

UNIVERSIDADE CANDIDO MENDES – UCAM  
PROGRAMA DE PÓS-GRADUAÇÃO EM PESQUISA OPERACIONAL E INTELIGÊNCIA  
COMPUTACIONAL  
CURSO DE MESTRADO EM PESQUISA OPERACIONAL E INTELIGÊNCIA  
COMPUTACIONAL

Marlito Pereira de Andrade

USO DE INTELIGÊNCIA COMPUTACIONAL NO RECONHECIMENTO  
ÓPTICO DE CARACTERES

CAMPOS DOS GOYTACAZES – RJ  
Março de 2015

UNIVERSIDADE CANDIDO MENDES – UCAM  
PROGRAMA DE PÓS-GRADUAÇÃO EM PESQUISA OPERACIONAL E INTELIGÊNCIA  
COMPUTACIONAL  
CURSO DE MESTRADO EM PESQUISA OPERACIONAL E INTELIGÊNCIA  
COMPUTACIONAL

Marlito Pereira de Andrade

## USO DE INTELIGÊNCIA COMPUTACIONAL NO RECONHECIMENTO ÓPTICO DE CARACTERES

Dissertação apresentada ao Programa de Pós-Graduação em  
Pesquisa Operacional Inteligência Computacional, da  
Universidade Candido Mendes–Campos/RJ, para obtenção do  
grau de MESTRE EM PESQUISA OPERACIONAL E  
INTELIGÊNCIA COMPUTACIONAL.

Orientador: Prof. Ítalo de Oliveira Matias, DSc.

CAMPOS DOS GOYTACAZES – RJ  
Março de 2015

## FICHA CATALOGRÁFICA

A553u Andrade, Marlito Pereira de.

Uso de inteligência computacional no reconhecimento óptico de caracteres/ Marlito Pereira de Andrade. – 2016.  
120 f.; il.

Orientador: Ítalo de Oliveira Matias.

Dissertação de Mestrado em Pesquisa Operacional e Inteligência Computacional – Universidade Candido Mendes – Campos. Campos dos Goytacazes, RJ, 2015.

Bibliografia: f. 73-80.

1. Inteligência computacional – caracteres (reconhecimento óptico). 2. Robótica inteligente. 3. Redes neurais. 4: Máquinas de vetores de suporte. Universidade Candido Mendes – Campos. II. Título.

CDU: 004.896

MARLITO PEREIRA DE ANDRADE

USO DE INTELIGÊNCIA COMPUTACIONAL NO RECONHECIMENTO  
ÓPTICO DE CARACTERES

Dissertação apresentada ao Programa de Pós-Graduação em  
Pesquisa Operacional Inteligência Computacional, da  
Universidade Candido Mendes–Campos/RJ, para obtenção do  
grau de MESTRE EM PESQUISA OPERACIONAL E  
INTELIGÊNCIA COMPUTACIONAL.

Aprovado em: março de 2015.

BANCA EXAMINADORA

---

Prof. Ítalo de Oliveira Matias, DSc. – Orientador.  
Universidade Candido Mendes

---

Prof. Dalessandro Soares Vianna, DSc.  
Universidade Candido Mendes

---

Prof.<sup>a</sup>. Marcilene de Fátima Dianin Vianna, DSc.  
Universidade Federal Fluminense

CAMPOS DOS GOYTACAZES – RJ  
2015

Dedico este Trabalho de Conclusão de Curso de Pós-Graduação (Mestrado) primeiramente a Deus, pois sem ele nada seria possível.

À minha esposa Denizeti e aos meus filhos pela compreensão e apoio nesta importante etapa de minha vida.

Aos meus pais e todos os meus irmãos pelo esforço, dedicação e compreensão, em todos os momentos desta caminhada.

## **AGRADECIMENTOS**

Primeiramente a Deus que me deu tudo, em especial, o Dom da Vida.

Agradeço a todas as pessoas que de alguma forma contribuiu e contribui para o meu crescimento pessoa e profissional.

Ao professor Italo de Oliveira Matias que me auxiliou e direcionou com sua vasta sabedoria e experiência. Ao Professor Dr. Dalessandro Vianna, Eduardo Shimoda, Aldo Shimoia e demais professores e colegas de curso.

Aos funcionários da UCAM, pelo apoio e bom trabalho realizado que me auxiliou sempre que necessário, em especial, à Aparecida Silva Pessanha “Cida”, que sempre me atendeu com muita presteza.

À minha irmã Tereza Pereira de Andrade que sempre acreditou e incentivou em buscar o melhor.

Aos meus amigos Nivaldo Antônio da Silva, Jocimar Fernandes, Marcelo Costalonga, Flavio Quirino do Nascimento e, em especial, ao grande amigo Ubiratan Roberte Cardoso Passos que fez o convite e me fez companhia por vários meses indo a UCAM CAMPOS e voltando, pois moramos no Espírito Santo e fazíamos esse trajeto quase que semanalmente, sem uma companhia seria muito difícil.

## RESUMO

### USO DE INTELIGÊNCIA COMPUTACIONAL NO RECONHECIMENTO ÓPTICO DE CARACTERES

Este trabalho apresenta um estudo de técnicas de inteligência computacional aplicado no reconhecimento óptico de caracteres OCR (Optical Character Recognition). Estuda-se a tarefa de classificação realizada pelos classificadores - Máquinas de Vetores de Suporte (SVM), Redes Neurais Artificiais (RNAs) - e algoritmo dos  $k$  vizinho mais próximo (K-Nearest Neighbor - KNN), com o objetivo de selecionar as características dos padrões que são caracteres ópticos presentes nas imagens e otimizar a configuração dos parâmetros, a fim de obter o melhor desempenho de classificação possível. O reconhecimento de caracteres é uma das várias aplicações em Reconhecimento de Padrões (RP). Há duas terminologias que são comumente empregadas para classificar os sistemas capazes de efetuar a leitura automática de caracteres e textos escritos à máquina ou manuscritos, em letra de forma ou cursivos. O termo mais antigo é OCR e o segundo é ICR que é um acrônimo para Intelligent Character Recognition. Para avaliar os algoritmos desenvolvidos foi utilizada uma base de dados com 20.000 vetores de características, juntamente com as respostas de letras latinas A-Z e, desse conjunto de dados, 50% foram utilizados para treinamento e os outros 50% para testes. Ao final dos testes, os resultados obtidos foram satisfatórios, com taxa de acerto superior a 98% ao classificador SVM, utilizando função kernel RBF, já o classificador RNA obteve taxa de acerto superior de 97%, ao passo que o classificador KNN obteve taxa de acerto superior de 93%.

**PALAVRAS-CHAVES:** Reconhecimento Óptico de Caracteres. OCR (*Optical Character Recognition*). Redes Neurais Artificiais (RNA). Perceptrons Multicamadas (MLP). Algoritmo dos  $k$  vizinhos mais próximos (*K-Nearest Neighbor* - KNN). Máquinas de Vetores de Suporte (SVM).

## **ABSTRACT**

### **INTELLIGENCE COMPUTER USE IN OPTICAL CHARACTER RECOGNITION**

This work presents a computational intelligence techniques applied to the study of optical recognition of characters OCR (Optical Character Recognition). We study the classification task performed by the classifiers - Support Vector Machines (SVM), Artificial Neural Networks (RNAS) - and algorithm of k nearest neighbor (K-Nearest Neighbor - KNN), with the objective of selecting the features the optical character patterns that are present in the images, and optimize the configuration parameters in order to obtain the best classification performance. The character recognition is one of several applications in Pattern Recognition (PR). There are two terms that are commonly used to classify systems capable of performing automatic reading of characters and texts typewritten or handwritten in block letters or cursive. The oldest term is OCR and ICR is the second which is an acronym for Intelligent Character recongnition. To evaluate the developed algorithms we used a database with 20,000 feature vectors, along with the responses of Latin letters AZ, and that dataset, 50% was used for training and 50% for testing. At the end of the tests, the results were satisfactory, with accuracy rate exceeding 98% per classifier SVM using RBF kernel function, since the RNA classifier obtained higher hit rate of 97%, while the KNN classifier obtained rate top hit 93%.

**KEYWORDS:** Optical Character Recognition. OCR (Optical Character Recognition). Artificial Neural Networks (ANN). Multilayer Perceptron (MLP). Algorithm of k-nearest neighbors (K-Nearest Neighbor - KNN). Support Vector Machines (SVM).

## LISTA DE FIGURAS

<b>Figura 1:</b>	Componentes de um sistema OCR.	23
<b>Figura 2:</b>	Os principais paradigmas da Inteligência Computacional.	27
<b>Figura 3:</b>	Esquema de um sistema e uma máquina de aprendizado.	29
<b>Figura 4:</b>	Rede Feedforward de Múltiplas Camadas.	31
<b>Figura 5:</b>	Processo de treinamento do algoritmo back propagation.	32
<b>Figura 6:</b>	Exemplo de hiperplano ótimo linearmente separáveis.	34
<b>Figura 7:</b>	Distância $d$ entre os hiperplanos $H1$ e $H2$ .	35
<b>Figura 8:</b>	Exemplo de classificação pelo método KNN.	38
<b>Figura 9:</b>	Representação de Imagens Digitais.	39
<b>Figura 10:</b>	Processo de Discretização de uma Imagem.	40
<b>Figura 11:</b>	Imagem digital em tons de cinza.	41
<b>Figura 12:</b>	Exemplo de limiarização com limiar = "128.	42
<b>Figura 13:</b>	Passos para o processamento digital de imagens.	44
<b>Figura 14:</b>	Linha e detecção dos limites dos caracteres.	50
<b>Figura 15:</b>	Confirmação dos limites dos caracteres.	51
<b>Figura 16:</b>	Caracteres segmentados utilizados como treinamento.	54
<b>Figura 17:</b>	Caracteres reconhecidos.	54
<b>Figura 18:</b>	Amostra de possíveis entradas para os métodos.	57
<b>Figura 19:</b>	Rede MLP do Projeto.	63

## LISTA DE GRÁFICOS

<b>Gráfico 1:</b>	Classificação KNN com variação do valor de K.	58
<b>Gráfico 2:</b>	Classificação SVM com função Kernel Linear.	59
<b>Gráfico 3:</b>	Classificação SVM com função Kernel Polinomial.	61
<b>Gráfico 4:</b>	Classificação SVM com função Kernel RBF.	62
<b>Gráfico 5:</b>	Representação gráfica da RNA para a variação no número de épocas.	64
<b>Gráfico 6:</b>	Gráfico da comparação direta entre os métodos.	66
<b>Gráfico 7:</b>	Gráfico de intervalos KNN, SVM RBF e RNA.	67
<b>Gráfico 8:</b>	Gráfico de Diferença de Médias para KNN, SVM RBF e RNA.	69
<b>Gráfico 9:</b>	Gráfico Resultado Especificidade e Sensibilidade.	69

## LISTA DE TABELAS

<b>Tabela 1:</b>	Resultados para a variação do valor de K.	57
<b>Tabela 2:</b>	Resultados da Classificação por SVM com função Kernel Linear.	59
<b>Tabela 3:</b>	Resultados da Classificação por SVM com função Kernel Polinomial.	60
<b>Tabela 4:</b>	Resultados da Classificação por SVM com função Kernel RBF.	61
<b>Tabela 5:</b>	Resultados da Classificação por SVM.	62
<b>Tabela 6:</b>	Resultados para a variação no número de épocas.	64
<b>Tabela 7:</b>	Pareamento dos resultados e desvio padrão dos métodos.	67
<b>Tabela 8:</b>	Cálculo da Variância.	68
<b>Tabela 9:</b>	Análise de Variância e Classificação por Teste de Tukey.	68

## LISTA DE EQUAÇÕES

<b>Equação 1:</b>	Equação de Superfície de Decisão.	34
<b>Equação 2:</b>	Equação com Base nas Classes $y = +1$ e $y = -1$ (1).	34
<b>Equação 3:</b>	Equação com Base nas Classes $y = +1$ e $y = -1$ (1).	34
<b>Equação 4:</b>	Equação de Representação de uma Superfície de Decisão Linear Multidimensional no Espaço de Entrada.	35
<b>Equação 5:</b>	Equação de Distância entre os Hiperplanos $H_1$ e $H_2$ .(1)	36
<b>Equação 6:</b>	Equação de Distância entre os Hiperplanos $H_1$ e $H_2$ .(2)	36
<b>Equação 7:</b>	Equação para Obter o Comprimento do Vetor Projetado.	37
<b>Equação 8:</b>	Equação da Distância Euclidiana.	41
<b>Equação 9:</b>	Equação da Conversão dos Tons de Cinza.	41
<b>Equação 10:</b>	Equação da Definição dos Tons de Cinza.	41
<b>Equação 11:</b>	Equação de Limiarização.	43

## LISTA DE ABREVIATURAS E SIGLAS

- ANFIS - O sistema de inferência *neuro-fuzzy* adaptativo
- HMM - *Hidden Markov Model*
- IC - Inteligência Computacional
- ICR - *Intelligent Character Recognition* (Reconhecimento Inteligente de Caracteres)
- IWR - *Intelligent Word Recognition* (*Reconhecimento Inteligente de Palavra*)
- KNN - K-Nearest Neighbor (k vizinho mais próximo)
- LMS - *Least Mean Square* (Algoritmo do mínimo quadrado médio)
- MA - Máquina de aprendizagem
- MLP- Multi Layer Perceptrons (Perceptrons Multicamadas)
- OCR - *Optical Character Recognition* (Reconhecimento Óptico de Caracteres)
- OMR - *Optical Mark Recognition* (Reconhecimento Óptico de Marca)
- OpenCV - *Open Source Computer Vision Library*
- PSO - *Particle Swarm Optimization* (Otimização de enxame de partículas)
- RBF- *Radius Basis Function* (Função de Base Radial)
- RGB - *Red, Green and Blue* (Vermelho, Verde e Azul)
- RNA - Redes Neurais Artificiais
- RP- Reconhecimento de Padrões
- SOM - *Self Organizing Map* (Mapa Auto Organizável de Kohonen)
- SVM - *Support Vector Machines* (Máquina de Vetor de Suporte)
- UCI - *Repository of Machine Learning* (Repositório de Aprendizado de Máquina)

## SUMÁRIO

<b>1:</b>	<b>INTRODUÇÃO.</b>	15
1.1:	MOTIVAÇÃO.	18
1.2:	OBJETIVOS.	19
<b>1.2.1:</b>	<b>Objetivo Geral.</b>	19
<b>1.2.2:</b>	<b>Objetivos Específicos.</b>	19
1.3:	ORGANIZAÇÃO DO TRABALHO.	20
<b>2:</b>	<b>FUNDAMENTAÇÃO TEÓRICA.</b>	22
2.1:	TRABALHOS RELACIONADOS.	23
2.2:	INTELIGÊNCIA COMPUTACIONAL.	26
2.3:	APRENDIZADO DE MÁQUINA.	27
2.4:	REDES NEURAIS ARTIFICIAIS.	29
<b>2.4.1:</b>	<b>Perceptrons Multicamadas (MLP).</b>	30
2.4.1.1:	Algoritmo de treinamento.	31
2.5:	MÁQUINAS DE VETORES SUPORTE.	32
2.6:	ALGORITMO DOS K VIZINHOS MAIS PRÓXIMOS (KNN).	36
2.7:	COMPUTAÇÃO GRÁFICA.	38
<b>2.7.1:</b>	<b>Imagem digital.</b>	39
<b>2.7.2:</b>	<b>Processamento Digital de Imagens.</b>	43
<b>3:</b>	<b>METODOLOGIA.</b>	46
3.1:	DETECÇÃO DE SÍMBOLO NA IMAGEM.	47
<b>3.1.1:</b>	<b>Determinar Linhas de Caractere.</b>	48
3.1.1.1:	Algoritmo (1).	48

<b>3.1.2:</b>	<b>Detecção de Símbolos Individuais.</b>	49
3.1.2.1:	Algoritmo (2).	49
3.2:	ATIVIDADES DESENVOLVIDAS PARA O CASO ESPECÍFICO DO PROJETO.	51
<b>4:</b>	<b>RESULTADOS E DISCUSSÕES.</b>	56
4.1:	CLASSIFICADOR KNN.	57
4.2:	CLASSIFICADOR SVM.	58
<b>4.2.1:</b>	<b>SVM Kernel Linear.</b>	58
<b>4.2.2:</b>	<b>SVM Kernel Polinomial.</b>	60
<b>4.2.3:</b>	<b>SVM Kernel RBF.</b>	61
4.3:	CLASSIFICADOR RNA.	63
<b>4.3.1:</b>	<b>Influência da variação de parâmetros.</b>	65
4.4:	COMPARAÇÃO ESTATÍSTICA ENTRE OS MÉTODOS.	66
<b>5:</b>	<b>CONCLUSÕES.</b>	71
<b>6:</b>	<b>REFERÊNCIAS BIBLIOGRÁFICAS.</b>	73
	<b>APÊNDICE A: USO DE REDES NEURAIS PERCEPTRONS MULTICAMADAS (MLP) EM SISTEMA DE RECONHECIMENTO ÓPTICO DE CARACTERES</b>	80
	<b>APÊNDICE B: SISTEMA DE SEGMENTAÇÃO DE IMAGEM DE CHAPAS DE GRANITO POLIDAS PARA GERAR UM BANCO DE IMAGENS COM PADRÕES DE DEFEITOS EM GRANITOS.</b>	81

## 1 INTRODUÇÃO

A troca de dados entre seres humanos e máquinas é um desafio que, basicamente, implica compreender que reconhecimento de caracteres é um processo que associa um significado simbólico com objetos desenhados em uma imagem (letras, símbolos e números), ou seja, é a técnica de reconhecimento de caracteres associam uma identidade simbólica à imagem de um carácter.

Portanto, o maior desafio na área de processamento de imagem consiste em reconhecer documentos, seja no formato impresso ou no manuscrito. Isso porque o reconhecimento de caracteres é uma das áreas mais amplas e nessa são utilizadas características biométricas para autenticação de pessoa, bem como de documento. O reconhecimento óptico de caracteres é um tipo de análise de imagem de documento no qual a imagem digitalizada contém impressos à máquina ou escritos à mão e a entrada em um software OCR converte para um formato de texto digital legível, editável por máquina. (VASUDEVA *et al.*, 2012).

Pal e Chaudhuri (2004) afirmam que a tecnologia OCR converte as imagens de caracteres impressos por máquinas em caracteres legíveis por máquinas. Isso porque os Sistemas OCR visam permitir que máquinas reconheçam símbolos ópticos sem intervenção humana. As técnicas de extração de caracteres e reconhecimento têm potencial aplicação em qualquer domínio onde as imagens de documentos de textos precisem ser interpretadas, analisadas e processadas (JAWAHAR *et al.*, 2002). Assim, a seleção da estratégia de segmentação, os recursos e o classificador são passos muito importantes na construção de um OCR.

De acordo com Kumbhar e Chandrachud (2012), o reconhecimento óptico de caracteres aborda o problema de processar e reconhecer caracteres ópticos. O

reconhecimento óptico é realizado *off-line* após a escrita ou a impressão ser concluída, ao contrário de reconhecimento *online* em que o computador reconhece os caracteres à medida que são desenhados. Ambos os caracteres - escritos à mão e impressos - podem ser reconhecidos, mas o desempenho é diretamente dependente da qualidade dos documentos de entrada. Quanto mais restrita a entrada for, melhor será o desempenho do sistema OCR. No entanto, quando se trata de escrita manual, totalmente irrestrita, as máquinas OCR ainda estão muito longe de ler, bem como seres humanos. No entanto, o computador lê, com rapidez, dispositivos técnicos e estão trazendo continuamente a tecnologia mais perto de seu ideal.

De acordo com Pradeep *et al.* (2010), as tecnologias mais comuns de reconhecimento padrão são ICR, OCR, IWR (*Intelligent Word Recognition*), OMR (*Optical Mark Recognition*), etc., mas dentre esses, o OCR é a tecnologia mais prevalente. Nos últimos anos, a tecnologia OCR tem sido aplicada em diversas áreas, revolucionando o processo de gestão de documento. O uso de OCR varia de acordo com as diferentes áreas. Os bancos o OCR são usados para controlar processos sem necessidade de envolvimento humano, como nos cuidados de saúde, no processamento de grandes volumes de formulários para cada paciente, em formulários de seguro, etc. OCR também é usado em muitas outras áreas como: educação, finanças, agências governamentais, etc..

Segundo Neeba *et al.* (2009), o reconhecimento óptico de caracteres é um campo de interesse que se evidencia por mais de uma década e é uma pesquisa ativa no reconhecimento de padrões. Tem sido proposto um grande número de algoritmos para o mesmo domínio, mas o que mais importa são a eficiência do algoritmo e sua viabilidade.

Segundo Osório (1991), os sistemas OCR são sistemas desenvolvidos, de certa forma, para reproduzir a capacidade humana de ler textos. Além de servirem como uma forma alternativa de interação homem-máquina, os sistemas de reconhecimento de caracteres têm sido utilizados para a compactação de imagens textuais. Esse caso se aplica muito na utilização de sistemas de *fac-símile*, para transmissão de texto. Através do reconhecimento dos caracteres, em uma imagem a ser transmitida por este tipo de equipamento, pode-se obter a compactação das informações, o que resulta em economia (menor número de informações transmitidas).

A técnica OCR estabeleceu as bases e a motivação para tornar o reconhecimento de padrões e a análise de imagens como campos individuais de interesse da ciência (RODRIGUES; THOMÉ, 2000).

A técnica de reconhecimento de caracteres se baseia na extração das características de uma imagem de um documento digitalizado para fazer a classificação das características obtidas. Um fator determinante para um bom desempenho do reconhecimento é a seleção do conjunto de características a serem extraídas dos caracteres (AIRES, 2005).

Segundo Vasudeva *et al.* (2012), a ferramenta de reconhecimento de caracteres é uma dessas ferramentas inteligentes para adquirir inteligência humana parcial, com a capacidade de capturar e reconhecer vários caracteres em diferentes idiomas. Isso porque o reconhecimento de caracteres (em geral, o reconhecimento de padrões) aborda o problema da classificação de dados de entrada, representado como vetores, por categorias. (LECUN; BENGIO, 1995).

De acordo com Arora *et al.* (2010), ao longo dos anos, a informatização assumiu grande número de operações manuais, um bom exemplo é o reconhecimento de caracteres que consiste na capacidade de um sistema de computador de receber e interpretar imagens digitalizadas de caracteres. No estágio inicial do desenvolvimento do OCR, foram utilizadas técnicas de reconhecimento baseado em modelo de correspondência. Os métodos iniciais foram concebidos, artificialmente, em modelos ou protótipos selecionados a partir de uma média ou de algumas amostras. À medida que o número de amostras aumentou essa metodologia de projeto simples se tornou insuficiente para acomodar a variabilidade sob forma de amostras e, assim, não foi capaz de gerar altas precisões no reconhecimento.

Os métodos de classificação de padrões, baseados na aprendizagem de exemplos, têm sido amplamente aplicados para reconhecimento de caracteres desde 1990 e têm trazido melhorias significativas de precisões de reconhecimento. Esses tipos de métodos incluem métodos estatísticos, redes neurais artificiais, máquinas de vetor de suporte, múltipla combinação classificador. (ARORA *et al.*, 2010).

Neste projeto, estudam-se as características de alguns métodos de inteligência computacional que têm sido aplicados com sucesso no reconhecimento de caracteres ópticos e os resultados de RNA's, SVM e KNN aplicados no

reconhecimento de caracteres ópticos. Os pontos fortes e as deficiências desses métodos de classificação também serão discutidos.

### 1.1: MOTIVAÇÃO

Na visão computacional, pode-se perceber um significativo aumento nos vários campos a serem desenvolvidos, o desafio é resolver o problema relacionado à visão computacional, produzindo técnicas inteligentes que reproduzam as capacidades do sistema visual humano, reagindo aos estímulos visuais de acordo com a área de pesquisa investigada. Nesse sentido, a identificação das funções do sistema visual dos seres humanos, o processo de aprendizagem e a capacidade de realizar inferências nos processos da pesquisa são de grande relevância para o desenvolvimento e a melhoria de técnicas para análise automática ou semiautomática de informações, sejam essas extraídas de uma imagem ou cena com o auxílio de computador (GUIMARÃES, 2013).

Através do trabalho com imagens é possível desenvolver e obter muitos dados que, usados de forma objetiva, podem trazer benefícios em vários contextos.

No início da década de 60, surgiu o termo reconhecimento de padrões que a princípio significava a detecção de formas simples. De acordo Hugo (1995), Duda e Hart (1973) deram uma definição interessante para o termo "Reconhecimento de padrões é um campo interessado no reconhecimento por máquinas de regularidades significativas em ambientes ruidosos ou complexos", ou a procura por uma estrutura nos dados.

Na visão computacional, juntamente com redes neurais, têm-se estudos sobre reconhecimento de padrões. O que para os seres humanos é uma habilidade simples e extremamente desenvolvida - por se tratar de reconhecer objetos do dia a dia, como: face, fisionomias, caligrafias e até mesmo estado de humor das pessoas que conhecem - no ambiente de computadores isso se torna um pouco mais complexo, por se tratar de algoritmos e técnicas que permitam que um computador aprenda, ou seja, que ele melhore o seu desempenho em determinada tarefa.

Diante desses pressupostos, para classificar as imagens e obter um padrão de reconhecimento, deve-se saber - previamente - para qual finalidade se almeja obter as informações. Segundo Mossman (2010), algumas áreas se destacam na

aplicação do reconhecimento de padrões, como: identificação através de impressões digitais (PEDRINI; SCHWARTZ, 2008) e análise da íris, diagnósticos médicos (STEINER, 1995), análise de imagens aeroespaciais, visão computacional (PERELMUTER et al., 1995), investigação da qualidade do papel industrial (STEINER, 1995), processamento de imagens (ALBUQUERQUE; ALBUQUERQUE, 2000), análise de peças para manutenção preventiva, análise de caracteres manuscritos (PRADO; ELFES, 1975), análise de eletrocardiogramas (MASCARENHAS, 1987), reconhecimento e identificação de cromossomos (TODESCO, 1995), dentre tantas outras.

