



BINARIZAÇÃO DE IMAGENS DE CHEQUES

Trabalho de Conclusão de Curso

Engenharia da Computação

Nome do Aluno: Maíra Saboia da Silva

Orientador: Prof. Carlos Alexandre Barros de Mello



MAÍRA SABOIA DA SILVA

**BINARIZAÇÃO DE IMAGENS DE
CHEQUES**

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia da Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Recife, Junho 2009.

*“A mente que se abre a uma nova idéia jamais
voltará ao seu tamanho original” (Albert Einstein)*

A Deus, aos meus pais e minhas irmãs.

Agradecimentos

Quero agradecer primeiramente a Deus que tornou tudo possível. Agradecer aos meus pais, Paulo e Mariândela, que abdicaram da convivência comigo, por quase nove anos, pois sabiam que estavam fazendo o melhor pra mim e para o meu futuro. Agradeço às minhas irmãs, Naara e Tainá, por todo apóio e amizade. Aos meus avôs e a toda minha família, que cuidaram de mim por todos esses anos que morei em Recife.

Tenho muito a agradecer ao meu Orientador, Carlos Alexandre, que sempre demonstrou interesse no meu crescimento e me apoiou muito durante minha graduação. Agradeço às pessoas que trabalham comigo na FITec. Amigos que me ajudaram e me apoiaram nesta fase final do curso.

Agradeço por um amigo especial que Deus colocou na minha jornada, Henrique Reinaux. Agradeço por todo apóio, compreensão, companhia e paciência que ele teve e tem comigo. Quero agradecer aos meus amigos, todos os amigos que fiz enquanto estive em Recife. Amigos que me acordavam no meio da noite para fazer a monografia, amigos que liam meu trabalho, amigos que escutavam meus desabafos, amigos que estiveram presente nos bons e maus momentos. Todos eles foram importantes para essa conquista e fizeram meus dias serem mais felizes.

Resumo

O processamento de cheque é um dos poucos processos bancários que não é totalmente automatizado. Por ser uma aplicação bancária, seu processamento exige segurança e precisão. Contudo, sua automatização não é uma tarefa simples, pois os bancos visando diminuir a possibilidade de falsificação adicionam elementos de complexidade ao background do cheque. Sendo assim, binarizar esse tipo de imagem não é uma tarefa trivial. Nesta pesquisa foi estudado e implementado um algoritmo de binarização proposto por Neves, Mello e Silva [14][15][16][17], ele é focado no processamento de imagens de cheques. Ele se baseia na entropia de Tsallis para a determinação do limiar de binarização das imagens. Esse algoritmo apresentou resultados satisfatórios ao ser comparado com outros sete algoritmos clássicos de binarização de documentos.

Abstract

The check processing is one of the few banks processes that are not totally automated. As a bank application, this processing requires precision and safety. However, its automation is not an easy task, since banks introduce complexity to the background of the checks in order to reduce the possibility of falsification. Then, thresholding this type of image is not a trivial task. With that motivation, this research studied and implemented a thresholding algorithm proposed by Neves, Mello e Silva [14][15][16][17], which is focused on the check images processing. This algorithm is based on the Tsallis entropy for determining the threshold of images. It achieved satisfactory results when compared with seven other classic thresholding algorithms of documents.

Sumário

Resumo	i
Abstract	ii
Sumário	iii
Índice de Figuras	v
Índice de Tabelas	vii
Índice de Quadros	viii
Tabela de Símbolos e Siglas	ix
Capítulo 1 Introdução	10
1.1 Objetivos	11
1.2 Estrutura da Monografia	12
Capítulo 2 Binarização de Imagens de Cheques Brasileiros	13
2.1 Cheque Bancário Brasileiro	13
2.2 Base de Imagens de Cheques Brasileiros	17
2.3 Binarização de Imagens de Cheques	18
2.4 Algoritmos Clássicos de binarização	19
2.4.1 Binarização Niblack	23
2.4.2 Binarização Sauvola	23
2.4.3 Binarização White	24
2.4.4 Binarização Johannsen	26
2.4.5 Binarização Bernsen	27
2.4.6 Binarização Rosenfeld	28

2.4.7 Binarização Wellner	31
2.5 Considerações finais sobre o capítulo 2	32
Capítulo 3 Algoritmo Neves-Mello-Silva	35
Capítulo 4 Experimentos e Resultados	45
Capítulo 5 Conclusão	53
5.1 Contribuições	53
5.2 Trabalhos Futuros	54
Bibliografia	56

Índice de Figuras

Figura 1.	Exemplo de imagem de cheque cedida pela <i>AILeader Technologies</i> . .12
Figura 2.	Imagem de cheque com linhas de segurança e marca d'água15
Figura 3.	Padrão para a distribuição dos dados no cheque.....15
Figura 4.	Amostra de cheques de bancos Brasileiros.17
Figura 5.	Cheques do Branco do Brasil para tipos de contas diferentes.18
Figura 6.	Histograma de uma imagem em tons de cinza.....21
Figura 7.	Binarização de White. A imagem a) é a imagem original de um CA em tons de cinza, as imagens b), c) e d) é o resultado da binarização White com peso igual a 1.2, 1.5 e 2, respectivamente26
Figura 8.	Histograma de uma imagem em tons de cinza envolvido por sua envoltória convexa.29
Figura 9.	Imagens binarizadas por algoritmos clássicos33
Figura 10.	Imagens binarizadas por algoritmos clássicos34
Figura 11.	Descrição de Curtose39
Figura 12.	Descrição da assimetria.39
Figura 13.	a) Histograma base; b) imagem de um CA e seu histograma; c) imagem do CA após a especificação de seu histograma e seus histograma.....41
Figura 14.	Fluxograma para o algoritmo Neves-Mello-Silva. (Fonte: dissertação de mestrado de Renata Freire de Paiva Neves, título: “Um Novo Algoritmo para Binarização do <i>Courtesy Amount</i> de Imagens de Cheques Brasileiros”, Programa de Pós-Graduação em Engenharia da Computação da Universidade de Pernambuco).....43

Figura 15. Exemplos da aplicação do algoritmo: (esquerda) imagem original e (direita) imagem binarizada.	44
Figura 16. Gráfico da eficiência dos algoritmos.	47
Figura 17. Definição de uma imagem ideal.	48
Figura 18. Imagens encontradas pelo algoritmo de Neves, Mello e Silva.	51

Índice de Tabelas

Tabela 1.	Parâmetros para o algoritmo de binarização Neves-Mello-Silva	42
Tabela 2.	Sumarização dos tempos dos algoritmos.....	46
Tabela 3.	Medidas de <i>Precision</i> , <i>Recall</i> , <i>Accuracy</i> e <i>Specificity</i> de cada algoritmo.	50
Tabela 4.	Comparativo entre as classes do algoritmo Neves-Mello-Silva.....	52

Índice de Quadros

Quadro 1. Pseudocódigos para os cálculos do a) histograma, b) média, c) variância e d) desvio padrão de uma imagem em tons de cinza **Erro! Indicador não definido.**

Quadro 2. Pseudocódigos para método de Niblack23

Quadro 3. Pseudocódigos para método de Sauvola.....24

Quadro 4. Pseudocódigos para método de White.....25

Quadro 5. Pseudocódigos para método de Johannsen27

Quadro 6 Pseudocódigos para método de Bernsen.....28

Quadro 7. Pseudocódigos para método de Rosenfeld.....29

Quadro 8. Pseudocódigo para o método de Graham modificado para aplicação em histograma.....30

Quadro 9. Pseudocódigo para o método de Wellner (*Moving Averages*).....32

Tabela de Símbolos e Siglas

CA – *Courtesy Amount*

LA – *Legal Amount*

CMC7 - *Character Magnetic Code*

DPI – dots per inch

OCR - *Optical Character Recognition* (Reconhecimento óptico de caractere)

IDE - *Integrated Development Environment* (Ambiente Integrado de Desenvolvimento)

Capítulo 1

Introdução

Desde os primórdios da civilização, a humanidade busca formas mais cômodas e eficazes de fazer suas transações comerciais. Inicialmente essas transações eram feitas pela simples troca de mercadorias, mas elas evoluíram e agregaram avanços tecnológicos em seus processos, avanços esses que garantissem ganhos de facilidade e segurança.

O advento da tecnologia trouxe consigo a capacidade de automatizar o processo comercial. Especialmente os processos que envolvem transações bancárias. Atualmente, as formas mais comuns de se fazer transações bancárias, além da utilização de dinheiro em espécie, são através do uso de cheques e cartões. O processo de utilização de cartões já é considerado totalmente automatizado, contudo, o processamento de cheques ainda não alcançou tais níveis de automatização.

Para chegar ao objetivo da automatização de um cheque, que é reconhecer corretamente as informações nele contidas, o cheque precisa passar por um processo extenso, geralmente dividido em cinco etapas: Digitalização, pré-processamento, binarização, extração de características e reconhecimento. Cada uma das etapas depende do sucesso das etapas anteriores para garantir seu próprio sucesso.

Para alcançar um reconhecimento eficaz das informações é necessário que essas informações tenham sido extraídas da imagem do cheque. Para que essas informações sejam extraídas de forma correta e completa é necessário que elas estejam lá, ou seja, a imagem resultante do processo de binarização deve conter todos os objetos desejáveis. Nesse tipo aplicação, o texto é o que se espera recuperar da imagem após sua binarização.

Contudo, os bancos visando aumentar a segurança adicionam elementos extras ao *background* (fundo) do cheque, tornando sua imagem mais complexa, e

assim, dificultando a etapa de binarização. Esses elementos extras podem conter características similares às do texto e serem classificados como tal, o que não é desejável.

1.1 Objetivos

Este trabalho está focado justamente na etapa de binarização de imagens de cheques. Onde se busca separar o texto e o *background* desse tipo de imagem. Esta etapa é fundamental para o bom desempenho das outras etapas do processo. Principalmente porque ela é a etapa que mais sofre influência das mudanças adicionadas pelos bancos para aumentar a segurança do processo.

É feita a implementação de um algoritmo originalmente elaborado para binarização de imagens de cheques. Este algoritmo foi desenvolvido por Neves, Mello e Silva [14][15][16][17], e mostrou-se muito eficaz na busca dos seus objetivos. Nesse algoritmo, a binarização é aplicada às imagens em tons de cinza do *Courtesy Amount* das imagens dos cheques. O *Courtesy Amount* corresponde ao valor numérico do cheque. Ele foi identificado como um campo ideal para esta aplicação.

Além disso, vários algoritmos clássicos foram implementados e utilizados para validar o real desempenho do algoritmo em estudo. Esses algoritmos foram aplicados a uma base de imagens de cheque cedida pela empresa *AILeader Technologies* [1]. Essas imagens são amostras de cheques de vários bancos no Brasil. Na Figura 1 é apresentado um exemplar das imagens de cheques da base. É importante enfatizar que as informações do cliente foram preservadas, e estão borradas na imagem.

Assim, os principais objetivos dessa monografia são:

1. Implementar algoritmos de binarização;
2. Testar esses algoritmos em imagens de cheques bancários brasileiros;
3. Testar o novo algoritmo de binarização de cheques, confrontando-o com os algoritmos implementados.

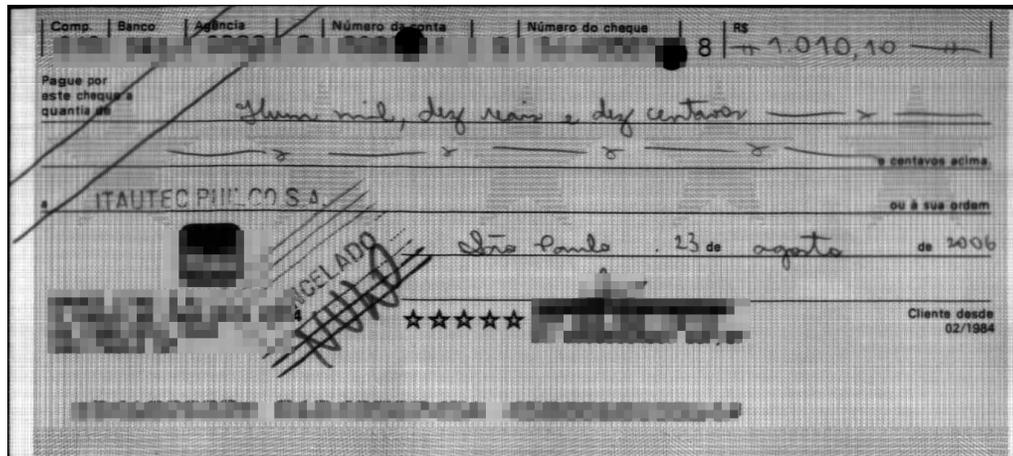


Figura 1. Exemplo de imagem de cheque cedida pela *AiLeader Technologies*.

1.2 Estrutura da Monografia

O Capítulo 2 apresenta uma visão geral sobre processo binarização de imagens de cheque. Esse capítulo relata sobre as características dos cheques brasileiros, e como essas características influenciam na binarização das imagens dos cheques. Além disso, no Capítulo 2 são descritos vários algoritmos clássicos de binarização de imagens. Esses algoritmos serão utilizados para validar os resultados deste trabalho.

O Capítulo 3 contém o foco deste trabalho. Nele é descrito o algoritmo Neves-Mello-Silva para binarização de imagens de cheques.

No Capítulo 4 são analisados os resultados do algoritmo descrito no Capítulo 3 e é apresentado um comparativo entre este algoritmo e os outros algoritmos de binarização implementados.

