



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

RODRIGO DALVIT CARVALHO DA SILVA

**UM ESTUDO SOBRE A EXTRAÇÃO DE CARACTERÍSTICAS E A
CLASSIFICAÇÃO DE IMAGENS INVARIANTES À ROTAÇÃO EXTRAÍDAS DE
UM SENSOR INDUSTRIAL 3D**

FORTALEZA

2014

RODRIGO DALVIT CARVALHO DA SILVA

UM ESTUDO SOBRE A EXTRAÇÃO DE CARACTERÍSTICAS E A
CLASSIFICAÇÃO DE IMAGENS INVARIANTES À ROTAÇÃO EXTRAÍDAS DE UM
SENSOR INDUSTRIAL 3D

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Teleinformática, do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial para obtenção do Título de Mestre em Engenharia de Teleinformática. Área de Concentração: Sinais e Sistemas

Orientador: Prof. Dr. George André Pereira Thé

FORTALEZA

2014

RODRIGO DALVIT CARVALHO DA SILVA

UM ESTUDO SOBRE A EXTRAÇÃO DE CARACTERÍSTICAS E A
CLASSIFICAÇÃO DE IMAGENS INVARIANTES À ROTAÇÃO EXTRAÍDAS DE UM
SENSOR INDUSTRIAL 3D

Dissertação de Mestrado apresentada ao
Programa de Pós-Graduação em Engenharia
de Teleinformática, do Centro de Tecnologia
da Universidade Federal do Ceará, como
requisito parcial para obtenção do Título de
Mestre em Engenharia de Teleinformática.
Área de Concentração: Sinais e Sistemas

Aprovada em: ____/____/_____.

BANCA EXAMINADORA

Prof. Dr. George André Pereira Thé (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Guilherme de Alencar Barreto
Universidade Federal do Ceará (UFC)

Prof^a. Dr^a. Fátima Nelsizeuma Sombra de Medeiros
Universidade Federal do Ceará (UFC)

Prof. Dr. Eduardo Furtado de Simas Filho
Universidade Federal da Bahia (UFBA)

AGRADECIMENTOS

Gostaria de começar agradecendo a Deus, minha esperança, quem esteve presente e que nos momentos difíceis e de desânimo alegrou meu coração e me inspirou dando-me ânimo para continuar a jornada e que sem o equilíbrio espiritual não teria conseguido concluir mais esta etapa em minha vida.

A minha namorada, Sabrina, que sempre esteve ao meu lado nas horas felizes e tristes a qual ao seu lado se torna insignificante. Pelo carinho, simpatia, dedicação e felicidade, e espero algum dia poder recompensar

A minha mãe, Gertrudes Dalvit, que está sempre mostrando o caminho do amor e da fé, da verdade e do saber, que com muita paciência ensinou-me o real significado do trabalho.

À amiga e irmã Nadia Dalvit, por ser quem ela é. Ajudando sempre que preciso seja a hora que for.

A minha família por toda paciência e incentivo tanto financeiro quanto moral durante todo esse caminho.

Um agradecimento especial ao meu orientador pela paciência, pela divisão de conhecimentos proporcionados durante a produção deste trabalho, e pela receptividade quando o procurei para que me orientasse. Agradeço-o pela competência e dedicação a qual teve e tem me inspirado a continuar trilhando o caminho da docência.

Agradecendo aos professores Guilherme Barreto e Fátima Sombra, que me auxiliaram dando-me uma direção a seguir. Agradeço também aos professores Paulo César Cortez, Giovanni Barroso, Carlos Estêvão, pela competência e dedicação ao ensino.

Não poderia deixar de agradecer aos meus amigos do CENTAURO, que além de serem fonte de inspiração para as minhas ideias, ajudaram-me com opiniões, críticas, elogios e muitas risadas.

Por fim, concludo esta seção com um grande obrigado a todas as pessoas conhecidas ou anônimas que passaram por minha vida e que deixaram algo de si.

*“Consulte não a seus medos mas a suas esperanças e sonhos.
Pense não sobre suas frustrações, mas sobre seu potencial não usado.
Preocupe-se não com o que você tentou e falhou, mas com aquilo que ainda é possível fazer.”*

Papa João XXIII

RESUMO

Neste trabalho, é discutido o problema de reconhecimento de objetos utilizando imagens extraídas de um sensor industrial 3D. Nós nos concentramos em 9 extratores de características, dos quais 7 são baseados nos momentos invariantes (Hu, Zernike, Legendre, Fourier-Mellin, Tchebichef, Bessel-Fourier e Gaussian-Hermite), um outro é baseado na Transformada de Hough e o último na análise de componentes independentes, e, 4 classificadores, *Naive Bayes*, *k*-Vizinhos mais Próximos, Máquina de Vetor de Suporte e Rede Neural Artificial-Perceptron Multi-Camadas. Para a escolha do melhor extrator de características, foram comparados os seus desempenhos de classificação em termos de taxa de acerto e de tempo de extração, através do classificador *k*-Vizinhos mais Próximos utilizando distância euclidiana. O extrator de características baseado nos momentos de Zernike obteve as melhores taxas de acerto, 98.00%, e tempo relativamente baixo de extração de características, 0.3910 segundos. Os dados gerados a partir deste, foram apresentados a diferentes heurísticas de classificação. Dentre os classificadores testados, o classificador *k*-Vizinhos mais Próximos, obteve a melhor taxa média de acerto, 98.00% e, tempo médio de classificação relativamente baixo, 0.0040 segundos, tornando-se o classificador mais adequado para a aplicação deste estudo.

Palavras - chave: Momentos Invariantes, Transformada de Hough, Análise de Componentes Independentes, *Naive Bayes*, *k*-Vizinhos mais Próximos, Máquina de Vetor de Suporte, Rede Neural Artificial-Perceptron Multi-Camadas.

ABSTRACT

In this work, the problem of recognition of objects using images extracted from a 3D industrial sensor is discussed. We focus in 9 feature extractors (where seven are based on invariant moments -Hu, Zernike, Legendre, Fourier-Mellin, Tchebichef, Bessel-Fourier and Gaussian-Hermite-, another is based on the Hough transform and the last one on independent component analysis), and 4 classifiers (Naive Bayes, k-Nearest Neighbor, Support Vector machines and Artificial Neural Network-Multi-Layer Perceptron). To choose the best feature extractor, their performance was compared in terms of classification accuracy rate and extraction time by the k -nearest neighbors classifier using euclidean distance. The feature extractor based on Zernike moments, got the best hit rates, 98.00 %, and relatively low time feature extraction, 0.3910 seconds. The data generated from this, were presented to different heuristic classification. Among the tested classifiers, the k -nearest neighbors classifier achieved the highest average hit rate, 98.00%, and average time of relatively low rank, 0.0040 seconds, thus making it the most suitable classifier for the implementation of this study.

Keywords: Invariant Moments, Hough Transform, Independent Component Analysis, *Naive Bayes*, k -Nearest Neighbors, Support Vector Machine, Artificial Neural Network – Multi-Layer Perceptron.

LISTA DE ILUSTRAÇÕES

| | |
|--|----|
| Figura 1 – (a) Imagem original, (b) rotacionada, (c) imagem que sofreu rotação e alteração na escala, e (d) rotação e translação. | 16 |
| Figura 2 - Polinômio radial de Zernike de ordem 0-5 e baixas repetições. | 24 |
| Figura 3 - Polinômio de Legendre $P_p(x)$ com $p = 0,1,\dots,5$ | 25 |
| Figura 4 - Polinômio radial $Q_p(r)$ dos momentos de Fourier-Mellin com $p = 0,1,\dots,5$ | 27 |
| Figura 5 - Polinômios escalados de Tchebichef para $N = 40$ | 28 |
| Figura 6 - Polinômio radial $J_1(\lambda_n x)$ dos momentos de Bessel-Fourier com $n = 0,1,\dots,5$ | 30 |
| Figura 7 - Polinômios de Gaussian-Hermite de graus $p = 0,1,\dots,5$ | 31 |
| Figura 8 - (a) Imagem original e (b) seu correspondente espaço de Hough. | 32 |
| Figura 9 - Passos para o processo de classificação. | 36 |
| Figura 10 - (a) Problema linearmente separável. O espaço entre as linhas tracejadas é a margem de separação ótima, máxima. (b) Problema não linearmente separável. As linhas tracejadas, margem de separação ótima, são encontradas de modo a ser a maior margem com menor erro no conjunto de treinamento. | 42 |
| Figura 11 - Representação das etapas para classificação dos objetos. | 49 |
| Figura 12 - Estrutura física mostrando os equipamentos utilizados. | 50 |
| Figura 13 - Caixas com dimensões $15 \times 10.5 \times 7.2$ cm, $15 \times 14 \times 6$ cm, e $21.5 \times 16.2 \times 9.6$ cm respectivamente, e com resolução 50×64 pixels. | 51 |
| Figura 14 - Taxa de acerto e Matriz de Confusão. | 57 |
| Figura 15 - Acurácia e Matriz de Confusão da distância City Block para $k = 1$ (Taxa de acerto = 98.00%). | 58 |
| Figura 16 - Acurácia e Matriz de Confusão da distância City Block para $k = 1$ (Taxa de acerto = 98.81%). | 59 |

| | |
|---|----|
| Figura 17 - Acurácia e Matriz de Confusão da distância City Block para $k = 1$ (Taxa de acerto = 100.00%)..... | 59 |
| Figura 18 - Acurácia e Matriz de Confusão à função <i>kernel</i> polinomial (a), rbf (b) e rbf / polinomial (c)..... | 60 |
| Figura 19 - Acurácia e Matriz de Confusão da função de ativação tangente hiperbólica (a), (b) e tangente hiperbólica / logística (c)..... | 62 |
| Figura 20 – Amostras classificadas × amostras reais dos classificadores <i>Naive Bayes</i> e RNA-MLP... | 63 |
| Figura 21 - Amostras classificadas × amostras reais classificador <i>k</i> -NN e SVM. | 64 |
| Figura 22 – Taxa Média de Acerto versus Tempo de Classificação. As barras representam os tempos mínimos e máximos de classificação para cada um dos classificadores e os símbolos seus respectivos tempos médios..... | 66 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 1 : <i>Kernels</i> para Máquina de Vetor de Suporte. | 43 |
| Tabela 2 : Matriz de Confusão..... | 47 |
| Tabela 3 : Número de características extraídas. | 51 |
| Tabela 4 : Taxa de reconhecimento dos momentos, HT e ICA, utilizando k -NN..... | 53 |
| Tabela 5 : Tempo extração e classificação..... | 53 |
| Tabela 6 : Taxa de acerto tomando 50% da base de dados para treinamento. | 55 |
| Tabela 7 : Taxa de acerto tomando 80% da base de dados para treinamento. | 55 |
| Tabela 8 : Tempo de treinamento dos classificadores..... | 65 |
| Tabela 9 : Tempo de classificação dos classificadores. | 65 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|--------------|--|
| BF | Momentos de Bessel-Fourier |
| FM | Momentos de Fourier-Mellin |
| FN | Falso Negativo |
| FNR | Razão de Falsos Negativos – Falsa Rejeição |
| FP | Falso Positivo |
| FPR | Razão de Falsos Positivos – Falsa Aceitação |
| GH | Momentos de Gaussian-Hermite |
| H | Momentos de Hu |
| HT | Transformada de Hough (<i>Hough Transform</i>) |
| ICA | Análise de Componentes Independentes (<i>Independent Component Analysis</i>) |
| ICs | Componentes Independentes (<i>Independent Components</i>) |
| <i>k</i> -NN | <i>k</i> -Vizinhos mais Próximos (<i>k-Nearest Neighbors</i>) |
| L | Momentos de Legendre |
| LoG | Laplaciana da Gaussiana |
| MLP | Perceptron Multi-Camadas (<i>Multi-Layer Perceptron</i>) |
| PCA | Análise de Componentes Principais (<i>Principal Component Analysis</i>) |
| rbf | Função de Base Radial (<i>Radial Basis Function</i>) |
| RNA | Rede Neural Artificial |
| T | Momentos de Tchebichef |
| TPR | Proporção de Verdadeiros Positivos - Sensibilidade |
| VP | Verdadeiro Positivo |
| Z | Momentos de Zernike |

LISTA DE SÍMBOLOS

| | |
|----------------------|--|
| \hat{H}_p | Polinômios de Gaussian-Hermite |
| δ_j | Gradiente Local |
| Δw_{ji} | Gradiente Descendente |
| B_{nm} | Momentos de Bessel-Fourier |
| C_{pq} | Momentos Complexos de Bessel-Fourier |
| I_{xy} ou I_{ij} | Imagem |
| J_1 | Função de Bessel-Fourier de 1° ordem |
| L_D | Lagrangiana |
| L_{pq} | Momentos de Legendre |
| M_{pq} | Polinômios Ortogonais de Fourier-Mellin |
| M_{pq}^* | Conjugado Complexo dos Polinômios Ortogonais de Fourier-Mellin |
| O_{pq} | Momentos Normalizados de Fourier-Mellin |
| P_p | Polinômio de Legendre |
| Q_p | Polinômios Radiais dos Momentos de Fourier-Mellin |
| R_{nm} | Polinômios Radiais de Zernike |
| T_{pq} | Momentos de Tchebichef |
| V_{nm} | Polinômios Complexos de Zernike |
| \bar{X} | Média de X |
| Z_{nm} | Momentos de Zernike |
| a_n | Constante de Normalização dos Momentos de Bessel-Fourier |
| m_{pq} | Momentos Geométricos / Regulares de uma Imagem |
| t_p | Polinômios Escalados de Tchebichef |
| $w^T x + w_0$ | Regra de Decisão |
| y_j | Função de Ativação |
| θ_{nm} | Ângulo entre o vetor ρ e o eixo x no sentido anti-horário |
| η_{pq} | Momentos Normalizados |
| λ_N | Fator de Normalização dos Momentos de Zernike |
| λ_n | 0° s da Função de Bessel-Fourier de 1° ordem |
| λ_{pq} | Constante de Normalização dos Momentos de Legendre |
| μ_{pq} | Momentos Centrais |
| $\phi_j(g_j(n))$ | Função de Ativação Logística |

| | |
|---------------|--|
| ε | Soma Instantânea dos Erros Quadráticos |
| $E\{x\}$ | Média de x |
| $P(Y_j)$ | Probabilidade a Priori da classe Y_j |
| $P(Y_j x_i)$ | Probabilidade a Posteriori |
| $P(x_i Y_j)$ | Função de Densidade de Probabilidade Condicional da classe Y_j |
| $d(x, x')$ | Distâncias Euclideana, City-Block, Cosseno e Correlação |
| g | Função Discriminante Linear |
| k | Função de Núcleo (<i>kernel</i>) |
| p ou n | Ordem |
| q ou m | Repetição |
| s | Componentes Independentes |
| z | Dados x branqueados |
| A | Matriz de Mistura |
| I | Imagem |
| M | Momentos de Gaussian-Hermite |
| T | Transposição da Matriz |
| V | Matriz de Branqueamento |
| W | Inversa da Matriz de Mistura (A^{-1}) |
| Γ | Função Gamma |
| η | Taxa de Aprendizado |
| ρ | Comprimento do vetor desde sua origem ao <i>pixel</i> (x, y) |
| $\rho(p, N)$ | Norma Quadrada dos Polinômios Escalados de Tchebichef |
| σ | Parâmetro Escalar |
| ϕ | Espaço de Características Transformadas |

SUMÁRIO

| | |
|--|-----------|
| 1. INTRODUÇÃO..... | 14 |
| 1.1 Caracterização do Problema e Motivação | 14 |
| 1.2 Objetivo e Contribuição..... | 17 |
| 1.2.1 Objetivo Principal..... | 17 |
| 1.2.2 Objetivo Específico | 17 |
| 1.3 Produção Científica durante o mestrado..... | 18 |
| 1.4 Organização da Dissertação..... | 19 |
| 2. EXTRAÇÃO DE CARACTERÍSTICAS | 20 |
| 2.1 Momentos Invariantes | 20 |
| 2.1.1 <i>Momentos de Hu</i> | 21 |
| 2.1.2 <i>Momentos de Zernike</i> | 22 |
| 2.1.3 <i>Momentos de Legendre</i> | 24 |
| 2.1.4 <i>Momentos de Fourier-Mellin</i> | 25 |
| 2.1.5 <i>Momentos de Tchebichef</i> | 27 |
| 2.1.6 <i>Momentos de Bessel-Fourier</i> | 29 |
| 2.1.7 <i>Momentos de Gaussian-Hermite</i> | 30 |
| 2.2 Transformada de Hough | 31 |
| 2.3 Análise de Componentes Independentes | 33 |
| 2.4 Considerações Finais deste Capítulo | 36 |
| 3. CLASSIFICADORES | 37 |
| 3.1 Classificador Bayesiano Linear | 38 |
| 3.1.1 <i>Classificador Naive Bayes</i> | 39 |
| 3.2 <i>k</i> -Vizinhos Mais Próximos | 40 |

| | | |
|-----------|---|-----------|
| 3.3 | Máquina de Vetor de Suporte | 41 |
| 3.4 | Rede Neural Artificial..... | 44 |
| 3.5 | Avaliação do Desempenho dos Classificadores | 46 |
| 3.5.1 | <i>Matriz de Confusão</i> | 46 |
| 3.6 | Considerações Finais deste Capítulo | 48 |
| 4. | METODOLOGIA E ANÁLISE DOS RESULTADOS | 49 |
| 4.1 | Metodologia..... | 49 |
| 4.2 | Resultados Experimentais I – Extração de Características..... | 52 |
| 4.3 | Resultados Experimentais II – Classificação..... | 56 |
| 4.3.1. | <i>Classificador Naive Bayes</i> | 56 |
| 4.3.2. | <i>Classificador k-NN</i> | 58 |
| 4.3.3. | <i>Classificador Máquina de Vetor de Suporte</i> | 60 |
| 4.3.4. | <i>Classificador Rede Neural – Perceptron Multi-Camadas</i> | 61 |
| 4.3.5. | <i>Representação das Classes</i> | 63 |
| 4.3.6. | <i>Tempo de Processamento dos Classificadores</i> | 64 |
| 4.4 | <i>Softwares Utilizados</i> | 66 |
| 5. | CONCLUSÃO..... | 68 |
| | REFERÊNCIAS..... | 70 |

1. INTRODUÇÃO

Esta dissertação apresenta uma investigação de desempenho entre diferentes métodos de extração de características e diferentes métodos de classificação visando uma aplicação específica no reconhecimento de imagens industriais.

Na Seção 1.1 são apresentadas a caracterização do problema e a motivação que influenciaram o desenvolvimento deste trabalho. Em seguida os objetivos geral e específicos são indicados na Seção 1.2.

1.1 Caracterização do Problema e Motivação

A capacidade humana de reconhecer objetos, independente de eventual rotação, translação ou mudança de escala, é uma das características mais básicas e importantes para a interação do homem com o ambiente (CICHY *et al.*, 2013). Esta capacidade de reconhecimento proporciona também aos seres humanos, a habilidade única de detecção e de ação em uma vasta gama de situações. Além disso, a mesma permite a rotulação de objetos presentes em qualquer lugar do espaço, independente de sua orientação.

Em aplicações de visão computacional, uma questão fundamental é o reconhecimento de objetos no espaço de trabalho, independentemente de eventuais transformações. O reconhecimento de objetos implica na atribuição de um rótulo de acordo com a sua descrição característica, onde a mesma entende-se como uma quantidade mínima de dados que permite representar este objeto.

No contexto de automação industrial, a visão computacional fornece soluções inovadoras. De fato, muitas atividades industriais, como sistema de inspeção visual automática (PARK *et al.*, 1989), controle de qualidade (ASOUDEGI *et al.*, 1991), sistemas de controle (CHEN *et al.*, 1996), micromecânica (BAIDYK *et al.*, 2009), entre outros, beneficiaram-se com a aplicação da tecnologia de visão de máquina em processos de manufatura. A tecnologia de visão de máquina melhora a produtividade e a gestão da qualidade, proporcionando uma vantagem competitiva para as indústrias que dela fazem uso (MALAMAS *et al.*, 2003).

Graças aos recentes avanços na aquisição de dados, processamento e sistemas de controle de processos, a eficiência de muitas das aplicações industriais, tais como a contagem e seleção de objetos em esteiras transportadoras (BOZMA *et al.*, 2002; GLUD 2010), foi melhorada com a ajuda de sistemas de processamento e classificação visual automatizada

(SELVER *et al.*, 2011). No entanto, a classificação de objetos tem-se revelado um problema complexo para a automação industrial em qualquer tipo de aplicação, pois envolve a aquisição de imagem, o pré-processamento, extração de características e a classificação. Assim, embora computadores de alta capacidade estejam disponíveis, a complexidade matemática na modelagem dessas tarefas torna o controle baseado em imagens um problema desafiador em ambiente industrial, particularmente porque os tempos nos processos produtivos exigem rápidas tomadas de decisão.

O reconhecimento de objetos invariantes à rotação tem recebido cada vez mais atenção em pesquisas da área de reconhecimento de padrões (BELKASIM *et al.*, 1991; FLUSSER *et al.*, 1994; SLUZEK *et al.*, 1995; MUKUNDAN *et al.*, 1996; NOVOTNI *et al.*, 2004; MERCIMEK *et al.*, 2005; FLUSSER *et al.*, 2006; WANG *et al.*, 2010). Desde reconhecimento de contornos de aeronaves, estimação de posição e altitude de objetos no espaço 3D, registro de imagens de satélites e controle de qualidade industrial, cada tipo de aplicação tem seus próprios requisitos e restrições. Desta forma, se torna impossível o desenvolvimento de uma técnica que resolva todos os requisitos necessários à verificação de diferentes técnicas para uma adequada aplicação.

Uma técnica largamente utilizada para a análise de imagens é a que se baseia nos momentos invariantes, estes são utilizados como descritores de formas em uma variedade de aplicações, reconhecimento de padrões, classificação objeto, reconhecimento de face, detecção de borda, visão computacional aplicadas na robótica, entre outras, (MINDRU *et al.*, 2004; MOKHATARIAN *et al.*, 2005; FLUSER *et al.*, 2006; NABTI *et al.*, 2008; ZHANG *et al.*, 2009; MA *et al.*, 2010; BELGHINI *et al.*, 2012; FLUSER *et al.*, 2013). Em muitas delas as características extraídas devem possuir invariância a escala, rotação e translação, como mostra a Figura 1.