Entretanto, as dificuldades de extrair as informações, classificar a imagem e definir um padrão estão na definição de quais técnicas e parâmetros serão utilizados, já que para cada objetivo de uso final da imagem há grande variedade de técnicas, podendo apresentar resultados totalmente diferentes de acordo com a técnica que se opte por aplicar.

A possibilidade de automatização desse processo de definição da ordem das técnicas e métodos que são aplicadas em cada imagem, para obter um objetivo final e uso da mesma, é que motiva a realização deste projeto. Através do desenvolvimento de Algoritmo de Redes Neurais (RNA's), Máquina de Vetores de Suporte (SVM) e k Vizinhos Mais Próximos (K-Nearest Neighbor – KNN), acredita-se que é possível obter sucesso para o objetivo final da dissertação.

## 1.2: OBJETIVOS

### 1.2.1: Objetivo Geral

O objetivo principal deste trabalho é estudar as técnicas de reconhecimento de padrões com uso de inteligência computacional no reconhecimento e classificação de caracteres ópticos.

### 1.2.2: Objetivos Específicos

Os objetivos específicos são avaliar, computacionalmente, Redes Neurais Artificiais (RNA's), Máquina de Vetores de Suporte (SVM) e k-Vizinhos Mais

Próximos (K-Nearest Neighbor - KNN) como método de classificação para a integração de informações presentes nas imagens ópticas de caracteres. Dentre esses objetivos estão:

- Revisar as questões e técnicas para OCR, que são:
  - Passo 1: Pré-processamento;
  - Passo 2: Pré-Classificação;
  - Passo 3: Extração de Características.
  
- Avaliar o potencial das Redes Neurais Artificiais (RNA's) como método de classificação para a integração de informações presentes nas imagens ópticas de caracteres.
  
- Descrever o potencial de Máquina de Vetores de Suporte (SVM) como método de classificação para a integração de informações presentes nas imagens ópticas de caracteres.
  
- Analisar o algoritmo dos k Vizinhos Mais Próximos (K-Nearest Neighbor – KNN) como método de classificação para a integração de informações presentes nas imagens ópticas de caracteres.

### 1.3: ORGANIZAÇÃO DO TRABALHO

O trabalho apresentado está organizado em 5 (cinco) capítulos e dois anexos conforme descritos a seguir.

**Capítulo 1: Introdução:** faz uma breve introdução do conteúdo que compõe o presente trabalho, apresentando a descrição do problema, a motivação, justificativa, os objetivos e a organização do trabalho;

**Capítulo 2: Fundamentação Teórica:** descreve, de forma sucinta, o conteúdo que compõe o presente trabalho bem como as fontes de dados e as principais referências;

**Capítulo 3: Metodologia:** apresenta o processo de elaboração do trabalho em suas fontes de dados e os aspectos de implementação utilizados para o desenvolvimento do sistema;

**Capítulo 4: Resultados:** descreve os resultados dos testes e treinamentos realizados com as diferentes técnicas de inteligência computacional no reconhecimento de caracteres.

**Capítulo 5: Conclusões:** Apresenta as conclusões obtidas a partir do desenvolvimento do projeto, fazendo uma análise retrospectiva e trazendo a perspectiva para trabalhos futuros que possam ser implementados, tendo como base este projeto.

**Apêndice A:** Artigo aprovado no **IX** Encontro Mineiro de Engenharia de Produção – EMEPRO.

**Apêndice B.** Artigo aprovado no **X** Encontro Mineiro de Engenharia de Produção – EMEPRO.

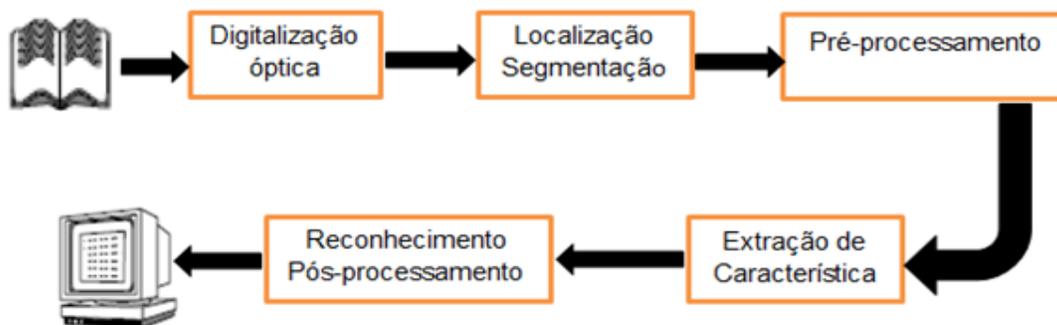
**A Seção Referências Bibliográficas:** detalha as bibliografias utilizadas no decorrer do período de estudos e as pesquisas adotadas no desenvolvimento do projeto, sendo que essas possuem grande importância para a consolidação de conceitos e conhecimentos.

## 2: FUNDAMENTAÇÃO TEÓRICA

A reprodução de funções humanas por máquina, como a leitura, é um sonho antigo. No entanto, ao longo das últimas cinco décadas, a leitura de máquina passou de um sonho para a realidade. Isso porque o reconhecimento óptico de caracteres se tornou uma das mais bem-sucedidas aplicações da tecnologia na área de reconhecimento de padrões e inteligência artificial. (TIWARI *et al.*, 2013).

A técnica de reconhecimento de caracteres se baseia na extração das características de uma imagem de documento digitalizado para fazer a classificação das características obtidas. Um fator determinante para um bom desempenho do reconhecimento é a seleção do conjunto de características a serem extraídas dos caracteres. (AIRES, 2005).

De acordo com Tiwari *et al.* (2013), um típico sistema de OCR é constituído por vários componentes. O primeiro passo, nesse processo, é digitalizar o documento utilizando um scanner óptico. Quando as regiões que contêm texto são localizadas, cada símbolo é extraído por meio de um processo de segmentação. Os símbolos extraídos podem, então, ser pré-processados, eliminando ruído, para facilitar a extração das características na sequência de passos. Em seguida, a identidade de cada símbolo é encontrada - por comparação das características extraídas com descrições das classes símbolos obtidos- através de uma fase de aprendizagem anterior. Finalmente, a informação contextual é usada para reconstruir as palavras e/ou números do texto original. A figura 1 ilustra uma configuração comum de um típico sistema de OCR.



**Figura 1:** Componentes de um sistema OCR.

Fonte: Adaptado de Tiwari (*et al.*, 2013)

De acordo com Pradeep *et al.* (2010), as tecnologias mais comuns de reconhecimento padrão são ICR, OCR, IWR (*Intelligent Word Recognition*), OMR (*Optical Mark Recognition*), mas dentre essas, a OCR é a tecnologia mais prevalente. Nos últimos anos, a tecnologia OCR tem sido aplicada em diversas áreas, revolucionando o processo de gestão de documento. O uso de OCR varia de acordo com as diferentes áreas. Nos bancos, por exemplo, a OCR é usada para controlar processos sem o envolvimento de humano; nos cuidados de saúde, é usada para processar grandes volumes de formulários por paciente, formulário de seguro. A OCR também é usada em muitas outras áreas como: educação, finanças, agências governamentais, etc.

O reconhecimento óptico de caracteres é um campo de pesquisa em reconhecimento de padrões, inteligência artificial, visão de máquina e processamento de sinais. Esse tipo de reconhecimento é, normalmente, referido como um processo de reconhecimento de caracteres *off-line*, o que significa dizer que o sistema processa e reconhece imagens estáticas de caracteres. Refere-se à tradução mecânica ou eletrônica de imagens de caracteres manuscritos ou textos impressos em código de máquina, sem qualquer variação. (BHATIA, 2014).

## 2.1: TRABALHOS RELACIONADOS

Bhatia (2014) apresenta uma revisão detalhada da área de reconhecimento óptico de caracteres, enfatizando as várias técnicas que têm sido propostas para os sistemas de reconhecimento ópticos de caracteres.

Bayat (2013) apresenta o desenvolvimento do projeto de um sistema com alta eficiência para reconhecimento de dígitos manuscritos. Primeiramente, propõe um sistema eficiente que inclui dois módulos principais: o módulo de extração de características e o módulo classificador. No módulo de extração de característica sete conjuntos de características discriminantes são extraídos e utilizados no sistema de reconhecimento. No módulo classificador, pela primeira vez nesta área, o sistema de inferência *neuro-fuzzy* adaptativo (ANFIS) é investigado. Os resultados experimentais mostram que o sistema proposto tem uma boa precisão de reconhecimento.

No entanto, os resultados iniciais mostram que, no treinamento ANFIS, o raio do vetor tem papel muito importante para precisão de reconhecimento. No segundo momento, propõe um sistema de inteligência com um novo módulo de otimização, ou seja, melhora no algoritmo IBA que é proposto para encontrar os melhores parâmetros do classificador. Nessa fase de teste, o método de validação cruzada foi aplicado ao banco de dados numérico manuscrito MNIST para avaliar os desempenhos do sistema proposto. Os resultados das simulações mostraram que o sistema proposto teve alta precisão de reconhecimento.

Nasir e Uddin (2013) desenvolveram um método que com base em pré-processamento, *k-means clustering*, teorema Bayesiano e SVM. O número de amostra de dígitos foi de 300 e a taxa de sucesso alcançado foi de 99,33%. Sahu e Kubde (2013) relatam que os métodos de classificação, baseados na aprendizagem de exemplos, têm sido amplamente aplicados para reconhecimento de caracteres desde 1990 e eles têm trazido melhorias significativas de precisão no reconhecimento. Isso porque essa classe de métodos inclui métodos estatísticos, redes neurais artificiais, máquinas de vetor suporte e classificador de combinação múltipla. Assim, o presente trabalho detalha as características dos métodos de classificação que têm sido aplicados com sucesso para o reconhecimento de caracteres e aponta os problemas pendentes que podem ser - potencialmente - resolvidos pelos métodos de aprendizagem.

Singh *et al.* (2013) apresentam uma análise detalhada no domínio do reconhecimento *off-line* de caracteres manuscritos, pois segundo os autores citados o reconhecimento de escrita manual, no entanto, pode ser considerado um problema ainda aberto, devido à sua variação substancial na aparência. Embora, tenham sido realizados estudos suficientes, o trabalho descreve as técnicas para a conversão de

conteúdo textual, partindo de um documento em formato de papel para o formato digital. Desse modo, o reconhecimento *off-line* de caracteres manuscritos é um processo em que o computador entende automaticamente a imagem de manuscrito. Este material serve como um guia e atualização para os leitores que trabalham na área de reconhecimento de caracteres.

Abaynarh *et al.* (2012) apresentam um reconhecedor de caracteres Amazighe manuscrito, irrestrito, baseado em momentos ortogonais e classificador de redes neurais. Nesse, o resultado mostra que, se o número de nós escondidos propicia o aumento do número de épocas (iterações), também há aumento se for levado em consideração o carácter manuscrito representado. O sistema proposto extraiu as características momentos a partir de imagens de caracteres e a precisão obtida foi 97,46%.

Sok e Taing (2014) descrevem o uso do método de classificação SVM para reconhecimento de caracteres *Khmer* impresso em documento *bitmap*. A língua *Khmer* foi identificada como uma das línguas mais complexas, essa possui um total de 74 alfabetos e a composição de palavras pode ter até 5 níveis verticais. O trabalho propõe um novo método SVM para o sistema de classificação de caracteres *Khmer*, usando para tal fim três diferentes *Kernels* SVM: o Gaussiano, o de polinomiais e o de *Kernel* Linear, bem como a formatação dos dados e o reconhecimento adotado para descobrir o melhor *kernel* para língua *Khmer*.

O sistema utilizou uma fonte "*Khmer OS Content*" e dados de treinamento com fonte de tamanho 32 pt para reconhecer três fontes de tamanhos diferentes. Sendo que a precisão da fonte de tamanho 28 pt é 98,17%, 32 pt é 98,62% e 36 pt é 98,54%, respectivamente.

Salouan *et al.* (2014) apresentam uma comparação entre dois métodos de aprendizagem-classificação: O primeiro é chamado Rede Kohonen ou Mapas *Self-Organizing* (SOM) - é caracterizado por um aprendizado não supervisionado. O segundo é chamado de Máquina de Vetor de Suporte - baseia-se em uma aprendizagem supervisionada. Tais técnicas são utilizadas para reconhecimento de manuscritos numéricos latinos, que são extraídos da base de dados MNIST. Na fase de pré-processamento, utiliza-se limiar as técnicas de centralização e de esqueletização; Na extração de características, utiliza-se o método de zoneamento. Assim, o resultado da simulação obtido demonstra que a SVM é mais robusta do que o método SOM, no reconhecimento de manuscritos numéricos latinos.

Vikram *et al.* (2013), apresentam um novo sistema para classificar caracteres Telugu - uma língua do sul da Índia - usando MLP. Inicialmente, caracteres escritos à mão foram extraídos, medindo coordenadas X, Y e Z de *pixels* dos caracteres reproduzíveis. No passo seguinte, contemplaram-se as coordenadas X, Y, Z de cada pixel dos caracteres manuscritos Telugu e o algoritmo proposto, isso possibilitou que o sistema fosse treinado para distinguir e classificar, ainda, outra matriz de caracteres pertencentes a essas classes treinadas. Sahu e Kubde (2013) apresentam um sistema de OCR para caracteres em Inglês, usando a rede neural artificial que é um método eficiente de extração de característica, esse é baseado e proposto em diagonal e pode dar alta precisão de reconhecimento, além de exigir menos tempo para treinamento e classificação, o método de extração de características que se baseia em diagonal dá precisão de até 98,8%.

Salouan *et al.* (2015) apresentam duas comparações: A primeira entre o desempenho em termo da precisão e da rapidez dos três métodos híbridos de extração de características utilizadas em OCR, que é o método de região estrutural combinado com três métodos estatísticos que são *Krawtchouk*, pseudo momentos Zernike invariantes e transformada de Fourier-Mellin, com análise invariante. A segunda comparação é realizada entre três classificadores que se compõe pela programação dinâmica, a Naive Bayes e as máquinas de vetores suporte.

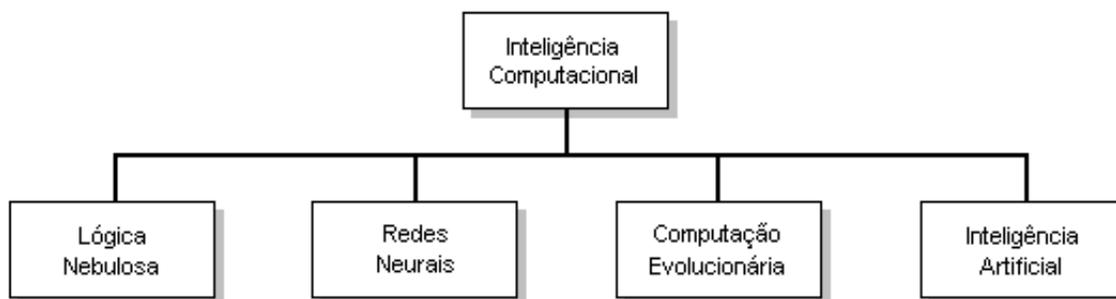
Fenwa *et al.* (2014) apresenta uma integração do método enxame de partículas otimizado (*Particle Swarm Optimization* - PSO) com KSVM, ou seja, uma melhoria do SVM para assegurar um melhor desempenho de classificação em termos de precisão de reconhecimento e de tempo de reconhecimento. Os experimentos foram conduzidos nos classificadores KSVM e PSO - *Based* KSVM, usando a ferramenta de aprendizado de máquina WEKA no conjunto de dados de dígitos manuscrito *off-line* MNIST. A melhor precisão de reconhecimento dentre os três os *kernels* foi determinada por experimentação na função polinomial, função RBF e *kernel Dot*. No entanto, os experimentos foram concebidos para permitir uma comparação no conjunto de dados, em uma validação cruzada os resultados indicaram que PSO - *Based* KSVM teve melhor desempenho do que KSVM, com uma precisão máxima de 98,57%.

## 2.2: INTELIGÊNCIA COMPUTACIONAL

De acordo com Kruse *et al.* (2013), a área de pesquisa inteligência computacional (IC) consiste em conceitos, paradigmas, algoritmos e o desenvolvimento de ferramentas que, em teoria, reconhecem e respondem aos comportamentos inteligentes, seja em ambientes extremos ou complexos. Assim, a Inteligência Computacional é identificada, fortemente, por sub-simbólicos de predominância de análogos de sistemas naturais ou, de forma mais direta, ligados à natureza.

Através da inteligência computacional é possível que sistemas inteligentes tenham características próximas ao raciocínio do ser humano, como por exemplo: Na extração de conhecimento dos dados, no aprendizado por padrões, na generalização do conhecimento aprendido, na identificação do processamento da informação incerta e imprecisa, na explicação das decisões tomadas, na identificação e na solução dos problemas complexos, na adaptação e nas melhorias do processo por otimização de parâmetros de pesquisa (JAIN *et al.* 2008).

Segundo Jain *et al.* (2008), os principais paradigmas de modelagem da IC, são: Conexionista, Lógica Difusa, Evolucionário e Híbrido. A figura 2 apresenta uma taxonomia com os principais paradigmas da Inteligência Computacional.



**Figura 2:** Os principais paradigmas da Inteligência Computacional.  
Fonte: Goldschmidt (2010)

### 2.3: APRENDIZADO DE MÁQUINA

De acordo com Alpaydin (2004), aprendizado de máquinas é uma subárea da inteligência computacional direcionada para a criação de algoritmos, técnicas e métodos com objetivos voltados para o processo de aprendizagem que permitem que as máquinas aprendam. Nesse sentido, a resolução de grandes problemas

complexos, através de programas de computador, é um desafio e muitas vezes impossíveis se consideradas as restrições técnicas e tecnológicas atuais.

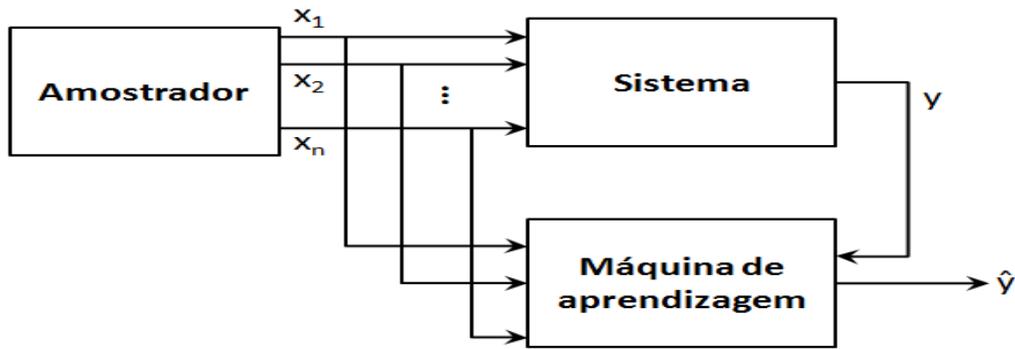
Assim, Mika (2002) descreve que o objetivo da aprendizagem de máquinas é construir dispositivos capazes de realizar tarefas complexas, começando-os através do aprendizado de um problema e não desenvolvendo sua construção.

Para Mitchell (1997), o aprendizado de máquina auxilia no estudo da construção de programas, melhorando seu desempenho de uma nova tarefa, a partir de uma primeira tarefa já realizada, ou seja, utilizando a experiência da solução inicial do problema. Nesse contexto, o processo de aprendizagem pode ser definido como por situações que o desempenho de alguma tarefa pode ser medido, "Diz-se que um programa computacional aprende a partir da experiência. E, em relação a uma classe de tarefas T, com medida de desempenho P, se seu desempenho nas tarefas T, medida por P, melhora com a experiência E." (MITCHELL, 1997, p. 02).

A partir da evolução dos computadores os aspectos técnicos ganharam força à medida que as pesquisas relevantes nessa área foram e têm sido realizadas, o que demonstra que as máquinas podem obter um nível significativo de aprendizado, mesmo que as fronteiras dessa habilidade não estejam claramente definidas. (PRATI, 2006).

Segundo Barone e Bazzan (2003), aprendizado de máquina: "é o processo pelo qual um sistema inteligente é adaptado através do estímulo do ambiente no qual está inserido", sendo que o tipo de aprendizado é determinado pela maneira que essa adaptação é realizada.

A forma como os estímulos do ambiente se relaciona com os ajustes, realizados na máquina de aprendizado, é o que define os diversos métodos de aprendizado. Basicamente, podem-se considerar os seguintes métodos de aprendizado: Supervisionado, Não-Supervisionado e Semi-Supervisionado, Indutivos e Transdutivos. Um esquema de forma de aprendizado é apresentado na figura 3.



**Figura 3:** Esquema de um sistema e uma máquina de aprendizado.  
Fonte: Silva (2008).

Nota-se, pela figura 3, que o sistema obtém as informações das observações realizadas a partir de  $x_i$ , sendo as saídas dadas por  $y_i$ , e as respostas da máquina de aprendizado buscam emular o comportamento do sistema dado por  $\hat{y}$ . O aprendizado de máquina apresenta a vantagem de apresentar boas soluções para problemas onde não se podem encontrar soluções exatas.

A referida técnica reduz a necessidade de inserção de conhecimento de um especialista para que seja possível encontrar a solução do problema, além de permitir o encontro de regras e relacionamentos implícitos nos dados, mas dificilmente observáveis por especialistas humanos. Assim, o aprendizado dos sistemas pode ser classificado basicamente como indutivo, podendo ser **não supervisionado** e **supervisionado**, e dentro dos estudos de aprendizado supervisionado, classificam-se como classificação e regressão. (RUSSELL; NORVING, 2004).

## 2.4: REDES NEURAIIS ARTIFICIAIS

Segundo Leoesch e Sari (1996), devido às suas características inerentes, as redes neurais são utilizadas em ferramentas computacionais para os recursos de reconhecimento e classificação de padrões, podendo se adaptar facilmente para responder a um padrão específico através da aprendizagem.

Osório (1991) identifica as redes neurais ou, mais precisamente, as redes neurais artificiais em simulações de um ambiente para solução de aplicação em

diversas áreas do conhecimento da ciência e da vida humana. Assim, considerando as aplicações existentes no mercado, destacam-se:

- Análise de imagens;
- Classificação de dados;
- Diagnóstico médico;
- Identificação de fraude de cartão de crédito;
- Processamento de sinais e eliminação de ruídos;
- Reconhecimento e síntese de fala;
- Reconhecimento óptico de caracteres (OCR);
- Robótica;

De acordo com Braga *et al.* (2000), as RNA são modelos matemáticos que se assemelham às estruturas neurais biológicas e que têm capacidade computacional adquirida por meio da aprendizagem e da generalização.

Para Haykin (2001), rede neural é uma máquina adaptativa com um processador maciço e paralelamente distribuído, constituído de unidades de processamento simples e que têm a propensão natural para armazenar conhecimento experimental e, posteriormente, torná-lo disponível para uso.

Segundo (HAYKIN, 2001; SILVA, 2009), a arquitetura da rede é definida pela forma na qual esses neurônios estão organizados e interconectados, ou seja, pelo número de camadas, número de neurônios por camada, tipos de conexão entre os neurônios e a topologia da rede. Existem diversos modelos para implementação de uma estrutura de rede neural artificial, como a SOM (*Self organizing map*), RBF (*Radius Basis Function*), LMS (*Least Mean Square*) e a MPL (*Multi Layer Perceptrons*).

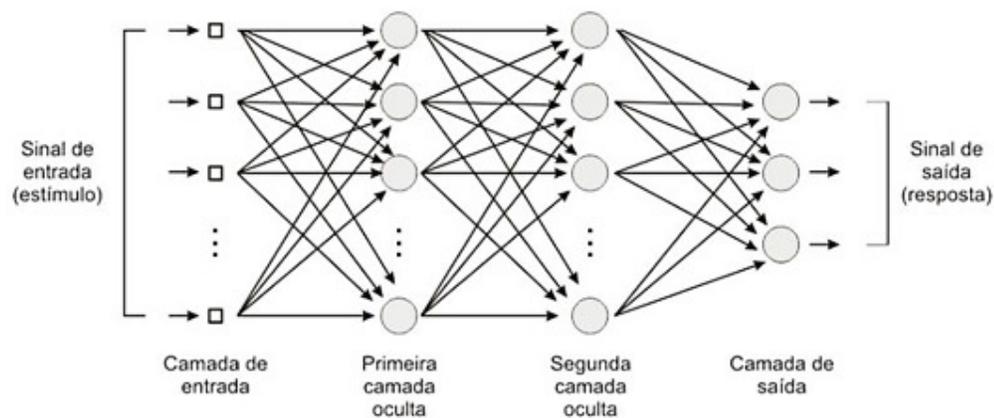
#### **2.4.1: Perceptrons Multicamadas (MLP)**

As redes de múltiplas camadas se distinguem das redes de camada simples pelo número de camadas intermediárias que são aquelas situadas entre a camada de entrada e a de saída. Essa arquitetura possui uma ou mais camadas ocultas, que são compostas por neurônios computacionais, também chamados de neurônios

ocultos. Nesse contexto, Haykin (2001) define que a função dos neurônios ocultos é interagir entre as camadas de entrada externa e a saída da rede.

Quando se adiciona uma ou mais camadas ocultas, a rede será capaz de extrair estatísticas de ordem elevada. Nesse sentido, e considerando as habilidades dos neurônios ocultos de extraírem estatísticas de ordem elevada, é extremamente importante para a função que o tamanho da camada de entrada (estímulo) seja relativamente grande.

Na Figura 4 é apresentado um grafo arquitetural de um *perceptron* multicamadas com duas camadas ocultas e uma camada de saída.



