negativo em % na detecção de floresta com o modelo SCRF e o classificador I-PyraNet

Tipo de Erro	Falso positivo (%)	False negativo (%)	Quantidade de floresta (%)
Jundiai-1	11,97	0,86	37,85
Jundiai-2	6,05	5,08	36,05
Jundiai-3	2,57	5,61	35,94
Manaus-1	1,08	4,43	56,92
Manaus-2	1,13	4,23	54,21
Manaus-3	1,25	6,12	16,11
Manaus-4	6,61	2,42	31,88
Recife-1	1,28	1,21	48,74
Recife-2	1,42	1,31	47,44
$\overline{x}$	3,71	3,47	40,57

Tabela 5.4 *Probabilistic Rand Index* sobre Jundiai-1

Método	PR
SCRF-IPN	**0,88**
SCRF-PN	0,77
SCRF-NN	0,61
PN	0,73
100-NN	0,49
MLP	0,75
k-Means	0,64
Otsu	0,64
FCM	0,64

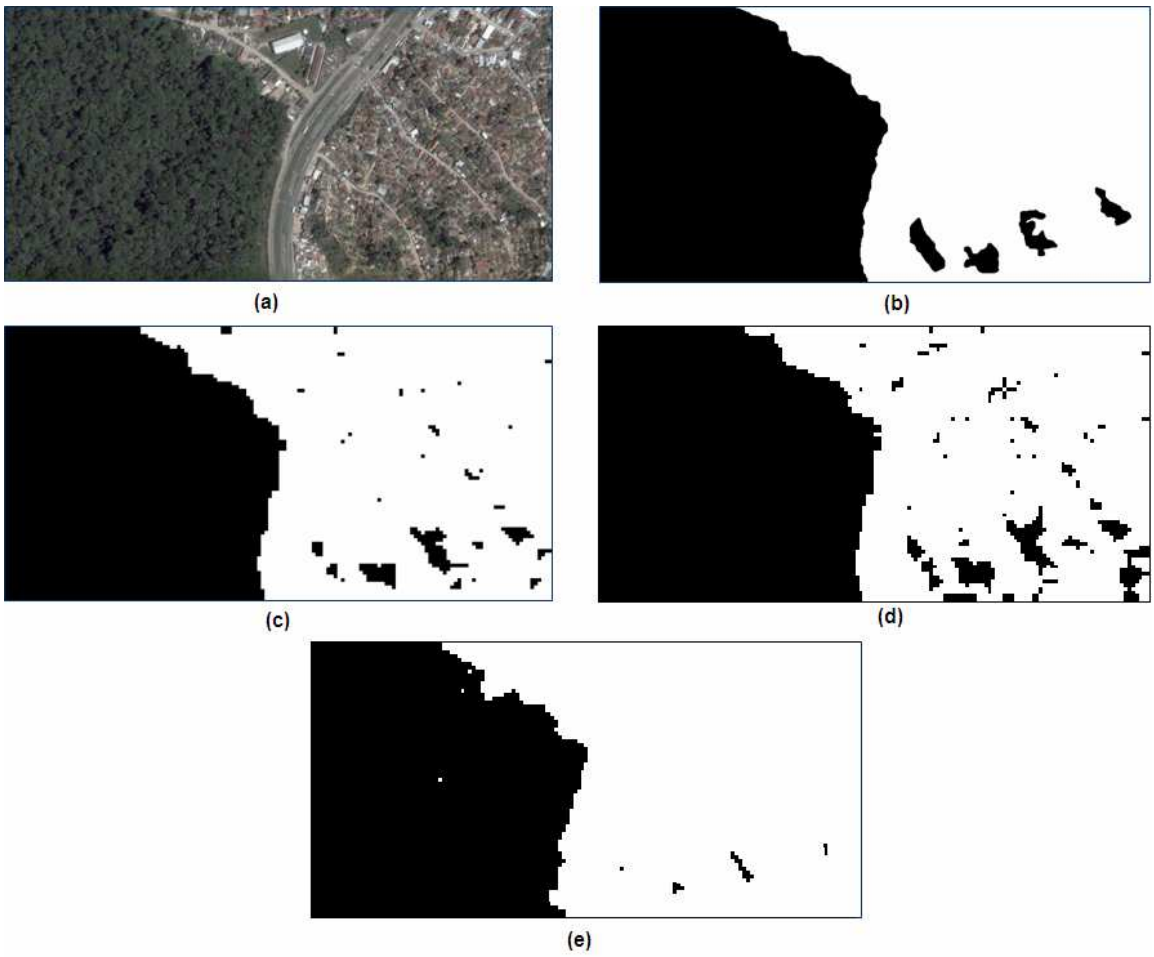


Figura 5.14 Exemplo de imagem segmentada, (a) imagem original, (b) imagem manualmente segmentada e imagem segmentada pelo modelo SCRF com o classificador (c) I-PyraNet, (d) PyraNet e (e) k-NN.

sificador k-NN foram os maiores. Particularmente, o 100-NN realizando uma classificação pixel-a-pixel levou 7 horas para segmentar uma imagem. O classificador MLP também apresentou vagarosidade no seu tempo de classificação. Tal demora pode ser atribuída ao fato da MLP ser executada pixel-a-pixel. Os algoritmos não-supervisionados executaram mais rapidamente que o k-NN e o MLP, contudo eles demonstraram ser mais lentos que os demais métodos de classificação supervisionados, com exceção ao método de Otsu que é o mais rápido dentre todos. É importante notar que o uso do modelo SCRF diminuiu mais de trezentas vezes o tempo de classificação em comparação com a abordagem pixel-a-pixel do k-NN. Enquanto que o uso de campos inibitórios aumentou em apenas 3,6% o tempo da classificação obtido pela combinação do modelo SCRF com a PyraNet. Então, é possível afirmar que o modelo SCRF e o classificador I-PyraNet trouxeram ganhos na taxa de classificação sem interferir negativamente no tempo gasto com a tarefa de reconhecimento.

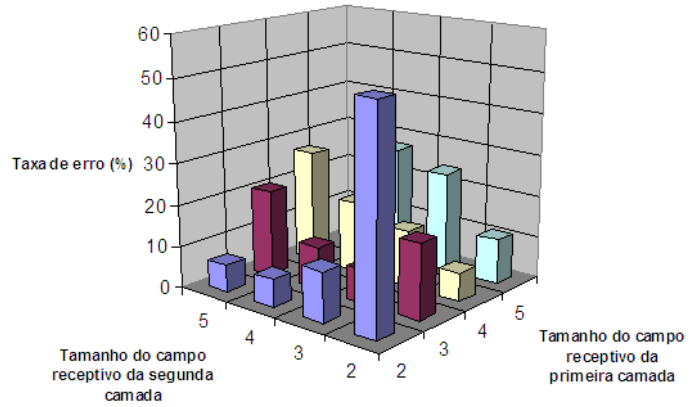
Tabela 5.5 Tempo de classificação em segundos por imagem de 900×450 pixels

Método	Tempo gasto (s)
SCRF-IPN	0,58
SCRF-PN	0,56
SCRF-NN	77,25
PN	0,47
100-NN	25200,00
MLP	12,73
K-Means	2,05
Otsu	0,18
FCM	5,60

5.3.2 Experimentos com Imagens SAR Simuladas

Nesse experimento foram utilizadas as imagens SAR geradas sinteticamente. As configurações para o modelo SCRF e da I-PyraNet são as mesmas que as utilizadas nos experimentos com imagens reais de satélite. Nesse experimento não foram realizados testes com os classificadores pixel-a-pixel. Isso se deve ao fato do k-NN pixel-a-pixel ter obtido os piores resultados nos experimentos anteriores, inclusive em termos de custo computacional, e ao fato de não existir imagens coloridas geradas sinteticamente para a aplicação da MLP pixel-a-pixel.

O resultado também advém de uma comparação pixel-a-pixel entre a imagem gerada pelo algoritmo de segmentação e a segmentação original da mesma. Assim como no experimento anterior, cada imagem foi testada 10 vezes e o resultado apresentado é a média da taxa de erro.