Por fim, o Capítulo 5 apresenta a conclusão e contribuições deste trabalho e o que é proposto para pesquisas futuras.

Capítulo 2

Binarização de Imagens de Cheques Brasileiros

Este capítulo está dividido em quatro seções. Na seção 2.1 é feito um estudo sobre os padrões e características dos cheques brasileiros e como essas características interferem no processamento automático de cheques bancários. Na seção 2.2 é descrita a base de imagens utilizada nesta pesquisa. Na seção 2.3 é feito um estudo sobre a binarização de imagens de cheque e os benefícios desta aplicação. Logo em seguida, na seção 2.4 vários algoritmos clássicos de binarização são descritos.

2.1 Cheque Bancário Brasileiro

Pesquisas revelam que a utilização de cheques está em constante crescimento no Brasil [21]. Em 2008 o montante em transações utilizando cheque foi de 2,48 trilhões, um aumento de 13,14% em relação ao ano de 2007 [10]. Em contrapartida, o montante em transações com cartões de crédito foi de 212 bilhões, representando menos de 10% do total em transações com cheques bancários. Parte deste aumento foi devido ao aumento na utilização de cheques pré-datados. Entre novembro de 2007 e novembro de 2008, houve um aumento de 8,56% na utilização deste tipo de transação.

Pela primeira vez, o uso de cartões de débito ultrapassou o uso de cheque [10]. Segundo o Banco Central, em 2008, ocorreram 2,1 bilhões de transações com cartões de débito e 1,94 bilhão de transações com cheques. Apesar dessa ultrapassagem, a diferença no número de transações é proporcionalmente pequena. No entanto, o valor agregado por transações com cheque é cerca de dez vezes maior. O valor médio de pagamentos com cheque foi de R\$ 835,00 e o valor médio com cartões eletrônicos foi de R\$49,00 para cartões de débito e de R\$ 86 para cartões de crédito.

Em 2009, observa-se a manutenção do aumento na utilização de cheques pré-datados [7]. Entre o mês de março e abril houve um aumento de 0,44% [28]. Além disso, Pernambuco foi considerado o segundo estado no Brasil com maiores índices de cheques pré-datados, com 90,21% do total de transações utilizando cheques, ficando atrás apenas do estado do Rio Grande do Norte, com um índice de 90,27%.

Um dos principais motivos da utilização comercial dos cheques bancários no Brasil é a segurança envolvendo transações de altos valores, visto que elas não são comumente realizadas por meio de cheques eletrônicos. No entanto, o processamento de cheque é um dos poucos processos bancários que ainda não é totalmente automatizado, grande parte dele é feito manualmente, tornando-o um processo lento e com alto custo associado.

Com o avanço da tecnologia os bancos estão investindo cada vez mais na automatização de seus processos. Contudo, automatizar o processamento de cheques bancário não é uma tarefa simples. Pois, os bancos, visando dificultar a falsificação, adicionam elementos de complexidade ao *background* (fundo) do cheque. Esses elementos geralmente são marcas d'água, linhas de segurança, modificação no padrão de cores ou modificação no padrão de contrastes na imagem.

A Figura 2 mostra uma imagem de cheque que contém tanto marca d'água quanto linhas de segurança. Nela, o nome do banco, SUDAMERIS, posicionado no centro da imagem, está escrito como uma marca d'água. A marca d'água pode estar localizada em qualquer posição do cheque, já as linhas de segurança são dispostas no sentido vertical no lado direito do cheque. O Banco Central [2] estabelece que as linhas aleatórias devam cobrir os campos de valor numérico, nome do favorecido, local e data de emissão e assinatura. Essas linhas são impressas segundo ângulo ou sinuosidades aleatórias, sendo encontrado o mesmo padrão das linhas a cada mil cheques impressos.

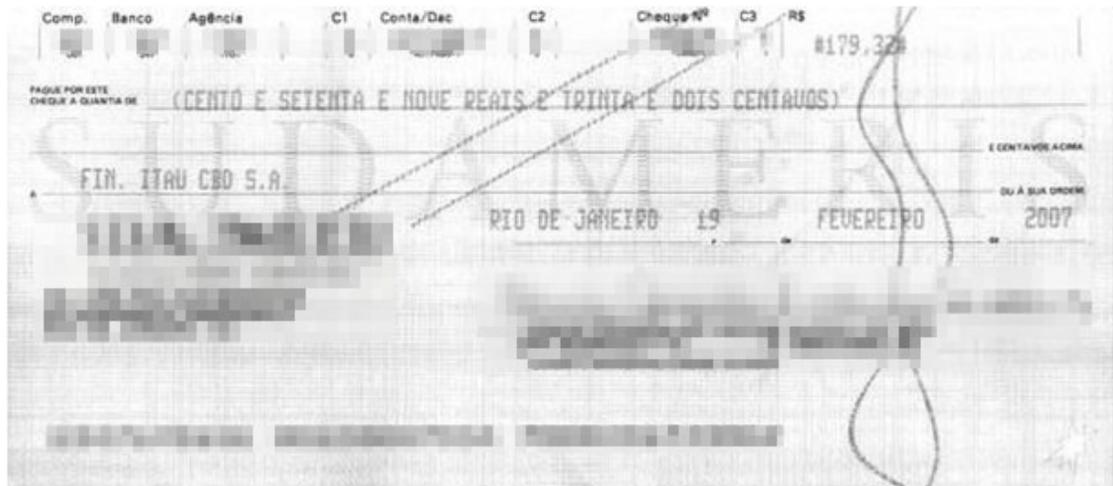


Figura 2. Imagem de cheque com linhas de segurança e marca d'água

No Brasil, o padrão dos cheques foi estabelecido por normas do Banco Central [2]. Suas características foram projetadas para garantir maior agilidade e segurança nas transações. O Conselho Monetário Nacional define um único modelo de cheque sendo composto por quatro áreas distintas, como mostrado na Figura 3. Uma pequena variação na sua construção é permitida. Essa variação pode ser no tamanho e cor da fonte utilizada.

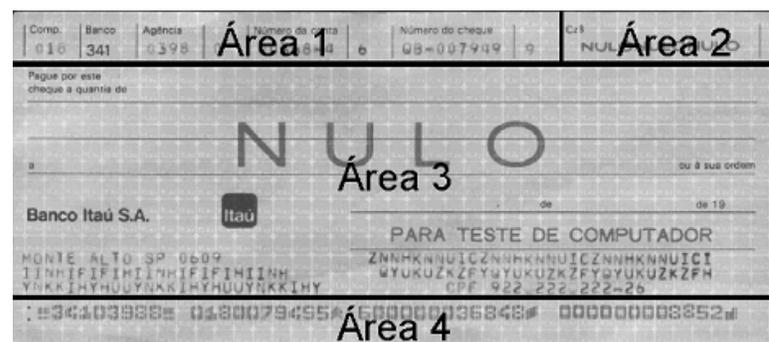


Figura 3. Padrão para a distribuição dos dados no cheque.

1. A área 1 possui informações do banco e da conta do cliente (número da compensação, número do banco, número da agência, número da conta, número do cheque e dígitos verificadores);
2. A área 2 corresponde ao valor numérico do cheque (CA – *Courtesy Amount*);

3. A área 3 contém o valor do cheque por extenso, o nome do favorecido, logotipo do banco, o nome do banco, assinatura, nome do correntista, número do CPF/CGC, data e local de emissão e outras informações do banco e da conta (LA – *Legal Amount*);
4. A área 4 contém um código magnético para validação do cheque codificado em sete barras, (CMC7 - *Character Magnetic Code*).

Os objetos presentes em um cheque podem ser divididos em quatro categorias: *background*, fixas, variáveis e inseridas [9]. Quando se fala em *background*, estamos fazendo referência ao fundo artístico, ele geralmente é composto por pinturas e desenhos coloridos, e varia de cheque para cheque. As informações fixas são aos dados que estão presentes em todos os cheques, como por exemplo, as linhas horizontais e verticais que delimitam os campos do cheque. Informações variáveis são aquelas que são diferentes para cada cliente, mas que são impressas no cheque pelo próprio banco, por exemplo, o número da agência e o número da conta. Por fim, as informações inseridas são aquelas introduzidas pelo próprio cliente, tais como o valor numérico do cheque (*Courtesy Amount*), o valor do cheque por extenso (*Legal Amount*), a data, a cidade e a assinatura.

Esta monografia está focada na binarização do *Courtesy Amount* das imagens dos cheques. O CA é identificado como um campo ideal, tanto nas etapas de binarização e reconhecimento quanto para a eficiência da automatização do processamento da imagem dos cheques.

Primeiramente, porque o reconhecimento automático de algarismos é mais fácil do que o reconhecimento de letras alfabéticas. Pois, além de existir uma quantidade menor de símbolos (intervalo de 0-9), fazendo com que haja uma diminuição na quantidade de possibilidades para o reconhecimento, os algarismos apresentam menor variação na forma da escrita enquanto que as letras alfabéticas podem ser escritas como letra de forma ou como letra cursiva.

Outra vantagem advinda da escolha da região do *Courtesy Amount* é que essa pequena região da imagem contém todas as informações essenciais para se realizar uma boa binarização. Essas informações são o padrão do *background*, que

é o mesmo para o restante do cheque, e a tinta utilizada no seu preenchimento. Essa tinta, na maioria dos casos, também é a mesma utilizada para o preenchimento do valor do cheque por extenso, o *Legal Amount*. Sendo assim, se houver a necessidade de uma posterior binarização da imagem completa, todas as informações necessárias já foram de alguma forma processadas na etapa de binarização do CA. O tom encontrado pode ser ideal ou perto do ideal para a binarização da imagem completa.

2.2 Base de Imagens de Cheques Brasileiros

As imagens utilizadas nesta pesquisa foram cedidas pela empresa *AILeader Technologies* [1]. Foram no total 1.584 imagens de cheques de 34 bancos brasileiros distintos, com diversas variações de cheques de cada banco. As imagens foram digitalizadas a 200 dpi de resolução, e estão em tons de cinza. Veja na Figura 4 imagens de cheques de bancos diferentes. Observe como os *backgrounds* dos cheques variam de banco para banco.



Figura 4. Amostra de cheques de bancos Brasileiros.

Veja na Figura 5 imagens de contas diferentes, porém de um mesmo banco, Banco do Brasil. Pela simples observação fica evidente que cheques de mesmo banco não necessariamente possuem o mesmo padrão. Veja que os padrões do *background* da área 1 e da área 2 variam de um cheque para outros.

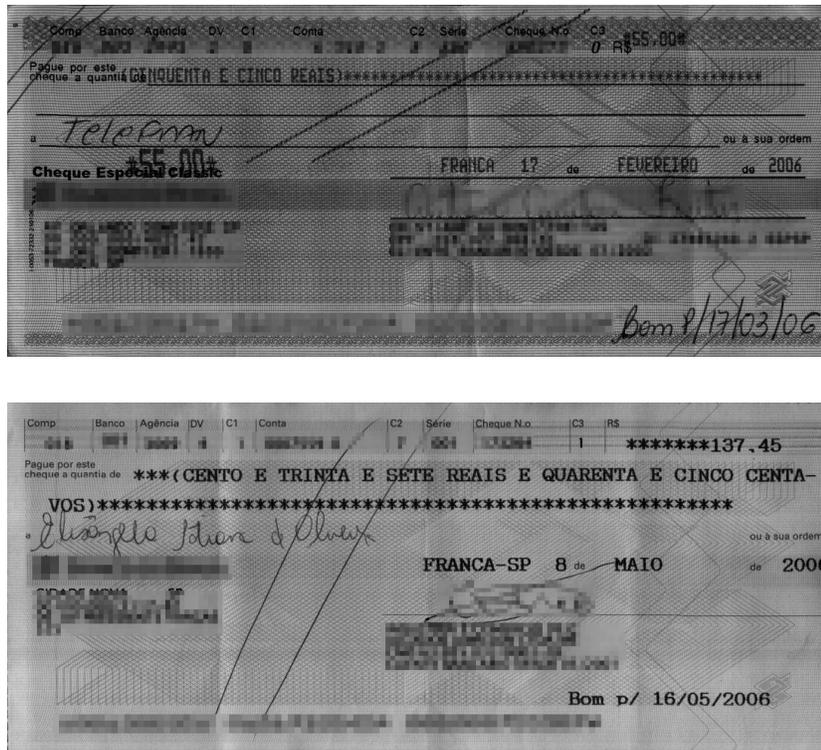


Figura 5. Cheques do Banco do Brasil para tipos de contas diferentes.

2.3 Binarização de Imagens de Cheques

Os cheques bancários, depois de digitalizados, são muitas vezes representados em duas cores, preto e branco. A binarização é a técnica utilizada para fazer a transformação no conjunto de cores da imagem. Essa representação binária traz economia de recurso em relação à representação colorida ou em tons de cinza, pois é necessário apenas 1 bit para representar cada cor, e conseqüentemente cada *pixel* [5].

A primeira etapa da binarização é, para uma imagem em tons de cinza, encontrar um valor de nível de cinza t considerado o limiar, ou ponto de corte, da imagem. Todos os *pixels* que possui tons de cinza menor que t são considerados texto e convertidos para preto e o restante, considerados como pertencentes ao fundo da imagem, são convertidos para branco.

