Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Detecção de Fraudes em Leitores de Impressões Digitais sem Contato Utilizando Descritores de Texturas e Redes Neurais Artificiais

Mateus Mendelson Esteves da Silva

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Orientador
Prof. Dr. Alexandre Zaghetto

Brasília
2015

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Detecção de Fraudes em Leitores de Impressões Digitais sem Contato Utilizando Descritores de Texturas e Redes Neurais Artificiais

Mateus Mendelson Esteves da Silva

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Prof. Dr. Alexandre Zaghetto (Orientador)
CIC/UnB

Prof. Dr. Flávio de Barros Vidal
Universidade de Brasília

Prof. Dr. Marcus Vinícius Chaffim Costa
Universidade de Brasília

Prof. Dr. Ricardo Zelenovsky
Coordenador do Curso de Engenharia da Computação

Brasília, 14 de dezembro de 2015

Dedicatória

Dedico este trabalho ao único Deus, que tem me dado o dom da vida e iluminado os meus caminhos.

Agradecimentos

Agradeço aos meu pais, **Romualdo Silveira da Silva** e **Márcia Esteves Silveira**, por seu amor e apoio incondicionais, além do incentivo diário a perseguir o perfeito e indubitável conhecimento da verdade.

Agradeço, também, ao meu amigo **Vitor Quaresma**, que tem me acompanhado e ajudado nas mais diversas empreitadas acadêmicas e profissionais. Também aos meus amigos de infância **Paulo Rolemberg** e **Diógenes Néris** por me apoiarem nos momentos mais difíceis da minha vida.

Agradeço ao professor **Dr. Alexandre Zaghetto**, que me guiou de forma irrepreensível durante a construção deste trabalho e em demais áreas de cunho acadêmico e pessoal.

Agradeço ao professor **Dr. Flávio de Barros Vidal**, que, com suas dicas e sugestões excelentes, me permitiu encontrar o caminho a ser trilhado em diversos trabalhos acadêmicos ao longo da minha graduação, inclusive neste.

À todos os membros do **Capítulo Estudantil da IEEE RAS na UnB**, do **Projeto Edubot** e do **LARA** (Laboratório de Automação e Robótica), agradeço pelo imenso ganho de maturidade, pelas oportunidades, amizades e prêmios que me foram proporcionados durante minha graduação.

Agradeço, ainda, à equipe de desenvolvimento da BRISA-DF pela compreensão e palavras de incentivo durante a construção deste trabalho, em especial aos colegas **Felipe Lopes**, **Jefferson Chaves** e **Vitor Filincowsky**, que contribuíram fortemente para a minha formação profissional.

Por fim, agradeço a todos os meus colegas do **LISA** (Laboratório de Imagens, Sinais e Áudio) que, de alguma forma, me auxiliaram na aplicação e compartilhamento de seu conhecimento.

Resumo

Com o advento dos sistemas biométricos de impressão digital, surgem também as mais diversas técnicas de ataque que visam enganar a segurança imposta. Quando um atacante investe contra um sistema, três possíveis situações merecem destaque. Em uma delas, o atacante fornece uma impressão digital falsa com o objetivo de se passar por um terceiro. Na outra, o atacante altera sua impressão digital de forma a não ser reconhecido. Ainda, há situações nas quais um usuário insere objetos no leitor, ou seja, não são apresentadas impressões digitais. Para que tais tentativas de fraude sejam combatidas, este trabalho apresenta um método capaz de classificar imagens provenientes de um leitor biométrico de impressões digitais sem contato em categorias que indicam a presença de dedos reais, dedos oclusos e objetos que não são dedos. A classificação é realizada com uma combinação de redes neurais artificiais e os descritores de textura ILBP e GLCM.

Palavras-chave: biometria, antifraude, LBP, ILBP, GLCM

Abstract

Along with the advent of biometric systems, many techniques have been created in order to trick them. Many times, these techniques are performed at the sensor level and, according to the way it is done, they may be classified into 3 subcategories. Two of them are worth mentioning for the purpose of this paper. In one of them, the attacker provides a fake fingerprint to make the system believe that the attacker is a valid user. In the other, the attacker obfuscates his fingerprint so that the system will not be able to identify him. There is also the situation that a user presents an object to the sensor, in which case there is no fingerprint. This paper proposes an anti-spoofing method that classifies images acquired from a touchless fingreprint biometric sensor into three categories: real fingerprint, obfuscated fingerprint or not even a finger. The proposed method makes use of artificial neural networks and a combination of two texture descriptors: ILBP and GLCM.

Keywords: biometrics, antispoofing, LBP, ILBP, GLCM

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

AFIS Automated Fingerprint Identification System. 7, 8, 11, 12

FAR False Accept Rate. 19, 52–54, 57, 59

FFANN Feed Forward Artificial Neural Network. 41

FNF Finger or Not Finger. xi, 43–45, 47–50, 52, 54, 57, 59, 60

FRR False Reject Rate. 19, 52–54, 57, 59

GLCM Gray-Level Co-occurrence Matrix. xi, 1, 2, 31–33, 47, 59

ILBP Improved Local Binary Pattern. xi, 1, 2, 31, 32, 47, 59

LBP Local Binary Pattern. xi, 29–31

RTFI Reflection-based Touchless Finger Imaging. 13, 49

TTFI Transmission-based Touchless Finger Imaging. 13

Capítulo 1

Introdução

A necessidade crescente de monitorar e proteger o acesso à informações e/ou ambientes tem impulsionado grandes esforços em direção aos mais diversos mecanismos de proteção. Um dos mecanismos de proteção mais comuns são os de controle de acesso.

Para que o controle de acesso seja eficaz, é necessário o uso de um sistema capaz de identificar cada usuário e decidir se o mesmo possui as devidas permissões. A identificação, portanto, deve ser feita com base em características que sejam universais (que todos os possíveis usuários possuam), únicas, permanentes (que não variem, possibilitando sempre medições suficientemente próximas) e cuja coleta seja viável.

Surge, então, o uso de impressões digitais como credenciais de acesso, pois são universais, únicas, permanentes e de fácil coleta. Apesar de serem vastamente adotadas, impressões digitais podem ser facilmente obtidas e reproduzidas por terceiros, como veremos no Capítulo 3.

Os sistemas biométricos de leitura de impressões digitais são passíveis de ataques nos quais um usuário tenta se passar por outro ou simplesmente deseja não ser identificado. Tentativas por parte de terceiros de se passar por outro usuário são chamadas de *fraude*. Tentativas de não ser identificado por meio da alteração da própria impressão digital são chamadas de *oclusão*.

Com o objetivo de oferecer uma alternativa para combater engajamentos maliciosos, este trabalho apresenta um método capaz de distinguir se um usuário está apresentando à um leitor biométrico de impressões digitais sem contato um dedo real, um dedo ocluso ou um objeto qualquer.

O método proposto faz uso de uma combinação de dois descritores de textura bem conhecidos no contexto de processamento de imagens, o *Improved Local Binary Pattern (ILBP)* e o *Gray-Level Co-occurrence Matrix (GLCM)*, sendo que os resultados são animadores.

No Capítulo 2, são apresentados os conceitos e jargões adotados na área de biometria em geral e, depois, volta-se para impressões digitais.

O Capítulo 3 classifica os tipos de ataque à sistemas biométricos, maneiras de obter moldes de impressões digitais falsas e algumas técnicas já existentes para a detecção de fraudes, bem como apresenta as métricas mais comumente utilizadas no meio acadêmico para avaliar o desempenho de um método contra ataques.

O Capítulo 4, por sua vez, apresenta as técnicas de processamento de imagens necessárias ao entendimento do método proposto por este trabalho.

No Capítulo 5, são definidos os conceitos e estruturas básicos das redes neurais, que desempenham papel crucial na detecção de tentativas de fraude.

O método proposto é, então, apresentado no Capítulo 6, trazendo as ideias e conceitos apresentados até então para um único contexto. Dessa forma, redes neurais e descritores de textura (ILBP e GLCM) são utilizados em conjunto para que o objetivo deste trabalho seja alcançado.

Os experimentos e os resultados obtidos são apresentados no Capítulo 7. Aqui, os cenários de teste, a descrição da base de dados utilizada, a forma de treinamento das redes neurais e as métricas obtidas são apresentados.

Por fim, o Capítulo 8 reapresenta a linha de pensamento que levou à implementação do método aqui proposto e apresenta futuras atividades a serem desenvolvidas como continuação à este trabalho.

Capítulo 2

Biometria

Este capítulo visa apresentar os conceitos e conhecimentos iniciais necessários ao entendimento dos jargões, classificações e técnicas comumente utilizados no contexto de biometria. O leitor é encorajado a buscar informações complementares nos ítens bibliográficos que o auxiliem no entendimento e/ou aprofundamento do conteúdo aqui apresentado.

2.1 Traços Biométricos

A capacidade de autenticar indivíduos de forma eficiente e precisa tem se tornado um requisito cada vez mais imprescindível aos sistemas computacionais. Seu uso varia desde aplicações comuns ao nosso dia-a-dia (como o desbloquear da tela de um celular por meio de impressão digital) até aplicações em que a segurança no acesso é um ítem crítico (tais como controle de acesso a determinados espaços físicos, informações altamente sigilosas e objetos de alto valor). Sendo assim, é necessário escolher características intrínsecas à cada indivíduo e elaborar técnicas que sejam capazes de extraí-las de forma confiável.

O processo de autenticação pode ser feito por meio de três credenciais, sendo elas: a posse de um objeto específico (como um cartão RFID); o conhecimento de certa informação (como uma senha); ou a presença de certa característica (como uma impressão digital) [21].

Define-se biometria como sendo a identificação automatizada de um indivíduo à partir de suas características comportamentais e/ou fisiológicas que sejam únicas e cuja imitação por um terceiro seja não trivial [9, 3].

Características comportamentais são aquelas relacionadas ao modo de agir de uma pessoa. Já as características fisiológicas são aquelas relacionadas à estrutura física do indivíduo.

A fim de ilustrar as definições acima, os seguintes traços biométricos são assim classificados [29, 4]:

1. Comportamentais
    - Assinatura
    - Marcas de pressão ao escrever
    - Voz
    - Modo de digitar
    - Modo de andar
2. Fisiológicas
    - Impressões digitais
    - Mãos
    - Face
    - Íris
    - Retina
    - Formato dos dedos
    - DNA

É fácil notar que, embora cada uma dessas características seja intrínseca ao indivíduo, um sistema baseado em características comportamentais tende a ser mais vulnerável do que outro que se baseia em características fisiológicas. Traços comportamentais são facilmente observáveis e, muitas vezes, podem ser copiados por outros seres humanos sem grandes dificuldades, pois se tratam de sequências de ações executadas de acordo com o padrão observado. Como exemplo, pode-se citar os corriqueiros e bem conhecidos casos de falsificação de assinaturas. Traços fisiológicos, por sua vez, são características físicas, não aprendidas. Para que se possa burlar tal sistema, é necessário o uso de materiais e técnicas específicos para a confecção das falsificações.

2.1.1 Características de traços biométricos

Diante de tantas características presentes nos seres humanos, é necessário definir os pré-requisitos para que uma simples característica possa vir a ser utilizada como traço biométrico.

Traços biométricos devem possuir os seguintes requisitos [11]:

1. **Universalidade**: todos os usuários do sistema devem possuir tal característica;

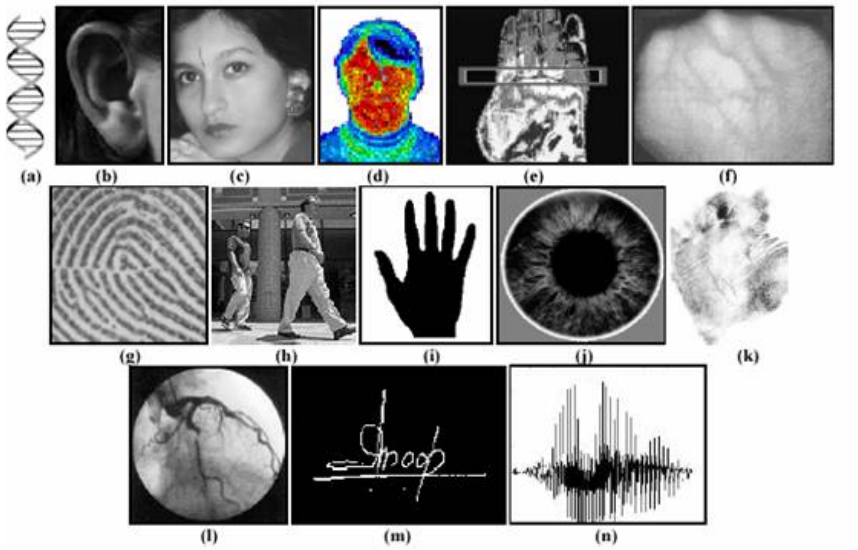

Figura 2.1: Exemplos de traços biométricos, sendo (a) DNA, (b) orelha, (c) face, (d) termograma facial, (e) termograma da mão, (f) veias da mão, (g) impressão digital, (h) modo de andar (marcha), (i) formato da mão, (j) íris, (k) impressão digital da mão, (l) retina, (m) assinatura, (n) espectro vocal. Imagem retirada de [11].

2. **Unicidade**: a característica deve ser única em cada indivíduo, ou seja, quaisquer dois indivíduos não devem ser capazes de apresentar a mesma formação para a característica em questão. Por exemplo, impressões digitais são uma característica que possui formações distintas em cada pessoa.

3. **Permanência**: a característica não deve ser variável, fornecendo sempre resultados suficientemente iguais em medições feitas em quaisquer instantes no tempo.

4. **Viabilidade de coleta**: é necessário que seja possível a obtenção de medições da característica.

Sendo atendidos os requisitos acima, ainda é necessário avaliar se, em termos práticos, a coleta de suas medições pode ser feita de forma cômoda.

Primeiramente, é preciso avaliar a aceitabilidade por parte dos usuários. Devem ser considerados o modo com o qual as medições serão feitas (conforto), questões de privacidade (desejo do usuário de fornecer ou não tais medições), ética, etc.

O quão fácil (ou difícil) seria para que um invasor conseguisse imitar o traço biométrico para burlar o sistema também é um ponto que requer atenção, pois de nada adianta um traço biométrico que pode ser facilmente copiado.

Por se tratar de uma abordagem computacional, é de grande importância avaliar o desempenho do sistema. O tempo tomado durante as medições e no processo de busca e autenticação são fatores de alto impacto, bem como a precisão e os recursos utilizados.

Tabela 2.1: Classificação de traços biométricos de acordo com suas características em alto (A), médio (M) ou baixo (B). Tabela adaptada de [11].

Traço biométrico	**Universalidade**	**Unicidade**	**Permanência**	**Coletabilidade**
DNA	A	A	A	B
Face	A	B	M	A
Impressão digital	M	A	A	M
Modo de andar	M	B	B	A
Geometria da mão	M	M	M	A
Íris	A	A	A	M
Modo de digitar	B	B	B	M
Retina	A	A	M	B
Assinatura	B	B	B	A
Voz	M	B	B	M

Dentre as características aceitas como traços fisiológicos, impressões digitais, face e íris são as mais utilizadas e aceitas nos sistemas biométricos atuais [9], pois estas alcançam níveis aceitáveis nos parâmetros aqui apresentados.

2.2 Impressão Digital

Impressão digital é o nome dado à marca formada pelo conjunto de dobras e vales presentes na ponta de cada dedo (falange). Essa marca é única para cada indivíduo e é imutável, ou seja, mesmo com o passar do tempo esse traço permanece o mesmo. Utilizando os conceitos na Seção 2.1.1, vê-se que se trata de uma característica que possui unicidade, permanência, é universal e de fácil coleta, e, por isso, a impressão digital é um dos traços biométricos mais aceitos e utilizados no mundo. Ela é uma característica tão única que até mesmo irmãos gêmeos possuem formações distintas. Tais marcas podem ser vistas a olho nu se observadas com atenção, como mostrado na Figura 2.2.

Há relatos na literatura de que, com o passar dos anos, algumas digitais podem acabar sendo confundidas como sendo falsas por alguns leitores biométricos. Isso não significa que as impressões digitais tenham se alterado, mas que sua qualidade sofre com o passar do tempo. Tal problema, entretanto, já possui algumas soluções, como pode ser visto em [2].

2.3 Sistemas Biométricos

Sistemas biométricos são os responsáveis por coletar as medições dos traços biométricos que, em conjunto, são utilizados como credencial de acesso e, ainda, decidem se o usuário possui as devidas permissões. Muitas aplicações fazem uso de mais de um traço biométrico

Figura 2.2: Exemplo de impressão digital.

ao mesmo tempo. Recomenda-se que a aquisição das medições dos traços seja realizada em apenas uma única interação com o usuário.

O processo de autenticação de credenciais é dividido em etapas, sendo elas [14, 4]: aquisição de credenciais, pré-processamento das credenciais fornecidas, extração de características (*feature extraction*), associação com as credenciais existentes no banco de dados (*matching*) e tomada de decisão.

O escopo de cada etapa e a ordem nas quais são executadas se definem da seguinte maneira:

1. **Aquisição de credenciais**: esta etapa é iniciada no engajamento[1] e seu objetivo é a obtenção das medições biométricas. Este processo deve ser capaz de obter amostras com qualidade suficiente e, ainda, manter sua viabilidade, ou seja, o conforto do usuário deve ser levado em consideração. Há três principais métodos de aquisição, que serão tratados nas Seções 2.3.3, 2.3.3 e 2.3.3.

2. **Pré-processamento**: esta etapa é opcional. Aqui, o intuito é realizar quaisquer atividades prévias com o intuito de preparar a imagem para o processo de extração de características. Tais atividades podem ser processos de melhoria na qualidade

[1]Defini-se *engajamento* como sendo a ação na qual o usuário interage com o sistema biométrico com o intuito de fornecer credenciais (sejam elas verdadeiras ou falsas).

das medições (filtragem de ruídos), segmentação da área de interesse, obtenção da medição equivalente em 2D, deslocamentos nas posições das medições, etc.

3. **Extração de características**: o processo de extração de características consiste em, à partir das medições obtidas, selecionar e preparar os dados que o algoritmo de associação utiliza pra fazer a autenticação do usuário.

4. **Associação**: consiste em utilizar os dados presentes no banco de dados do sistema de forma a compará-los com os dados fornecidos pela etapa de extração e, assim, calcular uma pontuação que indica o grau de similaridade entre eles.