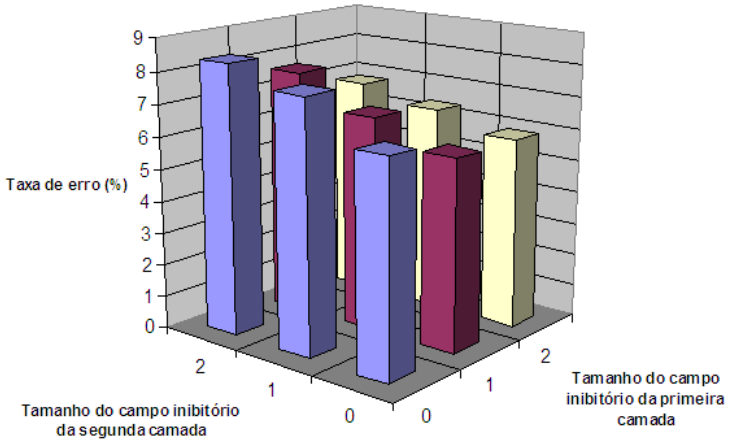
	5	4	3	2
2	6.56	6.71	11.94	52.25
3	20.73	10.08	7.79	17.69
4	27.43	17.51	13.61	6.57
5		27.67	24.12	10.78

Figura 5.15 Taxa de erro variando os campos receptivos para imagens simuladas.

É importante notar que as imagens de validação delimitam exatamente cada segmento nas imagens de teste, uma vez que cada imagem de teste foi gerada a partir da sua imagem de validação correspondente. Dessa forma, ao contrário do experimento anterior, as imagens de validação possuem uma taxa de acerto de 100% na classificação dos pixels, sendo então perfeitamente exato o percentual de erro da comparação entre a imagem gerada pelos algoritmos de segmentação e a de validação.

Primeiramente, foi assumido que a taxa de sobreposição para as camadas da I-PyraNet serão as mesmas obtidas pelas melhores configurações do experimento anterior. No caso, a taxa de sobreposição é de 1 para ambas as camadas 2-D. Então, foi calculada a melhor configuração para o tamanho dos campos receptivos. Na Figura 5.15 é apresentada a taxa de erro para as configurações de campo receptivo com o tamanho variando entre 2 e 5 para ambas as camadas. A I-PyraNet com os campos receptivos de tamanho 2 e 5 para a primeira e para segunda camada 2-D, respectivamente, obteve o melhor resultado com uma taxa de erro de 6,56%.

Na Figura 5.16 são apresentadas as diferentes taxas de erro para a I-PyraNet com os tamanhos dos campos inibitórios para ambas as camadas variando entre 0 e 2. Os tamanhos dos campos receptivos são os mesmos que obtiveram os melhores resultados na Figura 5.15, com o fator de sobreposição fixado em 1 para ambas as camadas. Assim, a presença de campos inibitórios de tamanho 2 pixels na primeira camada e a ausência do mesmo na segunda camada



	2	1	0
0	8.35	7.79	6.56
1	7.62	6.64	5.92
2	6.85	6.39	5.88

Figura 5.16 Taxa de erro usando a melhor configuração para campos receptivos e fator de sobreposição para diferentes tamanhos de campos inibitórios.

2-D apresentou a menor taxa de erro, qual seja 5,88%.

Finalmente, na Tabela 5.6 é apresentada a taxa de erro obtida por cada classificador na tarefa de detecção de floresta em imagens SAR simuladas. Nela, pode ser visto que os métodos não-supervisionados alcançaram as piorer taxas de erro, novamente. Enquanto que a combinação do modelo SCRF e do classificador I-PyraNet obteve a menor taxa de erro em todas as imagens.

5.4 Detecção de Faces

Detecção de faces consiste na tarefa de apontar numa dada imagem os locais nos quais existe uma face. Esse problema pode ser sumarizado pela dicotomia entre dizer se um dado padrão é ou não uma face. O objetivo desse experimento é analisar a qualidade da classificação da I-PyraNet na resolução do problema de detecção de faces em comparação com os resultados obtidos pela PyraNet e por outros classificadores.

A área de reconhecimento e detecção de faces tem sido um alvo recente de estudos, onde a aplicação dos mais diversos tipos de classificadores tem se mostrado útil em diferentes situações. Duas abordagens bastantes utilizadas são as de *Eigenfaces* e a de *Fisherfaces*, descritas

Tabela 5.6 Taxa de erro em % para detecção de floresta com métodos supervisionados

Classificador	SCRF-IPN	SCRF-PN	SCRF-NN	PN	k-Means	Otsu	FCM
Jundiai-1	**7,30**	7,98	10,79	9,49	20,91	20,77	20,91
Jundiai-2	**7,73**	8,51	11,05	9,91	21,49	21,20	21,34
Jundiai-3	**6,60**	7,41	9,77	8,67	21,51	21,20	21,36
Manaus-1	**6,21**	6,73	9,27	7,85	16,70	16,71	16,70
Manaus-2	**4,48**	4,95	6,35	5,88	17,13	17,12	17,13
Manaus-3	**4,87**	5,78	7,46	7,62	28,28	27,71	28,28
Manaus-4	**10,28**	11,07	15,16	12,67	22,41	22,41	22,60
Recife-1	**3,01**	3,58	4,99	4,75	18,21	18,17	18,21
Recife-2	**2,40**	3,00	4,11	4,02	18,52	18,46	18,52
$\bar{x}$	**5,88**	6,56	8,77	7,87	20,57	20,56	20,42

a seguir:

- *Eigenfaces* (Zhao *et al.* 2006, Turk e Pentland 1991), que é um método baseado na técnica de PCA. Seu objetivo principal é encontrar um conjunto de vetores-base mutuamente ortogonais que capturem as direções da máxima variância nos dados. Contudo, sua eficácia na área de reconhecimento de imagens é reduzida pelo fato do mesmo não conseguir se adequar bem a mudanças de luminosidade e expressão facial;

- *Fisherfaces* (Belhumeur *et al.* 1997), que é um algoritmo de aprendizagem supervisionado baseado no método de LDA. Tal método busca derivar uma base de projeção que separe as diferentes classes o máximo possível, enquanto une os elementos da mesma classe o máximo, também. Ou seja, ele procura maximizar a variância inter-classes enquanto minimiza a variância intra-classes.

Em outro trabalho, Makinen e Raisamo (Makinen e Raisamo 2008) apresentam um estudo na área de detecção automática e alinhamento de faces. Nele, várias combinações de algoritmos são testadas para realizar a detecção automática, alinhamento e determinação do gênero de faces. Ficou demonstrado que o alinhamento manual foi o único que conseguiu alguma melhora sobre os resultados, enquanto que os métodos que o fazem automaticamente não trouxeram ganhos de classificação. Em qualquer dos casos analisados, a SVM conseguiu as melhores taxas de classificação. Osuna et al. (Osuna *et al.* 1997) também demonstrou as grandes vantagens em se utilizar uma abordagem com SVM para realizar a detecção de faces.

No segmento de softwares para detecção de faces, o OpenCV 1.0 (*OpenCV 1.0* 2006) baseado no dectetor em cascata de Viola e Jones (Viola e Jones 2004) tem lugar de destaque.

Nele, a detecção de faces se dá com uma janela deslizante percorrendo toda a imagem, indicando se dentro da janela existe ou não uma face.

A seguir são apresentados os resultados obtidos sobre um banco criado neste trabalho, especificamente para essa tarefa, e com o banco MIT CBCL que possui mais padrões e é bastante referenciado por diversos trabalhos (Heisele *et al.* 2000, Alvira e Rifkin 2001).

5.4.1 Experimentos em Detecção de Faces

Os experimentos aqui realizados fazem uso do banco exibido em (Fernandes e Cavalcanti 2008) e criado para este trabalho. A I-PyraNet testada nesse experimento possui duas camadas 2-D e uma camada 1-D com dois neurônios, cada um estimando a probabilidade de um dado padrão de entrada ser ou não uma face. A metodologia de teste utilizada nesse experimento foi a validação-cruzada *10-fold* repetida 10 vezes. A taxa de erro apresentada é a média obtida entre todos os testes realizados.

De forma a detectar os melhores parâmetros da rede, várias configurações foram testadas. Primeiro, os tamanhos das sobreposições e dos campos inibitórios de ambas as camadas 2-D foram fixados em 0 e apenas o tamanho dos campos receptivos variaram. As taxas de erro para tais configurações são apresentadas na Tabela 5.7, onde a primeira linha apresenta o tamanho dos campos receptivos para a primeira e a segunda camada, respectivamente.

Tabela 5.7 Taxas de erro em % variando os campos receptivos

Campos receptivos	1,1	2,1	4,2	5,1	5,2	5,4
Taxa de erro	$42,0$	$13,0$	$9,1$	$8,0$	$\mathbf{7,7}$	$7,9$

A melhor configuração para as dimensões dos campos receptivos foram de 5×5 e 2×2 para a primeira e para segunda camada 2-D, respectivamente. Então, de forma a encontrar a melhor configuração para os tamanhos das sobreposições entre os campos receptivos adjacentes, novos testes foram realizados variando apenas o fator de sobreposição para cada camada. Tais resultados são apresentados na Tabela 5.8, onde a primeira linha define o tamanho das sobreposições para a primeira e para a segunda camada 2-D, respectivamente.