**Figura 4:** Rede Feedforward de Múltiplas Camadas.  
Fonte: Haykin (2001).

Uma rede típica *Feed for ward* tem neurônios dispostos em uma topologia de camadas distintas. A camada de entrada não é realmente neural a todas: essas unidades servem simplesmente para introduzir os valores das variáveis de entrada. Os neurônios das camadas ocultas e de saída são ligados cada um a todas as unidades da camada anterior.

#### 2.4.1.1: Algoritmo de treinamento

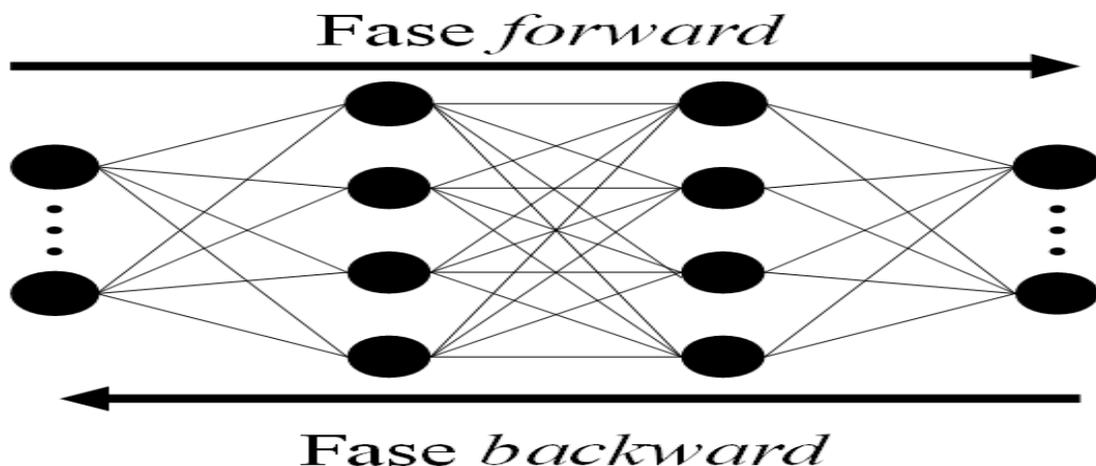
O algoritmo de treinamento mais utilizado em modelos MLP é o *Backpropagation*, esse se baseia na aprendizagem por correção de erros. O algoritmo de *Backpropagation* é um tipo de aprendizado supervisionado, dessa forma, quando o

valor de saída é gerado, o erro é calculado e seus valores são retro propagados para a entrada, os pesos são ajustados e os valores são novamente calculados.

O funcionamento algoritmo de *Backpropagation*, segundo Ferrari *et al.* (2006), é o seguinte:

- Apresenta-se um padrão à camada de entrada da rede;
- O padrão é processado, camada por camada, até que a camada de saída forneça a resposta processada;
- A resposta é comparada com a resposta desejada, se estiver errada, o erro é calculado.
- Os valores são retro propagados da camada de saída para a camada de entrada, os pesos são ajustados e o processamento é feito novamente, até que se obtenha a resposta desejada.

A figura 5 ilustra o processo de treinamento do algoritmo *back propagation* feito em duas fases, a fase **forward** e a fase **backward**.



**Figura 5:** Processo de treinamento do algoritmo back propagation.  
Fonte: Elaborado pelo Autor (2015).

## 2.5: MÁQUINAS DE VETORES SUPORTE

De acordo com Gunn (1998), máquinas de vetores suportes (*Support Vector Machines* – SVM's), possuem a capacidade de resolver problemas de regressão e classificação, considerando o aprendizado realizado na etapa de treinamento e da sua capacidade de generalização. Assim, em um problema binário, através da SVM, o objetivo é separar as instâncias em duas classes, isso através de uma função que será obtida a partir dos exemplos conhecidos na fase de treinamento e aprendizagem.

O funcionamento básico do SVM é dado por duas classes e um conjunto de instâncias de treinamento, cujas amostras concernem com essas classes e o SVM constrói um hiperplano que segmenta o espaço de características em duas regiões, maximizando a fronteira de separação entre as mesmas. Esse hiperplano é conhecido como hiperplano de separação ótima. As amostras estranhas são mapeadas para esse mesmo espaço e atribuídas a uma das classes. (ALPAYDIN, 2010).

Dentro do contexto Alpaydin (2010) destaca que o SVM funciona da seguinte forma: Considerando duas classes e um conjunto de instâncias de treinamento, cujas amostras pertencem a essas classes, a SVM cria um hiperplano que divide o espaço de características em duas regiões, maximizando a margem de separação. Esse hiperplano é conhecido como hiperplano de separação ótima e as amostras desconhecidas são mapeadas para esse mesmo espaço e, na sequência, atribuídas à uma das classes.

SVM são hiperplanos que separam os dados de treinamento por uma margem máxima e, esse recebe como entrada um conjunto de instâncias previamente rotuladas para a sua fase de treinamento, como qualquer classificador supervisionado. As instâncias de treinamento que se encontram mais próximas do hiperplano são chamadas de vetores de suporte (BURGES, 1998). Um SVM linear constrói um hiperplano que separa classes positivas (+1) e negativas (-1) com a margem máxima.

De acordo com Silva e Scarpel (2007), na classificação binária, o problema é encontrar uma função paramétrica, linear ou não, que defina um hiperplano de separação das classes. Sendo  $T$  um conjunto de entrada do tipo  $(X_i, Y_i)$ , onde  $X_i$  representa a “i-ésima” tupla de entrada e  $Y_i$  sua respectiva classe, sendo  $Y = \{-1, +1\}$ ,  $T$  é linearmente separável se houver possibilidade de separação dos dados

das classes -1 e +1 por um hiperplano (LORENA; CARVALHO, 2007). A superfície de decisão, na forma de um hiperplano que realiza essa separação, é obtida pela equação 1.

$$\mathbf{w}^T \cdot \mathbf{x} + b = 0$$

**Equação 1:** Equação de Superfície de Decisão  
Fonte: Elaborado pelo Autor (2015).

Sendo  $x$  um vetor de entrada,  $w$  um vetor de peso ajustável e  $b$  um bias. Assim, a equação 1 é reescrita com base nas classes  $Y = +1$  e  $Y = -1$ , como mostra a equação 2 e a equação 3.

$$\mathbf{w}^T \cdot \mathbf{x} + b \geq 0 \text{ para } Y = +1$$

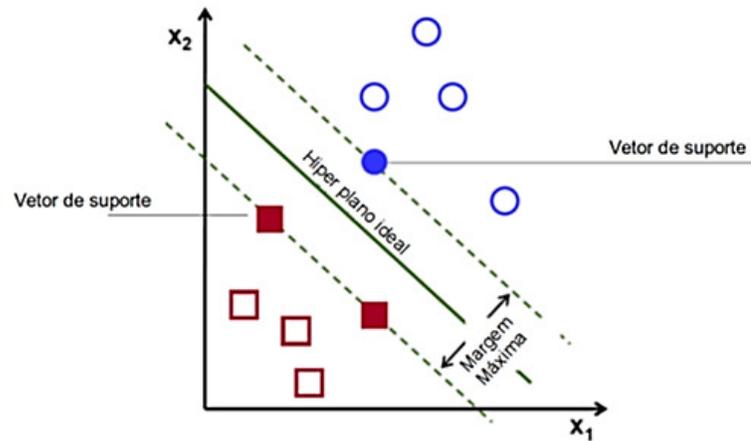
**Equação 2:** Equação com base nas classes  $Y = +1$  e  $Y = -1$  (1)  
Fonte: Elaborado pelo Autor (2015).

$$\mathbf{w}^T \cdot \mathbf{x} + b < 0 \text{ para } Y = -1$$

**Equação 3:** Equação com base nas classes  $Y = +1$  e  $Y = -1$  (2)  
Fonte: Elaborado pelo Autor (2015).

De acordo com Haykin (2001), dado um vetor  $w$  e um bias  $b$ , a separação entre o hiperplano definido na equação 1 e o ponto de dado mais próximo é denominado margem de separação, representada por  $\rho$ . Sempre que for possível obter um  $\rho > 0$ , existirão infinitos hiperplanos entre as duas classes.

Sendo duas classes linearmente separáveis, deseja-se encontrar um hiperplano de separação que produz o menor erro de generalização entre o número infinito de hiperplanos. O hiperplano ideal é aquele que possui a margem máxima de separação entre duas classes, onde a margem corresponde à soma das distâncias do hiperplano para os pontos/dados mais próximos de cada uma das duas classes. Esses pontos próximos são chamados de vetores de suporte. A figura 6 ilustra um hiperplano ideal para um exemplo linearmente separável.



**Figura 6:** Exemplo de hiperplano ótimo linearmente separáveis.  
Fonte: Cristianini; Shawe-Taylor, (2000).

Nesse sentido, considerando os termos conceituais, Haykin (2001) define que, os vetores de suporte são os pontos de dados que se encontram mais próximos da superfície de decisão, portanto, são os mais difíceis de classificar.

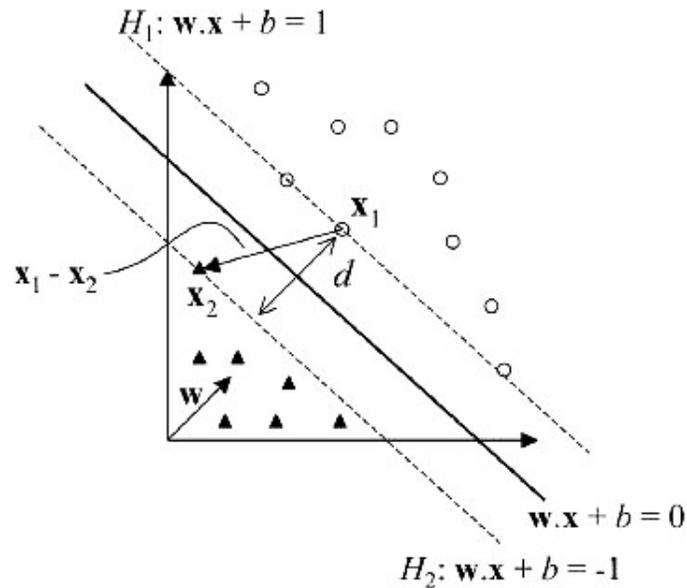
Ainda de acordo com Haykin (2001), considerando que  $w_o$  e  $b_o$  revelem os valores ótimos do vetor peso e bias, o hiperplano ótimo representa uma superfície de decisão linear multidimensional no espaço de entrada, e está representado pela equação 4.

$$w_o^T \cdot x + b_o = 0$$

**Equação 4:** Equação de representação uma Superfície de Decisão Linear Multidimensional no Espaço de Entrada.

Fonte: Elaborado pelo Autor (2015).

Sendo  $x_1$  um ponto no hiperplano  $H_1: w^T \cdot x + b = +1$  e  $x_2$  um ponto no hiperplano  $H_2: w^T \cdot x + b = -1$ , como ilustrado na figura 7.



**Figura 7:** Distância  $d$  entre os hiperplanos  $H_1$  e  $H_2$ .  
Fonte: Lorena; Carvalho (2007).

De acordo com Lorena e Carvalho (2007), projetando  $x_1 - x_2$  na direção de  $w$ , perpendicular ao hiperplano separador  $w^T \cdot x + b = 0$ , pode-se obter a distância entre os hiperplanos  $H_1$  e  $H_2$ , que é a projeção ilustrada na equação 5:

$$d = (x_1 - x_2) \left( \frac{w}{\|w\|} \cdot \frac{(x_1 - x_2)}{\|x_1 - x_2\|} \right)$$

**Equação 5:** Equação de Distância entre os Hiperplanos  $H_1$  e  $H_2$  (1)  
Fonte: Elaborado pelo Autor (2015).

Tendo  $w^T \cdot x_1 + b = +1$  e  $w^T \cdot x_2 + b = -1$ . A diferença entre essas equações fornece  $w^T \cdot (x_1 - x_2) = 2$ . Substituindo esse resultado na equação 5, obtêm-se a equação 6:

$$d = \frac{w}{\|w\|} \cdot \frac{(x_1 - x_2)}{\|x_1 - x_2\|}$$

**Equação 6:** Equação de Distância entre os Hiperplanos  $H_1$  e  $H_2$  (2)  
Fonte: Elaborado pelo Autor (2015).

Como se deseja obter o comprimento do vetor projetado, toma-se a norma da equação 6, obtendo a equação 7:

$$d = \frac{2}{\|w\|}$$

**Equação 7:** Equação para Obter o Comprimento do Vetor Projetado  
Fonte: Elaborado pelo Autor (2015).

Essa é a distância  $d$ , ilustrada na figura 6, entre os hiperplanos  $H_1$  e  $H_2$ , paralelos ao hiperplano separador, onde  $w$  e  $b$  foram escalados de forma a não haver exemplos entre  $H_1$  e  $H_2$ ,  $1/\|w\|$  é a distância mínima entre o hiperplano separador e os dados de treinamento. A maximização da margem de separação dos dados em relação ao hiperplano ótimo pode ser encontrada pela minimização de  $\|w\|$ .

## 2.6: ALGORITMO DOS K VIZINHOS MAIS PRÓXIMOS (KNN)

Aha *et al.* (1991), descreve que o algoritmo de classificação dos  $k$  vizinhos mais próximos ou, simplesmente kNN, insere-se no âmbito da aprendizagem baseada em casos (*Instance Based Learning*). Esse algoritmo possui objetivo e princípio em que casos semelhantes deverão ter a mesma classe. Esse resultado de continuidade é - de certa forma - comum a todos os algoritmos, no entanto, utiliza o conceito de semelhança de forma direta para identificar a classe. A semelhança é comparada através de uma função de distância que é aplicada aos pares de casos.

De acordo Theodoridis e Koutroumbas (2006), KNN é uma técnica de classificação, baseada em instâncias, que permite atribuir a classe de cada elemento desconhecido, partindo da classe majoritária obtida entre os seus vizinhos mais próximos, esses são identificados no conjunto de treinamento. Para encontrar seus vizinhos é criada uma medida de similaridade que normalmente é uma medida de distância calculada no espaço de características de cada vizinho.

Uma das medidas de similaridade mais utilizadas na literatura é distância Euclidiana. Tal medida é calculada pela equação a seguir:

$$d(x, y) = \sqrt{\sum_i^n (x_i - y_i)^2}$$

**Equação 8:** Equação da Distância Euclidiana  
Fonte: Elaborado pelo Autor (2015).

Conforme a equação descrita em (8),  $x$  é um exemplo de treino, representado por  $x = [x_1, \dots, x_n] \in \mathfrak{X}^n$ , onde  $n$  é dimensão do vetor de características e  $y$  representa a amostra de teste, sendo  $y = [y_1, \dots, y_n] \in \mathfrak{Y}^n$ .

De acordo com Theodoridis e Koutroumbas (2006), o KNN é um dos algoritmos mais simples dentre as técnicas existentes de aprendizagem de máquina. Seu funcionamento será descrito a seguir:

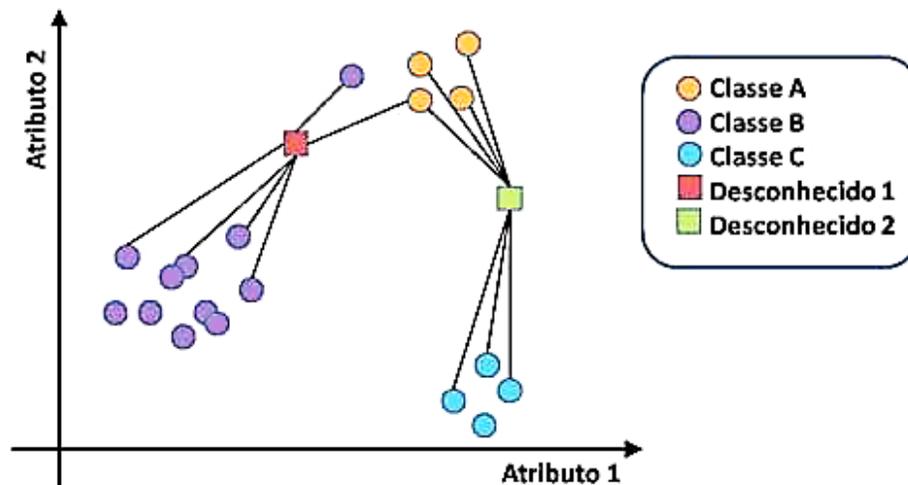
1. Identifique o valor de  $k$ , ou seja, o número de vizinhos mais próximos;
2. Calcule a distância da nova amostra a ser classificada entre todas as amostras de treinamento;
3. Identifique os  $k$  vizinhos mais próximos, independentemente do rótulo das classes;
4. Conte o número de vizinhos mais próximos que pertencem a cada classe do problema;
5. Classifique a nova amostra, atribuindo-lhe a classe mais constante na vizinhança.

De acordo com Mitchell (1997), o KNN é um método de classificação de padrão que não possui processamento na fase de treinamento, para cada padrão de teste é medida a distância entre ele e todos os padrões de treinamento. Verificam-se quais classes pertencem os  $k$  padrões mais contíguos e a classificação é feita associando o padrão de teste à classe que for predominante.

Nesse contexto, considerando a classificação de um padrão onde o parâmetro  $k$  é igual a 1 (NN), e se o  $k$  for maior que 1, por exemplo,  $k = 3$ , na qual são considerados três vizinhos do novo padrão, são analisadas as três menores distâncias do novo padrão para os padrões de treinamento. A classe que obtiver o maior número de padrões dentre essas distâncias será a classe determinante do novo padrão. A figura 8 representa esse processo de classificação. Para classificar

um ponto, primeiro se tomam os k-vizinhos mais próximos dele e, dentro desse conjunto, encontra-se a classe mais significativa.

O ponto **desconhecido 1** será classificado como classe B e o ponto **desconhecido 2** será classificado como classe A.



**Figura 1** - Exemplo de classificação pelo método KNN.  
Fonte: Martins (et al., 2005).

## 2.7: COMPUTAÇÃO GRÁFICA

Segundo Marques Filho e Vieira Neto (1999), dentro da ciência da computação, a computação gráfica se destaca como área responsável pela geração e manipulação de imagens em geral, através de sistemas computadorizados. Geralmente, está relacionada aos jogos digitais, além de possuir um conjunto de técnicas aplicáveis a qualquer ramo da informática, tal como a geração de interfaces gráficas com usuário, artes digitais, softwares de uso geral e em muitos trabalhos, como neste, essa técnica será aplicada na preparação de imagens correspondentes às derivações de sinais impressos de caracteres, permitindo que as mesmas sejam analisadas por um sistema de inteligência artificial.

### 2.7.1: Imagem digital

Segundo Maillard (2001), imagem digital é a discretização de uma cena ou de uma imagem em formato analógico (foto, sinal de televisão ou de vídeo) e é representada através de uma matriz bidimensional ou com mais dimensões para imagens coloridas ou multiespectrais de valores numéricos.

De acordo com Pedrini (2015), imagem digital pode ser representada por uma matriz bidimensional, onde cada elemento da matriz corresponde a um *pixel* da imagem. A figura 9 ilustra a representação matricial de uma imagem.



**(a) Representação matricial**

120	138	120	151	139
110	129	129	139	146
150	138	137	138	129
137	129	129	128	137
146	145	131	132	145

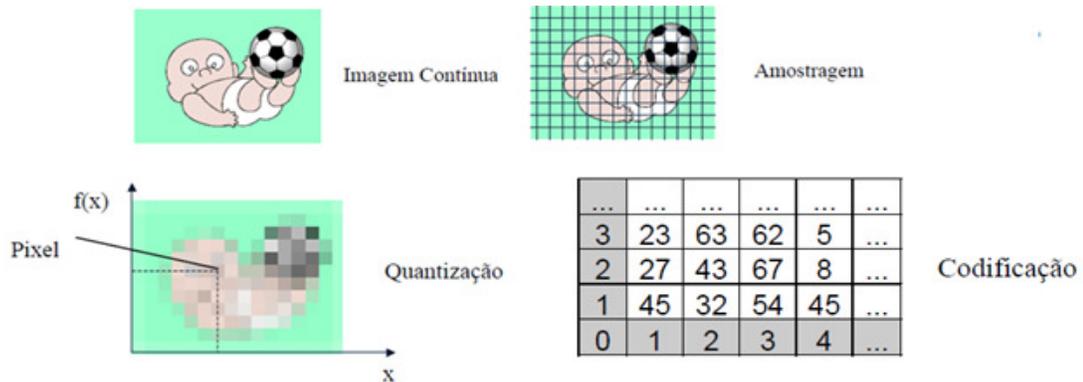
**(b) Região da imagem**

**Figura 9:** Representação de Imagens Digitais.  
Fonte: Pedrini (2015).

Uma pequena região da imagem é destacada, sendo formada por números inteiros correspondendo aos níveis de cinza dos *pixels* da imagem.

De acordo com Gonzalez e Woods (1992), para possibilitar a representação de uma função contínua em ambientes computacionais, é preciso uma função discreta equivalente, ou seja, uma função cujo domínio e contradomínio sejam números naturais, o que significa um conjunto finito de valores, ao contrário das funções contínuas que podem possuir infinitos valores em seus domínios e contradomínios.

A figura 10 ilustra o processo de discretização de imagem e mostra uma possível codificação da cor para cada elemento da matriz.



**Figura 10:** Processo de Discretização de uma Imagem.  
Fonte: Adaptada Scuri (2015).

Imagens digitais podem ser representadas pela atribuição discreta de uma característica de tons/cor, brilho e transparência para cada ponto da função discreta, representada ou através da atribuição de um nível de cinza, por exemplo, para imagens monocromáticas, como representado na equação que segue.

**Equação 9** - Conversão de tons de cinza, onde  $x$  e  $y$  são números reais no intervalo 0 a  $X$  e 0 a  $Y$ :

$$f(x,y) = \text{cor no ponto } (x,y); \text{ para } x \in [0,X] \text{ e } y \in [0,Y]$$

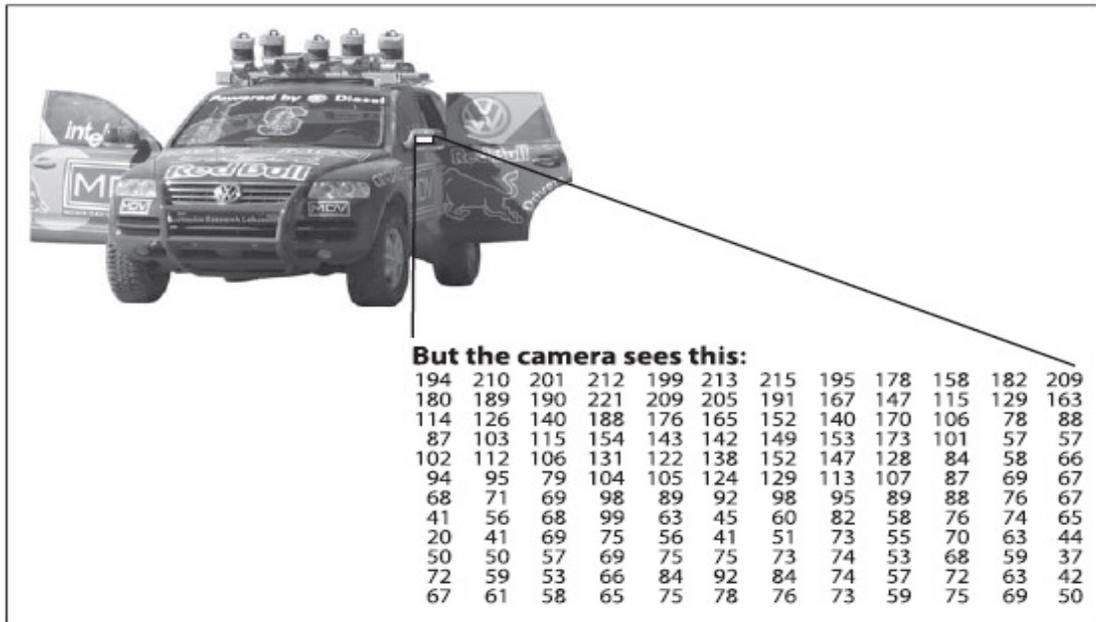
**Equação 9:** Equação da Conversão de Tons de Cinza  
Fonte: Elaborado pelo Autor (2015).

Assim sendo, uma imagem digital monocromática será o resultado da atribuição dos níveis de cinza para cada uma das coordenadas  $(x, y)$  de uma função discreta. A intensidade da imagem de saída é definida como nível de cinza ( $L$ ), para cada um dos pontos  $(x, y)$ , estando a intensidade dentro dos limites de intervalo, como descrito na equação representada a seguir.

$$L_{min} \leq L \leq L_{max}, \text{ sendo } L_{min} \text{ e } L_{max} \text{ valores finitos e positivos.}$$

**Equação 10:** Equação da Definição dos Tons de Cinza.  
Fonte: Elaborado pelo Autor (2015).

A figura 11 ilustra a representação de uma imagem digital em tons de cinza.



**Figura 11:** Imagem digital em tons de cinza.  
Fonte: Bradski; Kaebler (2008).

Criar imagens digitais, partindo de uma imagem real, é um processo composto por duas etapas que juntas permitirão a representação digital da informação que foi capturada pelo dispositivo, as referidas etapas são a aquisição e a digitalização. (GONZALEZ; WOODS, 1992). A **aquisição** é o primeiro momento da criação de uma imagem digital, a partir de uma imagem real e a **digitalização** é o resultado do processo anterior e que deve ser transformado para um formato manipulável, para o sistema computacional.

Dadas as características de algumas imagens, em alguns casos é necessário um processo de limiarização, ou seja, é preciso realizar a detecção e a separação das informações contidas em uma determinada imagem digital, permitindo a posterior criação de uma imagem binária que seja constituída de apenas 0's e 1's.