5. **Tomada de decisão**: de acordo com a pontuação obtida pelo processo de associação, o sistema utiliza um limiar para decidir se o usuário será aceito ou não. Caso a pontuação esteja dentro do limiar, o usuário será aceito; caso contrário, recusado.

Previamente, é necessário que todas as pessoas que utilizarão o sistema façam o cadastramento de suas credenciais. Tal etapa é denominada **credenciamento**.

O processo acima descrito pode ser organizado das mais diversas maneiras, sendo algumas etapas fundidas, separadas ou recebendo nomenclaturas diferentes, variando de acordo com a literatura e objetivos da aplicação. As atividades realizadas, entretanto, permanecem as mesmas, conforme Figura 2.3.

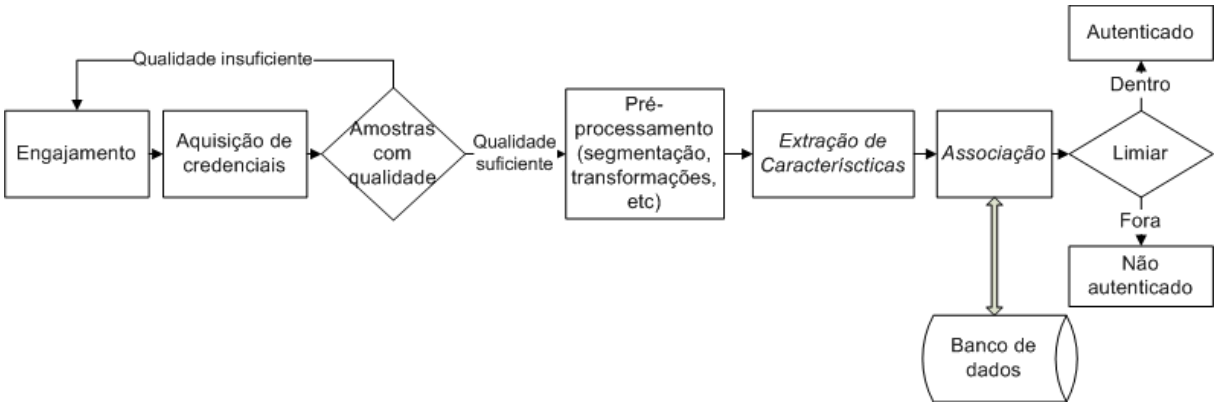

Figura 2.3: Fluxograma básico de um sistema biométrico.

Deste ponto em diante, adotaremos o termo *Automated Fingerprint Identification System (AFIS)* para denotar sistemas de identificação de impressões digitais automatizados.

2.3.1 Protocolos de autenticação

Dependendo da política do sistema, pode ser que apenas a apresentação do(s) traço(s) biométrico(s) cadastrado(s) em seus bancos de dados seja suficiente para a autenticação.

Há aplicações, entretanto, que também exigem saber se o usuário é realmente quem alega ser. Partindo deste ponto, há dois tipos de protocolos que definem quais informações formam a credencial de acesso de um usuário, sendo eles:

- **Verificação (de identidade)**: no momento do engajamento, o usuário deve dizer ao AFIS quem ele alega ser por meio de um PIN (ou seu CPF, ou seu RG, etc.) e, então, o(s) seu(s) traço(s) biométrico(s). O sistema, por sua vez, busca em seu banco de dados o PIN fornecido e checa se as medições biométricas do usuário são compatíveis com as medições relacionadas ao PIN no banco de dados. Dessa forma, o sistema faz uma busca (1:1). Sendo assim, a verificação exige pelo menos duas informações e diminui o custo computacional ao buscar apenas as medições biométricas ligadas ao PIN fornecido.
- **Identificação**: neste caso, no momento do engajamento, o usuário fornece apenas o(s) seu(s) traço(s) biométrico(s). O AFIS, por sua vez, irá buscar em seu banco de dados se alguma de suas medições é compatível com as medições do usuário e retornar uma lista contendo as identificações cujas pontuações ficaram dentro do limiar. Sendo assim, o processamento tende a ser mais custoso, pois a busca é (1:N).

A decisão a respeito de qual protocolo deve ser implementado depende do nível de segurança requerido, dos recursos computacionais disponíveis, tempo de resposta desejado, etc.

Por se tratar do foco deste trabalho, à partir deste ponto trataremos apenas o traço biométrico da impressão digital e suas técnicas.

2.3.2 Agente supervisor

Outra ideia que, apesar de simples, merece atenção é a ausência ou presença de um agente (uma pessoa) que supervisiona as interações de um usuário com o sensor biométrico (seja de forma presencial ou por câmeras). Ambientes que possuem um agente supervisior são chamados de “supervisionados”; ambientes nos quais não há um agente supervisor, “não supervisionados”.

Deve-se destacar que ambientes não supervisionados tendem a ser mais suscetíveis a tentativas de ataque direto. Tal assunto, entretanto, será abordado no Capítulo 3.

2.3.3 Métodos de aquisição de impressões digitais

Aquisição por deslize

Este método exige que o usuário deslize seu dedo sobre a superfície do sensor. Conforme o dedo é deslizado, um sequência de imagens é capturada. Essas imagens são, então, utilizadas para montar uma única imagem da digital completa.

Como a superfície de contato é pequena, o custo deste tipo de sensor tende a ser menor do que o de outros tipos, porém pode apresentar altas taxas de falhas de leitura [26].

Seu uso é muito comum em notebooks e celulares, geralmente apresentando um formato similar ao da Figura 2.4.

Figura 2.4: Leitor biométrico de impressão digital *BIOC-SW* com aquisição por deslize da *Videx Security*.

Aquisição com contato

O processo de aquisição de impressões digitais com contato (mais conhecido como *touch-based fingerprinting*) consiste em pressionar o dedo sobre a superfície do sensor e, dessa

forma, a leitura das medições da impressão digital é realizada. É o tipo de aquisição mais comum nos sistemas atuais [14].

Os dados extraídos neste processo representam a versão 2D da impressão digital, comumente representada em níveis de cinza. Um exemplo pode ser visto na Figura 2.5.

Figura 2.5: Impressão digital obtida pelo método de aquisição com contato, retirada de [17].

Este processo, entretanto, implica em algumas desvantagens, que são inerentes ao modo com o qual o engajamento é realizado.

A pele humana possui um certo grau de elasticidade. Ao realizar o contato entre o dedo e o sensor, a impressão digital sofre deformações. Dessa forma, as medições obtidas pelo sensor podem não ser suficientemente fieis à impressão digital original e sua replicabilidade pode ser afetada. Outras fontes de deformações que também atuam são doenças de pele, umidade do ar, suor, etc.

É importante ressaltar que este método também pode representar uma grande falha de segurança, pois é possível que resíduos do dedo permaneçam na superfície de contato. Dessa forma, torna-se possível que um agente mal intencionado consiga obter uma digital válida à partir destes resíduos, principalmente em ambientes não supervisionados.

Aquisição sem contato

O processo de aquisição de impressões digitais sem contato (mais conhecido como *touchless fingerprinting*) consiste em posicionar o dedo em frente à um sensor que não exige contato. Tal sensor costuma fazer uso de uma (ou mais) câmera(s) de tal forma que as medições da impressão digital sejam obtidas à partir de imagens do dedo.

Devido a ausência de contato físico com a superfície do sensor, este método não deforma a impressão digital e é menos afetado por fatores como sujeira, umidade e demais condições da pele e ambiente.

Duas abordagens são comumente utilizadas para a aquisição sem contato. A aquisição 2D é feita com o uso de apenas uma câmera. A imagem do dedo é capturada por apenas um ângulo e essa imagem é utilizada para extrair a impressão digital do dedo. O ítem *a)* na Figura 2.7 seria a aquisição realizada. É importante notar que este método pode acabar não capturando toda a digital, pois apenas um ângulo do dedo é utilizado.

Também há a aquisição multivista, que, comumente, é feita com o uso de três ou cinco câmeras posicionadas ao redor do dedo (esquema na Figura 2.6). Cada câmera faz a captura da imagem do dedo por um ângulo diferente e, dessa forma, a impressão digital é capturada por completo. Essas imagens são processadas e as partes da digital capturadas em cada imagem são sobrepostas de forma que a impressão digital completa é reconstruída. Figura 2.7 mostra um exemplo de conjunto de imagens que formam uma aquisição feita por um sensor sem contato multivista.

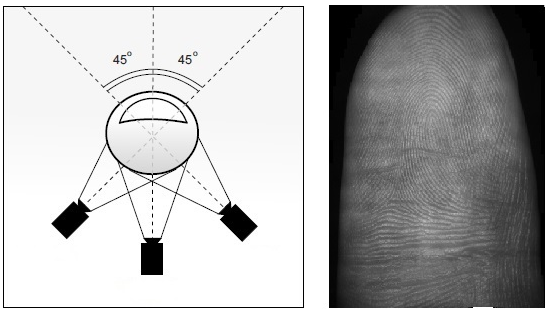

Figura 2.6: Esquema de câmeras em um sistema de aquisição sem contato multivista, à esquerda, retirada de [31]. À direita, imagem completa do dedo que o sistema obtém ao combinar as aquisições de suas câmeras..

As abordagens mais utilizadas no processo de captura das imagens do dedo são descritas abaixo.

Sistema legado: sabe-se que o método de aquisição mais utilizado nos AFISs é o com contato. Ao criar um novo método de aquisição (sem contato), torna-se inconveniente ter

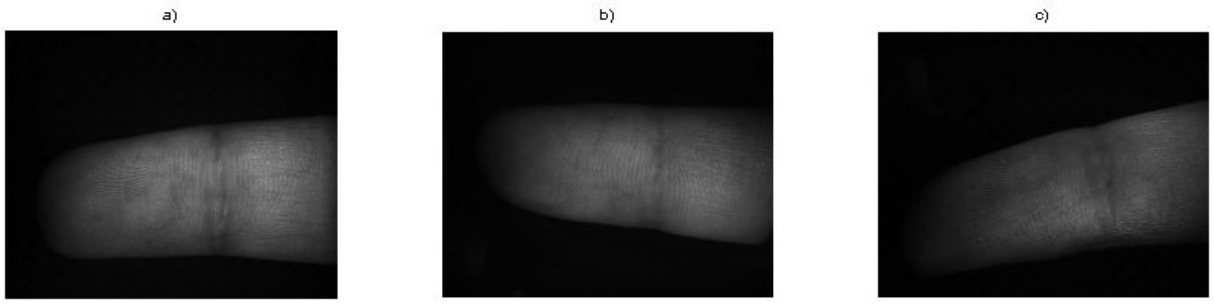

Figura 2.7: Aquisição realizada por um sensor sem contato multivista. *a)*, *b)* e *c)* representam imagens obtidas por cada uma das câmeras.

que recadastrar todas as credenciais já existentes no sistema. Sendo assim, é importante garantir que haja compatibilidade entre ambos.

Ainda, note que, no escopo desta discussão, apenas a etapa de aquisição está sendo alterada, sendo que as outras etapas também devem continuar sendo compatíveis com o novo método de aquisição. Nos AFISs atuais, costuma-se utilizar, nas etapas de extração de características e de associação, algoritmos concebidos para imagens obtidas por sistemas com contato [25].

Com o objetivo de unir essas ideias, AFISs de aquisição sem contato costumam transformar suas representações de impressões digitais em equivalentes 2D de sistemas com contato. Dessa forma, consegue-se conciliar os dados obtidos pelos dois métodos no mesmo sistema. Por exemplo, uma aquisição semelhante à Figura 2.7 seria processada de tal forma a obter uma medição semelhante à Figura 2.8

Além de lidar com os problemas intrínsecos da aquisição com contato, sistemas legados se beneficiam da aquisição sem contato ao passo que este método, de acordo com [25], também reduz a ocorrência de:

- Derrapagem e borramento devido a umidade.
- Contatos impróprios causados por pele seca.
- Acúmulo de sujeira na superfície de aquisição.
- Diminuição da qualidade da imagem adquirida, devido ao desgaste da superfície de contato.
- Erros de medições causados pela diferença de temperatura entre a superfície de contato do dedo e a superfície de contato do sensor.

Imageamento por Reflexão: este método, também conhecido pelo termo em inglês *Reflection-based Touchless Finger Imaging (RTFI)*, baseia-se na maneira com que a luz

Figura 2.8: Impressão digital 2D equivalente de uma aquisição feita por um sensor multivista.

é refletida na superfície de um dedo. Fontes luminosas são colocadas em frente a região da impressão digital em conjunto com sensores (comumente câmeras), que quantificam as reflexões.

Para que seja possível obter medições com constraste suficiente, é necessário que os seguintes requisitos sejam atendidos [25]:

- A pele do dedo deve refletir a maior parte da luz incidente sobre ela, ou seja, a escolha da fonte luminosa e até mesmo do tipo de luz utilizada devem ser tais que a pele humana absorva o mínimo possível.
- As quantidades de luz refletidas pelas dobras e vales devem ser diferentes.
- Fontes luminosas e sensores devem ser posicionados o mais perpendicular possível da superfície do dedo, para que penumbras sejam evitadas e para que os raios refletidos possam ser capturados pelo sensor.

Imageamento por Transmissão: este método, também conhecido pelo termo em inglês *Transmission-based Touchless Finger Imaging (TTFI)*, baseia-se na maneira com que a luz atravessa o dedo.

As fontes luminosas são posicionadas acima da área na qual a unha deve estar e, abaixo do dedo, são posicionados os sensores (comumente câmeras) que quantificam a quantidade de luz que atravessa o dedo. Tal método assemelha-se a colocar uma lanterna acesa em um lado do dedo (fonte luminosa) e observar o lado oposto (os olhos são nossos sensores).

Método	**Vantagens**	**Desvantagens**
Deslize	Sensores que o implementam tendem a ser de menor custo.	Tende a apresentar altas taxas de falhas de leitura.
Com contato	Largamente utilizado, bem consolidado e apresenta menores taxas de falhas de leitura.	Deforma a digital, sensível à doenças de pele e umidade. Permite que um atacante se beneficie dos resíduos deixados pelo dedo sobre o sensor.
Sem contato	Não deforma a digital, resistente à umidade, sujeira e doenças de pele.	Sensores tendem a ser de maior custo.

Figura 2.9: Vantagens e desvantagens de cada método de aquisição de impressões digitais.

Capítulo 3

Ataques à Sistemas Biométricos

A segurança nunca foi um assunto tão em destaque como nos dias atuais e, como é de se esperar, até mesmo sistemas biométricos precisam de ferramentas que os protejam de ataques.

A grande vantagem da utilização de traços biométricos é que eles permitem concluir se alguém é de fato quem diz ser, pois, diferentemente de senhas e cartões, são intrínsecos ao indivíduo. O problema reside na privacidade de tais credenciais. Traços faciais são vistos por todos e podem ser registrados em fotos; a voz também é perceptível à todos e pode ser gravada; uma impressão digital deixa marcas em praticamente todos os objetos que tocamos. Dessa forma, são válidos os esforços direcionados à lidar com esta peculiaridade.

Define-se que o ato de tentar burlar um sistema biométrico é chamado de *ataque* [24]. Um ataque pode ser realizado das mais diversas maneiras e em diferentes pontos do sistema, podendo ser classificado como:

- **Ataque indireto**: este tipo de ataque consiste em uma ação interna ao sistema. O foco do ataque está em nível de software. Alteração em bancos de dados, alteração de *features* capturadas, desvios no fluxo de execução do sistema, etc. são exemplos de ataques indiretos. Como medidas preventivas, é comum o uso de *firewalls*, anti-vírus e encriptação.
- **Ataque direto**: neste tipo de ataque, o atacante interage com o sistema biométrico diretamente através do sensor de aquisição. É uma interação puramente física, não havendo qualquer tipo de alteração no sistema e si. Conforme dito anteriormente, este tipo de ataque é mais comum em ambientes não supervisionados.

Um ataque direto, ainda, pode ser classificado em 3 subcategorias: quando um atacante altera seus traços biométricos para que o sistema não seja capaz de reconhecê-lo, diz-se que ouve *oclusão*; quando um atacante diz ser um usuário válido e simplesmente apresenta seus traços biométricos inalterados, diz-se que houve um *ataque de esforço zero*;

quando um atacante forja um traço biométrico, diz-se que houve um *ataque de apresentação*.

3.1 Fraude e Anti-fraude

O termo *fraude* é empregado como um sinônimo para ataque de apresentação.

Uma definição mais completa, que será aqui adotada, é apresentada por [17], em tradução livre:

> *"Fraude, também conhecida como ataque de apresentação, é um ataque direto realizado no nível de sensor fora dos limites digitais do sistema. Portanto, mecanismos digitais de proteção não são eficazes contra isto. Em uma tentativa de fraude contra um sistema biométrico, um intruso tenta se mascarar como um usuário válido forjando uma amostra biométrica falsa e a apresentando ao sensor biométrico para ser capturada. Anti-fraude (ou detecção de ataque de apresentação) se refere às contramedidas para detectar e impedir tais tentativas. Produtos comerciais de autenticação biométrica sem módulos anti-fraude colocariam a segurança pessoal em risco."*

Sendo assim, tentativas de fraude representam um grande risco, pois não exige que um intruso tenha conhecimentos de programação e suas técnicas são de fácil acesso. Por exemplo, diversas técnicas para forjar impressões digitais podem ser encontradas facilmente na Internet.

Para que se possa forjar um traço biométrico, é necessário obtê-lo de alguma forma. Essa forma pode ser classificada da seguinte maneira [13]:

- **Cooperativa**: a captura é realizada com o consentimento do portador dos traços. A Figura 3.1 mostra um molde de impressão digital obtido desta forma.
- **Latente**: consiste na obtenção dos traços de forma indireta, pela captura de amostras latentes. É o famoso caso da obtenção da impressão digital de terceiros por meio de marcas em copos, maçanetas e outras superfícies.
- **Gravação**: a captura é realizada por meio de gravação em alguma mídia, como fotos, vídeos e gravações de áudio.
- **Regeneração de padrão**: consiste em obter os traços biométricos à partir de amostras incompletas do mesmo. À partir destas amostras, o traço original é obtido.
- **Sintética**: utilização de características sintéticas, não obtidas diretamente de uma pessoa.
- **Personificação**: conversão de traços originários de uma pessoa em traços de outra pessoa (por exemplo, programas de reconstrução vocal).

Figura 3.1: Molde de impressão digital feito de cera de vela derretida e obtido de forma cooperativa.