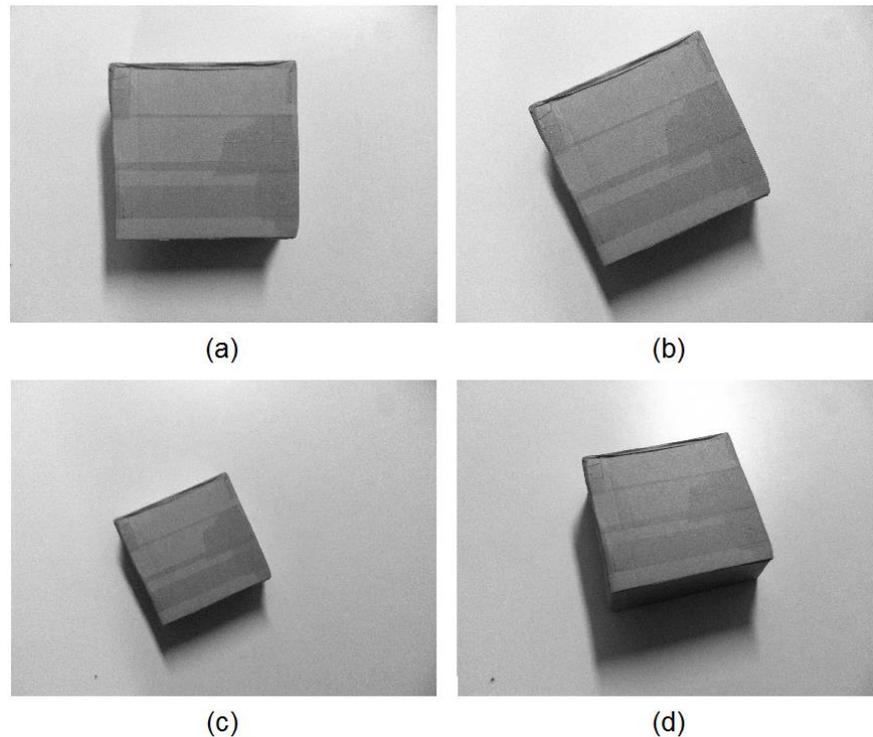


Figura 1 – (a) Imagem original, (b) rotacionada, (c) imagem que sofreu rotação e alteração na escala, e (d) rotação e translação.

Outra abordagem para análise de imagem é a transformada de Hough, que é uma técnica elegante e versátil (HAULE *et al.*, 1989) a qual mapeia pontos do espaço de imagem em curvas em um espaço de parâmetros ou espaço de acumulador, e tem sido utilizada em vários problemas incluindo a detecção de formas (MAJI *et al.*, 2009; BARINOVA *et al.*, 2010).

Uma terceira e mais recente técnica para representação de dados, mais poderosa no que se diz respeito à descrição das características e à quantidade de descritores, é a Análise de Componentes Independentes (HYVÄRINEN *et al.*, 2000).

Uma pergunta natural que pode ser colocada neste cenário diz respeito a qual técnica melhor se adapta às particularidades de aplicações industriais típicas. A escolha de uma ou outra técnica deve ser guiada, primordialmente, pelos tempos de processamento envolvidos e pela taxa de acerto nesta fase de classificação.

Portanto é notória a importância da comparação entre diferentes métodos de extração de características e classificadores aplicados ao reconhecimento de objetos, para a automação industrial, e, a partir deste contexto, o presente trabalho visa mostrar a viabilidade na implantação prática de sistemas de detecção de objetos. Em vista disso, pretende-se aplicar

as técnicas de reconhecimento de padrões utilizando as abordagens dos Momentos Invariantes, Transformada de Hough e Análise de Componentes Independentes.

Por consequência, é de fundamental importância avaliar se é possível realizar tarefas de classificação por visão, em tempos breves com taxas de acerto satisfatórias, sem que para isso se faça uso de grandes volumes de dados (tais quais os gerados por câmeras de alta qualidade de imagem). Assim, esta dissertação trata da investigação de algoritmos que permitem a classificação de objetos através de sistemas de visão por computador. Isto levará à tentativa de resolução de problemas recorrentes do cotidiano industrial, gerados pela necessidade de aumento da produtividade.

Com o desenvolvimento deste tema, a detecção de objetos com um sistema automático irá melhorar a produtividade através da automatização e redução dos tempos de processamento.

1.2 Objetivo e Contribuição

1.2.1 Objetivo Principal

O objetivo geral desta dissertação é avaliar o desempenho de 9 algoritmos de extração de características baseados nos Momentos Invariantes, na Transformada de Hough e na Análise de Componentes Independentes, buscando a melhora no processo de classificação de imagens feitas a partir de um sensor industrial 3D de baixa resolução maximizando assim seu potencial em aplicações industriais.

1.2.2 Objetivo Específico

No intuito de alcançar o objetivo geral definido para esta pesquisa, os seguintes objetivos específicos foram designados.

- Definir um extrator de características através de comparações entre vários métodos de extração.
- Selecionar um classificador de padrões de diferentes heurísticas através de avaliações de desempenho em termos de acerto.
- Analisar e discutir os resultados obtidos na execução dos processos de avaliação.

1.3 Produção Científica durante o mestrado

SILVA, R. D. C., NUNES, T. M., PINHEIRO, G. J. B., ALBUQUERQUE, V. H. C. CUSTOMIZAÇÃO DA TRANSFORMADA IMAGEM-FLORESTA PARA GRAFOS DENSOS UTILIZANDO ALGORITMO DE FLOYD-WARSHALL – UM ESTUDO INICIAL, In: **XXIII Congresso Brasileiro de Engenharia Biomédica (CBEB)**. 2012, Porto de Galinhas, Anais do XXIII Congresso Brasileiro de Engenharia Biomédica, Porto de Galinhas, 2012.

NUNES, T. M., SILVA, R. D. C., ALBUQUERQUE, V. H. C. DESEMPENHO PRELIMINAR DE UMA ABORDAGEM VETORIAL SOBRE MÉTODOS DE CONTORNOS ATIVOS, In: **XXIII Congresso Brasileiro de Engenharia Biomédica (CBEB)**. 2012, Porto de Galinhas, Anais do XXIII Congresso Brasileiro de Engenharia Biomédica, Porto de Galinhas, 2012.

A dissertação produziu os seguintes artigos:

SILVA, R. D. C., THÉ, G. A. P. Moment Invariant based Classification of objects from low-resolution Industrial Sensor Images, In: **11th Brazilian Congress (CBIC), on Computational Intelligence**, 2013, Porto de Galinhas, *Anais do 11th Brazilian Congress on Computational Intelligence*, Porto de Galinhas, 2013.

SILVA, R. D. C., THÉ, G. A. P. Comparison Between Hough Transform and Moment Invariant to the Classification of Objects from low-resolution Industrial Sensor Images, In: **XI Simpósio Brasileiro de Automação Inteligente (SBAI)**, 2013, Fortaleza, *Anais do XI Simpósio Brasileiro de Automação Inteligente*, Fortaleza, 2013.

SILVA, R. D. C., COELHO, D. N., THÉ, G. A. P. A Performance Analysis of Classifiers to Recognition of Objects from low-resolution Images Industrial Sensor, In: **XI Simpósio Brasileiro de Automação Inteligente (SBAI)**, 2013, Fortaleza, *Anais do XI Simpósio Brasileiro de Automação Inteligente*, Fortaleza, 2013.

Trabalhos submetidos e em desenvolvimento

SILVA, R. D. C., THÉ, G. A. P., MEDEIROS, F. N. S. Improved Independent Component Analysis for Rotation-Invariant Image Description, **Pattern Recognition Letters**, 2014. (Submetido)

SILVA, R. D. C., COELHO, D. N., THÉ, G. A. P. Comparison Between k-Nearest Neighbors, Neural Network – SOM and Optimum-Path Forest to Recognition of Objects using Image Analysis by Zernike Moments, **11th IEEE Latin American Robotics Symposium / 2nd Brazilian Robotics Symposium**, São Carlos, 2014. (Submetido)

SILVA, R. D. C., THÉ, G. A. P. Image Processing to Objects Classify Invariant to Rotation using a Resolution Low 3D Industrial Sensor, **The International Journal of Advanced Manufacturing Technology**, 2014. (Em desenvolvimento)

1.4 Organização da Dissertação

O restante desta dissertação está organizado conforme as descrições dos capítulos a seguir:

No Capítulo 2 são apresentados os extratores abordados nesta pesquisa. São eles: Momentos de Hu, Zernike, Legendre, Fourier-Mellin, Tchebichef, Bessel-Fourier e Gaussian-Hermite, Transformada de Hough e a Análise de Componentes Independentes.

O Capítulo 3 introduz as diferentes heurísticas de classificação. Este capítulo apresenta os seguintes classificadores: *Naive Bayes*, *k*-Vizinhos mais Próximos, Rede Neural Artificial-Perceptron Multi-Camadas e Máquina de Vetor de Suporte.

Capítulo 4 descreve a metodologia utilizada para pré-processamento, extração de características e classificação de objetos. O primeiro experimento é conduzido com vistas à discussão dos resultados das taxas de acertos de todos os extratores utilizando o classificador *k*-NN com distância euclidiana. O segundo por sua vez, utiliza o melhor extrator obtido no primeiro experimento, permite comparar e discutir as várias heurísticas de classificação.

O Capítulo 5 apresenta as conclusões do trabalho e sugestões de trabalhos futuros.

2. EXTRAÇÃO DE CARACTERÍSTICAS

Em visão computacional, o conjunto de dados é formado por imagens que, a princípio, possuem considerável quantidade de informação irrelevante, o que tornaria qualquer computação sobre elas mais custosa. Neste contexto, a extração de características, pode ser visto como um meio de encontrar um conjunto de vetores que representem uma observação, enquanto reduzem a dimensionalidade do conjunto original de características. Assim, após a extração de características, obtemos um novo conjunto de dados com dimensão menor do que o conjunto original, o que pode diminuir consideravelmente o custo computacional exigido no processo de reconhecimento (FARIAS, 2012).

Em problemas de classificação de padrões, é desejável extrair características que introduzam alta discriminação entre as possíveis classes dos dados de entrada, eliminando as características redundantes que não contribuem no processo de classificação. Entretanto, a diminuição da dimensão dos dados não deve comprometer o desempenho do sistema de classificação. Deste modo, o processo de extração de características deve ser dirigido a fim de que as características geradas possibilitem ao classificador generalizar o problema eficientemente e obter taxa de acerto elevada.

Em resumo, o objetivo principal da extração de características, é simplificar um vetores de características, que possam representar uma observação, sem entretanto diminuir o poder de discriminação entre as classes.

O objetivo deste capítulo é apresentar as técnicas de extração de características para a classificação de objetos representados por imagens extraídas do sensor 3D.

2.1 Momentos Invariantes

Momentos invariantes têm sido extensivamente utilizados na extração de características, em reconhecimento de padrões e classificação de objetos. Uma das propriedades mais importantes dos momentos é sua invariância a transformações afins, como por exemplo, invariância à rotação, translação e escala. Momentos são quantidades escalares utilizadas para caracterizar uma função e capturar suas características mais significativas (FLUSSER *et al.*, 2009). Do ponto de vista matemático, momentos são projeções de uma função em uma base polinomial.

2.1.1 Momentos de Hu

O conceito de momento foi inicialmente apresentado por Hu (1962), e tem sido amplamente utilizado no campo da análise de imagem e reconhecimento de padrões (RANI *et al.*, 2007; BELGHINI *et al.*, 2012; YANG *et al.*, 2012).

Hu (1962) introduziu um conjunto de sete funções não lineares, sete momentos que são invariantes a translação, escala e rotação, os quais são calculados com os momentos geométricos/regulares de uma imagem $M \times N$

$$m_{pq} = \sum_{x=1}^M x^p y^q I_{xy}, \quad (1)$$

onde m_{pq} é o momento de ordem $(p + q)$ de uma imagem I_{xy} .

Essa equação permite calcular o centro de massa de uma imagem, e de uma região, no caso de uma máscara binária.

A partir dos momentos regulares podemos definir algumas medidas importantes sobre os objetos de interesse, e que são úteis na identificação de diferentes formas. Assim, os momentos regulares de ordem 0 e 1 são usados para o cálculo do centro de massa do objeto através de

$$(\bar{x}, \bar{y}) = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right). \quad (2)$$

Momentos centrais μ_{pq} são momentos geométricos da imagem calculada com relação ao centro de massa (\bar{x}, \bar{y}) :

$$\mu_{pq} = \sum_{x=1}^M \sum_{y=1}^N (x - \bar{x})^p (y - \bar{y})^q I_{xy}. \quad (3)$$

Momentos centrais são invariantes à translação, para que sejam invariantes à escala, são utilizados os momentos normalizados η_{pq} definidos pela seguinte fórmula:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{pq}^\gamma}, \quad \text{onde: } \gamma = \frac{p+q}{2} + 1, \forall p+q \geq 2. \quad (4)$$

Assim, os sete momentos são calculados a partir dos momentos normalizados η_{pq} até a terceira ordem, com as seguintes fórmulas:

$$M1 = \eta_{20} + \eta_{02}, \quad (5)$$

$$M2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2, \quad (6)$$

$$M3 = (\eta_{30} - 3\eta_{12})^2 + 3(\eta_{03} - 3\eta_{21})^2, \quad (7)$$

$$M4 = (\eta_{30} + \eta_{12})^2 + (\eta_{03} + \eta_{21})^2, \quad (8)$$

$$M5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{03} + \eta_{21})^2] + (3\eta_{21} - \eta_{03})(\eta_{03} + \eta_{21})[3(\eta_{30} + \eta_{12})^2 - (\eta_{03} + \eta_{21})^2], \quad (9)$$

$$M6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - 7(\eta_{03} + \eta_{21})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{03} + \eta_{21}), \quad (10)$$

$$M7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{03} + \eta_{21})^2] + (\eta_{30} - 3\eta_{12})(\eta_{03} + \eta_{21})[3(\eta_{30} + \eta_{12})^2 - (\eta_{03} + \eta_{21})^2]. \quad (11)$$

O inconveniente dos momentos regulares é que eles não são ortogonais, portanto, os momentos geométricos sofrem com o alto grau de redundância de informação, e os de ordem superior são sensíveis ao ruído (FU *et al.* 2007; SRIDHAR *et al.*, 2012).

Assim, Teague (1980), baseado na teoria de polinômios ortogonais contínuos, introduziu inicialmente os momentos de Zernike e Legendre. Posteriormente, Sheng (1994) introduziu os momentos ortogonais de Fourier-Mellin e mais recentemente, Mukundan (2001) mostrou os momentos de Tchebichef.

2.1.2 Momentos de Zernike

Momentos de Zernike são o mapeamento de uma imagem em um conjunto de polinômios de Zernike complexos (TEAGUE, 1980). Como estes polinômios de Zernike são ortogonais entre si, os momentos de Zernike podem representar as propriedades de uma imagem sem redundância ou sobreposição de informações entre os momentos (KHOTANZA *et al.*, 1990). Devido a estas características, os momentos de Zernike têm sido utilizados como recurso definitivo em aplicações como reconhecimento de padrões (KIM *et al.*, 1994; HSE *et al.*, 2004; QADER *et al.*, 2007; VOROBYOV, 2011), recuperação de imagens baseada em conteúdo (KIM *et al.*, 1998), classificação de objetos (ARIF *et al.*, 2009), reconstrução de

alfabeto (TRIPATHY, 2010) e outros sistemas de análise de imagem (WANG *et al.*, 1998; KIM *et al.*, 1999; SIT *et al.*, 2013).

Para calcular os momentos de Zernike, a imagem (ou região de interesse) é inicialmente mapeada em um círculo unitário, onde o centro da imagem é a origem do círculo. Os *pixels* que estão fora do círculo não são utilizados no cálculo. As coordenadas são então descritas pelo tamanho do vetor desde a origem ao ponto de coordenada.

Assim, o processo para calcular os momentos de Zernike para uma imagem consiste de três passos: cálculo do polinômio radial, cálculo das funções de base de Zernike e cálculo dos momentos de Zernike projetando a imagem para as funções de base (HWANG *et al.*, 2006).

A obtenção dos momentos de Zernike de uma imagem inicia-se com o cálculo dos polinômios radiais de Zernike $R_{nm}(\rho)$,

$$R_{nm}(\rho) = \sum_{s=0}^{\frac{n-|m|}{2}} c(n, m, s) \rho^{n-2s}, \quad (12)$$

onde

$$c(n, m, s) = (-1)^s \frac{(n-s)!}{s! \left(\frac{n+|m|}{2-s}\right)! \left(\frac{n-|m|}{2-s}\right)!}, \quad (13)$$

ρ é o comprimento do vetor desde a origem ao *pixel* (x, y)

$$\rho_{xy} = \frac{\sqrt{(2x - N + 1)^2 + (N - 1 - 2y)^2}}{N}, \quad (14)$$

e n e m são geralmente chamados ordem e repetição, respectivamente. A ordem n é um número inteiro não negativo, e a repetição m é um inteiro satisfazendo $0 \leq |m| \leq n$. A Figura 2 mostra os polinômios radiais de ordem zero a quinta ordem no intervalo $0 \leq \rho \leq 1$.

Para as funções de base, Zernike, (TEAGUE, 1980), introduziu um conjunto de polinômios complexos que formam um conjunto ortogonal completo sobre o interior de um círculo unitário, $x^2 + y^2 = 1$,

$$V_{nm}(x, y) = V_{nm}(\rho, \theta) = R_{nm}(\rho) \exp(jm\theta), \quad (15)$$

onde $j = \sqrt{-1}$, e θ é o ângulo entre o vetor ρ e o eixo x no sentido anti-horário

$$\theta_{xy} = \tan^{-1} \frac{N - 1 - 2y}{2x - N + 1}, \quad (16)$$

A ortogonalidade implica nenhuma redundância ou sobreposição de informações entre os momentos com diferentes ordens e repetições. Assim, o terceiro e último passo consiste no cálculo dos momentos de Zernike. A forma discreta dos momentos de Zernike de uma imagem de tamanho $N \times N$ é expressa como

$$Z_{nm} = \frac{n+1}{\lambda_N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} V_{nm}(x, y) I_{xy}, \quad (17)$$

onde $0 \leq \rho_{xy} \leq 1$ e λ_N é um fator de normalização, o qual deve ser o número de *pixels* localizado no interior do círculo unitário.

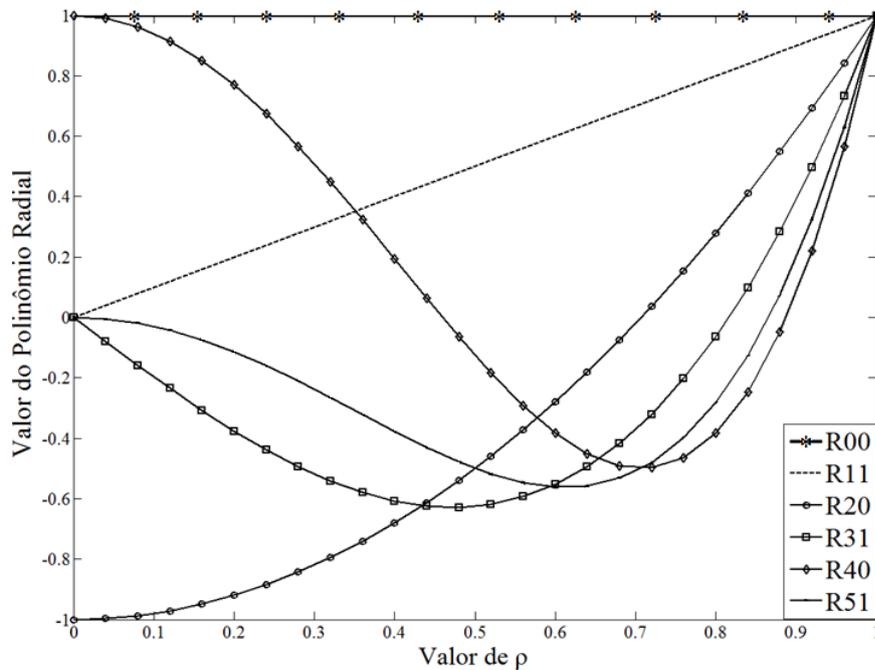


Figura 2 - Polinômio radial de Zernike de ordem 0-5 e baixas repetições.

2.1.3 Momentos de Legendre

Os momentos de Legendre foram introduzidos por Teague (1980), os quais são produzidos a partir da relação recursiva do polinômio de Legendre de ordem p que é definido como (CHONG *et al.*, 2004)

$$P_p(x) = \frac{(2p-1)xP_{p-1}(x) - (p-1)P_{p-2}(x)}{p}, \quad (18)$$

onde $P_0(x) = 1, P_1(x) = x$ e $p > 1$. Uma vez que a região de definição de polinômio de Legendre é o interior de $[-1, 1]$, uma imagem quadrada de $N \times N$ *pixels* com função intensidade $I_{ij}, 0 \leq x, y \leq (N - 1)$, é escalonada na região $-1 < x, y < 1$. A Figura 3 mostra o polinômio de Legendre de ordem zero até a quinta ordem definido no intervalo $[-1, 1]$.

A forma discreta dos momentos de Legendre de ordem $(p + q)$ pode ser expressa por:

$$L_{pq} = \lambda_{pq} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P_p(x_i) P_q(y_j) I_{ij} , \quad (19)$$

onde a constante de normalização é

$$\lambda_{pq} = \frac{(2p + 1)(2q + 1)}{N^2} , \quad (20)$$

x_i e y_j denotam as coordenadas dos *pixels* normalizados no intervalo $[-1, 1]$, o qual é dado por

$$x_i = \frac{2i}{N-1} - 1, \quad y_j = \frac{2j}{N-1} - 1. \quad (21)$$

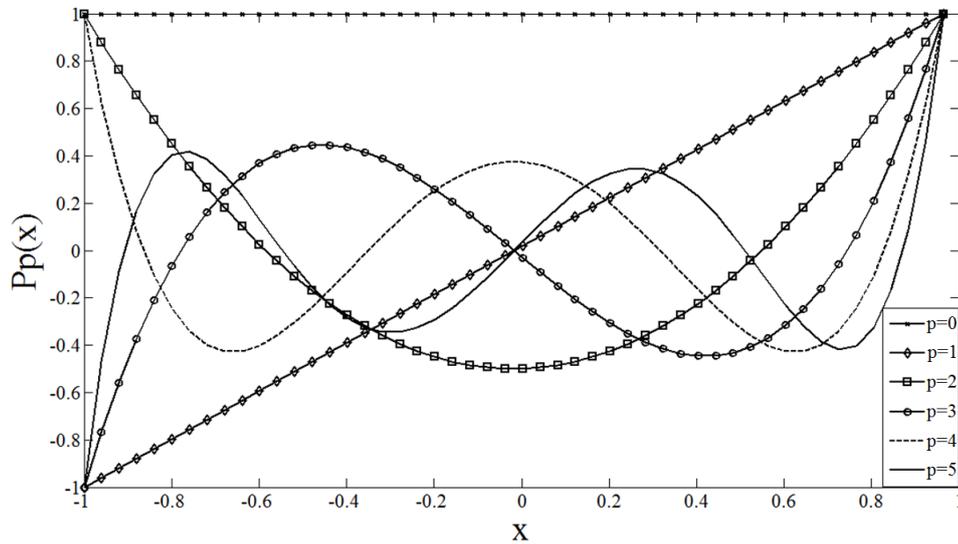


Figura 3 - Polinômio de Legendre $Pp(x)$ com $p = 0, 1, \dots, 5$.

2.1.4 Momentos de Fourier-Mellin

Os momentos ortogonais de Fourier-Mellin, introduzidos por Sheng (1994), pertencem a uma classe de momentos circularmente ortogonais. Estes momentos possuem características importantes, como a invariância a rotação, a redundância de informação mínima e robustez ao ruído de imagem.

