

**UNIVERSIDADE FEDERAL DE UBERLÂNDIA**  
**FACULDADE DE ENGENHARIA ELÉTRICA**  
**PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**



**UMA CONTRIBUIÇÃO PARA UM SISTEMA COMPUTACIONAL DE  
LEITURA AUTOMÁTICA DE VALORES EM CÉDULAS  
MONETÁRIAS**

**Camilo de Lelis Tosta Paula**

Orientando

**Keiji Yamanaka Dr.**

Orientador

**Uberlândia MG, 2013**

**Camilo de Lelis Tosta Paula**

**UMA CONTRIBUIÇÃO PARA UM SISTEMA COMPUTACIONAL DE  
LEITURA AUTOMÁTICA DE VALORES EM CÉDULAS  
MONETÁRIAS**

Dissertação apresentada por Camilo de Lelis Tosta  
Paula à Universidade Federal de Uberlândia para  
obtenção do título de Mestre em Ciências, avaliada  
em 12 de abril de 2013 pela banca examinadora

*Área de concentração:*

*Processamento da Informação, Inteligência  
Artificial.*

Banca Examinadora

*Prof. Dr. Keiji Yamanaka(Orientador)*

*Prof. Dr. Alexandre Cardoso – FEEL/UFU*

*Prof. Dr. Hipólito Barbosa Machado Filho – IFG*

**Uberlândia MG, 2013**

- P324c Paula, Camilo de Lelis Tosta, 1968-  
2013 Uma contribuição para um sistema computacional de leitura automática de valores em cédulas monetárias / Camilo de Lelis Tosta Paula. - 2013.  
67 p. : il.
- Orientador: Keiji Yamanaka.  
Dissertação (mestrado) – Universidade Federal de Uberlândia, Programa de Pós-Graduação em Engenharia Elétrica.  
Inclui bibliografia.
1. Engenharia elétrica - Teses. 2. Processamento de imagens - Técnicas digitais - Teses. 3. Redes neurais (Computação) - Teses. 4. Cédulas - Teses. I. Yamanaka, Keiji. II. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

# **UMA CONTRIBUIÇÃO PARA UM SISTEMA COMPUTACIONAL DE LEITURA AUTOMÁTICA DE VALORES EM CÉDULAS MONETÁRIAS**

**Camilo de Lelis Tosta Paula**

Texto da dissertação apresentada à Universidade Federal de Uberlândia perante a banca de examinadores abaixo, como parte dos requisitos necessários à obtenção do título de Mestre em Ciências.

---

**Prof. Dr. Keiji Yamanaka – UFU**  
**Orientador**

---

**Prof. Dr. Alexandre Cardoso – UFU**

---

**Prof. Dr. Hipólito Barbosa Machado Filho – IFG**

**Uberlândia**

**2013**

## **Dedicatória**

*Dedico este trabalho a minha mãe, meus filhos  
e minha esposa Patrícia Borges Hanaoka, que  
nos momentos mais difíceis, sempre estiveram  
ao meu lado.*

*Meu muito obrigado!*

## **AGRADECIMENTOS**

Ao Professor Keiji Yamanaka, pela orientação e pelo constante estímulo transmitido durante todo o trabalho.

Aos amigos, colegas de minter e colegas do IFTM – Campus Uberaba, que me auxiliaram nas horas de dificuldade, e a todos que colaboraram direta ou indiretamente na execução deste trabalho.

*Não importa aonde você parou...  
Em que momento da vida você cansou...  
O que importa é que sempre é possível e necessário "Recomeçar".  
Recomeçar é dar uma chance a si mesmo...  
É renovar as esperanças na vida e o mais importante...  
Acreditar em você de novo.*

*(Carlos Drummond de Andrade)*

## **RESUMO**

As técnicas de processamento digital de imagens, reconhecimento de padrões e redes neurais artificiais não são utilizadas frequentemente no mercado tecnológico, pois são trabalhosas e acabam se tornando caras para quem irá utilizar. Quando pesquisamos no mercado ferramentas tecnológicas utilizando estes recursos, encontramos de maneira fragmentada ou apenas para resolver um ou outro problema, encontramos também em softwares que são proprietários. Assim, propõe-se uma **CONTRIBUIÇÃO A UM SISTEMA COMPUTACIONAL DE LEITURA AUTOMÁTICA DE VALORES EM CÉDULAS MONETÁRIAS**, que irá juntar as técnicas de programação difundidas no meio acadêmico para desenvolver um software livre, disponível a ser utilizado no mercado e também para estudos acadêmicos. A finalidade é a construção de um software com aplicações em captura de imagens, processamento digital de imagens, e reconhecimento de padrões utilizando técnicas de RNA – **REDES NEURAIS ARTIFICIAIS**, acoplado a um sintetizador de voz, que irá atender às pessoas com necessidades visuais e até mesmo pessoas com dificuldades em atingir o foco necessário em seus trabalhos. Assim, podendo ser executado em qualquer tipo de PDV (Ponto de venda), ou mesmo em locais onde se realiza a contagem de valores altos, trazendo a confiabilidade necessária para se fechar o dia trabalhado sem falhas que podem ocasionar perdas para a empresa e mesmo para clientes.



## ABSTRACT

*Image Digital Processing techniques, standards recognizing, and neural artificial nets are not often used in the technology market, because they are hard to work with and expensive for who uses them. When we make research on technology tools we find them fragmented as they solve few problems and come in software with no license for copywriting. Thus, we propose a Computer System for Automatic Reading of Money Banknotes, who will join programming techniques widespread in academia to develop free software that can be used both in the market and for studies in academia, a software with applications on image capture, digital image processing, pattern recognition that uses artificial neural network techniques, coupled to a voice synthesizer, which will assist people with visual needs and even people with difficulties in achieving the necessary focus such as supermarkets. The system can be adapted to any sales place, or even where people work counting high money values. It brings the reliability needed without errors that can cause losses to the company and people as well.*

## SUMÁRIO

1. Introdução.....	16
1.1 Objetivo .....	17
1.1.1 Objetivo geral .....	17
1.1.2 Objetivos específicos .....	17
1.2 Justificativa .....	18
1.3 Organização da Dissertação .....	18
2. Fundamentação Teórica .....	20
2.1 Reconhecimento de padrões .....	20
2.2 JAI (Java Advanced Imaging). .....	20
2.3 Pré-Processamento da imagem .....	22
2.4 Redes Neurais Artificiais (RNA) – Reconhecimento das cédulas.....	22
2.5 Arquitetura de uma RNA .....	24
2.6 Redes MLP ( <i>Perceptron</i> Multi Camadas) .....	25
2.7 Algoritmo <i>Backpropagation</i> .....	26
2.8 Usando a API Java <i>Speech</i> e o sintetizador FREETTS .....	26
3. Análise de Correlatos .....	28
3.1 Introdução .....	28
3.2 Slep – Scanner Leitor Portatil.....	28
3.3 Auire Prisma .....	29
3.4 Look Tell Money Reader for iPhone.....	30
3.5 Sistema de Identificação de Cédulas Monetárias Para Indivíduos Portadores de Deficiência Visual .....	31
3.6 Comparação entre as características dos softwares. ....	32
4. Arquitetura do Sistema.....	33
4.1 Introdução: .....	33
4.2 Especificações do Sistema .....	33
4.3 Diagrama de Caso de Uso.....	36
4.4 Treinamento da rede neural artificial .....	38
5. Detalhes da Implementação .....	41
5.1 Introdução. ....	41
5.1 Aquisição de Imagens .....	41
5.2 Binarização da Imagem .....	44

5.3	Isolando o valor da cédula. ....	47
5.4	Centralizando a Imagem .....	49
5.5	Diminuir Imagem.....	50
5.6	Executando a RNA .....	51
5.7	Comparando os índices. ....	51
5.8	Dificuldades encontradas .....	53
5.8.1	Rotacionar a imagem .....	53
5.8.2	Reconhecimento de cores .....	57
6.	Resultados obtidos.....	60
7.	Conclusões e trabalhos futuros.....	61
7.1	Contribuições .....	63
7.2	Trabalhos Futuros .....	64
8.	Referências .....	65

## LISTA DE ILUSTRAÇÕES

Figura 2.1 - Processamento da imagem.....	22
Figura 2.2 - RNA, modelo Matemático de um neurônio artificial (Moreto & Rolim, 2010)...	23
Figura 2.3 – Feedforward de camada simples .....	24
Figura 2.4 – Feedforward de camadas duplas .....	25
Figura 2.5 - Efeito do algoritmo Backpropagion (Ramos, 2003).....	26
Figura 2.6 - Sintetizador de voz .....	27
Figura 4.1 - Nível 1 de uso do sistema .....	34
Figura 4.2 – Etapa 2 de uso do Sistema – OCR .....	35
Figura 4.3 – Etapa 2 de uso do Sistema – Pré-Processamento .....	35
Figura 4.4 - Diagrama de caso de uso .....	37
Figura 4.5 - Treinamento da Rede Neural Artificial .....	39
Figura 5.1 - módulos do projeto .....	41
Figura 5.2 - Modulo de captura .....	42
Figura 5.3 – Interface do Sistema de captura .....	42
Figura 5.4 - Botão vinculado a classe Deletar .....	43
Figura 5.5 – Imagem das cédulas monetárias.....	43
Figura 5.6.1 – Em (A) o resultado da aplicação do método de Otsu. Em (B) o resultado da aplicação do método Niblack. (RAMALLHO, 2005).....	45
Figura 5.7 - Imagem Original .....	46
Figura 5.8 - O resultado da aplicação do método de Otsu. ....	46
Figura 5.9 - O resultado da aplicação do método Niblack .....	46
Figura 5.10 - Diferença de iluminação nas notas (RAMALLHO, 2005), p. 28.....	47
Figura 5.11 - Imagem com o limite do corte .....	47
Figura 5.12 - Imagem resultado do corte.....	47
Figura 5.13 - área definida para filtrar.....	48
Figura 5.14 – área filtrada.....	48
Figura 5.15 – Nota após a passagem do filtro .....	48
Figura 5.16 – Histograma Vertical .....	49
Figura 5.17 – a imagem Centralizada.....	50
Figura 5.18 – sem a mudança de tom .....	50
Figura 5.19 - Condição de comparação para retornar o índice.....	52
Figura 5.20 - imagem para rotacionar A.....	53

Figura 5.21 - Rotação ao contrário .....	54
Figura 5.22 - imagem recebida .....	54
Figura 5.23 – Imagem com maior rotação.....	55
Figura 5.24 - figura rotacionada .....	56
Figura 5.25 - Imagem manchada. ....	56

## Lista de Tabelas

Tabela 3.1 - Comparação dos softwares .....	32
Tabela 4.1 - Documentação do caso de uso.....	38
Tabela 5.1 - Tabela referência de valores .....	51
Tabela 5.2 - Tabela de cores para a nota 2 reais.....	57
Tabela 5.3 - Tabela de cores para a nota de 5 reais .....	57
Tabela 5.4 - Tabela de cores para a nota 10 reais.....	58
Tabela 5.5 - Tabela de cores para a nota 20 reais.....	58
Tabela 5.6 - Tabela de cores para a nota 50 reais.....	58
Tabela 5.7 - Tabela de cores para a nota 100 reais.....	59
Tabela 5.8 - tabela de resultados para comparação das cores .....	59
Tabela 6.1 – A tabela de resultados dos testes com amostra de 628 imagens capturadas pelo sistema, as duas primeiras colunas mostram o valor da cédula monetária e o número de imagens capturadas por cada valor. As colunas 3 a 9 apresentam a matriz confusão para os testes. A coluna 10 apresenta a quantidade de respostas certas, a colunas 11 mostra o percentual de erros e a coluna 12 o percentual de acertos.....	60
Tabela 6.2 - Tabela de classificação .....	60
Tabela 6.3 - Tabela de resultados .....	61

## Lista de Siglas

R.N.A – Redes Neurais Artificiais.

P.D.V. – Ponto de Venda.

O.M.S. – Organização Mundial de Saúde.

J.A.I. – Java Advanced Imaging.

A.P.I. – Application Programming Interface.

J.T.A.P.I. – Java Telefonia.

M.L.P. – Multi Layer Perceptron

ICMC – Instituto de Ciências Matemáticas e de Computação.

USP – Universidade de São Paulo

T.A. – Tecnologia Assistiva.

C.E.D.I. – Centro Especializado em Desenvolvimento Infantil.

PUC – Pontifícia Universidade Católica.

UML - *Unified Modeling Language*.

HD – Hard Disk.

RN – Rede Neural.

JSP – Java Server Pages.

## 1. Introdução

O desenvolvimento de softwares quando se trata de auxiliar o ser humano nas tarefas do dia a dia é de uma abrangência enorme. Para realizar uma simulação de um evento real são necessárias várias técnicas de programação para definição de padrões a fim de atingir o objetivo principal.

O reconhecimento destes padrões são habilidades extremamente desenvolvidas por seres humanos que durante suas ações diárias executam atividades de forma automática, tudo se deve a informação passada ano após ano, pessoa para pessoa, ensinamentos distribuídos para o maior número de indivíduos.

Fazer com que um equipamento realize essa ação sem a intervenção do homem com a certeza que não haverá um segundo de desatenção para cometer um erro, passou a ser um desafio de pesquisadores. As ações que o ser humano realiza, são na verdade um conjunto de padrões aprendidos durante vários anos de existência e com a capacidade de aperfeiçoamento, as técnicas foram se tornando mais fáceis e menos complexas.

A distribuição da informação como elemento fundamental da sociedade pode ser realizada através da cultura e educação, o que as torna um veículos de disseminação dos conhecimentos mais eficazes para uma gradual formação de uma sociedade cidadã. **(TORRES & MAZZONI, 2004)** cita que “*As possibilidades de utilização dos recursos multimídia trazem grandes contribuições para a produção de conteúdos didáticos digitais, permitindo o respeito ao estilo de aprendizagem dos usuários e às suas preferências quanto à forma de recepção da informação*”, seguindo esse conceito podemos citar softwares para leitura de tela do computador, cadeiras de roda eletrônicas para locomoção rápida, softwares e impressoras para impressão em Braille, softwares de captura de imagem via celular para tradução de Libras, etc.

Diante disso, houve a necessidade do desenvolvimento de um sistema computacional que aborda várias técnicas de programação, podendo ser utilizado no mercado financeiro como também para atender à tecnologia assistiva.



## **1.1 Objetivo**

## **1.2 Objetivo geral**

Realizar o reconhecimento de cédulas monetárias nacionais a partir de imagens capturadas via webcam

## **1.3 Objetivos específicos**

1. Encontrar na linguagem Java as melhores técnicas para a realização da captura de imagem a partir de uma webcam, realizada em ambientes com variação de iluminação;
2. Buscar os métodos de processamento digital de imagens que realizem o tratamento da imagem capturada através de técnicas de limiarização, retirada de ruídos e movimentação de pixels;
3. Pesquisar as redes neurais artificiais, avaliando os métodos existentes que possam realizar o reconhecimento do valor encontrado com maior eficácia e rapidez no treinamento;
4. Analisar os sintetizadores de voz existentes que melhor se adaptam à linguagem Java.

## 1.4 Justificativa

A necessidade de um reconhecedor de números em cédulas monetárias é ampla e pode ser utilizado para várias situações, como por exemplo, inseri-lo em um sistema *mobile* e assim favorecer o deficiente visual no reconhecimento dos possíveis valores monetários. De acordo com o projeto “Pequenos Olhares” (abcegos, 2012), do Conselho Brasileiro de Oftalmologia apontou números semelhantes aos da Organização Mundial de Saúde (OMS), “em 2004 havia 4.000.000 de brasileiros com intensidade visual no melhor olho entre 20/60 e 20/400, o que de acordo com o site da Associação Baiana de Cegos varia entre visão próxima do normal e baixa visão severa”.