Após a obtenção dos moldes, são necessárias técnicas capazes de reproduzir os traços biométricos. Elas se classificam em [13]:

- Criação de modelos 3D por meio de negativação do molde.
- Aplicação de material sobre o molde, como, por exemplo, na Figura 3.2 que foi obtida à partir do molde da Figura 3.1.
- Mascaramento por meio da associação de objetos ao molde (tal como a utilização de óculos ou chapeu por cima de uma máscara facial).
- Renderização de molde, seja por meio de impressão 3D, 2D, pintura, etc.
- Reprodução em mídias digitais, como telas de celulares e reprodução de arquivos de áudio.

Como foi possível ver, há inúmeros métodos existentes e muito utilizados em tentativas de fraude. Cada um voltado à um tipo de sistema e de sensor específicos. Por conta disso, também se faz necessário desenvolver inúmeras técnicas para combater fraudes, que recebem o nome de *anti-fraude*.

Basicamente, há 3 maneiras de se combater tentativas de fraude, que podem acontecer durante a aquisição ou durante a extração de características:

- Uso de hardware adicional para identificar se o traço biométrico é real. Esta técnica implica em aumento de custos financeiros.

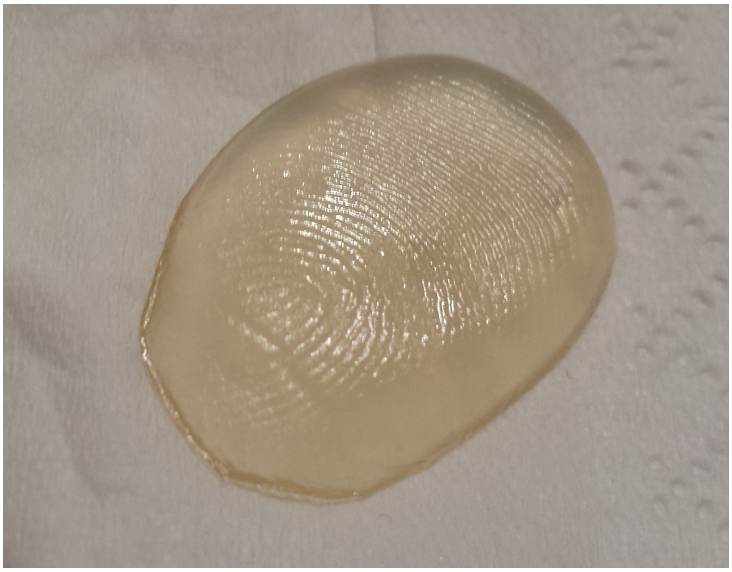

Figura 3.2: Impressão digital forjada com o uso de gelatina, chá verde e água.

- Ao longo do tempo, armazenar novas aquisições de forma a ter uma base de dados maior para poder extrair características de forma mais precisa.
- Utilizar informações de autenticação intrínsecas ao traço biométrico.

As técnicas anti-fraude também podem ser classificadas em 3 grupos, de acordo com sua abordagem:

- **Checagem de propriedades intrínsecas**: tais propriedades são inerentes ao traço biométrico. Pode-se utilizar coloração, textura, formato, umidade, resistência elétrica, dureza, reflectância, etc.
- **Detecção de vida (*liveness detection*)**: um traço biométrico real pertence à um usuário vivo e, para isso, há técnicas capazes de detectar se a aquisição realizada se refere a um dedo vivo. Geralmente, mede-se sinais involuntários (pulso, pressão sanguínea e transpiração), sinais provenientes de reflexos (como a dilatação da pupila e piscar de olhos) ou respostas a desafios (também um piscar de olhos). Na maioria das vezes, métodos de *liveness detection* são implementados em software. Nestes casos, eles são também classificados em outras duas categorias [26]: métodos estáticos, que fazem uso de uma única aquisição para comparar com seus bancos de dados; e métodos dinâmicos, que consistem em realizar múltiplas aquisições de cada vez, podendo tirar proveito de características como poros sudoríparos e elasticidade da pele ao rolar.

- **Detecção de falsificação**: este método busca por marcas de falsificação nas aquisições realizadas e é aferido de forma visual por especialistas. Aqui, a técnica não se baseia no uso de computadores, mas na perícia humana.

3.2 Métricas

Todo e qualquer sistema carece de análises, com o fim de determinar se o seu comportamento é aceitável ou não. Sistemas anti-fraude não são diferentes.

No contexto deste trabalho, serão adotados 3 parâmetros para a análise de um sistema [9].

O primeiro parâmetro, chamado de *Hit Rate*, indica a taxa de acerto do sistema. Após a avaliação de todo o conjunto de teste, aqui é indicada a porcentagem de aquisições que se originaram de dedos reais que foram classificadas pelo sistema como sendo reais e a de aquisições originárias de dedos falsos que foram classificadas pelo sistema como sendo falsas. Indica a taxa de acerto obtida.

O segundo parâmetro, chamado de *False Accept Rate (FAR)*, indica a porcentagem de aquisições originárias de um dedo falso que foram classificadas pelo sistema erroneamente como sendo reais.

O terceiro e último parâmetro, chamado de *False Reject Rate (FRR)*, indica a porcentagem de aquisições originárias de um dedo real que foram classificadas pelo sistema erroneamente como sendo falsas.

Capítulo 4

Processamento de Imagens

Este capítulo tem por objetivo apresentar algumas técnicas de processamento de imagens para que o entendimento do Capítulo 6 seja facilitado. Algumas dessas técnicas, entretanto, podem causar uma certa confusão no leitor e, nesses casos, serão utilizados exemplos que tornarão sua compreensão mais clara.

4.1 Imagens em níveis de cinza

No dia a dia, entramos em contato com, basicamente, dois tipos de imagens: imagens coloridas e imagens "em preto e branco".

Cada valor possível à um pixel representa um nível de cinza. Quantos níveis de cinza são possíveis é algo que depende da quantidade de bits disponíveis para representá-los. Quanto maior for a quantidade de bits, maior é a tendência à sensação de qualidade da imagem (conforme Figura 4.1), pois mais valores podem ser representados e, assim, aumenta-se o nível de detalhamento permitido. A Equação 4.1 mostra como o número de bits e os níveis de cinza se relacionam.

$$L(b) = 2^b \quad , \tag{4.1}$$

no qual b é a quantidade de bits utilizada para representar um nível de cinza e $L(b)$ é a quantidade de níveis de cinza representáveis com b bits.

Apenas imagens em níveis de cinza fazem parte do escopo deste trabalho e, portanto, assume-se o seu uso deste ponto em diante.

Figura 4.1: *(a)* imagem original com 8 bits e 256 níveis de cinza. *(b)* imagem com 4 bits e 16 níveis de cinza. *(c)* imagem com 3 bits e 8 níveis de cinza. *(d)* imagem com 2 bits e 4 níveis de cinza. *(e)* imagem com 1 bit e 2 níveis de cinza.

4.2 Filtro de média

Um filtro de média consiste em substituir o valor de um dado pixel pelo valor da média dos pixels em sua vizinhança, incluindo ele próprio. De forma geral, o cálculo consiste em uma média ponderada. Um filtro de média pode variar de tamanho, de acordo com a necessidade da aplicação em desenvolvimento, e é utilizado para suavizar uma imagem, reduzindo ruídos.

Dado um pixel no ponto p, sua matriz de vizinhança N e a matriz de pesos W, o novo valor para o pixel em p é dado pela Equação 4.2.

$$F(p) = \sum_{p \in N} N(p) \times W(p) \tag{4.2}$$

Tomando como exemplo a Figura 4.2, podemos expandir a Equação 4.2:

$$f(p) = 7 \times \frac{1}{9} + 19 \times \frac{1}{9} + 35 \times \frac{1}{9} + 0 \times \frac{2}{15} + 5 \times \frac{1}{15} + 20 \times \frac{2}{15} + 4 \times \frac{1}{9} + 50 \times \frac{1}{9} + 90 \times \frac{1}{9}$$

$$\implies f(p) \approx 25,8 \tag{4.3}$$

Dessa forma, o filtro substitui o pixel central pelo valor 25,8.

7	19	35
0	5	20
4	50	90

(a)

1/9	1/9	1/9
2/15	1/15	2/15
1/9	1/9	1/9

(b)

Figura 4.2: *(a)* mostra a matriz de vizinhança do pixel central, de valor 5. *(b)* mostra a matriz de pesos.

O filtro do exemplo acima é um filtro de média retangular. Um filtro de média circular, entretanto, consiste no mesmo funcionamento, diferenciando-se apenas pela distribuição dos pesos. Figura 4.3 é um exemplo de filtro de média circular.

0	0	0	0.0012	0.0050	0.0063	0.0050	0.0012	0	0	0
0	0	0.0062	0.0124	0.0127	0.0127	0.0127	0.0124	0.0062	0	0
0	0.0062	0.0127	0.0127	0.0127	0.0127	0.0127	0.0127	0.0127	0.0062	0
0.0012	0.0124	0.0127	0.0127	0.0127	0.0127	0.0127	0.0127	0.0127	0.0124	0.0012
0.0050	0.0127	0.0127	0.0127	0.0127	0.0127	0.0127	0.0127	0.0127	0.0127	0.0050
0.0063	0.0127	0.0127	0.0127	0.0127	0.0127	0.0127	0.0127	0.0127	0.0127	0.0063
0.0050	0.0127	0.0127	0.0127	0.0127	0.0127	0.0127	0.0127	0.0127	0.0127	0.0050
0.0012	0.0124	0.0127	0.0127	0.0127	0.0127	0.0127	0.0127	0.0127	0.0124	0.0012
0	0.0062	0.0127	0.0127	0.0127	0.0127	0.0127	0.0127	0.0127	0.0062	0
0	0	0.0062	0.0124	0.0127	0.0127	0.0127	0.0124	0.0062	0	0
0	0	0	0.0012	0.0050	0.0063	0.0050	0.0012	0	0	0

Figura 4.3: Filtro de média circular de raio 5 gerado pelo MATLAB mostra a matriz de pesos (posição do pixel central em destaque).

4.3 Segmentação

O processo de segmentação de uma imagem consiste em dividi-la em duas ou mais regiões. Tal processo pode ter por objetivo, por exemplo, encontrar objetos específicos, localizar áreas de desmatamento, detectar a presença de pele humana, etc.

Dentre os diversos métodos existentes que realizam segmentação em imagens, apenas um deles compõe o escopo deste trabalho e é chamado de limiarização global simples.

A limiarização global simples consiste em estabelecer uma ou mais faixas de valor que definem o grupo ao qual cada pixel pertence ao realizar a segmentação. Esta técnica possui duas variantes que merecem atenção, descritas a seguir.

4.3.1 Limiar único

A limiarização global simples de limiar único consiste em definir um único valor T que será utilizado para classificar cada pixel de uma imagem em dois grupos: pixels com valor menor ou igual ao T e pixels com valor maior do que T.

Em termos matemáticos, temos

$$s(i) = \begin{cases} 1, & \text{se } i > T \\ 0, & \text{se } i \leq T \end{cases}, \tag{4.4}$$

no qual $s(i)$ é a função que classifica um pixel de intensidade i.

De posse da Equação 4.4, ainda é necessário definir um método que determine qual o valor mais adequado para o limiar T.

O seguinte algoritmo para o cálculo de T é definido por [5]:

1. Escolher o quanto se deseja de precisão para o cálculo de T. Seja ϵ este valor.
2. Escolher um valor inicial para T, que pode ser aleatório.
3. Classificar os pixels da imagem I em dois grupos, conforme a Equação 4.4. Sejam os dois grupos chamados de S_1 e S_0.
4. Calcular o valor da intensidade média dos pixels em S_1 e S_0, sendo eles m_1 e m_0, respectivamente.
5. Guardar o valor antigo de T em T_0 e adotar $T = \frac{m_1+m_2}{2}$
6. Se $|T - T_0| < \epsilon$, então deve-se aplicar o valor de T; caso contrário, retornar ao passo 3.

A Equação 4.4 deve, então, ser aplicada em todos os pixels da imagem utilizando o limiar definido pelo algoritmo acima.

A Figura 4.4 mostra uma imagem segmentada pelo método do limiar único.

4.3.2 Múltiplos limiares

Enquanto o limiar único segmenta a imagem em apenas dois grupos, a segmentação com múltiplos limiares permite a segmentação em mais de dois grupos.

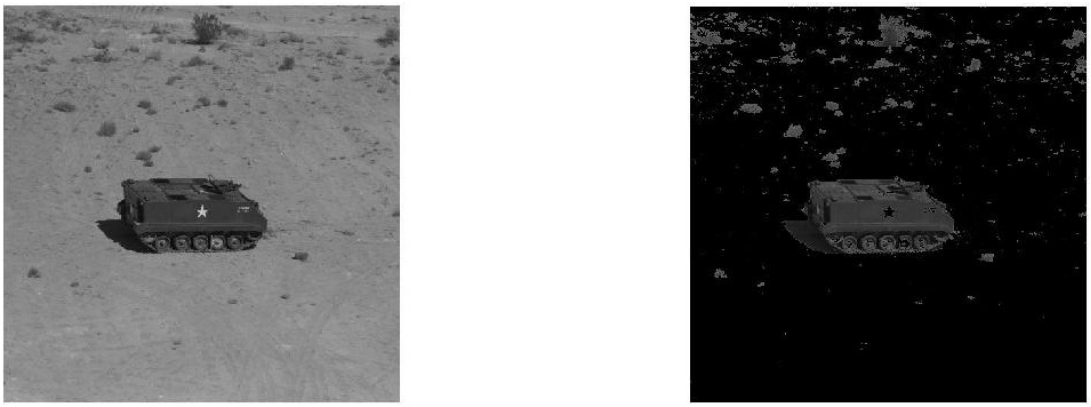

Figura 4.4: *(a)* mostra a imagem original, disponibilizada em `http://sipi.usc.edu/database/`. *(b)* mostra o tanque segmentado. Note que a imagem segmentada carece de tratamentos adicionais para remover objetos indesejados.

Matematicamente, temos

$$s(i) = \begin{cases} c_1, & \text{se } i \leq T_1 \\ c_2, & \text{se } T_1 < i \leq T_2 \\ \vdots & \vdots \\ c_k - 1, & \text{se } T_{k-2} < i \leq T_{k-1} \\ c_k, & \text{se } i > T_{k-1} \end{cases}, \tag{4.5}$$

onde $s(i)$ é a função que classifica um pixel de intensidade i e k indica a quantidade de classes desejadas.

O seguinte algoritmo implementa a classificação descrita pela Equação 4.5 (encoraja-se que o leitor busque aprofundamento caso sinta necessidade em [5]):

1. Escolher o quanto se deseja de precisão. Seja ϵ este valor.
2. Coloca-se k centroides($x_1 \dots x_k$) em posições aleatórias da imagem I.
3. Para cada pixel de I:
   (a) Encontrar o centroide x_j mais próximo.
   (b) Designar o pixel atual à classe c_j
4. Para cada classe c_j:
   (a) Guardar o valor antigo de c_j em c_{j_0} e calcular a nova localização para c_j = média dos pixels pertencentes ao cluster c_j.

5. Para cada classe c_j, checar se $|c_j - c_{j_0}| < \epsilon$. Se a condição falhar em qualquer uma das classes, retornar ao passo 3; caso contrário, a classificação está concluída.

4.4 Operação morfológica

Algumas imagens, após a etapa de segmentação, apresentam objetos e falhas indesejados. Tais artefatos devem ser eliminados.

O processo morfológico aqui aplicado consiste em analisar a vizinhança de um pixel e decidir se este deve ser incorporado à classe de interesse (caso ainda não faça parte), se deve ser removido (caso faça parte) ou se deve permanecer na classe em que já está.

No contexto deste trabalho, considere que, apenas nesta seção, as imagens são binárias (os pixels assumem apenas os valores 0 ou 1). Sendo assim, o objeto de processamento da operação morfológica é uma imagem binária segmentada na qual os pixel de valor 1 indicam a área de interesse (classe de interesse).

Esta técnica segue o seguinte algoritmo:

1. Adotar um valor d que indique a largura do quadrado que definirá a vizinhança de cada pixel.
2. Para cada pixel da imagem I:
   (a) Contar as ocorrências de 0's e 1's dentro do quadrado, incluindo o próprio pixel em análise.
   (b) Se a ocorrência de 1's for maioria, o pixel central será substituído por 1; caso contrário, por 0.

A Figura 4.5 apresenta dois exemplos da operação morfológica com uma vizinhança de dimensões 3x3.

Para efeito de comparação, observe a imagem na Figura 4.4 *(b)* e veja que o resultado obtido na Figura 4.6 diminuiu os ruídos, suavizou a imagem e removeu considerável quantidade de artefatos indesejados, além de preencher os "buracos" antes existentes na da área do tanque.

4.5 Histograma

Um histograma de uma imagem I com L níveis de cinza é definido pela Equação 4.6, como adota em [5].

$$h(r_k) = n_k \quad , \tag{4.6}$$

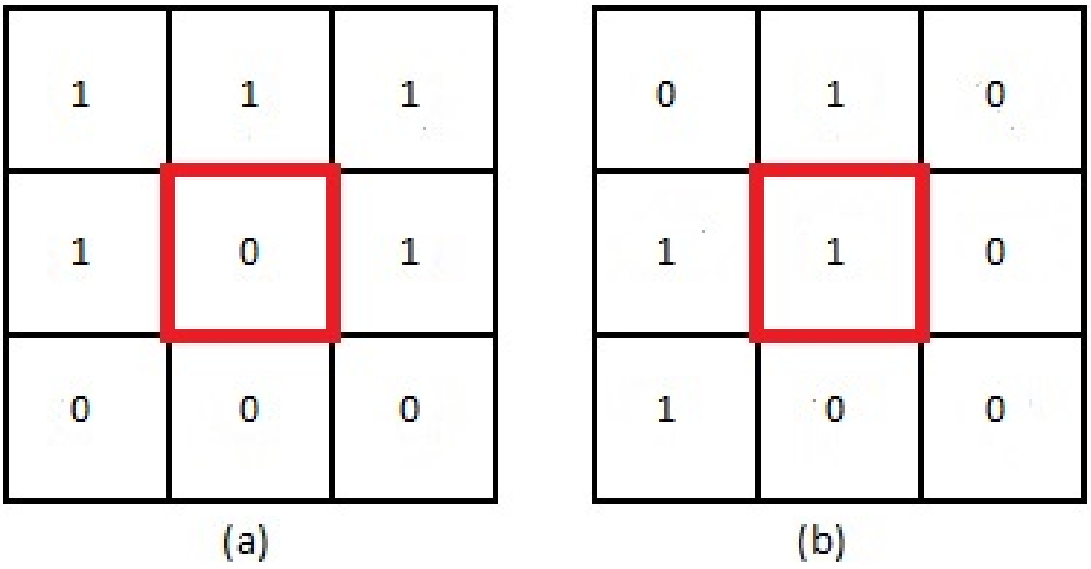

Figura 4.5: Matrizes de vizinhança de um pixel central, que *(a)* assumirá o valor 1 após a operação morfológica. *(b)* assumirá o valor 0 após a operação morfológica. Pixel central, à ser substituído, em destaque.