Assim, estes momentos são baseados no conjunto de polinômios radiais onde os polinômios radiais de uma imagem I_{xy} são dados por (SINGH *et al.*, 2012)

$$Q_p(x, y) = \sum_{s=0}^p (-1)^{p+2} \frac{(p+s+1)! (x^2 + y^2)^{s/2}}{s! (p-s)! (s+1)!}, \quad (22)$$

onde p é um inteiro tal que $p \geq 0$ e $|q| \geq 0$. Estes polinômios também possuem formato polar bastando apenas substituir r por x, y onde $r = \sqrt{x^2 + y^2}$. A Figura 4 mostra os polinômios radiais $Q_p(r)$ para os valores de $p = 0, 1, \dots, 5$.

A função imagem é definida ao longo do domínio quadrado discreto de *pixels* $N \times N$, e $M_{pq}^*(x, y)$ são os conjugados complexos dos polinômios ortogonais $M_{pq}(x, y)$ dado por

$$M_{pq}(x, y) = Q_p(x, y) e^{jp\theta}, \quad (23)$$

onde $j = \sqrt{-1}$ e $\theta = \tan^{-1}\left(\frac{y}{x}\right)$, $\theta \in [0, 2\pi]$.

A forma discreta dos momentos normalizados de Fourier-Mellin é

$$O_{pq} = \frac{p+1}{\pi} \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} I(x_i, y_k) M_{pq}^*(x_i, y_k) \Delta x_i \Delta y_k, \quad (24)$$

onde $x_i^2 + y_k^2 \leq 1$,

$$x_i = \frac{2i+1-N}{D}, \quad y_k = \frac{2k+1-N}{D}, \quad (25)$$

$$\Delta x_i = \Delta y_k = \frac{2}{D}, \quad i, k = 0, 1, \dots, N-1, \quad (26)$$

e

$$D = \begin{cases} N, & \text{para círculo inscrito} \\ N\sqrt{2}, & \text{para círculo circunscrito} \end{cases}. \quad (27)$$

A escolha de D depende se os momentos devem ser calculados para imagem circular inscrita, $D = N$, ou para a região circular exterior, $D = N\sqrt{2}$. Quando $D = N\sqrt{2}$ são tomados, todos os *pixels* da imagem que fazem parte do cálculo dos momentos.

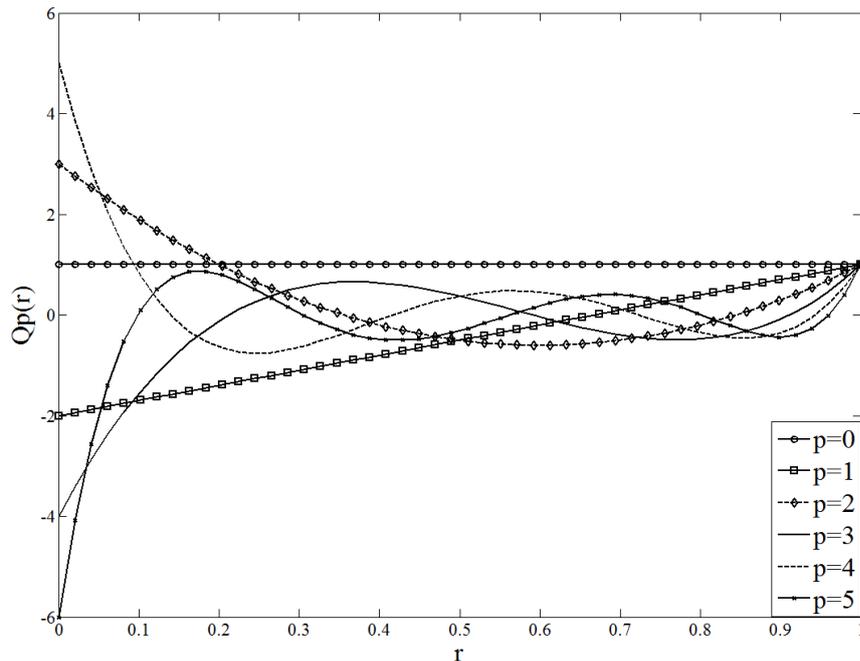


Figura 4 - Polinômio radial $Q_p(r)$ dos momentos de Fourier-Mellin com $p = 0, 1, \dots, 5$.

2.1.5 Momentos de Tchebichef

Os momentos e funções de momentos discutidos nas seções anteriores apresentam limitações como descritas a seguir. De fato, os polinômios de Zernike são definidos apenas dentro de um círculo unitário, os polinômios de Legendre são válidos apenas entre os intervalos $[-1, 1]$ e os momentos de Fourier-Mellin podem ser inscritos em um círculo ou no exterior deste círculo, porém, assumindo qualquer uma das duas formas, ocorre falta ou excesso de *pixels* que comprometem os momentos de Fourier-Mellin.

Desta forma, o cálculo desses momentos exige uma transformação de coordenadas e aproximações adequadas a partir dos momentos contínuos. Isto irá levar a um erro de discretização devido a aproximações numéricas de momentos contínuos (TIAGRAJAH *et al.*, 2011).

Para solucionar este problema, um conjunto de funções de momentos ortogonais discretos com base nos polinômios de Tchebichef foi introduzido por Mukundan (2001). A implementação dos momentos de Tchebichef não envolve qualquer aproximação numérica, uma vez que o seu conjunto de base é ortogonal no domínio discreto das coordenadas espaciais da imagem.

Os polinômios discretos de Tchebichef foram definidos por Erdelyi (1953) e baseado nestes polinômios, Mukundan (2001) definiu os polinômios escalados de Tchebichef como

$$t_p(x) = \frac{(2p-1)t_1(x)t_{p-1}(x) - (p-1)\left(1 - \frac{(p-1)^2}{N^2}\right)t_{p-2}(x)}{p}, \quad (28)$$

$$p = 2, 3, \dots, N-1,$$

onde $t_0(x) = 1$, $t_1(x) = \frac{(2p+1-N)}{N}$. A Figura 5 mostra os valores dos polinômios $t_p(x)$ para $p = 0, 1, \dots, 5$.

De acordo com a transformação citada anteriormente, na Eq. 28, a norma quadrada dos polinômios escalados é modificada de acordo com a fórmula:

$$\rho(p, N) = \frac{N\left(1 - \frac{1}{N^2}\right)\left(1 - \frac{2^2}{N^2}\right)\dots\left(1 - \frac{p^2}{N^2}\right)}{2p+1}, \quad (29)$$

$$p = 0, 1, \dots, N-1.$$

Então, os momentos de Tchebichef são definidos como:

$$T_{pq} = \frac{1}{\rho(p, N)\rho(q, N)} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} t_p(x)t_q(y)I_{xy}, \quad (30)$$

$$x, y = 0, 1, \dots, N-1.$$

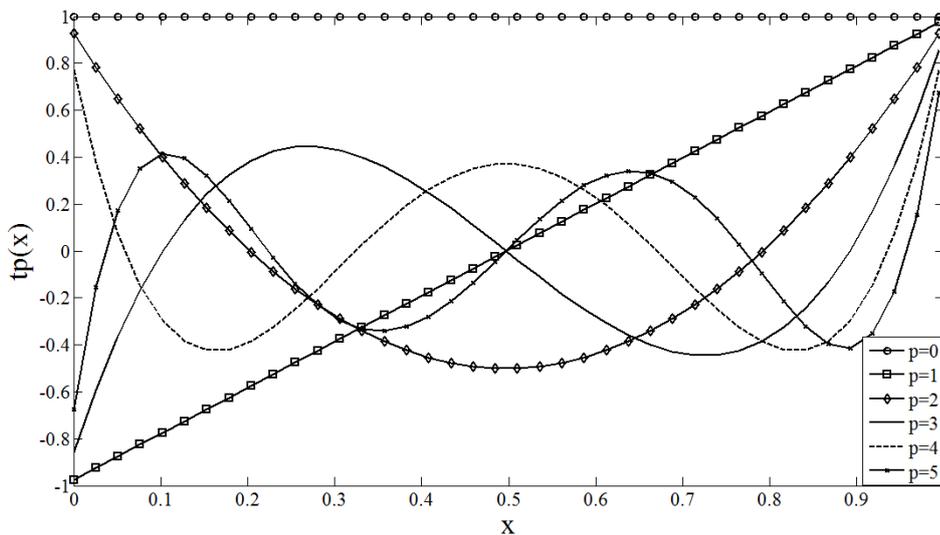


Figura 5 - Polinômios escalados de Tchebichef para $N = 40$.

Como descrito anteriormente, a implementação dos momentos de Tchebichef não envolve qualquer aproximação numérica.

Os momentos de Legendre, Tchebichef, e outros similares momentos ortogonais discretos, tais como os momentos de Krawtchouk (YAP *et al.*, 2003), dual Hahn (ZHU *et al.*, 2007a) e Racah (ZHU *et al.*, 2007b), caem na mesma classe de momentos ortogonais definidos no espaço de coordenadas cartesianas, onde momentos invariantes, particularmente invariantes à rotação, não estão prontamente disponíveis. No entanto, os momentos de Zernike e Fourier Mellin podem ser definidos em coordenadas polares, de modo que a rotação da imagem não altera a magnitude dos seus momentos.

Recentemente, um novo conjunto de momentos ortogonais definidos em coordenadas polares foi apresentado por Xiao (2010), os momentos de Bessel-Fourier, os quais também discute a invariância a rotação. Além deste, mais recentemente ainda, Yang (2011) reportou um sistemático e completo estudo relativo aos momentos de Gaussian-Hermite, suas implementações discretas e formulações.

2.1.6 Momentos de Bessel-Fourier

Os momentos de Bessel-Fourier são um conjunto de momentos baseados na função de Bessel de primeira ordem. Os momentos de Bessel-Fourier podem ser expressos como segue

$$B_{nm} = \frac{1}{2\pi a_n} \sum_k b_{nk} C_{pq}, \quad (31)$$

ond, $a_n = \frac{[J_{\nu+1}(\lambda_n)]^2}{2}$ é a constante de normalização, b_{nk} é

$$b_{nk} = \frac{(-1)^k}{k! \Gamma(\nu + k + 1)} \left(\frac{\lambda_n}{2}\right)^{\nu+2k}, \quad (32)$$

onde ν , é uma constante real, $\Gamma(a)$ é a função gamma e λ_n são os 0's' da função de Bessel de primeira ordem (ABRAMOWITZ *et al.*, 1965; AMOS, 1986)

$$J_\nu(\lambda_n r) = \sum_{k=0}^{\infty} \frac{(-1)^k}{k! \Gamma(\nu + k + 1)} \left(\frac{\lambda_n r}{2}\right)^{\nu+2k}, \quad (33)$$

e, os momentos complexos são definidos por Abu-Mostafa (1984) como:

$$C_{pq} = \iint_{-\infty}^{\infty} f(x, y) (x + jy)^p (x - jy)^q dx dy, \quad (34)$$

onde

$$p = \frac{v + 2k - m}{2}, \quad q = \frac{v + 2k + m}{2}. \quad (35)$$

Os polinômios $J_1(\lambda_n r)$ no intervalo $0 \leq r \leq 1$ são dados na Figura 6.

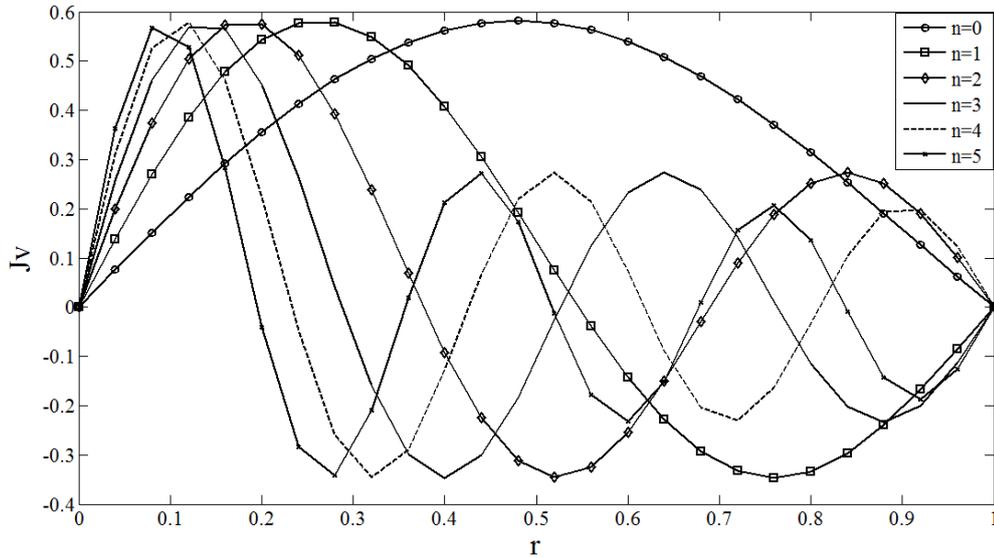


Figura 6 - Polinômio radial $J_1(\lambda_n x)$ dos momentos de Bessel-Fourier com $n = 0, 1, \dots, 5$.

2.1.7 Momentos de Gaussian-Hermite

Como outro tipo de momentos ortogonais contínuos, os momentos de Gaussian-Hermite foram inicialmente introduzidos por Shen (1997), contudo, as pesquisas para estes tipos de momentos são relativamente iniciais comparadas as demais pesquisas. Recentemente, Yang (2011b) voltou sua atenção para o desenvolvimento das invariâncias a rotação e translação dos momentos de Gaussian-Hermite e Yang (2011a), descreve um sistemático e completo estudo relativo a estes momentos.

Assim, os polinômios de Gaussian-Hermite com parâmetro de escala σ têm a seguinte definição:

$$\hat{H}_p(x; \sigma) = \frac{\exp\left(\frac{-x^2}{2\sigma^2}\right) H_p\left(\frac{x}{\sigma}\right)}{\sqrt{2^p p! \sqrt{\pi} \sigma}}, \quad (36)$$

onde

$$H_{p+1}(x) = 2xH_p(x) - 2pH_{p-1}(x), \quad (37)$$

for $p \geq 1$,

com condições iniciais $H_0(x) = 1$ e $H_1(x) = 2x$.

Logo, os momentos de Gaussian-Hermite, em forma de matriz, são então definidos como (YANG *et al.*, 2011a):

$$\mathbf{M} = \mathbf{H}\mathbf{I}\mathbf{H}^T. \quad (38)$$

Em que \mathbf{I} é uma matriz de imagem digital, e \mathbf{T} denota a operação de transposição de matriz.

A Figura 7 mostra algumas ordens dos polinômios de Gaussian-Hermite com valor de parâmetro de escala $\sigma = 0.1$.

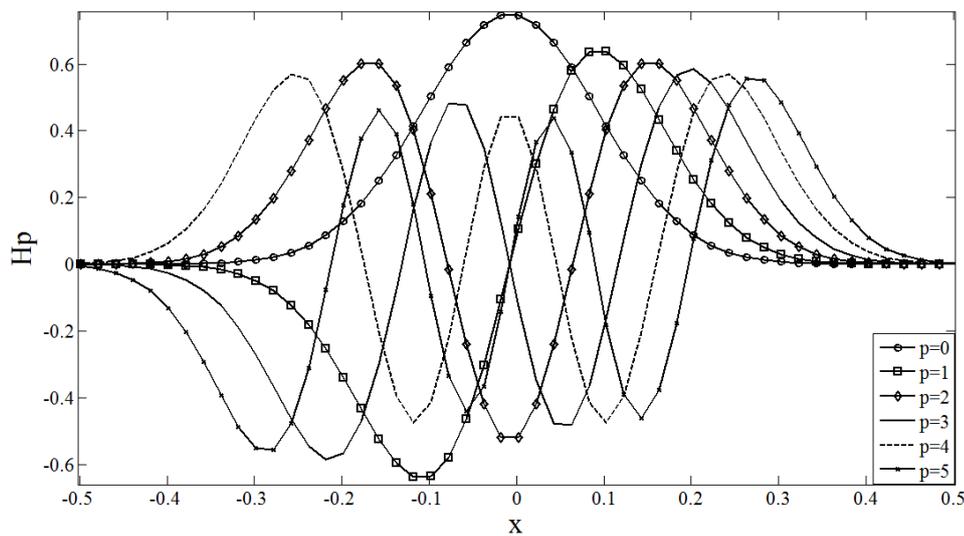


Figura 7 - Polinômios de Gaussian-Hermite de graus $p = 0, 1, \dots, 5$.

Como já mencionado, momentos e funções de momentos, devido à sua capacidade de representar os recursos globais de uma imagem, têm encontrado amplas aplicações nas áreas de processamento de imagens e reconhecimento de padrões. Contudo em aplicações onde as imagens são sujeitas a distorções e ruídos, os momentos têm encontrado dificuldades para a caracterização das imagens (YANG *et al.*, 2011b).

Outra abordagem para análise de imagem é a transformada de Hough, que é uma técnica elegante e versátil (HAULE *et al.*, 1989).

2.2 Transformada de Hough

A Transformada de Hough (HT) é uma técnica para detectar características de uma forma particular, que pode ser parametrizada como segmentos de retas e círculos em imagens binárias. Foi proposta por Hough em 1959 e modificada por ele em 1962, e é considerada uma técnica clássica de visão computacional. A HT é amplamente utilizada para

detectar a posição e orientação dos segmentos de linha reta em uma área da imagem de interesse. É uma transformação a partir do espaço de imagem para outro espaço de parâmetros, conhecido como o espaço de Hough com o intuito de detectar linhas retas (INRAWONG, 2012). A ideia é considerar as características da linha reta não como pontos da imagem, mas em termos de seus parâmetros. A intenção é reduzir cada linha para um ponto no espaço de parâmetros facilitando o processo para fins de detecção.

A funcionalidade básica e inicial da HT é detectar linhas retas. A equação de uma linha reta é dada pela equação (HAULE *et al.*, 1989; INRAWONG, 2012):

$$y = m * x + b , \quad (39)$$

onde (x, y) são coordenadas de pontos no espaço da imagem e (m, b) são dois parâmetros, o declive e a respectiva interseção y .

Devido ao fato das linhas, perpendiculares ao eixo x , poderem dar valores ilimitados para os parâmetros m e b , Duda e Hart (1972) parametrizaram as linhas em termos de θ e r tais que:

$$r = x * \cos(\theta) + y * \sin(\theta) , \quad (40)$$

em que r é o comprimento do vetor e $\theta \in [0, \pi]$ é o ângulo formado. Assim, dado x e y , cada linha que passa através do ponto (x, y) pode ser unicamente representada por (θ, r) . Ambos θ e r têm tamanhos finitos. A Figura 8 mostra um objeto (caixa) e sua correspondente representação no espaço de Hough.

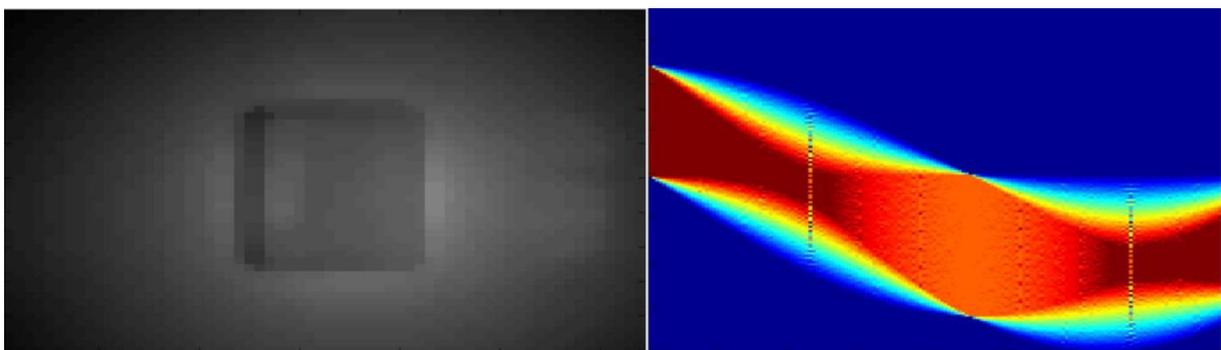


Figura 8 - (a) Imagem original e (b) seu correspondente espaço de Hough.

Quando a HT é utilizada como extrator de características, a definição do vetor característico que representa a imagem consiste em tomar características do espaço de Hough. Contudo, a tomada destas características a partir de um espaço dimensional elevado pode diminuir significativamente a eficiência do desempenho do sistema. Análise por Componentes

Principais (PCA) é amplamente utilizada para reduzir a dimensionalidade destes dados. A PCA (HOTELLING, 1933) é uma técnica matemática que utiliza uma transformação ortogonal para converter um conjunto de observações de dados possivelmente correlacionados em um conjunto de variáveis descorrelacionadas, chamadas de componentes principais (INRAWONG, 2012) onde cada um possui uma variância menor que a do anterior preservando coletivamente a variância total dos dados originais (HAROON *et al.*, 2009).

Seu objetivo consiste em reduzir a dimensão dos dados enquanto mantém a variação presente no conjunto de dados original. PCA permite calcular uma transformação linear de dados mapeados em um espaço dimensional elevado para um espaço dimensional mais baixo (LIU *et al.*, 2009), assim, utilizando a PCA é possível reduzir a dimensão do espaço de Hough.

PCA tem base nas estatísticas de segunda ordem (ZHANG *et al.*, 2006). Pode descorrelacionar os dados de entrada, mas não resolve as dependências de alta ordem (NAYAK *et al.*, 2006). Mas, em processamento de imagem, grande parte das informações importantes pode estar contida nas relações de ordem elevada (YANG, 2002).

Outro método de transformação de dados é Análise de Componentes Independentes. Atualmente, avanços significativos foram alcançados em termos de eficiência de algoritmos e em uma gama de aplicações onde a ICA pode ser usado. O interesse sobre esta técnica tem aumentado significativamente em áreas como sistemas de energia (LIMA *et al.*, 2012), visão computacional (PAN *et al.*, 2013), reconhecimento de face (SANCHETTA *et al.*, 2013), neuroimagem (TONG *et al.*, 2013), neurocomputação (ROJAS *et al.*, 2013), processamento de sinais biomédicos (SINDHUMOL *et al.*, 2013), estatística computacional (CHATTOPADHYAY *et al.*, 2013), modelagem econômica (LIN *et al.*, 2013).

2.3 Análise de Componentes Independentes

A Análise de Componentes Independentes (ICA) é uma técnica matemática que revela fatores que estão por trás de um conjunto de variáveis aleatórias que são assumidas não-gaussianas e mutuamente estatisticamente independentes; em outras palavras, é uma técnica de processamento de sinal estatístico cujo objetivo é decompor um vetor aleatório de forma linear em componentes que não são apenas descorrelacionadas (como na PCA) mas também o mais independentes possível (FAN *et al.*, 2002). Assim, ICA pode ser considerada uma generalização da análise de componentes principais (PCA). A PCA tenta obter uma representação das entradas a partir de variáveis não correlacionadas, enquanto a ICA fornece

uma representação com base em variáveis estatisticamente independentes (DÉNIZ *et al.*, 2003).