Adaptando o sistema no aparelho celular possibilita ao usuário realizar pagamentos e recebimentos com muito mais facilidade, estando segura ao receber um troco um troco ou efetuar pagamentos desnecessários.

Outra forma de uso desse sistema seria a implantação em um equipamento de contagem de cédulas, o que era apenas para um tipo de cédula (R\$2,00, R\$5,00, R\$10,00, R\$20,00, R\$50,00, R\$100,00), passa a receber notas variadas, o equipamento por si reconhece e apresenta o somatório final.

## 1.5 Organização da Dissertação

A dissertação está dividida nos seguintes capítulos:

No capítulo 2 são apresentados os fundamentos teóricos nos quais o software será desenvolvido, características que envolvem redes neurais, seu treinamento e aprendizado.

No capítulo 3 é realizado a análise de correlatos, os trabalhos relacionados ao projeto aqui proposto, as principais semelhanças e diferenças.

No capítulo 4 são apresentados os módulos de execução, métodos, classes e bibliotecas utilizadas para trabalhar com redes neurais e sintetizadores de voz.

No capítulo 5 apresenta a arquitetura do sistema, como foi realizada a captura da imagem, o pré-processamento da cédula e o pós-processamento que é o reconhecimento e por fim o acoplamento do sintetizador de voz para realizar a leitura da cédula.

No capítulo 6 é realizada a análise dos resultados dos testes no sistema, quantas vezes o sistema confundiu as notas e não trouxe o resultado correto e quantas vezes o sistema reconheceu corretamente.

No capítulo 7 apresenta as considerações e sugestões de novas aplicações a partir do que foi iniciado neste trabalho.

## 2. Fundamentação Teórica

### 2.1 Reconhecimento de padrões

São dois conceitos sobre reconhecimento de padrões que podem ser citados, o primeiro mais genérico, de acordo com (CASTRO & PRADO, 2002), “*padrões são os meios pelos quais o mundo é interpretado e, a partir dessa interpretação, elaboram-se atitudes e decisões*”. E o segundo mais detalhado que é de acordo com (ARAÚJO, 2010) apud Duda 2000, “... *um sistema de reconhecimento de padrões típico pode ser dividido em cinco subsistemas básicos: sensoriamento, segmentação, extração de características, classificação e pós-processamento*”. Ainda segundo (ARAÚJO, 2010):

“Na etapa de sensoriamento, um sensor ou transdutor converte um fenômeno físico em um conjunto de dados, que pode ser uma imagem ou um sinal de fala, por exemplo. Estes dados são compostos de objetos e plano de fundo, geralmente possuindo algum tipo de ruído. Nas etapas seguintes é necessário que os objetos estejam devidamente separados. Os objetos são, portanto, segmentados. Em seguida são extraídas as características dos objetos que são úteis na classificação. O classificador utiliza estas características para determinar a qual classe o objeto em questão pertence. No pós-processamento alguma técnica pode ponderar as saídas de diferentes classificadores e tomar a decisão final.”

### 2.2 JAI (Java Advanced Imaging).

Uma alternativa interessante é a criação de rotinas de processamento de imagens utilizando a linguagem Java em conjunto com a API (*Application Programming Interface*) JAI. De acordo com (BERNI, 2008) apud JAI *Guide* 1999, “o JAI permite virtualmente que qualquer algoritmo de processamento de imagens possa ser adicionado a sua API e ser usado como se fosse uma parte nativa da sua biblioteca”.

Outra vantagem do Java é a sua extensibilidade. Desde a versão original do Java, diversas extensões foram adicionadas ao código do núcleo, proporcionando maior flexibilidade e poder às aplicações. Estas extensões adicionam objetos e classes que melhoram a capacidade do programador Java para utilizar recursos como: Java Swing - um componente definido para criar interfaces gráficas multi-plataforma para o usuário ver e sentir; Java Sound - para alta qualidade de renderização com 32 canais de áudio e MIDI com síntese de som controlado; Java 3D - para a geometria avançada e som espacial 3D; Java Media Framework - para componentes de jogar e controlar o tempo baseados em mídias como áudio e vídeo; Java Telefonia (JTAPI) - para aplicações de telefonia computadorizada; Java Speech - para a inclusão de tecnologia de fala em applets e aplicações Java.(JAI *Guide*, 1999)

A API do Java chamada de JAI nos proporciona ser multi-plataforma, Orientada a Objetos, ser flexível e extensível, independência do dispositivo (pode-se configurar um gráfico de operações rederizáveis sem preocupar com a resolução da imagem ou tamanho da fonte), poderoso (suporta imagens com formatos complexos) alto desempenho e interoperável.

Utilizar a API JAI para criar uma imagem é mais difícil, mas permite grande flexibilidade nos formatos e estruturas internas. (SANTOS, 2004) sugere a seguinte sequencia de passos para criação de uma aplicação JAI:

1. “Criar os dados da imagem em um array unidimensional (o mapeamento das coordenadas da imagem deve ser feito pelo programador).
2. Com este array, criar uma instância de classe que herda de *DataBuffer* do tipo adequado.
3. Criar uma instância de *SampleModel* do tipo e dimensões adequadas. A classe *RasterFactory* contém métodos-fábrica que facilitam este passo.
4. Com a instância de *SampleModel* criar um *ColorModel* compatível.
5. Usar as instâncias de *DataBuffer* e *SampleModel* para criar uma instância de *WritableRaster*. Esta instância, depois de criada, pode ser também usada para manipular os pixels desta imagem.
6. Usar as instâncias de *SampleModel* e *ColorModel* para criar uma instância de *TiledImage*.
7. Associar o *Raster* à instância de *TiledImage*.

Ao fim da realização da sequencia a instancia pode ser manipulada, visualizada ou armazenada.”

Para a implementação de alguns algoritmos específicos de processamento de imagens, é necessário obter os valores dos pixels em determinadas posições ou regiões. A forma mais simples de fazer isto é usando uma imagem já na memória (instância de *BufferedImage* ou *PlanarImage*, por exemplo), obtendo o *Raster* associado a ela e usando um dos métodos *getSample* ou *getPixel*. Estes métodos recuperam um valor de um pixel ou todos os valores associados a um pixel, respectivamente.

E finalmente para exibirmos a imagem em um monitor que é uma função indispensável de um sistema de processamento digital de imagens usa-se classes da API Swing para exibir instâncias de *BufferedImages* através da criação de um objeto da classe *JLabel* que deve receber no seu construtor como argumento uma instância de *ImageIcon* que por sua vez recebe como argumento um objeto *BufferedImage*.

A instância de *JLabel* pode ser adicionada à interface gráfica de uma aplicação envolvida em um objeto de *JScrollPane* para conter barras de rolagem que são necessárias para imagens com áreas maiores do que a disponível na interface gráfica.

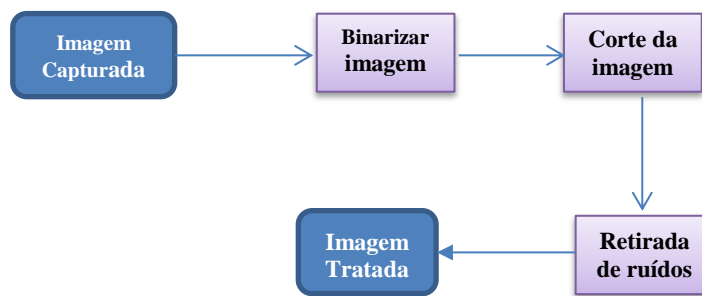
## 2.3 Pré-Processamento da imagem

“Processar uma imagem consiste em transformá-la sucessivamente com o objetivo de extrair mais facilmente a INFORMAÇÃO nela presente”. (ALBUQUERQUE & ALBUQUERQUE, 2000)

As necessidades de processamento de imagens são de acordo com Lindley (1993 apud (COSTA, 2001)):

Aplicadas quando se verificam:

- Alguns aspectos da imagem precisam ser melhorados devido a presença de ruídos tornando possível o reconhecimento.
- Elementos da imagem precisam ser caracterizados, classificados, comparados ou medidos, como por exemplo, a extração de atributos em imagens de impressões digitais que possibilitam realizar a classificação e verificação.
- Faz-se necessário combinar ou reorganizar determinadas regiões das imagens



**Figura 2.1 - Processamento da imagem**

Sendo assim, percebe-se que é necessário o uso de filtros para realçar certa imagem, para que a resultante seja mais adequada que a original em uma aplicação específica. O uso de filtros não é somente para a questão de realce ou retirada de ruídos, mas pode ser utilizado para demarcação de que área da imagem será utilizada para realizar a leitura no espaço definido. O fato de que nem todas as notas estarão em perfeito estado de conservação ou a luminosidade nem sempre será a mesma ao realizar a captura, em qualquer uma das situações anteriores irá gerar assim muitos ruídos.

## 2.4 Redes Neurais Artificiais (RNA) – Reconhecimento das cédulas.

De acordo com (SILVA, et al., 2010), “... são modelos computacionais inspirados no sistema nervoso de seres vivos”, que é baseado no neurônio do ser humano, recebe entradas através de conexões com outros neurônios por meio de impulsos elétricos, nas

RNAs os mesmos são substituídos por sinais representados por  $x_i$ , baseado em (NOVAIS, et al., 2010) “... são combinados aos pesos sinápticos associados a cada um, representados por  $w$ , e produz uma saída que pode ser passada para outros neurônios da rede”, como mostra a figura 1 abaixo:

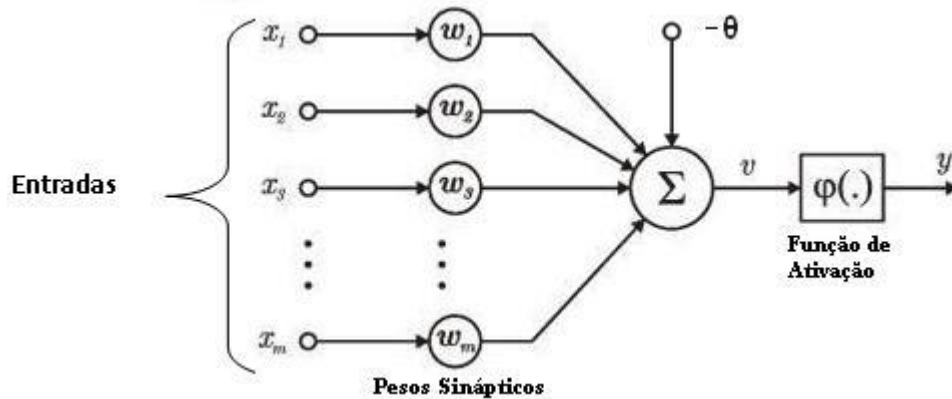


Figura 2.2 - RNA, modelo Matemático de um neurônio artificial (MORETO & ROLIM, 2010)

Percebe-se que o neurônio artificial é constituído de sete elementos básicos:

- ✓ sinais de entrada( $x_1, x_2, x_3, \dots, x_m$ ),
- ✓ pesos sinápticos( $w_1, w_2, w_3, \dots, w_m$ ),
- ✓ combinador linear( $\Sigma$ ),
- ✓ limiar de ativação( $\theta$ ),
- ✓ potencia de ativação( $v$ ),
- ✓ função de ativação( $\phi$ ) e o
- ✓ sinal de saída( $y$ ).

(SILVA, et al., 2010) resume que:

“... o funcionamento do neurônio por meio dos seguintes passos:

- Apresentação de um conjunto de valores que representam as variáveis de entrada do neurônio;
- Multiplicação de cada entrada do neurônio pelo seu respectivo peso sináptico;
- Obtenção do potencial de ativação produzido pela soma ponderada dos sinais de entrada, subtraindo-se o limiar de ativação;
- Aplicação de uma função de ativação apropriada, tendo-se como objetivo limitar a saída do neurônio;
- Compilação da saída a partir da aplicação da função de ativação neural em relação ao seu potencial de ativação.”

## 2.5 Arquitetura de uma RNA

Uma rede neural pode ser dividida em 3 camadas:

Camada de entrada → É responsável pela entrada de informações que podem ser sinais, características ou medições vindas do meio externo.

Camadas escondidas, ocultas ou invisíveis → São aquelas responsáveis por extrair as características associadas ao processo ou sistema inferido.

Camada de saída → É responsável pela produção e apresentação dos resultados obtidos.

De acordo com (NOVAIS, et al., 2010) no artigo sobre Redes Neurais Artificiais afirma que “... da mesma forma que o cérebro, as RNA são capazes de aprender com exemplos e utilizar o conhecimento adquirido para melhorar seu desempenho na realização de tarefas”, o processo de aprendizagem é essencial para a RNA realizar as tarefas propostas. Um exemplo disso seria inserir uma pessoa em um trabalho sem ocorrer um treinamento, curso ou procedimentos que deva seguir ao executar o trabalho.

O treinamento da rede é um processo de ajuste dos pesos sinápticos, para que possam determinar a resposta correta, assim que outros elementos forem lançados para a rede. Entre os principais tipos de redes existem arquiteturas *feedforward* de camada simples e de camadas duplas. Como mostra as figuras 2.3 e 2.4:

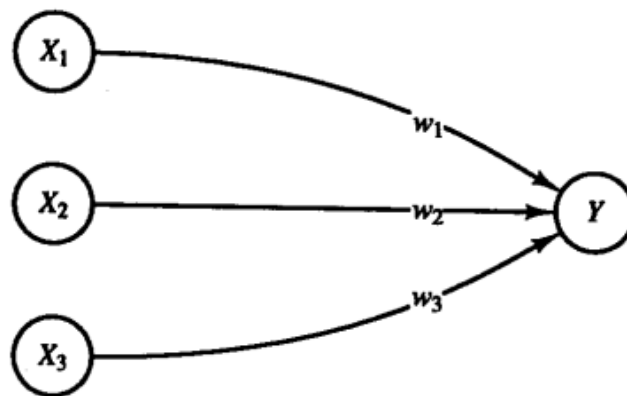


Figura 2.3 – Feedforward de camada simples (FAUSET, 1994)



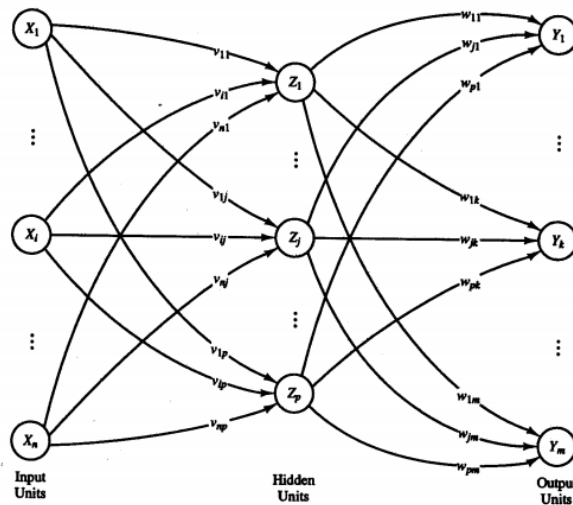


Figura 2.4 – Feedforward de camadas duplas (FAUSET, 1994)

Entre os principais tipos de redes com arquitetura *feedforward* de camadas múltiplas se encontra o *Perceptron* multicamadas (*Multlayer Perceptron* – MLP), cujo algoritmo de aprendizado utilizado em seus processos de treinamento são baseados na regra generalizada.