As técnicas de binarização podem ser divididas em “globais” e “locais”. Uma técnica global encontra um limiar único para toda a imagem. Costuma apresentar bons resultados quando o nível de cinza do texto é bastante distinto do nível de cinza do fundo. Uma técnica de binarização local encontra limiares distintos de acordo com as características da vizinhança de um *pixel*. As técnicas locais apresentam melhores resultados quando existe pouca variação do nível de cinza do texto e do fundo. Em geral as técnicas globais são mais eficientes que as locais porque exigem menos processamento, visto que uma vez definido um limiar, o único processamento restante é a separação dos objetos da imagem.

A escolha do limiar num processo de binarização é uma tarefa crítica. Um limiar inadequado pode deixar a imagem ilegível e prejudicar todas as outras etapas do processamento dos cheques bancário. Para garantir o sucesso da etapa de binarização, muitas vezes são utilizadas técnicas de pré-processamento da imagem em tons de cinza e pós-processamento da imagem binarizada.

A binarização também é utilizada para auxiliar o processo de trucagem de cheques. Na trucagem, a troca física de cheques em papel entre bancos para a compensação é substituída pela troca de imagens digitalizadas. A utilização de imagens digitalizadas no processo e trucagem eleva a eficiência operacional dessa atividade. Essa aplicação trás economia ao sistema financeiro com a redução do custo no transporte dos cheques em papel para as câmaras de compensação. Além do que, a imagem binarizada ocupa menos espaço que uma imagem que foi apenas digitalizada. Isso reduz custos em tecnologia de armazenamento. Contudo, a representação eletrônica de cheques para a trucagem traz alguns desafios adicionais. Como um banco poderá digitalizar os cheques de vários bancos diferentes, é necessário encontrar técnicas de binarização de imagens que sejam eficientes para uma grande variação no padrão de imagens.

2.4 Algoritmos Clássicos de binarização

Vários algoritmos de binarização de imagens de documentos podem ser encontrados na literatura. Nesta seção estão descritos alguns desses algoritmos: Niblack [18], Sauvola [24], White [30], Johannsen [18], Bernsen [3], Rosenfeld [23] e

Wellner [29]. Todos eles foram implementados como parte desta pesquisa, e são utilizados em comparação com o algoritmo de binarização Neves-Mello-Silva [14][15][16][17] desenvolvido para imagens de cheques bancários, que também foi implementado neste trabalho, como descrito na seção 3.

Funções estatísticas são comumente utilizadas nos métodos de processamento de imagem existentes. Algumas dessas funções são utilizadas como métodos auxiliares em vários dos algoritmos de binarização descritos neste trabalho. Sendo assim, com o intuito de evitar repetições excessivas de informações, as funções mais utilizadas estão escritas em pseudocódigo no Quadro 1, são elas as funções de histograma, média, variância e desvio padrão.

Em processamento de imagens, o histograma é uma representação gráfica, em duas dimensões, da distribuição cumulativa das cores presentes numa imagem. Existem vários tipos de histogramas, para imagens em tons de cinza o histograma é uma função entre os níveis de cinza e a frequência com que eles aparecem na imagem. Nesse gráfico, a origem cartesiana representa a tonalidade de menor luminosidade, e à medida que se afasta da origem, pelo eixo das abscissas, no sentido da direita, vai adquirindo mais luminosidade até chegar à tonalidade mais clara representável. Veja um exemplo de histograma na Figura 6. Para uma escala de tons de cinza, a origem, com valor 0 (zero) corresponde à cor preta, e a tonalidade mais à direita corresponde à cor branca, com valor de 255 (duzentos e cinquenta e cinco) associado. No pseudocódigo descrito no Quadro 1 a), o histograma é representado pelo vetor $histo(i,j)$, que possui 1 linha e 256 colunas, e armazena as probabilidades dos 256 possíveis níveis de cinza presentes na imagem.

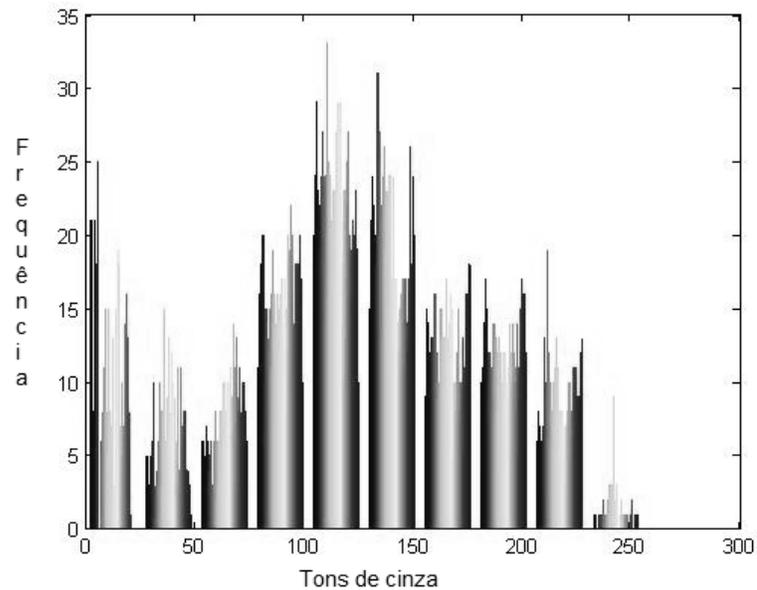


Figura 6. Histograma de uma imagem em tons de cinza

A função média descrita em pseudocódigo no Quadro 1 b), calcula o valor médio dos níveis de cinza presentes na imagem. A variância, descrita no Quadro 1 c), é uma medida de dispersão estatística, que indica quão longe, em geral, os valores dos níveis de cinza da imagem se encontram do valor médio. O desvio padrão também é uma medida de dispersão. Ele é calculado como a raiz quadrada da variância, veja seu pseudocódigo no Quadro 1 d) , ele pode ser entendido como a média dos valores absolutos dos desvios em relação à média que, neste caso, é o nível médio de cinza da imagem, sendo expresso em tons de cinza.

Quadro 1. Pseudocódigos para os cálculos do a) histograma, b) média, c) variância e d) desvio padrão de uma imagem em tons de cinza

<pre>a) HISTOGRAMA (imagem) INICIO Var lin, col, i, j histo <- vetor (1, 256) * ZERO [lin, col] <- TAMANHO (imagem) Para i <- 1 até lin faça Para j <- 1 até col faça histo (imagem (i, j)) <- histo (i, j) + 1 fim fim Retorne (histo) FIM</pre>
<pre>b) MEDIA (imagem) INICIO var i, j, media <- 0 [lin, col] <- TAMANHO (imagem) para i <- 1 até lin faça para j <- 1 até col faça media <- media + imagem (i, j) fim media <- media / (lin * col) fim Retorne (media) FIM</pre>
<pre>c) VARIANCIA (histo, media) INICIO Var i, j, variancia <- 0 Para i <- 1 até 256 faça variancia <- variancia + (histo (1, i) - media)^2 variancia <- variancia / 256 Retorne (variancia) FIM</pre>
<pre>d) DESVIO_PADRAO (histo, media) INICIO Var variancia, desvioPadrao variancia <- VARIANCIA (histo, media) desvioPadrao <- (variancia)^(1/2) Retorne (desvioPadrao) FIM</pre>

2.4.1 Binarização Niblack

O algoritmo proposto por Niblack [18], aparece em muitas pesquisas recentes como um excelente algoritmo de binarização. Nesse algoritmo o limiar varia por toda imagem, de acordo com a média local e o desvio padrão calculados na vizinhança de cada *pixel*, numa janela de tamanho $b \times b$. Além de definir o tamanho da janela, o usuário também pode definir o peso que o desvio padrão tem sobre o limiar. O valor do peso pode variar dependendo do tipo de imagem ao qual o algoritmo for aplicado. Neste trabalho, para a base de imagem utilizada, o tamanho da janela b igual a 100 e o valor do parâmetro peso igual a -0,3 foram achados satisfatórios. Esses resultados foram definidos, experimentalmente, através da comparação entre as imagens ideais definidas para cada uma das imagens e as imagens resultantes geradas, à medida que os parâmetros do algoritmo sofriam variações. A combinação entre janela e peso que gerou os melhores resultados foi utilizada como parâmetro para o algoritmo. O pseudocódigo para o algoritmo de Niblack está descrito no Quadro 2.

Quadro 2. Pseudocódigos para método de Niblack.

```
NIBLACK(janela, peso)

INICIO
  Histo <- HISTOGRAMA (janela)
  media <- MEDIA(janela)
  desvioPadrao <- DESVIO_PADRAO (histo, media)
  limiar <- media + peso* desvioPadrao

  Para i <- 1 até janela
    Para j <- 1 até janela
      Se janela( i, j) > limiar então
        janela( i, j) <- BRANCO
      Senão
        janela( i, j) <- PRETO
    fim
  fim

FIM
```

2.4.2 Binarização Sauvola

Em [24], Sauvola descreve uma melhoria no algoritmo de Niblack para imagens que não apresentam boa iluminação. Ele introduz uma nova variável R que modifica o

efeito introduzido pelo desvio padrão. Em seus experimentos, Sauvola utilizou R igual a 128, chegando assim em seus melhores resultados. O algoritmo de Sauvola busca melhorar os resultados da binarização de imagens que foram adquiridas em más condições de iluminação, e que apresentam altas taxas de ruídos. O pseudocódigo para o seu algoritmo está descrito no Quadro 3.

Quadro 3. Pseudocódigos para método de Sauvola.

```
SAUVOLA(janela, peso)

INICIO
  histo<-HISTOGRAMA (janela)
  media <- MEDIA(janela)
  desvioPadrao <- DESVIO_PADRAO (histo, media)
  R<-128

  limiar = media + 1 + peso* ((desvioPadrao/R) - 1)

  Para i <- 1 até janela
    Para j <- 1 até janela
      Se janela( i, j) > limiar então
        janela( i, j) <- BRANCO
      Senão
        janela( i, j) <- PRETO
    fim
  fim
fim
FIM
```

2.4.3 Binarização White

O algoritmo de White [30] compara o tom de cinza de cada *pixel* da imagem com a média dos valores dos seus vizinhos numa janela que seja aproximadamente o tamanho de um caractere. Para peso igual a 2, como sugerido em [25], se o *pixel* for significativamente mais escuro que a média, ele é classificado como caractere, caso contrário, ele é classificado como papel. No entanto, para imagens de cheque esse valor é relativamente alto. Pois, como os elementos textuais da imagem do cheque podem apresentar valores com tonalidades muito próximas ao do fundo, ao atribuir um valor muito alto ao peso, maior será a probabilidade do texto ser classificado como papel, pois o peso maior do que 1 (um) provoca o efeito de clareamento no *pixel* que está sendo analisado.

Veja na Figura 7 o efeito provocado pela variação do peso. Na imagem b) o peso é igual a 1.2, e apesar do aparecimento de ruídos, a informação de texto foi satisfatoriamente preservada, visto que o texto permaneceu quase todo na imagem. Já nas figuras c) e d), nota-se uma degradação significativa da informação textual. Em c) apesar dos traços dos caracteres estarem desconectados, ainda é possível a identificação dos algarismos ali presentes, porém, na figura d) não restou informação suficiente para que os algarismos sejam identificados. No Quadro 4, é apresentado o pseudocódigo para o método de White.

Quadro 4. Pseudocódigos para método de White.

```
WHITE(janela)

INICIO
  peso <- 1.2
  media <- MEDIA(janela)
  Para i <- 1 até janela
    Para j <- 1 até janela
      Se media < janela(i,j)*peso então
        janela( i, j) <- BRANCO
      Senão
        janela( i, j) <- PRETO
    fim
  fim
fim

FIM
```

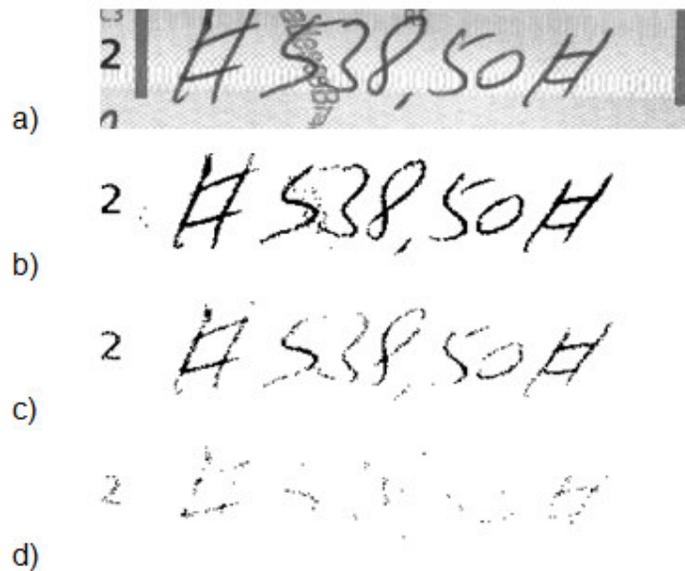


Figura 7. Binarização de White. A imagem a) é a imagem original de um CA em tons de cinza, as imagens b), c) e d) é o resultado da binarização White com peso igual a 1.2, 1.5 e 2, respectivamente

2.4.4 Binarização Johannsen

O método de Johannsen [18] calcula a binarização baseado na entropia da imagem. O seu objetivo é dividir uma imagem em duas partes minimizando as interdependências entre os seus níveis de cinza. Para que isso aconteça, é calculada a entropias para preto $S_b(t)$ e para branco $S_w(t)$. O pseudocódigo para o método de Johannsen está descrito no Quadro 5.