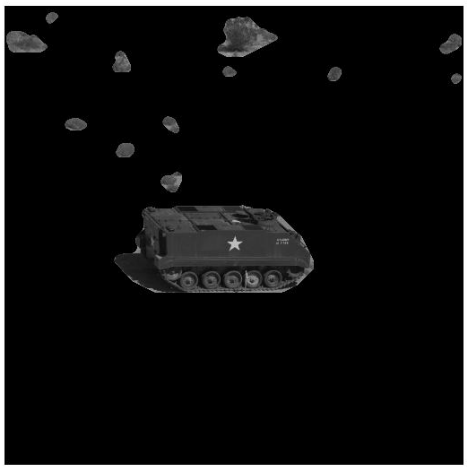

Figura 4.6: Resultado da operação morfológica realizada após a segmentação.

onde $0 \leq k \leq L-1$, r_k é o k-ésimo nível de cinza e n_k representa a quantidade de pixels em I cujo valor é r_k.

4.5.1 Normalização de histograma

Um histograma puro consiste na apresentação de valores absolutos, o que geralmente dificulta a análise da imagem. O desejado é que o histograma apresente valores relativos,

ou seja, porcentagens/probabilidades. Dessa forma, uma prática comum é normalizar o histograma [5], conversão tal que segue a Equação 4.7.

$$p(r_k) = \frac{n_k}{n} \quad , \tag{4.7}$$

onde $0 \leq k \leq L-1$, r_k é o k-ésimo nível de cinza, n_k representa a quantidade de pixels em I cujo valor é r_k e n é o número total de pixels.

Note que a soma de todos os elementos do histograma normalizado deve resultar em 1, por se tratar de uma distribuição de probabilidade.

4.5.2 Equalização de histograma

A equalização de histograma, por sua vez, é utilizada para realçar a imagem revelando detalhes não vistos (facilmente) na imagem original. Aqui, considera-se que a normalização de histograma já foi realizada.

Para que essa técnica seja alcançada, considere a relação de transformação na Equação 4.8.

$$s = T(r) \quad , \tag{4.8}$$

onde $0 \leq r \leq 1$ representa um nível de cinza e s é o novo nível de cinza correspondente à r.

A função de transformação $T(r)$ deve ser tal que as seguintes premissas sejam atendidas:

- ser monotonicamente crescente em $0 \leq r \leq 1$.
- $0 \leq T(r) \leq 1$ no intervalo $0 \leq r \leq 1$.

A primeira premissa garante que a operação inversa existe e que os valores não serão invertidos na realização da transformação.

A segunda premissa garante que o nível de cinza r será o responsável por definir o intervalo da transformação, ou seja, fica garantido que a função de transformação não irá gerar valores em s que extrapolam os valores em r.

Tomemos como exemplo a função na Figura 4.7, que atende às premissas citadas. Esta operação resulta no aumento do contraste da imagem.

Para demonstrar o efeito da equalização de histograma, observe a Figura 4.8 e compare-a com sua versão realçada, na Figura 4.9.

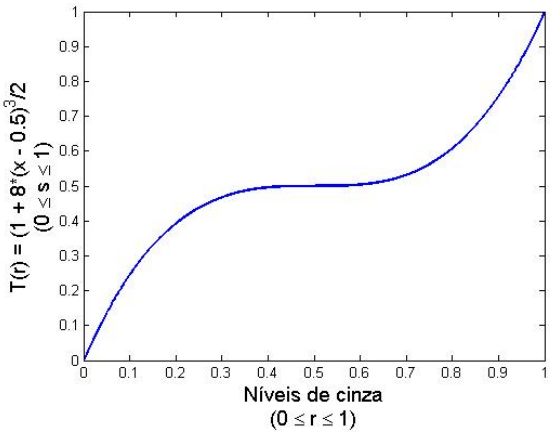

Figura 4.7: Possível função de transformação para equalização de histograma.

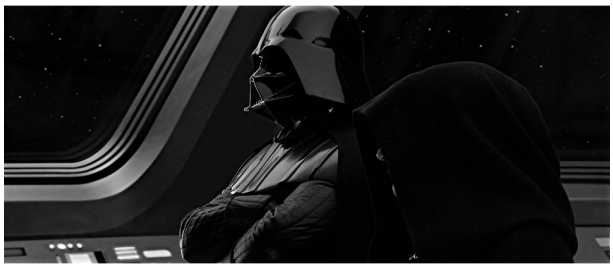

Figura 4.8: Imagem original em níveis de cinza.

Figura 4.9: Imagem em níveis de cinza após a aplicação da equalização de histograma.

4.6 Descritores de Texturas

Descritores de texturas são algoritmos que extraem características de certa área de uma imagem que são capazes de fornecer informações/descritores que a diferenciam de outras imagens. Para o contexto deste trabalho, duas premissas devem ser atendidas:

- O descritor deve ser invariante na escala de cinza.
- O descritor deve ser invariante na rotação da imagem.

As próximas seções são dedicadas a definição de técnicas de descritores de textura. Sendo assim, a Figura 4.10 será tomada como exemplo visual.

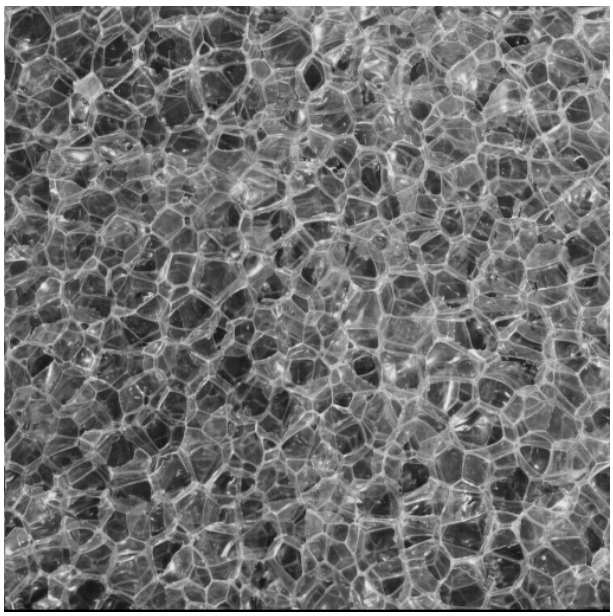

Figura 4.10: Textura de exemplo.

4.6.1 Padrão Binário Local

O Padrão Binário Local, do inglês *Local Binary Pattern (LBP)*, é uma técnica que compõe a classe dos descritores de texturas. Em [23], esta técnica é definida, sendo o algoritmo computacionalmente simples e invariante na escala de cinza, ou seja, que apresenta resultados suficientemente próximos para uma mesma imagem independente de quantos níveis de cinza forem utilizados.

O LBP consiste em analisar a vizinhança de cada pixel de uma região da imagem e gerar um código que a descreve. A equação de cálculo deste código é definida na Equação 4.10.

Para definir o conjunto de pixels considerados como sendo a vizinhança, há diversas técnicas. Aqui, adotaremos sempre a *vizinhança de 8*[1].

Considere g_p como sendo o conjunto de pixels que formam a vizinhança de um pixel central g_c. Para calcular o descritor de textura local de um pixel g_c, segue-se a forma genérica da Equação 4.9.

$$LBP(g_c) = t(s(g_0 - g_c), s(g_1 - g_c), \ldots, s(g_7 - g_c)) \quad , \tag{4.9}$$

[1] A vizinhança de 8 de um pixel p é composta por todos os 8 pixels que tocam as bordas e cantos de p.

sendo que definimos $s(x)$ como:

$$s(x) = \begin{cases} 1, & \text{se } x \geq 0 \\ 0, & \text{se } x < 0 \end{cases} \tag{4.10}$$

Com os valores da Equação 4.10, define-se o valor do descritor de textura local de g_c da seguinte maneira:

$$LBP(g_c) = \sum_{p=0}^{7} s(g_p - g_c)2^p \tag{4.11}$$

Em resumo, o uso de $s(x)$ permite gerar um vetor de números binários que é convertido para um valor decimal. Este valor decimal é o descritor local (código LBP) daquele pixel g_c. A Figura 4.11 exemplifica este cálculo.

De posse do código LBP de cada pixel da área de interesse da imagem, monta-se um histograma referente a ocorrência desses códigos. O histograma normalizado, então, é o descritor de textura fornecido pelo algoritmo LBP. Ele fornece um conjunto probabilístico para cada tipo de textura.

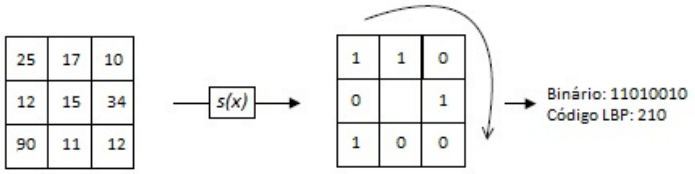

Figura 4.11: Exemplo de cálculo de um código LBP.

Ainda falta um método para garantir a invariância na rotação. A abordagem aqui adotada é bem simples. O vetor de números binários possui seus elementos deslocados de forma circular e, para cada deslocamento, calcula-se o código LBP. O menor código obtido é adotado como o código final daquele pixel.

$$LBP(g_c) = min\left\{\sum_{p=0}^{7} s(g_{(p+i \bmod 8)} - g_c)2^p | i = 0, 1, \ldots, 7\right\} \tag{4.12}$$

Seguindo o exemplo da Figura 4.11 e a Equação 4.12, temos $LBP(g_c) = 00101101_2 = 45_{10}$.

4.6.2 Padrão Binário Local Aperfeiçoado

Dentre as diversas variações derivadas do LBP [18, 8], merece atenção o Padrão Binário Local Aperfeiçoado, do inglês *Improved Local Binary Pattern (ILBP)*.

O ILBP possui as mesmas características do LBP original, porém aquele é capaz de detectar certas nuâncias/padrões [12] que este não consegue.

A única diferença no seu funcionamento em relação ao algoritmo original, é que o ILBP utiliza em seus cálculos a média de toda a vizinhança (incluindo o pixel central) no lugar do g_c. Ou seja, em vez da Equação 4.10, temos a Equação 4.13

$$ILBP(g_c) = \sum_{p=0}^{8} s(g_p - Avg(g_c))2^p \quad , \tag{4.13}$$

onde

$$Avg(g_c) = \frac{g_0 + g_1 + \ldots + g_7 + g_c}{9} \tag{4.14}$$

Desta forma, a Figura 4.11 dá espaço à Figura 4.12.

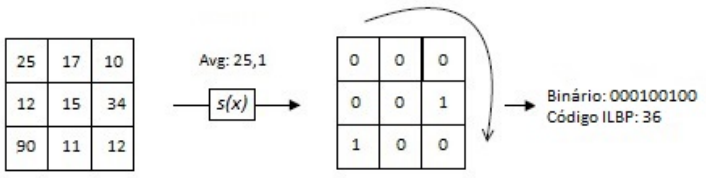

Figura 4.12: Exemplo de cálculo de um código ILBP.

Ainda, a Equação 4.12 é substituída pela Equação 4.15

$$ILBP(g_c) = min\left\{\sum_{p=0}^{8} s(g_{(p+i \bmod 9)} - Avg(g_c))2^p | i = 0, 1, \ldots, 8\right\} \tag{4.15}$$

e temos que $ILBP(g_c) = 000001001_2 = 9_{10}$.

Por fim, a Figura 4.13 mostra um exemplo do histograma do descritor ILBP final.

4.6.3 Matriz de Co-ocorrência de Níveis de Cinza

Outro descritor de texturas é a Matriz de Co-ocorrência de Níveis de Cinza [27], do inglês *Gray-Level Co-occurrence Matrix (GLCM)*.

Este método consiste em criar uma matriz quadrada de tamanho L que indica a quantidade de vezes em que dois pixels são adjacentes em determinada direção e, à partir disso, dados estatísticos podem ser extraídos.

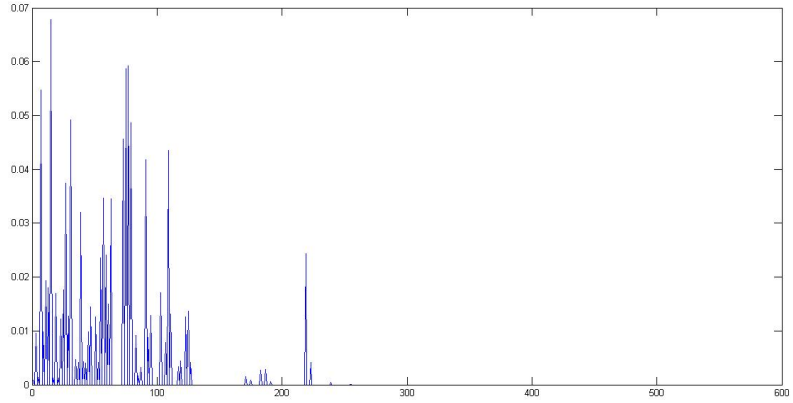

Figura 4.13: Exemplo do cálculo completo do descritor ILBP. As abscissas representam os códigos; as ordenadas, a frequência de ocorrência de cada código.

Para explicar o funcionamento desta técnica, será utilizado um exemplo. Tomando uma imagem como ponto de partida e analisando sempre o pixel à direita, o elemento da GLCM em $P = [i, j]$ representa a quantidade de vezes que um pixel de valor j (chamado de *pixel vizinho*) é adjacente à direita de um pixel de valor i (chamado de *pixel de referência*) na imagem em análise.

A Figura 4.14 mostra que a GLCM armazena em $[1, 2]$ o valor 2, pois, na matriz de entrada, a quantidades de vezes em que um pixel de valor 2 ocorre à direita de um pixel de valor 1 é 2. Note que é possível a existência de posições zeradas, ou seja, a não ocorrência de todos os padrões possíveis não gera nenhum tipo de problema.

É importante lembrar que este método não se limita apenas ao pixel à direita e nem apenas à um único pixel. De acordo com a aplicação, pode-se analisar quaisquer posições, sendo de livre escolha do usuário.

Após a montagem da matriz, há 4 métricas que são comumente extraídas.

Contraste

Medição do contraste entre um pixel e seus vizinhos. É cálculada através de

$$Contraste = \sum_{i,j} |i - j|^2 glcm(i, j) \tag{4.16}$$

Correlação

Medição de quão correlacionado está um pixel em relação a sua vizinhança.

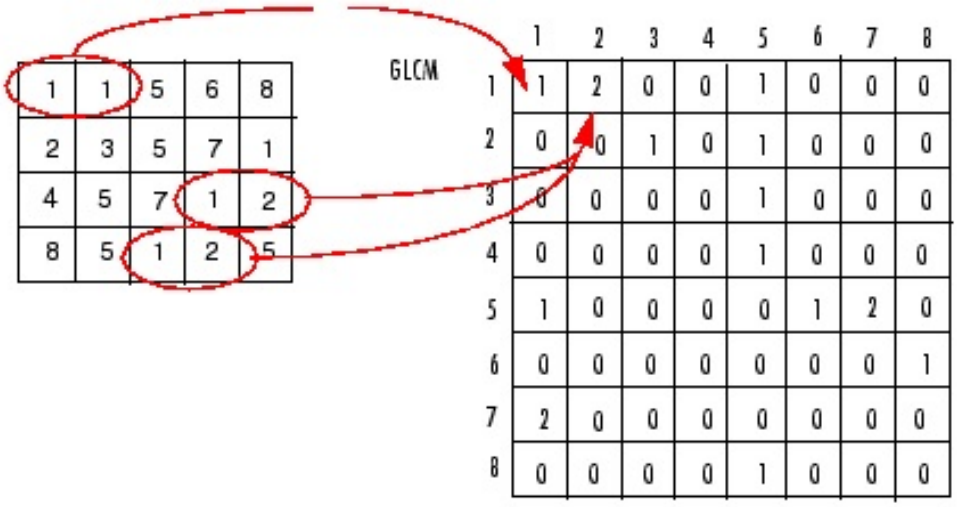

Figura 4.14: Exemplo da construção de uma GLCM. À esquerda, matriz de intensidade de níveis de cinza da imagem original. À direita, matriz GLCM referente à imagem da esquerda. Ilustração retirada de `http://www.mathworks.com/help/images/ref/graycomatrix.html`.

$$Correlação = \sum_{i,j} \frac{(i - \mu i)(j - \mu j) glcm(i,j)}{\sigma_i \sigma_j} \tag{4.17}$$

Energia

É a soma do quadrado de cada elemento da GLCM

$$Energia = \sum_{i,j} glcm(i,j)^2 \tag{4.18}$$

Homogeneidade

Medição da proximidade da distribuição dos elementos na GLCM em relação à diagonal da GLCM.

$$Homogeneidade = \sum_{i,j} \frac{glcm(i,j)}{1 + |i - j|} \tag{4.19}$$

Tais medições probabilísticas podem compor o descritor GLCM. Dependendo do objetivo, pode-se adotar apenas um subconjunto destas medições ou, ainda, adotar outros. Desde que sejam métodos que independam da rotação da matriz, são passíveis de uso.

Retomando a textura de exemplo na Figura 4.10, a técnica de GLCM gera os seguintes descritores:

- Contraste: 504.195398

- Correlação: 0.836115
- Energia: 0.000134
- Homogeneidade: 0.162959

Capítulo 5

Redes Neurais Artificiais

Um dos maiores desafios na computação é a definição de métodos que sejam capazes de classificar elementos em suas respectivas classes. Na maioria das vezes, essa é uma atividade que não oferece grandes dificuldades quando realizada por seres humanos. Entretanto, há situações complexas e também situações que envolvem uma grande quantidade de elementos. O desafio da classificação de elementos veio a se tornar, então, uma atividade desempenhada por computadores.

Há diversas abordagens computacionais voltadas à classificação, porém este ainda é um problema em aberto, não tendo sido encontrada, ainda, uma solução definitiva. Dentre tantas técnicas, uma delas tem se destacado por seu alto nível de aprendizado: as redes neurais artificiais [32].