## 2.6 Redes MLP (*Perceptron* Multi Camadas)

Redes MLP consistem em apresentar camadas intermediárias denominadas camadas ocultas, onde cada uma tem uma função definida. A camada de saída recebe estímulos das camadas intermediárias e organiza qual será a finalidade da ação. Estas camadas funcionam como extratoras de características, seus pesos são uma codificação de características apresentadas nos padrões de entrada e permitem que a rede crie sua própria representação do problema.

Para treinar uma rede MLP, é necessário a utilização de um algoritmo que permita estabelecer o conjunto de pesos ótimos. O mesmo é utilizado para o treinamento deste tipo de rede, que consiste numa generalização da regra Delta (técnica do gradiente descendente), conhecido com algoritmo *BackPropagation*. As redes MLP são também conhecidas como treinamento supervisionado, inclui-se pelo menos uma camada intermediária para realizar o reajuste dos pesos, logo resolve o problema por executar uma propagação recursiva dos erros.

## 2.7 Algoritmo *Backpropagation*

Este algoritmo de treinamento contém duas fases:

*Forward* → Fase correspondente à propagação progressiva do sinal de entrada para a camada de saída e com o cálculo do erro os pesos são ajustados através da regra de ajuste de pesos (regra delta generalizada).

*Backward* → Fase correspondente ao reajuste dos pesos, para que seja efetuado ocorre a retropropagação do erro. Todo o processo onde ocorre a assimilação é supervisionado, e o objetivo do algoritmo é de ajustar os pesos para que o erro entre a saída desejada e a saída calculada seja minimizado.

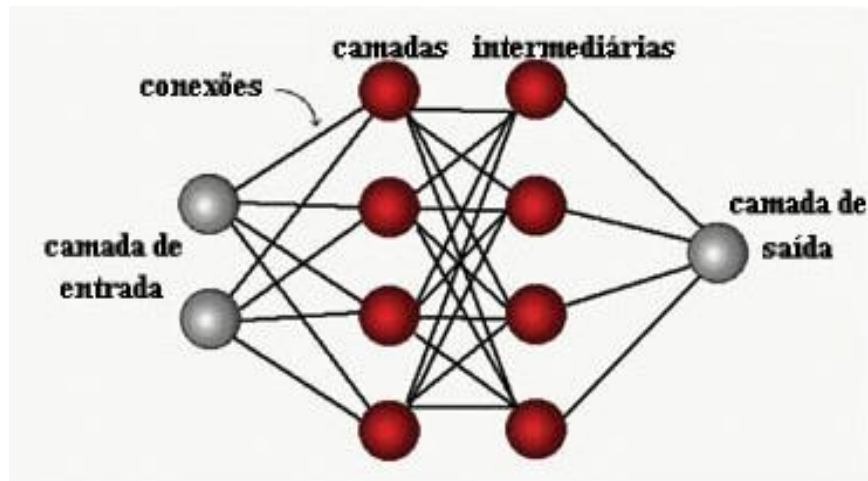


Figura 2.5 - Efeito do algoritmo *Backpropagation* (RAMOS, 2003)

De acordo com (FAUSET, 1994), “uma rede de múltiplas camadas é uma rede com uma ou mais camadas de nodos (as assim chamadas unidades escondidas) entre as unidades de entrada e das unidades de saídas”, como mostra (RAMOS, 2003) na figura 2.5 “A rede apresentada, possui todas as conexões, o que significa que um neurônio em qualquer camada da rede está conectado a todas as outras unidades (neurônios) na camada anterior. O fluxo de sinais através da rede é feito positivamente, da esquerda para a direita, camada a camada.”

## 2.8 Usando a API Java *Speech* e o sintetizador FREETS

(RODRIGUES JUNIOR & MOREIRA, 2000), cita que “Os sintetizadores, como o próprio nome diz, são as entidades responsáveis pela síntese de voz. Eles são criados

através da classe *Central* que implementa a interface *Engine* (herdada pelos sintetizadores) responsável por acessar a aplicação *Engine*.” As formas de instanciar um objeto capaz de sintetizar voz, ocorrem através do método estático *createSynthesizer*, classe principal do programa

De acordo com (MARANGONI & PRECIPITO, 2006):

“Java Speech API suporta dois tipos básicos de gramática: gramáticas das regras gramaticais e do ditado. Estes tipos da gramática diferem na maneira em que as aplicações ajustam as gramáticas, nos tipos de sentenças permitem, na maneira em que os resultados são fornecidos, na quantidade de recursos computacionais requeridos, e na maneira em que são usados eficazmente no projeto da aplicação.”

Esta API é a mais indicada para trabalhar com Java e assim realizar a conexão com o sintetizador Texto-Voz.



Figura 2.6 - Sintetizador de voz (MIRANDA, 2009)

O FreeTTS é um sintetizador de voz feito completamente em Java, sua licença permite o seu uso e sua distribuição livre de qualquer encargo ou restrição. O maior problema detectado foi a privação de suporte para língua portuguesa, então será instalada a biblioteca em português do sintetizador de voz conhecido como M-Brola. Apresenta um grande número de idiomas e não tem problemas com licença, porque é permitido ser usado livre de encargos para fins não comerciais e militares.

### 3. Análise de Trabalhos Correlatos

#### 3.1 Introdução

Nesse capítulo serão apresentados alguns softwares voltados para Tecnologia Assistiva (TA), com embasamento para a criação de um software melhor e com mais recursos para atenção ao deficiente.

Será aplicada a TA, que de acordo com **(BERSCH, 2008)** CEDI (Centro Especializado em Desenvolvimento Infantil em Porto Alegre, RS), “... a TA deve ser então entendida como um auxílio que promoverá a ampliação de uma habilidade funcional deficitária ou possibilitará a realização da função desejada e que se encontra impedida por circunstância de deficiência ou pelo envelhecimento”. **(BERSCH, 2008)** (RADABAUGH apud BERSCH, 1993) cita que “Para as pessoas sem deficiência, a tecnologia torna as coisas mais fáceis. Para as pessoas com deficiência, a tecnologia torna as coisas possíveis”.

Existem vários aplicativos com a proposta *Text-To-Speech*, são focados em Tecnologia Assistiva e foram testados alguns mais conhecidos no mercado, são eles:

#### 3.2 Slep – Scanner Leitor Portatil.

Realização da leitura de textos a partir da captura da imagem pela câmera de um celular, desenvolvido a partir do Sistema Operacional *Symbian* da empresa Nokia **(NNSolutions, 2010)**. A empresa NNSolutions, localizada em Salvador na Bahia, domina os direitos autorais do software e realiza a venda do produto. A empresa cita em seu site **(NNSolutions, 2010)** que o software traz como vantagem:

- ✓ Integração social;
- ✓ Autonomia;
- ✓ Portabilidade e
- ✓ Respeito ao meio ambiente.

A empresa não oferece uma versão para testes, e sendo assim fica atrelada a apenas um sistema operacional então é considerada como ponto negativo, se tornando inviável financeiramente, pois além de comprar o sistema, ainda tem o investimento no aparelho compatível.

### 3.3 Auire Prisma

Através da leitura, reconhece as cédulas monetárias e descobre as cores emitidas pelo objeto em contato com o aparelho. Consiste de um sensor de contato, que mostra o resultado desejado a partir de botões que definem o que será reconhecido. Desenvolvido pela Auire Tecnologias Acessíveis, (Gil, 2010) não publicou o endereço da empresa e também não oferece uma versão para testes.

Por ser uma empresa brasileira tem suporte para língua portuguesa e tem como principais pontos positivos:

- ✓ Reconhece cédulas de real;
- ✓ Não necessita de conexão com a internet;
- ✓ Realiza reconhecimento instantâneo.

São pontos negativos:

- ✓ Não acumula os valores identificados;
- ✓ É instalado em equipamento próprio (sem portabilidade).

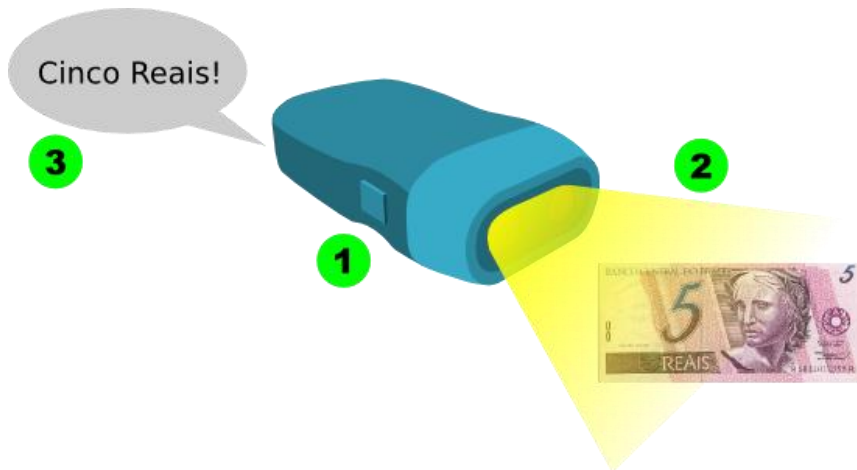


Figura 3.1 - Auire Prima (Gil, 2010)

### 3.4 *Look Tell Money Reader for iPhone.*

De acordo com a empresa Looktel (**LookTel, 2012**), o software “reconhece instantaneamente a moeda e fala sua denominação, permitindo às pessoas com deficiência visual total ou parcial, identificar facilmente o valor da moeda capturada.”. O software suporta moedas de várias nacionalidades, além do dólar dos EUA, euro, libra esterlina, dólar canadense e dólar australiano. Somente disponível para computadores da empresa Apple, originalmente criado para iPhones. A empresa disponibiliza um versão para testes (Beta) somente para iPhones, o que impossibilita a realização de testes para verificação dos pontos positivos e negativos.



Figura 3.2 - Iphone com o software da LookTel instalado - (**LookTel, 2012**)

Na pesquisa realizada foi apresentado como pontos positivos:

- ✓ O software acumula os valores reconhecidos;
- ✓ Realiza leitura de textos em inglês;
- ✓ Adiciona voz em rótulos;
- ✓ Não necessita de conexão com a internet;
- ✓ O reconhecimento é instantâneo;
- ✓ Não é somente para deficientes visuais, o que o deixa mais versátil.

Foi constatado que os pontos negativos são:

- ✓ Não reconhece cédulas brasileiras;
- ✓ Não tem portabilidade, executa apenas em Iphones;

### **3.5 Sistema de Identificação de Cédulas Monetárias Para Indivíduos Portadores de Deficiência Visual**

Desenvolvido pela PUC-MG de Belo Horizonte, como cita (CAMPOS & MACHADO, 2009), o método desenvolve uma rede neural de retropropagação que é treinada para reconhecer a cédula com base nos canais H e S do histograma de cores da imagem obtida por um aparelho celular.

A implementação foi executada em um aparelho da empresa Sony Ericson modelo K550i, o *firmware* apresentado é o *Firmware Flashing* de propriedade da própria Sony Ericson.

De acordo com (CAMPOS & MACHADO, 2009), “A vantagem dessa aplicação é o fato de ela funcionar em aparelhos celulares sem o uso de nenhum tipo de conexão com internet ou ligação telefônica, tornando possível que ela seja utilizada em qualquer ambiente”.

Os responsáveis pelo produto não ofereceu uma versão para testes e verificação de pontos positivos e negativos e as informações a respeito do software foram encontradas através de pesquisa textual.

As desvantagens encontradas foram:

- ✓ Não tem portabilidade;
- ✓ Não reconhece com exatidão as cédulas monetárias;
- ✓ Não acumula os valores reconhecidos;

### 3.6 Comparação entre as características dos softwares.

A tabela 1 apresenta o resumo das características dos softwares correlatos apresentadas anteriormente:

**Tabela 3.1 - Comparação dos softwares**

Software/Características	Suporte para língua Portuguesa?	Tipo de moeda (Real)?	Acumula valores?	Necessita conexão com a internet?	Reconhecimento instantâneo?	Somente para deficientes visuais?	Utiliza RNA	Sist.Operacional totalmente livre?	Equipamento embarcado em que equipamento?
Sistema computacional de reconhecimento de números de Cédulas monetárias	Sim	Real	Sim	Não	Sim	Não	Sim	Sim	Desktop/note book/net book
Look Tell Money Reader for iPhone.	Não	dólar EUA, euro, libra esterlina, dólar canadense dólar australiano	Sim	Não	Sim	Não	Não informado	Não	Computadores Apple
Auri Prisma	Sim	Real	Não	Não	Não	Sim	Não informado	Não	Aparelho com sensor de contato para reconhecimento.
Sistema de Identificação de Cédulas Monetárias Para Indivíduos Portadores de Deficiência Visual	Sim	Real	Não	Não	Sim	Sim	Sim	Não	Aparelho Sony Ericson Mod. K550i
Slep - Scanner Leitor Portátil.	Sim	Real	Não	Não	Sim	Sim	Não informado	Não	Aparelhos smartphones da Nokia.



## 4. Arquitetura do Sistema

### 4.1 Introdução:

Nesse capítulo é apresentado os requisitos e a arquitetura do Sistema Computacional de Leitura Automática de Cédulas Monetárias, desenvolvidos utilizando Java como linguagem padrão e técnicas de Processamento digital de imagem, redes neurais artificiais e conexão com sintetizador de voz.

“Tomando como exemplo a construção de um edifício, percebemos que ao projetar-se uma construção, esta não possui apenas uma planta, mas sim diversas, enfocando o projeto de construção do prédio sob diversas formas, algumas referentes ao layout dos andares, outras apresentando a planta hidráulica e outras ainda abordando a planta elétrica, por exemplo. Isso torna o projeto do edifício completo, abrangendo todas as características da construção. Da mesma maneira os diversos diagramas fornecidos pela UML, permitem analisar o sistema em diferentes níveis, podendo focar a organização estrutural do sistema, o comportamento de um processo específico, a definição de um determinado algoritmo ou até mesmo as necessidades físicas do sistema para eu este funcione adequadamente (GUEDES, 2004)”.

Serão apresentados os diagramas para demonstrar o sistema, estruturados em alguns níveis apresentados pela *Unified Modeling Language*. (UML). A documentação é necessária para um melhor entendimento do sistema, deixando na parte conceitual o que foi realizado na prática.

### 4.2 Especificações do Sistema

O Sistema é desenvolvido a partir da aquisição da imagem digital de uma cédula monetária nacional (Real), a imagem é capturada por uma webcam conectada a um computador.