Quadro 5. Pseudocódigos para método de Johannsen.

```

JOHANNSEN(imagem)

INICIO
[lin col] <- tamanho (imagem)

hist <- HISTOGRAMA(image)
hist <- hist/(lin*col) // histograma normalizado

pt(1) <- hist(1)
para i<-2 até 256 // probabilidade cumulativa
    pt(i) <- pt(i-1) + hist(i)
fim

t<-1
para j <- 2 até 256
    se(hist(j) > 0) então
        sb<- logaritmo[pt(j)]
        sb<-sb+[1/pt(j)]*{entropia[hist(j)]+entropia[pt(j-1)]}

        sw<- logaritmo(1-pt(j))
        sw<-sw+{1/[1-pt(j)]}*{entropia[hist(j)]+entropia[1-pt(j+1)]}

        F(j) <- sb + sw
        se (F(j) < F(limiar)) então
            limiar <- j
        fim
    fim
fim

Para i <- 1 até janela
    Para j <- 1 até janela
        Se limiar < image(i,j) então
            janela( i, j) <- BRANCO
        Senão
            janela( i, j) <- PRETO
        fim
    fim
fim

FIM

```

2.4.5 Binarização Bernsen

No método local de Bernsen [3] o limiar é escolhido como o um valor intermediário, que é a média entre o mínimo e o máximo tom de cinza da vizinhança do *pixel*. Se o contraste for menor que certo limiar dado, então o *pixel* é considerado de uma só classe com seus vizinhos, texto ou *background*, dependendo do valor do limiar. O

contraste é a diferença entre o máximo e o mínimo nível de cinza dado. Este método está descrito no Quadro 6.

Quadro 6. Pseudocódigos para método de Bernsen.

```
BERNSEN(janela, peso, contraste)
INICIO
  Min <- MINIMO(janela)
  max <- MAXIMO(janela)
  cont <- max - min
  media <- (max - min)/2

  Se(cont > contraste) então
    limiar <- media/2

  Senão se (media < 100) então
    limiar <- max
  Senão
    limiar <- min

  Para i <- 1 até janela
    Para j <- 1 até janela
      Se janela( i, j) > limiar então
        janela( i, j) <- BRANCO
      Senão
        janela( i, j) <- PRETO
    fim
  fim
fim
FIM
```

2.4.6 Binarização Rosenfeld

O método de binarização desenvolvido por Rosenfeld [23] é baseado na análise da envoltória convexa do histograma de uma imagem. Veja na Figura 8, a envoltória convexa de um histograma é o menor polígono convexo para o qual todos os níveis do histograma estejam no seu interior. No algoritmo de Rosenfeld, o ponto mais profundo entre a diferença da envoltória convexa de um histograma e o próprio histograma da imagem se torna candidato ao limiar. Se houver mais de um candidato, outras características da imagem podem ser definidas como padrões de desempate. O algoritmo para o método de Rosenfeld está descrito no Quadro 7. Para o algoritmo implementado foi utilizada uma técnica chamada de Varredura de Graham [4]. Essa varredura é um método para o cálculo da envoltória convexa de

um conjunto de pontos. Como pode ser observado no Quadro 8, este método foi adaptado para sua utilização em histogramas, onde só a parte superior do histograma é considerada para o cálculo, uma vez que sua parte inferior é fixa e plana para qualquer histograma.

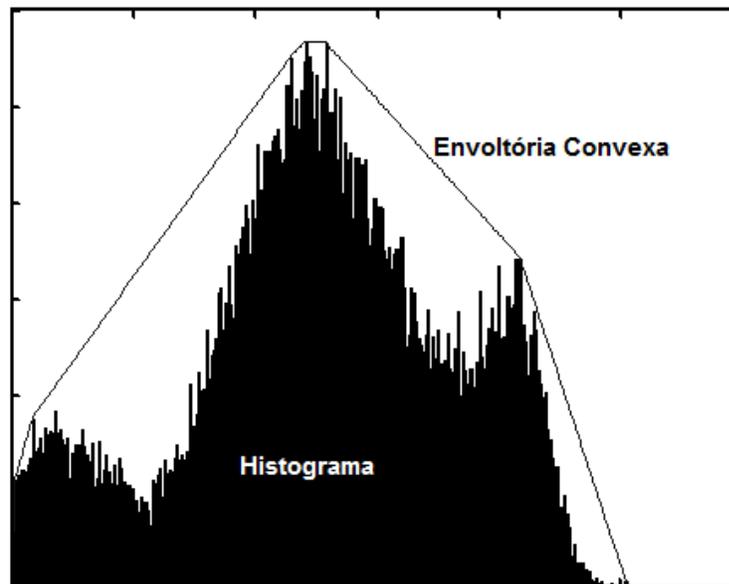


Figura 8. Histograma de uma imagem em tons de cinza envolvido por sua envoltória convexa.

Quadro 7. Pseudocódigos para método de Rosenfeld.

```
ROSENFELD (janela, peso, contraste)
INICIO
  histograma <- HISTOGRAMA(janela)
  envoltória <- Graham (histograma)

  candidatos <- envoltória - histograma
  limiar <- MINIMO(candidatos)

  Para i <- 1 até janela
    Para j <- 1 até janela
      Se janela( i, j) > limiar então
        janela( i, j) <- BRANCO
      Senão
        janela( i, j) <- PRETO
    fim
  fim
fim
FIM
```

Quadro 8. Pseudocódigo para o método de Graham modificado para aplicação em histograma.

```

GRAHAM (histograma)

INICIO

  a1 <- inicio válido do histograma
  an <- final válido do histograma
  pilha(1) <- a1;
  pilha(2) <- a1 + 1;
  topo <- 2;

  para i <- (a1 + 2) até an
    i1 <- pilha(topo - 1);
    i2 <- pilha(topo);
    i3 <- i;
    guinada <- sentido(i1, i2, i3)

    enquanto(sentido >= 0 && topo > 2) faça
      topo <- topo - 1;
      i1 <- pilha(topo -1);
      i2 <- pilha(topo);
      i3 <- i;
      guinada <- sentido(i1, i2, i3)

    fim
  se(guinada >= 0 && topo == 2)então
    pilha(2) <- i;
  senão
    topo <- topo + 1;
    pilha(topo) <- i;
  fim
fim

FIM

```

Na adaptação do método de Graham para a determinação da envoltória convexa em histogramas, o cálculo começa do primeiro nível com amplitude diferente de zero, $a1$, e termina no último nível com amplitude diferente de zero, an . Para saber se a guinada entre três níveis está no sentido horário ou anti-horário, é utilizado um método $\text{sentido}(i1, i2, i3)$. Se $\text{sentido}(i1, i2, i3)$ for positivo tem-se uma guinada no sentido anti-horário, se for negativo, a guinada é no sentido horário, se for nulo, os pontos são colineares. Neste último caso, o algoritmo pode eliminar o ponto intermediário, que é redundante.

2.4.7 Binarização Wellner

O método descrito por Wellner, em [29], para binarização de imagens, produz um limiar por pixel muito rápido. O *Moving Average* (Média móvel), como é chamado pelo autor, é uma média dos níveis de cinza dos últimos n pixel que foram vistos. A imagem pode ser tratada como um fluxo unidimensional de pixel. O algoritmo para o método de Wellner está descrito no Quadro 9.

Para evitar desvio de um lado da imagem para outro, um método chamado *Boustrophedon* ou *as the ox plows* (como o boi lavra) foi empregado. Isso significa atravessar a imagem em direções opostas em cada linha. O *pixel* seguido do último *pixel* de uma linha é o último *pixel* da próxima linha, seguido pelo penúltimo e assim chegando até o começo de linha, que é seguido pelo *pixel* do começo da próxima linha, até o final. Isso evita a descontinuidade no final das linhas.

A Soma é uma estimativa da média do *pixel* i , tendo nível de cinza imagem (i,j) e a média do nível imediatamente anterior. O valor do limiar é adquirido a partir de uma porcentagem fixa da média. Uma provável armadilha é o valor de porcentagem fixado usado para selecionar o limiar para a média. É improvável que um único valor seja apropriado para usar com a variação de tipo de imagem. Parker [19] detectou que para imagens que contêm texto uma porcentagem em torno de 85% seria suficiente.

Quadro 9. Pseudocódigo para o método de Wellner (*Moving Averages*)

```
WELLNER (imagem)
INICIO
  [lin,col] <- tamanho (imagem)
  N <- lin*col
  NC <- col
  S<-NC/8
  Soma <- 127*S

  Enquanto (existir linha na imagem)

    Para i <- 1 até lin

      Se (numero da linha for par) então
        Para j <- 1 até col
          Soma <- soma - (soma/S) + imagem[i,j]
          Media <- soma/S
          Porcentagem <- media(1-pct)
        fim
        Se imagem[i,j] > porcentagem então
          imagem[i,j]<- BRANCO
        Senão
          imagem[i,j]<- PRETO
        fim
      fim

      Se (numero da linha for impar) então
        Para j <- col até 1
          Soma <- soma - (soma/S) + imagem[i,j]
          Media <- soma/S
          Porcentagem <- media(1-pct)
        fim
        Se imagem[i,j] > porcentagem então
          imagem[i,j]<- BRANCO
        Senão
          imagem[i,j]<- PRETO
        fim
      fim

    fim

  Retorna(imagem)
FIM
```

2.5 Considerações finais sobre o capítulo 2

Por serem todos algoritmos clássicos de binarização, eles são utilizados como referência para a comparação com o método de binarização Neves-Mello-Silva

identificar, nem por inspeção visual, alguns deles. Em contrapartida, as imagens geradas pelos algoritmos de Wellner e Bernsen deixaram muita informação do papel, características que dificultam o reconhecimento dos caracteres, uma vez que tem muita informação de fundo considerada texto.

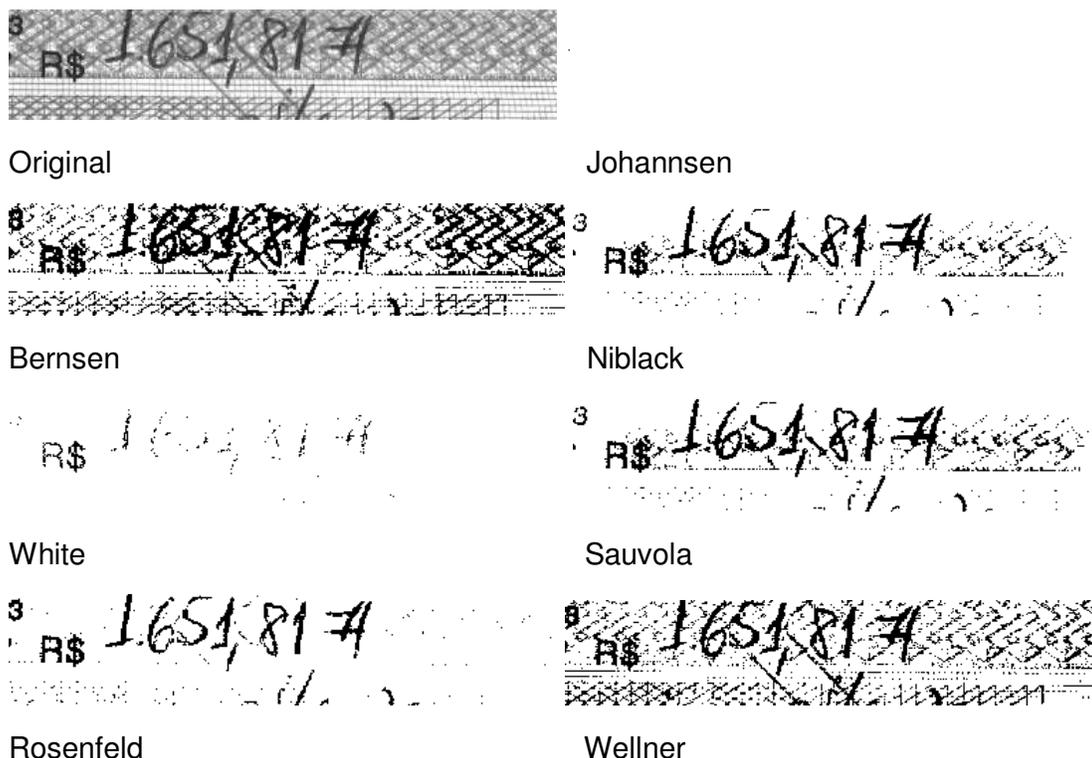


Figura 10. Imagens binarizadas por algoritmos clássicos

Apesar do algoritmo de Rosenfeld não ter gerado uma imagem ideal, as informações contidas nela têm uma probabilidade maior de serem reconhecida num processo de OCR (*Optical Character Recognition* - Reconhecimento óptico de caractere) [18], pois o fundo do cheque foi o praticamente todo removido, e o texto foi praticamente todo preservado. O processo de OCR é utilizado para fazer o reconhecimento de caracteres, ele é dividido em cinco principais etapas: digitalização, pré-processamento, binarização, extração de características e reconhecimento.

Capítulo 3

Algoritmo Neves-Mello-Silva

Após o estudo detalhado de vários algoritmos de binarização e a aplicação destes às imagens de cheques, Neves, Mello e Silva identificaram que o algoritmo proposto por Mello em [13], para binarização de imagens de documentos históricos, foi o que obteve melhores resultados, como pode ser visto nos resultados no próximo capítulo. Então, eles propuseram um novo algoritmo de binarização [14][15][16][17] baseado nas idéias de Mello, mas que se adequasse às características das imagens de cheque. Esse algoritmo foi projetado para realizar a binarização no *Courtesy Amount* do cheques bancários. Ele foi aplicado e validado em uma base dados com mais de 1,500 imagens de cheques brasileiros.