Tabela 5.8 Taxas de erro em % variando o fator de sobreposição

Fator de sobreposição	0,0	1,0	2,0	2,1	2,2
Taxa de erro	$7,7$	$10,6$	$\mathbf{7,2}$	$10,5$	$10,7$

A melhor configuração para os fatores de sobreposição foi de 2 e 0 para a primeira e para a segunda camada 2-D, respectivamente. Isso significa que os neurônios na segunda camada 2-D não irão compartilhar nenhum valor de entrada. O último parâmetro de configuração aqui testado é o tamanho do campo inibitório. Na Tabela 5.9 é apresentada a taxa de erro para a I-PyraNet com os campos receptivos 5×5 e 2×2 para a primeira e para a segunda camada 2-D, respectivamente, sendo o fator de sobreposição igual a 2 na primeira camada e inexistente na segunda. A primeira linha da Tabela 5.9 apresenta os tamanhos dos campos inibitórios para ambas as camadas 2-D. É importante notar que $0,0$ significa que nenhum campo inibitório é utilizado, funcionando igual a uma PyraNet. Pode-se constatar que a aplicação dos conceitos de campos inibitórios na primeira camada 2-D resultou na menor taxa de erro obtida.

Tabela 5.9 Taxas de erro em % variando os campos inibitórios

Campos inibitórios	0,0	1,0	1,1	2,1
Taxa de erro	$7,2$	$\mathbf{5,9}$	$9,1$	$8,9$

Finalmente, na Tabela 5.10 são apresentados os melhores resultados obtidos com a I-PyraNet contra os obtidos com a PyraNet e uma SVM. A SVM testada utiliza uma função de kernel polinomial de grau 3 que obteve bons resultados em (Heisele *et al.* 2000). A SVM apresentou a melgor taxa de classificação entre todos os métodos, atingindo uma taxa de erro de apenas $3,8\%$. Contudo, é importante notar que o tempo que a SVM leva para classificar uma imagem é muito superior ao da I-PyraNet. Enquanto a I-PyraNet leva apenas 0,06 milisegundos para classificar uma imagem, a SVM leva 7 milisegundos. Ou seja, a I-PyraNet classifica mais de 100 vezes mais rápido que a SVM.

Tabela 5.10 Comparação dos resultados

Classificador	I-PyraNet	PyraNet	SVM
Taxa de erro	$\mathbf{5,9}$	$7,2$	$3,8$

5.4.2 Experimentos em Detecção de Faces com Banco MIT CBCL

Os testes realizados com o banco MIT CBCL, assim como os demais, também utilizou uma I-PyraNet com duas camadas 2-D e uma camada 1-D com dois neurônios. Contudo, os resultados que são apresentados a seguir não detalham somente a taxa de erro do banco, devido ao fato do banco possuir muito mais padrões de não-face do que de face. Assim, os resultados

são apresentados na forma de uma curva ROC, Apêndice A. Em linhas gerais, a curva ROC apresenta as taxas de classificação do tipo verdadeiro positivo para diferentes erros do tipo falso positivo. As taxas de verdadeiro positivo são aqui apresentadas como resultado da divisão do total de faces corretamente classificadas pelo total de faces presentes na base de teste. Enquanto que as taxas de falso positivo representam a divisão do total de padrões não-face classificados erroneamente dividido pelo total de imagens não-face na base de teste. Considera-se que o resultado de uma curva ROC seja tão bom quanto a área que a mesma ocupa.

Primeiramente, de forma a encontrar a melhor configuração de campos receptivos da I-PyraNet, foram geradas as curvas ROC para diferentes tamanhos de campos receptivos, com os fatores de sobreposição fixados em 1 para ambas as camadas 2-D e sem a presença de campos inibitórios. Durante esses testes foram utilizadas apenas 1000 imagens de cada classe para treinamento e nenhum método de pré-processamento de imagem ou análise de textura. Na Figura 5.17 são apresentados vários gráficos com as curvas ROC para diferentes configurações de campos receptivos. Nela, cada gráfico indica o tamanho do campo receptivo na primeira camada e cada curva dentro do gráfico indica o tamanho do campo receptivo para a segunda camada. Os campos receptivos de tamanho 4 na primeira camada e 3 na segunda camada obtiveram os melhores resultados, ocupando uma área de $0,66$. Tal configuração para campos receptivos é utilizada para os demais experimentos realizados.

Então, baseado nos melhores resultados obtidos com os campos receptivos, foram calculadas as melhores configurações para os fatores de sobreposição nas camadas 2-D da I-PyraNet. Na Figura 5.18 são apresentados vários gráficos com as curvas ROC para diferentes tamanhos do fator de sobreposição na primeira camada. Cada gráfico aponta o tamanho do fator de sobreposição na primeira camada. Enquanto, cada curva dentro do gráfico indica o tamanho do fator de sobreposição na segunda camada 2-D. Pode ser visto que algumas configurações de fatores de sobreposição apresentaram a mesma área sob a curva. Contudo, a configuração com fator de sobreposição 1 para ambas as camadas será a utilizada por duas razões. A primeira é que a utilização de tal configuração possibilita tamanhos menores de camadas, fazendo com que a I-PyraNet funcione mais rapidamente. A segunda é que tal configuração apresentou uma melhor taxa de verdadeiro positivo no início da curva.

É importante notar que as imagens do banco MIT CBCL foram coletadas sob diferentes condições de ambiente e iluminação. Assim, na Figura 5.19 é apresentada a comparação dos resultados gerados sem a aplicação de técnicas de pré-processamento e com a aplicação da técnica de equalização de histograma. É fácil de perceber que a aplicação da equalização de histograma melhorou de uma forma bastante sensível a qualidade dos resultados obtidos.

Tendo em vista a melhora obtida pela aplicação da técnica de equalização de histograma,

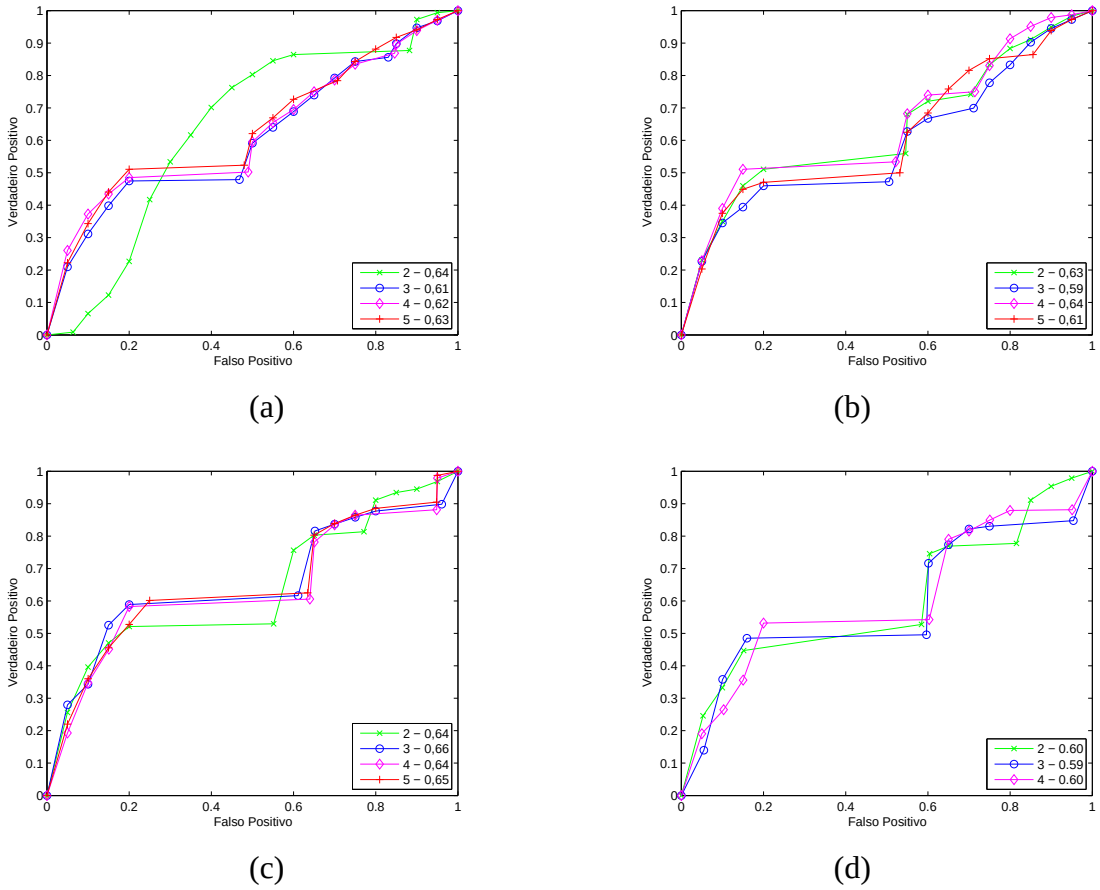


Figura 5.17 Curvas ROC para diferentes configurações de campos receptivos da I-PyraNet. Cada gráfico apresenta o tamanho do campo receptivo para a primeira camada 2-D, (a) 2, (b) 3, (c) 4, (d) 5. A legenda das curvas de cada gráfico indica o tamanho do campo receptivo na segunda camada seguido da área ocupada pela curva.