No Capítulo 5.2; foi criada uma classe de aquisição de imagem (captura.java) que realiza a conexão da câmera com o computador e gravação da imagem no HD (Hard Disk);

No capítulo 5.3; realiza a binarização da imagem, deixando-a em apenas dois tons de cinza;

No capítulo 5.4; é isolado o valor encontrado na cédula;

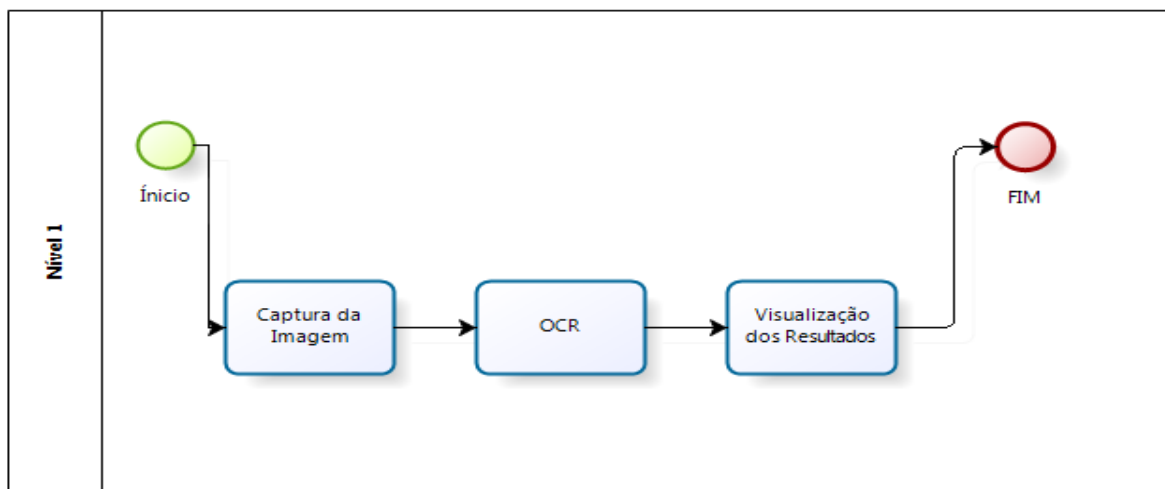
No capítulo 5.5; a imagem é centralizada;

No capítulo 5.6; a imagem é reduzida, são apresentados os métodos para essa redução;

No capítulo 5.7; foi montado um módulo de redes neurais que realiza o treinamento da rede para reconhecimento do valor da cédula.

Para facilitar o entendimento, o projeto foi dividido em três níveis, captura, pré-processamento (OCR) e a visualização dos resultados:

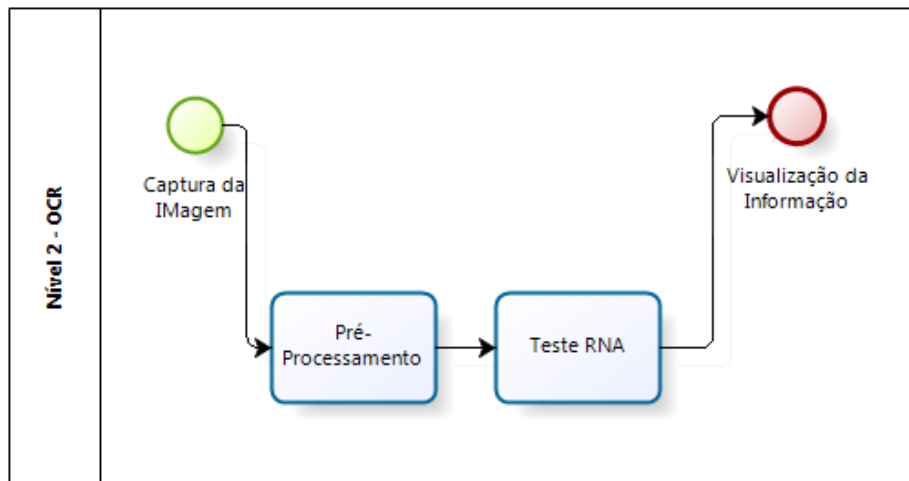
Nível 1 – apresenta os módulos superiores do sistema.



**Figura 4.1 - Nível 1 de uso do sistema**

O início se dá com a captura da imagem que logo é guardada em um diretório no disco rígido do equipamento em que o software foi instalado. Logo a imagem é encontrada automaticamente pelo módulo OCR, que irá realizar o pré-processamento da imagem através das classes que estão vinculadas. Passando o resultado final, que seria o índice retornado da rede que realiza o reconhecimento padrão e apresentando os resultados. O módulo OCR é onde todo o processamento da imagem acontece, como demonstra a figura 9.

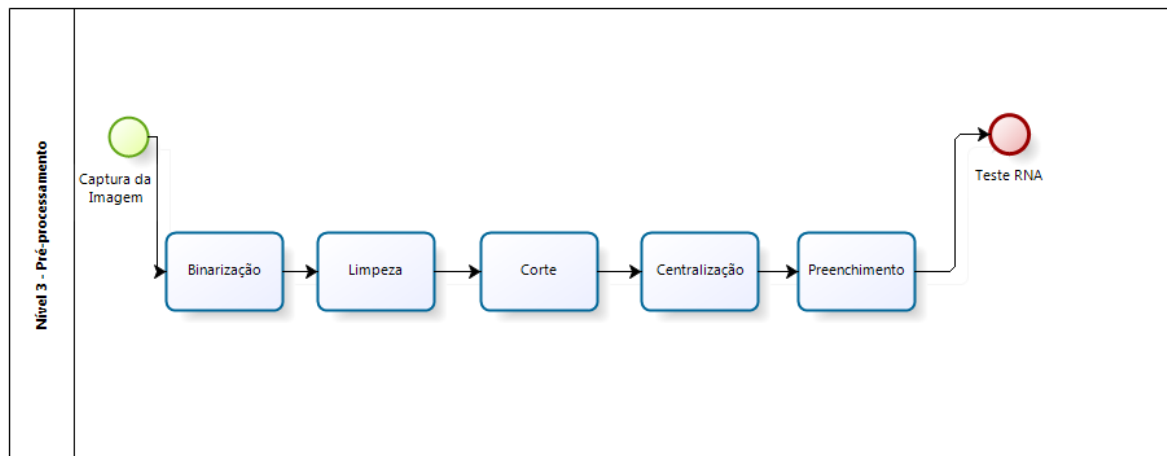
Nível 2 – apresenta os módulos de nível intermediário do sistema



**Figura 4.2 – Etapa 2 de uso do Sistema – OCR**

O nível intermediário foi criado para realização dos testes da rede neural, vindo da etapa pré-processamento onde o vetor que caracteriza a imagem é passado para a classe que realiza as comparações fornecidas pela rede. Assim retorna ao índice final, apresentando qual a cédula que configura aquele vetor.

Nível 3 – apresenta os módulos inferiores do sistema.



**Figura 4.3 – Etapa 2 de uso do Sistema – Pré-Processamento**

O nível 3 são as classes que estão vinculadas a classe OCR, onde são realizadas:

- ✓ binarização da cédula;
- ✓ retirada dos ruídos da imagem;
- ✓ definição dos pontos de corte da imagem;
- ✓ centralização da imagem e o
- ✓ preenchimento de uma imagem nova evitando ruídos causados pela mudança de arquivo.

Foi criado um módulo não vinculado ao sistema que executa as etapas de processamento da imagem e treinamento da rede, como demonstrado na figura 4.4.

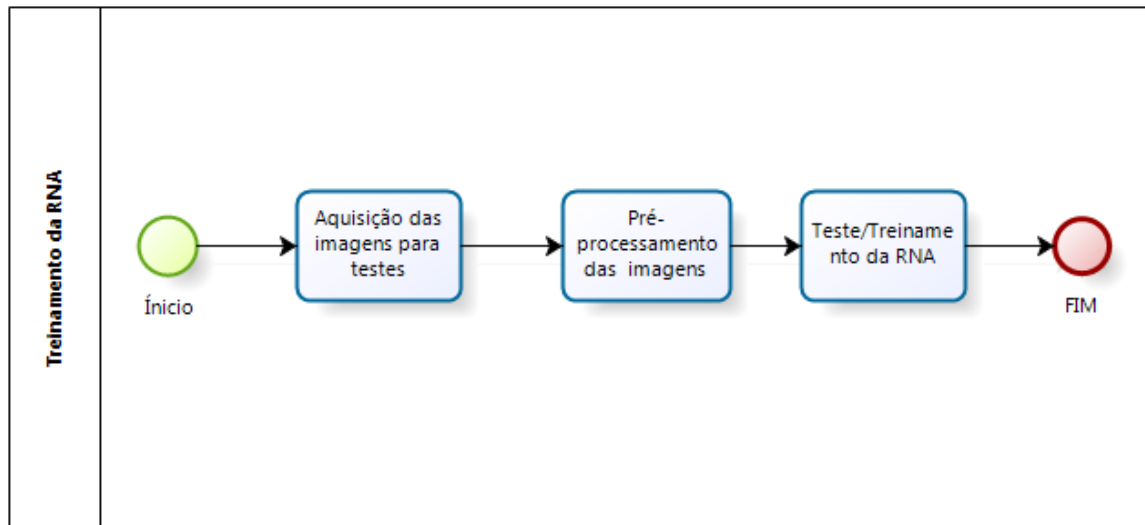
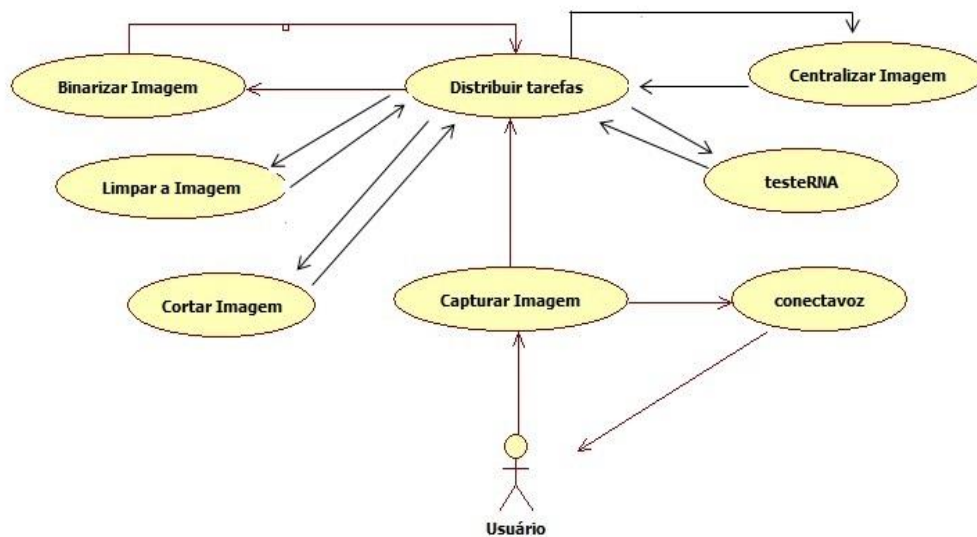


Figura 4.4 - Etapas para treinamento da rede

### 4.3 Diagrama de Caso de Uso.

(GUEDES, 2004) cita que “o diagrama de caso de uso é o diagrama mais geral e informal da UML, sendo utilizado normalmente nas fases de Levantamento e Análise de Requisitos, embora venha a ser consultado por todo o processo de modelagem”. O diagrama de caso de uso tem como finalidade apresentar o sistema de maneira simples com a maior clareza para o usuário.

Do ponto de vista do usuário, o acesso ao sistema é apenas na interface gráfica, onde contém a captura da imagem com alguns botões de funcionalidade e o retorno do resultado obtido. Apesar dessa visão superficial do cliente, foi demonstrado como as classes interagem entre si, sem que o usuário perceba todo o procedimento, como apresentado na figura 4.5.



**Figura 4.5 - Diagrama de caso de uso**

Será explicado o fluxo de dados que acontece no sistema:

A primeira funcionalidade do processo acontece quando o cliente ativa o processo de captura da imagem, por padrão da câmera a mesma tem o tamanho de 640x480 *pixels*. Logo essa figura é passada para o pré-processamento da imagem, onde começa a distribuição de tarefas do sistema, próximo passo é a binarização da imagem, processo definido para que os *pixels* existentes na figura permaneça com valores 0 e 255. Esse procedimento marca as bordas da figura, retorna para a classe distribuidora de tarefas, requisitando o processo Limpa Imagem, Ele transfere os pontos de corte para retirada de ruídos e os valores são encaminhados para o processo cortar Imagem, que seleciona somente o valor da cédula descartando o restante. Retorna para o OCR que solicita o processo de Centralizar Imagem, utilizando antes o processo Preencher para preparar um novo arquivo com fundo branco, para recebe-la na posição correta. Já com a imagem centralizada, volta para o OCR, onde ocorrerá a conversão da imagem em um vetor de 1 e -1 e passado para o processo TesteRNA. O OCR receberá um índice referente ao valor da cédula entrão enviará para o processo capturar imagem, verificando os valores passando por uma variável que acumulará com outros. Então, é exibida a somatório total(acumulador de resultados) das notas que foram lançadas ao sistema, ao mesmo tempo o capturar Imagem para em forma de texto o valor que foi reconhecido para o processo ConectaVoz, onde vai acontecer a conexão a um sintetizador de voz que revelará qual cédula foi reconhecida, retornando para o usuário a soma das entradas que aconteceram no sistema.

Tabela 4.1 - Documentação do caso de uso

Nome do Caso de Uso	Reconhecimento dos números
Caso de Uso Geral	
Ator Principal	Usuário
Atores Secundários	
Resumo	Este caso de uso descreve as etapas percorridas pelo sistema para o reconhecimento do valor apresentado na cédula.
Pré-condições	Iniciar o procedimento clicando com o botão do lado direito no centro da imagem apresentada pelo sistema
Pós-condições	
Ações do Ator	Ações do sistema
1. Apresentar a cédula para o sistema	
	2. Capturar a imagem
	3. Enviar a imagem para o distribuidor de tarefas
	4. Realizar a Binarização da Imagem
	5. Isolar o número encontrado na imagem
	6. Retirar os ruídos existentes na imagem
	7. Centralizar a imagem
	8. Realizar o teste com a RNA
	9. Retornar o índice encontrado
	10. Acumular os valores
	11. Retornar o som do valor encontrado
12. Ouvir o valor.	

#### 4.4 Treinamento da rede neural artificial

A figura 4.6 mostra como é realizado o treinamento da RNA. Através da captura de um conjunto de imagens das cédulas monetárias feita com o auxílio de um módulo a parte para treinar a rede.

Para treinar uma RNA, de acordo com (LIU, et al., 2008)

“... existem vários algoritmos para treinar as redes MLP. Dentre esses, o algoritmo de aprendizado mais conhecido para treinamento destas redes é o da retropropagação do gradiente do erro observado. Este é um algoritmo supervisionado, que usa a saída desejada para cada entrada fornecida para ajustar os parâmetros, denominados pesos da rede de acordo com a regra delta”.

Como cita (ARAÚJO, 2012),

O treinamento e funcionamento de uma RN compreendem as seguintes etapas:

- (1) Apresentação das entradas;
- (2) Propagação dos estímulos ao longo da rede;
- (3) Atualização das saídas da rede;
- (4) Adaptação dos valores das conexões entre unidades de processamento.

Foi definida a quantidade de 25 neurônios para a camada escondida através de tentativa e erro, e quando executados os testes, foi percebido que o uso de muitos neurônios na camada escondida durante o treinamento da rede pode ocorrer um sobreajuste (*overfitting*) e consequentemente acontecerá uma redução na taxa de acerto.

De acordo com (SANTOS, et al., 2005),

“Não existe um critério geral que permita definir o número de neurônios na camada escondida. Em geral, redes neurais com poucos neurônios escondidos são preferidas, visto que elas tendem a possuir um melhor poder de generalização, reduzindo o problema de sobreajuste (*overfitting*). Entretanto, redes com poucos neurônios escondidos podem não possuir a habilidade suficiente para modelar e aprender os dados em problemas complexos, podendo ocorrer *underfitting*, ou seja, a rede não converge durante o treinamento.”