Contudo, para que isso se torne possível, é necessário definir o nível de cinza **N**, de forma que qualquer valor inferior ou igual ao mencionado seja considerado como o objeto e todo valor superior seja considerado como o o fundo, possibilitando assim que no final do processo se tenha somente dois tipos de informação. Em notações matemáticas a limiarização é uma operação onde uma entrada  $f(x,y)$  de

nível cinza  $N$  gera uma saída  $g(x,y)$  denominada imagem limiarizada, representado na equação a seguir. (GONZALEZ; WOODS, 1992).

$$\begin{aligned} g(x,y) &= 1 \text{ se } f(x,y) \geq T \\ g(x,y) &= 0 \text{ se } f(x,y) < T \end{aligned}$$

**Equação 11:** Equação de Limiarização.  
Fonte: Elaborado pelo Autor (2015).

A figura 12 ilustra um exemplo de operação de limiarização que é uma técnica muito usada para reconhecimento de caracteres.



**Figura 12:** Exemplo de limiarização com limiar = "128".  
Fonte: Elaborado pelo Autor (2015).

Podem-se definir os *pixels* de valor igual a 1 como sendo objeto e os *pixels* com valor igual a 0 como sendo o fundo em que  $T$  é um nível de cinza predefinido. Quando se trabalha com imagens que possuem somente duas informações essa é técnica bastante útil, entretanto, se a imagem possuir uma série de informações, a melhor representação da mesma não seria uma imagem binária (GONZALEZ; WOODS, 1992). Assim sendo, dadas as características das imagens analisadas neste trabalho, Afinamento, *Thinning* ou Esqueletização foram aplicadas com o propósito de preparar as imagens para utilização na ferramenta. Esses conceitos serão apresentados de forma sucinta para que não haja fuga do escopo deste trabalho.

### **2.7.2: Processamento Digital de Imagens**

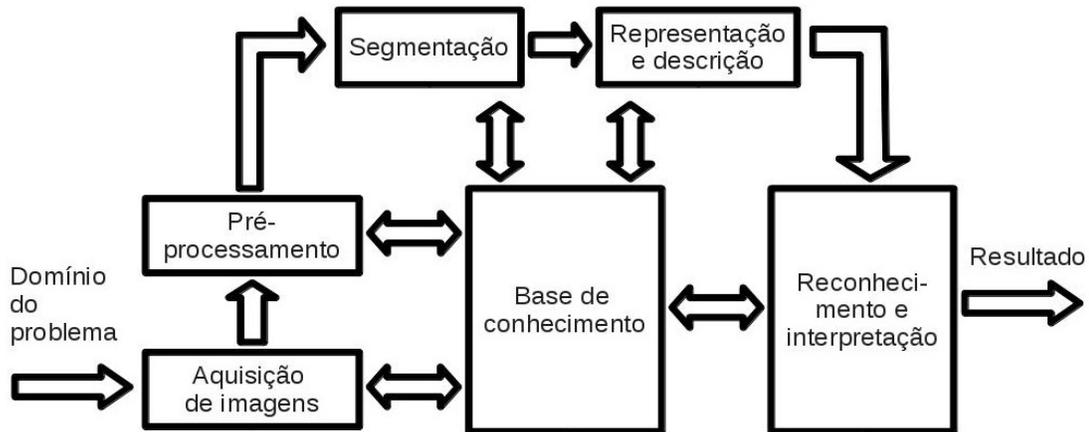
Processamento digital de imagens é a manipulação de uma imagem por computador de modo que a entrada e a saída do processo sejam também imagens. O propósito de se usar processamento digital de imagens é melhorar o aspecto visual de certas feições estruturais para o analista humano e para propiciar outros subsídios de sua apreciação, gerando resultados que - posteriormente - possam ser submetidos a outros processamentos. (JENSEN, 2000).

Ainda segundo Jensen (2000), há outras funções que se destacam ao utilizar o processamento digital de imagens que são a detecção, a identificação e a mensuração de objetos de uma cena, isso através de aplicações de multiconceitos - multiescala, multitemporal e multiespectral - que envolvem princípios estatísticos para reconhecimento de padrões, modelagem, sistemas de redes neurais, fotogrametria e interpretação de imagens.

Segundo Pedrini e Schwartz (2008) há várias técnicas para processamento de imagens e dentre essas é possível identificar as técnicas de realce, segmentação, compressão, análise de textura, morfologia matemática e outras. Juntas, todas essas técnicas podem ser utilizadas para uma vasta gama de aplicações, tais como: processamento de imagem de satélite, processamento de imagem médicas, detecção de face e de objetos, visão computacional para robôs em automação industrial, dentre muitas outras.

De acordo com Gonzalez e Woods (1992), geralmente, o primeiro passo para processar uma imagem é a segmentação, pois essa consiste em dividir uma imagem entre seus constituintes ou objetos. O nível até o qual essa divisão é levada depende do problema a ser resolvido.

A figura 13 ilustra os passos fundamentais para o processamento digital de imagens com enfoque no reconhecimento de imagens.



**Figura 13:** Passos para o processamento digital de imagens.  
Fonte: Gonzalez; Woods (2000).

A descrição da sequência padrão para o reconhecimento de imagens é a seguinte:

- **Domínio do problema:** Profunda análise do problema a ser tratado para se especificar o objetivo a ser alcançado;
- **Aquisição de imagens:** Consiste em utilizar um equipamento para adquirir a imagem, como uma câmera de vídeo, uma câmera fotográfica digital ou um digitalizador de imagens. Esse processo pode ser também chamado de digitalização que consiste na conversão de um sinal da forma analógica para a forma digital;
- **Pré-processamento:** Pode ser entendido como o processamento inicial da imagem para correção de distorções geométricas e a remoção de ruídos, modificando a imagem para evidenciar características a serem utilizadas nas próximas etapas;
- **Segmentação:** Consiste na etapa de processamento em que se analisa a imagem em relação à informação nela presente. A imagem é dividida em diferentes regiões que são, posteriormente, analisadas por algoritmos em busca de informações que a caracterizem. A segmentação consiste em extrair da imagem apenas as áreas que interessam para a resolução do problema;

- **Base de conhecimento:** Representa o conhecimento adquirido pelo ser humano, o qual é direcionado ao processamento da imagem para o melhor reconhecimento possível;
  
- **Representação e descrição:** No processo de representação as características principais devem ser enfatizadas e extraídas. O processo de descrição é também conhecido como seleção de características, pois nesse são extraídas as características principais que resultam em informação quantitativa, capaz de separar classes de objetos importantes para o reconhecimento dos padrões;
  
- **Reconhecimento e interpretação:** Consiste em atribuir rótulos aos objetos, classificando-os a partir das informações encontradas na imagem na etapa anterior. Essa etapa tenta aproximar o desempenho computacional ao do ser humano quando associado ao reconhecimento de padrões dentro de uma imagem.

### 3: METODOLOGIA

O trabalho foi dividido em vários experimentos com o objetivo de promover uma maior investigação na diferenciação dos caracteres presentes em imagens do banco de imagens de caracteres. Esse processo pode ser desenvolvido basicamente em três etapas. A primeira etapa se constitui como processamento digital da imagem que envolve tanto a melhoria na sua qualidade quanto a detecção do objeto de interesse. A segunda etapa diz respeito à extração de características que possam descrever o objeto e a seleção das mais significativas. Por fim, a terceira etapa é a classificação em que, após escolhidas as características - na etapa anterior - essas servirão de entrada para os métodos classificadores e esses fornecerão a tendência dos caracteres para os classificar.

Para alcançar os objetivos apresentados, realizou-se uma extensa revisão bibliográfica em busca de trabalhos e materiais sobre o uso da inteligência computacional no reconhecimento óptico de caracteres. A codificação dos algoritmos dos métodos de reconhecimento óptico de caracteres foi realizada na linguagem de programação *Python* com a biblioteca *OpenCV*. Tal escolha se deve à possibilidade de utilização dos diversos recursos oferecidos pela biblioteca *OpenCV*, elaborados para a área de processamento de imagens e visão computacional, de modo a simplificar o processo de codificação e garantir a estabilidade para a realização dos testes.

I: O ambiente utilizado para o desenvolvimento foi:

#### 1.1: **Softwares:**

- ✓ Sistema Operacional *Windows* 8.1;
- ✓ Plataforma de Desenvolvimento, IDLE (*Python* GUI);
- ✓ Linguagem de Programação *Python*;

### 1.2: Hardware:

- ✓ *Notebook*;
- ✓ Processador *Intel Core i7* – Terceira Geração;
- ✓ 6 GB Memória RAM;

## 3.1: DETECÇÃO DE SÍMBOLO NA IMAGEM

O processo de digitalização, segundo Osório (1991), consiste em realizar a aquisição de uma imagem e enviá-la ao computador em um formato adequado para que esse possa manipulá-la. As informações visuais são convertidas em sinais elétricos, que são convertidos em valores binários a serem armazenados na memória do computador. Nesse processo, os sinais são amostrados espacialmente e quantificados em amplitude, de forma a obter a imagem digital.

No processamento de imagens, a imagem digital é definida como uma matriz formada por  $m \times n$  elementos. Cada elemento dessa imagem é denominado *pixel* (*picture element*) e possui associação com informações referentes à cor e à luminosidade.

O processo de análise de imagem para detectar símbolos de caracteres, pelo exame dos *pixels*, é a parte central do conjunto de preparação - tanto na entrada do treinamento como na fase de testes. Extensões simbólicas são reconhecidas a partir de um arquivo de imagem de entrada com base no valor de cor de *pixels* individuais, o que para o limite desse projeto é assumido para ser preto RGB (0, 0,0) ou branco RGB (255, 255, 255). As imagens de entrada são assumidas em forma de *bitmap*, de qualquer resolução, que pode ser mapeado para um objeto *bitmap* interno no ambiente IDLE (*Python* GUI). O procedimento também assume que a imagem de entrada é composta apenas de caracteres e qualquer outro tipo de objeto delimitador, como uma linha de fronteira, não é considerado.

O procedimento para a análise de imagens para detecção de caracteres está listado nos seguintes algoritmos:

### 3.1.1: Determinar linhas de Caractere:

- Enumeração das linhas de caracteres em uma imagem de caractere, o que é essencial para a delimitação dos limites em que a detecção pode prosseguir.
- Assim, detectar o próximo caractere em uma imagem não envolve necessariamente a digitalização da imagem inteira novamente.

#### 3.1.1.1: Algoritmo (1):

1. Iniciar primeiro no  $x$  e no primeiro *pixel*  $y$  da imagem *pixel* (0,0) que ajusta o número das linhas para 0.
2. Verificar se a largura da imagem é a mesma no componente  $y$  da imagem, classificando-o como:
  - a. Se um *pixel* preto for detectado, registra-se  $y$  como topo da primeira linha;
  - b. Senão for para o próximo *pixel*;
  - c. Se nenhum *pixel* preto que for encontrado, aumenta-se a largura de  $y$  e se reinici  $x$  para verificar a próxima linha horizontal.
3. Inicia-se, no topo da linha, o primeiro componente  $x$  encontrando *pixel* (0, na linha\_superior).
4. Verificar se a largura da imagem é a mesma no componente  $y$  da imagem, obtendo:
  - a. Se um *pixel* preto for detectado, registra  $y-1$  como topo da primeira linha. Incrementa-se o número das linhas;
  - b. Senão for para o próximo *pixel*;

c. Se nenhum *pixel* preto for encontrado, incrementa-se largura de y e reinicia x para verificar a próxima linha horizontal.

5. Inicia-se abaixo da parte inferior da última linha encontrada e se repetem os passos 1 a 4 para detectar as linhas subsequentes.

6. Se a parte inferior da imagem (altura da imagem) for atingido pare.

### 3.1.2: Detecção de símbolos individuais:

- A detecção de símbolos individuais envolve a digitalização das linhas de caracteres, ortogonalmente, para separar *pixels* pretos compostos na imagem.

#### 3.1.2.1: Algoritmo:

1. Inicia-se no topo a primeira linha do componente x do caractere e a primeira linha do componente x.

2. Digitaliza-se a largura da imagem sobre o mesmo componente y.

a. Se o *pixel* preto é detectado, registra-se y como topo da primeira linha;

b. Senão for para o próximo *pixel*;

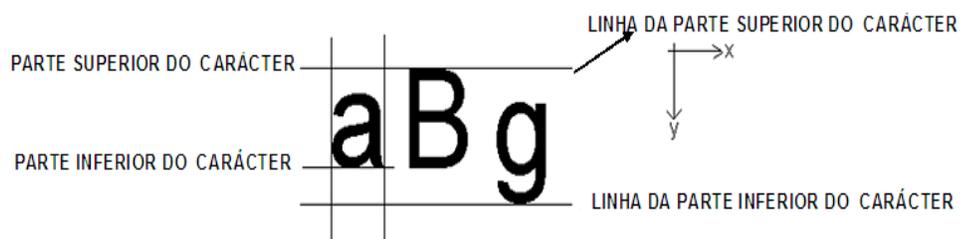
c. Inicia-se no topo do caractere encontrado e primeiro componente x *pixel* (0, superior\_caracter).

4. Digitaliza-se a linha até a parte inferior no mesmo componente x:

a. Se um *pixel* preto for encontrado, registra-se x como a esquerda do símbolo;

b. Senão for para o próximo *pixel*;

- c. Se nenhum *pixel* preto for encontrado, incrementa-se x e reinicia y para verificar a próxima linha vertical.
5. Inicia-se à esquerda do símbolo encontrado e na parte superior da linha corrente, pixel (esquerda \_caractere, topo\_linha).
6. Verificar se a largura da imagem é a mesma no componente x:
- a. Se nenhum caractere preto for encontrado, registra-se x-1 como direita do símbolo;
- b. Se um *pixel* preto for encontrado, incrementa-se x e reinicia y para digitalizar a próxima linha vertical;
7. Inicia-se na parte inferior da linha corrente e à esquerda do símbolo, *pixel* (esquerda caractere, linha\_inferior).
8. Digitaliza-se a primeira linha até a direita do caractere no mesmo componente y:
- a. Se um *pixel* preto for encontrado, registra-se como parte inferior do caractere;
- b. Se nenhum *pixel* preto for encontrado, decrementa-se y e se reinicia x para verificar a próxima linha vertical.

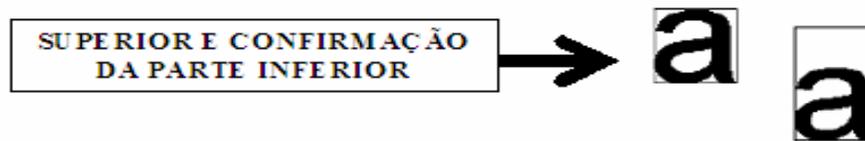


**Figura 14:** Linha e detecção dos limites dos caracteres.  
Fonte: Elaborado pelo Autor (2015)..

A partir do processo seguido e da figura acima, é evidente que o carácter detectado vinculado pode não ser o real destino ao caractere em questão. Essa é

uma questão que surge em decorrência da altura e da irregularidade existentes no alinhamento, na parte inferior com símbolos alfabéticos impressos. Assim, uma linha superior não significa, necessariamente, ser o topo de todos os caracteres e uma linha inferior pode não significar inferior de todos os caracteres também. Assim, uma confirmação da parte superior e inferior para o caractere é necessária e um algoritmo de confirmação opcional é implementada no projeto:

- a. Inicia-se no topo da linha atual e à esquerda do carácter.
- b. Digitalizar até a direita do caractere:
  1. Se um *pixel* preto for detectado, registra-se y como topo confirmado;
  2. Senão for para o próximo *pixel*;
  3. Se nenhum dos *pixels* pretos são encontrados, incrementa-se y e se reinicia x para verificar a próxima linha horizontal.



**Figura 15:** Confirmação dos limites dos caracteres.  
Fonte: Elaborado pelo Autor (2015).

### 3.2: ATIVIDADES DESENVOLVIDAS PARA O CASO ESPECÍFICO DESTE PROJETO

As atividades para o caso específico deste projeto foram desenvolvidas seguindo os passos abaixo:

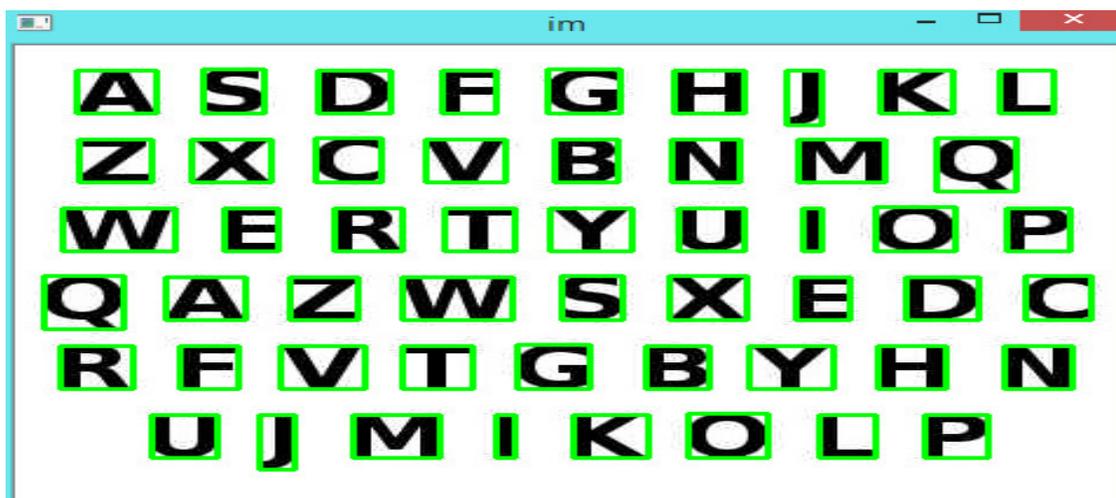
1. Estudo de estruturas de geração das imagens, redes neurais artificiais, SVM, KNN e aplicações de reconhecimento de padrões, programação em *Python*;

2. *Download*, instalação e estudo da biblioteca *OpenCV*;
3. Elaboração de exemplos e testes com a biblioteca *OpenCV*. Nessa etapa, foram desenvolvidos algoritmos em *Python* utilizando a biblioteca *OpenCV*, com propósitos voltados para o projeto, tais como: rotinas algoritmos de processamento de imagem, reconhecimento de padrões;
4. Análise e definição dos procedimentos utilizados para estruturação dos algoritmos. Nessa etapa foi feito um planejamento de como os algoritmos seriam implementados;
5. Desenvolvimento em *Python* utilizando a biblioteca *OpenCV* dos algoritmos para ler imagem e aplicar operações de processamento de imagem, efetuar reconhecimento de um padrão em uma sequência de imagens utilizando a rede neural, SVM e KNN para determinado padrão em uma imagem contendo caracteres;
6. Realização de testes e validação; para ler um arquivo de imagens, capturar a imagem, processar o conteúdo de uma imagem, entre outras.

Na etapa posterior, foi realizado o estudo e execução de exemplos de programas utilizando a biblioteca *OpenCV* que acompanham a ferramenta e estão disponíveis para execução após sua devida instalação e configuração. Além disso, foram desenvolvidos algoritmos em *Python*, utilizando a biblioteca *OpenCV* tais como: rotinas para ler um arquivo de imagens, processar o conteúdo da imagem, treinar e localizar um padrão em uma imagem, dentre outras. Essas rotinas foram usadas, posteriormente, para implementação dos algoritmos de processamento de imagem, reconhecimento de padrões e reconhecimento óptico de caractere.

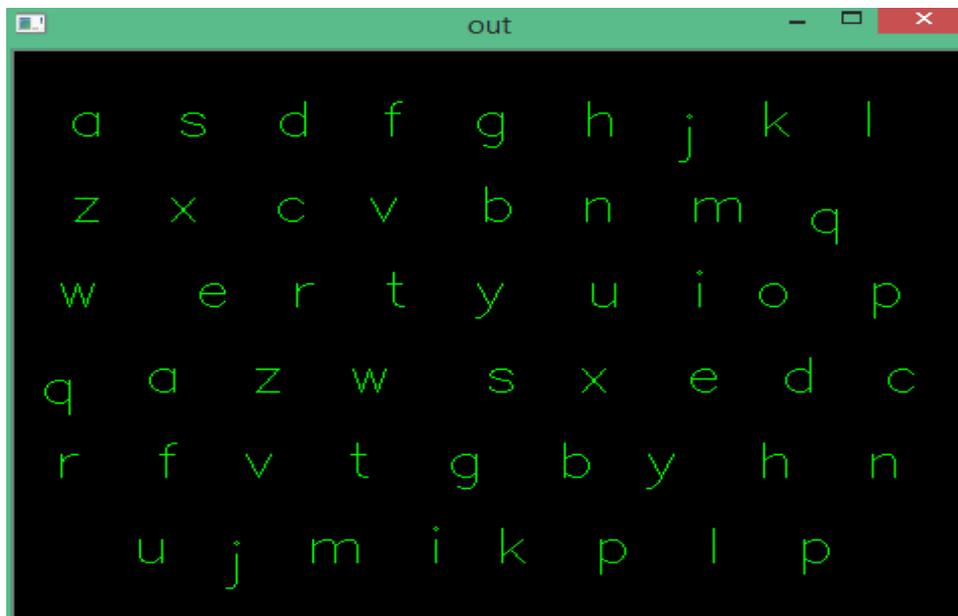
Na etapa 5, desenvolveu-se o sistema propriamente dito, sendo que o primeiro passo consistiu na aquisição de arquivos de imagens e foi, então, desenvolvido um algoritmo para processamento de imagem. Em seguida, foram esquematizado os algoritmos, a RNA do tipo MLP, SVM e KNN para treinamento posterior com arquivos de imagens de caracteres. Utilizou-se a RNA, SVM e KNN fornecida pelo próprio *OpenCV*.

Para caracteres impressos, foi utilizado o arquivo de imagem de caracteres no formato jpg com dimensões 473x371, com tamanho de 14 KB monocromático para treinamento e validação. A figura 16 ilustra uma sequência de caracteres segmentados que foi utilizado como treinamento.



**Figura 16:** Caracteres segmentados utilizados como treinamento.  
Fonte: Elaborado pelo Autor (2015).

A figura 17 ilustra uma sequência de caracteres reconhecidos após treinamento.



**Figura 17:** Caracteres reconhecidos.  
Fonte: Elaborado pelo Autor (2015).

Após a implementação de cada algoritmo, realizava-se a atividade 6, ou seja, eram efetuados testes para verificar o funcionamento e validar o que foi desenvolvido. Da mesma forma, ao longo de todas as etapas de desenvolvimento do sistema foi executada a atividade 6.

Utilizou-se também arquivo da base UCI (*Repository of machine learning*) criada por Slate (1991), para avaliar os algoritmos desenvolvidos. O conjunto de dados consiste em 20.000 vetores de características, juntamente com a respostas de letras latinas A-Z.

As primeiras 10.000 amostras são usadas para treinamento e o restante, as outras 10.000, para testar os classificadores.

## 4: RESULTADOS E DISCUSSÕES

Os resultados apresentados pelo processo de classificação e reconhecimento de caracteres ópticos consideram o modelo básico dos métodos (Redes Neurais Artificiais (RNA), Máquinas de Suporte de Vetores (SVM) e K-Vizinhos mais próximos analisados, buscando suas variações de melhores resultados.

As etapas de treinamento e testes efetuadas nos algoritmos foram realizadas utilizando 50% das amostras para treinar os classificadores e o restante para classificação posterior. Os experimentos foram executados 10 vezes com diferentes conjuntos de treinamento e avaliação com o objetivo de identificar, através do cálculo, a taxa média de acerto, bem como os tempos médios de execução de cada classificador.

Para validação dos classificadores e comprovação de sua eficiência e eficácia, o mesmo foi exposto a um processo de treinamento e experimento. Após treinamento, o algoritmo foi submetido à variadas simulações, foram utilizadas as informações da base UCI, previamente retiradas da mesma base de dados utilizadas para testes. Contudo, os resultados obtidos nos testes permitiram sua submissão à variadas análises, o que tornou possível avaliar seu desempenho e aplicabilidade.

A figura 18 apresenta exemplos de possíveis entradas para o propósito deste trabalho.

**A S D F G H J K L**  
**Z X C V B N M Q**  
**W E R T Y U I O P**  
**Q A Z W S X E D C**  
**R F V T G B Y H N**  
**U J M I K O L P**

**Figura 18:** Amostra de Possíveis Entradas para os Métodos.  
 Fonte: Elaborado pelo Autor (2015)..

Os procedimentos para preparação e pré-processamento, bem como as particularidades de cada algoritmo, não são discutidas, devendo cada um dos desses ser estudado separadamente - caso seja necessário. Cada algoritmo, depois de treinado, foi exposto a 10 testes, sendo que o resultado de cada um foi apresentado separadamente como segue nos tópicos a seguir.

#### 4.1: CLASSIFICADOR KNN

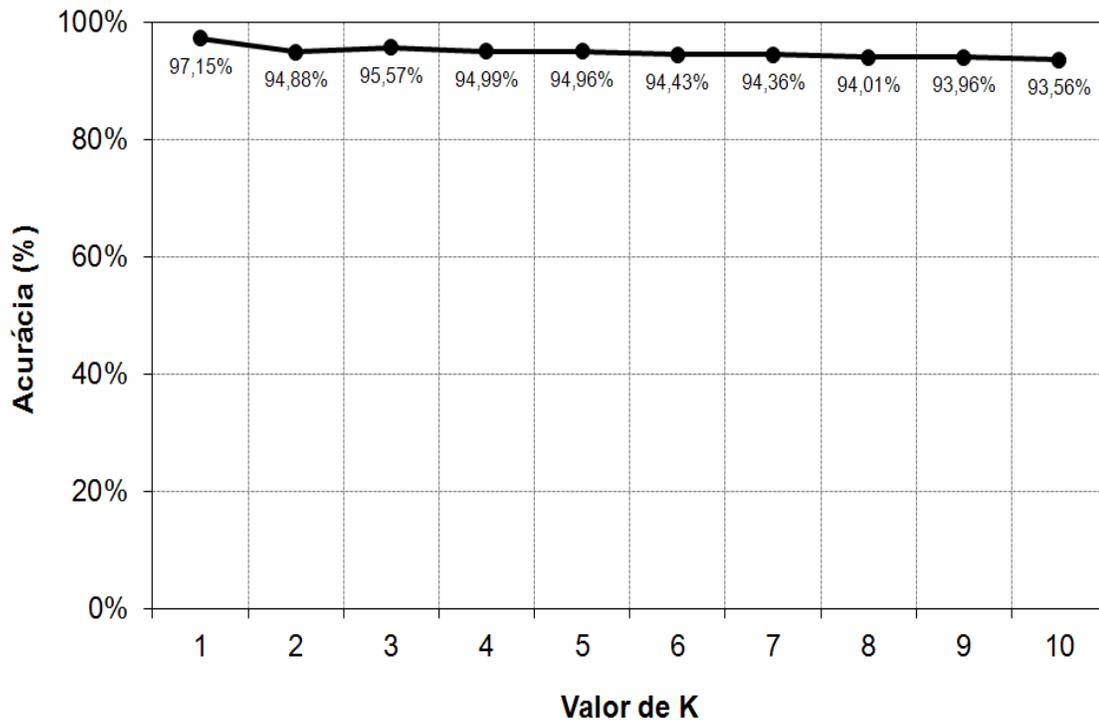
Depois de aplicados os procedimentos necessários, foram obtidos os valores descritos na Tabela 1.