Redes neurais artificiais surgiram inspiradas nas redes neurais biológicas presentes no sistema nervoso central humano. Tais redes consistem em um arranjo de neurônios interconectados que trocam informações entre si, permitindo a detecção, aprendizado e aplicação de padrões. Redes neurais artificiais são muito utilizadas em áreas ligadas à inteligência artificial, principalmente para resolver problemas de classificação, de ajuste de função (regressão), de robótica, de controle, etc.

Para compreender melhor o funcionamento de uma rede neural artificial, é preciso, primeiro, entender sua estrutura mais básica: o neurônio.

5.1 Neurônios

Assim como em uma rede neural biológica, as redes neurais artificiais são compostas por neurônios, que tentam modelar o funcionamento de um real.

Um neurônio real possui três estruturas básicas: o corpo celular, os dendritos e o axônio. O corpo celular contém o núcleo da célula e é responsável por praticamente toda a produção de proteínas e membranas, bem como a respiração celular. Em suma,

o corpo celular é responsável pelas funções vitais do neurônio. Os dendritos e os axônios são responsáveis por realizar a conexão entre neurônios. Dendritos se conectam apenas à axônios; axônios, apenas à dendritos. Os dendritos são responsáveis por receber os impulsos provenientes de outros neurônios e transmití-los para o corpo celular, que processa essas informações. Tendo processado as informações, o corpo celular excita o axônio, que transmite novas informações aos dendritos do neurônio conectado à ele.

Desta forma, os impulsos nervoso são transmitidos e processados em nosso cérebro. De forma geral, um neurônio possui a estrutura da Figura 5.1.

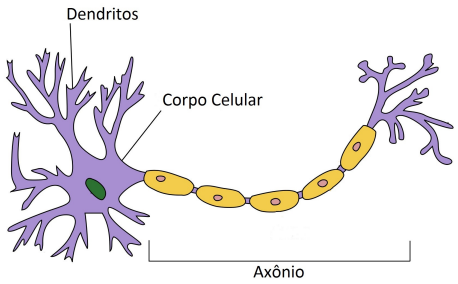

Figura 5.1: Estrutura básica de um neurônio real.

Analogamente, um neurônio artificial tem seu comportamento dividido em três partes, cujos nomes adotaremos da seguinte forma: núcleo, entrada e saída. Entrada e saída são, respectivamente, dendritos e axônio, tendo a função de apenas conectar um neurônio à outro. O núcleo é o responsável por realizar cálculos.

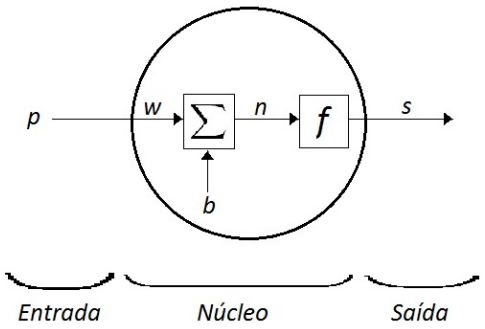

Figura 5.2: Estrutura básica de um neurônio artificial.

A forma com a qual um neurônio define sua saída em função de sua entrada, será melhor detalhada mais adiante. Neste ponto, entretanto, é necessário esclarecer quais tipos de cálculos são realizados em geral. A entrada p de um neurônio pode ser composta por um único valor escalar ou por um vetor de R elementos. Então, cada elemento de p é multiplicado por um peso e, depois, um deslocamento b (sempre um valor escalar)

pode ser adicionado. O conjunto contendo também R pesos é denotado por w. Neste ponto, o resultado é submetido à uma função (que chamaremos de função de ativação) que determina sua saída s (também um escalar). Tomando a Figura 5.2 como referência, temos:

$$n = \left(\sum_{i=1}^{R} p_i \times w_i \right) + b \tag{5.1}$$

De posse de n, podemos definir a saída s como:

$$s = f(p \cdot w + b) = f(n) \tag{5.2}$$

, onde $f(n)$ é a função de ativação, ainda a ser definida.

Há duas variantes predominantes quando se trata de tipos de neurônios artificiais [22, 7]: *perceptrons* e *sigmoides*.

5.1.1 Perceptrons

Os perceptrons são a categoria mais básica de neurônios artificiais. Eles seguem o mesmo modelo básico descrito acima, recebendo R entradas e fornecendo uma única saída s.

O que caracteriza um perceptron é o tipo de sua saída e suas entradas, sendo que s e cada elemento de p sempre assumirão apenas um de dois valores, 0 ou 1. Ao se utilizar este tipo de neurônio, é necessário determinar um parâmetro que terá influência direta em sua função de ativação. Tal parâmetro é indicado por T. Dessa forma, definimos a função de ativação $f(n)$ do perceptron como:

$$u(n) = f(n) = \begin{cases} 1, & \text{se } n > T \\ 0, & \text{se } n \leq T \end{cases} \tag{5.3}$$

Em resumo, um perceptron não faz nada mais, nada menos do que aplicar pesos sobre suas entradas e comparar o resultado com um limiar.

Um exemplo pode tornar seu funcionamento ainda mais claro. Inspirado pela Figura 4.8, considere a situação na qual um imperador do mal se depara com seu filho após muitos anos de busca e, pelas condições nas quais o momento de seu encontro se deu, o imperador deve decidir entre manter o seu império (saída $s = 0$) ou se unir a seu filho, livrando-o da morte (saída $s = 1$).

O imperador faz 3 perguntas a si próprio:

- O seu filho demonstrava algum sinal de que queria estar com ele?
- A queda de seu império era iminente?

- Ele tinha vontade de ter o filho a seu lado?

Ele responde negativamente para as duas primeiras perguntas e positivamente para a terceira. Forma-se, assim, o conjunto de entradas do perceptron: $p = [0\ 0\ 1]$. Além disso, o quão importante é a resposta de cada uma dessas perguntas, em uma escala de 0 à 10? Para o imperador, os pesos são: $w = [5\ 8\ 7]$.

Agora, só resta definir qual o limiar a ser usado. Vamos considerar que um ser humano que tenha o mínimo de amor à família geraria uma saída $s = 12$. Então $T = 12$.

Temos:

$$\begin{aligned} n = \left(\sum_{i=1}^{3} p_i \times w_i\right) + 0 = 0 \times 5 + 0 \times 8 + 1 \times 7 = 7 \\ \implies u(n) = u(7 < T = 12) \\ \implies s = 0 \end{aligned} \tag{5.4}$$

e, portanto, o imperador decide manter o império, deixando seu filho para a morte.

Note que o deslocamento b foi deixado em 0 por simplicidade. Quanto maior for o valor de b, mais fácil será atingir a condição de disparo ($s = 1$). Dessa forma, concluímos que a função dessa constante de deslocamento é simplesmente definir o grau de dificuldade para que sua saída seja 1.

5.1.2 Sigmoides

Apesar da simplicidade dos perceptrons, o mais comum é que outros modelos de neurônios sejam utilizados Saídas puramente binárias acabam deixando um sistema pouco preciso. Seria interessante ter saídas que pudessem ter, por exemplo, o valor 0,5 (no caso do imperador, poderia indicar que ele deveria encontrar um meio termo, mantendo o império e seu filho). Ainda, pequenas mudanças nos pesos poderiam causar uma mudança brusca na saída do perceptron. Buscamos uma solução na qual pequenas variações nos valores dos pesos e do deslocamento causem pequenas variações na saída. Como uma abordagem para tentar resolver esses problemas e prover mais adaptabilidade, surgem os neurônios artificiais sigmoides. Eles se destacam por uma característica em especial: o tipo de valor de suas entradas e saída.

Neurônios sigmoides são capazes de receber entradas de números reais, não se limitando apenas a 0's e 1's, mas podendo receber quaisquer valores dentro deste intervalo. Todavia, permitir valores reais nas entradas ainda não é o suficiente para que uma rede neural possa aprender algo.

Ainda, a função de ativação dos sigmoides não é uma simples função degrau. Ao adotar uma função que permita saídas entre 0 e 1, consequentemente o neurônio também

fornece saídas entre 0 e 1. Finalmente, temos um modelo que nos permite inserir valores reais e que respondem com valores reais.

Define-se a função de ativação deste neurônio da seguinte forma:

$$\sigma(p) = f(p) = \frac{1}{1 + e^{-(\sum_{i=1}^{R} p_i \times w_i) + b}} \tag{5.5}$$

que, para tornar a notação mais compacta, pode ser reescrita como:

$$\sigma(p) = f(n) = \frac{1}{1 + e^{-n}} \tag{5.6}$$

É interessante realizar uma análise rápida do comportamento de $\sigma(n)$ em alguns pontos. Com n positivo e muito grande, temos:

$$\lim_{n \to \infty} \sigma(n) = \lim_{n \to \infty} \frac{1}{1 + e^{-n}} = 1 \tag{5.7}$$

e, para n muito negativo, temos:

$$\lim_{n \to -\infty} \sigma(n) = \lim_{n \to -\infty} \frac{1}{1 + e^{-n}} = 0 \tag{5.8}$$

Nos extremos, o comportamento de $\sigma(n)$ e de $u(n)$ é praticamente o mesmo. Note, então, que a função de ativação sigmoide (também conhecida como log-sigmoide) consiste em uma versão suavizada da função de ativação do neurônio perceptron (Figura 5.3).

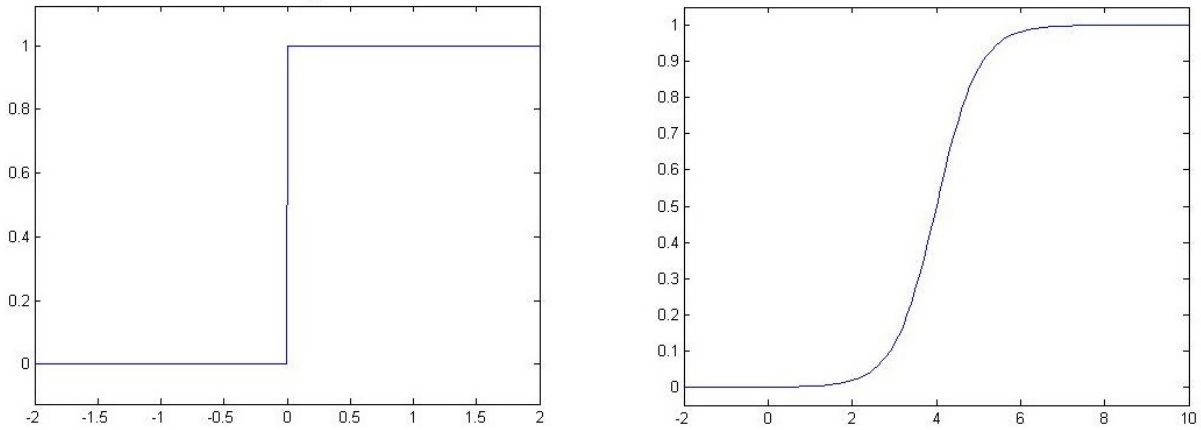

Figura 5.3: À esquerda, função de ativação de um neurônio perceptron; à direita, de um neurônio sigmoide.

Retornando ao exemplo do imperador e considerando que ele tenha alterado suas respostas, podemos dizer que seu filho tinha uma mínima vontade de estar com ele (0,1), que ele acreditava que a queda de seu império seria muito improvável (0,1) e que não queria tanto assim estar com o filho (0,1). Mantendo os pesos e o deslocamento inalterado, temos:

$$
\begin{aligned}
n = \left(\sum_{i=1}^{3} p_i \times w_i\right) + 0 = 0.1 \times 5 + 0.1 \times 8 + 0.1 \times 7 = 20 \\
\implies \sigma(n) = \sigma(20) = \frac{1}{1 + e^{-20}} \\
\implies s = 0.88
\end{aligned}
\tag{5.9}
$$

Provavelmente, o imperado ainda tomaria a mesma decisão, mas com menos convicção.

5.1.3 Funções de ativação

Como já foi possível notar, as funções de ativação são responsáveis por definir a maneira com a qual o neurônio irá tratar as entradas. Se de forma binária, se com variações mais suavez ou mais abruptas. Há diversas funções que podem ser utilizadas como funções de ativação para neurônios artificiais, entretanto, além das já citadas, vale a pena citar mais duas delas, geralmente utilizadas em redes de múltiplas camadas [1]. Note que o importante é o formato das funções de ativação. Ver Figura 5.4.

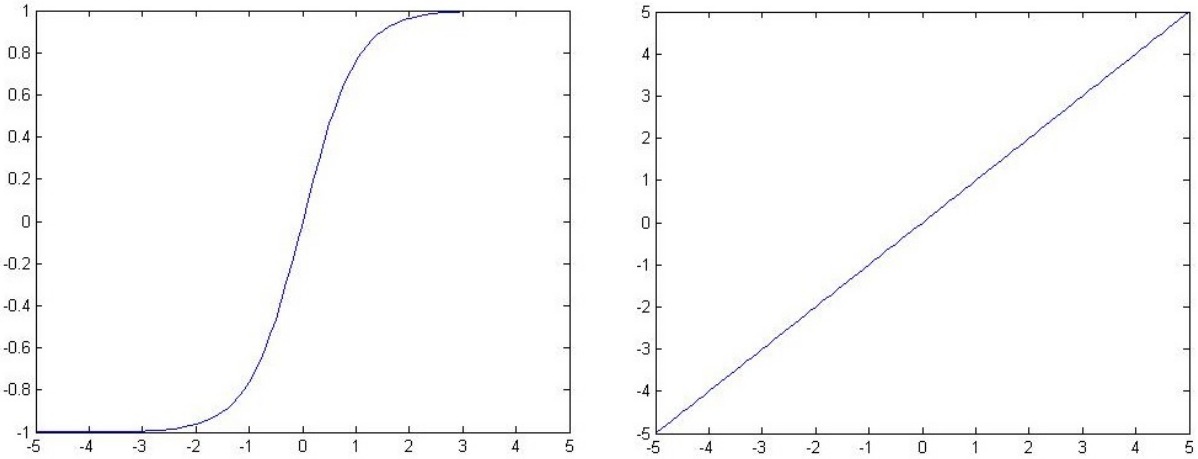

Figura 5.4: À esquerda, função de ativação tangente hiperbólica; à direita, de ativação linear.

Apesar de tudo, é fácil perceber que os neurônios artificiais ainda não chegaram nem perto de reproduzir o comportamento complexo de um neurônio real. Mesmo assim, a modelagem computacional atingida até o presente momento tem demonstrado ser válida para a resolução de problemas que são simples para humanos, mas difíceis para computadores.

5.2 Redes Neurais Artificiais *Feed-forward*

Tendo definido o modelo de neurônio a ser utilizado, uma rede neural é formada por uma combinação de ligações entre neurônios. Na Figura 5.5, podemos ver que há 3 colunas de neurônios. Cada uma dessas colunas formam o que é chamado de camada. Cada camada, com exceção da última, fornece novos valores de entrada para outras. A primeira camada é denominada camada de entrada; a última, de camada de saída; as demais, de camadas ocultas. Redes neurais podem ter uma ou múltiplas camadas.

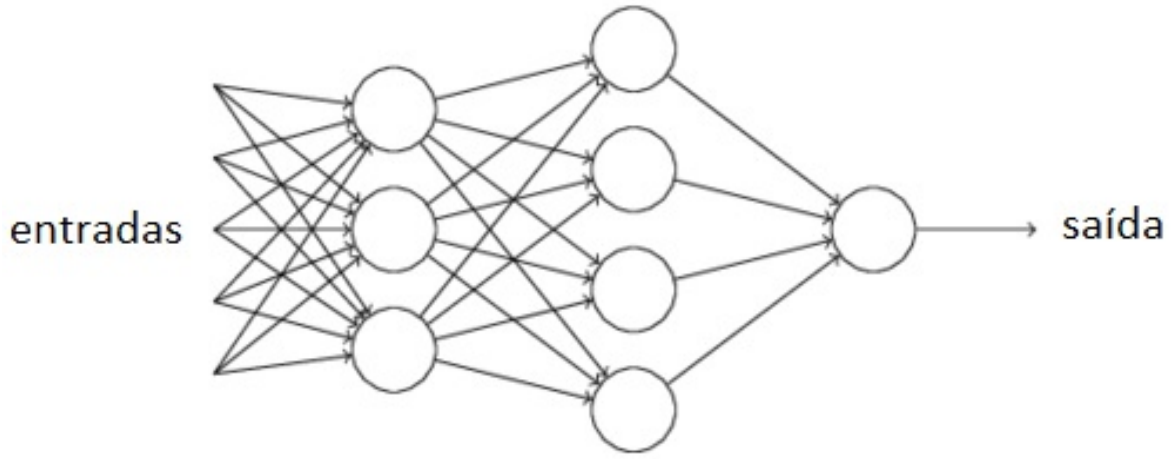

Figura 5.5: Rede neural artificial *feed-forward* de 3 camadas, adaptada de [22].

Por se tratar do escopo deste trabalho, apenas um tipo de rede neural será abordado neste texto.

Uma *Feed Forward Artificial Neural Network (FFANN)* é um tipo de rede neural com as seguintes características:

- Possuem pelo menos duas camadas.
- Cada neurônio de uma camada se conecta com todos os neurônios da camada seguinte e não se comunicam com camadas anteriores.
- Não há conexão entre neurônios de uma mesma camada.

Assim, uma FFANN não possui loops nem realimentação de neurônios. Os dados fluem à partir da entrada da rede sempre em direção à saída. Este tipo de rede é altamente utilizada na resolução de problemas de reconhecimento de padrões.

Há um tipo de FFANN que possui loops, conhecidas como redes neurais recorrentes, porém seus algoritmos de aprendizagem não são tão poderosos [22] e, por isso, não são tão utilizadas.

5.3 Treinamento

Para fazer uso de uma rede neural, é necessário definir 3 conjuntos de dados. Os elementos de cada conjunto devem pertencer somente à este conjunto, não estando presente em nenhum outro. Define-se:

- **Treinamento**: este conjunto deve ser utilizado para calibrar/treinar a rede, definindo os pesos w e deslocamentos b.
- **Validação**: para cada tentativa de treinamento, este conjunto é submetido à análise da rede para que se possa decidir se os valores atuais são ótimos.
- **Teste**: é o conjunto de elementos que realmente serão colocados à prova no sistema, é o conjunto de dados final.

Em outras palavras, o primeiro conjunto treina a rede, o segundo conjunto checa se a rede foi bem treinada e a rede neural é aplicada sobre o terceiro conjunto.