A figura 4.6 exemplifica o processo realizado pelo sistema:

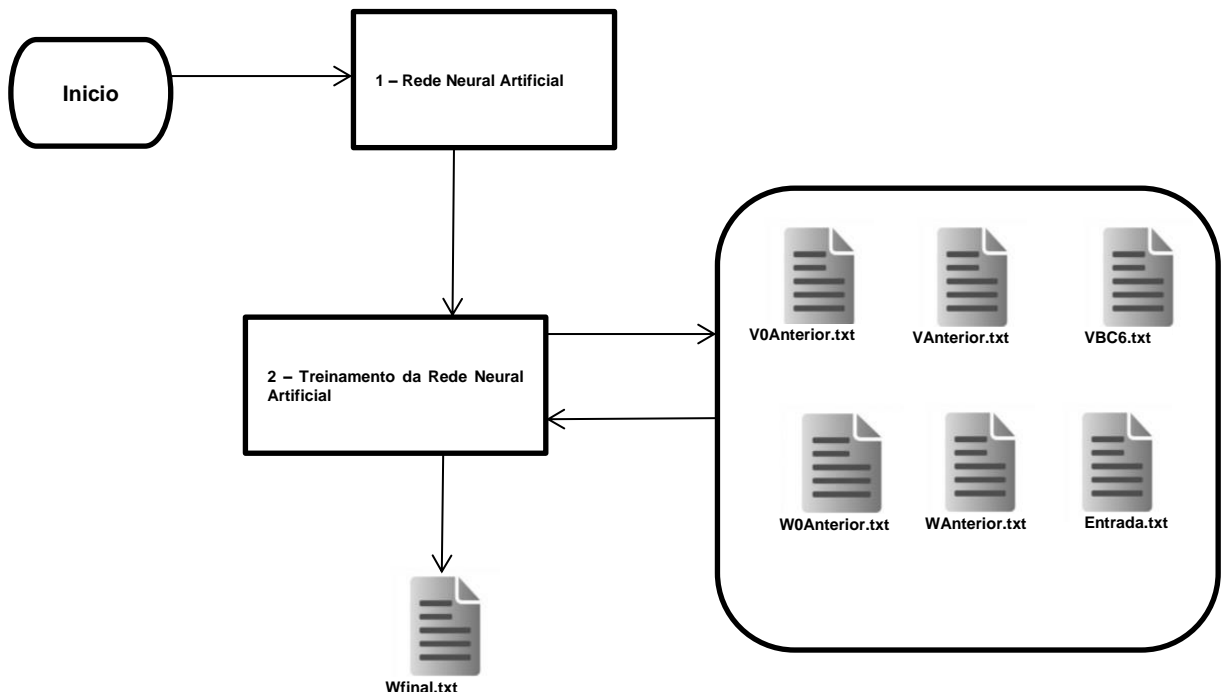


Figura 4.6 - Treinamento da Rede Neural Artificial

Para o treinamento da rede foi aplicada a regra de delta, medindo o curso a distancia entre a resposta atual e a desejada. O próximo passo é diminuir a distância entre os pesos das conexões criando arquivos textos com a dimensão desejada, logo são comparados e os valores ajustados, para então acontecer o reconhecimento da cédula.

(CASTRO & PRADO, 1999-2002) cita que “*Padrões representativos de cada classe estão disponíveis e o sistema é “ensinado” a reconhecer padrões por meio de esquemas de adaptação.*”, como apresentado na figura 4.6, é necessário oferecer parâmetros para realizar o reconhecimento, assim como seres humanos necessitam de conhecimento anterior para confirmar que o objeto avaliado é realmente o que é.

Segundo (CERA, 2005), “faz parte da natureza humana, ao observar um objeto, fazer uma coleta de informações, as quais são comparadas com as propriedades e comportamentos conhecidos e armazenados em sua mente. Através dessa comparação os seres humanos são capazes de reconhecer o alvo de sua observação”.

Sendo assim é necessária a criação de imagens das cédulas monetárias para servir como “alvo de observação”, ou seja, a rede terá como referência os parâmetros de treinamento.



## 5. Detalhes da Implementação.

### 5.1 Introdução.

Embora algoritmos de variados tipos de processamentos já integram softwares e sistemas embarcados, a tecnologia vem sofrendo avanços para a melhoria das técnicas de processamento de imagens.

A implementação foi realizada utilizando a linguagem Java, por ser multiplataforma e uma linguagem de âmbito comercial, que visa uma ampliação no uso do software procurando diversificar seu uso.

- ✓ Na seção 5.2 será apresentada como foi feita a aquisição da imagem;
- ✓ Na seção 5.3 será o pré-processamento da imagem adquirida, mostrando as técnicas de processamento digital de imagens;
- ✓ Na seção 5.4 as técnicas de RNAs, como foi construída e
- ✓ Na seção 5.5 o retorno da informação para o usuário através de um sintetizador de voz, como realça a figura 5.1.

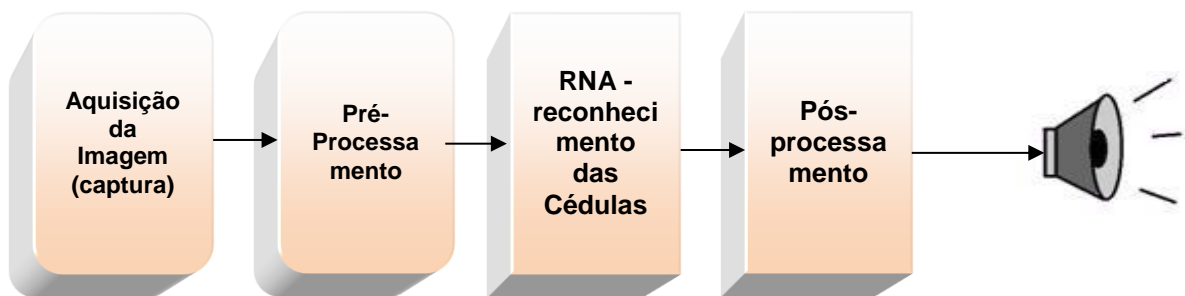


Figura 5.1 - módulos do projeto

### 5.2 Aquisição de Imagens

Realizada através de uma webcam da Microsoft modelo VX2000, apresentando os seguintes recursos:

- ✓ Ajuste automático para luz baixa;
- ✓ Com resolução de 1.3 megapixels.

Instalada em um suporte com hastes, deixando-a acima da cédula, facilitando o manuseio da nota, ou seja, passar as cédulas com facilidade e um computador, como mostra a figura 5.2.



Figura 5.2 - Modulo de captura

O processo foi idealizado a partir da classe `jwebCam`, onde há o método de conexão com a webcam (`MySnapshot`), a imagem é gravada em um diretório no disco rígido, e pode ser trabalhada utilizando as técnicas de processamento digital de imagens.

Segundo (SOUZA & PASSELA, 2011), a “*linguagem Java é a facilidade do uso de classes nativas para a manipulação de imagens, e a existência de várias APIs para a manipulação de imagens, OCR e inteligência artificial*”. O projeto utilizou o Java Media Framework(JMF) para realizar o reconhecimento da câmera, através de um toque com o botão do lado direito do mouse em cima da imagem, onde é realizada a captura e a gravação, como mostra a figura 5.3.

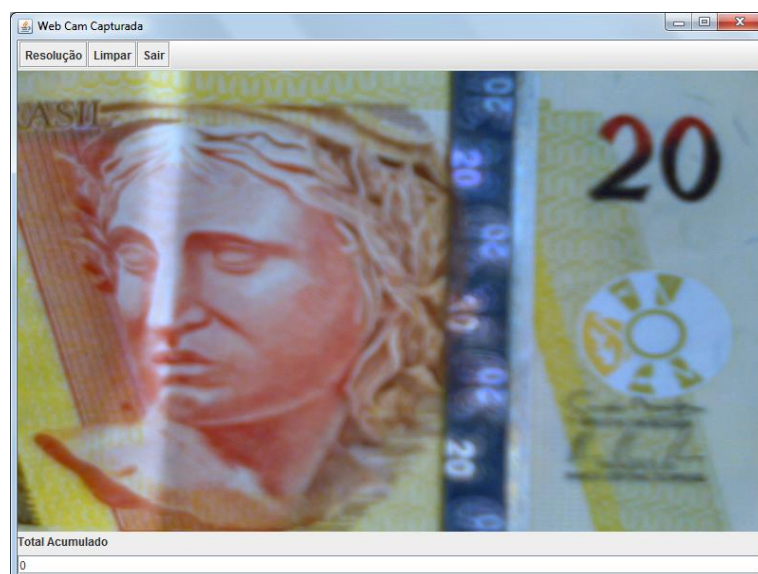


Figura 5.3 – Interface do Sistema de captura

Foram feitas adaptações na classe original para atender as demandas do projeto. Entre elas está o método *mouseClicked*, onde foi inserido um evento para inicializar o processo com um clique no botão do mouse.

A criação do botão “Limpar”, como é apresentado na Figura 5.4, tem a função de apagar todas as imagens gravadas no diretório onde se encontram as imagens capturadas.

O botão “Sair” foi uma adaptação, para evitar forçar a saída do sistema.

O campo “Total Acumulado” recebe o valor da cédula monetária que foi capturada e acumulada com valores anteriores, trazendo assim a contagem total.

Foi inserido também a comparação dos índices, onde é recebido do módulo de teste da RNA e feita a equivalência de valores.

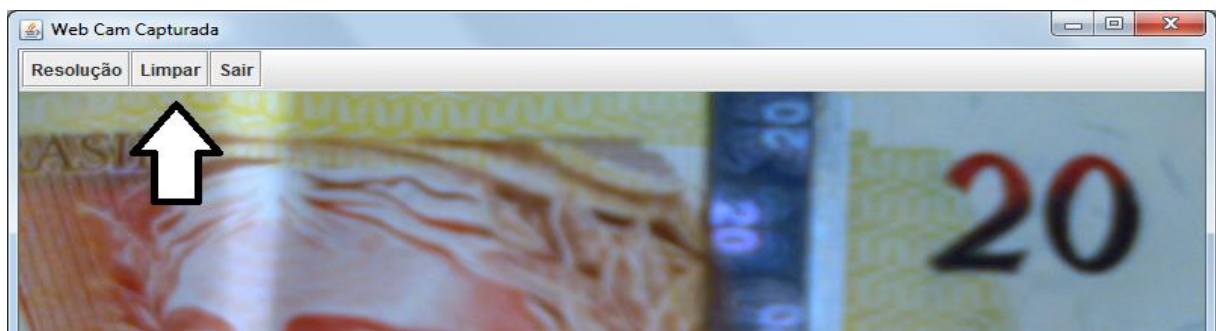


Figura 5.4 - Botão vinculado a classe Deletar

Na figura 5.5, um mosaico das imagens capturadas pelo sistema, em um ambiente com iluminação adequada as imagens tiveram maior nitidez o que trouxe bons resultados durante os testes realizados.



Figura 5.5 – Imagem das cédulas monetárias

No método *MySnapshot* foi instanciada a classe OCR do projeto e uma rotina de validação para reconhecimento do índice retornado pela classe “TesteRNA”. Este índice recebido é que determina qual o valor da cédula.

Depois de realizada a comparação, o valor encontrado na cédula é enviado para uma variável do tipo inteiro, que será somada com outra variável contendo os valores acumulados anteriormente, exibindo os valores acumulados durante o processo de captura.

Foi inserido um botão vinculado à classe Deletar, com a função de apagar todas as imagens existentes no diretório raiz do projeto, uma vez realizado esse processo acontece a inicialização do programa, voltando ao início e zerando os acumuladores.

### 5.3 Binarização da Imagem

A classe denominada de OCR foi criada para centralizar as chamadas de outras classes, conforme representado no diagrama de caso de uso da figura 4.5. É nessa classe que o resultado da imagem final é convertida em uma matriz de 1 e -1 para que seja testada com a rede neural.

Efetivada a binarização da imagem, apresentando apenas tons de cinza, portanto a sub-classe *BufferedImage* da classe *java.awt.image*, pois possui estrutura adequada para trabalhar com imagens em tons e cinza e coloridas. Agrega valores entre 0 e 255, para que haja um destaque maior das bordas, sendo 0 para a cor preta e 255 para a cor branca

(SOUZA & PASSELA, 2011) cita que “a *BufferedImage* possui duas classes internas, uma para armazenagem dos pixels (*Raster*), e outra para interpretar os valores de cada pixel(*ColorModel*)”.



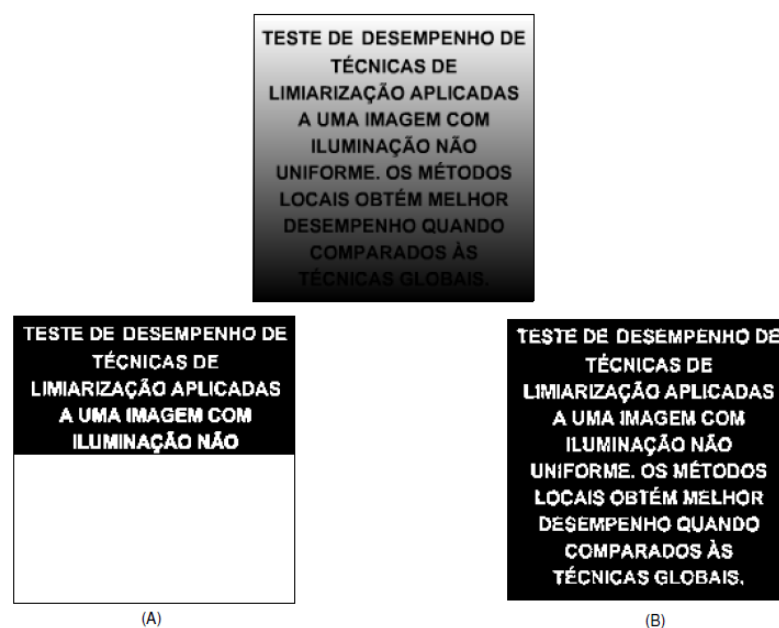
Figura 5.6 - Limiarização da imagem

A limiarização (*thresholding*) é uma das mais importantes abordagens para segmentação de imagens, técnica utilizada para dividir os níveis de cinza em uma ou duas categorias, acima ou abaixo de um determinado patamar (0 ou 255). Este é um dos vários métodos para criar uma máscara binária para a imagem, (Parker 1996 apud **(RODRIGUES & OLIVETE, 2005)**) cita que “*um método eficiente de limiarização deve ser capaz de fornecer automaticamente um valor (limiar) para o qual todos os pontos com valores de intensidade inferior a este devam ser eliminados*”, essa eliminação seria a inserção do valor 0 para compor a imagem. A intenção é usar essa etapa para eliminar pixels que não fazem parte do processo de separação do valor da cédula.

De acordo **(RODRIGUES & OLIVETE, 2005)** “*a técnica de Otsu calcula o valor da limiar obtido globalmente, supondo que os pontos da imagem podem ser classificados em fundo e objeto, o que pode ser eficiente em imagens com histogramas bimodais, nas quais a presença é notória das duas classes acima citadas.*”

A técnica NiBlack tem como característica ser adaptativa, pois consegue se ajustar a ambientes com luminosidade diferente contando com ponto positivo na qualidade da imagem limiarizada. Os efeitos da luz ambiente na imagem da cédula são minimizados, (figura 5.7), comparação entre as técnicas de otsu e NiBlack.

“O valor do limiar é obtido de forma adaptativa através da média e do desvio padrão nas regiões da imagem, esta técnica consiste em deslocar uma janela sobre a imagem original, calculando os limiares somente no local e assim diminuindo o problema com a luminosidade diferente existentes nos pontos da imagem.” **(RODRIGUES & OLIVETE, 2005)**



**Figura 5.7 – Em (A) o resultado da aplicação do método de Otsu. Em (B) o resultado da aplicação do método Niblack. (RAMALHO, 2005)**

As figuras 5.7, 5.8 e 5.9 apresentam os testes na imagem das cédulas monetárias:



Figura 5.8 - Imagem Original



Figura 5.9 - O resultado da aplicação do método de Otsu. .