também foi aplicado o filtro de Gabor sobre as imagens do banco. Resultados obtidos em (Valois *et al.* 1982) indicaram que as células no V1 de macacos possuíam uma orientação em torno de 65º. Assim, o valor para θ será de $\pi/3$. Na Figura 5.20 são apresentadas as curvas ROC para diferentes valores de frequência do filtro de Gabor, utilizando $\theta = \pi/3$, $\sigma_x = 4$ e $\sigma_y = 4$. Como pode ser visto, o valor de 8 para a frequência obteve o melhor resultado, ocupando uma área de $0,71$.

Dessa forma, na Figura 5.21 são apresentadas as curvas ROC sobre imagens sem pré-processamento obtidas com diferentes configurações para o tamanho da dispersão da janela Gaussiana do filtro de Gabor. A I-PyraNet é utilizada com os melhores parâmetros para os campos receptivos e fatores de sobreposição. Pode ser visto que a aplicação do filtro de Gabor com $\sigma_x = 4$ e $\sigma_y = 8$ obteve o melhor resultado, ocupando uma área de $0,8$.

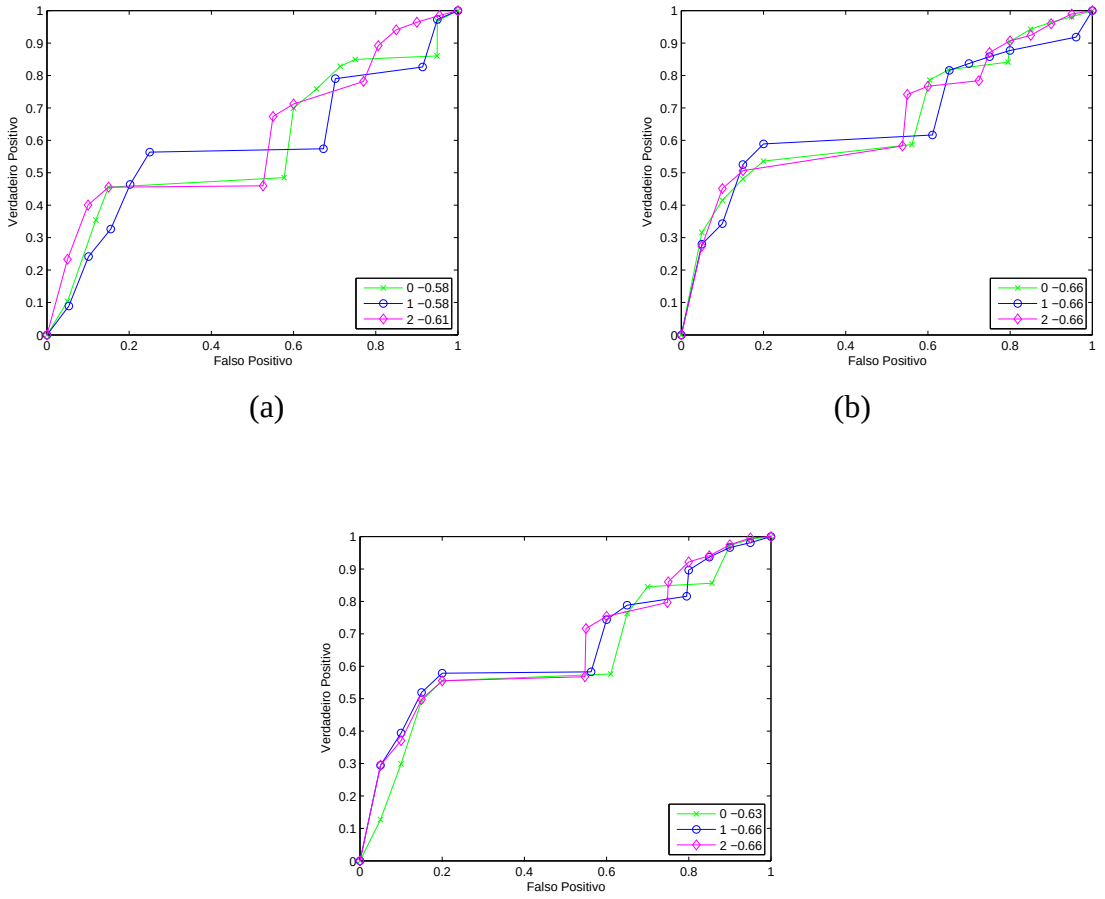


Figura 5.18 Curvas ROC para diferentes configurações de fator de sobreposição da I-PyraNet. Cada gráfico apresenta o tamanho do fator de sobreposição para a primeira camada 2-D, (a) 0, (b) 1, (c) 2. A legenda das curvas de cada gráfico indica o tamanho do fator de sobreposição na segunda camada seguido da área ocupada pela curva.

É importante notar que os experimentos apresentados até agora não estavam utilizando todas as imagens da base de treinamento do MIT CBCL e, também, não utilizavam os campos inibitórios. Na Figura 5.22 são apresentadas as curvas roc para diferentes configurações do campo inibitório, utilizando todas as imagens de treino do banco MIT CBCL. Nela, a I-PyraNet na sua melhor configuração além de fazer uso dos campos inibitórios, também utiliza imagens com histograma equalizadas e filtradas com Gabor com os melhores valores de dispersão, $\sigma_x = 4$ e $\sigma_y = 8$. Como pode ser visto, o resultado obtido aumentou o tamanho da área ocupada pela curva ROC para $0,82$ sem a presença de campos inibitórios. Já o uso de campos inibitórios de tamanho 0 na primeira camada e tamanho 1 ou 2 na segunda camada, aumentou a área ocupada pela curva para $0,83$, sendo esse o melhor resultado obtido nos experimentos com a I-PyraNet.

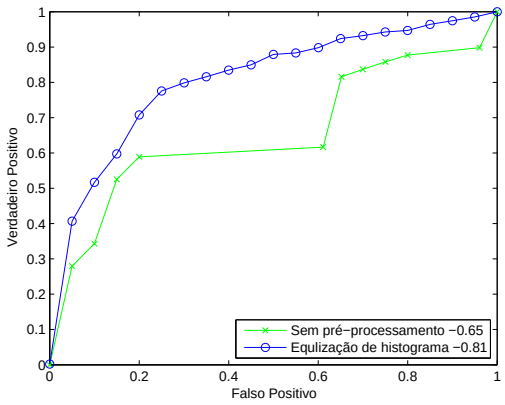


Figura 5.19 Comparação entre os resultados das imagens sem pré-processamento e com o histograma equalizado.

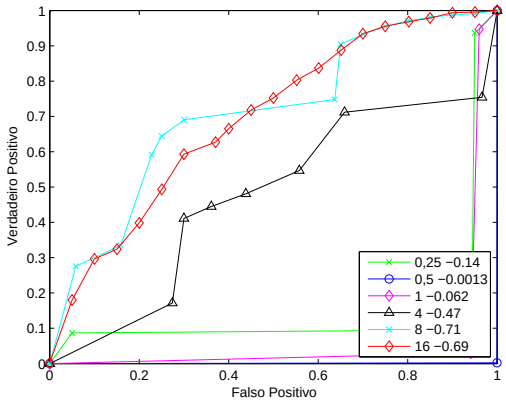


Figura 5.20 Comparação entre os resultados das imagens sem pré-processamento filtradas com o filtro de Gabor para diferentes valores de frequência. A legenda apresenta o valor utilizado para frequência seguida da área ocupada pela curva.