Dois paradigmas de treinamento merecem atenção:

- **Supervisionado**: este paradigma consiste em fornecer os elementos em conjunto com um gabarito. Durante o treinamento, o rede neural tem acesso tanto aos elementos quanto à classe em que eles devem ser classificados.
- **Não-supervisionado**: o paradigma não-supervisionado, por sua vez, somente tem acesso aos elementos e tenta separá-los em diferentes classes para distinguí-los.

Por fim, é necessário definir um método para calcular os valores ótimos para os pesos w e deslocamentos b. Dentro de cada paradigma, há diferentes técnicas de treinamento. Um dos algoritmos mais utilizados para treinamento supervisionado é o *backpropagation*. Ele consiste em tentar reduzir o erro entre as saídas obtidas e as desejadas. Para simplificar, pode-se dizer que isso é feito por meio do cálculo de o quanto o erro varia (gradiente) ao aumentar ou diminuir os pesos. Ao perceber que a taxa está aumentando demais, o treinamento é terminado e os pesos que geraram os menores erros são adotados.

Ainda, há outras técnicas que podem ser utilizadas que podem resultar em redes mais precisas. As técnicas de *Levenberg-Marquardt*, que consiste em uma variação do *backpropagation* mais eficiente [1, 15, 19, 16, 30, 6], e *BFGS Quasi-Newton* são as mais rápidas, apesar de tenderem a consumir mais memória. Para treinamentos com grandes bases de dados, *Scaled Conjugate Gradient* e *Resilient Backpropagation* costumam ser mais eficientes e consomem menos memória [1]. A escolha de qual método utilizar depende de cada caso e deve ser pensada com cuidado.

Capítulo 6

Método Proposto

De posse do conteúdo teórico apresentado nos capítulos anteriores e tomando os termos *credencial* e *impressão digital* como sinônimos, definimos aqui o método desenvolvido.

O método desenvolvido consiste em um algoritmo que pode ser incorporado a um sistema anti-fraude de propriedades intrínsecas que avalia as imagens capturadas por um leitor biométrico multivista de impressão digital sem contato e decide se as aquisições são compostas por um dedo válido, por um dedo ocluso ou por um objeto que não é dedo (tentativa de fraude). Este trabalho não se preocupa em associar credenciais à usuários, mas apenas em decidir se a credencial apresentada é válida ou não. Este algoritmo foi desenvolvido com o objetivo de ser adotado como o ponto de entrada de um sistema biométrico de autenticação de impressões digitais. Para simplificar as referências ao decorrer do texto, este algoritmo recebe o nome de *Finger or Not Finger (FNF)*.

O FNF recebe, como entrada, uma imagem e fornece, como saída um valor que indica se a credencial é válida ou não. Se a credencial não for válida, o sistema é encerrado. Caso a credencial seja válida, o FNF invoca o módulo responsável por realizar a autenticação da impressão digital, de acordo com uma das políticas descritas na Seção 2.3.1.

Utilizando a definição de sistema biométrico apresentada na Seção 2.3, podemos concluir que o FNF pode ser classificado como tal. Ele decide se a credencial apresentada possui permissão para ser processada pelo sistema de autenticação de digitais.

Com o objetivo de tornar o entendimento do FNF mais fácil, a descrição teórica de cada etapa estará acompanhada de seus respectivos resultados, exibidos de forma gráfica. Neste ponto, os resultados não são objeto de análise (o que ocorrerá no Capítulo 7).

6.1 Aquisição

Um leitor biométrico multivista realiza esta etapa. Cada aquisição é formada pelo conjunto de 3 imagens, cada uma representando um ângulo de visão da credencial apresen-

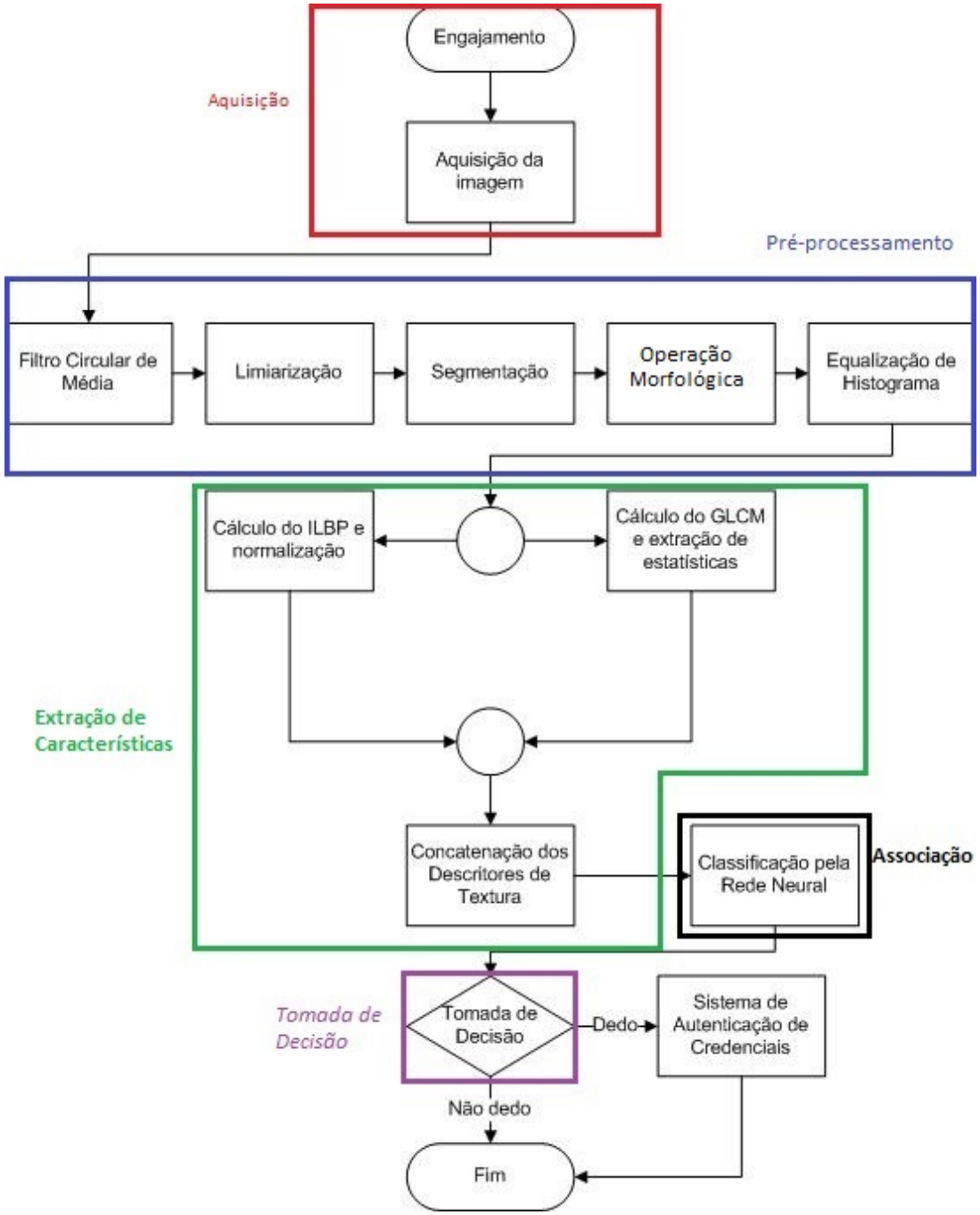

Figura 6.1: Fluxograma em alto nível do *Finger or Not Finger (FNF)*.

tada. É importante ressaltar que o FNF trata cada uma dessas imagens individualmente. A Figura 2.7 apresenta um exemplo de aquisição.

Como exemplo, tomemos a Figura 6.2 como imagem de entrada.

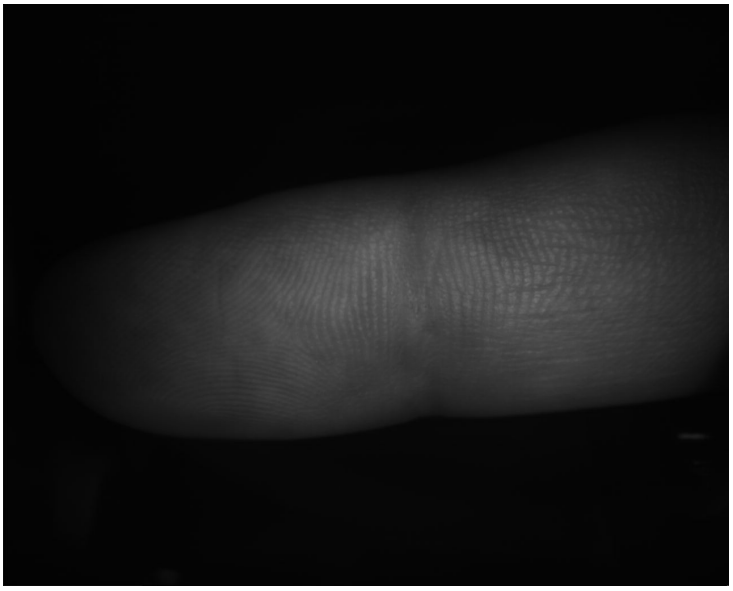

Figura 6.2: Vista lateral que compõe uma aquisição do leitor sem contato multivista.

6.2 Pré-processamento

Como dito anteriormente, esta etapa consiste em manipular a imagem de entrada de forma a prepará-la para as etapas seguintes. O resultado desta fase é uma imagem com a área do dedo segmentada e realçada. O pré-processamento do FNF consiste nos seguintes passos:

1. **Filtro circular de média**: todas as aquisições realizadas apresentam ruídos e, muitas delas, borrões indesejados (como reflexos). Ao aplicar este filtro, a imagem é suavizada e tais artefatos tendem a diminuir.
2. **Limiarização**: à partir da imagem suavizada, é realizado o processo de limiarização global simples de limiar único.
3. **Segmentação**: de posse do valor do limiar global, a imagem é segmentada.
4. **Operação Morfológica**: em certas situações, a aplicação do filtro de média não é suficiente para que todos os artefatos indesejados sejam removidos. Por causa disso, alguns borrões acabam, erroneamente, sendo segmentados como parte da credencial. Em outras situações, ocorre de parte da credencial não ser segmentada, gerando buracos e falhas no meio da área de interesse. Para a maior parte destes casos, a operação morfológica é suficientemente eficaz e remove tais inconvenientes, seja excluindo as áreas de borrões da segmentação ou preenchendo as falhas e buracos.

5. **Equalização de Histograma**: com a imagem já segmentada, é aplicado o processo de equalização de histograma, para que mais detalhes sejam realçados.

Os resultados intermediários desta etapa podem ser vistos na Figura 6.3.

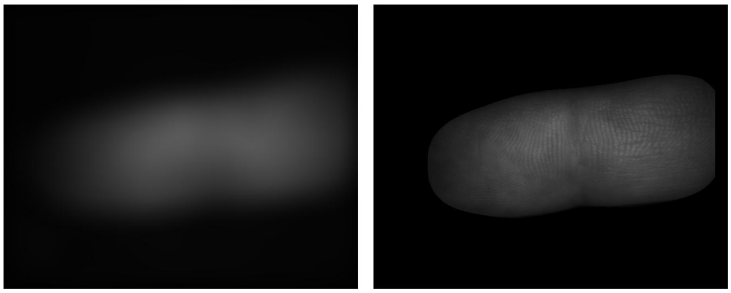

Figura 6.3: Na esquerda, imagem resultante da suavização com o filtro circular de média. Na direita, imagem resultante após a aplicação da operação morfológica.

O resultado final, na Figura 6.4.

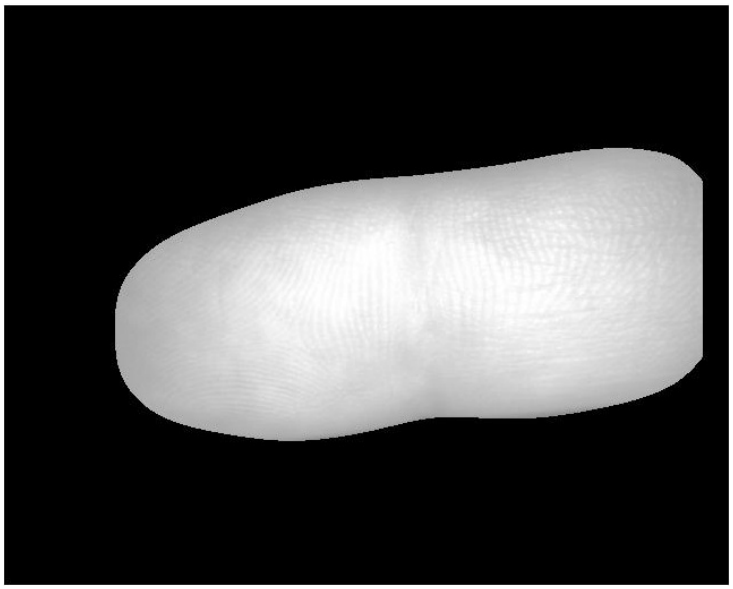

Figura 6.4: Imagem final gerada pela etapa de pré-processamento.

6.3 Extração de Características

Neste ponto, a imagem não é mais alterada. A área de interesse já está devidamente segmentada e com detalhes realçados. Resta, agora, extrair características que sejam capazes de descrever a credencial fornecida.

O FNF é um algoritmo que utiliza descritores de textura como forma de caracterizar credenciais. Sendo assim, ele utiliza uma combinação dos descritores ILBP e GLCM.

Com o objetivo de reduzir o consumo de valiosos recursos computacionais, o FNF apenas realiza o cálculo do ILBP para pixels pertencentes à área de interesse, não calculando nenhum código relativo aos pixels de fundo.

Para o algoritmo do GLCM, é necessário definir quais estatísticas comporão o descritor. O FNF utiliza quatro: contraste, correlação, energia e homogeneidade.

Após o cálculo de cada um dos descritores, a forma adotada para combiná-los consiste na concatenação de seus valores. Dessa forma, o vetor de características possui a seguinte forma:

$$descritor = [a_1 \quad \dots \quad a_{512} \quad b_1 \quad b_2 \quad b_3 \quad b_4] \quad , \tag{6.1}$$

sendo que os elementos de a_1 até a_{512} são compostos pelos códigos ILBP e os elementos b_1, b_2, b_3 e b_4 são, respectivamente, contraste, correlação, energia e homogeneidade relativas à matriz GLCM.

6.4 Associação

Após todas essas etapas, tem-se o resultado final capaz de descrever a textura de uma credencial. Este resultado, então, é utilizado para classificar o tipo de credencial.

Como o descritor de textura é composto por um grande conjunto de valores, uma simples classificação linear não seria suficiente. É necessária uma técnica capaz de identificar quais os padrões que cada tipo de textura segue.

Conforme visto no Capítulo 5, uma rede neural é suficiente para atender as necessidades do FNF. Dessa forma, é adotada uma rede neural com arquitetura do tipo *feed-forward*.

Essa rede neural é composta por 2 camadas, sendo que a camada de saída possui apenas um neurônio. A função de ativação dos neurônios nas duas camadas é a tangente hiperbólica, já que funções de ativação do tipo sigmoide são largamente utilizadas para a resolução de problemas de reconhecimento de padrões [1].

O neurônio de saída provê valores que permitem classificar cada descritor em uma das seguintes 3 categorias: “não dedo”, “dedo ocluso” e “dedo real”. A categoria “não dedo” indica que a credencial apresentada ao leitor não é um dedo, sendo um objeto qualquer; a categoria “dedo ocluso” indica um dedo que foi alterado com o objetivo de não ser reconhecido; a categoria “dedo real” indica que a credencial apresentada é composta por um dedo de verdade.

6.5 Tomada de decisão

Finalmente, a decisão é tomada. Caso a rede neural indique que a credencial não pertence à categoria "dedo real", a decisão tomada é que aquela credencial não possui permissão para ser processada pelo sistema de autenticação de digitais e o sistema é encerrado. Caso a credencial pertença à classe "dedo real", a aquisição realizada pelo sensor é fornecida ao sistema de autenticação de digitais, que assume o controle do fluxo de execução.

Sendo assim, o FNF é um sistema biométrico completo. Sua estrutura e divisões podem ser vistas em sua totalidade na Figura 6.1.

Capítulo 7

Experimentos e Resultados

Com o objetivo de avaliar o quão preciso o FNF pode ser na detecção de fraudes, alguns experimentos foram realizados utilizando o algoritmo *Finger or Not Finger (FNF)*.

Alguns objetos foram submetidos à avaliação do FNF. Cada um foi submetido à 10 aquisições e as imagens obtidas (3 por aquisição) compõem o banco de imagens. Para a realização dos experimentos, as aquisições foram realizadas utilizado-se um leitor biométrico de impressão digital sem contato multivista com 3 câmeras que utiliza RTFI. Os objetos utilizados nas aquisições, conforme na Figura 7.1, foram: fita adesiva (enrolada em dedo real para causar obfuscação), vela (suas duas extremidades foram capturadas), um lápis de escrever (duas extremidades capturadas), um removedor de clip (ambas as extremidades capturadas), uma pilha, um cartão de papel enrolado, um pedaço de copo plástico, um conjunto de clips, uma caneta de quadro branco, uma cola de bastão, uma boquilha de metal de saxofone soprano, uma borracha, uma palheta de bambu de saxofone soprano numeração 2 1/2, a parte de baixo de uma chave de fenda e uma embalagem de corretor.

Além destes objetos, mais um tipo de obfuscação foi submetido (também com 10 aquisições): um dedo com corretor (conforme Figura 7.2).

Por fim, foram capturadas mais 300 imagens de dedos reais. O banco de imagens utilizado no experimento, desta forma, é composto por 840 imagens: 300 imagens de dedos reais; 60 imagens de dedos oclusos; 480 imagens de não dedos.

7.1 Treinando as Redes Neurais

O FNF foi posto à prova na avaliação de 4 cenários, com o objetivo de avaliar sua capacidade discriminante de forma mais controlada. Para cada um destes cenários, uma rede neural específica foi treinada e somente as imagens pertencentes às classes participantes foram utilizadas. São eles:

Figura 7.1: Conjunto de objetos utilizados.

1. Classificação das imagens entre as classes “não dedo” e “dedo real”, deixando a classe “dedo ocluso” de fora.
2. Classificação das imagens entre as classes “dedo ocluso” e “dedo real”.
3. Classificação das imagens entre as classes “não dedo” e “dedo ocluso”.
4. Classificação das imagens entre as classes “não dedo” e “dedo real”, porém utilizando todo o banco de imagens. Neste caso, a classe “dedo ocluso” é incorporada à classe “não dedo”. É o cenário mais próximo à um cenário de uso real que o FNF suporta.