O algoritmo de Mello, definido originalmente em [12] e melhorado em [13], utiliza a idéia desenvolvida por Pun [22] para dividir a entropia da imagem completa em duas entropias, uma para o *background* (H_w) e uma para o *foreground* (tinta) (H_b). O limiar que divide essas entropias também é utilizado para o cálculo do limiar que será usado para a binarização da imagem. Mello utiliza a entropia de Tsallis [27] para calcular o limiar da imagem.

De acordo com Tsallis, uma definição universal de entropia é dada por:

$$H_{\alpha}(t) = \frac{1 - \sum_i p(s_i)^{\alpha}}{\alpha - 1}$$

Onde $p(i)$, assim como na entropia clássica de Shannon [26], é a probabilidade do *pixel* i aparecer na imagem e α é um parâmetro real cujo valor não é definido por Tsallis.

Segundo Shannon, se um sistema pode ser decomposto em dois sistemas estatisticamente independentes, A e B, então a entropia, H, possui a propriedade da adição [26]. Isso significa que $H(A + B) = H(A) + H(B)$.

Baseado na idéia de Pun, o algoritmo de Mello utiliza a cor mais freqüente para separar a entropia de Tsallis:

$$H_{\alpha}(S) = H_{\alpha}^b(t) + H_{\alpha}^w(t)$$

onde:

$$H_{\alpha}^b(t) = \frac{1 - \sum_{i=0}^t (p(s_i) / P_b(t))^{\alpha}}{\alpha - 1} \quad \text{e} \quad H_{\alpha}^w(t) = \frac{1 - \sum_{i=t+1}^{255} (p(s_i) / P_w(t))^{\alpha}}{\alpha - 1} ;$$

$$P_b(t) = \sum_{i=0}^t p(s_i) \quad \text{e} \quad P_w(t) = \sum_{i=t+1}^{255} p(s_i) ;$$

Onde, t é a cor mais freqüente na imagem. A probabilidade de S_i aparecer na imagem é representado por $p(s_i)$ e α é um valor real constante e diferente de 1.

O valor do limiar segundo o algoritmo de Mello é dado por:

$$th = m_b H_{\alpha}^b(t) + m_w H_{\alpha}^w(t)$$

Nele, m_b e m_w são constantes multiplicativas definidas experimentalmente.

Mello utiliza a entropia para identificar a quantidade de informação presente na imagem, assim, classificando-as em imagens com muito texto, pouco texto, e imagens que apresentam uma quantidade intermediária de texto. Para cada uma dessas classes são definidos valores adequados para os parâmetros do algoritmo.

Após Neves, Mello e Silva terem feito um estudo detalhado sobre o algoritmo de Mello, eles concluíram que os valores definidos para m_b , m_w e α , e a forma de classificar as imagens não eram ideais para a binarização de imagens de cheque. Então, modificaram a forma com que o algoritmo de Mello classificava as imagens para obter os valores adequados para m_b , m_w e α para as imagens desse tipo de aplicação.

A primeira etapa do algoritmo de Neves, Mello e Silva é definir um ponto de corte fixo que binarize as imagens sem remover totalmente o texto de nenhuma das imagens.

Após a binarização com esse ponto de corte fixo. O algoritmo calcula o quanto de informação permaneceu na imagem binarizada. Para isso, é feito o cálculo do vetor *stroke-run* da imagem. Essa técnica foi amplamente estudada por Liu e Shihari [11]. O vetor *stroke-run* armazena a quantidade de *stroke-length* com características de texto presentes na imagem. O *stroke-length* pode ser entendido como a largura de uma sequência de pontos conectados que estejam na mesma direção. Por exemplo, à primeira posição do vetor *stroke-run* é inserida a quantidade de pontos (largura 1) da imagem binarizada. À segunda posição é inserida a quantidade de sequências com largura 2 (dois). À terceira posição é inserida a quantidade de sequências com largura 3 (três). Seguindo esse raciocínio o vetor *stroke-run* é totalmente preenchido.

Análises feitas sobre a largura cursiva do texto indicam que para objetos textuais essa largura chega ao máximo 6. Valores maiores que 6 são geralmente de linhas ou elementos geométricos, por isso esses objetos são automaticamente desprezados. A partir deste ponto, ao ser mencionado o vetor *stroke-run*, está sendo feita referência a um vetor com 6 posições que armazena os *stroke-lengths* da imagem que foi binarizada com ponto de corte fixo.

Tendo definido o vetor *stroke-run* cada imagem é classificada como pertencente a um dentre 5 (cinco) grupos de acordo com a quantidade de texto preservado (TP). A quantidade de texto preservada foi definida como a soma dos elementos do vetor *stroke-run*. Para o *Courtesy Amount* do cheque os grupos foram empiricamente definidos da seguinte maneira:

- Classe 1: $TP < 700$ objetos
- Classe 2: $700 \text{ objetos} \leq TP < 1.500$ objetos
- Classe 3: $1.500 \text{ objetos} \leq TP < 2.200$ objetos
- Classe 4: $2.200 \text{ objetos} \leq TP < 3.000$ objetos

- Classe 5: TP \geq 3.000 objetos

Neves, Mello e Silva observaram que para as primeira e segunda classes as imagens que tinham pouca quantidade de texto apresentavam resultados ruins. Além de o fundo ter sido removido, grande parte do texto também foi removida durante o processo de binarização. Análises feitas nessas imagens mostraram que existia uma distribuição alta e concentrada de tons no lado direito do histograma. Além disso, a cor mais freqüente estava muito próxima de 255, inviabilizando a quebra da entropia de Tsallis.

Com a finalidade de identificar as imagens que apresentavam resultados insatisfatórios, as imagens das primeira e segunda classes tiveram seus histogramas analisados por três medidas diferentes: curtose [20], assimetria [20] e cor mais freqüente. Possibilitando uma reclassificação interna das imagens pertencentes a cada uma dessas classes.

A curtose é uma medida de dispersão que caracteriza o achatamento de uma função de distribuição. Veja na Figura 11, se o valor da curtose for igual a zero, o histograma tem o mesmo comportamento de uma gaussiana. Se o valor for menor que zero, o histograma é mais achatado que uma gaussiana, neste caso, as cores mais freqüentes não se encontram tão próximas à média quanto na gaussiana. Caso o valor seja maior que zero, o histograma tem uma distribuição concentrada e com um pico maior que uma gaussiana padrão.

A assimetria analisa a concentração em torno do valor médio do histograma. Veja na Figura 12, se a assimetria da distribuição for nula, a distribuição é semelhante a uma gaussiana. Se o valor da assimetria apresentar valores negativos, a concentração dos pontos se encontra no lado esquerdo do histograma. Caso contrário, a concentração do lado direito do histograma.

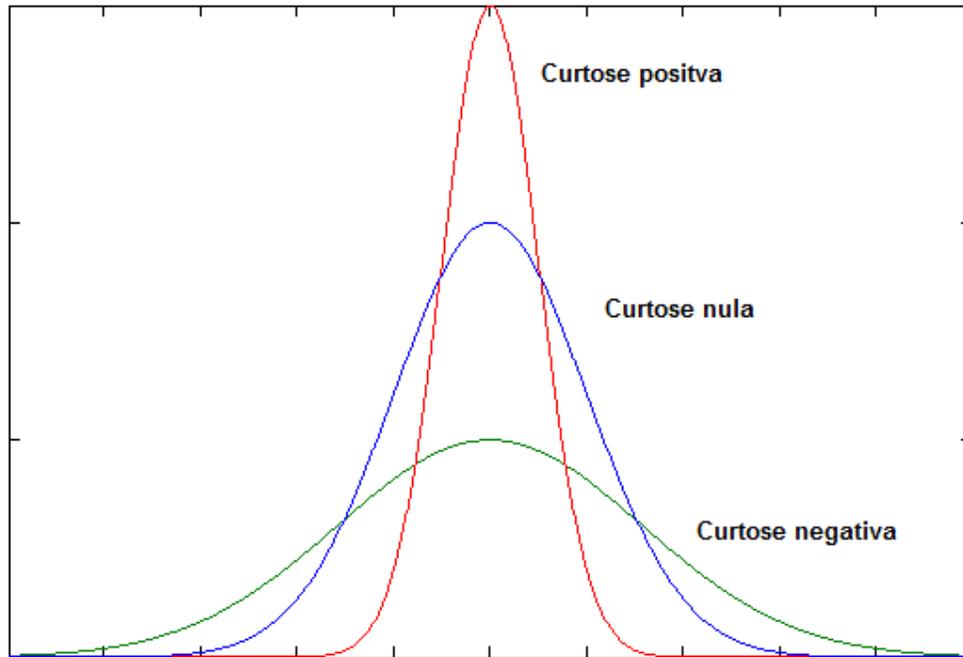


Figura 11. Descrição de Curtose

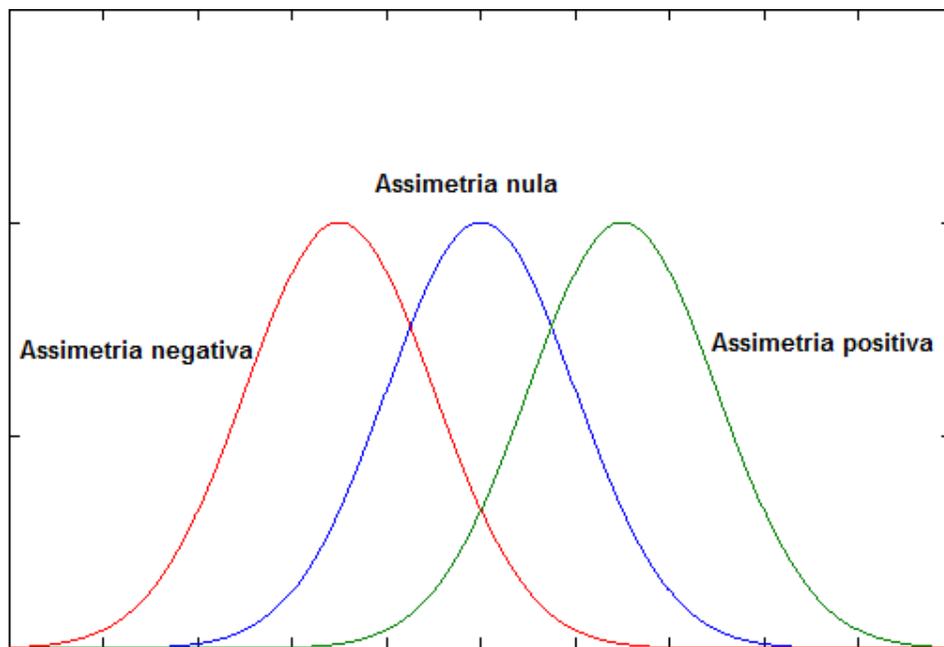


Figura 12. Descrição da assimetria.

Após terem sido definidas novas métricas, a primeira classe foi dividida em três grupos:

- Grupo FRE: Imagens que possuem cor mais freqüente maior que 250;
- Grupo CAF: Imagens que possuem cor mais freqüente maior que 200 e menor que 250, a curtose maior que 2 e assimetria maior que 1;
- As outras imagens desta classe;

A segunda classe foi dividida em dois grupos:

- Grupo CUR: Imagens que possui curtose maior que 20
- As outras imagens desta classe

Após a identificação das imagens claras que apresentam distorção no histograma, é necessário fazer ajustes em seus histogramas. Pois, essas imagens possuem seus tons mais freqüentes muito próximo de 255 e isso inviabiliza o cálculo da entropia de Tsallis. Com uma técnica chamada de Especificação do Histograma [8], é possível modificar uma imagem para que seu histograma fique semelhante a outro histograma base. O histograma base indica a distribuição à qual uma imagem deve se aproximar. Essa técnica foi utilizada para corrigir as distorções nas imagens identificadas como problemáticas a partir das medidas de curtose, assimetria e cor mais frequente.

A imagem escolhida para ser utilizada como base para a especificação foi aquela que seu histograma apresentou maior similaridade com a maioria dos histogramas das outras imagens de cheques disponíveis. Essa similaridade foi medida através da utilização de índice de fidelidade. A imagem que teve maior índice de fidelidade e menor desvio padrão em relação às outras imagens disponíveis foi considerada a imagem base e seu histograma foi utilizado para fazer a especificação dos outros histogramas.

A Figura 13b) contém uma imagem que teve seu histograma especificado c). A especificação do histograma aplicada utilizou o histograma base mostrado na Figura 13a). É possível observar que o histograma da imagem original do CA mudou seu histograma para seguir a forma do histograma base.