**Tabela 1** - Resultados para a variação do valor de K. Fonte: O Autor.

<b>CLASSIFICADOR KNN</b>				
<b>Parâmetro: K variando 1 a 10</b>				
<b>Valor de K</b>	<b>Treino %</b>	<b>Teste%</b>	<b>Acurácia %</b>	<b>Tempo de Treinamento</b>
1	100,00	94,30	97,15	0:00:01
2	97,10	92,67	94,88	0:00:00
3	97,75	93,39	95,57	0:00:00
4	96,74	93,25	94,99	0:00:00
5	96,70	93,22	94,96	0:00:00
6	95,93	92,93	94,43	0:00:00
7	95,93	92,78	94,36	0:00:00
8	95,39	92,63	94,01	0:00:00
9	95,47	92,45	93,96	0:00:00
10	95,07	92,05	93,56	0:00:00

Fonte: Elaborado pelo Autor (2015).

O gráfico 1 ilustra a representação gráfica dos resultados da classificação e reconhecimento pelo método K-Vizinhos mais próximos.



**Gráfico 1:** Classificação KNN com variação do valor de K.  
Fonte: Elaborado pelo Autor (2015).

A classificação e reconhecimento pelo método K-Vizinhos mais próximos, apresenta evolução inversa ao aumento do valor de K, como pode ser visto no gráfico1. No entanto, o algoritmo proposto apresenta taxas de acerto de até 97,15%.

#### 4.2: CLASSIFICADOR SVM

Foram realizados testes de SVM com três funções *kernel* distintas: a *Linear*, a *Radial - Basis Function- RBF* e a *Polinomial* e, assim como no método anterior, o classificador SVM foi submetido à 10 simulações.

##### 4.2.1: SVM Kernel Linear

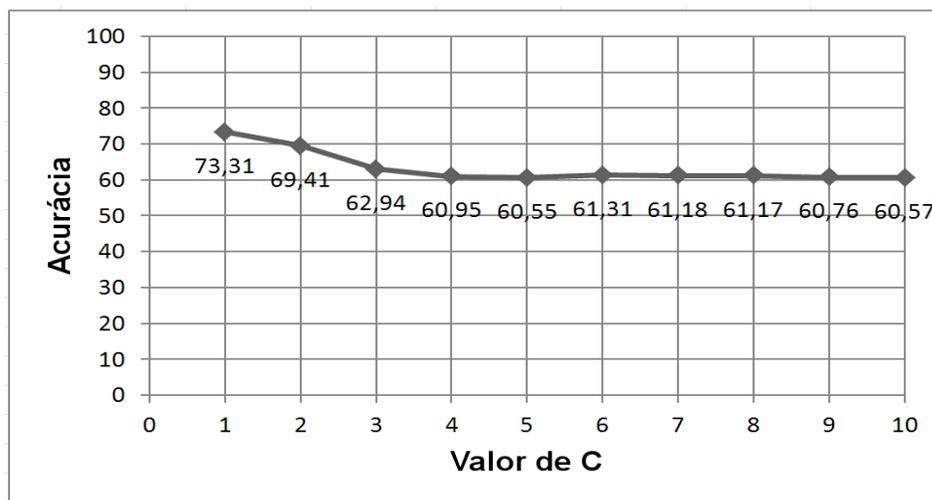
O classificador SVM com função *Kernel Linear* foi submetido à 10 simulações, tendo como configurações básicas os valores apresentados na Tabela 2, seguidos de sua respectiva taxa de acerto.

**Tabela 2:** Resultados da Classificação por SVM com função Kernel Linear.

CLASSIFICADOR SVM: Kernel LINEAR				
Parâmetros: gamma: 0,02 ; C: variando 1 a 10				
Valor de C	Treino %	Teste%	Acurácia %	Tempo de Treinamento
1	74,61	72,01	73,31	0:00:06
2	70,73	68,08	69,41	0:00:06
3	64,16	61,73	62,94	0:00:07
4	62,05	59,85	60,95	0:00:06
5	61,93	59,17	60,55	0:00:04
6	62,39	60,23	61,31	0:00:06
7	62,48	59,87	61,18	0:00:06
8	62,38	59,96	61,17	0:00:06
9	61,92	59,6	60,76	0:00:05
10	61,72	59,41	60,57	0:00:06

Fonte: Elaborado pelo Autor (2015).

Já o gráfico 2 ilustra representação gráfica dos resultados da classificação por SVM com função *Kernel Linear*.



**Gráfico 2:** Classificação SVM com função Kernel Linear.  
Fonte: Elaborado pelo Autor (2015).

A classificação e reconhecimento pelo método SVM com função *Kernel Linear*, apresenta evolução inversa ao aumento do valor de C, como pode ser visto no gráfico 2. No entanto, o algoritmo apresenta taxas de acerto de até 73,31% e tempo de treinamento de 6 segundos.

#### 4.2.2: SVM Kernel Polinomial

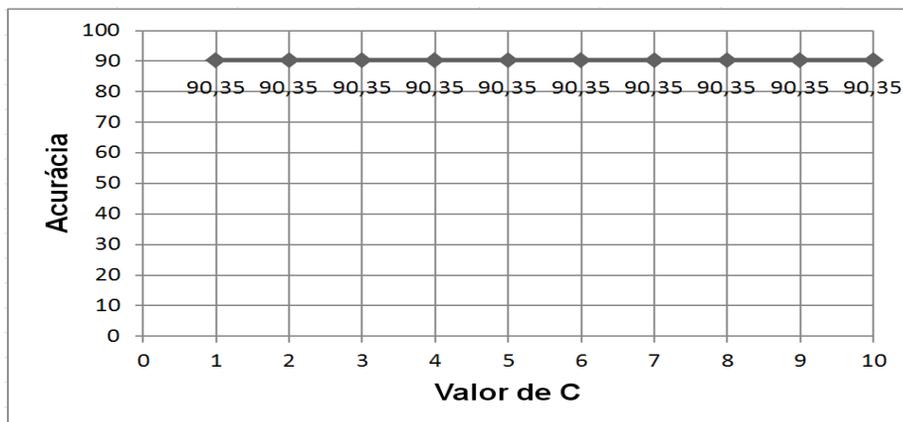
O classificador SVM com função *kernel* Polinomial foi submetido à 10 simulações, tendo como configurações básicas os valores apresentados na Tabela 3, seguidos de sua respectiva taxa de acerto.

**Tabela 3:** Resultados da Classificação por SVM com função Kernel Polinomial.

CLASSIFICADOR SVM: Kernel POLINOMIAL				
Parâmetros: gamma:2,0 ; Degree=2.3; C:variando 1 a 10				
Valor de C	Treino %	Teste%	Acurácia %	Tempo de Treinamento
1	92,65	88,06	90,35	0:00:06
2	92,65	88,06	90,35	0:00:06
3	92,65	88,06	90,35	0:00:06
4	92,65	88,06	90,35	0:00:07
5	92,65	88,06	90,35	0:00:07
6	92,65	88,06	90,35	0:00:07
7	92,65	88,06	90,35	0:00:07
8	92,65	88,06	90,35	0:00:06
9	92,65	88,06	90,35	0:00:07
10	92,65	88,06	90,35	0:00:07

Fonte: Elaborado pelo Autor (2015).

O gráfico 3 ilustra representação gráfica dos resultados da classificação por SVM com função *Kernel* Polinomial.



**Gráfico 3:** Classificação SVM com função Kernel Polinomial.  
Fonte: Elaborado pelo Autor (2015).

A classificação e reconhecimento pelo método SVM com função *Kernel* Polinomial é constante com o aumento do valor de C e não apresenta alteração, como pode ser visto no gráfico 3. No entanto, o algoritmo apresenta taxas de acerto de até 90,35% e tempo de treinamento de 7 segundos.

#### 4.2.3: SVM Kernel RBF

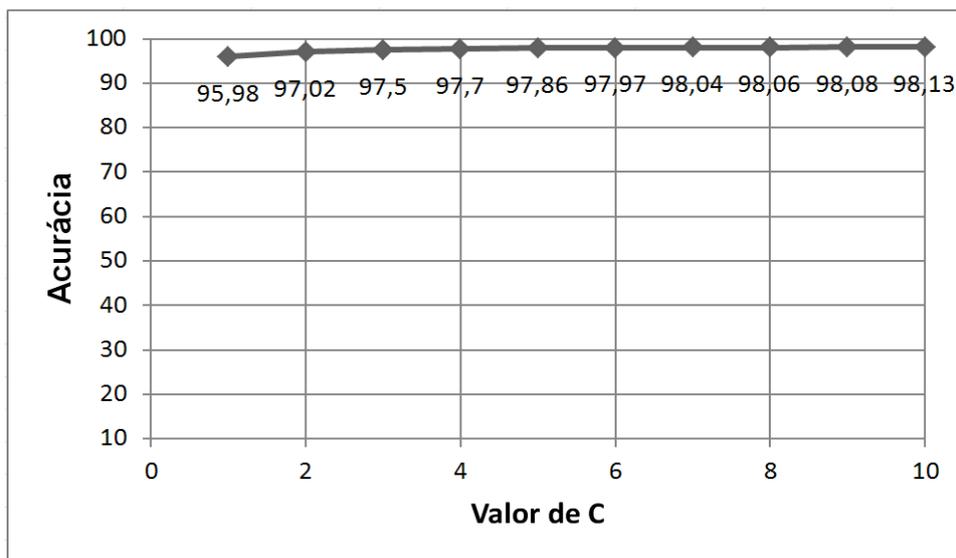
O classificador SVM com função *Kernel* RBF foi submetido à 10 simulações, tendo como configurações básicas os valores apresentados na Tabela 4, seguidos de sua respectiva taxa de acerto.

**Tabela 4:** Resultados da Classificação por SVM com função Kernel RBF.

CLASSIFICADOR SVM: Kernel RBF				
Parâmetros: gamma:0,02 ; C:variando 1 a 10				
Valor de C	Treino %	Teste%	Acurácia %	Tempo de Treinamento
1	97,34	94,62	95,98	0:00:03
2	98,41	95,62	97,02	0:00:03
3	98,91	96,1	97,5	0:00:04
4	99,28	96,12	97,7	0:00:03
5	99,42	96,29	97,86	0:00:03
6	99,55	96,4	97,97	0:00:03
7	99,57	96,5	98,04	0:00:03
8	99,68	96,43	98,06	0:00:03
9	99,71	96,46	98,08	0:00:03
10	99,75	96,51	98,13	0:00:03

Fonte: Elaborado pelo Autor (2015).

O gráfico 4 ilustra os resultados da classificação por SVM com função Kernel RBF.



**Gráfico 4:** Classificação SVM com função Kernel RBF.  
Fonte: Elaborado pelo Autor (2015).

A classificação e reconhecimento pelo método SVM com função *kernel* RBF, apresenta evolução crescente com o aumentar do valor de C, como pode ser visto no gráfico 4. No entanto, o algoritmo apresenta taxas de acerto de até 98,13% e tempo de treinamento de 3 segundos.

Depois de realizados os testes, foram obtidos melhor resultado ao ser utilizada a função *kernel* RBF, com 98,13% de acerto. A Tabela 5 apresenta os melhores resultados para cada uma das funções utilizadas no *kernel*.

**Tabela 5:** Resultados da Classificação por SVM.

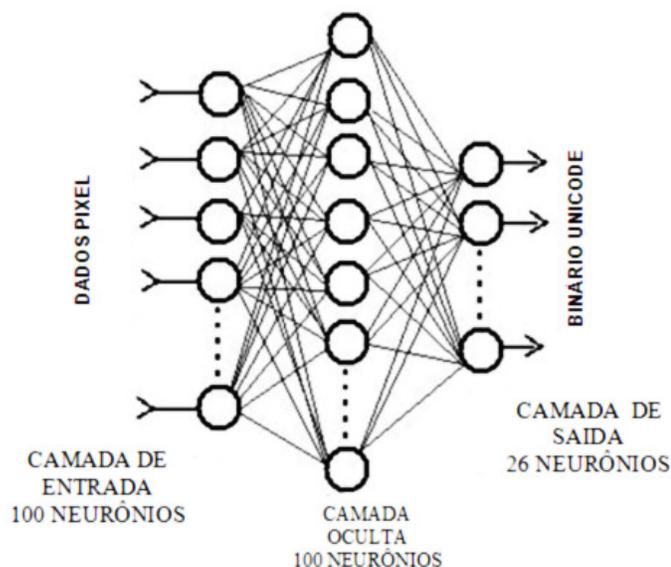
Classificador	Função Linear %	Função RBF %	Função Polinomial %
SVM	73,31	98,13	90,35

Fonte: Elaborado pelo Autor (2015).

Todos os resultados apresentados em cada uma das etapas do teste serão considerados para análises estatísticas, para obtenção de uma melhor avaliação do método.

#### 4.3: CLASSIFICADOR RNA

Para o referido trabalho, foi utilizada uma Rede Neural Artificial *Perceptron* Multicamadas que é composta por três camadas, sendo que uma dessas é a camada entrada, a outra é camada oculta e outra ainda que é a camada de saída. Logo, a RNA utilizada neste projeto possui a seguinte configuração: Camada de entrada - É composta de 100 neurônios; Camada oculta - É composta de 100 neurônios com valor definido, baseando-se nos melhores resultados obtidos nos testes; E a camada de saída - É composta de 26 neurônios, um para cada letra do alfabeto brasileiro. A figura 19 ilustra a topologia da rede. Foi utilizada sigmoide como função de ativação e 0,7 de taxa de aprendizado.



**Figura 19:** Rede MLP do Projeto.  
Fonte: Elaborado pelo Autor (2015).

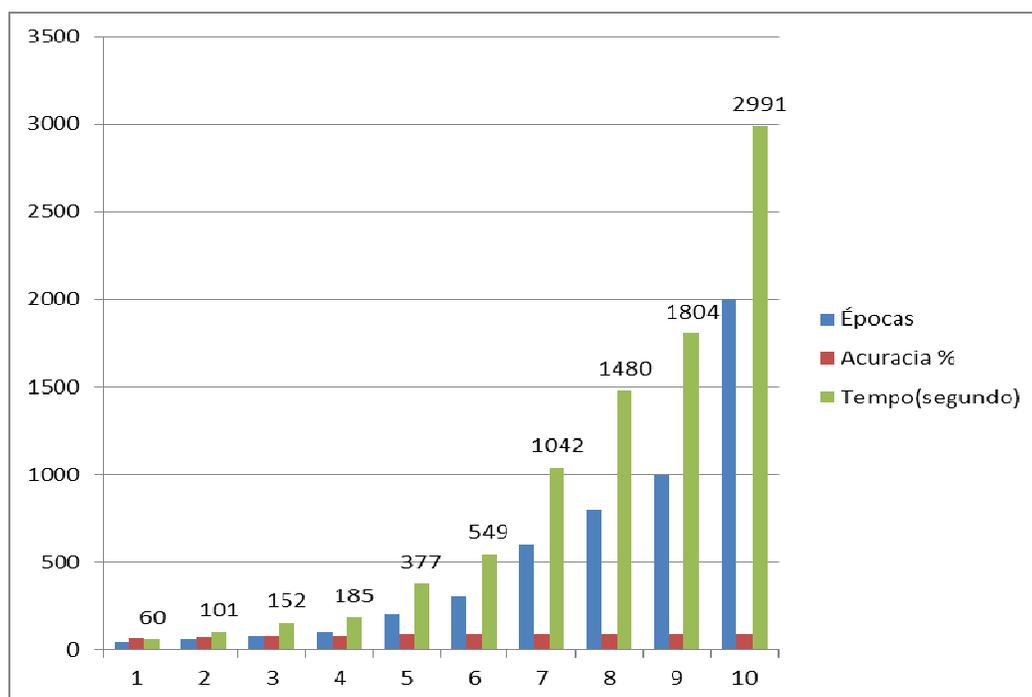
Os resultados dos testes realizados com a RNA são apresentados na Tabela 6 no qual o número de épocas foi alterado de acordo com a cardinalidade desse.

**Tabela 6:** Resultados para a variação no número de épocas.

RNA: MLP					
Parâmetros: Aprendizado=0,01; Camadas: Entrada=100; Oculta=100; Saída=26					
Nº do Teste	Épocas	Treino %	Teste%	Acurácia %	Tempo de Treinamento
1	40	71,46	70,25	70,86	60
2	60	77,93	77,10	77,52	101
3	80	83,53	81,85	82,69	152
4	100	85,44	84,12	84,78	185
5	200	93,13	90,74	91,94	377
6	300	94,53	92,09	93,31	549
7	600	96,15	93,80	94,97	1042
8	800	97,33	94,53	95,93	1480
9	1000	97,67	95,09	96,38	1804
10	2000	98,55	95,67	97,11	2991

Fonte: Elaborado pelo Autor (2015).

O gráfico 5 ilustra a representação gráfica da RNA para a variação no número de épocas.

**Gráfico 5:** Representação gráfica da RNA para a variação no número de épocas.

Fonte: Elaborado pelo Autor (2015).

No método RNA, quanto maior o valor da época (iterações), maior será a acurácia, em contrapartida, tem-se um considerável aumento no tempo de treinamento que é demonstrado sob forma de evolução no gráfico 5.

O melhor resultado foi apresentado quando o número de épocas foi estendido a 2.000 e o algoritmo de acerto médio obtido foi de 97,11%. Foram também realizados testes com números superiores de épocas, entretanto, a variação da taxa de acerto não foi significativamente expressiva.

#### **4.3.1: Influência da variação de parâmetros**

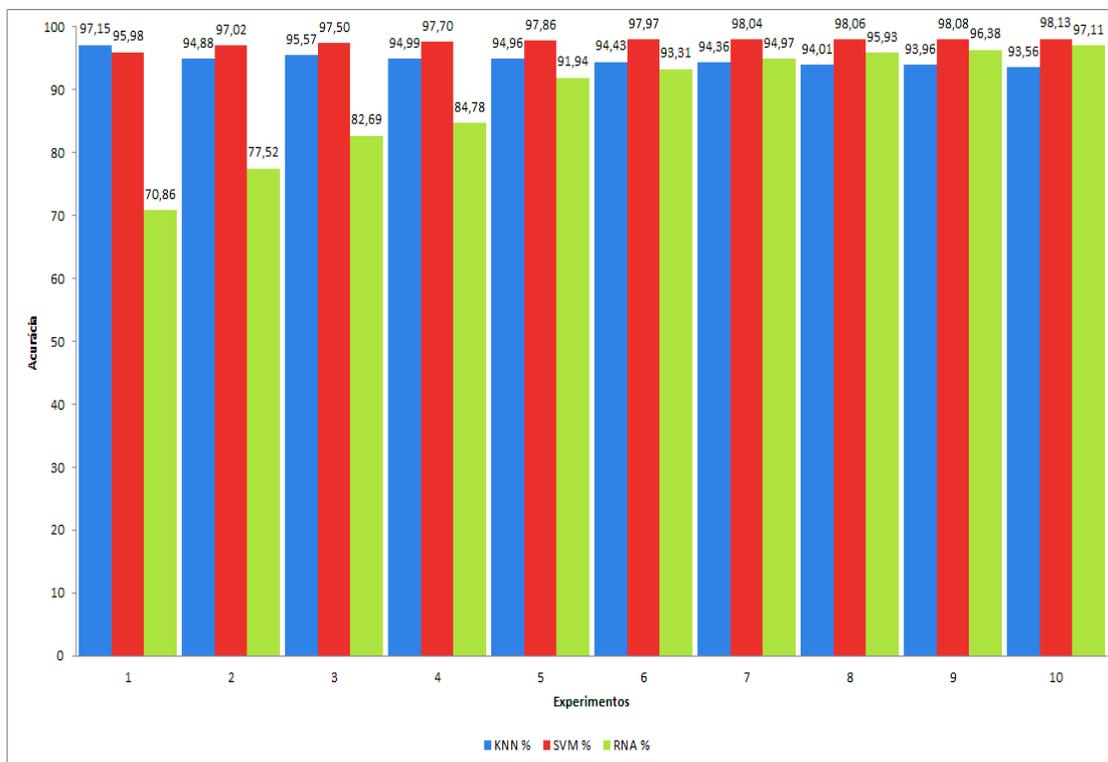
Aumentando o número de iterações, geralmente, há uma relação de proporcionalidade positiva para o desempenho da rede. No entanto, em certos casos, aumentando ainda mais o número de iterações se tem um efeito adverso de introdução de um maior número de reconhecimentos errados. Isso pode ser atribuído, em parte, ao valor elevado do parâmetro taxa aprendizagem, uma vez que como a rede se aproxima dos limites ideais, mais pesos resultam em alterações - ignorando o estado ideal. Com mais iterações, a rede vai tentar 'oscilar' de volta para o estado desejado para, novamente, continuar com uma boa chance de perder o estado ideal na época final. Esse fenômeno é conhecido como o excesso de aprendizagem.

O tamanho dos estados da entrada também é outro fator que influencia o desempenho direto. É natural que o maior número de símbolos de entrada defina que a rede deva ser treinada para ser mais sensíveis aos erros. Usualmente, o complexo conjunto de tamanhos - maiores - de entrada exige uma topologia de rede grande, com mais números de iterações.

A variação do parâmetro taxa de aprendizagem também afeta o desempenho da rede para um determinado limite de iterações. Quanto menor o valor desse parâmetro, menor será o valor que da rede ao atualizar seus pesos. Isso implica, intuitivamente, que será menos provável enfrentar a dificuldade de aprendizagem discutida acima, uma vez que serão atualizadas as suas ligações lentamente e de uma forma mais refinada.

#### 4.4: COMPARAÇÃO ESTATÍSTICA ENTRE OS MÉTODOS.

Todos os métodos foram submetidos á várias análises estatísticas, além de comparativos mais simples como a comparação direta entre os resultados, como pode ser observado no gráfico 6:



**Gráfico 6:** Gráfico da comparação direta entre os métodos.  
Fonte: Elaborado pelo Autor (2015).

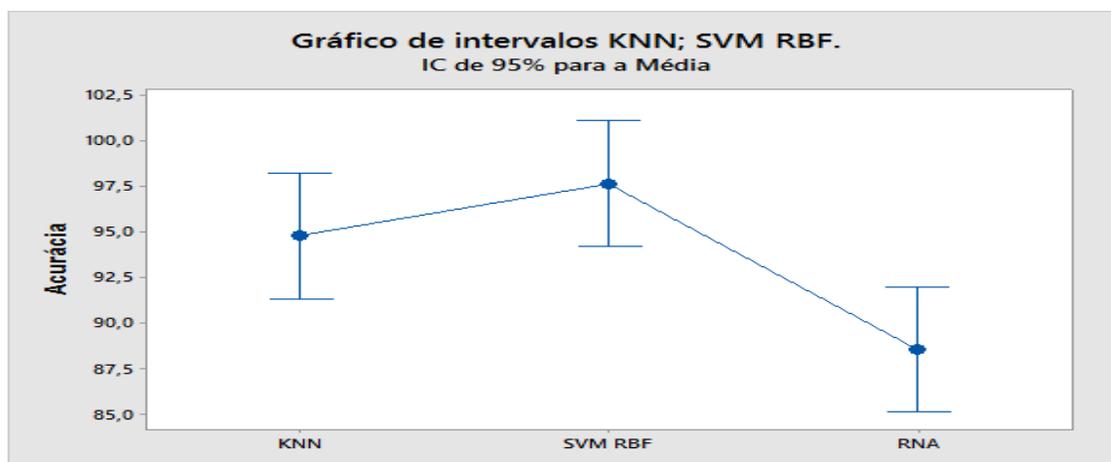
À luz de uma comparação direta, nota-se pouca diferença entre as taxas de acerto dos métodos analisados, sendo que os melhores são apresentados na seguinte ordem: SVM, KNN e RNA, respectivamente. No entanto, para que se possa afirmar que um método, apesar de apresentar maior taxa de acerto, é ou não mais adequado do que outro para o teste realizado, é necessário aplicar outros métodos de comparação estatística, sendo que o método de comparação aplicado neste trabalho foi a Análise de Variância (ANOVA), juntamente com o teste de Tukey para classificação. Pode-se observar na Tabela 7, o pareamento dos dados obtidos durante os 10 testes realizados em cada um dos métodos analisados. Observa-se, também, o desvio padrão entre os valores.

**Tabela 7:** Pareamento dos Resultados e Desvio Padrão dos Métodos.

	<b>KNN</b>	<b>SVM RBF</b>	<b>RNA</b>
	97,15	95,98	70,86
	94,88	97,02	77,52
	95,57	97,50	82,69
	94,99	97,70	84,78
	94,96	97,86	91,94
	94,43	97,97	93,31
	94,36	98,04	94,97
	94,01	98,06	95,93
	93,96	98,08	96,38
	93,56	98,13	97,11
<b>SOMA</b>	<b>947,87</b>	<b>976,34</b>	<b>885,49</b>
<b>DESVIO PADRÃO</b>	<b>1,02087</b>	<b>0,674145</b>	<b>9,11386</b>

Fonte: Elaborado pelo Autor (2015).