Para definir rigorosamente ICA (HYVÄRINEN *et al.*, 2000), dado um conjunto de observações de variáveis aleatórias $\{x_1(t), x_2(t), \dots, x_n(t)\}$, onde t é o tempo ou índice de amostras, assumimos que estas são geradas como uma mistura linear de componentes independentes $\{s_1(t), s_2(t), \dots, s_n(t)\}$:

$$\mathbf{x} = [x_1(t), x_2(t), \dots, x_n(t)]^T$$

$$\mathbf{x} = \mathbf{A}[s_1(t), s_2(t), \dots, s_n(t)]^T$$

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \quad (41)$$

em que \mathbf{A} é uma matriz de mistura desconhecida, $\mathbf{A} \in \mathbf{R}^{n \times n}$ (HUANG *et al.*, 2005). O modelo da ICA, equação (41), é dito ser um modelo generativo, o que significa que ele descreve como os dados observados são gerados por um processo de mistura das componentes independentes \mathbf{s} (ICs). As ICs são variáveis latentes, o que significa que elas não podem ser observadas diretamente. O problema clássico da ICA é o de estimar \mathbf{A} e \mathbf{s} , quando apenas \mathbf{x} é observado, desde que se obtenham observações que sejam independentes, de modo que \mathbf{A} seja inversível.

O problema dado pela equação (41) pode ser então reformulado, depois de estimar a matriz \mathbf{A} , como:

$$\mathbf{s} = \mathbf{A}^{-1} \mathbf{x} = \mathbf{W}\mathbf{x}, \quad (42)$$

de tal modo que uma combinação linear $\hat{\mathbf{s}} = \widehat{\mathbf{W}}\mathbf{x}$ é a estimativa otimizada dos sinais de fontes independentes \mathbf{s} .

Sob a hipótese de independência estatística dos componentes, cada um dos quais se caracterizam por uma distribuição não-gaussiana ou no máximo um que seja gaussiana, o problema básico da ICA dadas pelas equações (41) e (42) pode ser resolvido através da maximização da independência estatística das estimativas $\hat{\mathbf{s}}$ (BIZON *et al.*, 2013).

No processo de encontrar tal matriz $\widehat{\mathbf{W}}$, algumas técnicas de pré-processamento são úteis para tornar mais fácil seu cálculo. Há duas etapas de pré-processamento bastante utilizadas na ICA. Em primeiro lugar, a média dos dados é geralmente subtraída para centrar os dados sobre a origem, em outras palavras:

$$\tilde{\mathbf{x}} = \mathbf{x} - E\{\mathbf{x}\}. \quad (43)$$

O segundo passo é o branqueamento dos dados, isto significa transformar os dados de modo que os componentes não sejam mais correlacionados e tenham variância unitária

$$\mathbf{z} = \mathbf{V}\tilde{\mathbf{x}}, \quad (44)$$

onde \mathbf{V} é a matriz de branqueamento e \mathbf{z} os dados branqueados.

Aplicações da ICA para o reconhecimento de imagens rotacionadas, requer que as variáveis aleatórias sejam as imagens de treinamento. Dados x_i ser uma imagem vetorizada, podemos construir um conjunto de imagens de treinamento $\{x_1, x_2, \dots, x_n\}$ com n variáveis aleatórias que são assumidas como combinações lineares de m desconhecidas componentes independentes s , denotados por s_1, s_2, \dots, s_m , convertidas em vetores e denotados por $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ e $\mathbf{s} = [s_1, s_2, \dots, s_m]^T$. A partir desta relação, cada imagem x_i é representada por uma combinação linear de s_1, s_2, \dots, s_m , com pesos $\mathbf{s} = [s_1, s_2, \dots, s_m]^T$ respectivos da matriz \mathbf{A} . Quando a ICA é utilizada para a extração de parâmetros de imagens, as colunas de $\mathbf{A}_{\text{treinamento}}$ são as características das imagens, e os coeficientes $\mathbf{s}_{\text{treinamento}}$, sinalizam a presença e a amplitude da i -ésima característica nos dados observados $\mathbf{x}_{\text{treinamento}}$. Portanto, a matriz de mistura $\mathbf{A}_{\text{treinamento}}$ pode ser considerada como os vetores característicos que representam as características de todas as imagens de treinamento (YUEN *et al.*, 2002), e, assim, para achar as características $\mathbf{A}_{\text{teste}}$ das imagens $\mathbf{x}_{\text{teste}}$, este deve ser multiplicado pelo vetor $\mathbf{s}_{\text{treinamento}}$ como

$$\mathbf{A}_{\text{teste}} = \mathbf{x}_{\text{teste}} \mathbf{s}_{\text{treinamento}}^{-1}. \quad (45)$$

Finalmente, estas matrizes contêm os vetores característicos representativos das imagens e devem ser apresentadas ao classificador, como mostra a Figura 9.

Existem muitos algoritmos que realizam ou executam a ICA como FastICA (Hyvärinen, 1999) (Hyvärinen *et al.*, 2001), Jade (Cardoso, 1989), ProDenICA (Hastie e Tibshirani, 2003), Infomax (Bell e Sejnowski, 1995), *KernelICA* (Bach e Jordan, 2002). Para o cálculo da ICA neste trabalho, FastICA foi escolhido porque é um eficiente e popular algoritmo que permite uma fácil modificação e manutenção.

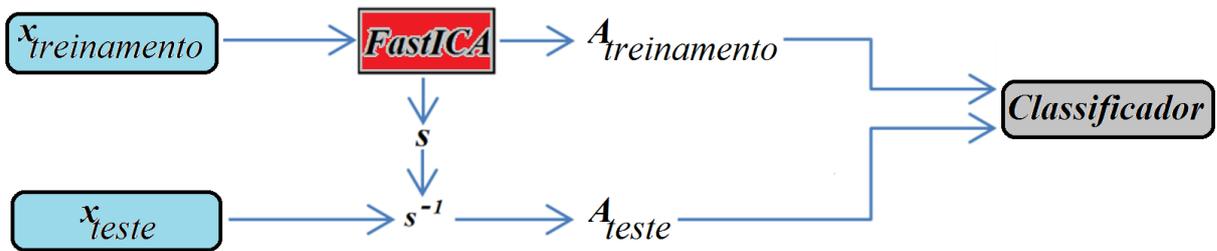


Figura 9 - Passos para o processo de classificação.

2.4 Considerações Finais deste Capítulo

Este capítulo apresentou os extratores de características, baseados nos momentos de Hu, Zernike, Legendre, Fourier-Mellin, Tchebichef, Bessel-Fourier e Gaussian-Hermite, foram abordadas também a Transformada de Hough e Análise de Componentes Independentes, bem como os passos necessários para as suas utilizações na extração de características em imagens invariantes à rotação.

O próximo capítulo descreverá os métodos de classificação *Naive Bayes*, *k*-Vizinhos mais Próximo, Máquina de Vetor de Suporte e Rede Neural Artificial – Perceptron Multi-Camadas.

3. CLASSIFICADORES

A classificação é a fase final de todo o sistema de processamento de imagem em que cada padrão desconhecido é atribuído a uma categoria. O grau de dificuldade do problema de classificação depende da variabilidade dos valores característicos dos objetos da mesma classe, em relação à diferença entre os valores característicos para objetos de diferentes classes (Mercimek, 2005).

Um sistema de classificação pode ser projetado de três formas distintas. Podemos utilizar um conjunto de observações com o objetivo de estabelecer a existência de classes ou *clusters* nos dados, baseando-se no princípio de que o algoritmo é capaz de identificar por si só as classes. Outro método é admitir conhecida a classe que gerou cada padrão no conjunto de dados. E, um último, é hibridizar os dois métodos quando normalmente as amostras rotuladas são difíceis de serem obtidas, porém as sem rótulo são abundantes e de fácil coleta. O primeiro tipo é conhecido como aprendizagem não-supervisionada (ou *Clustering*); o segundo é a aprendizagem supervisionada e o último, semi-supervisionada. Neste trabalho é feita classificação por aprendizagem supervisionada, possibilitando avaliar a qualidade do classificador.

Assim, vamos supor que temos um problema de classificação em que há M possíveis classes e há N amostras independentes e identicamente distribuídas $Z = \{(X_1, \theta(X_1)), (X_2, \theta(X_2)), \dots, (X_N, \theta(X_N))\}$, onde X_i é um vetor no espaço característico e θ corresponde à classe à qual a amostra pertence. O problema de classificação supervisionada consiste em utilizar esse conhecimento prévio para classificar novas amostras X_S a uma das M possíveis classes de uma forma que minimiza o erro de classificação (SOUZA *et al.*, 2012).

Assim, a aprendizagem supervisionada requer uma fase inicial denominada fase de treinamento, de modo que, nesta fase, são apresentados padrões de treinamento, que são definidos com o rótulo da classe a quais pertencem. O resultado da fase de treinamento é um conjunto de regras que exprimem os relacionamentos entre os atributos dos padrões de treinamento de modo a permitir a classificação de novos padrões nas classes existentes (HAYKIN, 2001).

Após o treinamento, dá-se a fase de teste. Nela são apresentados ao classificador o conjunto de regras, protótipos ou assinaturas, obtidos na fase anterior, e outros padrões, diferentes dos padrões usados para o treinamento, que também possuam rótulo das classes a quais pertencem. A finalidade disto é avaliar a consistência das regras, protótipos ou

assinaturas vindas da fase de treinamento, assim, esses novos padrões, chamados de padrões de teste, são classificados sem que a informação do rótulo de classe que carregam seja levada em consideração. Depois disso, os padrões de teste terão dois rótulos de classe, o rótulo inicial que informa precisamente a classe à qual eles pertencem e o rótulo calculado pelo classificador. Isto nos possibilita contar os erros e acertos do classificador, permitindo aferir sua precisão.

Com a disponibilidade e avanço computacional, o projeto e a utilização de métodos de classificação distintos tornaram-se práticos. Para várias aplicações, não existe somente uma abordagem para classificação, e, por isso, torna-se bastante necessária à comparação entre estes.

3.1 Classificador Bayesiano Linear

Classificadores bayesianos ou procedimentos de testes pelas hipóteses de Bayes são baseados na teoria de probabilidade conhecida como regra de Bayes, e a abordagem fundamental para o problema de classificação é a teoria de decisão de Bayes. O princípio da regra de decisão de Bayes é escolher a opção de menor risco. Suponha que existam amostras $\mathbf{X} = [X_1, X_2, \dots, X_N]$ onde N é o número de amostras que devem ser classificadas para as classes $\mathbf{Y} = [Y_1, Y_2, \dots, Y_M]$ e M é o número de classes. O vetor característico das amostras $X_i (1 \leq i \leq N)$ é denotado como $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T$, onde n é a dimensão do vetor \mathbf{x}_i . A probabilidade que uma amostra X_i com vetor característico \mathbf{x}_i pertença à classe $Y_j (1 \leq j \leq M)$ é $P(Y_j|\mathbf{x}_i)$, e é referida muitas vezes como uma probabilidade posterior. A classificação da amostra X_i com vetor característico \mathbf{x}_i é feita de acordo com as probabilidades posteriores. Pela regra de Bayes, a probabilidade posterior pode ser escrita como:

$$P(Y_j|\mathbf{x}_i) = \frac{P(\mathbf{x}_i|Y_j)P(Y_j)}{P(\mathbf{x}_i)}, \quad (46)$$

onde $P(\mathbf{x}_i|Y_j)$ é a função densidade de probabilidade de \mathbf{x}_i condicionada à classe Y_j no espaço e descreve o modelo de distribuição dos dados da classe Y_j . $P(Y_j)$ é a probabilidade a priori, da classe Y_j que descreve a probabilidade da classe Y_j antes da medição de todas as características (ZHAO *et al.*, 2013). Se as probabilidades a priori são desconhecidas, elas são muitas vezes estimadas pelas ocorrências relativas (JÚNIOR, 2004; KARCHER, 2009).

3.1.1. Classificador Naive Bayes

O classificador *Naive Bayes* é o mais simples classificador bayesiano e é um classificador probabilístico simplificado baseado na aplicação do teorema de Bayes, o qual possui a hipótese que todos os atributos são independentes (FRIEDMAN *et al.*, 1997; ZHAO *et al.*, 2013).

No classificador *Naive Bayes*, a probabilidade anterior $P(Y_j)$ pode ser calculada simplesmente por contagem do número de amostras da classe cujo rótulo é Y_j . Assim, o classificador introduz uma suposição de independência condicional na classe entre as características das amostras $\mathbf{X} = [X_1, X_2, \dots, X_N]$. O classificador Naive Bayes é obtido como se segue. Assumimos que a distribuição conjunta de classes e atributos pode ser escrita como:

$$P\left(\frac{X_i}{Y_j}\right)P(Y_j) = \prod_{i=1}^n P\left(\frac{X_i}{Y_j}\right)P(Y_j). \quad (47)$$

Este classificador classifica as amostras X_i como classe Y_j

$$Y_j = \arg \max \prod_{i=1}^n P(X_i|Y_j)P(Y_j). \quad (48)$$

A probabilidade $P(X_i|Y_j)$ é estimada por

$$P(X_i|Y_j) = \frac{Np + N_i^{Y_j}}{N + N^{Y_j}}, \quad (49)$$

onde p é a probabilidade anterior, N^{Y_j} é o número total de amostras de classe Y_j e $N_i^{Y_j}$ é o número de vezes que a amostra X_i com vetor característico $\mathbf{X} = [X_1, X_2, \dots, X_N]^T$ ocorre na classe Y_j (ZHAO *et al.*, 2013). O classificador *Naive Bayes* é conhecido por sua simplicidade e eficiência, pois apresenta estrutura fixa e parâmetros ajustáveis.

Os classificadores *Naive Bayes*, na presença de variáveis altamente correlacionadas (redundantes), podem ampliar desnecessariamente o peso da evidência destes atributos sobre a classe, o que pode prejudicar a assertividade das classificações. Outro problema que pode ocorrer nas aplicações com este classificador, é o superajuste (*overfitting*). Este problema é decorrente do grande número de parâmetros que a rede bayesiana construída pode apresentar o que pode degradar o desempenho do classificador (KARCHER, 2009).

3.2 *k*-Vizinhos Mais Próximos

O método *k*-vizinhos mais próximos (*k*-NN) é considerado um dos métodos mais antigos, simples e conhecidos para reconhecimento de padrões supervisionado. A primeira análise de uma regra de decisão do tipo vizinho mais próximo foi feita em uma série de dois artigos de Fix e Hodges (1951) e, Fix e Hodges (1952) para $k \rightarrow \infty$, posteriormente investigado por Cover e Hart (1967) para valores fixos de *k*, até ser estabelecido por Patrick e Fischer (1970) um classificador generalizado para múltiplas classes.

O *k*-NN é um classificador onde o aprendizado é baseado na analogia (MITCHEL, 1997). O conjunto de treinamento é formado por vetores *n*-dimensionais e cada elemento deste conjunto representa um ponto no espaço *n*-dimensional. Para determinar a classe de um elemento que não pertença ao conjunto de treinamento, o classificador *k*-NN procura *k* elementos do conjunto de treinamento que estejam mais próximos deste elemento desconhecido, ou seja, que tenham a menor distância. Estes *k* elementos são chamados de *k*-vizinhos mais próximos (SILVA, 2005). Verifica-se quais são as classes desses *k* vizinhos e a classe mais frequente será atribuída à classe do elemento desconhecido.

Considere um padrão de teste desconhecido *x*. Em geral, as seguintes etapas são executadas para algoritmo *k*-NN:

1. Escolha do valor de *k*: o valor *k* é completamente definido pelo usuários. Geralmente depois de algumas tentativas, o valor de *k* é escolhido de acordo com os resultados obtidos.
2. Cálculo da distância: Qualquer medida de distância pode ser utilizada para esta etapa.
3. Classificar as distâncias obtidas em ordem crescente: o valor escolhido de *k* também é importante nesta etapa. As distâncias encontradas são classificadas em ordem crescente e *k* distâncias mínimas são tomadas.
4. A classificação dos vizinhos mais próximos: as classes dos *k* vizinhos mais próximos são identificadas.

Existem várias formas de medir a distância entre os conjuntos de classes diferentes no espaço de características. Dentre elas, pode-se citar (WEBB, 2011):

Distância Euclidiana

Vamos considerar os casos de duas variáveis de entrada, uma vez que é fácil representar no espaço bidimensional. A distância entre estes dois pontos é calculada como a diferença do comprimento dos pontos. É denotada por:

$$d_e = \left\{ \sum_{i=1}^p (x_i - y_i)^2 \right\}^{\frac{1}{2}}. \quad (50)$$

Distância City Block (Manhattan)

A distância *City Block* entre dois pontos, x e y , é calculada como:

$$d_{cb} = \sum_{i=1}^p |x_i - y_i|. \quad (51)$$

Distância Cosseno

A distância Cosseno entre dois pontos é:

$$d_{cos} = 1 - \frac{\sum_{i=1}^p x_i y_i}{\left[\sum_{i=1}^p x_i^2 \sum_{i=1}^p y_i^2 \right]^{1/2}}. \quad (52)$$

Distância Correlação

A distância correlação entre dois pontos:

$$d_{corr} = 1 - \frac{\sum_{i=1}^p (x_i - \bar{x}_i)(y_i - \bar{y}_i)}{\left[\sum_{i=1}^p (x_i - \bar{x}_i)^2 \sum_{i=1}^p (y_i - \bar{y}_i)^2 \right]^{1/2}}, \quad (53)$$

onde \bar{x}_i e \bar{y}_i é a média de x_i e y_i respectivamente.

3.3 Máquina de Vetor de Suporte

Máquina de vetor de suporte (SVM) é uma técnica para classificação e regressão. Foi proposto inicialmente por Vapnik e Lerner (1963) como um classificador linear. Sua ideia é muito simples, ele mapeia os vetores padrões para um espaço característico de maior dimensão, onde um hiperplano de separação é melhor construído (o hiperplano de margem máxima). Boser *et al.* (1992) sugeriu um método para criação de um classificador não linear. A ideia principal consiste na construção de hiperplanos ótimos, ou seja, hiperplanos que maximizam a margem de separação das classes, com a finalidade de separar padrões de treinamento de diferentes classes, minimizando o número de erros no conjunto de

treinamento. A Figura 10a mostra a aplicação do SVM em um problema linearmente separável e a Figura 10b para um problema não linearmente separável.

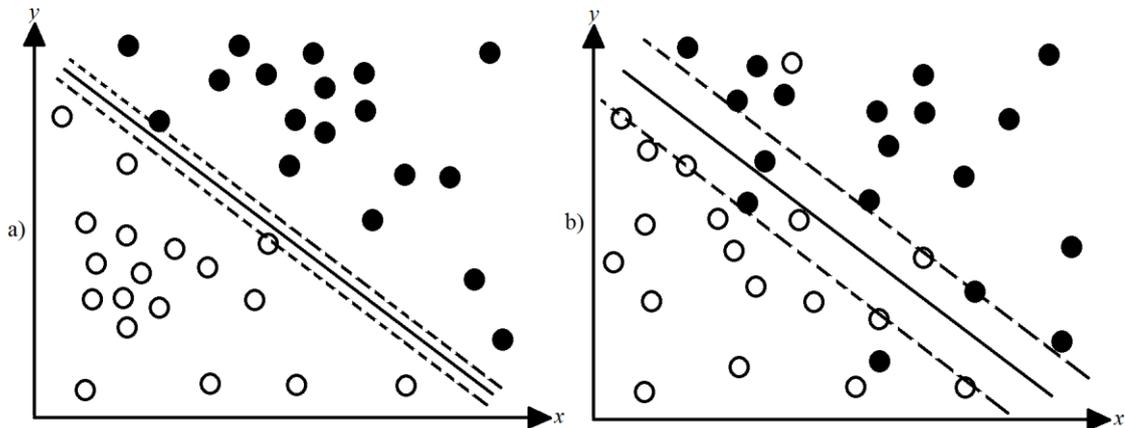


Figura 10 - (a) Problema linearmente separável. O espaço entre as linhas tracejadas é a margem de separação ótima, máxima. (b) Problema não linearmente separável. As linhas tracejadas, margem de separação ótima, são encontradas de modo a ser a maior margem com menor erro no conjunto de treinamento.

A classificação linear é frequentemente implementada pelo uso de uma função real $g(x)$ na seguinte forma: a entrada $\mathbf{x} = (x_1, \dots, x_n)^T$ para duas classes, w_1 e w_2 , com classe positiva, $y_i = 1$, e negativa caso contrário. A função discriminante linear é

$$g(x) = \mathbf{w}^T \mathbf{x} + \mathbf{w}_0, \quad (54)$$

com regra de decisão

$$\mathbf{w}^T \mathbf{x} + \mathbf{w}_0 \begin{cases} > 0 \\ < 0 \end{cases} \rightarrow x \in \begin{cases} w_1 \text{ com correspondente valor numérico } y_i = +1 \\ w_2 \text{ com correspondente valor numérico } y_i = -1 \end{cases}$$

podendo ser combinadas na seguinte inequação

$$y_i(\mathbf{w}^T \mathbf{x} + \mathbf{w}_0) > 0 \text{ para todo } i. \quad (55)$$

Normalmente em aplicações reais, os dados não são linearmente separáveis. Logo, o algoritmo de vetor de suporte pode ser aplicado em um espaço de características transformadas, $\phi(x)$, para alguma função ϕ não linear. Na verdade, este princípio constitui a base de muitos métodos de classificação de padrões: transformar as características de entrada não lineares para um espaço no qual métodos lineares possam ser aplicados, ou seja, fazer um mapeamento dos dados para um espaço onde os dados possam ser linearmente separáveis (WEBB *et al.*, 2011). Assim, a função discriminante é:

$$g(x) = \mathbf{w}^T \phi(x) + \mathbf{w}_0, \quad (56)$$

com regra de decisão

$$\mathbf{w}^T \phi(x) + \mathbf{w}_0 \begin{cases} > 0 \\ < 0 \end{cases} \rightarrow x \in \begin{cases} w_1 \text{ com correspondente valor numérico } y_i = +1 \\ w_2 \text{ com correspondente valor numérico } y_i = -1 \end{cases}.$$

A solução de margem máxima é feita através da maximização de uma lagrangiana. Assim, o problema dual pode ser formulado como:

$$L_D = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \phi^T(x_i) \phi(x_j), \quad (57)$$

onde $y_i = \pm 1$, $i = 1, \dots, n$, são valores do indicador de classe e $\alpha_i, i = 1, \dots, n$, são multiplicadores de Lagrange que satisfazem $0 \leq \alpha_i \leq C$, para um parâmetro de regularização C , e $\sum_{i=1}^n \alpha_i y_i = 0$.

Os vetores característicos transformados podem ser substituídos por uma função de *kernel*:

$$K(x, y) = \phi^T(x) \phi(y), \quad (58)$$

evitando assim o cálculo da transformação $\phi(x)$. Assim, a função discriminante torna-se

$$g(x) = \sum_{i \in SV} \alpha_i y_i K(x_i, x) + w_0, \quad (59)$$

em que SV é o conjunto de vetores de suporte que satisfazem $0 < \alpha_i < C$,

$$w_0 = y_i - \alpha_i y_i \phi^T(x_i) \phi(x_j). \quad (60)$$

Para a função de *kernel*, há muitos tipos que podem ser utilizados em uma SVM. A tabela abaixo lista algumas formas mais comumente utilizadas.