Conforme descrito na Seção 5.3, o banco de imagens, em cada cenário, foi dividido de forma homogênea em três conjuntos. O conjunto de treinamento foi utilizado para calibrar as redes; o de validação, para verificar se a rede havia sido bem calibrada; e o de teste, para avaliar seu desempenho. Ainda, conforme a Seção 6.4, todas as redes neurais são compostas por duas camadas (camada de entrada e camada de saída), há apenas um neurônio na camada de saída e todos os neurônios utilizam a função de ativação tangente hiperbólica.

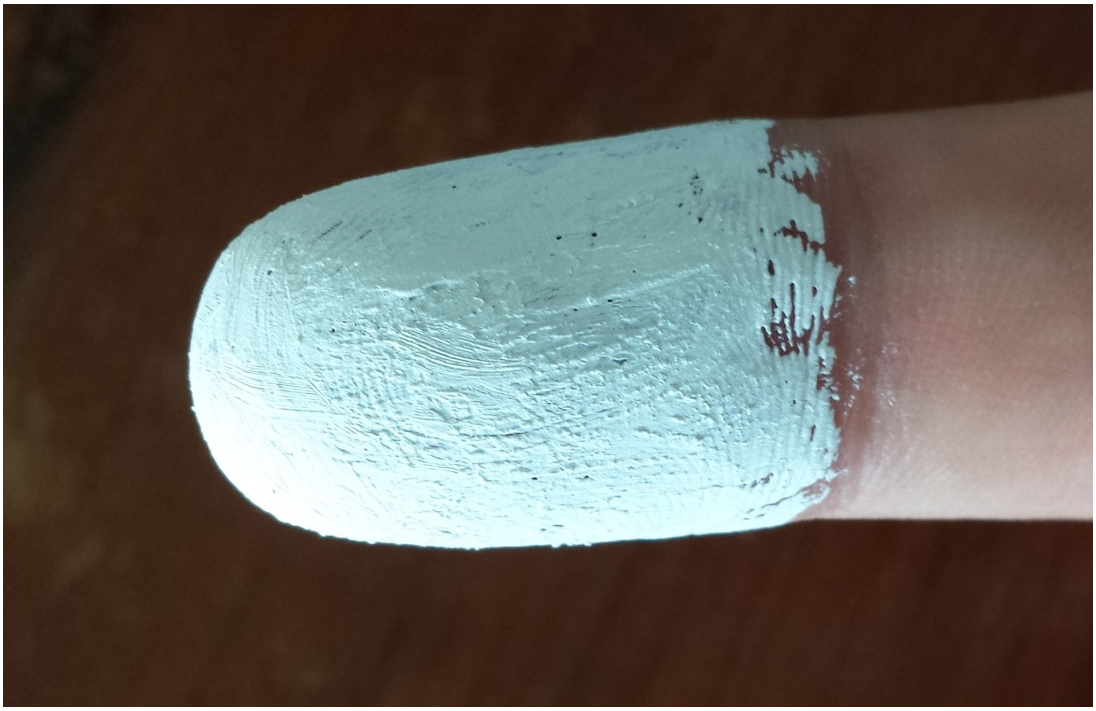

Figura 7.2: Dedo ocluso com o uso de corretor.

O treinamento de cada rede neural é do tipo supervisionado, o método de treinamento utilizado foi o Levenberg-Marquardt, o limite de épocas adotado foi de 1000 e sua realização consiste nas seguintes estapas:

1. A quantidade de neurônios na camada de entrada é 10.
2. Se a quantidade de neurônios na camada de entrada for maior do que 20, deve-se executar o passo 8.
3. Executar os passos de 4 à 6 por 20 vezes.
4. A rede neural é inicializada com valores aleatórios para os pesos w e deslocamentos b.
5. Realizar o treinamento da rede conforme já foi descrito.
6. Caso a rede treinada obtenha o melhor desempenho para o conjunto de teste até o momento, a rede é guardada.
7. A quantidade de neurônios na camada de entrada é incrementada em 1 e o passo 8 é retomado.
8. Adotar a rede neural que obteve melhor desempenho para o conjunto de teste.

Tabela 7.1: Resultados obtidos com a aplicação do conjunto de teste sobre as redes neurais já treinadas em cada cenário.

Cenário	FAR	FRR	Taxa de Acerto
"não dedo" e "dedo real"	0,75%	1,13%	98,11%
"dedo ocluso" e "dedo real"	0	0	100%
"não dedo" e "dedo ocluso"	0	1,09%	98,91%
"não dedo", "dedo ocluso" e "dedo real"	0,75%	1,13%	98,11%

É importante ressaltar que a razão para a execução dos passos de 4 à 6 por diversas vezes se dá com o objetivo de atingir o menor erro global. A escolha relativa à avaliação de 10 à 20 neurônios na camada de entrada foi feita de modo empírico, sendo esta a faixa que apresentou os melhores resultados.

7.2 Resultados

Após a divisão do banco de imagens e da calibração adequada das redes neurais, o conjunto de teste foi utilizado para a geração dos resultados aqui apresentados.

Nas imagens que serã apresentadas, note que os gráficos de regressão indicam os resultados obtidos nas classificações realizadas pelas redes neurais. O eixo das abscissas indica a classificação correta/esperada de cada elemento. O eixo das ordenadas representa a classificação realizada pelo FNF. Os valores de cada classe serão definidos logo adiante. Cada gráfico representa as classificações realizadas pelo FNF para cada grupo (de treinamento, validação e teste), sendo que o ideal seria que todos os pontos se localizassem sobre a reta pontilhada, ou seja, que a reta de regressão possuísse inclinação $R = 1$. O último gráfico, por sua vez, consiste na avaliação de todo o banco de imagens disponibilizado para o cenário em questão, ou seja, a união dos conjuntos de treinamento, validação e teste. Ainda, também é apresentado o gráfico de evolução da rede neural, indicando o momento em que o treinamento foi concluído.

7.2.1 Primeiro cenário: "não dedo" e "dedo real"

O primeiro cenário foi avaliado com o uso de 160 imagens da classe "não dedo" e 100 imagens da classe "dedo real", culminando em uma rede neural com 17 neurônios na camada de entrada e apresentando uma taxa de acerto de 98,11%, indicando que quase a totalidade das classificações foi realizada corretamente.

Além disso, foram obtidos os valores de FAR = 0,75% e FRR = 1,13%. Dessa forma, 0,75% dos objetos foram classificados como sendo de dedos reais e 1,13% dos dedos reais foram classificados como sendo objetos.

Analisando os gráficos na Figura 7.3, dados classificados no intervalo $[-1; 0]$ indicam a classe "não dedo". Os dados classificados no intervalo $(0; 1]$ indicam a classe "dedo real". Sendo assim, o funcionamento ideal do sistema ocorre quando a linha contínua se sobrepõe à linha pontilhada e todos os dados se posicionam sobre a linha contínua em um dos dois pontos (-1; -1) ou (1; 1). Com a fuga de poucos pontos no gráfico de teste em relação ao comportamento esperado e com a linha contínua se desviando pouco da pontilhada, vê-se que a precisão do algoritmo, para este cenário, é satisfatória.

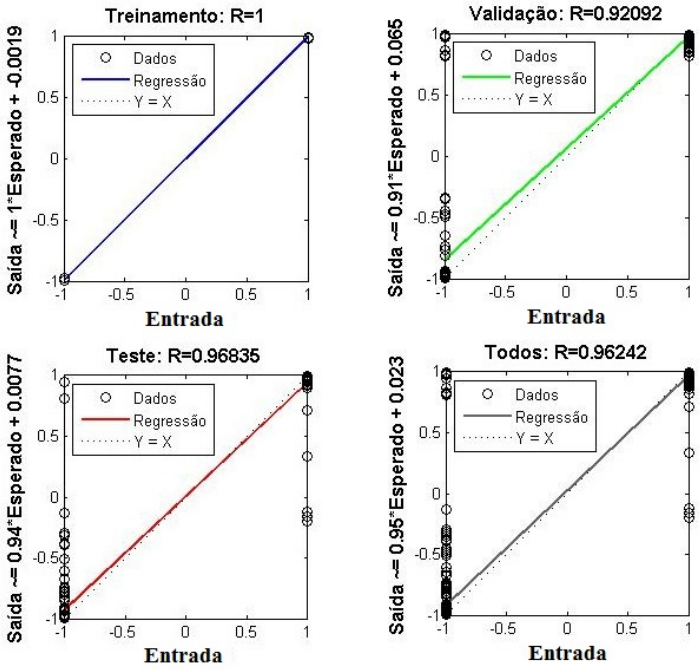

Figura 7.3: Resultados obtidos com a rede neural do primeiro cenário. O eixo das abscissas indica o valor da entrada real; as ordenadas, o valor no qual a rede neural classificou a entrada. Espera-se que a rede neural classifique todos os dados como sendo do mesmo valor da entrada, ou seja, $Y = X$. Neste caso, os valores de saída entre -1 e 0 são classificados como "não dedo". Valores entre 0 e 1, como "dedo real".

7.2.2 Segundo cenário: "dedo ocluso" e "dedo real"

O segundo cenário foi avaliado com o uso de 20 imagens da classe "dedo ocluso" e 100 imagens da classe "dedo real", culminando em uma rede neural com 20 neurônios na camada de entrada e apresentando uma taxa de acerto de 100%, indicando que a totalidade das classificações foi realizada corretamente.

Além disso, foram obtidos os valores de FAR = 0 e FRR = 0. Dessa forma, nenhum dedo real foi rejeitado e nenhum dedo ocluso foi aceito.

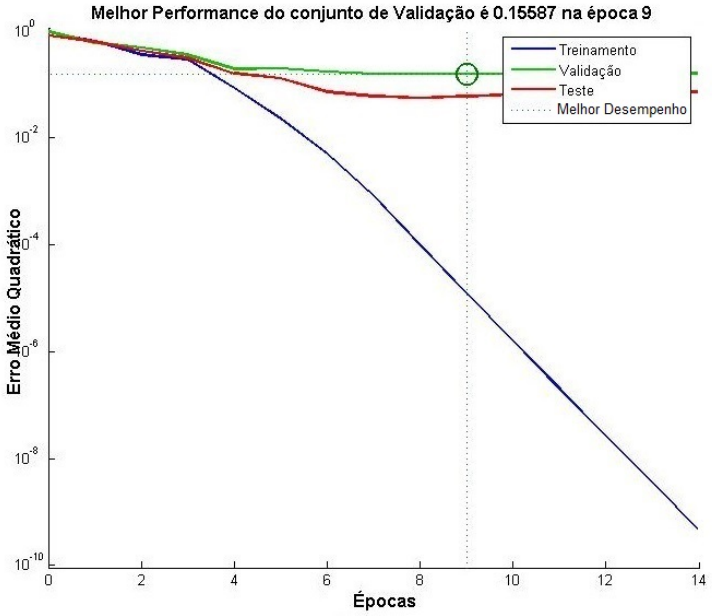

Figura 7.4: Gráfico de evolução da rede neural do primeiro cenário. O ponto sobre a curva de erro do conjunto de validação indica o momento em que o treinamento foi dado como concluído. Neste ponto, a rede neural adota os parâmetros que resultam em menores erros para o conjunto de validação sem que ela fique viciada no conjunto de treinamento. O eixo das abscissas indica a quantidade de épocas transcorridas na evolução da rede; o eixo das ordenadas, o erro médio quadrático sobre o conjunto de dados.

Analisando os gráficos na Figura 7.5, dados classificados no intervalo $[0; 0,5]$ indicam a classe "dedo ocluso". Os dados classificados no intervalo $(0,5; 1]$ indicam a classe "dedo real". Sendo assim, o funcionamento ideal do sistema ocorre quando a linha contínua se sobrepõe à linha pontilhada e todos os dados se posicionam sobre a linha contínua em um dos dois pontos (0; 0) ou (1; 1). Apesar de todos os dados terem sido classificados corretamente, é possível notar que a linha contínua apresenta um pequeno desvio e que os pontos não se localizam exatamente sobre as posições ideias. Isso significa que as classificações, mesmo estando corretas, não são sempre realizadas com a precisão ideal. O comportamento do FNF, para este cenário, é satisfatório.

7.2.3 Terceiro cenário: "não dedo" e "dedo ocluso"

O terceiro cenário foi avaliado com o uso de 160 imagens da classe "não dedo" e 20 imagens da classe "dedo ocluso", culminando em uma rede neural com 11 neurônios na camada de entrada e apresentando uma taxa de acerto de 98,91%, também indicando que quase a totalidade das classificações foi realizada corretamente.

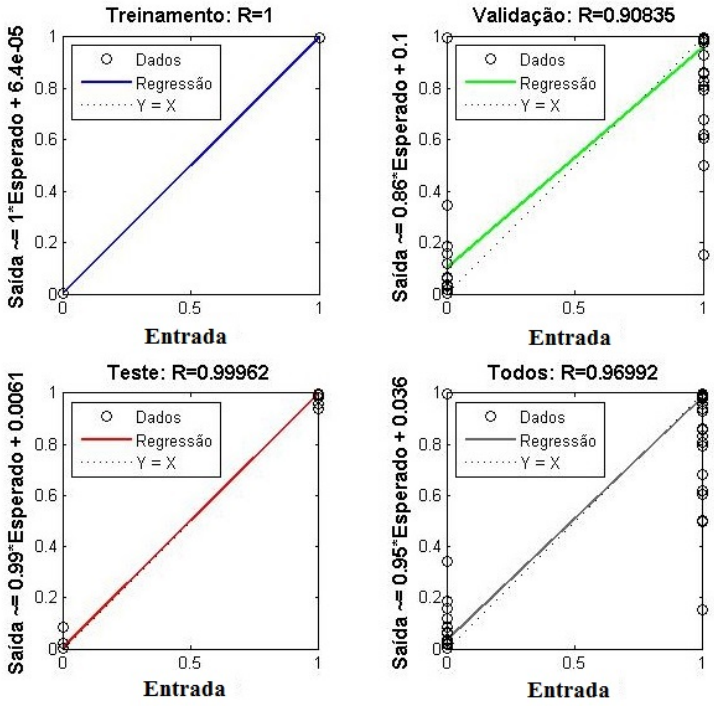

Figura 7.5: Gráficos resultantes do segundo cenário. O eixo das abscissas indica o valor da entrada real; as ordenadas, o valor no qual a rede neural classificou a entrada. Espera-se que a rede neural classifique todos os dados como sendo do mesmo valor da entrada, ou seja, $Y = X$. Neste caso, os valores de saída entre 0 e 0,5 são classificados como "dedo ocluso". Valores entre 0,5 e 1, como "dedo real".

Além disso, foram obtidos os valores de FAR = 0 e FRR = 1,09%, o que indica que nenhum "não dedo" foi classificado como "dedo ocluso". Ainda, 1,09% foram classificados como "não dedo" quando deveriam ter sido classificados como "dedo ocluso".

Analisando os gráficos na Figura 7.7, o funcionamento ideal do sistema ocorre quando a linha contínua se sobrepõe à linha pontilhada e todos os dados se posicionam sobre a linha contínua em um dos dois pontos (-1; -1) ou (0; 0). A classificação, entretanto, considera que dados classificados no intervalo $[-1; -0,5]$ indicam a classe "não dedo". Os dados classificados no intervalo $(-0,5; 0]$ indicam a classe "dedo ocluso". Apesar de a taxa de acerto ser alta, note que a linha contínua se desvia bastante da linha pontilhada e que os pontos se concentra de forma distante às localizações ideais. Como já foi possível notar, o espalhamento dos pontos e a divergência entre as linhas indicam o grau de precisão alcançado. Mesmo com tais distorções, o algoritmo ainda assim é capaz de classificar os elementos em suas respectivas classes de forma satisfatória.

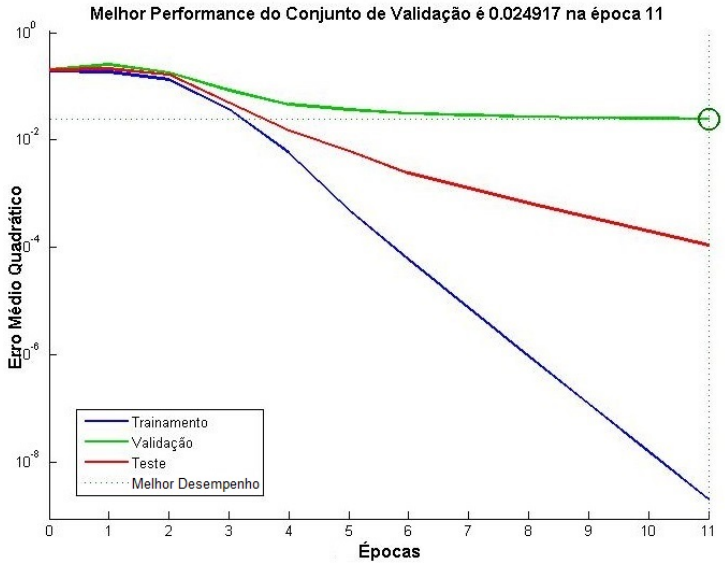

Figura 7.6: Gráfico de evolução da rede neural do segundo cenário. O ponto sobre a curva de erro do conjunto de validação indica o momento em que o treinamento foi dado como concluído. Neste ponto, a rede neural adota os parâmetros que resultam em menores erros para o conjunto de validação sem que ela fique viciada no conjunto de treinamento. O eixo das abscissas indica a quantidade de épocas transcorridas na evolução da rede; o eixo das ordenadas, o erro médio quadrático sobre o conjunto de dados.

7.2.4 Quarto cenário: banco de imagens completo

Por fim, o quarto e último cenário utiliza 160 imagens da classe "não dedo", 20 imagens da classe "dedo ocluso" e 100 imagens da classe "dedo real", sendo que esta representa uma combinação dos cenários anteriores. É importante ressaltar que, aqui, o FNF classifica internamente os dados entre as 3 classes ("não dedo", "dedo ocluso" e "dedo real"), porém a sua tomada de decisão é binária. Caso o dado seja classificado como "dedo ocluso", o comportamento do sistema será o mesmo do caso em que a classificação fosse "não dedo". Dessa forma, este cenário se aproxima ao uso real do sistema.

Foi obtida uma taxa de acerto de 93,71% e uma rede neural com 12 neurônios na camada de entrada. Nesta situação, a taxa de acerto considera 3 categorias. Assim, mesmo que a tomada de decisão seja a mesma para "não dedo" e "dedo ocluso", a classificação errônea em uma dessas duas classes também implica em queda na taxa de acerto.