Através do desvio padrão é possível medir o quanto de “dispersão” existe entre os dados em relação à média. Como se pode observar no gráfico 7, os dados tendem a estar próximos da média, o que aumenta a confiabilidade nos resultados. E o desvio padrão combinado foi usado para calcular os intervalos.



**Gráfico 7:** Gráfico de intervalos KNN, SVM RBF e RNA.  
Fonte: Elaborado pelo Autor (2015).

Foi realizado também o cálculo da variância dos dados coletados, sendo que o resultado obtido foi apresentado na Tabela 8.

**Tabela 8:** Cálculo da Variância.

RESUMO				
<i>Grupo</i>	<i>Contagem</i>	<i>Soma</i>	<i>Média</i>	<i>Variância</i>
KNN	10	947,87	94,787	1,042178889
SVM	10	976,34	97,634	0,454471111
RNA	10	885,49	88,549	83,06249889

Fonte: Elaborado pelo Autor (2015).

O cálculo da variância é fundamental para realização da análise de variância, e esse teste que permite definir e quantificar o valor de uma média que é ou não diferença estatística entre as médias comparadas. A Tabela 9 apresenta o resultado do teste de análise de variância, juntamente com a classificação realizada pelo teste de Tukey, o que permite agrupar os testes em níveis.

**Tabela 9:** Análise de Variância e Classificação por Teste de Tukey

ANOVA							TUKEY			
<i>Fonte da variação</i>	<i>SQ</i>	<i>gl</i>	<i>MQ</i>	<i>F</i>	<i>valor-P</i>	<i>F crítico</i>	<i>Fator</i>	<i>N</i>	<i>Média</i>	<i>Agrupamento</i>
Entre grupos	431,8509267	2	215,9254633	7,66063044	0,0023167	3,3541	SVM	10	97,634	A
Dentro dos grupos	761,03234	27	28,18638296				KNN	10	94,787	A
Total	1192,883267	29					RNA	10	88,549	B

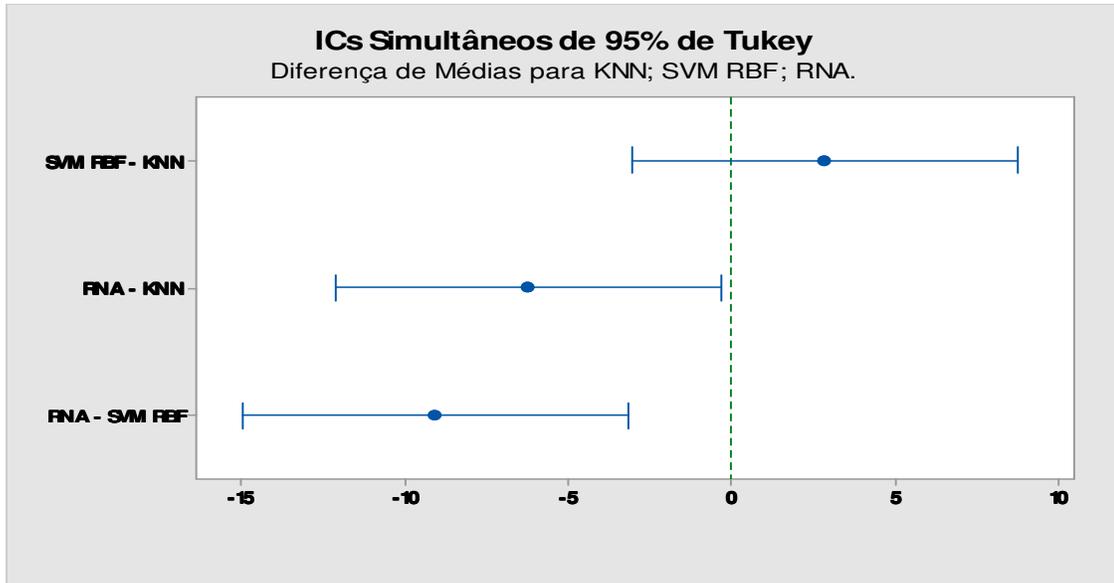
ANOVA – Análise de Variância

(b)TUKEY – Classificação

Fonte: Elaborado pelo Autor (2015).

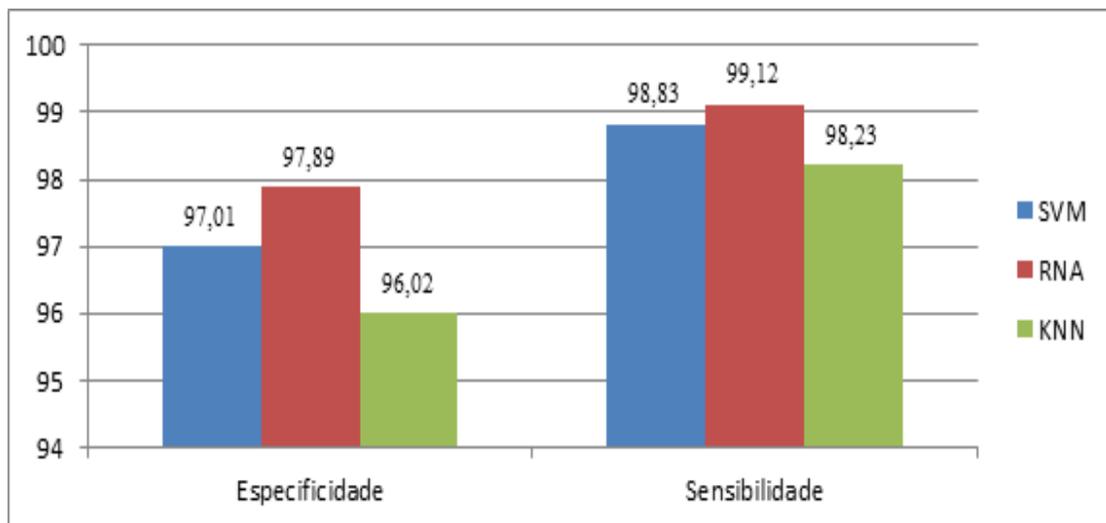
O resultado da análise de variância, juntamente com o teste de Tukey demonstra que existe diferença estatística entre as médias dos métodos SVM e RNA, assim como existem diferença estatística entre os métodos RNA e KNN considerando os resultados do agrupamento (A e B). No entanto, entre os métodos SVM e KNN não existem estatisticamente diferenças entre as médias (A e A). O gráfico 8 ilustra a representação gráfica da análise das comparações emparelhadas de Tukey, agrupamento usando método de Tukey e Confiança de 95%. O intervalo

que não contiver o zero, as médias correspondentes serão significativamente diferentes.



**Gráfico 8:** Gráfico de Diferença de Médias para KNN, SVM RBF e RNA.  
Fonte: Elaborado pelo Autor (2015).

Os métodos também foram testados quanto à sua especificidade e à sua sensibilidade. O gráfico 9 ilustra a representação gráfica dos resultados.



**Gráfico 9:** Gráfico Resultado Especificidade e Sensibilidade.  
Fonte: Elaborado pelo Autor (2015).

A especificidade e a sensibilidade medem, respectivamente, o número de vezes em que os métodos indicaram um carácter e realmente fizeram a indicação correta, quantas vezes indicaram ser outro carácter e não o carácter testado que realmente fizeram uma verdadeira indicação negativa. Testes como esses são, frequentemente, aplicados em questões relacionadas aos diagnósticos, procurando fornecer mais do que dados sobre a taxa de acerto, mas também validar as indicações positivas e negativas.

## 5: CONCLUSÕES

Após verificação e análise dos resultados, é possível observar que não existem parâmetros estatísticos, quando considerados os testes ANOVA e TUKEY realizados na pesquisa que permitam afirmar a existência de diferenças no percentual de acertos entre os métodos KNN e SVM, uma vez que, os ambos se encontram na mesma escala de classificação de Tukey - de acordo com a Tabela 9. Entretanto, comparados ao método RNA, o método SVM e o KNN apresentam maior eficácia, considerando que sua média de acertos é significativamente maior - de acordo com o Gráfico 8.

Quanto aos resultados para a variação do valor de K. – Classificador KNN, conforme observado no Gráfico 1, o algoritmo proposto apresenta taxas de acerto de até 97,15%.

Em relação à classificação e reconhecimento pelo método SVM com função kernel Linear, foi observada uma evolução inversa ao aumento do valor de C, o que está exposto no Gráfico 2. O algoritmo apresentou taxas de acerto de até 73,31% e tempo de treinamento de 6 segundos.

Com relação à classificação e reconhecimento pelo método SVM com função kernel Polinomial, foi observado um valor constante com o aumento do valor de C, o mesmo não apresentou alteração - como pode ser visto no Gráfico 3. No entanto, o algoritmo apresenta taxas de acerto de até 90,35% e tempo de treinamento de 7 segundos.

Considerando a classificação e o reconhecimento pelo método SVM com função kernel RBF, o resultado foi a evolução crescente com o aumentar do valor de

C, conforme visto no Gráfico 4. No entanto, o algoritmo apresentou taxas de acerto de até 98,13% e tempo de treinamento de 3 segundos.

Após resultados com os testes kernel, observou-se que se alcançou melhor resultado ao se utilizar a função kernel RBF, com esse foram obtidos 98,13% de acerto. A Tabela 5 apresenta os melhores resultados para cada uma das funções utilizadas no kernel.

Considerando os testes do método RNA, quanto maior o valor da época (iterações), maior será a acurácia, apresentando um considerável aumento no tempo de treinamento - conforme o Gráfico 5. Percebe-se que o melhor resultado apresentado foi quando o número de épocas foi alterado para 2.000, nesse o algoritmo obteve acerto médio de 97,11%. Foram, também, realizados testes com números superiores de épocas, entretanto, a variação da taxa de acerto não foi significativamente expressiva.

Assim sendo, considerando as diferentes técnicas de inteligência computacional, é possível obter o reconhecimento eficaz de caracteres com rapidez e segurança comparável à flexibilidade humana. Dessa forma, o processo composto pelo pré-processamento, a seleção, a classificação de imagens de caracteres e pelo aprendizado - com posterior reconhecimento através de SVM, RNA e KNN - perfaz-se a simulação de muitas tarefas monótonas ao homem na atualidade.

Isso porque atualmente, funções operacionais cotidianas poderiam ser usadas pelo reconhecimento de caracteres que livrariam o homem de rotinas trabalhosas como, por exemplo: a conferência de assinaturas, separação de correspondência nos correios, controle de infrações de trânsito e transcrição automática de textos. Como se pode perceber, o estudo das diferentes técnicas de inteligência computacional abre um novo caminho para novas relações no trabalho humano, retirando desse a parte operacional para que ele possa centrar sua atenção na parte criativa, dando mais sentido à natureza humana.

Como trabalho futuro, propõe-se a formação de outras bases para treinamento e testes para caracteres ópticos impressos e manuscritos, para avaliar os algoritmos e confirmar a robustez desses classificadores. Com relação ao contexto técnico, os resultados deste trabalho poderão servir de base para o desenvolvimento de futuros projetos, relacionados à mesma linha.

## 6: REFERÊNCIAS BIBLIOGRÁFICAS

ABAYNARH, M.; ELFADILI, H.; ZENKOUAR, L. Handwritten characters classification using neural networks and moments features. **International Journal of Modern Engineering Research (IJMER)**. Agdal Rabat, Morocco, v.2, n.5, p.3572-3577, sep-oct. 2012. Disponível em: <[http://www.ijmer.com/papers/Vol2\\_Issue5/CV2535723577.pdf](http://www.ijmer.com/papers/Vol2_Issue5/CV2535723577.pdf)>. Acesso em: 14 jun 2014.

AHA, D.W.; KIBLER, D.; ALBERT, M.K. **Instance-based learning algorithms**. Boston-US: Machine Learning, 1991.

AIRES, S. B. K. **Reconhecimento de caracteres manuscritos baseado em regiões perceptivas**. 2005. 97 f. Dissertação (Mestrado em Informática Aplicada) - Pontifícia Universidade Católica do Paraná, Curitiba. 2005. Disponível em: <[https://www.ppgia.pucpr.br/pt/arquivos/mestrado/dissertacoes/2005/2005\\_simone\\_bello\\_kaminski.pdf](https://www.ppgia.pucpr.br/pt/arquivos/mestrado/dissertacoes/2005/2005_simone_bello_kaminski.pdf)>. Acesso em: 14 jun 2014.

ALBUQUERQUE, M. P.; ALBUQUERQUE, M. P. **Processamento de imagens: métodos e análises**. Disponível em: <<http://www.cbpf.br/cat/lpdsi/pdf/ProcessamentoImagens.PDF>>. Acesso em: 27 fev. 2015.

ALPAYDIN, E. **Introduction to machine learning**. Massachusetts – London: The MIT Press Cambridge, 2010.

\_\_\_\_\_. **Introduction to machine learning: adaptive computation and machine learning**. 2. ed. Massachusetts – London: The MIT Press Cambridge, 2004.

ARORA, S. et al. Performance comparison of SVM and ANN for handwritten devnagari character recognition. **International Journal of Computer Science Issues**, v. 7, n. 3, may, 2010. Disponível em: <<https://arxiv.org/ftp/arxiv/papers/1006/1006.5902.pdf> >. Acesso em: 27 fev. 2015.

ASIR, M. K.; UDDIN, M. S. Hand written bangla numerals recognition for automated postal system. **IOSR Journal of Computer Engineering (IOSR-JCE)**. volume 8, n.6, p.43-48, Jan-Feb, 2013. Disponível em: <<http://www.iosrjournals.org/iosr-jce/pages/v8i6.html>>. Acesso em: 27 fev. 2015

BARONE, D. A. C.; BAZZAN, A. L. C. **Sociedades artificiais: a nova fronteira da inteligência nas máquinas**. Porto Alegre: Bookman, 2003.

BHATIA, N. Optical character recognition techniques: a review. **International Journal of Advanced Research in Computer Science and Software Engineering**, V. 4, Issue 5, May 2014. Disponível em: <[http://www.ijarcsse.com/docs/papers/Volume\\_4/5\\_May2014/V4I5-0677.pdf](http://www.ijarcsse.com/docs/papers/Volume_4/5_May2014/V4I5-0677.pdf)>. Acesso em: 27 fev. 2015

BAYAT, A. B. Recognition of handwritten digits using optimized adaptive neuro-fuzzy inference systems and effective features. **Journal of Pattern Recognition and Intelligent Systems**, V. 1, p.25-37, aug. 2013. Disponível em: <<http://www.academicpub.org/pris/paperInfo.aspx?PaperID=14532>>. Acesso em: 27 fev. 2015

BRADSKI, G.; KAEHLER, A. **Learning OpenCV: computer vision with the opencv library**. Sebastopol, California-US: O' Reilly, 2008.

BRAGA, A.P.; CARVALHO, A. C. P. L. F.; LUDERMIR, T. B. **Redes neurais artificiais: teoria e aplicações**. Rio de Janeiro: LTC, 2000.

BURGES, C. J. C. A tutorial on support vector machines for pattern recognition. **Data Mining Knowledge Discovery in Databases**, v.2, n.121, p.167, jun. 1998. Disponível em: <<http://research.microsoft.com/pubs/67119/svmtutorial.pdf>>. Acesso em: 27 fev. 2015.

CRISTIANINI, N.; SHAWE-TAYLOR, J. **An introduction to support vector machines and other kernel-based learning methods**. New York: Cambridge University Press. 2000.

FENWA, O. D.; ALADE, M. O.; ADEDEJI, O. T. Evaluation of performance of PSO-based kernel support vector machine in offline digit recognition. **International Journal for Advance Research in Engineering and Technology**, v.2,jul; 2014. Disponível em: <<http://www.ijaret.org/EVALUATION%20OF%20PERFORMANCE%20OF%20PSO-BASED%20KERNEL%20SUPPORT%20VECTOR%20MACHINE%20IN%20OFFLINE%20DIGIT%20RECOGNITION.pdf>>. Acesso em: 27 fev. 2015

GOLDSCHMIDT. R. R. **Uma introdução à inteligência computacional: fundamentos, ferramentas e aplicações**. Rio de Janeiro: IST-Rio, 2010.

GONZALEZ, R. C.; WOODS, R. E. **Digital image processing**. 3.ed. New York-US: Addison-Wesley, 1992.

GUIMARÃES, S. J. F. **Processamento digital de imagens**. Disponível em: <[http://inf.pucminas.br/projetos/viplab/team/prof/silvio/disc/pdi/slides/01\\_conceitos.pdf](http://inf.pucminas.br/projetos/viplab/team/prof/silvio/disc/pdi/slides/01_conceitos.pdf)>. Acesso em: 26 Fev 2015.

GUNN, S. R. **Support vector machine for classification and regression**. Southampton-UK: University of Southampton, 1998. Disponível em: <<http://users.ecs.soton.ac.uk/srg/publications/pdf/SVM.pdf>>. Acesso em 24 fev. 2015.

HAYKIN, S. **Redes neurais: princípios e prática**. 2. ed. Porto Alegre: Bookman, 2001.

HUGO, M. **Uma interface de reconhecimento de voz para o sistema de gerenciamento de central de informação de fretes**. 1995. 120f. Dissertação (Mestrado em Engenharia) - Universidade Federal de Santa Catarina, Florianópolis, 1995.

JAIN L. C; et al.. **Computational intelligence paradigms: innovative applications**. Berlin: Springer-Verlag, 2008.

JAWAHAR, C. V. et al. **Recognition of indian language characters using support vectors machines**: technical report TR-CVIT-22. Bangalore-Índia: International Institute of Information Technology Hyderabad, 2002.

JENSEN, J. R. **Remote sensing of the environment: an earth resource perspective**. New Jersey: Prentice Hall, 2000.

KRUSE, R. et al. **Computational Intelligence: a methodological introduction**. New York: Springer, 2013.

KUMBHAR, M. S.; CHANDRACHUD, Y. Y. Handwritten marathi character recognition using neural network. **International Journal of Emerging Technology and Advanced Engineering**, v. 2, n. 9, sep., 2012. Disponível em: <[http://www.ijetae.com/files/Volume2Issue9/IJETAE\\_0912\\_07.pdf](http://www.ijetae.com/files/Volume2Issue9/IJETAE_0912_07.pdf)>. Acesso em 24 fev. 2015.

LECUN, Y.; BENGIO, Y. Pattern recognition and neural networks. In: ARBIB, Michael A (ed.). **The handbook of brain theory and neural networks**. Massachusetts: Massachusetts Institute of Technology. 2003. Disponível em: <[http://www.federaljack.com/ebooks/My%20collection%20of%20medical%20books,%20208%20Books%20\(part%203%20of%203\)/The%20Handbook%20of%20Brain%20Theory%20and%20Neural%20Networks%20-%20ISBN%200262511029.pdf](http://www.federaljack.com/ebooks/My%20collection%20of%20medical%20books,%20208%20Books%20(part%203%20of%203)/The%20Handbook%20of%20Brain%20Theory%20and%20Neural%20Networks%20-%20ISBN%200262511029.pdf)>. Acesso em 24 fev. 2015.

LEOESCH, C.; SARI, S. T. **Redes neurais artificiais: fundamentos e modelos**. Blumenau: FUB, 1996

LORENA, A. C.; CARVALHO, A. C. P. L. F. Uma Introdução às support vector machines. **RITA: Revista de Informática Teórica e Aplicada**, v.14, n.2, p.43-67, 2007. Disponível em: <[http://www.seer.ufrgs.br/rita/article/view/rita\\_v14\\_n2\\_p43-67](http://www.seer.ufrgs.br/rita/article/view/rita_v14_n2_p43-67)>. Acesso em 24 fev. 2015.

MAILLARD, P. **Introdução ao processamento digital de imagens**. Disponível em: <<http://www.csr.ufmg.br/geoprocessamento/publicacoes/cursopdi.pdf>>. Acesso em: 05 de Fev. 2015.

MARQUES FILHO, O; VIEIRA NETO, H. **Processamento digital de imagens**. Rio de Janeiro: Brasport, 1999.

MARTINS, Elias Ribeiro da Silva; et al. Caracterização de lesões intersticiais de pulmão em radiograma de tórax utilizando análise local de textura. **Radiologia Brasileira**, v.38, n.6; p.421–426, 2005. Disponível em: <[http://www.scielo.br/scielo.php?pid=S0100-39842005000600008&script=sci\\_abstract&tlng=pt](http://www.scielo.br/scielo.php?pid=S0100-39842005000600008&script=sci_abstract&tlng=pt)>. Acesso em: 05 de Fev. 2015.

MASCARENHAS, N. D. A. Breve introdução ao reconhecimento estatístico de padrões. In: SIMPÓSIO DE INTELIGÊNCIA ARTIFICIAL E REUNIÃO ANUAL DA SOCIEDADE BRASILEIRA PARA O PROGRESSO DA CIÊNCIA, 39, 1987, Brasília.

**Anais....**São Paulo: SBC, 1988. p.12. Disponível em: < <http://mtc-m12.sid.inpe.br/col/sid.inpe.br/iris@1912/2005/07.18.23.52.36/doc/INPE%204206.pdf?metadataarepository=&mirror=sid.inpe.br/banon/2001/04.06.10.52.39>>. Acesso em: 05 de Fev. 2015.

MIKA, S. **Kernel fisher discriminants**. 2003. 156 f. Dissertação (Doutorado em Ciência da Computação) – Technischen Universität Berlin, Berlin, 2003. Disponível em: < [https://depositonce.tu-berlin.de/bitstream/11303/872/1/Dokument\\_32.pdf](https://depositonce.tu-berlin.de/bitstream/11303/872/1/Dokument_32.pdf)>. Acesso em: 05 fev. 2015.

MITCHELL, T.M. **Machine learning**. New York: McGraw Hill, 1997.

MOSSMANN, S. R. **Extração e análise de características de imagens digitais para construção de um modelo de reconhecimento de padrões**. 2010. 64f. Monografia (Graduação em Sistemas de Informação) - Universidade Feevale, Novo Hamburgo, 2010.

NEEBA, N.V.; et al. **Recognition of malayalam documents**. In: GOVINDARAJU, Venu (ed.); SETLUR, Srirangaraj Ranga (ed.). **Advances in pattern recognition: guide to OCR for Indic** London-UK: Scripts. Springer, 2009.

OSÓRIO, F. S. **Um estudo sobre o reconhecimento visual de caracteres através de redes neurais**. 1991. Dissertação (Mestrado em Computação) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 1991.

PAL, U.; CHAUDHURI, B. B. Indian script character recognition: a survey. **Pattern Recognition**, v. 37, p. 1887-1899, 2004. Disponível em: <[https://www.researchgate.net/publication/222197898\\_Chaudhuri\\_BB\\_Indian\\_script\\_character\\_recognition-A\\_survey\\_Pattern\\_Recognition\\_37\\_1887-1899](https://www.researchgate.net/publication/222197898_Chaudhuri_BB_Indian_script_character_recognition-A_survey_Pattern_Recognition_37_1887-1899)>. Acesso em: 05 de Fev. 2015.

PEDRINI, H. **Processamento e Análise de Imagens (MC940)**, 2015. Disponível em <[http://www.ic.unicamp.br/~helio/disciplinas/MO445/aula\\_fundamentos.pdf](http://www.ic.unicamp.br/~helio/disciplinas/MO445/aula_fundamentos.pdf)>. Acesso em: 05 fev 2015.

PEDRINI, H; SCHWARTZ, W. R. **Análise de imagens digitais: princípios, algoritmos, e aplicações**. São Paulo: Thompson Learning, 2008.

PERELMUTER, G., CARRERA, E. V., VELLASCO, M., PACHECO, A. reconhecimento de imagens bidimensionais utilizando redes neurais artificiais. In: SIMPÓSIO BRASILEIRO DE COMPUTAÇÃO GRÁFICA E PROCESSAMENTO DE IMAGENS, 8, 1995, 31 out. – 30 dez; Campinas-SP. **Anais...** Campinas-SP:

UNICAMP, 1995. p. 197-203. Disponível em: <<http://www.bv.fapesp.br/pt/auxilios/36051/anais-do-sibgrapi-95-viii-simposio-brasileiro-de-computacao-grafica-e-processamento-de-imagens/>>. Acesso em: 12 jan 2015.

PRADO JÚNIOR, A, ELFES, A. **Um projeto em reconhecimento de padrões de forma**. 1975. 70 f. Monografia (Graduação em Ciência da Computação) - Instituto Tecnológico da Aeronáutica, São José dos Campos- SP, 1975.

PRADEEP, A. *et al.* Malayalam character recognition using singular value decomposition. **International Journal of Computer Applications**, v. 92, n. 12, p. 6-11, New York-US, 2014. Disponível em: <<http://www.research.ijcaonline.org/volume92/number12/pxc3895167.pdf>>. Acesso em: 12 jan 2015.

PRATI, R. C. **Novas abordagens em aprendizado de máquina para geração de regras, classes desbalanceadas e ordenação de casos**. 2006. Tese (Doutorado em Ciência da Computação e Matemática Computacional) – Universidade de São Paulo, São Carlos, 2006. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-01092006-155445/pt-br.php>>. Acesso em: 12 jan 2014.

RODRIGUES, R. J.; THOMÉ, A. C. G. Reconhecimento de dígitos cursivos: um método de segmentação por histogramas. In: SIMPÓSIO BRASILEIRO DE REDES NEURAIS, 6., 2000, 2-5 abr., Rio de Janeiro. **Anais...** Porto Alegre: Sociedade Brasileira de Computação: 2001. Disponível em: <<http://www.sbc.org.br> >. Acesso em: 12 fev. 2014.

RUSSELL, S.; NORVING, P. **Inteligência Artificial**. 2 ed. Rio de Janeiro: Elsevier, 2004.