Figura 5.10 - O resultado da aplicação do método Niblack

Não é perceptível a diferença nas imagens binarizadas pelo método de otsu e pelo método Niblack a olho nu. A opção de se trabalhar com essa técnica vem pela motivação da luminosidade, considerando que em um mesmo ambiente pode ocorrer variações de luz durante o mesmo dia.

Considerando uma dimensão  $N \times N$  pixels e uma imagem  $I$ , o limiar é calculado segundo a equação:

$$L = \mu(i; j) + \text{Bias}(i; j)$$

Onde:

$$i = 1, \dots, N;$$

$$j = 1, \dots, N;$$

Os parâmetros para essa técnica são tamanho  $N$ , em pixels, da janela  $N \times N$  e o peso Bias, que é o fator de peso (desvio padrão). E apesar das variâncias de iluminação as características da imagem são preservadas, como demonstra a Figura 5.11. Quando se trata de imagens coletadas em lugares indeterminados, essa preservação é muito importante, pois cada imagem capturada a iluminação vai ser diferente.





Figura 5.11 - Diferença de iluminação nas notas (RAMALLHO, 2005), p. 28.

A diferença entre ambientes com pouca, média ou alta iluminação influenciam diretamente no reconhecimento da imagem, ressaltando que ao trabalhar com imagens vetoriais é comum alternar seus valores de acordo com a intensidade da luz. O resultado é a imagem binarizada em dois tons de cinza (0 e 255)

#### 5.4 Isolando o valor da cédula.

Após a binarização o valor da cédula passa a ser isolado, e através de tentativas foi encontrado quarenta por cento da imagem composta por linhas e sessenta por cento da imagem composta por colunas, medidas que encaixam qualquer um dos valores da cédula situados à direita da imagem.



Figura 5.12 - Imagem com o limite do corte



Figura 5.13 - Imagem resultado do corte

A imagem nova é enviada para a classe Clean.java, que tem a função de realiza a passagem de um filtro para suavizar a imagem, e se encontrar pixels corrompidos na imagem,

estes são alterados para o maior valor (255) e assim retirar eventuais ruídos causados pelo corte. Para suavizar a imagem foi usado o conceito de vizinhança, onde foi criada uma janela 3X3, como demonstra a Figura 5.14:

0	255	0
255	0	255
0	255	0

Figura 5.14 - área definida para filtrar

Diante da figura 5.14, observa-se que há em determinadas casas o algarismo 0. Ressaltando que nesse procedimento o canto superior esquerdo e direito, o canto inferior esquerdo e direito e o pixel central representarem 0, todo o conteúdo da janela recebe o valor máximo (255), consequentemente todo o espaço ficará branco (Figura 5.15)

255	255	255
255	255	255
255	255	255

Figura 5.15 – área filtrada

Deixando a imagem em forma de esqueleto, como mostra a Figura 5.16.

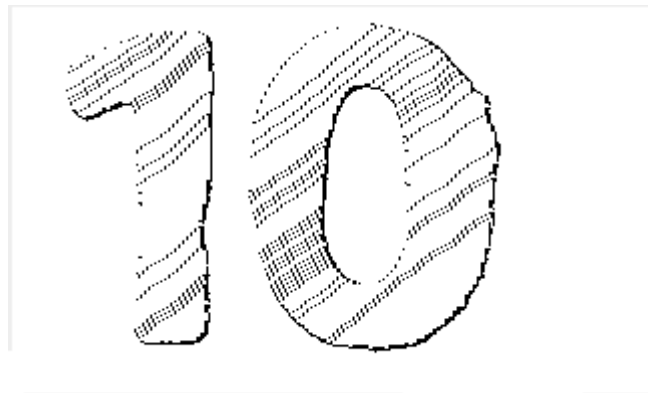
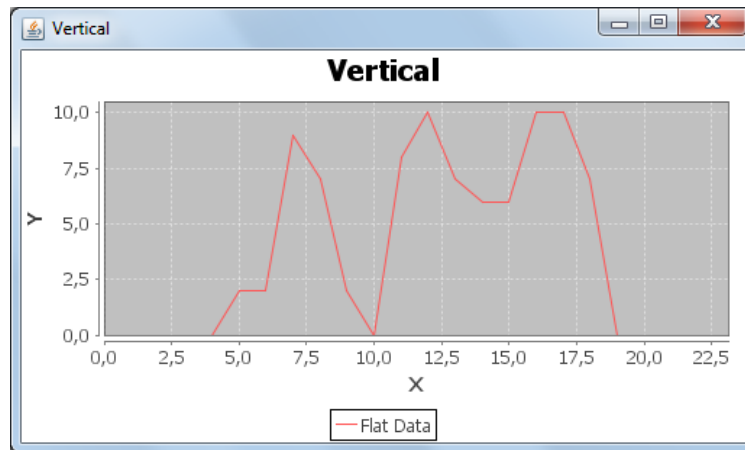


Figura 5.16 – Nota após a passagem do filtro

A seguinte etapa é realizar um histograma vertical definindo novos pontos de corte, selecionando os pontos iniciais e finais da imagem (Figura 5.17).





**Figura 5.17 – Histograma Vertical**

Definindo os pontos de corte, a figura anterior é enviada para a classe `cortaImagem.java`, uma nova imagem em tons de cinza, por padrão é preenchida com todos os pixels valendo 0, ocasionando mais ruídos para a imagem após a transferência. Para eliminar os ruídos ou mesmo manchas na imagem é preciso acontecer uma rotina onde o 0 é convertido para 255. Aproveitando os pontos de corte outra rotina é feita para transferir apenas o valor da cédula para nova imagem, já preparada para receber o desenho.

## 5.5 Centralizando a Imagem

O próximo passo é a centralização da imagem.

Feito um histograma vertical da figura, encontrada a posição inicial e a final sendo que a última foi descoberta pela inversão do vetor que acontece contando os pixels do final para o começo. A captura do valor foi descoberta através de uma variável do tipo inteiro, que insere uma condição para acumular os primeiros valores encontrados e de acordo com o resultado ocorre alteração encontrando as posições.

Assim teremos o tamanho da imagem através da subtração da posição final da posição inicial, esse valor será dividido por 2, e inserido no centro da imagem, demonstrado pela Figura 5.18.

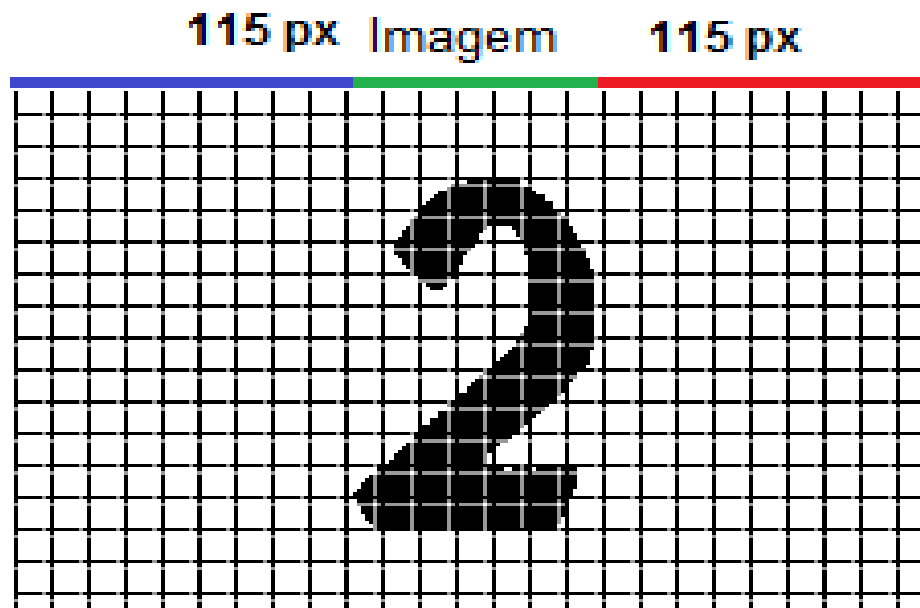


Figura 5.18 – a imagem Centralizada

Foi usado um algoritmo para inserir o valor 255 em todos os pixels, após a criação da nova imagem o sistema insere a área do valor da cédula e parte da imagem não receberá alteração, como mostra a figura 5.19:



Figura 5.19 – sem a mudança de tom

## 5.6 Diminuir Imagem

Com o auxílio do objeto `BufferedImage` e outro da classe `Graphics2D` é possível converter o tamanho da imagem ao padrão desejado.

Esse método foi utilizado duas vezes no projeto:

Na primeira conversão, após a centralização da imagem, passou de 320x192 para 230x150

Na segunda conversão, de 230x150 para 23x15.

Esse tamanho irá aperfeiçoar o trabalho da Rede Neural Artificial(RNA), conseguindo aumentar a performance sem perder a qualidade.

## 5.7 Executando a RNA

É criada uma matriz bidimensional que receberá os pixels da imagem, estes convertidos em -1 para todos os pixels com valor 0 e 1 para todos os pixels com valor 255. Após serão inseridos em uma matriz unidimensional (Vetor) de tamanho 345, que é exatamente a multiplicação das linhas (15) pelas colunas (23). Então é enviado para a classe testeRna.java de onde será calculado e retornado um índice entre 0 e 5, de acordo com a tabela 5.1.

**Tabela 5.1 - Tabela referência de valores**

Valor da nota	Índice
2	0
5	1
10	2
20	3
50	4
100	5

## 5.8 Comparando os índices.

Finalizados os índices, uma estrutura condicional (if's encadeados) verificará em qual nota ele irá se encaixar após ter feito o reconhecimento do valor, então é acumulado em uma variável do tipo *integer* que irá guardar o valor total das notas passadas pelo sistema e cada valor reconhecido é enviado para a classe conectaVoz.java. O sintetizador de voz será conectado ao programa replicando a mensagem que vem do módulo principal, ou seja o valor da nota reconhecida (Figura 5.20).

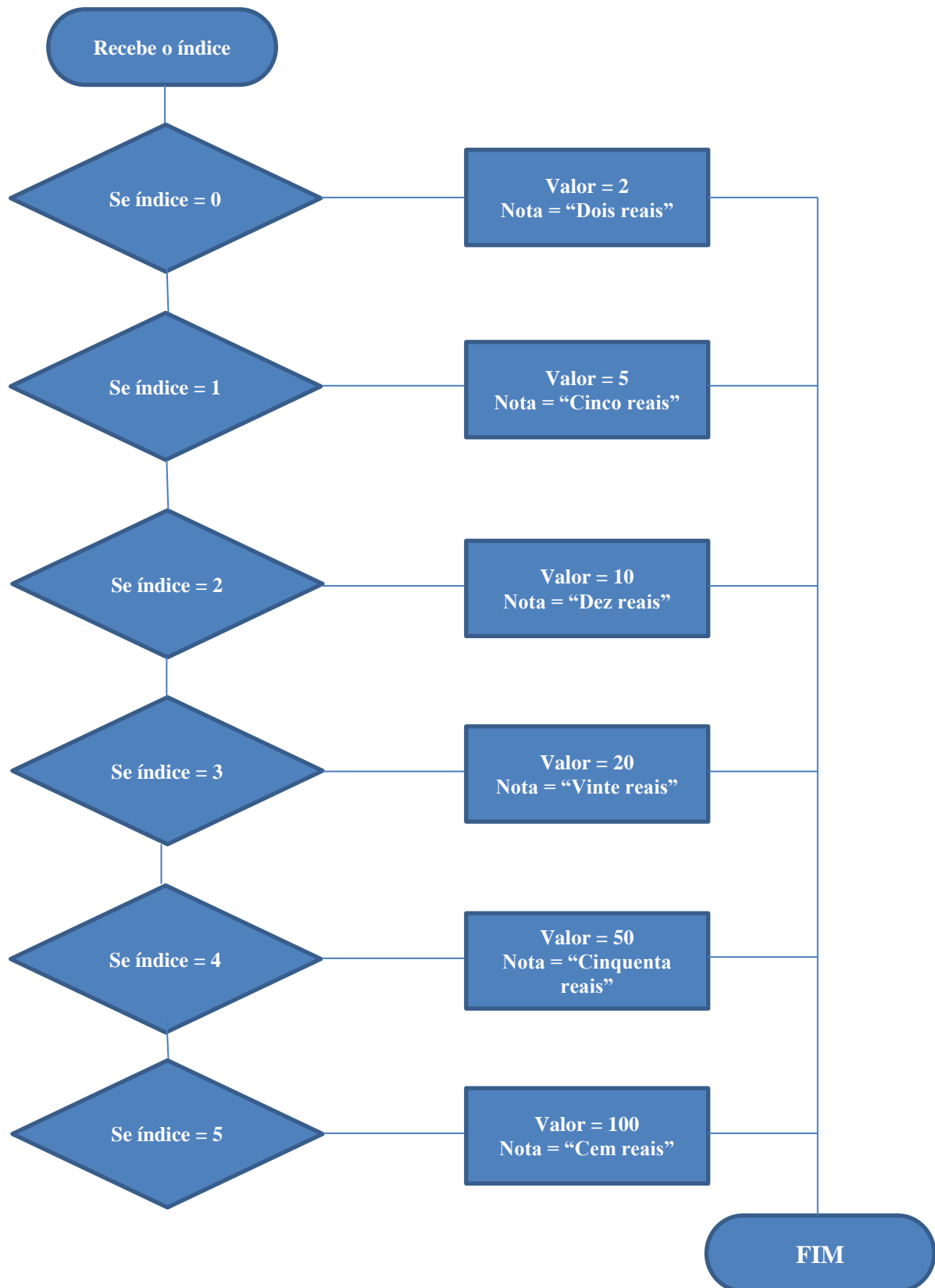


Figura 5.20 - Condição de comparação para retornar o índice.

## 5.9 Dificuldades encontradas

No decorrer do estudo, houve momentos em que classes criadas não concluíram a ideia esperada.

### 5.9.1 Rotacionar a imagem

Para alinhar a imagem, foi aplicada a seguinte formula:

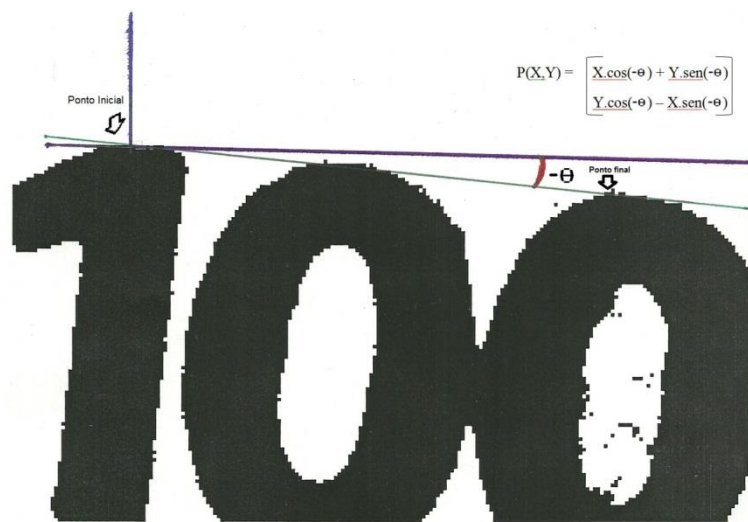


Figura 5.21 - imagem para rotacionar A

Foi calculado o ângulo da imagem, utilizando a formula:

$$P(x,y) = P'(x',y')$$

$$P'(x',y') = \begin{bmatrix} \cos\theta - \text{sen}\theta \\ \text{sen}\theta + \cos\theta \end{bmatrix} P(x,y)$$

onde:

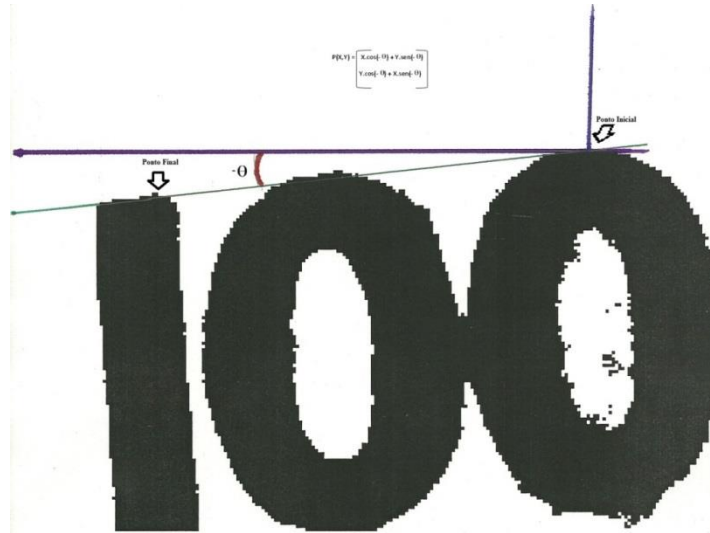
$$x' = \cos\theta - \text{sen}\theta$$

$$y' = \text{sen}\theta + \cos\theta$$

Realizado o ajuste do valor da cédula, foi encontrado o ponto inicial, seguindo os pixels da esquerda para a direita, de cima para baixo e o ponto final da direita para a esquerda,

também de cima para baixo, como mostra a figura 5.21, calculado o ângulo é realizada a rotação.