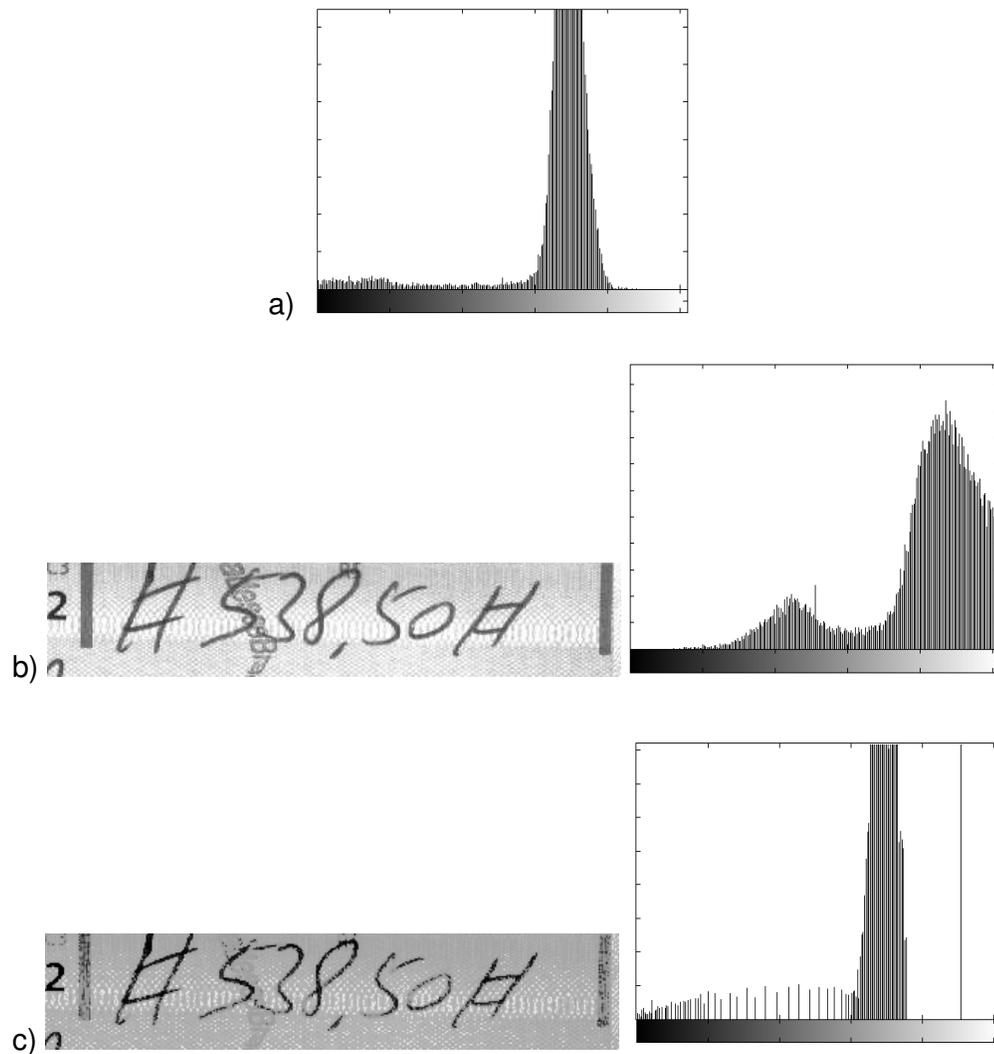


Figura 13. a) Histograma base; b) imagem de um CA e seu histograma; c) imagem do CA após a especificação de seu histograma e seus histograma.

As imagens pertencentes à Classe 1 que possuem a cor mais freqüente maior que 250 ou maior concentração no final do histograma devem ter seu histograma especificado. As imagens da Classe 2 que apresentam extrema concentração também devem ter seus histogramas especificados. Apesar de a especificação modificar a imagem para que ela fique mais escura, em algumas imagens a cor mais freqüente continua muito próxima a 255. Por isso, valores maiores que 250 são desprezados na escolha da cor mais freqüente.

Após a definição dessas métricas, a nova classificação das imagens ficou da maneira como descrita na Tabela 1. Também foram definidos os valores para as constantes m_b , m_w e α que são usados no cálculo da quebra da entropia de Tsallis.

Tabela 1. Parâmetros para o algoritmo de binarização Neves-Mello-Silva

Classes	Números de Objetos	Constantes
1	TP < 700 objetos	<p>Se cor mais freqüente > 250</p> <ul style="list-style-type: none"> ▪ $m_b = 1$; ▪ $m_w = 1$; ▪ $\alpha = 0,19$; <p>Se cor mais freqüente > 200, curtose > 2 e assimetria > 1</p> <ul style="list-style-type: none"> ▪ $m_b = 1$; ▪ $m_w = 1$; ▪ $\alpha = 0,20$; <p>Caso contrário</p> <ul style="list-style-type: none"> ▪ $m_b = 1$; ▪ $m_w = 1$; ▪ $\alpha = 0,18$;
2	700 objetos \leq TP < 1.500 objetos	<p>Se curtose > 20</p> <ul style="list-style-type: none"> ▪ $m_b = 1,5$; ▪ $m_w = 0,5$; ▪ $\alpha = 0,27$; <p>Caso contrário</p> <ul style="list-style-type: none"> ▪ $m_b = 1$; ▪ $m_w = 2,5$; ▪ $\alpha = 0,25$;
3	1.500 objetos \leq TP < 2.200 objetos	<ul style="list-style-type: none"> ▪ $m_b = 1$; ▪ $m_w = 2$; ▪ $\alpha = 0,32$;
4	2.200 objetos \leq TP < 3.000 objetos	<ul style="list-style-type: none"> ▪ $m_b = 1$; ▪ $m_w = 1$; ▪ $\alpha = 0,35$;
5	TP \geq 3.000 objetos	<ul style="list-style-type: none"> ▪ $m_b = 1$; ▪ $m_w = 1$; ▪ $\alpha = 0,4$;

O fluxograma completo para o algoritmo Neves-Mello-Silva, para binarização de imagens, está mostrado na Figura 14.

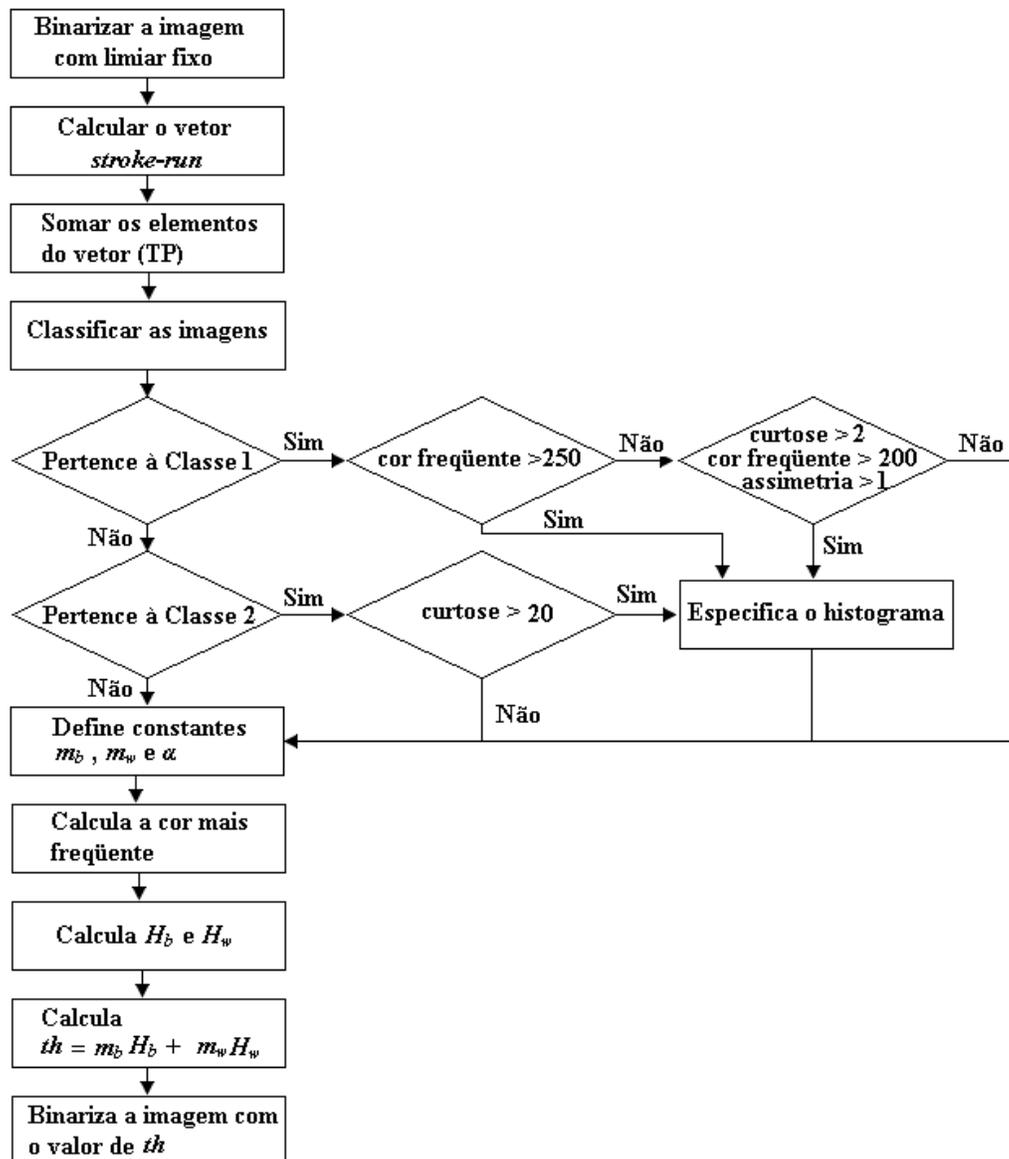


Figura 14. Fluxograma para o algoritmo Neves-Mello-Silva. (Fonte: dissertação de mestrado de Renata Freire de Paiva Neves, título: “Um Novo Algoritmo para Binarização do *Courtesy Amount* de Imagens de Cheques Brasileiros”, Programa de Pós-Graduação em Engenharia da Computação da Universidade de Pernambuco).

O algoritmo descrito neste Capítulo foi publicado previamente em [14][15][16][17].

Na Figura 15 são apresentadas algumas imagens resultantes da aplicação do algoritmo. É apresentada uma imagem para cada classe. A imagem superior é da classe 1, a segunda é da classe 2, até chegar à imagem mais inferior que é da classe 5.

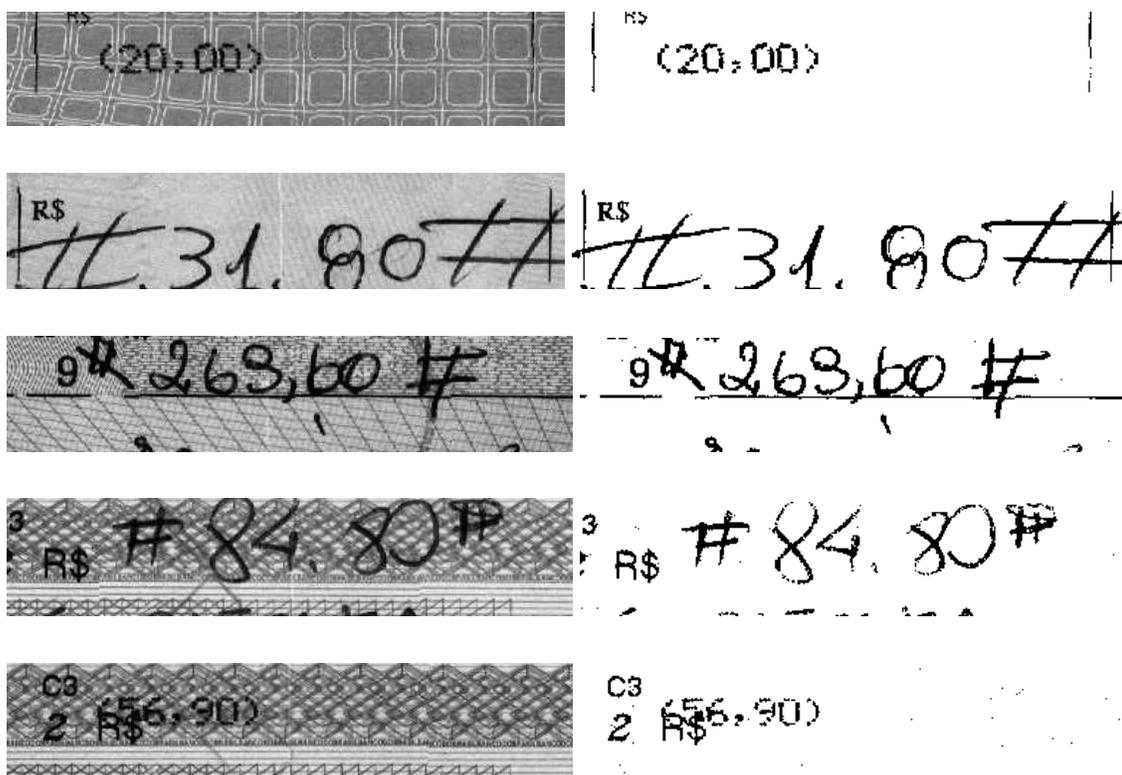


Figura 15. Exemplos da aplicação do algoritmo: (esquerda) imagem original e (direita) imagem binarizada.

Capítulo 4

Experimentos e Resultados

Para avaliar os resultados do algoritmo em estudo, foram utilizados os resultados dos 7 algoritmos de binarização descritos na seção 2.4. Imagens resultantes e tempo de processamento desses algoritmos foram comparados com as imagens resultantes e o tempo de processamento gerados pelo algoritmo Neves-Mello-Silva.

A primeira análise feita foi em relação à eficiência dos algoritmos. Para isso, a eficiência do algoritmo desenvolvido por Neves, Mello e Silva foi comparada com a eficiência apresentada pelos outros algoritmos de binarização. Esses algoritmos só puderam ser utilizados neste tipo de avaliação pois foram implementados pelo mesmo programador, utilizado-se da mesma linguagem de programação e da mesma IDE (Ambiente Integrado de Desenvolvimento, do termo em inglês *Integrated Development Environment*), a saber, Matlab. Isso elimina variações na eficiência relativa às características de uma linguagem, eficiência da ferramenta, ou habilidade do programador.