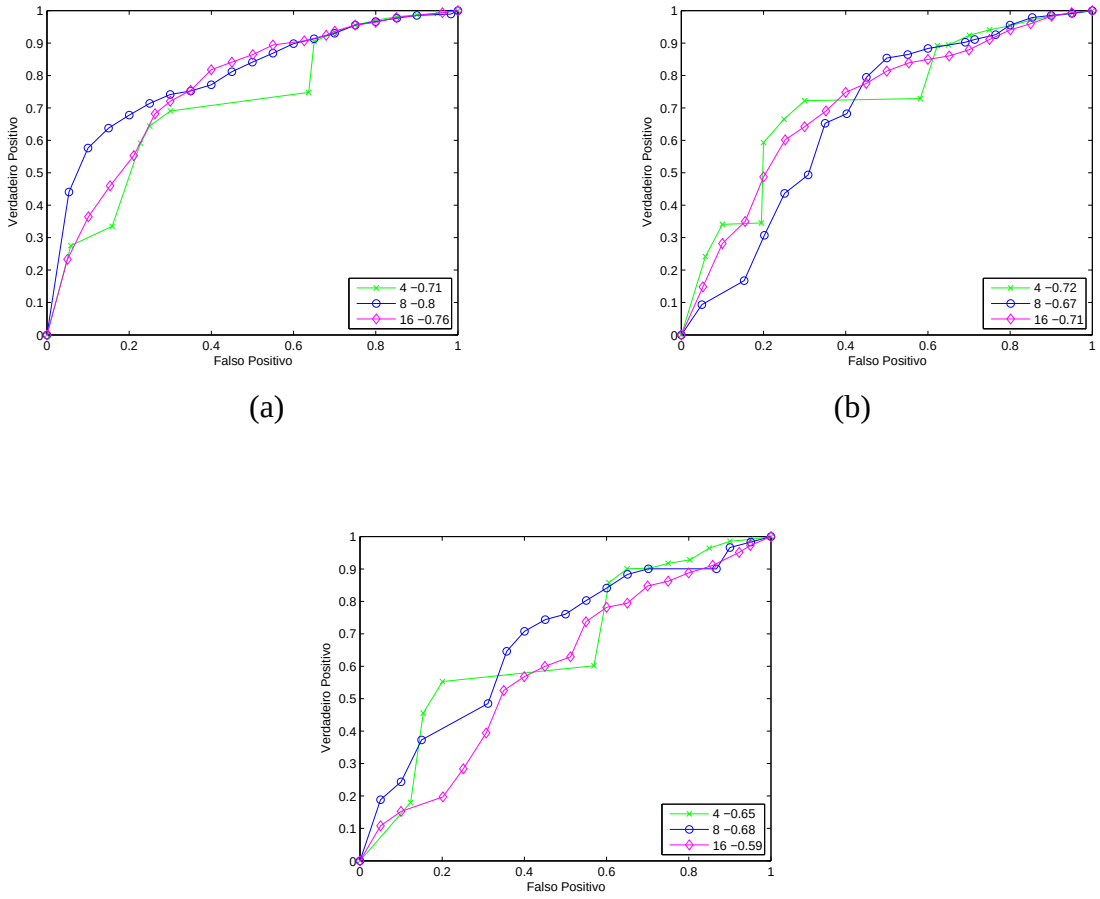


Figura 5.21 Curvas ROC para diferentes tamanhos de dispersão da janela Gaussiana do filtro de Gabor sem pré-processamento da imagem. Cada gráfico apresenta o tamanho da dispersão para σ_x, (a) 4, (b) 8, (c) 16. A legenda das curvas de cada gráfico indica o tamanho da dispersão para σ_y seguido da área ocupada pela curva.

Por outro lado, também foram realizados experimentos utilizando o classificador SVM. Na Figura 5.23 são apresentados os resultados obtidos pela SVM sem utilizar técnicas de pré-processamento de imagens, utilizando a técnica de equalização de iluminação, utilizando o filtro de Gabor e utilizando a equalização do histograma seguida do filtro de Gabor. O filtro de Gabor em todos os experimentos utiliza $f = 8$ (ou $T = 0,125$), $\sigma_x = 4$ e $\sigma_y = 8$. É possível observar que a SVM utilizando o filtro de Gabor sobre imagens com os histogramas equalizados obteve os melhores resultados, ocupando uma área de $0,9$.

Finalmente, na Figura 5.24 são comparados os resultados obtidos pela I-PyraNet, PyraNet e pela SVM. Pode-se observar que a I-PyraNet tem um resultado melhor que a PyraNet, mas a SVM apresenta a maior área entre os três classificadores. Contudo, o uso da I-PyraNet pode

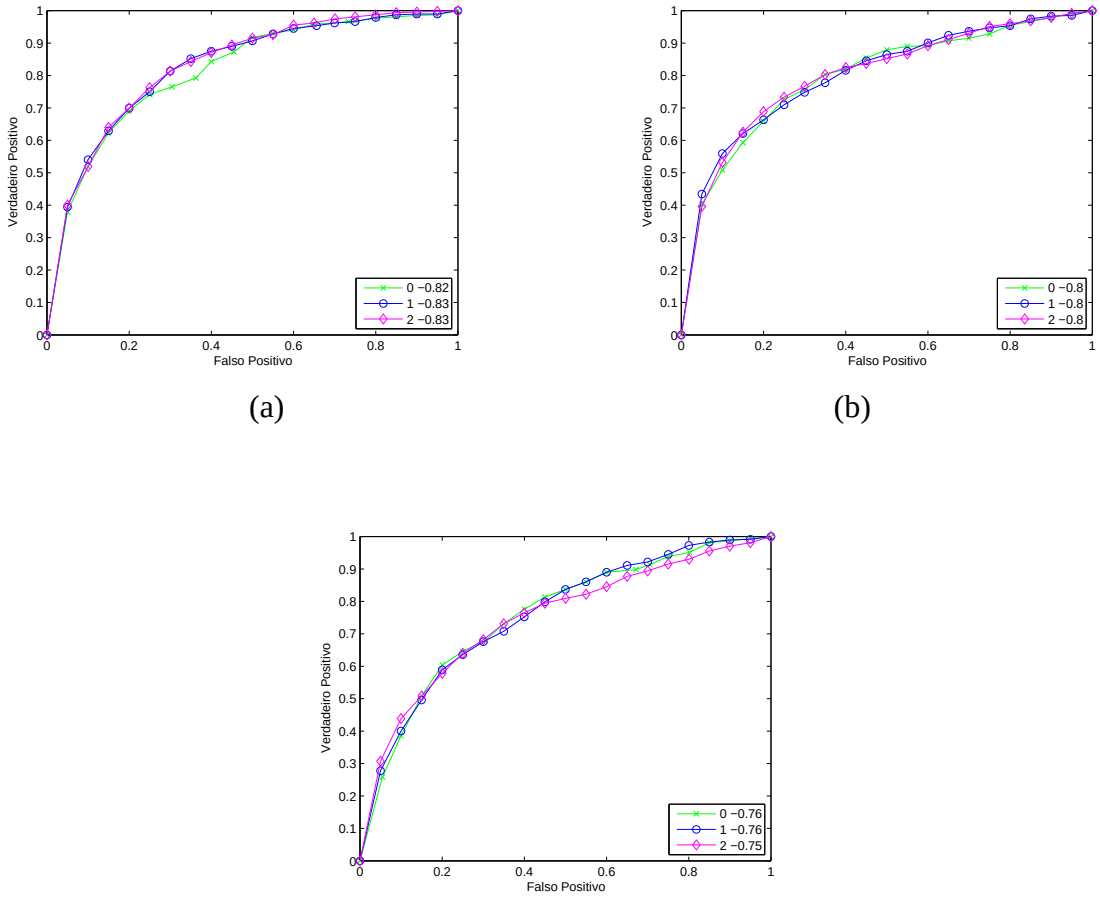


Figura 5.22 Curvas ROC para diferentes tamanhos de campos inibitórios com histograma da imagem equalizado para todas as imagens da base de treinamento. Cada gráfico apresenta o tamanho do campo inibitório para a primeira camada, (a) 0, (b) 1, (c) 2. A legenda das curvas de cada gráfico indica o tamanho do campo inibitório para a segunda camada seguido da área ocupada pela curva.

ser justificado devido ao seu rápido tempo de classificação. Enquanto a I-PyraNet leva $0,04$ milisegundos para classificar um padrão, a SVM leva em torno de 7 milisegundos. Ou seja, a I-PyraNet, com a configuração proposta nesse experimento, é 175 vezes mais rápida que a SVM.