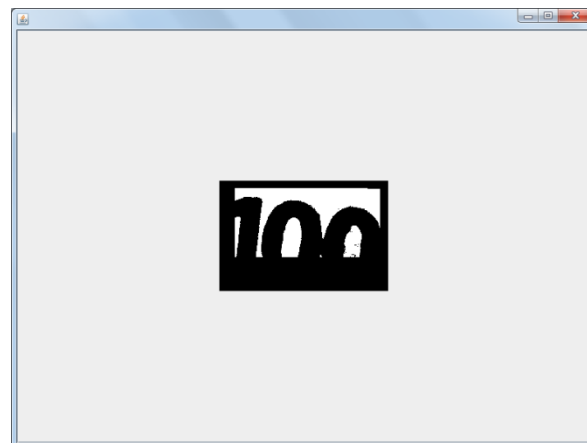
Caso a imagem necessite rotacionar ao contrário, como mostra a imagem 5.22.



**Figura 5.22 - Rotação ao contrário**

É verificado se a posição inicial do lado esquerdo está abaixo da posição final do lado direito, caso positivo então é invertido o processo de busca, procurando o ponto inicial utilizando os pixels da direita para a esquerda, de cima para baixo e o ponto final utilizando os pixels da esquerda para a direita, também de cima para baixo.

Os testes realizados com as imagens fora de alinhamento mostraram que, aplicando as técnicas elas realmente alinham-se, porém devido ao deslocamento dos pixels fora da área da imagem ocasiona a formação de ruídos do tipo “pimenta”, o que não é interessante para o sistema, como mostra as figuras 5.23 e 5.24



**Figura 5.23 - imagem recebida**

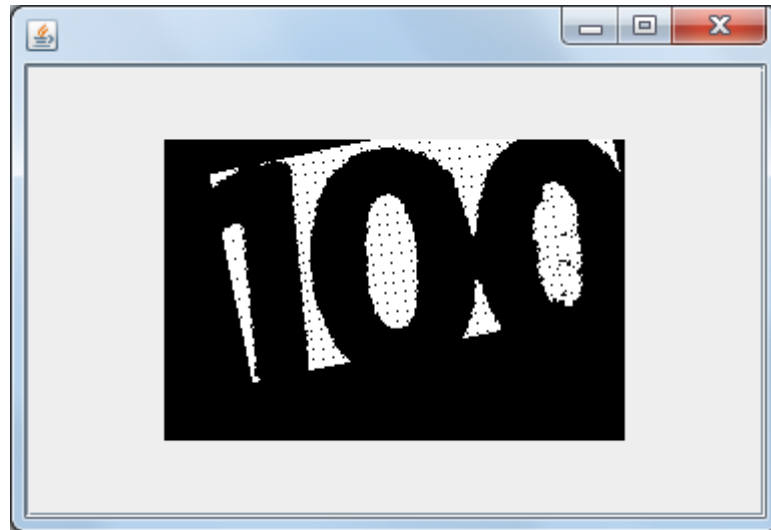


Figura 5.24 – Imagem com maior rotação..

A figura 5.24 apresenta com nitidez o erro cometido, observa-se também que a imagem está com um deslocamento maior, realizado propositalmente para salientar os ruídos que o método apresenta.

```

novo_x = (int) ( x * Math.cos(ang) + y * Math.sin(ang));
novo_y = (int) ( y * Math.cos(ang) - x * Math.sin(ang));

if ((novo_y < 640) && (novo_x < 480) && (novo_y > -1) && (novo_x > -1))
{
    System.out.println("x "+novo_x+" - "+novo_y);
    color = new Color(original.getRGB(y, x));
    raster.getPixel(y, x, pix);
    p = color.getBlue();
    raster.setSample(novo_y, novo_x, 0, p);
}

```

Calculo para encontrar o novo ponto

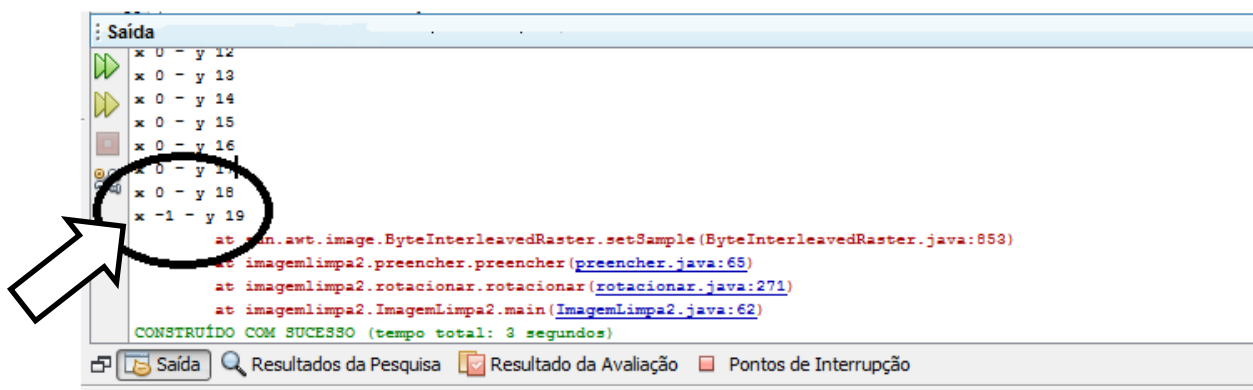
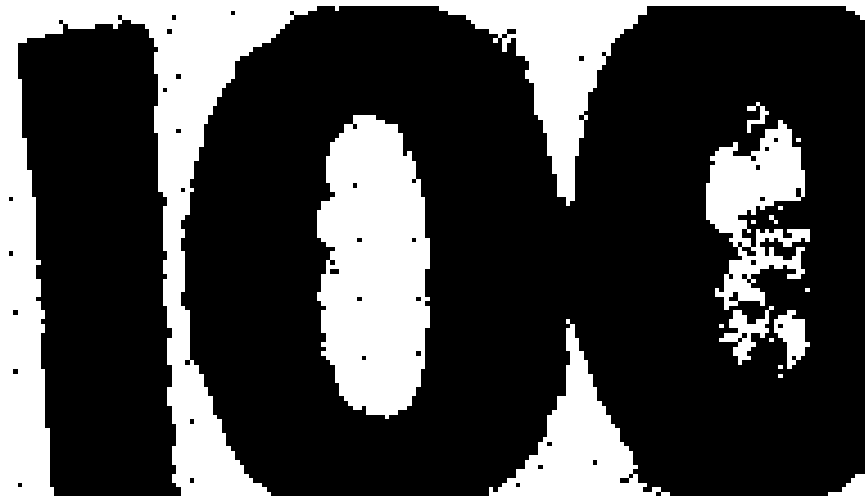


Figura 5.25 - erro ocorrido durante o calculo de rotação

Na figura 5.25, o cálculo da rotação para a posição novo\_x tende a ir para um valor abaixo de zero, que está fora da matriz da imagem, ocasionando erro no sistema.

Para evitar o erro foi criada uma estrutura condicional que mantém o valor antigo do pixel, o que acaba produzindo o efeito pimenta.

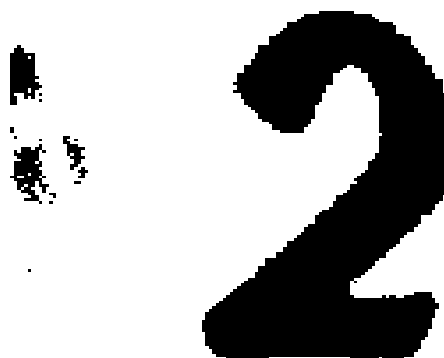
A figura 5.26 mostra a imagem rotacionada de maneira correta com o efeito pimenta.



**Figura 5.26 - figura rotacionada**

Apesar de rotacionada corretamente ainda apresenta excesso de ruído, interferindo na imagem quando lançada na rede neural, essa sujeira causa anomalia na rede e evita que o valor da cédula seja reconhecida corretamente.

Outro problema ocasionado pela captura da imagem é a sujeira que impede a rotação da figura, nessa condição ela trabalha de maneira incorreta (Figura 5.27 ).



**Figura 5.27 - Imagem manchada.**

A utilização de outros filtros de limpeza seria uma imprudência, visto que pode ocorrer uma confusão e o ruído não se desfazer, e ainda o risco de remover um número da imagem.



### 5.9.2 Reconhecimento de cores

Esse módulo do sistema seria responsável por realizar o reconhecimento da cédula monetária através das cores que cada uma apresenta. (CAMPOS & MACHADO, 2009) cita que “Quando a cor é a principal característica envolvida no problema, é necessário escolher um modelo para o espaço de cores.” A identificação da cédula baseia-se no valor dos pixels da imagem natural, ou seja, a primeira imagem capturada e armazenada. Foi realizada uma rotina agregando uma área pré-determinada na imagem, retirando os valores médios de cada pixel da área e assim retornando o índice equivalente a nota encontrada.

Abaixo as tabelas com os valores referentes às cédulas monetárias:

**Tabela 5.2 - Tabela de cores para a nota 2 reais**

				SlateBlue	106-090-205	#6A5ACD
				SteelBlue	070-130-180	#4682B4
				LightSteelBlue	176-196-222	#B0C4DE
Nota	R	G	B			
2	177	167	157			
	142	152	161			
	167	166	146			
	167	159	140			
	170	172	159			
	184	180	168			
	140	34	158			
Média dos valores						
R	G	B				
163,86	147,14	155,57				



**Tabela 5.3 - Tabela de cores para a nota de 5 reais**

				Orchid4	139-071-137	#8B4789	
				Plum4	139-102-139	#8B668B	
				MediumOrchid4	122-055-139	#7A378B	
				DarkOrchid4	104-034-139	#68228B	
				Purple2	145-044-238	#912CEE	
				Purple3	125-038-205	#7D26CD	
				Purple4	085-026-139	#551A8B	
				MediumPurple3	137-104-205	#8968CD	
				MediumPurple4	093-071-139	#5D478B	

Tabela 5.4 - Tabela de cores para a nota 10 reais

Nota	R	G	B	IndianRed2	2389999	#EE6363
10	131	103	92	Firebrick2	2384444	#EE2C2C
	147	130	136			
	129	94	100			
	133	94	89			
	126	129	134			
Média dos valores						
R	G	B				
133,2	110	110,2				




Tabela 5.5 - Tabela de cores para a nota 20 reais

Nota	R	G	B	Orange1	255-016-050	#FFA500
20	111	68	62	Orange2	238-150-040	#EE9A00
	108	86	65	Orange3	205-133-0	#CD8500
	93	70	62	Orange4	139-090-0	#8B5A00
	102	64	51	DarkOrange1	255-127-0	#FF7F00
	95	56	39	DarkOrange2	238-118-0	#EE7600
				DarkOrange3	205-102-0	#CD6600
Média dos valores						
R	G	B				
101,8	68,8	55,8				




Tabela 5.6 - Tabela de cores para a nota 50 reais

Nota	R	G	B	Sienna1	25513071	#FF8247
50	153	81	96	Sienna2	23812166	#EE7942
	60	65	69	Sienna3	20510457	#CD6839
	67	63	64			
	72	70	75			
	84	76	73			
Média dos valores						
R	G	B				
87,2	71	75,4				





Tabela 5.7 - Tabela de cores para a nota 100 reais

Nota	R	G	B		
100	53	77	77	RoyalBlue2	67110238 #436EEE
	45	71	84	RoyalBlue3	5895205 #3A5FCD
	44	73	81	RoyalBlue4	3964139 #27408B
	47	66	64		
	58	74	74		
Média dos valores					
	R	G	B		
	49,4	72,2	76		



A tabela 5.8 mostra o resultado atingido por essa técnica utilizando a cédula de valor R\$5,00 para os testes.

Tabela 5.8 - tabela de resultados para comparação das cores

Valores das cores	Resultado retirado da cédula	Valor encontrado	Valor correto
1 – true && true && false 137<141 && 130<131 && 131<130	Cedula2 Altura: 20 - largura: 40 Vermelho: 137 - Verde: 130 – Azul: 131	R\$ 50,00	R\$ 5,00
2 – true && false && false 140<141 && 146<140 && 140<146	Cedula8 Altura: 20 - largura: 40 Vermelho: 140 - Verde: 146 – Azul: 140	Nota Desconhecida	R\$ 5,00
3 – true && true && true 133<141 && 130<133 && 129<130	Cedula14 Altura: 20 - largura: 40 Vermelho: 133 - Verde: 130 - Azul: 129	R\$ 5,00	R\$ 5,00
4 – true && false && true 140<141 && 147<140 && 139<147	Cedula20 Altura: 20 - largura: 40 Vermelho: 140 - Verde: 147 – Azul: 139	Nota Desconhecida	R\$ 5,00
5 – true && true && false 136<141 && 130<136 && 132<130	Cedula26 Altura: 20 - largura: 40 Vermelho: 136 - Verde: 130 - Azul: 132	R\$ 50,00	R\$ 5,00

Um dos principais motivos para esse método não funcionar foi à ausência ou excesso de luminosidade aplicada na cédula, uma vez que a intensidade da luz altera o resultado. Outro fator importante é a condição da cédula, caso ela esteja muito gasta ou tenha sido lavada, passa a alterar as suas cores primárias, devido a pouca confiabilidade informada nas tabelas apresentadas anteriormente.

## 6. Resultados obtidos

A tabela 6.1 ilustra os resultados obtidos através dos testes verificados com o sistema, onde “nv” é o número de vezes que a nota foi capturada e transferida para sistema, as colunas “2”, “5”, “10”, “20”, “50”, “100” são as vezes que o sistema confundiu o valor das notas, coluna de percentual de erros e coluna de percentual de acertos.