Foi calculado o tempo utilizado para cada algoritmo fazer a binarização das 1584 imagens dos *Courtesy Amount* dos cheques disponíveis. Para cada algoritmo, todas as imagens foram binarizadas três vezes. O tempo total, em segundos, utilizado para binarizar todas as imagens pela primeira vez, e o tempo total utilizado para a segunda e terceira rodada, estão representados na Tabela 2 por Tempo total 1, Tempo total 2 e Tempo total 3, respectivamente. O tempo médio, por algoritmo, para binarizar todas as imagens está representado por Tempo total médio. Enfim, na última coluna é representado o tempo médio que o algoritmo em questão leva pra binarizar uma imagem de CA.

Tabela 2. Sumarização dos tempos dos algoritmos, medidos em segundos.

Algoritmo	Tempo total 1	Tempo total 2	Tempo total 3	Tempo total médio	Tempo médio por imagem
NevesMelloSilva	71,378979	64,780653	65,465341	67,2083	0,0424
Bersnsen	42,215872	47,689614	47,156697	45,6874	0,0288
Johannsen	98,055850	87,199109	85,196426	90,1505	0,0569
Niblack	345,118025	304,676953	295,905662	315,2335	0,1990
Rosenfeld	65,954010	55,520562	58,366235	59,9469	0,0378
Sauvola	537,401374	538,854241	526,955635	534,4000	0,3374
Wellner	42,568365	37,777763	39,995694	40,1139	0,0253
White	284,744959	269,048882	271,028621	274,9408	0,1736

Os melhores resultados encontrados foram os tempos gerados pelos algoritmos desenvolvidos por Wellner, Bersnsen, Rosenfeld e de Neves, Mello e Silva, respectivamente. Onde o tempo médio encontrado pelo algoritmo de Neves, Mello e Silva, por imagem, foi apenas 17,1 milissegundos mais lento do que o algoritmo que apresentou melhor eficiência. Por outro lado, o algoritmo de Neves, Mello e Silva foi 295 milissegundos mais rápido que o algoritmo de Sauvola, que se apresentou como o algoritmo mais lento. Na Figura 16, os tempos dos algoritmos foram plotados num gráfico. Nele, é possível observar a variação no desempenho dos algoritmos em relação à eficiência na binarização das imagens dos CAs. A partir da análise do gráfico, é possível observar que os cinco primeiro algoritmos apresentaram um desempenho relativamente melhor que os outros algoritmos testados.

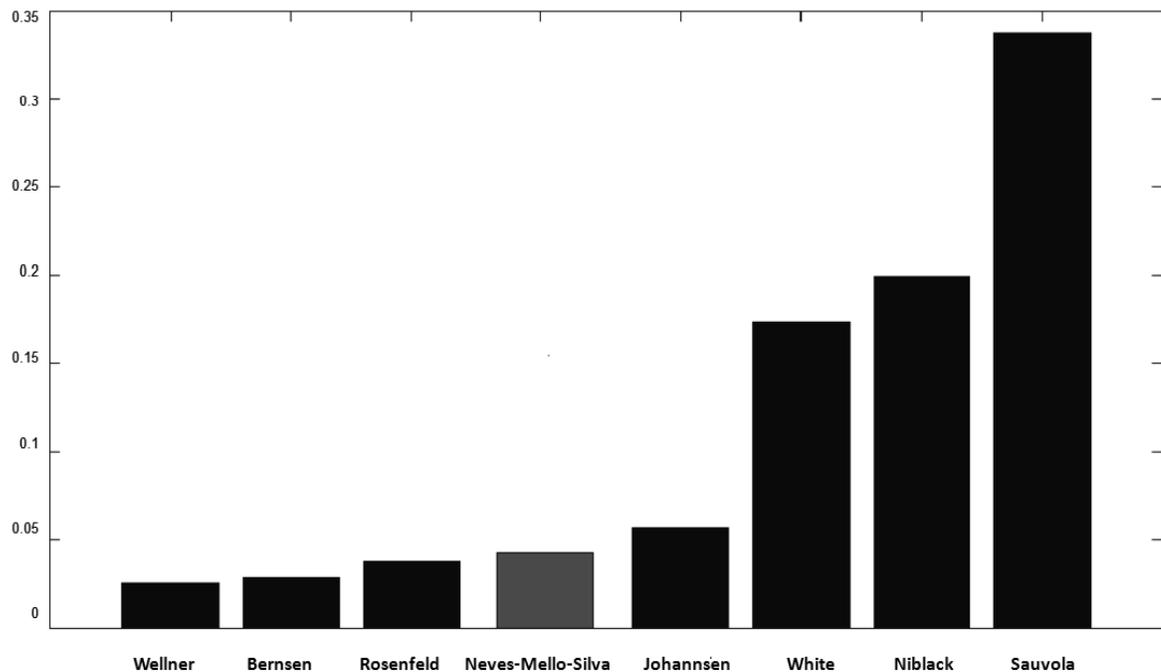


Figura 16. Gráfico da eficiência dos algoritmos.

É válido enfatizar que o tempo de processamento é um fator importante para binarização de cheques. Porque, além da aplicação bancária exigir eficiência nos seus processos, a binarização é apenas uma das etapas do processamento de cheques, e se ela não for realizada de forma eficiente, pode se tornar um gargalo para o restante da aplicação.

A segunda análise feita é em relação à eficácia da binarização gerada pelos algoritmos. Isso significa analisar o quanto os resultados obtidos atingiram o objetivo pretendido. Além disso, é feita uma comparação entre as imagens obtidas pelo algoritmo em estudo e os resultados dos outros algoritmos.

Para constatar se os objetivos foram alcançados, é preciso possuir objetivos bem definidos. Para isso, foram encontradas as imagens *Gold* do *Courtesy Amount* de 550 imagens de cheque, mais de um terço das imagens disponíveis. Imagem *Gold* é uma imagem dita ideal, ela representa o resultado desejado do processo de binarização. Essas imagens foram geradas manualmente a partir de inspeção visual. Para se obter uma imagem ideal é necessário saber balancear os dois principais objetivos da binarização, preservar o texto e eliminação do *background*.

Veja na figura Figura 17a) uma imagem de CA em tons de cinza. Nesta imagem, apesar de a tinta utilizada para o seu preenchimento ser escura, o padrão do *background* também é escuro. Isso é um dos fatores que dificultam o processo de binarização. Outro fator que dificulta a binarização é quando a tinta é muito clara, e neste caso, independente de o *background* ser claro ou escuro, a binarização será prejudicada. Tanto no caso do *background* escuro, como no caso da tinta clara, o contraste da imagem é pequeno, ou seja, os tons do texto e do fundo da imagem são muito próximos, dificultado na definição da imagem ideal.

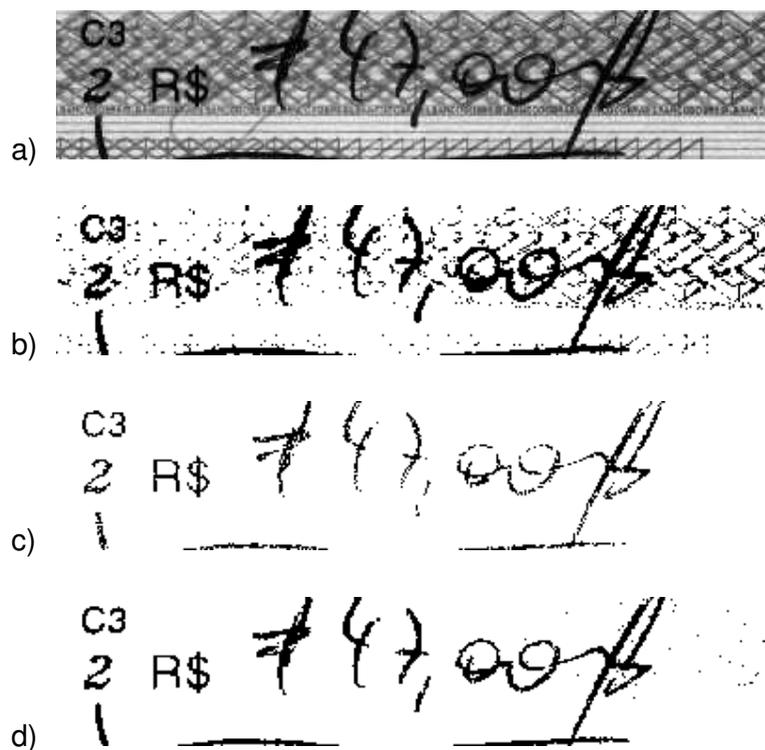


Figura 17. Definição de uma imagem ideal.

Na tentativa de gerar uma imagem que recupere todo o texto presente na Figura 17, chegou-se a imagem b). Nela, além do texto ter sido recuperado, grande parte do *background* também foi preservado na imagem. Numa outra tentativa, agora de gerar uma imagem que remova totalmente o *background*, chegou-se a Figura 17 c). Nesta imagem, o *background* foi totalmente removido, mas houve uma corrosão acentuada do texto recuperado. Sendo assim, apesar de eficazes nos seus objetivos específicos, tanto para imagem b) quanto c), a binarização não gerou uma imagem ideal.

Na binarização que gerou a imagem d) da Figura 17, buscou-se balancear a necessidade da preservação do texto, com a necessidade da remoção do *background*. Apesar da permanência de algum ruído do *background*, o texto foi praticamente todo preservado, gerando-se uma imagem com melhor qualidade do que as apresentadas nas imagens b) e c) da mesma figura. Sendo assim, a imagem d) é considerada uma imagem ideal.

Para avaliar se o algoritmo alcançou bons resultados, cada imagem gerada foi comparada *pixel-a-pixel* com sua respectiva imagem ideal. Dessa comparação são tiradas as quatro seguintes variáveis:

- TP (verdadeiro positivo): indica que o *pixel* foi classificado como tinta na imagem binarizada, assim como na imagem ideal;
- FP (falso positivo): indica que o *pixel* foi classificado como tinta na imagem binarizada, quando ele deveria ter sido classificado como papel, como na imagem ideal;
- TN (verdadeiro negativo): indica que o *pixel* foi classificado como papel na imagem binarizada, assim como na imagem ideal;
- FN (falso negativo): indica que o *pixel* foi classificado como papel na imagem binarizada, quando ele deveria ter sido classificado como tinta, como na imagem ideal.

Sendo:

- P (positivo): indica o total de *pixels* de tinta na imagem ideal, onde:

$$P = TP + FN$$

- N (negativo) indica o total de *pixels* de papel na imagem ideal, onde:

$$N = TN + FP$$

Da análise dessas variáveis pode-se identificar o real comportamento do binarizador. Para isso, quatro medidas de desempenho são calculadas e analisadas. São elas: *precision*, *recall*, *accuracy* e *specificity* [6]. Um bom binarizador deve ter todas essas medidas tendendo para 1 (um).

Precision calcula o erro na classificação do papel de acordo com a equação:

$$Precision = TP / (TP + FP).$$

Recall calcula o erro de classificação da tinta e seu valor é obtido pela equação:

$$Recall = TP/P$$

Accuracy reflete a quantidade de acertos de forma geral. Seu valor é obtido pela equação:

$$Accuracy = (TP + TN)/(P + N)$$

Specificity calcula o acerto na classificação do papel e é dado através da fórmula:

$$Specificity = TN/N$$

Na Tabela 3 está contido o resultado da comparação entre a eficácia do algoritmo Neves-Mello-Silva com outros algoritmos. Para todos os algoritmos é apresentada sua medida de *Precision*, *recall*, *accuracy* e *specificity* em termos de média e desvio padrão.

Tabela 3. Medidas de *Precision*, *Recall*, *Accuracy* e *Specificity* de cada algoritmo.

Algoritmo	<i>Precision</i>	<i>Recall</i>	<i>Accuracy</i>	<i>Specificity</i>
NevesMelloSilva	0,91211(0,4858)	0,92406(0,13472)	0,98810(0,01232)	0,99339(0,01149)
Bernsen	0,99727(0,02092)	0,47151(0,27113)	0,85525(0,14101)	0,99972(0,00231)
Johannsen	0,88682(0,24912)	0,92600(0,13412)	0,98343(0,05064)	0,99337(0,01572)
Niblack	0,96129(0,06171)	0,70385(0,19266)	0,96176(0,03590)	0,99689(0,00519)
Rosenfeld	0,99900(0,01257)	0,63889(0,21440)	0,93458(0,13226)	0,99991(0,00110)
Sauvola	0,96330(0,05913)	0,69672(0,19348)	0,96055(0,03709)	0,99705(0,00498)
Wellner	0,98225(0,03607)	0,46722(0,18747)	0,90225(0,06709)	0,99803(0,00411)
White	0,69393(0,22382)	0,99017(0,05213)	0,97753(0,01805)	0,97680(0,01923)

Observe que a maioria dos algoritmos teve excelentes resultados, mas o único que obteve todas as quatro medidas acima de 0,9 foi o algoritmo implementado nesta monografia.

A Figura 18 contém algumas imagens de CA e o resultado destas imagens após sua binarizadas pelo algoritmo Mello-Neves-Silva. Elas foram algumas das imagens que obtiveram todas as medidas de *Precision*, *recall*, *accuracy* e *specificity* iguais a 1, ou seja, o algoritmo gerou uma imagem ideal para cada uma dessas imagens.