Tabela 1 : *Kernels* para Máquina de Vetor de Suporte.

| | Forma matemática $K(x, y)$ |
|---|---|
| Polinômio de grau d | $(1 + x * y)^d$ |
| Gaussiano (rbf) | $\exp\left(\frac{- x - y ^2}{2\sigma^2}\right)$ |
| Perceptron Multi-Camadas (mlp) | $\tanh(x * y - \theta)$ |

3.4 Rede Neural Artificial

Redes Neurais Artificiais (RNA) foram inicialmente idealizadas por McCulloch e Pitts (1943) que desenvolveu um estudo sobre o comportamento do neurônio biológico, com o objetivo de criar um modelo matemático para este. McCulloch e Pitts sugeriram como modelo semelhante para o processamento lógico, uma possível estrutura baseada na forma como os neurônios biológicos processam informação, e demonstrou que este modelo pode ser utilizado para criar qualquer expressão lógica finita (MICHIE *et al.*, 1994).

Desta forma, RNA são máquinas desenvolvidas para modelar a forma com que o cérebro executa determinadas tarefas. Além disso, uma RNA pode ser vista como um processador de distribuição paralela que tem uma propensão natural para acumular conhecimento e fazê-lo disponível para uso. Este método baseia-se em adquirir conhecimento, através de um processo de aprendizagem, e guardar as informações adquiridas a partir de interconexões, sinapses, entre os neurônios. O exemplo mais antigo de RNA são as redes Perceptron a qual se caracterizam por possuir apenas uma camada de saída conectada às entradas por conjuntos de pesos.

Minsky e Papert (1969) analisaram matematicamente o Perceptron e expuseram que redes de uma camada não são capazes de solucionar problemas que não sejam linearmente separáveis devido às restrições de representação. Como não acreditavam na possibilidade de se construir um método de treinamento para redes com mais de uma camada, eles concluíram que as redes neurais seriam sempre suscetíveis a essa limitação.

Contudo, o desenvolvimento do algoritmo de treinamento retropropagação do erro (*backpropagation*) por Rumelhart (1986), mostrou que é possível treinar eficientemente redes com camadas intermediárias, resultando no modelo de RNA mais utilizado atualmente, as redes Perceptron Multi-Camadas (MLP), e desde então tem sido utilizada extensivamente em várias aplicações de reconhecimento de padrões.

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 (HAYKIN, 2001). 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. Além disso, o ajuste de pesos utiliza o método da retropropagação do gradiente para definir as correções a serem aplicadas.

$$w_{ji}(n + 1) = w_{ji}(n) + \Delta w_{ji}(n) , \quad (61)$$

o neurônio j é um nó de saída, na iteração n .

Gradiente descendente pode ser definido por:

$$\Delta w_{ji}(n) = -\eta \frac{\partial \varepsilon(n)}{\partial w_{ji}(n)}, \quad (62)$$

onde η é a taxa de aprendizagem e $\varepsilon(n)$ é a soma instantânea dos erros quadráticos, na iteração n .

O algoritmo de treinamento *backpropagation*, consiste basicamente de dois passos:

- *Propagação positiva do sinal funcional*: durante este processo todos os pesos da rede são mantidos fixos, e
- *Retropropagação do erro*: durante este processo os pesos da rede são ajustados tendo por base uma medida de erro.

Assim, o sinal de erro é propagado em sentido oposto ao de propagação do sinal funcional, por isso o nome de retropropagação do erro.

A variação dos pesos sinápticos, é:

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta \delta_j(n) y_i(n), \quad (63)$$

onde $\delta_j(n)$ é o gradiente local, $y_i(n)$ é o sinal funcional que aparece no neurônio i , na iteração n , η é a taxa de aprendizagem da rede que tem como função escalar o gradiente do neurônio permitindo variações mais ou menos rápidas, e α é a constante de momento que modifica a taxa de aprendizado, alterando, assim, a instabilidade (HAYKIN, 2001).

O MLP básico produz uma transformação de um padrão $x \in R^p$ para um espaço n' -dimensional de acordo com:

$$g_j(n) = \sum_{i=0}^m w_{ji}(n) y_i(n), \quad (64)$$

onde m é o número total de entradas aplicados no neurônio j e $y_i(n)$ é uma função de ativação, sendo comumente utilizada a função logística:

$$y_i(n) = \phi_j(g_j(n)) = \frac{1}{1 + \exp(-a * g_j(n))}, \quad (65)$$

e tangente hiperbólica:

$$y_j(n) = a * \tanh(b * g_j(n)), \quad (66)$$

onde a e b são constantes positivas.

Assim, a transformação consiste em projetar os dados para cada uma das m orientações; em seguida, transformar os dados projetados pelas funções não lineares $y_i(n)$; e finalmente, formar uma combinação linear usando os pesos w_{ji} (WEBB *et al.*, 2011).

Uma rede MLP típica possui três características principais; os neurônios das camadas intermediárias possuem uma função de ativação não-linear do tipo sigmoideal, a rede possui uma ou mais camadas intermediárias e a rede possui um alto grau de conectividade.

3.5 Avaliação do Desempenho dos Classificadores

Um dos passos mais importantes em sistemas de reconhecimento de padrões é a avaliação do desempenho dos classificadores. Podemos obter números que indicam quais foram às performances obtidas pelos classificadores utilizados através de métodos de medição do erro e, juntamente com as taxas de acerto, podemos escolher o classificador ideal para uma dada aplicação. Como método de avaliação do desempenho dos classificadores, iremos apresentar a matriz de confusão.

3.5.1 Matriz de Confusão

Vários métodos de avaliação de acurácia têm sido discutidos na literatura. Os métodos mais utilizados, ainda hoje, são baseados na matriz de confusão ou de erros. Uma análise consistente do comportamento do classificador pode ser fornecida pela matriz de desempenho semi-global, matriz de confusão. Esta matriz fornece uma representação de desempenho quantitativo para cada classificador em termos de reconhecimento de classe. A matriz de confusão pode ser definida por:

$$A = \begin{bmatrix} RR_{11} & \cdots & RR_{1N} \\ \vdots & \ddots & \vdots \\ RR_{N1} & \cdots & RR_{NN} \end{bmatrix}, \quad (67)$$

onde RR_{ij} corresponde ao número total de entidades de classe C_i que foram classificados na classe C_j . Os elementos da diagonal principal indicam o número total de amostras na classe C_i reconhecidos corretamente pelo sistema. Pela matriz A , é possível calcular uma taxa de desempenho global para o classificador (FREITAS *et al.*, 2007)

$$RR^A = \frac{1}{N} \sum_{i,j=1}^N RR_{i,j}, \quad (68)$$

assim, pode-se construir a matriz e confusão como a Tabela 2,

Tabela 2 : Matriz de Confusão.

| | | Entrada | | | FPR |
|-------|----------|----------|----------|----------|------|
| | | Classe 1 | Classe 2 | Classe 3 | |
| Saída | Classe 1 | VP | FN2/FP1 | FP3/FP1 | FPR1 |
| | Classe 2 | FN1/FP2 | VP | FP3/FP2 | FPR2 |
| | Classe 3 | FN1/FP3 | FN2/FP3 | VP | FPR3 |
| FNR | | FNR1 | FNR2 | FNR3 | TPR |

onde:

VP (Verdadeiro Positivo) - O elemento de entrada é genuíno (positivo) e o classificador o classifica como positivo.

FP (Falso Positivo) - O elemento de entrada é impostor (negativo) e o classificador o classifica como positivo.

FN (Falso Negativo) – O elemento de entrada é genuíno (positivo) e o classificador o classifica como negativo.

FNR (Razão de Falsos Negativos – Falsa Rejeição) – Proporção de amostras genuínas erroneamente classificadas como impostoras.

FPR (Razão de Falsos Positivos – Falsa Aceitação) – Proporção de impostores erroneamente classificados como genuínos.

TPR (Proporção de Verdadeiros Positivos - Sensibilidade) – Proporção de genuínos de uma classe que foram classificados como genuínos (GONZAGA, 2014).

3.6 Considerações Finais deste Capítulo

Este capítulo apresentou os classificadores *Naive Bayes*, *k*-NN, SVM e RNA-MLP, mostrando seus diferentes parâmetros que devem ser escolhidos ou selecionados quando aplicados.

O que se pretende com este trabalho é focar apenas no problema da rotação, e assim, selecionar um algoritmo que obtenha tempos baixos de extração de características e classificação bem como boas taxas de acerto, considerando as imagens extraídas do sensor 3D em vários ângulos. Assim, o próximo capítulo mostra a metodologia empregada neste trabalho bem como os resultados obtidos pelos diferentes extratores de características e classificadores.

4. METODOLOGIA E ANÁLISE DOS RESULTADOS

O planejamento de um sistema de reconhecimento invariante à rotação é feito em vários estágios, entre eles destacam-se: Aquisição dos Dados, Pré-processamento, Extração de Características e Classificação. Vários métodos de reconhecimento podem ser definidos em relação à etapa de aquisição de dados.

O presente trabalho propõe um estudo comparativo entre extratores e classificadores para classificação de imagens invariantes à rotação extraídas de um sensor industrial 3D. Nesse capítulo é apresentado um estudo envolvendo a aplicação das técnicas apresentadas nos capítulos anteriores, tanto para extração de características como para classificação de padrões, aplicados a um problema de classificação de objetos invariantes à rotação. Assim, inicia-se o estudo mostrando a metodologia utilizada. Posteriormente, os extratores são comparados. Por fim, utilizando os extratores que obtiveram as melhores taxas, foi feito um estudo comparativo também entre os classificadores. Para este fim, foi montado um aparato, descrito a seguir.

4.1 Metodologia

O diagrama do sistema para a extração, processamento e classificação das imagens pode ser visto na Figura 11, onde o funcionamento global do sistema, atividade principal, é representado pelos blocos em negrito e as atividades secundárias ao fundo são representadas pelos blocos em cinza. O bloco pré-processamento está sendo representado por linha tracejada, pois vários extratores exigem pré-processamento.

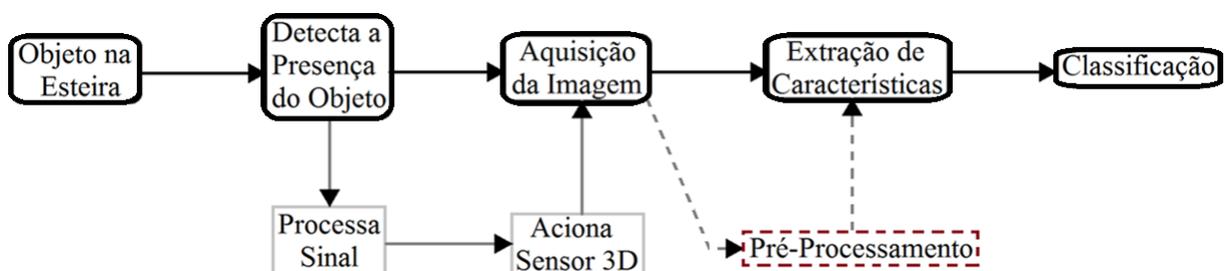


Figura 11 - Representação das etapas para classificação dos objetos.

O aparato experimental, por sua vez, é mostrado na Figura 12, e consiste de sensores que monitoram uma área de trabalho sobre uma esteira transportadora. O movimento da esteira é iniciado a partir de um motor trifásico e, o acionamento é feito por meio de um inversor de frequência PowerFlex 40P. Este inversor de frequência está ligado as saídas do

CLP Micrologix 1200. Neste mesmo CLP está conectado um sensor óptico, responsável pela detecção da presença do objeto e por disparar a aquisição da imagem através do sensor 3D (montado na barra superior).

O hardware utilizado para a aquisição das imagens foi o sensor 3D *effector pmd E3D200* da ifm *electronic*®, com resolução de 50×64 *pixels*. Este sensor possui interface Ethernet, permitindo assim, a implementação de aplicações em tempo real de algoritmos de classificação. Após a aquisição, as imagens são transferidas para processamento e classificação no MATLAB da *MathWorks*®. Os experimentos foram executados em uma máquina física com as seguintes características:

- CPU: *Intel Core i5-3210M 2.5 GHz*
- Memória RAM: 6 GB
- *Hard disk*: 500GB – 5.400 rpm
- Sistema operacional: *Windows 7 Home Basic (64 Bits)*

A solução escolhida para estabelecer a comunicação entre dois dispositivos ou aplicações que implementem diferentes protocolos, foi utilizar um servidor OPC (*OLE for Process Control*). Nesta solução utilizamos o *software RSLinx* da *Rockwell Automation*® para comunicar os dados gerenciados pelo CLP *Micrologix 1200*®, também da *Rockwell Automation*®, e a aplicação implementada. Uma descrição aprofundada pode ser encontrada em (SERPA, 2014), onde os algoritmos descritos neste trabalho estavam em execução e foram apresentados através de um vídeo.

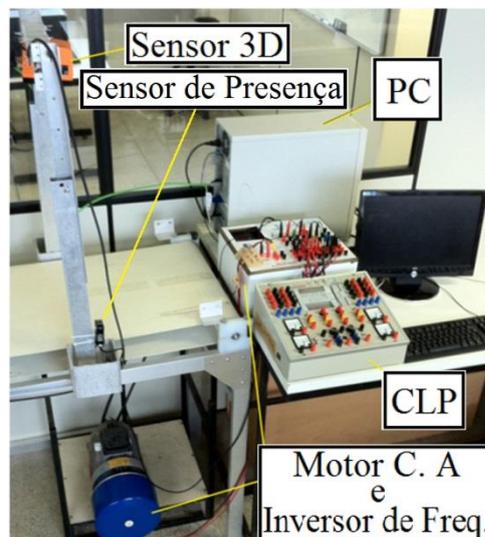


Figura 12 - Estrutura física mostrando os equipamentos utilizados.

Na fase de testes, 3 caixas com geometrias muito próximas, $15 \times 10.5 \times 7.2$ cm, $15 \times 14 \times 6$ cm, e $21.5 \times 16.2 \times 9.6$ cm, foram utilizadas. A Figura 13 mostra as imagens das 3 caixas selecionadas para teste. É possível verificar a má qualidade da imagem adquirida e a influência da iluminação.

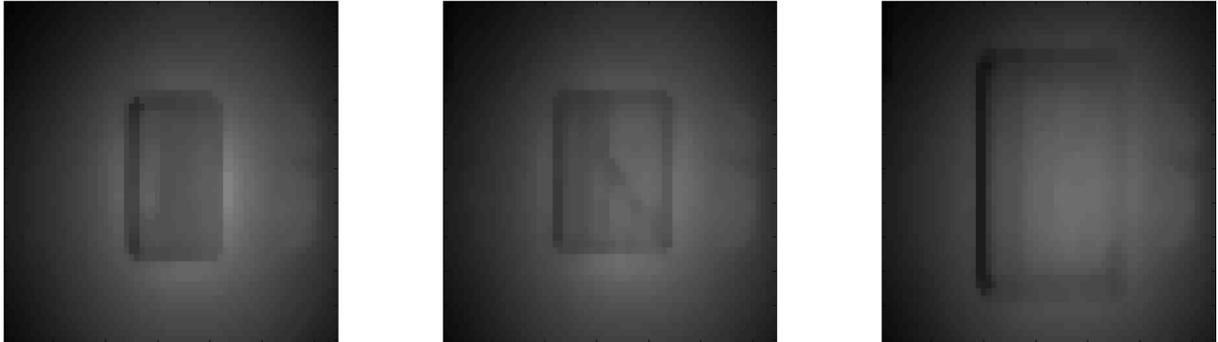


Figura 13 - Caixas com dimensões $15 \times 10.5 \times 7.2$ cm, $15 \times 14 \times 6$ cm, e $21.5 \times 16.2 \times 9.6$ cm respectivamente, e com resolução 50×64 pixels.

Os experimentos são baseados em apenas 3 classes, onde há 6 protótipos por classe, cada qual correspondendo aos seis lados de cada caixa. Desta forma, o banco de dados para o treinamento contém 18 objetos. Para o conjunto de dados de testes, este possui 150 imagens que sofreram rotação

No experimento realizado, o número de características extraídas para cada extrator é mostrado na Tabela 3.

Tabela 3 : Número de características extraídas.

| | Número de Entrada |
|-------------------------|--------------------------|
| Hu | 7 |
| Zernike | 36 |
| Legendre | 36 |
| Fourier-Mellin | 36 |
| Tchebichef | 36 |
| Bessel-Fourier | 36 |
| Gaussian-Hermite | 36 |
| HT | 36 |
| ICA | 18 |

Para os momentos, as dimensões de seus vetores característicos dizem respeito às suas particularidades individuais bem como às características de seus polinômios. Estas características são descritas a seguir:

- **Hu** – possui apenas 7 momentos;

- **Legendre, Fourier-Mellin, Tchebichef, Bessel-Fourier e Gaussian-Hermite** – seus vetores característicos são formados pelas ordens de 0 a 5 e repetições de 0 a 5, formando vetores característicos com dimensões 36: $[(0, 0), (0, 1), \dots, (5, 5)]$.
- **Zernike** – ordens de 0 – 10 e repetições de 0 – 10. Esta escolha deve-se ao fato de que vários momentos, tais como os momentos de ordens e repetições (0, 1), (0, 2), (0, 3), ..., entre outros, são nulos, assim, foram necessárias 10 ordens e 10 repetições de modo a formar um vetor característico de mesmo comprimento que os momentos anteriores, 36 características.

A escolha das ordens e repetições foi feita visando obterem-se os menores tempos possíveis para a extração das características dos momentos, pois se trata de um processo industrial em tempo real. Em muitos problemas em análise de imagens, a ordem e as repetições são aleatórias, como pode ser visto em (TEAGUE, 1980; MUKUNDA *et al.*, 2001).

A extração do vetor característico através da HT consiste em tomar as características do espaço de Hough através do método da Análise de Componentes Principais (PCA). Deste modo, foi possível reduzir a dimensão do espaço de Hough, 161×180 , para um vetor com 36 características. Além da PCA, para melhorar a descrição do vetor característico e conseqüentemente aumentar as taxas de classificação do HT, foi utilizado um filtro Laplaciano da Gaussiana (LoG) pois o desempenho da HT é altamente dependente dos resultados a partir da detecção de arestas, já que as imagens de entrada não puderam ser cuidadosamente escolhidas para uma melhor detecção destas.

Para a ICA, como foi utilizado o algoritmo FastICA, este algoritmo mantém o tamanho padrão do vetor característico, sendo igual ao número de dados de entrada. Assim, manteve-se o padrão para a escolha do tamanho do vetor característico no algoritmo FastICA.

4.2 Resultados Experimentais I – Extração de Características

A etapa de extração de características do experimento consiste na execução dos algoritmos de extração de características baseado nos momentos de Hu, Zernike, Legendre, Fourier-Mellin, Tchebichef, Bessel-Fourier e Gaussian-Hermite, e nos algoritmos Transformada de Hough e Análise de Componentes Independentes. Para a escolha dos melhores extratores de características e posteriormente suas comparações com diferentes heurísticas de classificação, estes vetores característicos, foram apresentados para o classificador k -NN, utilizando a distância euclidiana, visto ser um classificador de simples

utilização. Os resultados experimentais podem ser vistos na Tabela 4, onde **Min** - Mínimo, **Max** - máximo, **Me** - Média, **Med** - Mediana e **STD** - Desvio padrão, referentes às taxas de acerto (%) para os valores de k de 1 a 18, e, os momentos de **H** – Hu, **Z** - Zernike, **L** - Legendre, **FM** - Fourier-Mellin, **T** - Tchebichef, **BF** - Bessel-Fourier e **GH** - Gaussian-Hermite.

Tabela 4 : Taxa de reconhecimento dos momentos, HT e ICA, utilizando k -NN.

| | Min | Max | Me | Med | STD |
|------------|--------------|--------------|--------------|--------------|------------|
| H | 42.00 | 68.67 | 59.37 | 62.33 | 8.770 |
| Z | 70.00 | 97.33 | 81.63 | 79.00 | 10.18 |
| L | 62.67 | 87.33 | 76.93 | 79.00 | 8.880 |
| FM | 60.67 | 83.33 | 70.30 | 68.00 | 7.440 |
| T | 70.67 | 89.33 | 81.07 | 82.33 | 6.990 |
| BF | 66.00 | 78.00 | 72.70 | 72.67 | 3.330 |
| GH | 56.67 | 68.00 | 63.48 | 64.67 | 3.323 |
| HT | 70.00 | 81.33 | 73.89 | 71.67 | 3.980 |
| ICA | 57.33 | 72.00 | 64.51 | 65.66 | 3.910 |

Como este trabalho concentra-se em uma aplicação industrial em tempo real, é de fundamental importância haver um compromisso entre os dois aspectos, rapidez \times eficiência de classificação, assim, é necessário determinar o tempo médio para extrair as características de cada extrator utilizado, e o tempo médio para treinamento e classificação do classificador k -NN, levando em consideração todo o algoritmo para classificação. Assim, a Tabela 5 mostra o tempo médio para cada extrator, os quais são dados em segundos.

Tabela 5 : Tempo extração e classificação.

| | Extração de Características (segundos) | Treinamento / Classificação (segundos) |
|------------|--|--|
| H | 0.3128 | 0.0197 |
| Z | 0.3910 | 0.0173 |
| L | 0.2986 | 0.0177 |
| FM | 3.7672 | 0.0164 |
| T | 0.3041 | 0.0176 |
| BF | 0.3673 | 0.0193 |
| GH | 0.3113 | 0.0183 |
| HT | 0.3614 | 0.0184 |
| ICA | 0.2901 | 0.3546 |

Os momentos de Zernike e Tchebichef têm melhores capacidades de representação de imagem do que os momentos contínuos tradicionais, porque esses momentos preservam quase toda a informação da imagem em poucos coeficientes (HUAZHONG *et al.*, 2010). Como exemplo, temos os polinômios de Legendre afetados quando a imagem é discretizada. Como consequência, tais discretizações podem causar erros numéricos quando os momentos são computados. Descritores de recursos que são invariantes em relação a rotações no plano da imagem podem ser facilmente construídos utilizando os momentos de Tchebichef e Zernike (SILVA *et al.*, 2013a; SILVA *et al.*, 2013b; SILVA *et al.*, 2013c).

A ótima taxa de classificação dos momentos de Zernike pode estar relacionada com o uso dos momentos de ordens e repetições 0-10, em comparação com apenas os momentos de ordens e repetições 0-5 dos outros extratores.

Os momentos de Fourier-Mellin demandam, contudo, maiores esforços computacionais que os outros momentos. Além disto, o grande tempo de duração na extração de características dos momentos de Fourier-Mellin limita a sua utilização nesta tarefa. Este fato pode ser atribuído ao grande número de somas nas equações. Quando levado ao nível computacional, são traduzidos em laços iterativos, o que, por sua vez, demanda muito esforço computacional.

Os momentos de Legendre e Tchebichef caem na mesma classe de momentos ortogonais definidos no espaço de coordenadas cartesianas, onde momentos invariantes, particularmente invariantes à rotação, não estão prontamente disponíveis.

Para os momentos de Gaussian-Hermite, podemos relacionar suas taxas com a escolha de $\sigma = 0.1$. Este foi escolhido após testes para vários valores de σ , contudo, pode-se ainda variar o seu valor e assim obter-se uma melhora na taxa de acerto destes momentos. Para as taxas dos momentos de Hu, além de não se encontrar nas classes dos momentos ortogonais, este possui poucas características que podem ser extraídas para descrever uma imagem. Para os momentos de Bessel-Fourier, o baixo grau das ordens e repetições pode estar relacionado com sua taxa de acerto, já que em outros trabalhos estas taxas são melhores (MA *et al.*, 2011; GAO *et al.*, 2013; FEN *et al.*, 2011).