SAHU, V. L.; KUBDE, B. Offline english hand written character recognition using neural network. **International Journal of Scientific Research in Computer Science (IJSRCS)**, v. 1, n. 2, sep. 2013. Disponível em: <[http://www.iret.co.in/Docs/IJSCS/Volume 2/Issue1/4. Offline English Hand. written character recognition using neural network.pdf](http://www.iret.co.in/Docs/IJSCS/Volume%202/Issue1/4.Offline%20English%20Hand.written%20character%20recognition%20using%20neural%20network.pdf)>. Acesso em: 12 fev. 2014.

\_\_\_\_\_; \_\_\_\_\_. Offline hand written character recognition techniques using neural network: a review. **International Journal of Science and Research (IJSR)**, v. 2, n.1, p.87-94, jan. 2013. Disponível em: <<http://ijsr.net/archive/v2i1/IJSR13010129.pdf>>. Acesso em: 12 fev. 2014.

SALOUAN, R.; SAFI, S.; BOUIKHALENE, B. A comparison between the self-organizing maps and the support vector machines for handwritten latin numerals recognition. **International Journal of Innovation and Scientific Research**, v. 7, n. 1, p.50-56, aug, 2014. Disponível em: <<http://www.issr-journals.org/links/papers.php?journal=ijisr&application=pdf&article=IJISR-14-153-03>>. Acesso em: 12 fev. 2014

\_\_\_\_\_; \_\_\_\_\_.; \_\_\_\_\_.; Isolated handwritten roman numerals recognition using dynamic programming, naïve bayes and support vectors machines. **International Journal of Computer Applications**. v. 113, n.19, mar.; 2015. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.695.2095&rep=rep1&type=pdf>>. Acesso em: 12 fev. 2014

SCURI, A. E. **Fundamentos da imagem digital**. Disponível em: <<http://webserver2.tecgraf.puc-rio.br/~scuri/download/fid.pdf>>. Acesso em: 12. fev 2015.

SLATE, D. J. **Letter recognition data set**.. Disponível em <<http://archive.ics.uci.edu/ml/datasets/Letter+Recognition>>. Acesso em: 8 jan. 2015.

SINGH, G. *et al*. Feature extraction of gurmukhi script and numerals: a review of offline techniques. **International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)**, v.3, n.1, p.257-263; jan; 2013. Disponível em: <[http://www.ijarcsse.com/docs/papers/Volume\\_3/1\\_January2013/V3I1-0213.pdf](http://www.ijarcsse.com/docs/papers/Volume_3/1_January2013/V3I1-0213.pdf)>. Acesso em: 12. fev 2015.

SILVA, A. M. **Utilização de redes neurais artificiais para classificação de spam**. 2009. Dissertação (Mestrado em Modelagem Matemática e Computacional) - Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte, 2009.

SILVA, M. M. **Uma abordagem evolucionária para o aprendizado semi-supervisionado em máquinas de vetores de suporte**. 2008. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal de Minas Gerais, 2008.

SILVA, V. D.; SCARPEL, R. A. Detecção de fraudes na distribuição de energia elétrica utilizando support vector machine. **Investigação Operacional**, v. 27, n.2, p. 139 – 150, 2007. Disponível em: < <http://apdio.pt/documents/10180/15550/n9.pdf>>. Acesso em: 12. fev 2015.

SOK, P.; TAING, N. **Support Vector Machine (SVM) based classifier for Khmer printed character-set recognition**. In: ASSOCIATION ANNUAL SUMMIT AND CONFERENCE (APSIPA), 2014, 9-10 dec. Siem –Reap – Camboja. **Processing...** New York - US: IEEE, 2015. p.1-12 Disponível em: <[http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=7041823&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs\\_all.jsp%3Farnumber%3D7041823](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=7041823&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D7041823)>. Acesso em: 12 fev. 2015.

STEINER, M. T. A. **Uma metodologia para o reconhecimento de padrões multivariados com resposta dicotômica**. 1995. Tese (Doutorado em Engenharia de Produção) - Universidade Federal de Santa Catarina. Florianópolis-SC, 1995.

THEODORIDIS, S.; KOUTROUMBAS, K. **Pattern recognition**. 3.ed. New York-US: Academic Press, 2006.

TIWARI, S. *et al.* Optical character recognition using matlab. **International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)**. v. 2, n. 5, may, 2013. Disponível em: <<http://eprints.manipal.edu/138900/>>. Acesso em: 12 fev. 2015.

TODESCO, J. L. **Reconhecimento de padrões usando rede neuronal artificial com uma função de base radial**: uma aplicação na classificação de cromossomos humanos. 1995. Tese. (Doutorado em Engenharia de Produção) - Universidade Federal de Santa Catarina, Florianópolis, 1995.

VASUDEVA, N.; PARASHAR, H.J.; VIJENDRA, S. Offline character recognition system using artificial neural network. **International Journal of Machine Learning and Computing**, v. 2, n. 4, aug., 2012. Disponível em: <<http://search.proquest.com/openview/c22d4dce2184dc16502f414c8acd8cf5/1?pq-origsite=gscholar>>. Acesso em: 12 fev. 2015.

VIKRAM, C.; BINDU, C.S; SASIKALA, C. Handwritten character recognition for telugu scripts using Multi Layer Perceptrons (MLP). **International Journal of Advanced Research in Computer Engineering and Technology (IJARCET)**, v. 2, n.12, dec., 2013. Disponível em: <<http://ijarcet.org/wp-content/uploads/IJARCET-VOL-2-ISSUE-12-3105-3110.pdf>>. Acesso em: 12 fev. 2015.

## **APÊNDICE A. USO DE REDES NEURAIS PERCEPTRONS MULTICAMADAS (MLP) EM SISTEMA DE RECONHECIMENTO ÓPTICO DE CARACTERES**

Marlito Pereira de Andrade, B.Sc ( UCAM- Campos – RJ / Unes - ES) – [marlitopa@gmail.com](mailto:marlitopa@gmail.com)  
Italo de Oliveira Matias, D.Sc ( UCAM- Campos – RJ) - [italo@ucam-campos.br](mailto:italo@ucam-campos.br)  
Ubiratan Roberte Cardoso Passos B.Sc ( UCAM- Campos – RJ / Unes - ES) – [upassos@gmail.com](mailto:upassos@gmail.com)

**RESUMO:** O objetivo central deste projeto é demonstrar o uso das Redes Neurais Artificiais para reconhecer conjuntos de símbolos de caracteres ópticos. A dificuldade clássica de reconhecer corretamente símbolos ópticos é a complexa irregularidade entre representações gráficas do mesmo caracter devido as variações em fontes, estilos e tamanhos. Esta irregularidade, sem dúvida, aumenta quando se lida com caracteres manuscritos. Uma técnica emergente nesta área de aplicação particular é o uso de Redes Neurais Artificiais. Tais redes podem ser alimentadas com dados a partir da análise gráfica de imagem de entrada e treinadas para caracteres de saída em uma ou mais formas. Para reconhecimento óptico dos caracteres empregou-se Redes Neurais Artificiais Perceptrons Multicamadas (MLP), cuja capacidade de generalização e adaptabilidade são características importantes para o problema em questão. A rede foi treinada e testada no software desenvolvido para o trabalho na linguagem de programação C#, com um número de tipo de fonte amplamente utilizado no alfabeto latim. Os resultados obtidos foram satisfatórios, com taxa de acerto superior a 95%.

**PALAVRAS-CHAVES:** Reconhecimento óptico de caracteres. OCR (Optical Character Recognition). Redes Neurais Artificiais (RNA). Perceptrons Multicamadas (MLP).

### **1: INTRODUÇÃO**

Segundo Osório (1991), os sistemas de reconhecimento óptico de caracteres – OCR (Optical Character Recognition) são sistemas desenvolvidos de certa forma para reproduzir a capacidade humana de ler textos. Além de servirem como uma

forma alternativa de interação homem-máquina, os sistemas de reconhecimento de caracteres tem sido utilizados para a compactação de imagens textuais. Este caso se aplica muito na utilização de sistemas de fac-símile para transmissão de texto. Através do reconhecimento dos caracteres em uma imagem a ser transmitida por este tipo de equipamento, pode-se obter a compactação das informações, resultando em economia (menor número de informações transmitidas).

Segundo Rodrigues e Thomé (2000), a técnica OCR “estabeleceu as bases e a motivação para tornar o reconhecimento de padrões e a análise de imagens, campos individuais de interesse da ciência”.

A técnica de reconhecimento de caracteres se baseia na extração das características de caracteres de uma imagem de documento digitalizado para que se possa fazer a classificação destas características obtidas. Para Aires (2005), “um fator determinante para um bom desempenho do reconhecimento é a seleção do conjunto de características a serem extraídas dos caracteres”.

A dificuldade clássica em reconhecer corretamente símbolos ópticos é a complexa irregularidade entre representações gráficas do mesmo carácter, devido as variações em fontes, estilos e tamanhos. Esta irregularidade, sem dúvida, aumenta quando se lida com caracteres manuscritos. Por isso, os métodos convencionais de programação de mapeamento símbolos da imagem em matrizes, analisa pixels e/ou dados de vetor tentando decidir qual símbolo corresponde a qual carácter daria pouco ou nenhum resultado realista.

Uma técnica emergente nesta área de aplicação particular é o uso de redes neurais artificiais, redes empregam guias específicos (regras de aprendizagem) para atualizar as ligações (pesos) entre seus nós. Tais redes podem ser alimentadas com dados a partir da análise gráfica da imagem de entrada e treinadas para caracteres de saída em várias formas. Mais especificamente alguns modelos de rede utilizam um conjunto de saídas desejadas para comparar com a saída, calcula o erro e faz uso de ajustar em seus pesos. Tais regras de aprendizagem são denominadas como aprendizado supervisionado.

Uma rede, com a regra de aprendizado supervisionado é o modelo Perceptrons Multicamadas(MLP). O modelo de Rede Neural Perceptrons Multicamadas(MLP) utiliza aprendizado supervisionado, no qual usa regra de aprendizagem delta para ajustar seus pesos e podem ser treinadas para um conjunto de entrada / saída em valores desejados de uma série de iterações. A

própria natureza deste modelo em particular é que vai forçar a saída de um dos valores próximos, se uma variação de entrada é alimentada à rede que não está treinada, resolvendo assim questão de proximidade. Ambos os conceitos serão discutidos na parte de revisão de literatura.

Este trabalho utiliza a rede MLP e os resultados foram obtidos para uma serie de tipos de fontes utilizadas. A abordagem técnica seguida no processamento das imagens de entrada detectando símbolos gráficos, análise e mapeamento dos símbolos e treinamento da rede para um conjunto desejado de caracteres Unicode correspondentes às imagens de entrada são discutidos nas seções subsequentes.

## **2: REDES NEURAIIS ARTIFICIAIS**

Segundo Leoesch e Sari (1996) as redes neurais deixaram de ser um sonho para se tornarem realidade na computação. Elas possuem a característica de serem apropriadas ao reconhecimento de padrões, graças a sua capacidade de sofrer aprendizado, modificando seu comportamento frente a um conjunto de estímulos de entrada. Portanto, a rede pode aprender a dar resposta específica para um determinado conjunto de estímulos fornecidos. Devido a suas características inerentes, as redes neurais são muito adequadas para reconhecimento e classificação de padrões, podem se adaptar para responder a um padrão específico.

De acordo com Osório (1991), as redes neurais ou, mais precisamente redes neurais artificiais, representam uma ferramenta computacional de aplicação nas mais diversas áreas da ciência e da vida humana. Dentre as aplicações mais comuns que encontramos para as redes neurais, destacam-se:

- Análise de imagens;
- Classificação de dados;
- Diagnóstico médico;
- Identificação de fraude de cartão de crédito;
- Processamento de sinais e eliminação de ruídos;
- Reconhecimento e síntese de fala;
- Reconhecimento óptico de caracteres (OCR);
- Robótica;

As RNA são modelos matemáticos que se assemelham às estruturas neurais biológicas e que têm capacidade computacional adquirida por meio de aprendizagem e generalização (BRAGA, 2000). Esses modelos almejam semelhança com o sistema nervoso dos seres vivos com sua capacidade de processar informações. Trata-se de uma metáfora da maneira como o cérebro humano processa as informações utilizadas em computação. A rede neural, vista como uma máquina adaptativa é definida por Haykin (2001) como:

Um processador maciçamente paralelamente distribuído constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para uso. Ela se assemelha ao cérebro em dois aspectos: 1. O conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem; 2. Forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.

Para Zurada (1992), as redes neurais são sistemas celulares físicos com a capacidade de adquirir, armazenar e utilizar conhecimento experimental. Esse conhecimento está embutido na rede sob a forma de estados estáveis, que podem ser lembrados, em resposta à apresentação de estímulos. A RNA tem capacidade adquirida por meio de aprendizado e generalização. O aprendizado se dá por meio de exemplos e a generalização é a capacidade de dar respostas coerentes para dados não apresentados na etapa de aprendizado. As RNA's são amplamente empregadas em tarefas de aproximações de funções, previsão de séries temporais, classificação e reconhecimento de padrões.

A arquitetura da rede é definida pela forma na qual esses neurônios estão organizados e interconectados, ou seja, o número de camadas, o número de neurônios por camada, tipos de conexão entre os neurônios e a topologia da rede Haykin (2001) e Silva (2009). Existem diversos modelos para implementação de uma estrutura de rede neural artificial, como a SOM (Selforganizing map), RBF (Radius Basis Function), LMS (Least Mean Square), MPL (Multi Layer Perceptron).

## 2.1: PERCEPTRONS MULTICAMADAS (MLP)

Para capturar a essência dos sistemas biológicos neurais, um neurônio artificial é definido como:

- Ele recebe um número de entradas (que a partir de dados originais, ou a partir da saída de outros neurônios na rede neural). Cada entrada vem através de uma ligação que tem uma força (ou peso); estes pesos correspondem a eficácia sináptica num neurônio biológico. Cada neurônio tem também um valor de limiar individual. A soma ponderada das entradas é formada e subtraída do limiar, para compor a ativação do neurônio (também conhecido como o potencial pós-sináptico, ou PSP, do neurônio);
- O sinal de ativação é transmitido através de uma função de ativação (também conhecido como uma função de transferência), para produzir a saída do neurônio.

Se a função de ativação degrau é utilizada (isto é, a saída do neurônio é 0, se a entrada for menor do que zero e 1 se a entrada é maior ou igual a 0), então o neurônio age como o neurônio biológico descrito anteriormente (subtraindo-se o limiar a partir da soma ponderada e comparando com zero é equivalente a comparar a soma ponderada para o limiar). Na verdade, a função degrau é raramente utilizada nas redes neuronais artificiais, como será discutido. Note-se também que os pesos podem ser negativos, o que implica que a sinapse tem um efeito inibitório e não excitatório sobre o neurônio: neurônios inibidores são encontrados no cérebro.

Isso descreve um neurônio individual. A próxima pergunta é: como devem ser conectados neurônios juntos? Se uma rede estiver sobre qualquer uso, deve haver entradas (que transportam os valores das variáveis de interesse no mundo exterior) e saídas (como forma previsões ou sinais de controle). Entradas e saídas correspondem aos nervos sensoriais e motores, tais como os provenientes dos olhos e que conduz para as mãos. No entanto, podem também ser neurônios ocultos que desempenham um papel na rede interna. Os neurônios de entrada, ocultos e de saída deve estar ligados entre si.

Uma rede típica feedforward tem neurônios dispostos em uma topologia de camadas distintas. A camada de entrada não é realmente neural a todas: estas unidades simplesmente servem para introduzir os valores das variáveis de entrada. Os neurônios da camada oculta e de saída são ligados cada um a todas as unidades

da camada anterior.

As redes neurais Perceptrons Multicamadas é talvez a arquitetura de rede mais popular em uso hoje. Uma soma ponderada parcial de cada uma de suas unidades entradas é realizada e passa este nível de ativação através de uma função de ativação para produzir a sua saída, e as unidades são dispostas em uma topologia em camadas feedforward.

## 2.2: SÍMBOLOS ÓPTICOS DA LINGUAGEM

Vários idiomas são caracterizados por terem suas próprias representações simbólicas escritas (caracteres). Estes caracteres ou são um representante de um sotaque *audioglyph* palavras específicas ou conjunto de palavras em alguns casos. Em termos de estrutura, caracteres de idiomas do mundo manifestam vários níveis de organização. Com respeito a esta estrutura não é sempre um problema de compromisso entre a facilidade de construção e de conservação do espaço. Alfabetos altamente estruturados, como o conjunto de Latim permite fácil construção de elementos de linguagem, forçando a utilização de espaço adicional, alfabetos de estrutura média como o Etíope (Ge'ez) conserva o espaço devido a representação de conjunto grifos de áudio e os tons de um símbolo, mas dita a necessidade de ter conjuntos estendidos de símbolos e, assim, um nível de dificuldade de utilização e de aprendizagem. Alguns alfabetos, nomeadamente os alfabetos orientais, apresentam uma quantidade muito baixa de estruturação e palavras inteiras são delegadas por símbolos individuais. Tais linguagens são compostas de vários milhares de símbolos e são conhecidos por necessitar de um ciclo de aprendizagem abrangendo vidas inteiras.

## 3: VISÃO TÉCNICA GERAL DO PROJETO

As operações da rede implementadas no projeto podem ser resumidas pelos seguintes passos:

### 3.1: FASE DE TREINAMENTO

- Analisar a imagem de caracteres;
- Converter símbolos para matrizes de pixels;
- Recuperar caracteres correspondente saída desejada e converter para Unicode;
- Linearizar a matriz e alimentar a rede;
- Calcular a saída;
- Comparar a saída com o valor desejado de saída Unicode e calcular erro;
- Ajustar os pesos de acordo e repetir o processo até o número predefinido de iterações.

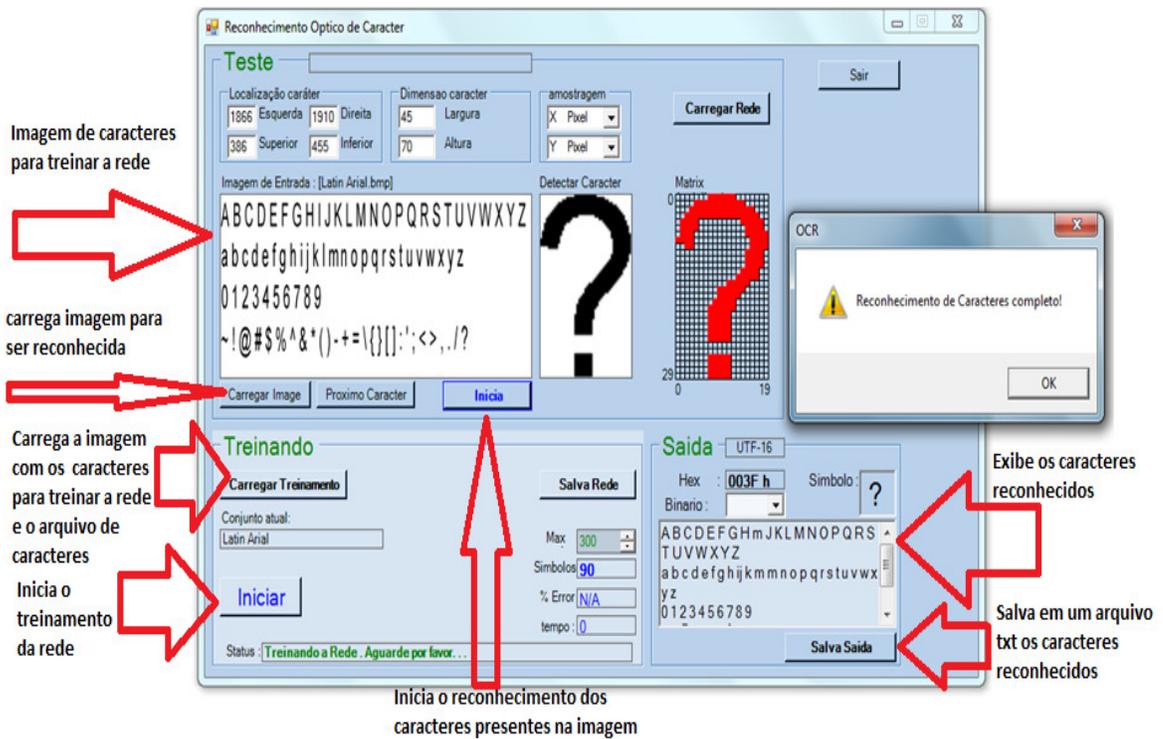
### 3.2: FASE DE TESTES

- Analisar a imagem de caracteres;
- Converter símbolos para matrizes de pixels;
- Calcular a saída;
- Mostrar representação de caracteres Unicode de saída.

### 3.3: COMPONENTES ESSENCIAIS DA IMPLEMENTAÇÃO SÃO:

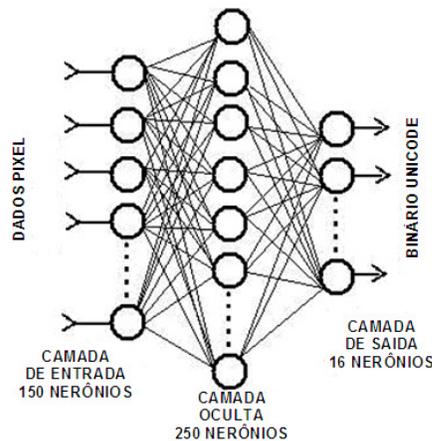
- Topologia da rede e rotina de inicialização de peso;
- Análise de pixel das imagens para detecção de símbolo;
- Rotinas de carregamento das imagens para treinamento de entrada e de saída correspondentes aos caracteres desejados em arquivos especiais chamados conjuntos de caracteres de treinamento (\*. Cts);
- Rotinas de carregamento e salvamento para redes treinadas (valores de peso);
- Rotinas de conversão de caracteres binária para Unicode e vice-versa ;
- Rotinas para cálculo de erro de saída e peso.

#### **3.3.1: Tela principal do software desenvolvido para os testes:**



**Figura 1:** Tela principal do software desenvolvido para os teste do trabalho  
Fonte: Ealborado pelo Autor (2015)

### 3.4: TOPOLOGIA DA REDE



**Figura 2:** Rede MLP do Projeto  
Fonte: Ealborado pelo Autor (2015)

### 3.5: DESCRIÇÃO DA TOPOLOGIA DA REDE UTILIZADA NO PROJETO

A rede MLP implementada para o propósito deste projeto é composta de três camadas, uma de entrada, uma oculta e uma de saída.

A camada de entrada constitui de 150 neurônios que recebem os dados binários de pixel de uma matriz de pixel 10x15 símbolo. O tamanho desta matriz foi decidida tendo em consideração a altura média e a largura da imagem de carácter que pode ser mapeado sem a introdução de qualquer ruído de pixel significativa.

A camada oculta constitui de 250 neurônios cujo número é decidido com base nos melhores resultados nas tentativas e com base nos erros. A camada de saída é constituída por 16 neurônios correspondentes aos 16-bits da codificação Unicode.

Para inicializar os pesos, uma função aleatória foi usada para atribuir um número aleatório inicial que está entre dois inteiros pré-nomeados  $\pm$  peso\_bias. O peso bias é seleccionado a partir de tentativa e erro observando o correspondente peso médio para uma convergência rápida.

#### **4: DETECÇÃO DE SÍMBOLO NA IMAGEM**

O processo de digitalização, segundo Osório (1991), consiste em realizar a aquisição de uma imagem e enviá-la ao computador em um formato adequado para que esse possa manipulá-la. As informações visuais são convertidas em sinais eléctricos, que são convertidos em valores binários a serem armazenados na memória do computador. Neste processo, os sinais são amostrados espacialmente e quantificado em amplitude, de forma a obter a imagem digital.

A segmentação é um passo extremamente necessário e muito crítico em sistemas OCR. Qualquer erro nessa etapa não poderá ser corrigido no passo seguinte, o reconhecimento. Em outras palavras, erros afetam diretamente o reconhecimento (CASEY, 1983). Segmentação é um processo que determina os constituintes de uma imagem. É necessário localizar as regiões dos documentos onde os dados foram impressos e distingui-los de figura e gráficos (EIKVIL, 1993).

No processamento de imagens, a imagem digital é definida com sendo uma matriz formada por  $m \times n$  elementos. Cada elemento dessa imagem é denominado pixel (picture element), que possui associado, uma informação referente a cor e luminosidade.

O processo de análise de imagem para detectar símbolos de caracteres pelo

exame dos pixels é a parte central do conjunto de preparação tanto na entrada do treinamento e fase de testes. Extensões simbólicas são reconhecidas a partir de um arquivo de imagem de entrada com base no valor de cor de pixels individuais, o que para os limites desse projeto é assumido para ser preto **RGB (0, 0, 0)** ou branco **RGB (255, 255, 255)**. As imagens de entrada são assumidas como sendo em forma de bitmap de qualquer resolução, que pode ser mapeada para um objeto bitmap interno no ambiente Microsoft Visual Studio. O procedimento também assume que a imagem de entrada é composta apenas de caracteres e qualquer outro tipo de objeto delimitador, como uma linha de fronteira não é levado em consideração.

#### 4.1: PROCEDIMENTO PARA A ANÁLISE DE IMAGENS PARA DETECÇÃO DE CARACTERES

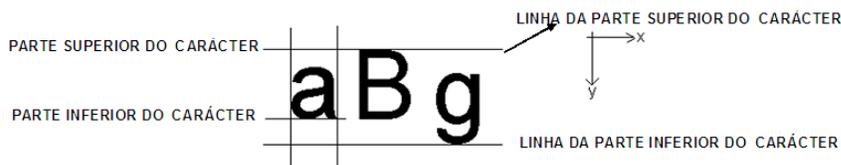
O procedimento para a análise de imagens para detecção de caracteres está listado a seguir:

##### 4.1.1: Determinar Linhas de Caractere

Enumeração das linhas de caracteres em uma imagem de caracter é essencial para a delimitação dos limites em que a detecção pode prosseguir. Assim detectar o próximo caractere em uma imagem não necessariamente envolve a digitalização da imagem inteira novamente;

##### 4.1.2: Detecção de Símbolos Individuais

Detecção de símbolos individuais envolve a digitalização das linhas de caracteres ortogonalmente para separar pixels pretos compostos na imagem.