**Tabela 6.1 – A tabela de resultados dos testes com amostra de 628 imagens capturadas pelo sistema, as duas primeiras colunas mostram o valor da cédula monetária e o número de imagens capturadas por cada valor. As colunas 3 a 9 apresentam a matriz confusão para os testes. A coluna 10 apresenta a quantidade de respostas certas, a colunas 11 mostra o percentual de erros e a coluna 12 o percentual de acertos.**

Resultados										
Valor	Nv	2	5	10	20	50	100	Respostas Certas	% Erros	% Acertos
2	109		6					103	5,50	94,50
5	112	2				2		109	3,79	97,32
10	100							100	0,00	100,00
20	100							100	0,00	100,00
50	105							105	0,00	100,00
100	102							102	0,00	100,00

Onde:

**Tabela 6.2 - Tabela de classificação**

Percentual	Classificação
>95%	Ótimo
>85% e <94%	Muito Bom
>65% e <84%	Bom
>50% e <64%	Regular
<49%	Péssimo

De acordo com a tabela 6.2, foi considerado satisfatório o desempenho do reconhecimento da rede neural, visto que houve apenas uma nota abaixo de 95% considerado muito bom.

Após cento e nove tentativas houveram seis respostas erradas, totalizando uma taxa de erro de 5,5%, considerando que as cédulas que obtiveram falhas são antigas e em contraposição as outras que obtiveram bom resultado são novas lançadas pelo Banco Central como a segunda família do Real.

De acordo com o site de economia da uol (UOL, 2012), as novas cédulas estão sendo desenvolvidas desde de 2003 pelo Banco Central em conjunto com a Casa da Moeda do Brasil(CMB). A expectativa é que todas as déduas venham a ser da segunda família, onde os melhores resultados foram obtidos.

**Tabela 6.3 - Tabela de resultados**

Valor	nv	Respostas Certas	% Erros	% Acertos	Resultado
2	109	103	5,50	94,50	Muito Bom
5	112	109	3,79	97,32	Ótimo
10	100	100	0,00	100,00	Ótimo
20	100	100	0,00	100,00	Ótimo
50	105	105	0,00	100,00	Ótimo
100	102	102	0,00	100,00	Ótimo

A tabela 6.3 apresenta o resultado de acordo com a classificação da tabela 6.2, apesar do bom desempenho encontrado nas cédulas, é necessário considerar que a taxa de erros tem que ser 0%, garantindo que o sistema alcançará nível de excelência, ou seja 100%.

“Software permeia o dia a dia das pessoas no mundo atual. Existem aplicações em elevadores, em aparelhos de TV, em automóveis, em celulares, etc. Além disso, existem também as aplicações consideradas críticas como software de controle de usinas nucleares, de apoio em aeronaves, embutido em mísseis e radares, controle de pacientes em hospitais, etc.

Devido a esta forte dependência, a ocorrência de algum problema em aplicações de software, pode levar a situações de grandes financeiras e até mesmo perda de vidas humanas.

Exemplos:

Ariane: Explosão do foguete 40 segundos após a decolagem;

Therac: Entre 1985 e 1987, 6 acidentes causando mortes por overdoses de radiação;

Denver airport: Atraso na abertura do aeroporto devido a erros no sistema automático de transporte de bagagens com perdas estimadas em US\$360 Milhões;” (CRESPO, 2011)

## 7. Conclusões e trabalhos futuros

Nesse trabalho foi desenvolvido um estudo no intuito de investigar as técnicas de processamento digital de imagens, a proposta de um sistema computacional de leitura automática de valores em cédulas monetárias. Para maior esclarecimento foi necessário ressaltar alguns conceitos:

- ✓ Aquisição de imagens utilizando java;
- ✓ Reconhecimento de padrões;
- ✓ Processamento digital de imagens utilizando a API JAI como ferramenta de manipulação de imagens;
- ✓ Redes Neurais Artificiais do tipo MLP e a
- ✓ Uso da API JSP com o sintetizador de voz.

Os estudos conduzidos nessa pesquisa deram segmento a uma contribuição para um sistema computacional de leitura automática de valores em cédulas monetárias. Foram analisados aspectos de implementação, trabalhando com programação Orientada a Objetos com a ferramenta Java, rotinas de interação com o usuário visando a facilidade no manuseio do sistema, rotinas de trabalho com processamento digital de imagens, redes neurais artificiais e finalmente a rotina de conexão com o sintetizador de voz.

Pesquisas relacionadas à programação orientada a objetos com java, permitiram a aquisição de conhecimentos relacionados a programação para aquisição de imagem via software, bem como a aplicação de algoritmos para reconhecimento de câmeras, acumulador de valores e comparação de índices, envolvidos com o reconhecimento das cédulas.

É importante destacar que foram encontradas na literatura, pesquisas semelhantes ao desenvolvimento do trabalho proposto, porém pelos diferentes tipos de aplicação existentes e pelo fato da maioria fazerem parte de softwares proprietários, não atendem na sua totalidade ao público acadêmico. Neste sentido, técnicas de captura de imagens, processamento digital de imagens, reconhecimento de padrões já conhecidos, mas encontrados de forma fragmentada dentro da bibliografia estudada, e assim, foram executadas de maneira ordenada o que auxiliou de forma significativa a montagem do projeto.

De acordo com os objetivos propostos:

1. Encontrar na linguagem Java as melhores técnicas para a realização da captura de imagem a partir de uma webcam, realizada em ambientes com variação de iluminação;

SOLUÇÃO – Foi usado a aplicação *Java Media Framework* (JMF) para aplicações multimídia, usado principalmente para realizar captura de vídeo e áudio com a linguagem Java, disponibiliza o recurso de

alteração de valor do pixel, o que facilitou o desenvolvimento do trabalho.

2. Buscar os métodos de processamento digital de imagens que realizem o tratamento da imagem capturada através de técnicas de limiarização, retirada de ruídos e movimentação de pixels;

SOLUÇÃO – A subclasse *BufferedImage* gerada pela aplicação JMF, proporcionou a realização da binarização nas imagens capturadas, a passagem dos filtros de redução de ruídos, a translação e a redução da imagem, com isso foi possível testar vários métodos de processamento digital de imagens e adequá-los às necessidades do sistema.

3. Pesquisar as redes neurais artificiais (RNA), avaliando os métodos existentes que possam realizar o reconhecimento do valor encontrado com maior eficácia e rapidez no treinamento;

SOLUÇÃO – Por ser uma rede já testada, de resposta rápida e com ótima confiabilidade foi escolhida a RNA *Multilayer Perceptron* (MLP), que trouxe para o projeto um ganho de tempo, qualidade e desempenho, tanto para o uso do hardware quanto para a qualidade do software.

4. Analisar os sintetizadores de voz existentes que melhor se adaptam à linguagem Java.

SOLUÇÃO – Após a realização dos testes com os sintetizadores disponíveis no mercado, optou-se pelo FreeTTS que é um sintetizador de voz feito totalmente na linguagem java que em conjunto com o mbrola converte o texto em fala português.

## 7.1 Contribuições

Durante o desenvolvimento deste projeto de pesquisa buscou-se acrescentar conhecimentos à comunidade acadêmica, com destaque às principais vantagens;

- ✓ Aplicação de diferentes técnicas de Processamento digital de imagens, destacando as técnicas de limiarização estudadas (global e multiníveis);
- ✓ Aplicação das técnicas de captura de imagem diretamente de uma webcam, apresentada pela linguagem Java, utilizando a API JAI;
- ✓ A adaptação de algoritmos existentes, na execução de filtros para retirada de ruídos da imagem limiarizada, gravação de arquivos do tipo imagem em disco rígido, sendo utilizados para captura inicial e para testes das classes que manipulam imagens;
- ✓ A adaptação de um sistema de redes neurais artificiais do tipo MLP para o reconhecimento da cédula;
- ✓ Técnicas apresentadas para diminuição da imagem sem a perda da qualidade e consequentemente causando um aumento na performance da RNA, tanto durante o seu treinamento quanto na execução da mesma no reconhecimento da cédula pelo sistema;
- ✓ A aplicação de algoritmos para movimentação de pixels na imagem, realizando o ajuste sem perder a qualidade e sem ocasionar ruídos;

## **7.2 Trabalhos Futuros**



Sugestões para trabalhos futuros:

- ✓ A ampliação do sistema com um “localizador” da cédula monetária na imagem original;
- ✓ A implantação do projeto em um sistema operacional mobile android, atendendo assim à Tecnologia Assistiva;
- ✓ A criação de um modulo de reconhecimento de notas falsas, oferecendo mais confiabilidade no sistema;
- ✓ A implantação do projeto em um hardware de contagem de cédulas, alterando assim o funcionamento do equipamento e trazendo benefícios de desempenho com o trato do dinheiro;
- ✓ Realização de novos estudos envolvendo a Tecnologia Assistida adaptando o sistema para textos;

## **8. Referências**

abcegos, 2012. *Associação Baiana de Cegos*. [Online]  
Available at: <http://www.abcegos.org.br/deficiencia-visual/>  
[Acesso em 01 02 2013].

ALBUQUERQUE, M. P. & ALBUQUERQUE, M. P., 2000. *Processamento de Imagens: Métodos e Análises*. Rio de Janeiro: Centro Brasileiro de Pesquisas Físicas.

ARAÚJO, A. F. R., 2012. *Redes Neurais Fundamentos*, Pernambuco: UFPE.

ARAÚJO, G. M., 2010. *Algoritmo para Reconhecimento de Características Faciais Baseado em Filtros de Correlação*, Rio de Janeiro: COOPE/UFRJ.

BERNI, J. C. A., 2008. *Desenvolvimento e Implementação de Métodos de Correção de Iluminação para Imagens Digitais*, Santa Maria - RS: Universidade Federal de Santa Maria.

BERSCH, R. d. C. R., 2008. *Introdução à Tecnologia Assistiva*, Porto Alegre: CEDI - Centro Especializado em Desenvolvimento Infantil.

CAMPOS, D. A. R. & MACHADO, A. M. C., 2009. Um sistema de identificação de cédulas monetárias para indivíduos portadores de deficiência visual. *REIC - REVISTA ELETRÔNICA DE INICIAÇÃO CIENTÍFICA*, 01 09, p. 11.

CASTRO, A. A. M. d. & PRADO, P. P. L. d., 1999-2002. Algoritmos para Reconhecimento de Padrões. *Revista Ciências Exatas Taubaté*, 15 08, pp. 129-145.

CASTRO, A. A. M. d. & PRADO, P. P. L. d., 2002. ALGORITMOS PARA RECONHECIMENTO DE PADRÕES. *Rev. Ciênc. Exatas*, 15 08, p. 17.

CERA, M. C., 2005. Uso de Redes Neurais para o Reconhecimento de Padrões. 20 10.

COSTA, S. M. F., 2001. *Classificação e Verificação*, São Paulo: Escola Politécnica da Universidade de São Paulo.

CRESPO, A. N., 2011. *5CQualiBR*. [Online]  
Available at: <http://www.softwarepublico.gov.br/5cqualibr/xowiki/Teste-item3>  
[Acesso em 01 03 2013].

FAUSET, L., 1994. *Fundamentals of Neural Networks - Architectures, Algorithms and Applications*. 1ª ed. New Jersey: Prentice Hall.

Gil, F. d. O., 2010. *AUIRE Tecnologias Acessíveis*. [Online]  
Available at: <http://www.auire.com.br/>  
[Acesso em 12 Fevereiro 2013].

GUEDES, G. T. A., 2004. *UML Uma abordagem Prática*. 1ª ed. São Paulo: Novatec.

LIU, Y. L. A. et al., 2008. Uso de Rede Neural Percéptron Multi-Camadas na Classificação de Patologias Cardíacas. *TEMA - Tendências em Matemática Aplicada e Computacional*, 02 09, pp. 255-264.

LookTel, 2012. LookTel. [Online]  
Available at: <http://www.looktel.com/>  
[Acesso em 10 Fevereiro 2013].

MARANGONI, J. B. & PRECIPITO, W. B., 2006. Reconhecimento e Sintetização de Voz Usando Java Speech. *REVISTA CIENTÍFICA ELETRÔNICA DE SISTEMAS DE INFORMAÇÃO*, I(4), pp. 1-10.

MENDES, D. Q. & OLIVEIRA, M. F. d. S., 2013. *Tutorial de Redes Neurais - Aplicações em Bioinformática*. [Online]  
Available at: [http://www.lncc.br/~labinfo/tutorialRN/frm4\\_perceptronMultiCamadas.htm](http://www.lncc.br/~labinfo/tutorialRN/frm4_perceptronMultiCamadas.htm)  
[Acesso em 22 Março 2013].

MIRANDA, R., 2009. COLMEIA. [Online]  
Available at: <http://www.colmeia.blog.br/sintetizador-de-voz-gratis-e-online.html>  
[Acesso em 25 Março 2013].

MORETO, M. & ROLIM, J. G., 2010. Análise automática de oscilografias em sistemas elétricos de potência. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, 1 Julho/Agosto, 21(4), pp. 347-362.

NNSolutions, 2010. *NNSolutions Desenvolvimento de Sistemas Integrados*. [Online]  
Available at: <http://www.nnsolutions.srv.br/slep.html>  
[Acesso em 10 Fevereiro 2013].

NOVAIS, A., NOVAIS, F., ARAÚJO, R. & NEVES, T., 2010. *Redes Neurais Artificiais*, Bahia - Ba: Universidade Federal da Bahia.

PFÜTZENREUTER, E., 2012. Epx. [Online]  
Available at: <http://epx.com.br/artigos/economianeural.php>  
[Acesso em 25 Março 2013].

RAMALLHO, M. C., 2005. *Ambiente Computacional Para Identificação da Ordem De Produção Estampadas Em Tarugos de Aço da Açominas*, Belo Horizonte - MG: UFMG.

RAMOS, J. P. S., 2003. Redes neurais artificiais na classificação de frutos: cenário bidimensional. *Ciência e Agrotecnologia*, Março/Abril, 27(2), pp. 356-362.

RODRIGUES JUNIOR, J. F. & MOREIRA, D. A., 2000. *Estudo e Desenvolvimento de Aplicações Java com Reconhecimento e Síntese de Voz*, São Paulo: Instituto de Ciências Matemáticas e de Computação .

RODRIGUES, E. L. L. & OLIVETE, J. C., 2005. Comparação entre Métodos de Limiarização Aplicados na Extração de Características de Imagens Carpais. *X Congresso Brasileiro de Física Médica*, 26 05, p. 4.

SANTOS, A. M. d., SEIXAS, J. M. d., PEREIRA, B. B. & MEDRONHO, R. A., 2005. USANDO REDES NEURAS ARTIFICIAIS E REGRESSÃO LOGÍSTICA NA PREDIÇÃO DA HEPATITE A. *Revista Brasileira de Epidemiologia*, 8(2), pp. 117-126.

SANTOS, R., 2004. JAI: Java Advanced Imaging.. *Revista de Informática Teórica e Aplicada*, Volume 11, pp. 93-124.

SILVA, I. N. d., SPATTI, D. H. & FLAUZINO, R. A., 2010. *Redes Neurais Artificiais para engenharia e ciências aplicadas*. 1ª ed. São Paulo: ArtLiber.

SOUZA, G. S. S. d. & PASSELA, P. H., 2011. *Reconhecimento Automático de Placas de Veículos*, Ribeirão Preto: UniSEB COC.

TORRES, E. F. & MAZZONI, A. A., 2004. Conteúdos digitais multimídia: o foco na usabilidade e acessibilidade. 26 julho, p. 5.

UOL, 2012. *UOL Notícias*. [Online]  
Available at: <http://economia.uol.com.br/ultimas-noticias/redacao/2012/07/23/banco-central-lanca-hoje-novas-cedulas-de-r-10-e-r-20-veja.jhtm>  
[Acesso em 27 Fevereiro 2013].