Além disso, foram obtidos os valores de FAR = 0,70% e FRR = 1,75%. Por serem métricas calculáveis com apenas duas categorias, seus cálculos consideram "não dedo" e "dedo ocluso" como sendo a mesma classe. Em outras palavras, elas indicam as taxas de tomadas de decisão errôneas. 0,70% das decisões resultou na chamada do sistema biométrico acoplado quando não deveria. 1,75% encerraram a execução sem nem chamar o sistema acoplado, sendo que deveriam ter chamado.

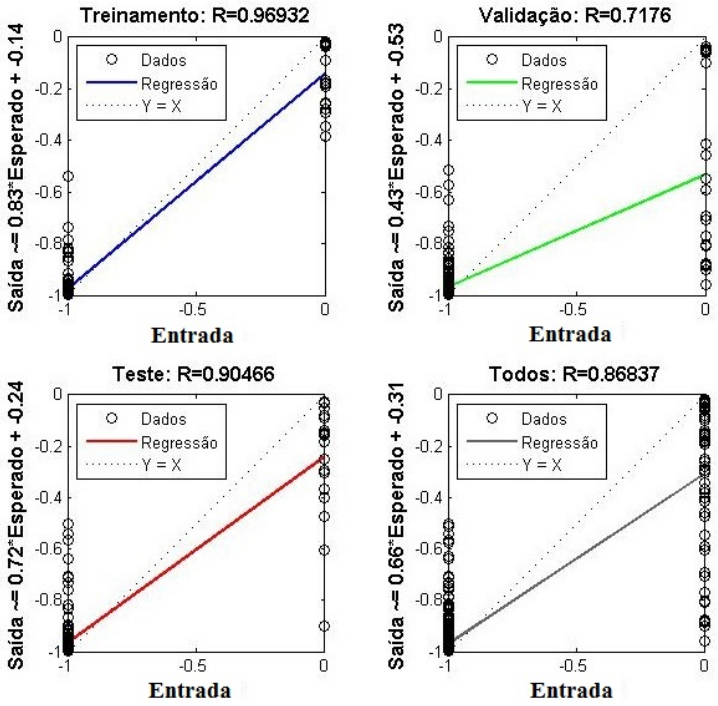

Figura 7.7: Gráficos resultantes do terceiro cenário. O eixo das abscissas indica o valor da entrada real; as ordenadas, o valor no qual a rede neural classificou a entrada. Espera-se que a rede neural classifique todos os dados como sendo do mesmo valor da entrada, ou seja, $Y = X$. Neste caso, os valores de saída entre -1 e -0,5 são classificados como "não dedo". Valores entre -0,5 e 0, como "dedo ocluso".

A classificação é feita da seguinte maneira: os dados de "não dedo" se localizam no intervalo $[-1; -0,5]$; os de "dedo ocluso", em $(-0,5; 0,5]$ e os de "dedos reais", em $(0,5; 1]$.

Analisando os gráficos na Figura 7.7, o funcionamento ideal do sistema ocorre quando a linha contínua se sobrepõe à linha pontilhada e: os dados de "não dedo" se localizam em (-1; -1); os de "dedo ocluso", em (0; 0) e os de "dedos reais", (1; 1).

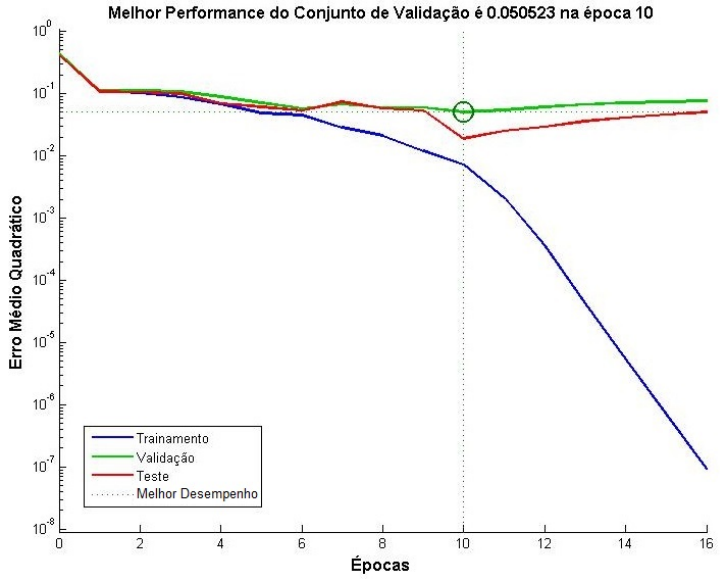

Figura 7.8: Gráfico de evolução da rede neural do terceiro cenário. O ponto sobre a curva de erro do conjunto de validação indica o momento em que o treinamento foi dado como concluído. Neste ponto, a rede neural adota os parâmetros que resultam em menores erros para o conjunto de validação sem que ela fique viciada no conjunto de treinamento. O eixo das abscissas indica a quantidade de épocas transcorridas na evolução da rede; o eixo das ordenadas, o erro médio quadrático sobre o conjunto de dados.

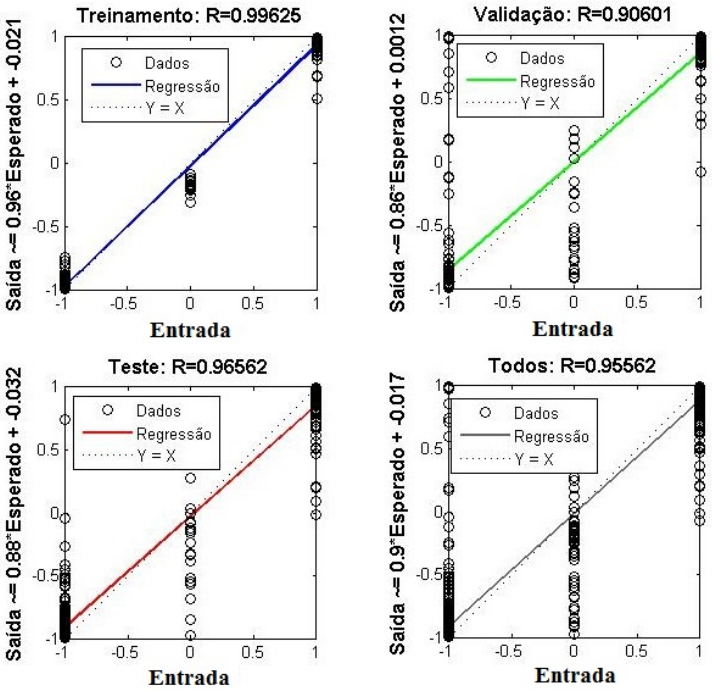

Figura 7.9: Gráficos resultantes do quarto cenário. O eixo das abscissas indica o valor da entrada real; as ordenadas, o valor no qual a rede neural classificou a entrada. Espera-se que a rede neural classifique todos os dados como sendo do mesmo valor da entrada, ou seja, $Y = X$. Neste caso, os valores de saída entre -1 e -0,5 são classificados como "não dedo". Valores entre -0,5 e 0,5, como "dedo ocluso". Valores entre 0,5 e 1, como "dedo real".

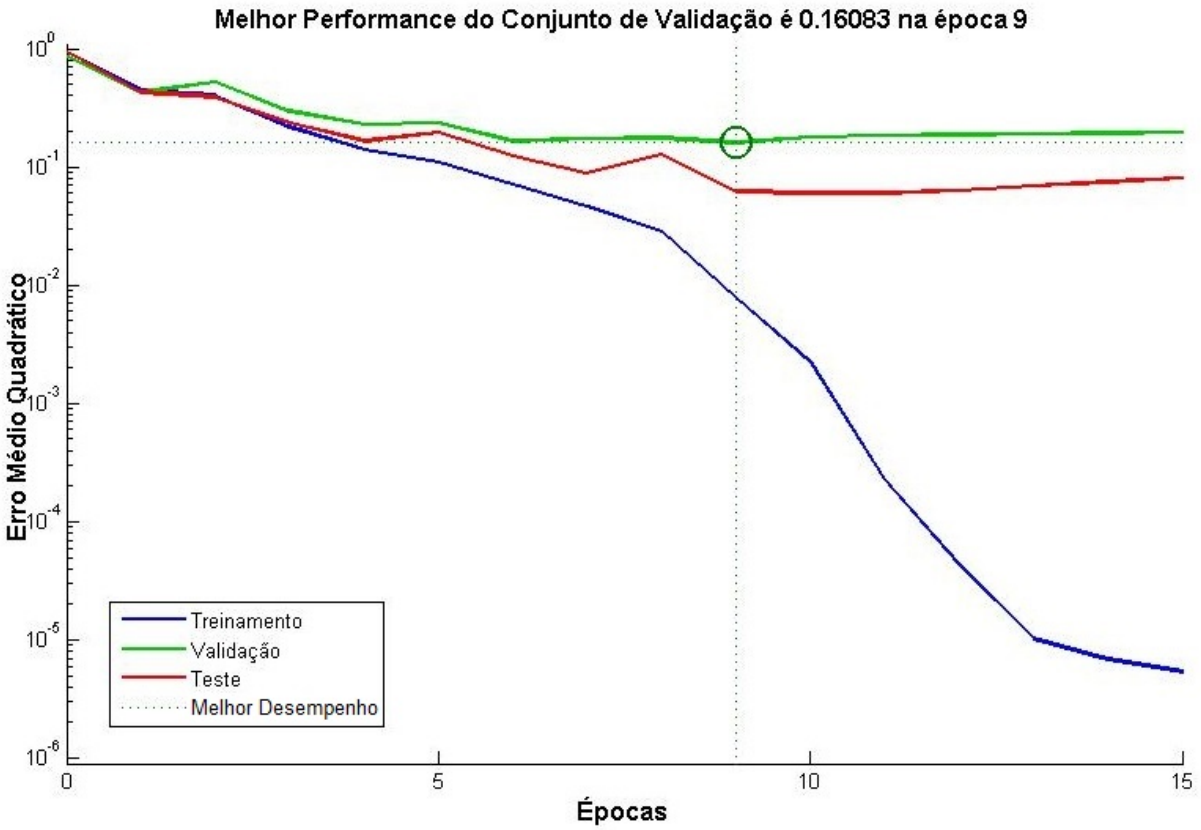

Figura 7.10: Gráfico de evolução da rede neural do quarto cenário. O ponto sobre a curva de erro do conjunto de validação indica o momento em que o treinamento foi dado como concluído. Neste ponto, a rede neural adota os parâmetros que resultam em menores erros para o conjunto de validação sem que ela fique viciada no conjunto de treinamento. O eixo das abscissas indica a quantidade de épocas transcorridas na evolução da rede; o eixo das ordenadas, o erro médio quadrático sobre o conjunto de dados.

Capítulo 8

Conclusão

Com o objetivo de fornecer mais uma alternativa de controle de acesso, surgiram os sistemas biométricos, que utilizam traços intrínsecos à cada indivíduo para indentificá-lo. Dentre os traços biométricos mais bem aceitos, estão as impressões digitais, foco deste trabalho. Elas são de fácil aquisição, únicas, universais e permanentes.

Como é de se esperar, até mesmo sistemas que tem por objetivo prover proteção são passíveis de ataques. Um desses tipos de ataques é chamado de *fraude*. Este tipo de ataque consiste na tentativa, por parte de um invasor, de se passar por um usuário válido. Tal tipo de ataque consiste em explorar erros no sistema que permitem que informações falsas sejam interpretadas como verdadeiras.

Diversas técnicas foram e continuam sendo desenvolvidas, como em [28, 20, 10], para combater tentativas de fraude. Essas técnicas recebem o nome de anti-fraude. O foco, neste contexto, está no desenvolvimento de sistemas em software, comumente através do processamento de imagens.

Por fim, este trabalho propôs um método capaz de detectar fraudes no contexto de leitores biométricos multivista de impressões digitais sem contato que recebe o nome de *Finger or Not Finger (FNF)* e que se utiliza de duas técnicas de descritores de textura: Padrão Binário Local Aperfeiçoado ILBP e Matriz de Co-ocorrência de Níveis de Cinza GLCM.

O FNF encontra maior aplicabilidade em ambientes não supervisionados e seu maior enfoque é a distinção entre a apresentação de credenciais que não são dedos, que são dedos oclusos e dedos reais.

Ao analisar o cenário mais próximo ao seu uso real, o FNF somente realizou tomadas de decisões erradas em 2,45% das avaliações (FAR = 0,70% e FRR = 1,75%). Além disso, obteve resultados ainda melhores em sub-cenários, apresentando sempre taxas de acerto acima de 98%.

Ainda é necessária, entretanto, a realização de experimentos com bases de imagens maiores e mais abrangentes, além de estudos voltados ao seu comportamento contra ataques de apresentação. Contudo, o FNF se apresenta como uma solução promissória aos algoritmos anti-fraude.

Referências

[1] Mark Hudson Beale, Martin T. Hagan, e Howard B. Demuth. *Neural Network Toolbox: User's Guide*. MathWorks, 2015. 41, 43, 48

[2] George D. C. Cavalcanti, Luis Filipe A. Pereira, Hector N. B. Pinheiro, Jose Ivson S. Silva, Anderson G. Silva, Thais M. L. Pina, Daniel B. O. Carvalho, e Tsang Ing Ren. A modular architecture based on image quality for fingerprint spoof detection. *IEEE International Conference on Systems, Man, and Cybernetics*, Outubro 2012. 6

[3] Roger Clarke. Human identification in information systems: Management challenges and public policy issues. *Information Technology and People*, 1994. 3

[4] Luciano R. Costa, Rafael R. Obelheiro, e Joni S. Fraga. Introdução à biometria. In *Livro-texto dos Minicursos, VI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pages 103–151. SBSeg, 2006. 3, 7

[5] Rafael C. Gonzalez e Richard E. Woods. *Digital Image Processing*. Prentice Hall, second edition, 2002. 24, 25, 26, 28

[6] Martin T. Hagan e Mohammad B. Menhaj. Training feedforward networks with the marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6), Novembro 1994. 43

[7] Simon O. Haykin. *Neural Networks and Learning Machines*. Prentice Hall, 2008. 38

[8] Di Huang, Caifeng Shan, Mohsen Ardebilian, Yunhong Wang, e Liming Chen. Local binary patterns and its application to facial image analysis: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 2011. 32

[9] Anil Jain, Brendan Klare, e Arun Ross. Guidelines for best practices in biometrics research. *8th IAPR International Conference on Biometrics*, 2015. 3, 6, 20

[10] Anil Jain e Sharath Pankanti. Fingerprint classification and matching. In *HANDBOOK FOR IMAGE AND VIDEO PROCESSING*. Academic Press, 2000. 62

[11] Anil K. Jain, Arun Ross, e Salil Prabhakar. An introduction to biometric recognition. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, 14(1), Janeiro 2004. x, xiv, 4, 5, 6

[12] Hongliang Jin, Qingshan Liu, Hanqing Lu, e Xiaofeng Tong. Face detection using improved lbp under bayesian framework. *Multi-Agent Security and Survivability, 2004 IEEE First Symposium on*, 2004. 32

[13] Peter Johnson, Richard Lazarick, Emanuela Marasco, Elaine Newton, Arun Ross, e Stephanie Schuckers. Biometric liveness detection: Framework and metrics. *International Biometric Performance Conference (IBPC)*, 2012. 17, 18

[14] Ruggero Donida Labati, Angelo Genovese, Vincenzo Piuri, e Fabio Scotti. Touchless fingerprint biometrics: a survey on 2d and 3d technologies. *Journal of Internet Technology*, 15(3):325–332, Maio 2014. 1607-9264. 7, 11

[15] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *The Quarterly of Applied Mathematics*, 1944. 43

[16] K. Madsen, N.B. Nielsen, e O. Tingleff. Methods for nonlinear least squares problems. Technical report, Technical University of Denmark, 2004. 43

[17] Sébastien Marcel, Mark S. Nixon, e Stan Z. Li. *Handbook of Biometric Anti-Spoofing: Trusted Biometrics under Spoofing Attacks*. Springer-Verlag London, 2014. x, 11, 17

[18] Sébastien Marcel, Yann Rodriguez, e Guillaume Heusch. On the recent use of local binary patterns for face authentication. *INTERNATIONAL JOURNAL OF IMAGE AND VIDEO PROCESSING, SPECIAL ISSUE ON FACIAL IMAGE PROCESSING*, 2007. 32

[19] D.W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 1963. 43

[20] Shahzad Ahmed Memon. *Novel Active Sweat Pores Based Liveness Detection Techniques for Fingerprint Biometrics*. PhD thesis, Brunel University, Abril 2012. 62

[21] Benjamin Miller. Vital signs of identity. *IEEE Spectrum*, 1994. 3

[22] Michael A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, Agosto 2015. xi, 38, 42

[23] Timo Ojala, Matti Pietikäinen, e Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2002. 30

[24] Nalini K. Ratha, Jonathan H. Connell, e Ruud M. Bolle. *Audio- and Video-Based Biometric Person Authentication*. Springer Berlin Heidelberg, 2001. 16

[25] Nalini K. Ratha e Venu Govindaraju. *Advances in Biometrics: Sensors, Algorithms and Systems*. Springer-Verlag London Limited, 2008. 13, 14

[26] Ctirad Sousedik e Christoph Busch. Presentation attack detection methods for fingerprint recognition systems: a survey. *IET Biometrics*, 3, Dezembro 2014. 10, 19

[27] Vishal S.Thakare, Nitin N. Patil, e Jayshri S. Sonawane. Survey on image texture classification techniques. *International Journal of Advancements in Technology*, 2013. 32

[28] Elham Tabassi, Charles L. Wilson, e Craig I. Watson. *Fingerprint Image Quality*. National Institute of Standards and Technology, 2004. 62

[29] James Wayman, Anil Jain, Davide Maltoni, e Dario Maio. *Biometric Systems: Technology, Design and Performance Evaluation*. Springer-Verlag London Limited, 2005. 3

[30] Hao Yu e Bogdan M. Wilamowski. Levenberg–marquardt training. In *Intelligent Systems*. Fevereiro 2011. 43

[31] Caue Zaghetto, Alexandre Zaghetto, Flávio de B. Vidal, e Luiz H. M. Aguiar. Touchless multiview fingerprint quality assessment: Rotational bad-positioning detection using artificial neural networks. *Biometrics (ICB), 2015 International Conference on*, pages 394 – 399, Maio 2015. x, 12

[32] Guoqiang Peter Zhang. Neural networks for classification: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 30:451 – 462, 2000. 36