Quanto à transformada de Hough e a Análise de Componentes Independentes, embora uma investigação mais profunda seja necessária, pode-se dizer que a rotação das imagens afeta o seu desempenho.

Para os momentos de Zernike, de fato, sua precisão obtida atingiu 97.33 %, no entanto, obteve um desvio padrão de 10.18 % e esta é uma incerteza muito grande para

considerar o reconhecimento de imagens em tempo real. Este fato pode ser relacionado com a falta de qualidade, ruído, imperfeições na iluminação contida nas imagens extraídas a partir do sensor de 3D (baixa resolução), pequena quantidade de ordens e repetições utilizadas, (0 - 10), que formam o vetor de entrada de 36 características e, uma quantidade muito baixa de amostras para treinamento, 18 amostras.

Assim, como o conjunto de dados para treinamento foi de apenas 18 amostras torna-se interessante verificar as taxas quando o conjunto de treinamento for maior. Portanto, separando 50% da base de dados, 28 amostras por classe, para o treinamento e 50% para teste, alcançou-se os resultados apresentados na Tabela 6, e, separando 80% da base de dados para treinamento, 45 amostras por classe, e 20% para teste, obteve-se os resultados mostrados na Tabela 7.

Tabela 6 : Taxa de acerto tomando 50% da base de dados para treinamento.

| | Min | Max | Me | Med | STD |
|------------|--------------|--------------|--------------|--------------|------------|
| H | 33.33 | 85.71 | 60.47 | 63.1 | 16.96 |
| Z | 64.28 | 100.0 | 79.96 | 77.38 | 10.38 |
| L | 39.28 | 89.28 | 59.15 | 57.73 | 13.29 |
| FM | 50.00 | 91.66 | 72.03 | 73.81 | 10.76 |
| T | 51.19 | 98.81 | 73.38 | 73.81 | 14.76 |
| BF | 63.09 | 85.71 | 76.21 | 75.59 | 4.210 |
| GH | 44.44 | 84.85 | 61.29 | 62.63 | 12.73 |
| HT | 76.19 | 85.71 | 81.22 | 80.95 | 1.730 |
| ICA | 54.76 | 92.85 | 75.82 | 78.57 | 11.91 |

Tabela 7 : Taxa de acerto tomando 80% da base de dados para treinamento.

| | Min | Max | Me | Med | STD |
|------------|--------------|--------------|--------------|--------------|------------|
| H | 33.33 | 84.84 | 64.42 | 69.69 | 16.78 |
| Z | 75.75 | 100.0 | 89.78 | 90.90 | 6.470 |
| L | 57.57 | 100.0 | 71.98 | 69.69 | 10.19 |
| FM | 66.66 | 90.90 | 79.79 | 81.81 | 4.560 |
| T | 69.69 | 96.96 | 79.86 | 72.72 | 10.27 |
| BF | 72.72 | 93.93 | 84.66 | 84.84 | 4.720 |
| GH | 45.45 | 100.0 | 77.76 | 87.88 | 20.39 |
| HT | 69.69 | 90.90 | 80.69 | 78.78 | 7.250 |
| ICA | 66.66 | 100.0 | 88.53 | 90.90 | 10.72 |

De acordo com as Tabelas II, III, IV e V, fica evidente a evolução e, portanto, o uso de extração de características baseado em momentos de Zernike, que oferece boas taxas de acertos e tempo de extração de características baixo.

4.3 Resultados Experimentais II – Classificação

A etapa de classificação do experimento consiste na execução dos algoritmos de classificação *Naive Bayes*, *k*-NN, RNA-MLP e SVM, onde a entrada dos algoritmos foi obtida pelo extrator baseado em momentos de Zernike. Para comparações de possíveis erros entre classes, foi utilizado o método da matriz de confusão. Além deste método os dados foram divididos em:

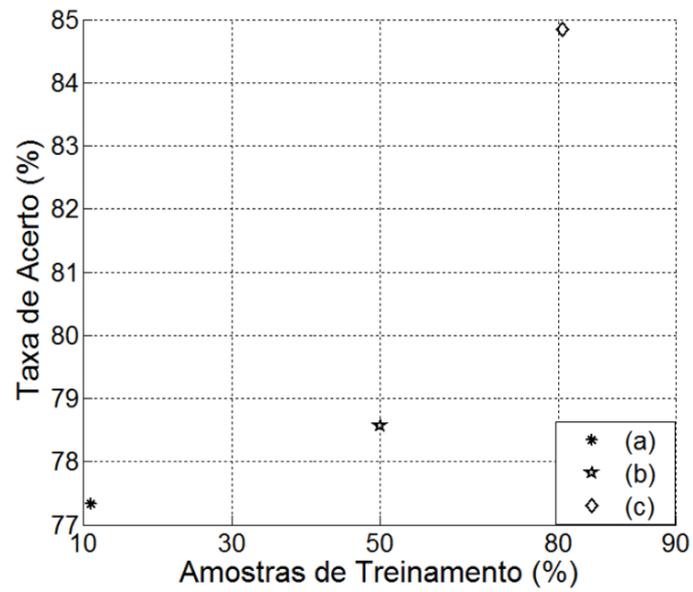
- (a) 18 amostras para treinamento (referentes aos 6 lados de cada caixa) / 150 amostras para teste (10.71% treinamento – 89.28% teste);
- (b) 84 amostras para treinamento (aleatórias) / 84 amostras para teste (50% treinamento – 50% teste), e;
- (c) 135 amostras para treinamento (aleatórias) / 33 amostras para teste (80.35% treinamento – 19.64% teste).

A divisão de dados (a) é a investigação deste trabalho. As investigações (b) e (c), foram desenvolvidas para averiguar a possível evolução/comportamento de cada classificador e, conseqüentemente, identificar qual ou quais classes são mais difíceis de serem representadas/identificadas, utilizando o método da matriz de confusão.

4.3.1. Classificador *Naive Bayes*

Para este classificador, a Figura 14 mostra o resultado e a matriz de confusão para os casos de divisão de dados (a), (b) e (c).

Através das matrizes de confusão, inicialmente (divisão de dados (a)) fica evidente a maior falha na representação da classe 2, a qual classificou a amostra como pertencente a classe 1 em 46.00%. Para as demais matrizes de confusão (divisão de dados (b) e (c)) as maiores confusões dos dados se encontram na classe 1, onde em (b) há 35.71% de confusão com a classe 2 e, em (c) há 27.27% de confusão com a classe 3 classificada como 2. Como a matriz relevante para este estudo é a relacionada à divisão de dados (a), o classificador *Naive Bayes* classificou amostras da classe 2 como sendo da classe 1 há uma razão $FNR1 = 0.4600$.



| (a) | 1 | 2 | 3 | FPR |
|------------|--------|--------|--------|--------|
| 1 | 1.0000 | 0.4600 | 0.1800 | 0.2424 |
| 2 | 0.0000 | 0.5400 | 0.0400 | 0.0196 |
| 3 | 0.0000 | 0.0000 | 0.7800 | 0.0000 |
| FNR | 0.0000 | 0.4600 | 0.2200 | 0.7733 |

| (b) | 1 | 2 | 3 | FPR |
|------------|--------|--------|--------|--------|
| 1 | 0.6428 | 0.0000 | 0.1290 | 0.0667 |
| 2 | 0.3571 | 1.0000 | 0.1290 | 0.1918 |
| 3 | 0.0000 | 0.0000 | 0.7419 | 0.0000 |
| FNR | 0.3571 | 0.0000 | 0.2581 | 0.7857 |

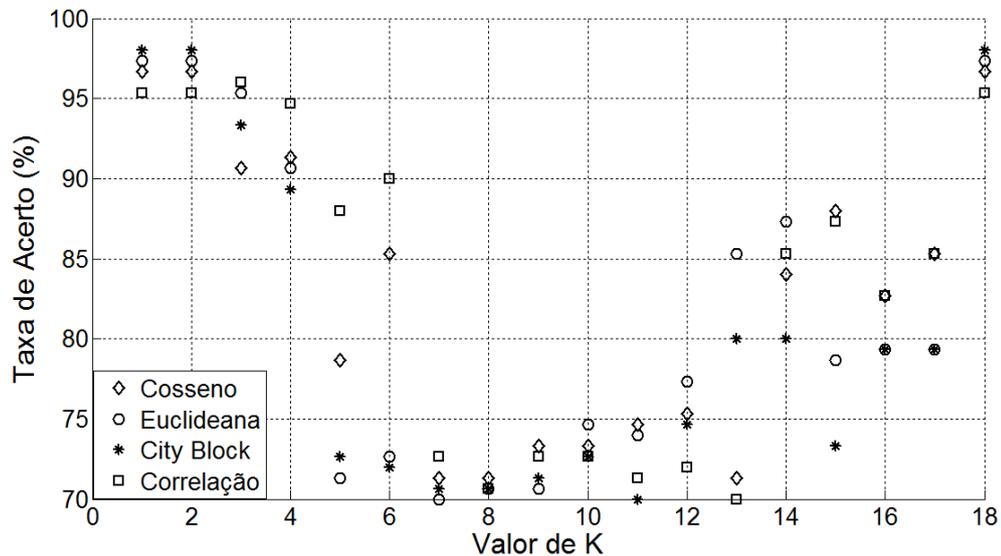
| (c) | 1 | 2 | 3 | FPR |
|------------|--------|--------|--------|--------|
| 1 | 0.9230 | 0.0000 | 0.0909 | 0.0476 |
| 2 | 0.0769 | 1.0000 | 0.2727 | 0.1429 |
| 3 | 0.0000 | 0.0000 | 0.6363 | 0.0000 |
| FNR | 0.0769 | 0.0000 | 0.3636 | 0.8485 |

Figura 14 - Taxa de acerto e Matriz de Confusão.

4.3.2. Classificador k -NN

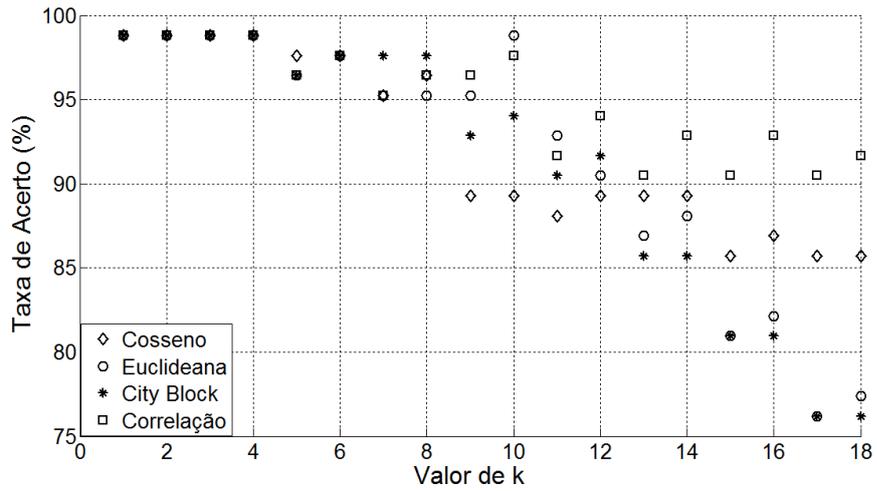
Para este classificador, as Figuras 15, 16 e 17 mostram as taxas de acerto e a matriz de confusão da distância que conquistou o melhor acerto para os casos de divisão de dados **(a)**, **(b)** e **(c)** respectivamente.

A matriz de confusão da Figura 15 mostra que o maior erro entre classes se encontra na classe 3 onde há um erro de 4.00% com a classe 2. Para os dados **(b)**, Figura 16, o erro entre classes permanece entre as classes 3 e 2, contudo, há um decréscimo indo para 3.23%. Finalmente os dados **(c)**, Figura 17, possuem taxa de acerto de 100.00%. Como o conjunto de dados que possui maior relevância para esta investigação é o **(a)**, este classificou como classe 2 uma parte da classe 3, gerando uma $FNR1 = 0.04$.



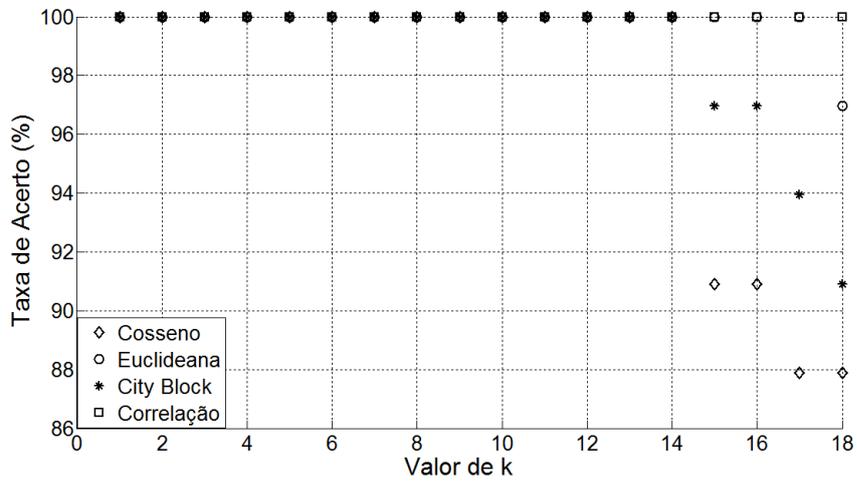
| (a) | 1 | 2 | 3 | FPR |
|-----|--------|--------|--------|--------|
| 1 | 1.0000 | 0.0200 | 0.0000 | 0.0099 |
| 2 | 0.0000 | 0.9800 | 0.0400 | 0.0196 |
| 3 | 0.0000 | 0.0000 | 0.9600 | 0.0000 |
| FNR | 0.0000 | 0.0200 | 0.0400 | 0.9800 |

Figura 15 - Acurácia e Matriz de Confusão da distância City Block para $k = 1$ (Taxa de acerto = 98.00%).



| (b) | 1 | 2 | 3 | FPR |
|------------|--------|--------|--------|--------|
| 1 | 1.0000 | 0.0000 | 0.0000 | 0.0000 |
| 2 | 0.0000 | 1.0000 | 0.0322 | 0.0167 |
| 3 | 0.0000 | 0.0000 | 0.9677 | 0.0000 |
| FNR | 0.0000 | 0.0000 | 0.0322 | 0.9881 |

Figura 16 - Acurácia e Matriz de Confusão da distância City Block para $k = 1$ (Taxa de acerto = 98.81%).



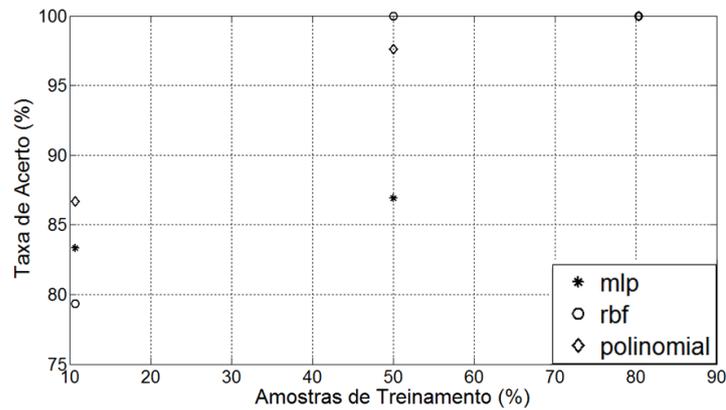
| (c) | 1 | 2 | 3 | FPR |
|------------|--------|--------|--------|--------|
| 1 | 1.0000 | 0.0000 | 0.0000 | 0.0000 |
| 2 | 0.0000 | 1.0000 | 0.0000 | 0.0000 |
| 3 | 0.0000 | 0.0000 | 1.0000 | 0.0000 |
| FNR | 0.0000 | 0.0000 | 0.0000 | 1.0000 |

Figura 17 - Acurácia e Matriz de Confusão da distância City Block para $k = 1$ (Taxa de acerto = 100.00%).

4.3.3. Classificador Máquina de Vetor de Suporte

Para o classificador SVM, alguns parâmetros foram testados, e conseqüentemente adotados, priorizando a melhora na taxa de acerto. Para a função *kernel* rbf, foi adotado $\sigma = 1$, para *kernel* polinomial, foi adotado $d = 1$, e para *kernel* mlp, foi adotado o vetor de dois elementos [2.4, -31] que especifica os parâmetros de escala e limiar $\tanh(2.4 * x * y - 31)$. Assim, a Figura 18 mostra os resultados dos 3 *kernels* e, conseqüentemente, a matriz de confusão do *kernel* que atingiu a melhor taxa de acerto, para as diferentes divisões de dados. Para todos estes *kernels*, foi utilizado na fase de treinamento programação quadrática (QP).

O maior erro encontrado entre classes, conjunto (a), está entre na classe 2 a qual classificou em 16,00% como pertencente a classe 1. Além disto, esta classe possui uma FNR1 = 0.2400. Para os dados (b) e (c), foi obtido uma taxa de acerto de 100%.



| (a) | 1 | 2 | 3 | FPR | (b) | 1 | 2 | 3 | FPR |
|------------|--------|--------|--------|--------|------------|--------|--------|--------|--------|
| 1 | 1.0000 | 0.1600 | 0.1200 | 0.1228 | 1 | 1.0000 | 0.0000 | 0.0000 | 0.0000 |
| 2 | 0.0000 | 0.7600 | 0.0400 | 0.0196 | 2 | 0.0000 | 1.0000 | 0.0000 | 0.0000 |
| 3 | 0.0000 | 0.0800 | 0.8400 | 0.0385 | 3 | 0.0000 | 0.0000 | 1.0000 | 0.0000 |
| FNR | 0.0000 | 0.2400 | 0.1600 | 0.8667 | FNR | 0.0000 | 0.0000 | 0.0000 | 1.0000 |

| (c) | 1 | 2 | 3 | FPR |
|------------|--------|--------|--------|--------|
| 1 | 1.0000 | 0.0000 | 0.0000 | 0.0000 |
| 2 | 0.0000 | 1.0000 | 0.0000 | 0.0000 |
| 3 | 0.0000 | 0.0000 | 1.0000 | 0.0000 |
| FNR | 0.0000 | 0.0000 | 0.0000 | 1.0000 |

Figura 18 - Acurácia e Matriz de Confusão à função *kernel* polinomial (a), rbf (b) e rbf / polinomial (c).

4.3.4. *Classificador Rede Neural – Perceptron Multi-Camadas*

O classificador Rede Neural Artificial – Perceptron Multi-Camadas, teve de ser verificado e seus parâmetros tiveram de ser exaustivamente alterados para que fosse possível obter taxas boas de acertos. Por fim, os parâmetros encontrados podem ser observados abaixo:

Função de Ativação Tangente Hiperbólica – neurônio ocultos [7 5], algoritmo de treinamento *backpropagation* gradiente descendente.

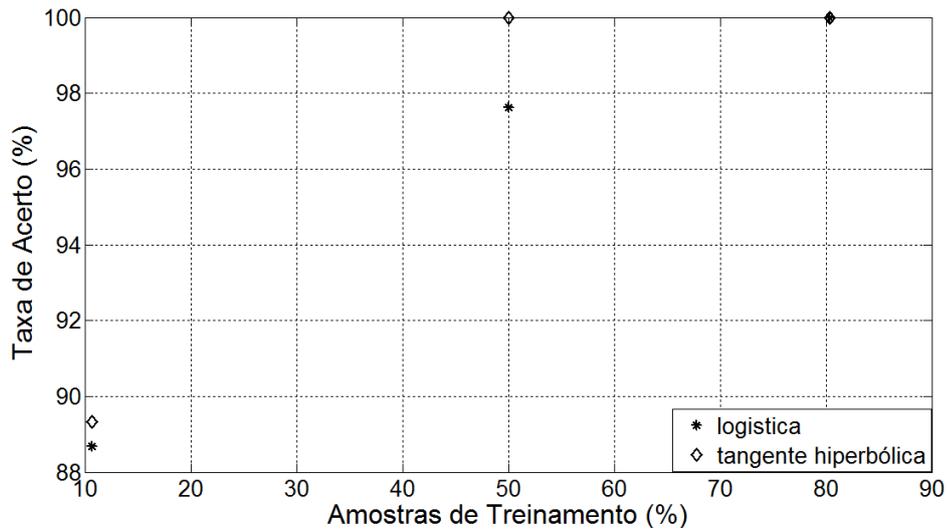
Função de Ativação Logística – neurônio ocultos [8 8], algoritmo de treinamento *backpropagation* gradiente descendente.

Para a fase de treinamento, foram adotados os seguintes parâmetros:

- Épocas – 1400,
- Erro final desejado – 0,
- Taxa de aprendizagem – 0.01,
- Desempenho mínimo de gradiente – 10^{-10} .

Assim, Figura 19 mostra os resultados das classificações dos dados **(a)**, **(b)** e **(c)**, e suas respectivas matrizes de confusão referente a função de ativação que alcançou a melhor taxa de acerto.

Para o conjunto **(a)**, a classe que obteve a maior quantidade de erros foi a classe 2 onde, ocorreu 24.0% de equívoco entre esta classe e a classe 1. A maior FNR1 = 0.2600, foi encontrada nesta mesma classe, conseqüentemente entre as classes 2 e 1. Para os demais dados, **(b)** e **(c)**, a taxa de acerto obtida foi de 100%.



| (a) | 1 | 2 | 3 | FPR |
|------------|--------|--------|--------|--------|
| 1 | 0.9400 | 0.2400 | 0.0000 | 0.1071 |
| 2 | 0.0600 | 0.7400 | 0.0000 | 0.0291 |
| 3 | 0.0000 | 0.0200 | 1.0000 | 0.0099 |
| FNR | 0.0600 | 0.2600 | 0.0000 | 0.8933 |

| (b) | 1 | 2 | 3 | FPR |
|------------|--------|--------|--------|--------|
| 1 | 1.0000 | 0.0000 | 0.0000 | 0.0000 |
| 2 | 0.0000 | 1.0000 | 0.0000 | 0.0000 |
| 3 | 0.0000 | 0.0000 | 1.0000 | 0.0000 |
| FNR | 0.0000 | 0.0000 | 0.0000 | 1.0000 |

| (c) | 1 | 2 | 3 | FPR |
|------------|--------|--------|--------|--------|
| 1 | 1.0000 | 0.0000 | 0.0000 | 0.0000 |
| 2 | 0.0000 | 1.0000 | 0.0000 | 0.0000 |
| 3 | 0.0000 | 0.0000 | 1.0000 | 0.0000 |
| FNR | 0.0000 | 0.0000 | 0.0000 | 1.0000 |

Figura 19 - Acurácia e Matriz de Confusão da função de ativação tangente hiperbólica (a), (b) e tangente hiperbólica / logística (c).

Após a análise das taxas de acerto e matrizes de confusão, para o conjunto de dados (a), é perceptível que alguns classificadores encontram dificuldades em representar algumas classes e conseqüentemente, a falha em suas eventuais classificações. Assim, torna-se fundamental verificar como cada classificador representa os dados de treinamento.