5.5 Considerações Finais

A partir dos experimentos aqui realizados, pode-se concluir que os modelos propostos neste trabalho trazem ganhos na realização de tarefas de processamento de imagens. O modelo de

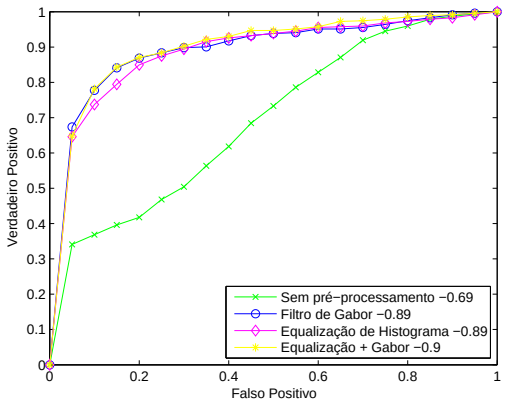


Figura 5.23 Comparação entre os resultados obtidos pela SVM sem pré-processamento da imagem, utilizando equalização de histograma, utilizando filtro de Gabor e utilizando a equalização do histograma seguida do filtro de Gabor.

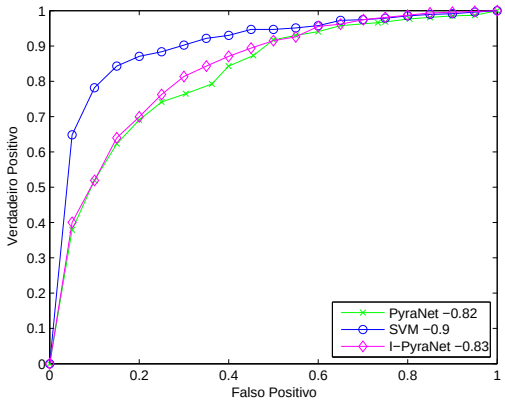


Figura 5.24 Comparação entre os resultados obtidos pela I-PyraNet, pela PyraNet e pela SVM. Todos os classificadores utilizaram o filtro de Gabor sobre imagens equalizadas.

segmentação SCRF mostrou ser uma boa abordagem para a realização de segmentação supervisionada, alcançando os melhores resultados na tarefa de detecção de floresta em imagens de satélite.

Por outro lado, o classificador I-PyraNet também apresentou excelentes resultados na realização de suas tarefas. A I-PyraNet foi o classificador que atuou melhor em conjunto com o modelo SCRF, superando, inclusive, a versão que o motivou, a PyraNet. Já no contexto da detecção de faces, a I-PyraNet, apesar de se mostrar melhor que a PyraNet na resolução das tarefas, apresentou piores resultados que o de uma SVM. Contudo, o uso da I-PyraNet pode ser motivado devido ao baixo tempo de classifição da mesma. A I-PyraNet funciona mais de 100 vezes mais rápido do que uma SVM.

Devido ao seu baixo tempo de classificação, a I-PyraNet pode ser empregada para classificar um novo padrão como face ou não-face por várias vezes, sem prejudicar seu tempo de classificação. Dessa forma, a classificação final é o resultado de uma operação sobre todas as classificações realizadas para o novo padrão. Por demandar pouco processamento, a I-PyraNet também tem seu uso motivado em sistemas embarcados, onde as restrições de processamento são grandes.

CAPÍTULO 6

Conclusão

6.1 Considerações Finais

Esta dissertação apresentou de forma suscinta e elaborada todos os conceitos e objetivos que pretendia abordar. Foram revisados a estrutura e o comportamento do sistema visual humano, apresentando o funcionamento dos campos receptivos e inibitórios, os quais motivaram os modelos propostos nesta dissertação. Também foi apresentado o tópico de processamento de imagens, com suas tarefas, técnicas e aplicações. Dentre as técnicas apresentadas, destacam-se a equalização de histograma e o filtro de Gabor, pela melhora que a aplicação dos mesmos apresentaram nos experimentos. Destacam-se, também, o modelo de segmentação pixel-a-pixel e os métodos de segmentação não-supervisionados. Além deles, também é importante mencionar a PyraNet e a SVM que obtiveram excelentes resultados nos experimentos.

Nesta dissertação também foram apresentados dois novos modelos. Um para realização de segmentação supervisionada, chamado SCRF, que funciona baseado nos conceitos de campos receptivos. E outro para classificação de padrões bidimensionais (imagens), baseado numa combinação entre a PyraNet e os conceitos de campos inibitórios, chamado I-PyraNet.

As técnicas aqui apresentadas foram empregadas nos experimentos de detecção de floresta em imagens de satélite e detecção de faces. Os modelos propostos obtiveram, quase sempre, os melhores resultados, tendo sua justificativa para o uso explicada não somente pela taxa de acerto, mas também pelo rápido desempenho na execução das suas tarefas.

6.2 Trabalhos Futuros

Nos trabalhos que se realizarão, ou que já estão em etapa inicial, pretende-se atacar os seguintes pontos:

- Utilização de técnicas de pós-processamento sobre o resultado das imagens segmentadas;
- Desenvolvimento de um modelo capaz de aproveitar o resultado de cada etapa do processamento da imagem para melhorar o resultado obtido pela etapa prévia;

- Aplicação do modelo SCRF e do classificador I-PyraNet em diferentes estudos de caso, como detecção de pele, por exemplo;
- Enriquecimento do banco de imagens de satélite Google Maps™;
- Revisão mais ampla das técnicas desenvolvidas na área de processamento de imagens de forma a encontrar novas técnicas para a realização de tarefas de processamento de imagens.

APÊNDICE A

Curva ROC

A.1 Introdução

Uma curva ROC (*Receive Operating Characteristics*) é uma técnica para visualização, organização e seleção de classificadores baseados nos seus desempenhos (Fawcett 2006). Curvas ROC têm sido usadas há muito tempo na teoria de detecção de sinais para descrever o balanceamento entre as taxas de acerto e as taxas de falso alarme dos classificadores (Swets e R.M. Dawes 2000).

Um dos primeiros trabalhos a adotarem a curva ROC na aprendizagem de máquina foi realizado por Spackman (Spackman 1989), que demonstrou os ganhos de utilizar a curva ROC para avaliar e comparar algoritmos. Recentemente, as curvas ROC têm sido cada vez mais empregadas na área de aprendizagem de máquina, devido, em parte, ao entendimento de que a simples medida da classificação é, geralmente, uma medida pobre do desempenho de um método (Provost *et al.* 1998). Além da sua utilidade devido à boa visualização do desempenho, a curva ROC é especialmente adequada para problemas que possuem a distribuição de classes desiguais.

A seguir, é descrito o modelo do gráfico ROC. Então, é apresentado o modo como deve ser gerado uma curva ROC e a utilidade de se calcular a área de uma curva ROC. Finalmente, são apresentadas algumas considerações finais.

A.2 Gráfico ROC

Primeiramente, é considerado que o problema de classificação utiliza apenas duas classes. Formalmente, uma instância I num processo de classificação deve ser mapeada com um rótulo de positiva ou negativa. Assim, cada classificador pode resultar para uma dada instância quatro tipos diferentes de saída:

- Se ela é uma instância positiva e sua classificação foi positiva, o resultado é verdadeiro positivo;
- Se ela é uma instância positiva e sua classificação foi negativa, o resultado é falso negativo;
- Se ela é uma instância negativa e sua classificação foi positiva, o resultado é falso positivo;
- Se ela é uma instância negativa sua classificação foi negativa, o resultado é verdadeiro negativo;

A taxa de verdadeiro positivo de um dado classificador é dada por

$$taxa\ vp \approx \frac{Positivos\ corretamente\ classificados}{Total\ de\ positivos} \tag{A.1}$$

Enquanto que a taxa de falso positivo de um classificador é dada por

$$taxa\ fp \approx \frac{Negativos\ erroneamente\ classificados}{Total\ de\ negativos} \tag{A.2}$$

Os gráficos ROC são gráficos bidimensionais nos quais a taxa de verdadeiro positivo é exibida no eixo Y e a taxa de falso positivo é exibida no eixo X. Vários pontos no gráfico ROC são importantes de serem notados. O ponto $(0,0)$ no gráfico indica que o classificador classificou todos os negativos corretamente, mas errou todos os positivos. Enquanto que o outro extremo, $(1,1)$, indica o oposto, que ele errou todos os negativos e acertou todos os positivos. Já o ponto $(0,1)$ representa a classificação perfeita, onde todos os elementos foram corretamente classificados. Informalmente, um ponto no gráfico ROC é dito melhor que outro se ele estiver acima e a esquerda do mesmo.