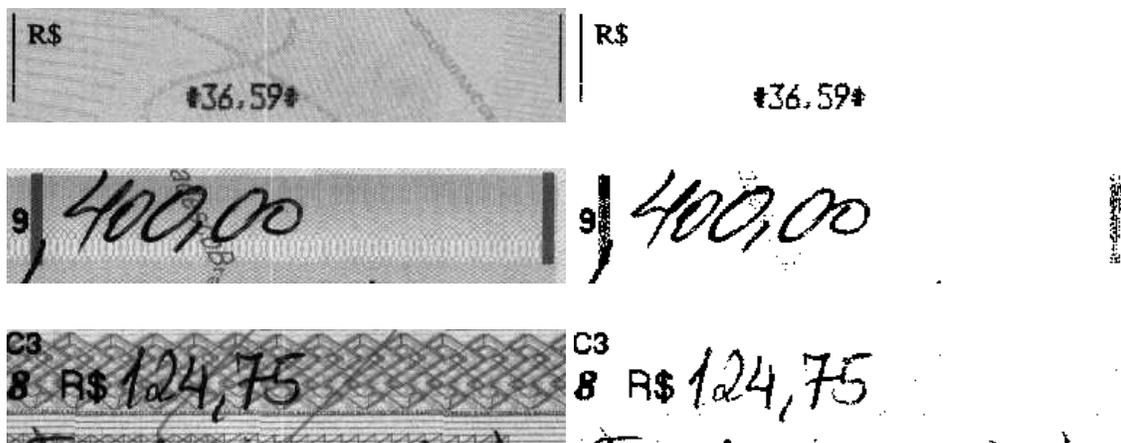


Figura 18. Imagens encontradas pelo algoritmo de Neves, Mello e Silva.

Após o comparativo entre os diversos algoritmos de binarização implementados. É feito um estudo sobre a eficácia de cada uma das classes em que o algoritmo Neve, Mello e Silva foi dividido. Na Tabela 4 os valores de *Precision*, *Recall*, *Accuracy* e *Specificity* são definidos para cada umas das classes e grupos desse algoritmo.

Tabela 4. Comparativo entre as classes do algoritmo Neves-Mello-Silva

Classe	<i>Precision</i>	<i>Recall</i>	<i>Accuracy</i>	<i>Specificity</i>
Classe 1	0,86227(0,19618)	0,92306(0,14163)	0,99292(0,01196)	0,98810(0,01118)
<i>Grupo FRE</i>	0,82385(0,18186)	0,92548(0,10128)	0,98451(0,01185)	0,98723(0,01479)
<i>Grupo CAF</i>	0,94411(0,13507)	0,75603(0,21814)	0,98358(0,01452)	0,99490(0,01552)
Classe 2	0,93630(0,10596)	0,94697(0,08333)	0,98871(0,01140)	0,99385(0,01071)
<i>Grupo CUR</i>	0,88963(0,11037)	0,97124(0,02876)	0,99222(0,00494)	0,99334(0,00666)
Classe 3	0,94799(0,07634)	0,91370(0,14795)	0,98539(0,01548)	0,99300(0,01079)
Classe 4	0,93383(0,12768)	0,97503(0,03638)	0,99179(0,00994)	0,99406(0,01065)
Classe 5	0,90300(0,12414)	0,96472(0,08661)	0,99032(0,01187)	0,99278(0,01046)

Pode-se observar que as 3 últimas classes apresentaram todas suas medidas maiores que 0,9. Comprovando a eficácia na escolha dos parâmetros para estas classes. Contudo, as duas primeiras classes não tiveram o mesmo êxito. As classes 1 e 2, e seus grupos, tiveram valores de *precision* menores que 0,9. O valor do *Precision* indica a quantidade de erros na classificação do papel. O grupo CAF teve um valor de *recall* em torno de 0,7. Isso indica que houve uma quantidade razoável erros na classificação da tinta.

Capítulo 5

Conclusão

Neste trabalho foi estudado e implementado um algoritmo de binarização para imagens de cheques bancários. Esse algoritmo foi desenvolvido por Neves, Mello e Silva. Numa primeira instância, a binarização foi aplicada ao *Courtesy Amount* das imagens do cheque, pois esse campo foi identificado como um campo ideal para essa aplicação.

A eficiência do algoritmo foi analisada em relação à eficiência de outros algoritmos clássicos de binarização. Apesar deste algoritmo estar entre os 4 melhores resultados, esse resultado não é o ideal, pois o problema para o qual o algoritmo foi proposto tem eficiência como um fator crítico.

Ao ser analisado em termos de eficácia, o algoritmo estudado apresentou o melhor desempenho, foi o único dentre os algoritmos implementados que obteve todas as medidas de *Precision*, *Recall*, *Accuracy* e *Specificity* maiores do que 0,9. Ou seja, ele foi o algoritmo que em média gerou os resultados mais próximos do resultado ideal.

Em seguida, foi feita uma análise sobre as classes que o algoritmo divide as imagens. Notou-se que as duas primeiras classes apresentam uma taxa maior de erro em relação às outras classes. Principalmente na classificação do papel representada pela medida de *precision*.

5.1 Contribuições

Apesar da utilização de cheque ter diminuído em todo o mundo, ele ainda é bastante utilizado em transações de altos valores. Sendo assim, o trabalho proposto visa resolver um problema enfrentado pela sociedade atualmente. Ele contribui para a evolução no processo das transações comerciais e bancárias. Essas contribuições podem ser divididas em:

1. Impacto econômico: A automatização dos processos bancários com segurança e precisão leva a uma evolução nos processos bancários elevando a velocidade das transações e diminuindo o índice de erros ocorridos.
2. Impacto social: A melhoria na agilidade e na segurança dos processos bancários aumenta a satisfação e o bem estar da comunidade que utiliza esse tipo de serviço.
3. Impacto Científico: A pesquisa aqui descrita foi publicada nos artigos:
 - I. *A New Technique to Threshold the Courtesy Amount of Brazilian Bank Checks. 15th International Conference on Systems, Signals and Image Processing (IWSSIP), 2008 [15].*
 - II. *A New Algorithm for Thresholding the Courtesy Amount of Brazilian Bank Checks. IEEE International Conference on Systems, Man and Cybernetics (SMC), 2008 [16].*
 - III. *Limiarização do Courtesy Amount de Cheques Brasileiros Baseada em Entropia de Tsallis. Revista IEEE América Latina, 2009 [17]. (Aceito para publicação)*

Com participação da autora desta monografia.

5.2 Trabalhos Futuros

Para trabalhos futuros, algumas melhorias no algoritmo podem ser feitas:

1. Como o fator tempo é um requisito importante para o processamento automático dos cheques. Para trabalhos futuros, pode ser feito um estudo sobre técnicas ou métodos que possam melhorar a eficiência do algoritmo proposto.

2. O algoritmo pode evoluir para ser utilizado na imagem inteira do cheque. O limiar encontrado para a binarização do CA pode ser ideal ou perto do ideal para a binarização da imagem do cheque por completo.
3. Com a finalidade de melhorar o desempenho das classes 1 e 2, que apresentaram um desempenho inferior ao desejado, é proposto um estudo sobre técnicas de pré-processamento que possa melhorar a eficácia da binarização.
4. Ainda com a finalidade de melhorar o desempenho das classes 1 e 2, pode-se adicionar um processo de definição dos parâmetros de cada classe baseado no uso das imagens *gold*, através de treinamento de um algoritmo de aprendizagem de máquina supervisionado.

Bibliografia

- [1] *AiLeader Technologies*. Acessível em: "<http://www.aileader.com.br/>". Acesso em: 22 de maio de 2008.
- [2] Banco Central do Brasil. Disponível em: "<http://www.bcb.gov.br/>." Acesso em: 15 de maio de 2009
- [3] Bernsen, J., "*Dynamic thresholding of gray level images*," ICPR'86: Proc. International Conference on Pattern Recognition, p. 1251–1255, 1986.
- [4] Cormen, T. H.; Leiserson, C. E.; Leiserson, R. L.; Stein, C., "Algoritmos - Teoria e Prática". Campus, 2002
- [5] Dias, D., "Modelo para Representação Eletrônica de Cheques". Brasília: 2006. p.83. Dissertação (Mestrado) - Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, 2006.
- [6] Fawcett, T., "*An introduction to ROC analysis*". Pattern Recognition Letters, vol. 27, p. 861-874, 2006.
- [7] Folha Online. Acessível em: "<http://www1.folha.uol.com.br/folha/dinheiro/ult91u564317.shtml>". Acesso em: 18 de junho de 2008.
- [8] Gonzalez, R. C.; Woods, R.E., "*Digital Image Processing*". New York: Addison-Wesley, 1993.
- [9] Impedovo, S.; Wang, P. S., Bunke, H., "*Automatic bankcheck processing*". World Scientific, vol. 28, 1997
- [10] Jornal da Mídia. Acessível em: "http://www.jornaldamidia.com.br/noticias/2009/04/27/Brasil/Operacoes_com_cartoes_de_debito_s.shtml". Acesso em: 18 de junho de 2008.
- [11] Liu, Y.; Srihari, S. N., "*Document image binarization based on texture feature*", *IEEE Trans. Pattern Anal. Machine Intell.* Vol. 19, p. 533-540, 1997.

- [12] Mello, C. A. B; Schuler, L. A., “*Tsallis Entropy-Based Thresholding Algorithm for Images of Historical Documents.*” *IEEE Int. Conference on Systems, Man and Cybernetics, Montreal, 2007.* p.1112-1117, 2007.
- [13] Mello, C. A. B; Oliveira, A. L. I.; Sanchez, A., “*Historical Document Image Binarization.*” *International Conference on Computer Vision Theory and Applications, 2008, Funchal. Proceedings of the International Conference on Computer Vision Theory and Applications.* INSTICC Press, Portugal, 2008. p.108-113, 2008.
- [14] Neves, R. F. P.; Mello, C. A. B.; Silva, M. S.; Bezerra, B., “*A New Technique to Threshold the Courtesy Amount of Brazilian Bank Checks.*” *15th International Conference on Systems, Signals and Image Processing (IWSSIP).* Bratislava, Junho 2008. Vol. 25, p.93 – 96, 2008.
- [15] Neves, R. F. P.; Mello, C. A. B.; Silva, M. S.; Bezerra, B., “*A New Algorithm for Thresholding the Courtesy Amount of Brazilian Bank Checks.*” *IEEE International Conference on Systems, Man and Cybernetics (SMC).* Cingapura, 2008. p. 1226-1230, 2008.
- [16] Neves, R. F. P., “*Um Novo Algoritmo para Binarização do Courtesy Amount de Imagens de Cheques Brasileiros.*” 2008. p.88. Dissertação (Mestrado) – Departamento de Engenharia da Computação, Universidade de Pernambuco, Recife. 2008.
- [17] Neve, R. F. P. ; Mello, C. A. B. ; Silva, M. S. ; Bezerra, B., “*Limiarização do Courtesy Amount de Cheques Brasileiros Baseada em Entropia de Tsallis.*” *Revista IEEE América Latina*, 2009. (Aceito para publicação)
- [18] Niblack, W., “*An Introduction to Image Processing.*” Englewood Cliffs, NJ: Prentice-Hall, 1986. p. 115–116, 1986.
- [19] Parker, J. R., “*Algorithms for Image Processing and Computer Vision.*” John Wiley and Sons, 1997.
- [20] Pedrini, H.; Schwartz, W. R., “*Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações.*” 1ª ed. São Paulo: Thompson Learning, 2008

- [21] Pimenta Comunicação. Acessível em:
“<http://www.pimenta.com/lermais_materias.php?cd_materias=3711>”.
Acesso em: 18 de junho de 2008.
- [22] Pun, T., “*Entropic Thresholding, the new Approach*”. *Computer Graphics and Image Processing*, 1981. vol. 16, p. 210-239, 1981.
- [23] Rosenfeld, A.; Torre, P., “*Histogram concavity analysis as an aid in threshold selection*”. *IEEE Trans. Syst. Man Cybern. SMC-13*, 1983. Vol.13, p.231–235, 1983.
- [24] Sauvola, J.; Pietaksinen, M., “*Adaptive document image binarization*”, *Pattern Recogn*, 2000. Vol. 33, p. 225–236, 2000.
- [25] Sezgin, M.; Sankur, B., “*Survey over image thresholding techniques and quantitative performance evaluation*”. *Jornal of Eletronic Imaging*, 2004. Vol.13, p.146-165, 2004.
- [26] Shannon, C. E., “*A Mathematical Theory of Communication*”, *The Bell System Technical Journal*, Julho e Outubro, 1948. Vol. 27, p. 379–423/623–656, 1948.
- [27] Tsallis, C, “*Possible Generalization of Boltzmann-Gibbs statistics.*”, *Journal of Statistical Physics*, 1988. vol. 52, p. 479-487, 1988.
- [28] Uol Economia. Acessível em:
“<http://economia.uol.com.br/ultnot/infomoney/2009/05/21/ult4040u19476.jhtm>”.
Acesso em: 18 de junho de 2008.
- [29] Wellner, P., “*Interacting with Paper on the Digital Desk*”. *Communications of the ACM*, 1993. Vol 36, p. 86-96, 1993.
- [30] White, J. M.; Rohrer, G. D., “*Image thresholding for optical character recognition and other applications requiring character image extraction,*” *IBM J. Res. Dev.* 27(4), 1983. P.400–411, 1983.