4.3.5. Representação das Classes

Para verificar quais classes são melhores representadas pelos classificadores, foram utilizados, como treinamento e teste, os dados de treinamento do conjunto (a).

A Figura 20 mostra o comportamento das classes classificadas para os classificadores *Naive Bayes* e RNA-MLP com função de ativação tangente hiperbólica, os quais são representados por (*), em confronto com os rótulos reais das classes representadas por (o). É evidente o erro cometido por estes classificadores ao representarem a amostra 7, a qual pertence a classe 2 e é classificada como classe 1. Este erro possivelmente deve-se a geometria muito próxima entre as duas imagens.

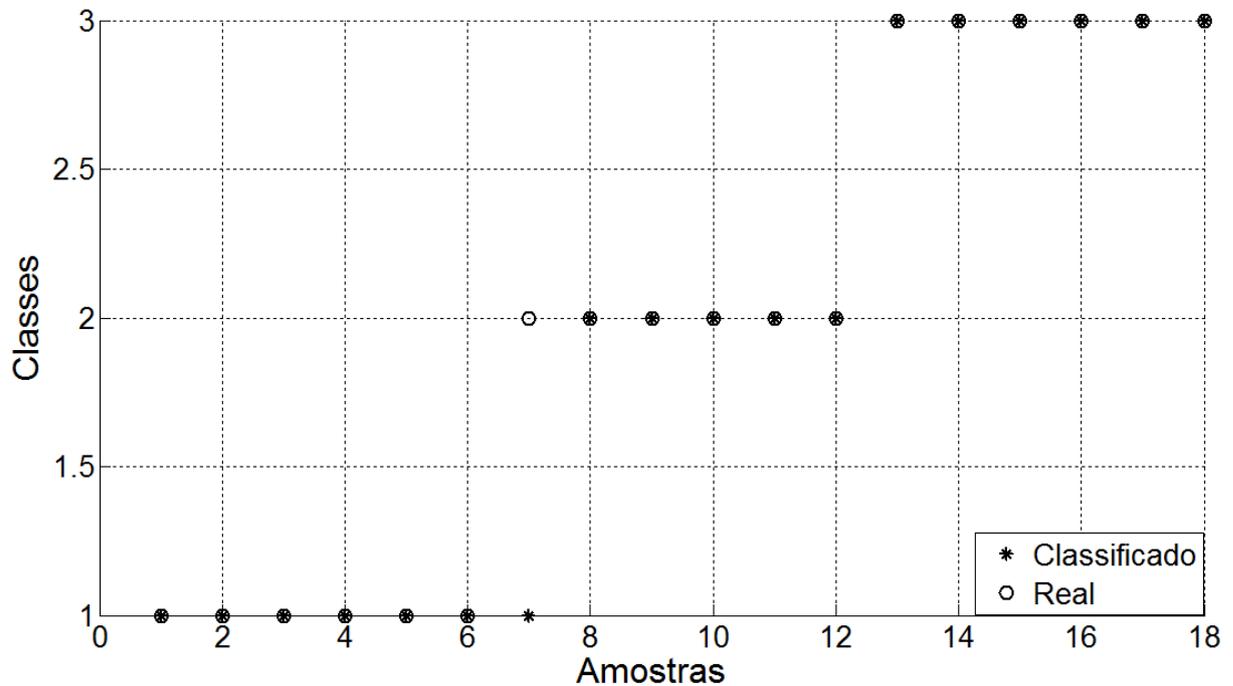


Figura 20 – Amostras classificadas × amostras reais dos classificadores *Naive Bayes* e RNA-MLP.

Ao confrontar a Figura 20 com as respectivas matrizes de confusão dos classificadores *Naive Bayes* e RNA-MLP, encontradas nas Figuras 14 e 19, fica evidente que os erros nas representações da amostra 7 afeta suas eventuais classificações.

Para os classificadores *k*-NN e SVM, a Figura 21 representa os dados classificados e reais. Estes dois classificadores obtiveram acurácia máxima, consequentemente, conseguem representar bem as amostras.

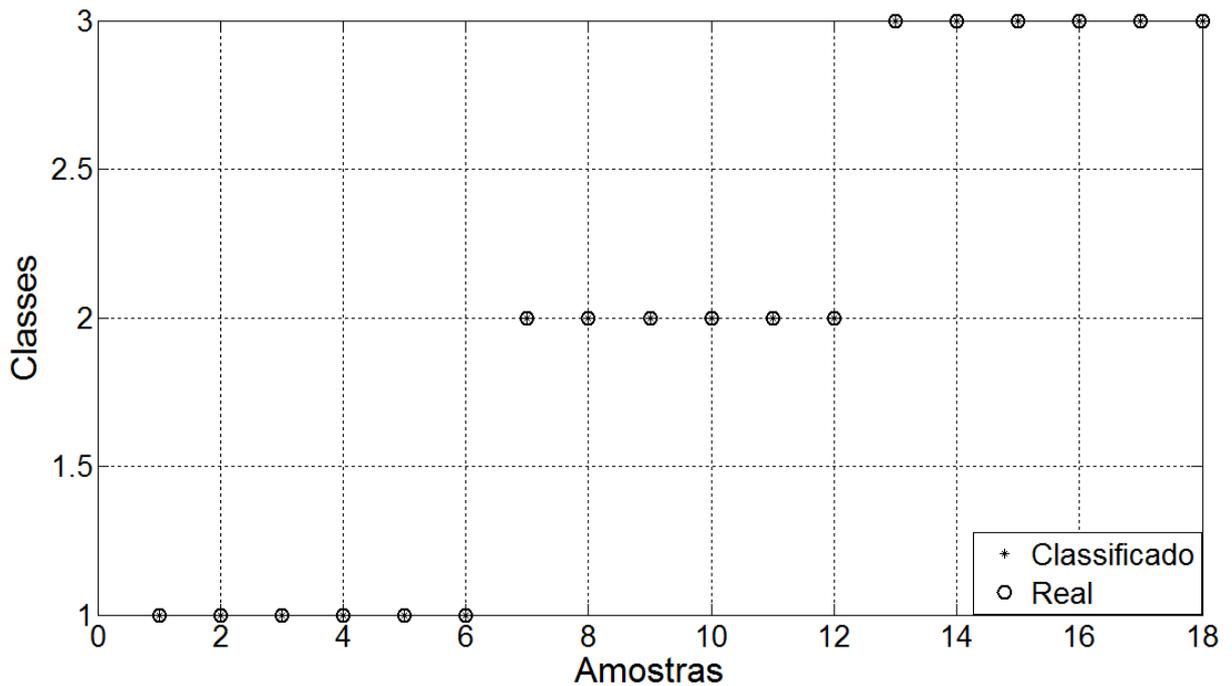


Figura 21 - Amostras classificadas \times amostras reais classificador k -NN e SVM.

Como a intenção deste trabalho é investigar, e por fim, mostrar qual extrator e classificador é mais adequado para classificação de objetos em tempo real, uma última verificação se torna fundamental, com que velocidade cada classificador treina e classifica cada amostra?

4.3.6. Tempo de Processamento dos Classificadores

Os tempos de treinamento e classificação para todos os classificadores são descritos nas Tabelas 8 e 9.

Na Tabela 8, o classificador RNA-MLP, teve maior tempo de treinamento, contudo isto já era previsto, visto que este fato é conhecido em redes neurais utilizando *backpropagation* (normalmente atribuído a necessidade de milhares de ciclos para se chegar a níveis de erros aceitáveis, tempo de aprendizado, grande quantidade de padrões, quantidade de camadas), contudo outras adaptações do algoritmo *backpropagation*, como o *resilient backpropagation*, podem obter melhores tempos de treinamento. Já o classificador k -NN atingiu os menores tempos de treinamento.

A Tabela 9 mostra que o classificador que levou mais tempo para classificar as amostras foi o SVM (provavelmente está relacionado com a dimensionalidade dos dados, a separação das classes, de 2 em 2, favorecendo o desempenho ao custo de maior complexidade, sua função *kernel*). Na fase de treinamento, a SVM foi treinada por

programação quadrática (QP), e o tempo de treinamento é geralmente proporcional ao quadrado do número de amostras ocasionando um tempo baixo de treinamento. O classificador que alcançou o menor tempo de classificação foi o *Naive Bayes*, contudo, este classificador possui uma taxa baixa de acertos.

Tabela 8 : Tempo de treinamento dos classificadores.

| | Treinamento (segundos) | | | | |
|--------------------|-------------------------------|------------|-----------|------------|------------|
| | Min | Max | Me | Med | STD |
| Naive Bayes | 0.0006 | 0.0954 | 0.0006 | 0.0006 | 0.0008 |
| k-NN | 0.0003 | 0.0595 | 0.0003 | 0.0003 | 0.0002 |
| SVM | 0.0228 | 1.4075 | 0.0346 | 0.242 | 0.1129 |
| RNA-MLP | 0.4279 | 13.8281 | 3.7361 | 1.7394 | 4.3255 |

Tabela 9 : Tempo de classificação dos classificadores.

| | Classificação (segundos) | | | | |
|--------------------|---------------------------------|------------|-----------|------------|------------|
| | Min | Max | Me | Med | STD |
| Naive Bayes | 0.0005 | 0.0831 | 0.0006 | 0.0005 | 0.0007 |
| k-NN | 0.0038 | 0.0723 | 0.0041 | 0.0040 | 0.0005 |
| SVM | 0.0811 | 0.1721 | 0.0913 | 0.0884 | 0.0109 |
| RNA-MLP | 0.0094 | 0.0187 | 0.0103 | 0.0102 | 0.0009 |

Por fim, podemos gerar um gráfico representando as taxas médias de acerto *versus* os tempos que classificação visto que estes sejam de maior interesse para este trabalho. A Figura 22 apresenta este gráfico.

As taxas médias de acerto é o ponto principal a ser verificado. Um segundo ponto de fundamental importância é a dispersão (desvio padrão) dos tempos de classificação. Dependendo das características do sistema, como por exemplo, a velocidade de deslocamento da esteira, torna-se inadequada a aplicação de um determinado classificador, visto que a aquisição das imagens e, posteriormente, a tomada de decisão do sistema (identificação da classe do objeto) é limitada por esta.

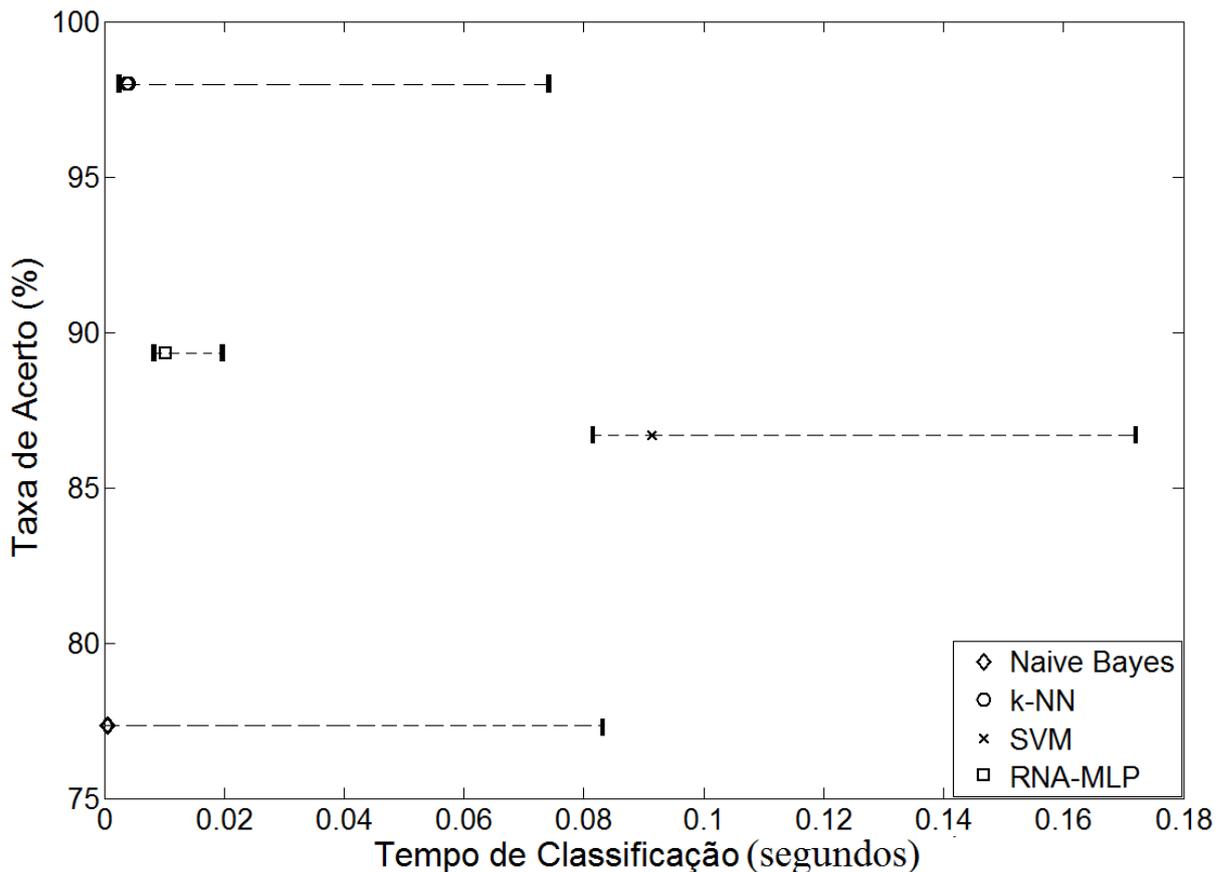


Figura 22 – Taxa Média de Acerto versus Tempo de Classificação. As barras representam os tempos mínimos e máximos de classificação para cada um dos classificadores e os símbolos seus respectivos tempos médios.

Como pode ser observado no gráfico, o SVM é o principal classificador que é limitado pela velocidade de deslocamento da esteira, já que seus tempos de classificação se encontram entre 0.0811 e 0.1721 segundos.

4.4 Softwares Utilizados

O *software* base para o desenvolvimento deste trabalho foi o MATLAB, pois muitos classificadores, como *Naive Bayes*, *k-NN* e *RNA-MLP*, e um extrator de características, Transformada de Hough, já estão implementados em *toolboxes* neste *software*. Para o classificador SVM, a *toolbox* implementada em MATLAB classifica apenas 2 classes, logo, foi utilizado o código desenvolvido por Cody Neuburger, da Florida Atlantic University, Florida USA, o qual pode ser encontrado em:

<http://www.mathworks.com/matlabcentral/fileexchange/39352-multi-class-svm/content/multisvm.m>

Para o extrator Análise de Componentes Independente, foi utilizada a ferramenta desenvolvida para MATLAB FastICA versão 2.5 de 19 de outubro de 2005. Este *toolbox* foi

desenvolvido por Hugo Gävert, Jarmo Hurri, Jaakko Särelä, e Aapo Hyvärinen, e pode ser encontrado em:

<http://research.ics.aalto.fi/ica/fastica/code/dlcode.shtml>

Para os extratores de características baseado nos momentos, foram desenvolvidas *scripts* próprios com base nos trabalhos:

(Hu) - Mercimek, M., Gulez, K., Mumcu, T. V. Real Object Recognition using Moment Invariants, **Sadhana**, v. 30, part 6, p. 765–775, 2005.

(Zernike) – HWANG, S. K., KIM, W. Y. A Novel Approach to the Fast Computation of Zernike Moments, **Pattern Recognition**, v. 39, p. 2065-2076, 2006.

(Legendre) – CHONG, C. W., RAVEENDRAN, P., MUKUNDAN, R. Translation and Scale Invariants of Legendre Moments, **Pattern Recognition**, v. 27, p. 119-129, 2004.

(Fourier-Mellin) – SINGH, C., UPNEJA, R. Accurate Computation of Orthogonal Fourier-Mellin Moments, **Journal of Mathematical Imaging and Vision**, v. 44, issue 3, p. 411-431, 2012.

(Tchebichef) – MUKUNDAN, R., ONG, S. H., LEE, P. A. Image Analysis by Tchebichef Moments, **IEEE Transactions on Image Processing**, v. 10, n. 9, p. 1357-1364, 2001.

(Bessel-Fourier) – XIAO, B., MA, J. F., WANG, X. Image Analysis by Bessel-Fourier Moments, **Pattern Recognition**, v. 43, p. 2620-2629, 2010.

(Gaussian-Hermite) - YANG, B., DAI, M. Image Analysis by Gaussian-Hermite Moments, **Signal Processing**, v. 91, issue 10, p. 2290-2303, 2011.

5. CONCLUSÃO

Este trabalho apresentou um estudo comparativo entre sete conjuntos de momentos invariantes (Hu, Zernike, Legendre, Fourier-Mellin, Tchebichef, Bessel-Fourier e Gaussian-Hermite), a Transformada de Hough e a Análise de Componentes Independentes como métodos de extração de características, para o reconhecimento de objetos invariantes à rotação, cujas imagens foram obtidas através de um sensor industrial 3D de baixa resolução. Inicialmente, foi utilizado o classificador k -NN, usando distância euclidiana, para verificar qual destes extratores representaria melhor os objetos. Os resultados experimentais mostraram que a taxa de reconhecimento do classificador k -NN com representação dos momentos de Zernike, são mais elevadas do que dos demais momentos, transformada de Hough e da análise de componentes independentes, e possui um tempo de extração, treinamento e classificação, relativamente baixo.

De fato, a precisão obtida com o melhor extrator, Zernike, atingiu 97.33 %, no entanto, 10.18 % de desvio padrão é uma incerteza muito grande para considerar o reconhecimento de imagens em tempo real. Uma possível solução para este problema, como mostrado neste estudo, é o aumento da quantidade de amostras de treinamento, neste caso, verificou-se um aumento significativo na média e nas taxas de sucesso dos momentos de Zernike.

Uma verificação a posteriori se deu na análise da melhoria da classificação utilizando métodos distintos (*Naive Bayes*, k -NN, SVM e RNA-MLP). Após investigações exaustivas podemos concluir que o classificador *Naive Bayes* não pode representar bem as amostras de treinamento alcançando apenas 77.33% de acerto no conjunto principal deste trabalho, (a). Por outro lado, o classificador k -NN atingiu boas taxas de representação das classes, acerto, 98.00%, e tempos de treinamento e classificação baixos.

Para a SVM o seu desempenho depende da seleção do tipo de *kernel* e dos seus parâmetros, o qual conseguiu uma boa representação das classes, taxas de acerto de 86.67% e tempos moderados para treinamento e classificação.

Para o classificador RNA-MLP, pode-se concluir que o seu desempenho é sensível ao tamanho da estrutura logo, este classificador sofreu com a pouca quantidade de representantes dos objetos, vetores característicos para o treinamento, desta forma, este classificador não adquiriu uma boa taxa de representação de classe esperada, resultando em uma taxa de acerto de 89.33% além de um tempo de treinamento elevado, contudo, com um tempo de classificação relativamente baixo.

Ao final, a Figura 22 confrontou a taxa média de acerto com o tempo necessário para classificação de cada classificador, e, foi discutida a influência do sistema físico (real) na escolha do melhor método de classificação, visto que estes são limitados pelos tempos de classificação.

Assim, podemos concluir que os classificadores individualmente apresentaram um comportamento interessante. Para a divisão de dados **(a)**, o melhor desempenho em geral foi do classificador k -NN. Para as divisões de dados **(b)** e **(c)**, todos, com exceção do classificador *Naive Bayes*, obtiveram taxa máxima de acerto. Isto demonstra as diferentes capacidades de cada classificador para tratar com atributos distintos de treinamento.

Em suma, para o propósito de classificação dos objetos que constituíram o presente estudo de caso, a melhor solução encontrada foi aquela baseada na análise de imagem por momento de Zernike, juntamente com o classificador k -Vizinhos mais Próximos.

REFERÊNCIAS

- ABRAMOWITZ, M., STEGUN, I. A., **Handbook of Mathematical Functions, National Bureau of Standards**, Applied Mathematics Series, v. 55, Dover Publications, 1965.
- ABU-MOSTAFA, Y. S., PSALTIS, D. Recognitive Aspect of Moment Invariants, **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. PAMI-6, p. 698–706, 1984.
- AIRF, T., SHAABAN, Z., KREKOR, L., BABA, S. Object Classification via Geometrical, Zernike na Legendre Moments, **Jornal of Theoretical & Applied Information Technology**, v. 6, n. 3, p. 31, 2009.
- AMOS, D. E., **A Portable Package for Bessel Functions of a Complex Argument and Nonnegative Order, ACM Transactions on Mathematical Software**, v. 12, n. 3, 1986 265–273.
- ASOUDEGI, E., PAN, Z. Computer Vision for Quality Control in Automated Manufacturing Systems, **Computers & Industrial Engineering**, v. 21, n. 1-4, p. 141-145, 1991.
- BACH, F. R., JORDAN, M. I. Kernel Independent Component Analysis, **Journal of Machine Learning Research**, v. 3, p. 1-48, 2002.
- BAIDYK, T., KUSSUL, E., MAKEYEV, O. Computer Vision System for Manufacturing of Micro Workpieces, **Proceedings of AI-2008, the Twenty-eighth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Applications and Innovations in Intelligente Systems XVI**, p. 19-32, 2009.
- BARINOVA, O., LEMPITSKY, V., KOHLI, P. On Detection of Multiple Object Instances using Hough Transforms, **IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, p. 2233-2240, 2010.
- BELGHINI, N., ZARGHILI, A., KHARROUBI, J. 3D Face Recognition using Gaussian Hermite Moments, **Internacional Journal of Computer Applications, Special Issue on Software Engineering, Databases and Expert Systems**, SEDEX, n. 1, p. 1-4, 2012. Published by Foundation of Compute Science, New York, USA.
- BELKSIM, S. O., SHRIDHAR, M., AHMADI, M. Pattern Recognition with Moment Invariants: A Comparative Study and New Results, **Pattern Recognition**, v. 24, p. 1117-1138, 1991.
- BELL, A. J., SEJNOWSKI, T. J. An Information-Maximization Approach to Blind Separation and Blind Deconvolution, **Neural. Computation**, v. 7, p. 1129–1159, 1995.

BIZON, K., LOMBARDI, S., CONTINILLO, G., MANCARUSO, E., VAGLIECO, B. M. **Analysis of Diesel Engine Combustion using Imaging and Independent Component Analysis**, Proceedings of the Combustion Institute, v. 34, p. 2921-2931, 2013.

BOSER, B. E., GUYON, I. M., VAPNIK, V. N. A Training Algorithm for Optimal Margin Classifiers, **5th Annual ACM Workshop on COLT**, ed. by D. Haussler, p. 144-152, Pittsburgh, PA, USA, ACM Press, 1992.

BOZMA, H. I., YALÇIN, H. Visual Processing and classification of Items on a Moving Conveyor: A Selective Perfection Approach, **Robotics and Computer Integrated Manufacturing**, v. 17, p. 125-133, 2002.

CARDOSO, J. F. Source Separation using Higher Order Moments, ICASSP, in: **Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing**, v. 4, p. 2109–2112, 23-26, 1989.

CHATTOPADHYAY, A. K., MONDAL, S., CHATTOPADHYAY, T. Independent Component Analysis for the Objective Classification of Globular Clusters of the Galaxy NGC 5128, **Computational Statistics & Data Analysis**, v. 57, n. 1, p. 17-32, 2013.