A.3 Geração de Curvas ROC

Dado um conjunto de teste, para gerar uma curva ROC a partir do mesmo, deve-se explorar a monotonicidade dos limiares de classificação. Qualquer instância classificada como positiva com respeito a um limiar será classificada positiva para todos os limiares abaixo desse. Assim, pode-se simplesmente ordenar as instâncias de teste pela probabilidade de serem falsos e mover pela lista, processando cada instância por vez e atualizando os falsos positivos e negativos.

O objetivo é então encontrar as taxas de verdadeiro positivo para diferentes taxas de falso negativo.

A.4 Área sob a Curva ROC

Uma curva ROC é uma descrição bidimensinal do desempenho de um classificador. Contudo, no intuito de comparar dois ou mais classificadores deve-se reduzir o resultado da curva ROC para um simples valor escalar representando o desempenho esperado. Um método comum é calcular a área sob a curva ROC (Bradley 1997), abreviada para AUC (*Area Under the Roc Curve*). Desde que a AUC é uma porção da área de um quadrado unitário, seu valor será sempre entre $0,0$ e $1,0$.

A AUC tem uma característica estatística importante, a AUC de um classificador é equivalente a probabilidade de que esse classificador irá ordenar uma instância positiva escolhida aleatoriamente acima de uma instância negativa escolhida aleatoriamente numa fila ordenada pela probabilidade de a instância ser positiva.

A.5 Considerações Finais

Curvas ROC são ferramentas bastante utéis para visualização e avaliação de classificadores. Elas estão aptas a prover uma rica medida de desempenho de classificação. Contudo, assim como qualquer outra métrica de avaliação, o uso da curva ROC deve ser realizado sabendo as suas características e as suas limitações. O trabalho desenvolvido por Fawcett (Fawcett 2006) apresenta um amplo estudo acerca das curvas ROC.

Referências Bibliográficas

Alvira, M. e Ryan Rifkin (2001). An empirical comparison of SNoW and SVMs for face detection. Technical Report 2001-004. Artificial Intelligence Laboratory, MIT. Cambridge, MA.

Belhumeur, P., J.P. Hepanha e D.J. Kriegman (1997). 'Eigenfaces vs fisherfaces: recognition using class specific linear projection'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (19), 711–720.

Bhuiyan, A.-A. e Chang Hong Liu (2007). 'On Face Recognition using Gabor Filters'. *Proceedings of the World Academy of Science, Engineering and Technology* **22**, 51–56.

Bishop, C. M. (2007). *Neural Networks for Pattern Recognition*. Oxford, U.K.: Clarendon.

Blakemore, C. e E.A. Tobin (1972). 'Lateral inhibition between orientation detectors in the cats visual cortex'. *Exp. Brain Res.* **15**, 439–440.

Bradley, A. (1997). 'The use of the area under the ROC curve in the evaluation of machine learning algorithms'. *Pattern Recognition* **30**(7), 1145–1159.

Bridle, J. S. (1990). 'Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition'. *Neurocomputing: Algorithms, Architectures and Applications* pp. 227–236.

Chellappa, R. e S. Chatterjee (1985). 'Classification of textures using gaussian markov random fields'. *IEEE Transactions on Acoustic, Speech, and Signal Processing* **33**, 959–963.

Chen, C. H., L. F. Pau e P. S. P. Wang (1998). *The Handbook of Pattern Recognition and Computer Vision (2nd Edition)*. World Scientific Publishing Co.

Cohen, F. e D.B. Cooper (1987). 'Simple parallel hierarchical and relaxation algorithms for segmenting noncausal markovian random fields'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **9**, 195–219.

Daugman, J. (1980). 'Two-dimensional analysis of cortical receptive field profiles'. *Vision Research* **20**, 846–856.

Duda, R., P.E. Hart e D.H. Stork (2000). *Pattern Classification*. Wiley Interscience.

Dunn, J. C. (1973). 'A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters'. *Journal of Cybernetics* **3**(3), 32–57.

Fawcett, T. (2006). 'An introduction to ROC analysis'. *Pattern Recognition Letters* **27**, 861–874.

Fernandes, B. J. T. e G. D. C. Cavalcanti (2008). 'A pyramidal neural network based on non-classical receptive field inhibition'. *20th IEEE International Conference on Tools with Artificial Intelligence* pp. 227–230.

Fernandes, B. J. T., G. D. C. Cavalcanti e T. I. Ren (2008). 'Classification and segmentation of visual patterns based on receptive and inhibitory fields'. *Eighth International Conference on Hybrid Intelligent Systems* pp. 126–131.

Fisher, R. (1936). 'The Use of Multiple Measurements in Taxonomic Problems'. *Annals of Eugenics* **7**, 179–188.

Fix, E. e J. Hodges (1989). 'Discriminatory analysis. nonparametric discrimination: Consistency properties'. *International Statistical Review* **57**(3), 238–247.

Frery, C., H. Muller, C. Yanasse e S. Sant'Anna (1997). 'A model for extremely heterogeneous clutter'. *IEEE Transaction on Geoscience and Remote Sensing* **35**(3), 648–659.

Fukushima, K. e N. Wake (1991). 'Handwritten alphanumeric character recognition by the neocognitron'. *IEEE Transactions on Neural Networks* **2**(3), 355–365.

Gabor, D. (1946). 'Theory of Communication'. *Journal of the Institute of Electrical Engineers* **93**(26), 429–457.

Gonzalez, R. C. e R. E. Woods (2007). *Digital Image Processing*. Prentice-Hall.

Grigorescu, C., N. Petkov e M. A. Westenberg (2003*a*). 'Contour detection based on nonclassical receptive field inhibition'. *IEEE Transactions on Image Processing* **12**(7), 729–739.

Grigorescu, C., N. Petkov e M. A. Westenberg (2003*b*). 'The role of non-crf inhibition in contour detection'. *Journal of Computer Graphics, Visualization, and Computer Vision* **11**(2), 197–204.

Guarnieri, A. e A. Vettore (2002). Automated technique for satellite images segmentation. In 'Symposium on Geospatial Theory, Processing and Applications'.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. IEEE Press.

Heisele, B., T. Poggio e M. Pontil (2000). Face detection in still gray images. Technical Report 1687. Center for Biological and Computational Learning, MIT. Cambridge, MA.

Hong, L., Y. Wan e A.K. Jain (1998). 'Fingerprint image enhancement: Algorithm and performance evaluation'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(8), 777–789.

Horng, M.-H., Y.-N. Sunb e X.-Z. Lin (2002). 'Texture feature coding method for classification of liver sonography'. *Computerized Medical Imaging and Graphics* **26**, 33–42.

Hu, H. (2008). ‘Orthogonal neighborhood preserving discriminant analysis for face recognition’. *Scientific American* (41), 2045–2054.

Hubel, D. H. (1963). ‘The visual cortex of the brain’. *Scientific American* (209), 54–62.

Jain, A. e F. Farrokhnia (1991). ‘Unsupervised texture segmentation using Gabor filters’. *Pattern Recognition* **24**(12), 1167–1186.

Jones, J. e P. Palmer (1987). ‘An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex’. *Journal of Neurophysiology* **58**, 1233–1258.

Jones, M. e J.M. Rehg (2002). ‘Statiscal color models with application to skin detection’. *Int’l J. Computer Vision* **46**(1), 81–96.

Kwak, N. (2008). ‘Feature extraction for classification problems and its application to face recognition’. *Pattern Recognition* **41**, 1701–1717.

Lecun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard e L. D. Jackel (1989). ‘Backpropagation applied to handwritten zip code recognition’. *Neural Computing* **1**(4), 541–551.

Lecun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard e L. D. Jackel (1990). ‘Handwritten digit recognition with a back-propagation network’. *Advances in Neural Information Processing Systems 2* pp. 598–605.

Levine, M. e J.M. Shefner (2000). *Fundamentals of sensation and perception*. Oxford University Press.

Li, W., C.J. Wang, D.X. Xu e S.F. Chen (2004). ‘Illumination invariant face recognition based on neural network ensemble’. *16th IEEE International Conference on Tools with Artificial Intelligence* pp. 486–490.

Lim, J. S. (1990). *Two-dimensional Signal and Image Processing*. Prentice-Hall Signal Processing Series.

Machado, A. (1993). *Neuroanatomia Funcional*. Atheneu.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In ‘Berkeley Symposium on Mathematical Statistics and Probability’. Vol. 1. pp. 281–297.

Makinen, E. e R. Raisamo (2008). ‘Evaluation of gender classifications methods with automatically detected and aligned faces’. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30**(3), 541–547.

Mautner, P., O. Rohlik, V. Matousek e J. Kempf (2002). ‘Signature verification using art-2 neural network’. *Proceedings of the 9th International Conference on Neural Information Processing* **2**, 636–639.

Meurie, C., G. Lebrun, O. Lezoray e A. Elmoataz (2003). 'A supervised segmentation scheme for cancerology color images'. *Signal Processing and Information Technology* **14**, 664 – 667.

Müller, S., R. Q. Feitosa, G. L. A. Mota, D. P. da Costa, V. V. Silva e K. Tanisaki (2003). Geoaida applied to spot satellite image interpretation. In 'Remote Sensing and Data Fusion over Urban Areas'. pp. 220–224.

Nelson, J. I. e B. J. Frost (1978). 'Orientation-selective inhibition from beyond the classic visual receptive field'. *Brain Research* **139**, 359–365.

OpenCV 1.0 (2006). `http://www.intel.com/technology/computing/opencv/`.

Osuna, E., R. Freund e E. Girosit (1997). 'Training support vector machines: an application to face detection'. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* pp. 130–136.

Otsu, N. (1979). 'A threshold selection method from gray-level histograms'. *IEEE Transactions on Systems, Man, and Cybernetics* **9**, 62–66.

Pal, N. R. e S. K. Pal (1993). 'A review on image segmentation techniques'. *Pattern Recognition* **26**(9), 1277–1294.

Paschos, G. (2000). 'Fast color texture recognition using chromaticity moments'. *Pattern Recognition Letters* **21**, 837–841.

Phillips, P., H. Moon, S. Rizvi e P. Rauss (2000). 'The feret evaluation methodology for face recognition algorithms'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(10), 1090–1104.

Phung, S. L., A. Bouzerdoum e D. Chai (2005). 'Skin segmentation using color pixel classification: analysis and comparison'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(1), 148–154.

Phung, S. L. e A. Bouzerdoum (2007). 'A pyramidal neural network for visual pattern recognition'. *IEEE Transactions on Neural Networks*.

Predini, H. e W. R. Schwartz (2008). *Análise de Imagens Digitais: Príncipios, Algoritmos e Aplicações*. Thomson Learning.

Provost, F., T. Fawcett e R. Kohavi (1998). 'The case against accuracy estimation for comparing induction algorithms'. *Proceedings of the Fifteenth International Conference on Machine Learning* pp. 445–453.

Rand, W. M. (1971). 'Objective criteria for the evaluation of clustering methods'. *Journal of the American Statistical Association* **66**(336), 846–850.

Rizzolatti, G. e R. Camarda (1975). 'Inhibition of visual responses of single units in the cat visual area of the lateral suprasylvian gyrus (clare-bishop area) by the introduction of a second visual stimulus'. *Brain Res.* **88**(2), 357–361.

Rosenblatt, F. (1958). 'The perceptron: a probabilistic model for information storage and organization in the brain'. *Psychological Review* **65**(6), 386–408.

Roula, M., A. Bouridane, F. Kurugollu e A. Amira (2002). 'Unsupervised segmentation of multispectral images using edge progression and cost function'. *International Conference on Image Processing* **3**, 781–784.

Rowley, H. A., S. Baluja e T. Kanade (1998). 'Neural network-based face detection'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(1), 23–38.

Rumelhart, D., G. Hinton e R. Williams (1986). 'Learning internal representations by back-propagation'. *Nature* **323**, 533–536.

Russell, S. J. e Peter Norvig (2003). *Artificial Intelligence: A Modern Approach*. Pearson Education.

Samaria, F. e A. Harter (1994). 'Parameterisation of a stochastic model for human face identification'. *Proceedings of the 2nd IEEE Workshop on Applications of Computer Vision* pp. 138–142.

Sharma, M. e S. Singh (2001). 'Evaluation of texture methods for image analysis'. *Seventh Australian and New Zealand Intelligent Information Systems Conference* pp. 117– 121.

Shimabukuro, Y., V. Duarte, M. Moreira, E. Arai, B. Rudorff, L. Anderson, F. Santo, R. Freitas, L. Aulicino, L. Maurano e J. Aragão (2005). Detecção de áreas desflorestadas em tempo real: conceitos basicos, desenvolvimento e aplicacao do projeto DETER. Technical report. Instituto Nacional de Pesquisas Espaciais.

Sin, C. e C.K. Leung (2001). 'Image segmentation by changing template block by block'. *Electrical and Electronic Technology* **1**, 302–305.

Souza, R. (1999). Classificação de imagens sar baseada em uma abordagem simbólica. Master's thesis. Universidade Federal de Pernambuco. Recife, PE.

Spackman, K. (1989). 'Signal detection theory: Valuable tools for evaluating inductive learning'. *Proceedings of the Sixth International Workshop on Machine Learning* pp. 160–163.

Stockham, T. G. (1972). 'Image processing in the context of a visual model'. *Proceedings of the IEEE* **60**, 828–842.

Sung, K.-K. (1996). Learning and Example Selection for Object and Pattern Recognition. PhD thesis. MIT, Artificial Intelligence Laboratory and Center for Biological and Computational Learning. Cambridge, MA.

Swets, J. e J. Monahan R.M. Dawes (2000). 'Better decisions through science'. *Scientific American* **283**, 82–87.

Tamura, H., S. Mori e Y. Yamawaki (1978). 'Textural features corresponding to visual perception'. *IEEE Transactions on Systems, Man, and Cybernetics* **8**, 460–47.

Tateyama, T., Z. Nakao, X.Y. Zeng e Y.-W. Chen (2004). 'Segmentation of high resolution satellite images by direction and morphological filters'. *Fourth International Conference on Hybrid Intelligent Systems* pp. 482– 487.

Triola, M. F. (2005). *Introdução a Estatística*. LTC.

Tsai, F., M.-J. Chou e H.-H. Wang (2005). 'Texture analysis of high resolution satellite imagery for mapping invasive plants'. *IEEE International Geoscience and Remote Sensing Symposium* **4**, 3024–3027.

Turk, M. e A. Pentland (1991). 'Eigenfaces for recognition'. *J. Cognitive Neurosci.* (3), 71–86.

Unnikrishnan, R. e M. Herbet (2005). Measures of similarity. In 'IEEE Workshop on Applications of Computer Vision'. pp. 394–400.

Valois, R. D., W. Yund e N. Hepler (1982). 'The orientation and direction selectivity of cells in macaque visual cortex'. *Vision Research* **22**, 531–544.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.

Venkatalakshmi, K., S. Sridhary e S. MercyShaliniez (2006). 'Neuro-statistical classification of multispectral images based on decision fusion'. *Neural Network World* **16**(2), 97–107.

Venkatesh, Y. e S. K. Raja (2002). 'On the classication of multispectral satellite images using the multilayer perceptron'. *Pattern Recognition*.

Viola, P. e M. J. Jones (2004). 'Robust real-time face detection'. *Int'l J. Computer Vision* **57**(2), 137–154.

Yang, J., Lifeng Liu, Tianzi Jiang e Yong Fan (2003). 'A modified Gabor filter design method for fingerprint image enhancement'. *Pattern Recognition Letters* **24**, 1805–1817.

Yang, M.-H., D.J. Kriegman e N. Ahuja (2002). 'Detecting faces in images: a survey'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**, 34–58.

Yang, M.-H. e N. Ahuja (1999). 'Gaussian mixture model for human skin color and its applications in image and video databases'. *SPIE Storage and Retrieval for Image and Video Databases* **3656**, 45–466.

Zhang, X., Y. G. Maylor e K.H. Leung (2006). 'Automatic texture synthesis for face recognition from single views'. *18th International Conference on Pattern Recognition* **3**, 1151–1154.

Zhang, Y. (1997). Information system for monitoring the urban environment based on satellite remote sensing: Shanghai as an example. In ‘Geoscience and Remote Sensing’. Vol. 2. pp. 842–844.

Zhang, Y. J. (1996). ‘A survey on evaluation methods for image segmentation’. *Pattern Recognition* **29**(8), 1335–1346.

Zhao, H., P. Yuen e J.T. Kwok (2006). ‘A novel incremental principal component analysis and its application for face recognition’. *IEEE Transactions on Systems, Man, and Cybernetics* (36), 873–886.

Zhou, M. e H. Wei (2006). ‘Face verification using Gabor Wavelets and AdaBoost’. *18th International conference on Pattern Recognition* pp. 404–407.

Zucker, S. W. e K. Kant (1981). ‘Multiple-level representations for texture discrimination’. *IEEE Conference on Pattern Recognition and Image Processing* **8**, 609–614.