CHEN, F. L., SU, C. T. Vision-Based Automated Inspection System in Computer Integrated Manufacturing, **The International Journal of Advanced Manufacturing Technology**, v. 1, n. 3, p. 206-213, 1996.

CHONG, C. W., RAVEENDRAN, P., MUKUNDAN, R. Translation and Scale Invariants of Legendre Moments, **Pattern Recognition**, v. 27, p. 119-129, 2004.

CICHY, R., PANTAZIS, D., OLIVA, A. Mapping Visual Object Recognition in the Human Brain with Combined MEG and fMRI, **Journal of Vision**, v. 13, n. 9, p. 659, 2013.

COVER, T. M., HART, P. E. Nearest Neighbor Pattern Classification, **IEEE Transactions on Information Theory**, v.13, n.1, p. 21-27, 1967.

DÉNIZ, O., CASTRILLÓN, M., HERNÁNDEZ, M. Face Recognition using Independent Component Analysis and Support Vector Machines, **Pattern Recognition Letters**, v. 24, n. 13, p. 2153-2157, 2003.

DUDA, R. O., HART, P. E. Use of the Hough Transform to Detect Lines and Curves in Pictures, **Communications of the ACM**, v. 15, number 1, p. 11-15, 1972

ERDELYI, A., MAGUNUS, W., OBERHETTINGER, F., TRICOMI, F. G. **Higher Transcendental Functions**, New York: McGraw-Hill, v. 2, 1953.

FAN, L., LONG, F., ZHANG, D., GUO, X., WU, X. Applications of Independent Component Analysis to Image Feature Extraction, **Second International Conference on Image and Graphics**, v. 4875, p. 471-476, 2002.

FARIAS, T. **Metodologia para reconstrução 3d baseada em imagens**, 2012. Tese de Doutorado (Ciência da Computação) - Centro de Informática - Universidade Federal de Pernambuco, Recife, 2012.

FEN, L., QINGQI, P., LIAOJUN, P. Robust Image Watermarking Based on Bessel-Fourier Moments, **International Journal of Digital Content Technology and its Applications**, v. 5, n. 11, p. 394, 2011.

FIX, E., HODGES, J. L. Discriminatory Analysis: Small Sample Performance, **USAF School of Aviation Medicine**, Randolph Field, Tex., Project 21-49-004, Rept. 11, 1952.

FIX, E., HODGES, Jr., J. L. Discriminatory Analysis, Nonparametric Discrimination, Consistency Properties, **USAF School of Aviation Medicine**, Randolph Field, Tex., Project 21-49-004, Rcpt. 4, Contract AF41(128)-31, 1951.

FLUSSER, J., SUK, T. A Moment-Based Approach to Registration of Images with Affine Geometric Distortion, **IEEE Transactions on Geoscience and Remote Sensing**, v. 32, n. 2, p. 382-387, 1994.

FLUSSER, J., SUK, T. Rotation Moment Invariants for Recognition of Symmetric Objects, **IEEE Transactions on Image Processing**, v. 15, n. 12, p. 3784-3790, 2006.

FLUSSER, J., SUK, T., ZITOVÁ, B. **Moments and Moment Invariants in Pattern Recognition**, Wiley, Chichester, 2009.

FLUSSER, J., SUK, T., ZITOVÁ, B. On the Recognition of Wood Slices by Means of Blur Invariants, **Sensors and Actuators A: Physical**, v. 198, p. 113-118, 2013.

FREITAS, C. O. A., CARVALHO, J. M., OLIVEIRA, J. J., AIRES, S. B. K., SABOURIN, R. Confusion Matrix Disagreement for Multiple Classifiers, **Progress in Pattern Recognition, Image Analysis and Applications, 12th Iberoamericann Congress on Pattern Recognition, CIARP 2007**, Valparaiso, Chile, v. 4756, p. 387-396, 13-16, 2007.

FRIEDMAN, N., GEIGER, D., GOLDSZMIDT, M. **Bayesian Network Classifiers**, Machine Learning, v. 29, p. 131-163, 1997.

FU, B., ZHOU, J., LI, Y., ZHANG, G., WANG, C. Image Analysis by Modified Legendre Moments, **Pattern Recognition**, v. 40, p. 691-704, 2007.

GAO, G., JIANG, G. Bessel-Fourier Moment-Based Robust Image Zero-Watermarking, **Multimedia Tools and Applications**, p. 1-18, 2013.

GLUD, L. **Spoon Detection and Classification on Conveyor Belts using Computer Vision**, Tese de Mestrado, Universidade do Sul da Dinamarca (SDU) Faculdade de Engenharia, 2010.

GONZAGA, A. Departamento de Engenharia Elétrica, Escola de Engenharia de São Carlos – USP, Transparência da Aula 9 (pdf) da disciplina Visão Computacional. Disponível em: http://iris.sel.eesc.usp.br/sel886/Aula_9.pdf, acessado em: 11 de Março de 2014.

HAROON, M. A., RASUL, G. Principal Component Analysis of Summer Rainfall and Outgoing Long-Wave Radiation over Pakistan, **Pakistan Journal of Meteorology**, v. 5, n. 10, p. 109-114, 2009.

HASTIE, T., TIBSHIRANI, R. In: Becker, S., Obermayer, K. (Eds.), Independent Component Analysis through Product Density Estimation, in **Advances in Neural Information Processing System**, v. 15, MIT Press, Cambridge, MA, p. 649–656, 2003.

HAULE, D. D., MALOWANY, A. S. Object Recognition using Fast Adaptive Hough Transform, **IEEE Pacific Rim Conference on Communications, Computers and Signal Processing**, 1989.

HAYKIN, S. O. **Neural Networks and Learning Machines**, 3^a Edição, Prentice Hall / Pearson, 2009.

HOTELLING, H. Analysis of a Complex of Statistical Variables into Principal Components, **Journal of Educational Psychology**, v. 25, p. 417-441, 1933

HOUGH, P. V. C. Machine Analysis of Bubble Chamber Pictures, **2nd International Conference on High-Energy Accelerators (HEACC 59)**, p. 554-558, 1959.

HOUGH, P. V. C. **Method and Means for Recognizing Complex Patterns**. U. S. Patent n. 3069654, 1962.

HSE, H., NEWTON, A. R. Sketched Symbol Recognition using Zernike Moment, **International Conference on Pattern Recognition**, p. 367-370, 2004.

HU, M. K. Visual Pattern Recognition by Moment Invariants, **IRE, Transaction on information theory**, p. 179-187, 1962.

HUANG, X., WANG, B., ZHANG, L. A New Scheme for Extraction of Affine Invariant Descriptor and Affine Motion Estimation based on Independent Component Analysis, **Pattern Recognition Letters**, v. 26, n. 9, p. 1244-1255, 2005.

HUAZHONG, S., HUI, Z., BEIJING, C., HAIGRON, P., LIMIN, L. Fast Computation of Tchebichef Moments for Binary and Grayscale Images, **IEEE Transactions on Image Processing**, v. 19, n. 12, 2010.

HWANG, S. K., KIM, W. Y. A Novel Approach to the Fast Computation of Zernike Moments, **Pattern Recognition**, v. 39, p. 2065-2076, 2006.

HYVÄRINEN, A. Fast and Robust Fixed-Point Algorithms for Independent Component Analysis, **IEEE Transactions on Neural Networks**, v. 10, p. 626-634, 1999.

HYVÄRINEN, A., KARHUNEN, J., OJA, E. **Independent Components Analysis**. John Wiley & Sons, Inc., Canada (Chapter 6–8), 2001.

HYVÄRINEN, A., OJA. Independent Component Analysis: Algorithms and Applications, **Neural Networks**, v. 13, p. 411-430, 2000.

INRAWONG, P. **Application of PCA and Hough Transform to Classify Features in Optical Images**, Tese de Doutorado, Universidade de Nottingham, 208 p., 2012.

JÚNIOR, M. P. P. **Combinação de Múltiplos Classificadores para Identificação de Materiais em Imagens Ruidosas**, Dissertação de Mestrado, Universidade Federal de São Carlos, 67 p., 2004.

KARCHER, C. **Redes Bayesianas Aplicadas à Análise do Risco de Crédito**, Dissertação de Mestrado, Escola Politécnica da Universidade de São Paulo, Departamento de Engenharia de Sistemas Eletrônicos, 103 p., 2009.

KHOTANZAD, A., HONG, Y. h. Invariant Image Recognition by Zernike Moments, **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v.12, n.5, p.489-497, 1990.

KIM, W. Y., KIM, Y. S. Robust Rotation Angle Estimator, **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 21, n. 8, p. 768-773, 1999.

KIM, W. Y., YUAN, P. A Practical Pattern Recognition System for Translation, Scale and Rotation Invariance, n, p. 391–396, 1994.

KIM, Y. S., KIM, W.Y. Content-Based Trademark Retrieval System Using Visually Salient Feature, Computer Vision and Pattern Recognition, 1997, **Image and Vision Computing**, v. 16, n. 12-13, p. 931-939, 1998.

LIMA, M. A. A., CERQUEIRA, A. S., COURY, D. V., DUQUE, C. A. A Novel Method for Power Quality Multiple Disturbance Decomposition based on Independent Component Analysis, **International Journal of Electrical Power & Energy Systems**, v. 42, n. 1, p. 593-604, 2012.

LIN, T.-Y., CHIU, S.-H. Using Independent Component Analysis and Network DEA to Improve Bank Performance Evaluation, **Economic Modelling**, v. 32, p. 608-616, 2013.

LIU, L. F., JIA, W., ZHU, Y. H. Gait Recognition using Hough Transform and Principal Component Analysis, **Emerging Intelligent Computing Technology and Applications, 5th International Conference on Intelligent Computing, ICIC 2009**, Ulsan, South Korea, v. 5754, p. 363-370, September 16-19, 2009.

MA, X., PAN, R., WANG, L. License Plate Character Recognition Based on Gaussian-Hermite Moments, **Education Technology and Computer Science (ETCS), Second International Workshop**, v. 3, p. 11-14, 2010.

MA, Z. P., KANG, B. S., XIAO, B. A Study of Bessel Fourier Moments Invariants for Image Retrieval and Classification, **Image Analysis and Signal Processing (IASP), 2011 International Conference on**, p. 316-320, 2011.

MAJI, S., MALIK, J. Object Detection using Max-Margin Hough Transform, **IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, p. 1038-1045, 2009.

MALAMAS, E. N., PETRAKIS, E. G. M., ZERVAKIS, M., PETIT, L., LEGAT, J. A Survey on Industrial Vision Systems, **Applications and Tools, Image and Vision computing**, v. 21, p. 171-188, 2003.

MCCULLOCH, W. S., PITTS, W. H. A Logical Calculus of the Ideas Immanent in Nervous Activity, **The Bulletin of Mathematical Biophysics**, v. 5, n. 4, p. 115-133, 1943.

MERCIMEK, M., GULEZ, K., MUMCU, T. V. Real Object Recognition using Moment Invariants, **Sadhana**, v. 30, part 6, p. 765-775, 2005.

MICHIE, D., SPIEGELHALTER, D. J., TAYLOR, C. C. **Machine Learning, Neural and Statistical Classification**, Ellis Horwood, NJ, 1994.

MINDRU, F., TUYTELAARS, T., GOOL, L. V., MOONS, T. Moment Invariants for Recognition Under Changing Viewpoint and Illumination, **Computer Vision and Image Understanding**, v. 94, n. 1, p. 3-27, 2004.

MINSKY, M. L., PAPERT, S. A. **Perceptrons: An Introduction to Computational Geometry**, The MIT Press, Cambridge, 1969.

MITCHELL, T. M. **Machine learning**. WCB/McGraw-Hill, 1997.

MOKHTARIAN, F., ABBASSI, S. Robust Automatic Selection of Optimal Views in Multi-View Free-Form Object Recognition, **Pattern Recognition**, v. 38, n. 7, p. 1021-1031, 2005.

MUKUNDAN, R., ONG, S. H., LEE, P. A. Image Analysis by Tchebichef Moments, **IEEE Transactions on Image Processing**, v. 10, n. 9, p. 1357-1364, 2001.

MUKUNDAN, R., RAMAKRISHNAN, K. R. An Iterative Solution for Object Pose Parameters using Image Moments, **Pattern Recognition Letters**, v. 17, p. 1279-1284, 1996.

NABTI, M., BOURIDANE, A. An Effective and Fast Iris Recognition System Based on a Combined Multiscale Feature Extraction Technique, **Pattern Recognition**, v. 41, n. 3, p. 868-879, 2008.

NAYAK, P. K., CHOLAYYA, N. U. Independent Component Analysis of Electroencephalogram, **IEE Japan Papers of Technical Meeting on Medical and Biological Engineering**, v. 6, number 95-115, p. 25-28, 2006.

NOVOTNI, M., KLEIN, R. Shape Retrieval using 3D Zernike Descriptors, **Computer Aided Design**, v. 36, n. 11, p. 1047-1062, 2004.

PAN, H., ZHU, Y., XIA, L. Efficient and Accurate Face Detection using Heterogeneous Feature Descriptors and Feature Selection, **Computer Vision and Image Understanding**, v. 117, n. 1, p. 12-28, 2013.

PARK, H. D., MITCHEL, O. R. Automated Computer Vision Inspection System for Quick Turnaround Manufacturing, **Proc. SPIE 1004, Automated Inspection and High-Speed Vision Architectures II**, p. 114-125, 1989

PATRICK, E. A., FISCHER III, F. P. A Generalizes k-Nearest Neighbor Rule, **Information and Control**, v. 16, n. 2, p. 128-152, 1970.

QADER, H. A., RAMLI, A. R., HADDAD, S. A. Fingerprint Recognition using Zernike Moments, **The International Arab Journal of Information Technology**, v. 4, n. 4, p. 372-376, 2007.

RANI, S. J., Devaraj, D., Sukanesh, R.. A Novel Feature Extraction Technique for Face Recognition, **Conference on Computational Intelligence and Multimedia Applications, International Conference on**, v. 2, p. 428-435, 2007.

ROJAS, F., GARCÍA, R. V., GONZÁLEZ, J., VELÁZQUEZ, L., BECERRA, R., VALENZUELA, O., B. ROMÁN, S. Identification of Saccadic Components in Spinocerebellar Ataxia Applying an Independent Component Analysis Algorithm, **Neurocomputing**, v. 121, p. 53-63, 2013.

RUMELHART, D. E., HINTON, G. E., WILLIAMS, R. J. Learning Representation by Back-Propagating Errors, **Nature**, v. 323, p. 533-536, 1986.

SANCHETTA, A. C., LEITE, E. P., HONÓRIO, B. C. Z. Facies Recognition using a Smoothing Process Through Fast Independent Component Analysis and Discrete Cosine Transform, **Computers & Geosciences**, v. 57, p. 175-182, 2014.

SELVER, M. A., Akay, O., Alim, F., Bardakçi, S., Olmez, M. An Automated Industrial Conveyor Belt System using Image Processing and Hierarchical Clustering for Classifying Marble Slabs, **Robotics and Computer-Integrated manufacturing**, v. 27, p. 164-176, 2011.

SERPA, T. P. **Um Sistema de Visão para Classificação de Produtos em Transportadores Industriais**, 2014. 60 f. Trabalho de Conclusão de Curso (Engenharia de Teleinformática) – Departamento de Engenharia de Teleinformática, Universidade Federal do Ceará, Fortaleza, 2014.

SHEN, J., Orthogonal Gaussian–Hermite Moments for Image Characterization, **Proceedings of the SPIE The International Society for Optical Engineering, Intelligent Robots and Computer Vision XVI: Algorithms, Techniques, Active Vision, and Materials Handling**, v. 3208, p. 224-233, Pittsburgh, 1997.

SHENG, Y., SHEN, L. Orthogonal Fourier-Mellin Moments for Invariant Pattern Recognition, **Journal of the Optical Society of America A**, v. 11, n. 6, pp1748-1757, 1994.

SILVA, L. M. O. da. **Uma Aplicação de Árvores de Decisão, Redes Neurais e KNN para a Identificação de Modelos ARMA Não-Sazonais e Sazonais**. Tese (Doutorado), Pontifícia Universidade Católica do Rio de Janeiro, 2005.

SILVA, R. D. C., COELHO, D. N., THÉ, G. A. P. A Performance Analysis of Classifiers to Recognition of Objects from low-resolution Images Industrial Sensor, In: **XI Simpósio Brasileiro de Automação Inteligente (SBAI)**, 2013, Fortaleza, *Anais do XI Simpósio Brasileiro de Automação Inteligente*, Fortaleza, 2013c.

SILVA, R. D. C., THÉ, G. A. P. Comparison Between Hough Transform and Moment Invariant to the Classification of Objects from low-resolution Industrial Sensor Images, In: **XI Simpósio Brasileiro de Automação Inteligente (SBAI)**, 2013, Fortaleza, *Anais do XI Simpósio Brasileiro de Automação Inteligente*, Fortaleza, 2013b.

SILVA, R. D. C., THÉ, G. A. P. Moment Invariant based Classification of objects from low-resolution Industrial Sensor Images, In: **11th Brazilian Congress (CBIC), on Computational Intelligence**, 2013, Porto de Galinhas, *Anais do 11th Brazilian Congress on Computational Intelligence*, Porto de Galinhas, 2013a.

SINDHUMOL S., KUMAR, A., BALAKRISHNAN, K. Spectral Clustering Independent Component Analysis for Tissue Classification from Brain MRI, **Biomedical Signal Processing and Control**, v. 8, n. 6, p. 667-674, 2013.

SINGH, C., UPNEJA, R. Accurate Computation of Orthogonal Fourier-Mellin Moments, **Journal of Mathematical Imaging and Vision**, v. 44, n. 3, p. 411-431, 2012.

SIT, A., MITCHEL, J. C., PHILLIPS, G. N., WRIGHT, S. J. An Extension of 3D Zernike Moments for Shape Description and Retrieval of Maps Defined in Rectangular Solids, **Molecular Based Mathematical Biology**, v. 1, p. 75-89, 2013.

SLUZEK, A. Identification and Inspection of 2-D Objects using New Moment-Based Shape Descriptors, **Pattern Recognition Letters**, v. 16, p. 687–697, 1995.

SMEREKA, M., DULEBA, I. Circular Object Detection using a Modified Hough Transform, **International Journal of Applied Mathematics and Computer Science**, v. 18, n. 1, p. 85-91, 2008.

SRIDHAR, D., KRISHNA, Dr I. V. M. Face Recognition using Tchebichef Moments, **International Journal of Information & Network Security**, v. 1, n. 4, p. 243-254, 2012.

STONE, M. Cross-Validatory Choice and Assessment of Statistical Predictions, **Journal of the Royal Statistical Society, Series B (Methodological)**, v. 36, n. 2, p. 111-147, 1974.

TEAGUE, M. R. Image Analysis via the General Theory of Moments, **Journal of the Optical Society of America**, v. 70, n. 8, p. 920–930, 1980.

TIAGRAJAH, V. J., JAMALUDIN, O., FARRUKH, H. N. Discriminant Tchebichef Based Moment Features for Face Recognition, **IEEE International Conference on Signal and Image Processing Applications (ICSIPA)**, p. 192-197, 2011.

TONG, Y., HOCKE, L.M., NICKERSON, L.D., LICATA, S.C., LINDSEY, K.P., FREDERICK, B.D. Evaluating the Effects of Systemic Low Frequency Oscillations Measured in the Periphery on the Independent Component Analysis Results of Resting State Networks, **NeuroImage**, v. 76, p. 202–215, 2013.

TRIPATHY, J. Reconstruction of Oriya Alphabet using Zernike Moments, **International Journal of Computer Applications**, v. 8, n. 18, 2010.

VAPNIK, V. N., LERNER, A. Pattern Recognition using Generalized Portrait Method, **Automation and Remote Control**, v. 24, n. 6, p. 774-780, 1963.

VOROBYOV, M. **Shape Classification using Zernike Moments**, Technical Report. iCamp-Universidade da California Irvine, 2011.

WANG, L., HEALEY, G. Using Zernike Moments for the Illumination and Geometry Invariant Classification of Multispectral Texture, **IEEE Transaction Image Processing**, v. 7, n. 2, p. 196–203, 1998.

WANG, X., GUO, F. X. XIAO, B. Rotation Invariant Analysis and Orientation Estimation Method for Texture Classification Based on Radon Transform and Correlation Analysis, **Journal of Visual Communication and Image Representation**, v. 21, n. 1, p. 29-32, 2010.

WEBB, A. R., COPSEY, K. D. **Statistical Pattern Recognition**, 3^a Edição, Wiley, 2011, p. 666.

XIAO, B., MA, J. F., WANG, X. Image Analysis by Bessel-Fourier Moments, **Pattern Recognition**, v. 43, p. 2620-2629, 2010.

YANG, B., DAI, M. Image Analysis by Gaussian-Hermite Moments, **Signal Processing**, v. 91, n. 10, p. 2290-2303, 2011a.

YANG, B., LI, G., ZHANG, H., DAI, M. Rotation and Translation Invariants of Gaussian-Hermite Moments, **Pattern Recognition Letters**, v. 32, n. 9, p. 1283-1298, 2011b.

YANG, M. H. Kernel Eigenfaces vs. Kernel Fisherfaces: Face Recognition using Kernel Methods, In: **Processings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition**, p. 215-220, 2002.

YANG, W. X., NO, M. E., YING, Y. H. A New SVM-based Image Watermarking using Gaussian-Hermite Moments, **Applied Soft Computing**, v. 12, p. 887-903, 2012.

YAP, P., PARAMEDRAN, R., ONG, S. H. Image Analysis by Krawtchouk Moments, **IEEE Transactions on Image Processing**, v. 12, n. 11, p. 1367–1377, 2003.

YUEN, P. C., LAI, J. H. Face Representation using Independent Component Analysis, **Pattern Recognition**, v. 35, p. 1247-1257, 2002.

ZHANG, F., LIU, S. Q., WANG, D. B., GUAN, W. Aircraft Recognition in Infrared Image using Wavelet Moment Invariants, **Image and Vision Computing**, v. 27, n. 4, p. 313-318, 2009.

ZHANG, K., CHAN, L. W. **ICA by PCA** Approach: Relating Higher-Order Statistics to Second-Order Moments, **Independent Component analysis and Blind Signal Separation, 6th International Conference, ICA 2006**, Charleston, SC, USA, v. 3889, p. 311-318, March 5-8, 2006.

ZHAO, C., ZHANG, B., HE, J. Vision-based Classification of Driving Postures by Efficient Feature Extraction and Bayesian Approach, **Journal of Intelligent and Robotic Systems**, v. 72, n. 3-4, p. 483-495, 2013.

ZHU, H. Q., SHU, H. Z., LIANG J., LUO, L. M., COATRIEUX, J. L. Image Analysis by Discrete Orthogonal dual-Hahn Moments, **Pattern Recognition Letters**, vol 28, n. 13, p. 1688–1794, 2007a.

ZHU, H. Q., SHU, H. Z., LIANG J., LUO, L. M., COATRIEUX, J. L. Image Reconstruction and Compression by Discrete Orthogonal Racah Moments, **Signal Processing**, v. 87, n. 4, p. 687–708, 2007b.