**Figura 3:** Linha e detecção dos limites dos caracteres  
Fonte: Elaborado pelo Autor (2015).

A partir do processo seguido e a figura acima, é evidente que o caractere detectado vinculado pode não ser o real destino ao caractere em questão. Esta é uma questão que surge com a altura e irregularidade que existe no alinhamento na parte inferior com símbolos alfabéticos impressos. Assim, uma linha superior não significa, necessariamente, topo de todos os caracteres e uma linha inferior pode não significar inferior de todos os caracteres também. Assim, uma confirmação da parte superior e inferior para o caractere é necessário. Um algoritmo de confirmação opcional é implementado no projeto:

- A. Inicia no topo da linha atual e à esquerda do caractere;
- B. Digitalizar até a direita do caractere:
  - Se um pixel preto é detectado registra y como topo confirmado;
  - Se não continuar para o próximo pixel;
  - Se nenhum pixel preto for encontrado incrementar y e reinicia x para verificar a próxima linha horizontal.



**Figura 4:** Confirmação dos limites dos caracteres.  
Fonte: Elaborado pelo Autor (2015).

#### 4.2: MAPEAMENTO DA IMAGEM EM SÍMBOLO NA MATRIZ.

O próximo passo é mapear a imagem em símbolo correspondente na matriz binária bidimensional. Uma questão importante a considerar aqui será decidir a dimensão da matriz. Se todos os pixels do símbolo são mapeados para a matriz, com certeza gostaria de poder adquirir todas as características distinguindo pixel do símbolo e minimizar a sobreposição com outros símbolos. No entanto, esta estratégia implicaria a manutenção e o processamento de uma matriz muito grande (até 1500 elementos para um 100x150 pixel da imagem). Por isso, uma

compensação razoável é necessária a fim de minimizar o tempo de processamento, que não vai afetar significativamente a separabilidade dos padrões. O projeto empregou uma estratégia de amostragem que mapeia o símbolo da imagem em uma matriz de símbolos binária 10x15 com apenas 150 elementos. Uma vez que a altura e largura das imagens individuais variam, um algoritmo adaptativo de amostragem foi implementado. O algoritmo é listado abaixo:

#### **4.2.1: Algoritmo:**

4.2.1.1: Para a largura (inicialmente 20 elementos de largura):

I: Mapeia primeiro (0, y) e última (largura, y) pixel dos elementos diretamente para o primeiro (0, y) e último (20, y) elementos da matriz;

II: Mapear o elemento de pixel intermédia (largura / 2, y) para o elemento de matriz 10;

III: Subdividir outras divisões e mapear de acordo com a matriz;

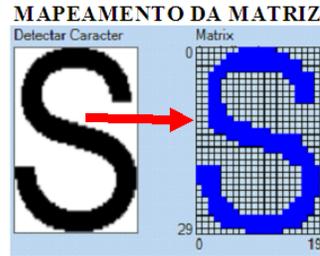
4.2.1.2: Para a altura (inicialmente 30 elementos altura):

I: Mapear o primeiro x, (0) e anterior (x, altura) pixel dos elementos diretamente para a primeira (x, 0) e anterior (x, 30) elementos da matriz ;

II: Mapear o elemento de pixel central (x, altura / 2) para o elemento de matriz ;

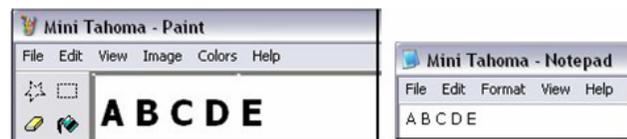
III: Subdividir outras divisões e mapear de acordo com a matriz;

4.2.1.3: Reduzir ainda mais a matriz de 10x15 por amostragem por um fator de 2 em ambas, a largura e a altura:



**Figura 5:** Mapeamento de símbolo da imagem em uma matriz binária  
Fonte: Elaborado pelo Autor (2015).

A fim de ser capaz de alimentar os dados da matriz da rede (que é de uma única dimensão) da matriz devemos primeiro linearizar com uma única dimensão. Uma amostra do conjunto do mini treinamento é mostrada abaixo:



**Figura 6:** Imagem de entrada e arquivo texto de saída desejada de exemplo treinamento Mini-Tahoma  
Fonte: Elaborado pelo Autor (2015).

#### 4.3: TREINAMENTO

- Quando a rede for inicializada e dados de entrada de treinamento preparado, a rede está pronta para ser treinada. Algumas questões que precisam ser abordadas no treinamento da rede são:
  - Como são caóticos os dados de entrada? Uma entrada caótica varia de forma aleatória e no intervalo extremo sem qualquer fluxo previsível entre os seus membros;
  - Como são complexos os padrões para treinarmos a rede? Padrões complexos são geralmente caracterizados por sobreposição e elevado tamanho de dados;
  - O que deve ser usado para os valores de:

- a. Taxa de aprendizagem;
  - b. Inclinação sigmoide;
  - c. Peso bias.
- Quantas iterações (épocas) são necessárias para treinar a rede para um determinado número de conjunto de entrada ?
  - Qual valor limite do erro deve ser utilizado para comparar com ordem de parada prematura das iterações caso seja necessário?

Alfabéticos de símbolos ópticos são um dos conjuntos de entrada mais caóticos em estudos de reconhecimentos de padrões. Isto é devido à natureza imprevisível da sua representação gráfica, que foi visto a partir da sequência de sua ordem. Por exemplo, o caractere alfabético latino consecutivo 'A' e 'B' tem pouca similaridade na característica quando representados simbolicamente em sua forma gráfica. A figura abaixo demonstra a sequência de ponto caótico e não-caótico com o latim e alguns conjuntos de caracteres factícios:

**A B C    Γ Γ Γ**

**Figura 7:** Exemplo de seqüências de símbolos caóticos e não-caóticos.  
Fonte: Elaborado pelo Autor (2015).

A complexidade do padrão individual dos dados também é outro problema no reconhecimento de caracteres. Cada símbolo tem um grande número de características distintas, que precisam ser contadas na ordem correta para reconhecê-lo. Eliminação de algumas características pode resultar na sobreposição de padrões e a quantidade mínima de dados necessários a torna uma das classes mais complexas do espaço de entrada no reconhecimento dos padrões.

Diferente das questões conhecidas mencionadas, os outros parâmetros numéricos são determinados em tempo real. Eles também variam muito de uma implementação a número de símbolos de entrada de alimentação e a topologia da rede. Para o propósito deste projeto os parâmetros utilizados são:

- Taxa de aprendizagem = 150;

- Inclinação sigmoide = 0,014 ;
- Peso de Bias = 30 (determinado por tentativa e erro);
- Número de Épocas = 300-600 (dependendo da complexidade dos tipos de fonte) ;
- A média limiar de erro = 0,0002 (determinado por tentativa e erro).

#### 4.4: TESTE

A etapa de implementação dos testes é simples e direta. Os passos básicos nas imagens de caracteres de entrada para testes podem ser resumidos como se segue:

##### 4.4.1: Algoritmo:

- Carga do arquivo de imagem;
- Analisar a imagem para linhas de caracter;
- Para cada linha de caracter detectar símbolos de caracteres consecutivos:
  - a. analisar e processar a imagem de símbolo mapeando para um vetor de entrada;
  - b. alimentar vetor de entrada para a rede e calcular a saída ;
  - c. converter a saída binária Unicode para o caracter correspondente renderizar em uma caixa de texto.

## 5: RESULTADOS E DISCUSSÃO

A rede foi treinada e testada por um número de tipo de fonte amplamente utilizado no alfabeto latim. O código do programa é escalável, a inclusão do maior número de fontes a partir de qualquer alfabeto do tipo linguagem é direto.

Os passos necessários são: preparar a sequência de símbolos das imagens de entrada em um único arquivo de imagem (\*.bmp [bitmap] extensão), digitar os

caracteres correspondentes em um arquivo de texto extensão (\*.cts [conjunto de caracteres de treinamento]) e salvar os dois na mesma pasta (ambos devem ter o mesmo nome de arquivo exceto as suas extensões). O aplicativo vai oferecer um diálogo de abertura de arquivo para que o usuário localize o arquivo de texto \*.cts e carregara automaticamente o arquivo de imagem correspondente.

Embora os resultados listados nas tabelas seguintes sejam a partir de um processo de treino/teste das imagens criadas com 72pt. A fonte pode ser de outro tamanho que também é direto para preparar a entrada/saída do conjunto de caracteres desejado como explicado.

Observação: Devido ao valor aleatório na inicialização dos pesos, os valores listados representam apenas o desempenho típico da rede e reprodução exata não poderá ser obtida com outros testes.

**Tabela 1:** Resultados para a variação no número de épocas

Tipo da fonte	300		600		800	
	Nº de caracteres errados	% Erro	Nº de caracteres errados	% Erro	Nº de caracteres errados	% Erro
Latim Arial	4	4,44	3	3,33	1	1,11
Latim Tahoma	1	1,11	0	0	0	0
Latim Times	0	0	0	0	1	1,11
Roman						

Número de caracteres=90, Taxa de aprendizagem=150, Inclinação sigmoide = 0,014

Fonte: Elaborado pelo Autor (2015).

**Tabela 2:** Resultados para a variação no número de caracteres de entrada

Tipo da fonte	20		50		90	
	Nº de caracteres errados	% Erro	Nº de caracteres errados	% Erro	Nº de caracteres errados	% Erro
Latim Arial	0	0	6	12	11	12,22
Latim Tahoma	0	0	3	6	8	8,89
Latim Times	0	0	2	4	9	10
Roman						

Número de caracteres=100, Taxa de aprendizagem=150 Inclinação sigmoide = 0,014

Fonte: Elaborado pelo Autor (2015).

**Tabela 3:** Resultados para a variação no parâmetro taxa de aprendizagem

Tipo da fonte	50		100		120	
	Ne de errados	caracteres %	Ne de errados	caracteres %	Ne de errados	caracteres %
Latim Arial	82	91,11	18	20	3	3,33
Latim Tahoma	56	62,22	11	12,22	1	1,11
Latim Times Roman	77	85,56	15	16,67	0	0

Número de caracteres=90 Número de época=600 Inclinação sigmoide = 0,014

Fonte: Elaborado pelo Autor (2015).

## 5.1: OBSERVAÇÃO DE DESEMPENHO

### 5.1.1: Influência da variação de parâmetros

- Aumentando o número de iterações tem geralmente uma relação de proporcionalidade positiva para o desempenho da rede. No entanto, em certos casos, aumentando ainda mais o número das épocas tem um efeito adverso de introdução de um maior número de reconhecimentos errados. Isto pode ser atribuído em parte ao valor elevado do parâmetro taxa aprendizagem, como a rede se aproxima dos limites ideais e mais peso resultam em alterações ignorando o estado ideal. Com mais iterações a rede vai tentar 'ocilar' de volta para o estado desejado e novamente continuar com uma boa chance de perder o estado ideal na época final. Este fenômeno é conhecido como o excesso de aprendizagem;
- O tamanho dos estados da entrada também é outro fator que influencia o desempenho direto. É natural que o maior número de símbolos de entrada defina que a rede deva ser treinada para ser mais sensíveis a erros. Usualmente o complexo, conjunto de tamanho grande de entrada exigem uma topologia de rede grande, com mais números de iterações. Para obter o número máximo acima de 90 símbolos do conjunto a topologia ótima alcançada foi uma camada oculta de 250 neurônios;
- Variação do parâmetro taxa de aprendizagem também afeta o desempenho da rede para um determinado limite de iterações. Quanto menor o valor deste parâmetro, um valor menor do que a rede atualiza seus pesos. Isso implica intuitivamente que será menos provável enfrentar a dificuldade de aprendizagem

discutido acima, uma vez que será atualizada as suas ligações lentamente e de uma forma mais refinada. Mas infelizmente, isso também implica que um maior número de iterações seja necessária para atingir o seu estado ideal. Assim, uma troca na ordem é necessária para otimizar o desempenho da rede. O valor ideal decidido para o parâmetro de aprendizagem é 150.

## **6: CONCLUSÃO**

Neste trabalho utilizou-se redes neurais Perceptrons Multicamadas (MLP) para reconhecimento óptico de caracteres. Durante o desenvolvimento do projeto, concluiu-se que a fase de segmentação é uma das etapas mais críticas, pois, qualquer problema nessa etapa irá afetar diretamente o desempenho.

Durante o decorrer do trabalho, foram observadas vantagens e desvantagens do uso de redes neurais. A principal vantagem observada foi a característica de auto aprendizado, já que as redes neurais não necessitam de reconhecimento de especialista para tomar decisões, baseando-se unicamente nos exemplos que lhe são fornecidos. Entre as desvantagens observadas da redes neurais, destacam-se:

- Caixa Preta: é impossível saber como a rede chegou a uma determinada conclusão, os pesos utilizados não aceitam uma interpretação lógica, sabe-se apenas que funcionam;
- Trabalho Artesanal: não há regras gerais para se determinar o volume de dados para treinamento, quantas camadas utilizar, a melhor estratégia de treinamento, enfim, como desenvolver a rede. Esses parâmetros só podem ser estabelecidos através de experiências com redes e através de tentativas e erros.

Considera-se que o objetivo do trabalho foi alcançado. No entanto, por se tratar de um protótipo, existem várias melhorias que podem ser implementadas. O protótipo desenvolvido apresenta as seguintes limitações:

- A rede só foi treinada para reconhecer letras em fontes Latin Arial, Latin Tahoma, Latin Times Roman e números de 0 a 9;

- A rede não foi treinada para reconhecer caracteres com acentuação.

## 7: REFERÊNCIAS BIBLIOGRÁFICAS

AIRES, S.B.K. **Reconhecimento de caracteres manuscritos baseados em regiões perceptivas**. 2005. Dissertação (Mestrado em Informática Aplicada) - Pontifícia Universidade Católica do Paraná. Curitiba, 2005.

BRAGA, A.P.; CARVALHO, A.C.P.L.F.; LUDERMIR, T.B. **Redes neurais artificiais: teoria e aplicações**. Rio de Janeiro: LTC, 2000.

CASEY, R.G.; JIH, C. R. A processor-based OCR system. **IBM Journal the Research and Development**. Armonk, v.27, n.4, p389-399, jul. 1983. Disponível em:  
<[http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5390436&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs\\_all.jsp%3Farnumber%3D5390436](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5390436&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D5390436) >. Acesso em: 12 fev. 2015.

EIKVIL, L. **OCR: optical character recognition**. Disponível em:  
<<https://www.nr.no/~eikvil/OCR.pdf> >. 12 mar. 2014.

HAYKIN, S. **Redes neurais : princípios e prática**. 2. ed. Porto Alegre: Bookman, 2001

LEOESCH, C.; SARI, S.T. **Redes neurais artificiais: fundamentos e modelos**. Blumenau: FUB, 1996

OSÓRIO, F.S. **Um estudo sobre o reconhecimento visual de caracteres através de redes neurais**. 1991. Dissertação (Mestrado em Computação) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 1991.

RODRIGUES, R. J.; THOMÉ, A. C. G. Reconhecimento de dígitos cursivos: um método de segmentação por histogramas. In: SIMPÓSIO BRASILEIRO DE REDES NEURAS, 6., 2000, 2-5 abr., Rio de Janeiro. **Anais...** Porto Alegre: Sociedade Brasileira de Computação: 2001. Disponível em: <<http://www.sbc.org.br> >. Acesso em: 12 fev. 2014.

SILVA, A. M. **Utilização de redes neurais artificiais para classificação de spam.** 2009. Dissertação (Mestrado em Modelagem Matemática e Computacional) - Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte, 2009.

ZURADA, J.M. **Introduction to artificial neural systems.** Boston: PWS Publishing Company, 1992.

## **APÊNDICE B: SISTEMA DE SEGMENTAÇÃO DE IMAGEM DE CHAPAS DE GRANITO POLIDAS PARA GERAR UM BANCO DE IMAGENS COM PADRÕES DE DEFEITOS EM GRANITOS.**

**Marlito Pereira de Andrade, B.Sc ( UCAM- Campos – RJ / Unes - ES) – marlitopa@gmail.com**  
**Italo de Oliveira Matias, D.Sc ( UCAM- Campos – RJ) - italo@ucam-campos.br**  
**Nivaldo Antonio da Silva, B.Sc ( UCAM- Campos – RJ) – nivaldosilva13@live.com**  
**Ubiratan Roberte Cardoso Passos B.Sc ( UCAM- Campos – RJ / Unes - ES) –**  
**upassos@gmail.com**

**RESUMO:** O objetivo central deste projeto é demonstrar o uso da biblioteca OpenCV para segmentar conjuntos de defeitos presente em chapas de granito polidas. Mesmo depois da automação de muitos setores industriais, a inspeção de chapas de granito polidas ainda é realizada por humanos com melhor qualidade do que por meios artificiais. Grande parte das abordagens automáticas para esta tarefa concentra-se na análise de algum tipo de imagem digital do espectro visível ou não visível, tais como os raios X, para avaliar chapas de granito polidas. A análise de imagens do espectro visível de particular interesse neste trabalho, pode então ser entendida como a automação do processo de inspeção visual realizado por especialistas humanos utilizando técnicas da visão computacional. As inspeções de chapas de granito polidas são realizadas por humanos que usualmente tomam como base padronizações internos da empresa. Para o caso específico chapas de granito polidas no Brasil, uma das escassas referências é o Centro Tecnológico do Mármore e Granito (Cetemag), que ministra o curso “Inspeção de Chapas”. Foi utilizado imagem de chapas de granito polidas, nessas imagens foi aplicada filtro de suavização (cvSmooth), na sequência foi feito a conversão da cor de RGB para HSV, após feito a conversão da imagem para HSV foi utilizado uma função limiar (threshold) com valores limiares inferiores e superiores de H, S e V definidos. Os resultados obtidos foram satisfatórios atendendo as expectativas esperadas com os valores padrões definidos baseado nas cores das imagens do material analisado. Para a implementação, foi utiliza a linguagem de programação C, juntamente com a biblioteca para visão computacional Open Source, OpenCV.  
**PALAVRAS-CHAVES:** Segmentação. OpenCV(Open Source Computer Vision). RGB(Red, Green, Blue). HSV(Hue, Saturation, Value).

### **1: INTRODUÇÃO**

A segmentação de imagens está presente nos principais programas de computação gráfica. Ela é um processamento crucial para a análise e o entendimento de imagens digitais. Este trabalho procura unir técnicas de segmentação de imagem para auxiliar na inspeção de imagens de chapas de granito polidas identificando de forma semiautomática áreas defeituosas.

Abordagem de segmentação utilizada nesse trabalho é classificada como uma segmentação por cor. Ela utiliza-se das características de cor, saturação e luminosidade presentes nos objetos das imagens de chapas de granito polidas para identificar uma área desejada, informada pelo profissional que analisa a imagem. Para atingir uma boa precisão deve ser informado manualmente o fator de tolerância que a técnica irá trabalhar, podendo este no futuro ser calculado de forma automática para o aperfeiçoamento dos resultados.

A todo momento estamos fazendo reconhecimento de padrões em nossas vidas. Neste exato instante, ao ler este texto, você reconhece estes traços pretos como letras, reconhece um grupo de letras como uma palavra e reconhece o significado de cada palavra. Enfim, reconhecer o rosto de uma pessoa, distinguir um cachorro de um gato, compreender a fala, ler as mais diversas caligrafias e até mesmo interpretar um exame de eletrocardiografia, tudo isso é reconhecimento de padrões (HENRIQUE, 2003).

Mesmo depois da automação de muitos setores industriais, a inspeção de chapas de granito polidas ainda é realizada por humanos com melhor qualidade do que por meios artificiais. Grande parte das abordagens automáticas para esta tarefa concentra-se na análise de algum tipo de imagem digital do espectro visível ou não visível, tais como os raios X, para avaliar chapas de granito polidas. A análise de imagens do espectro visível, de particular interesse neste trabalho, pode então ser entendida como a automação do processo de inspeção visual realizado por especialistas humanos utilizando técnicas da visão computacional.

As inspeções de chapas de granito polidas são realizadas por humanos usualmente tomam como base padronizações internas da empresa. Para o caso específico chapas de granito polidas no Brasil, uma das escassas referências é o Centro Tecnológico do Mármore e Granito (Cetemag), que ministra o curso “Inspeção de Chapas”.

O presente trabalho tem como objetivo desenvolver um sistema para fazer a segmentação de imagens de chapas de granito polidas para gerar um banco de imagem com padrões para uso futuro em uma rede neural para fazer a classificação de chapas de granito polidas baseada em informações visuais, com ênfase no tratamento do padrão de cor para reconhecer se imagem da chapa fornecida possui ou não algum defeito.

A proposta é trabalhar com um sistema que atenda aos requisitos citados, com custo reduzido e que tenha as seguintes características:

- Obtenção de um sistema de classificação de chapas de granito polidas que permita o reconhecimento a partir de uma imagem de boa qualidade;
- A imagem capturada pela câmera é enquadrada no software de análise de imagens, e quando solicitado a imagem é analisada apresentando como resultado a determinação dos valores de cor (RGB/HSV);
- Os valores médios obtidos serão utilizados como entradas no software de rede neural que oferecerá na saída a classificação do defeito na chapa caso exista;

Reconhecimento de padrões (RP) é, por natureza, uma ciência inexata, e assim admite muitas abordagens, algumas vezes complementares, outras competitivas, para aproximar, indicar ou encontrar soluções a um dado problema. Em problemas complexos de RP, a abordagem comum utilizada é dividir o problema em dois módulos subsequentes: a extração de características e o módulo de classificação. Tanto para o primeiro como para o segundo módulo já existem várias técnicas desenvolvidas que vão desde técnicas estatísticas e matemáticas até técnicas de Inteligência Artificial (ARTUR, 1999; BISHOP, 1995).

## **2: PROCESSAMENTO DIGITAL DE IMAGENS**

Uma imagem poder ser definida como uma função bidimensional  $f(x,y)$ , onde  $x$  e  $y$  são coordenadas espaciais e a amplitude de  $f$  em um par de coordenadas  $(x,y)$

é chamada de intensidade ou nível de cinza de uma imagem naquele ponto (GONZALEZ , WOODS, 2009). Quando os valores de  $x,y$ , e a amplitude de  $f$  forem definidos, discreto, chama-se esta imagem de uma imagem digital.

Em diversos campos de pesquisa e desenvolvimento, há inúmeros problemas que poderiam ser resolvidos com métodos de processamento de imagens. A partir de uma imagem digital, podem ser feitas diversas interpretações e processamento na imagem até encontrar a solução para estes problemas.

Existem varias técnicas para processamento de imagens. Dentre elas podemos classificar as técnicas de realce, segmentação, compressão, análise de textura, morfologia matemática e outras (PEDRINI, SCHWARTZ, 2008). Juntas essas técnicas podem ser utilizadas para inúmeras aplicações como processamento de imagem de satélite, processamento de imagem medicas, detecção de face e objetos, visão computacional para robôs em automação industrial e muitas outras. A figura 2 mostra um exemplo de limiarização usado no projeto para segmentar a imagem.

## 2.1: SEGMENTAÇÃO DA IMAGEM

O processo de digitalização, segundo Osório (1991), consiste em realizar a aquisição de uma imagem e enviá-la ao computador em um formato adequado para que esse possa manipulá-la. As informações visuais são convertidas em sinais elétricos, que são convertidos em valores binários a serem armazenados na memória do computador. Neste processo, os sinais são amostrados espacialmente e quantificado em amplitude, de forma a obter a imagem digital.

Segmentação de imagens é o termo usado em visão computacional para o agrupamento de partes de uma imagem genérica em unidades que são homogêneas com respeito a uma ou várias características (ou atributos), resultando em uma imagem segmentada (BALLARD; BROWN, 1982).

A segmentação por regiões é utilizada para separar os objetos de interesse. Neste caso, a imagem é particionada em diferentes regiões, ficando cada uma relacionada a certas características e conjuntos de pixels conectados. Assim, a partir do particionamento da imagem, em regiões, podem ser realizadas medidas sobre cada região e as relações entre as regiões adjacentes podem ser estabelecidas

(GONZALEZ; WOODS, 2009).

No âmbito de reconhecimento de padrões a segmentação de imagens é um passo preliminar e essencial para passos subsequentes de análise de imagem. Neste caso, o reconhecimento pode ser realizado a partir do relacionamento espacial entre os objetos de interesse presentes na cena.

No processamento de imagens, a imagem digital é definida com sendo uma matriz formada por  $m \times n$  elementos. Cada elemento dessa imagem é denominado pixel (picture element), que possui associado, uma informação referente a cor e luminosidade.

O processo de análise de imagem para detectar defeitos pelo exame dos pixels é a parte central do conjunto de preparação dos padrões que serão usados futuramente num algoritmo de rede neural para reconhecer defeitos presente em chapas de granito polidas. Defeitos serão reconhecidas a partir de um arquivo de imagem de entrada com base no valor de cor de pixels individuais, o que para os limites desse projeto é assumido para ser **RGB**. As imagens de entrada são assumidas como sendo em forma de bitmap de qualquer resolução, que pode ser mapeada para um objeto bitmap interno no ambiente Microsoft Visual Studio juntamente com a biblioteca OpenCV.

## 2.2: VISÃO COMPUTACIONAL.

Na visão computacional pode ser percebido um aumento significativo nos vários campos a serem desenvolvidos, o desafio é desenvolver sistemas autônomos que reproduzam as capacidades do sistema visual humano e que sejam capazes de reagir a estímulos visuais de forma adequada à área específica sob investigação. A compreensão do funcionamento do sistema visual dos seres humanos, sua capacidade de aprendizagem e sua habilidade em realizar inferências e ações baseadas em estímulos visuais é de grande importância para o desenvolvimento e aprimoramento de técnicas para análise automática ou semi-automática de informações extraídas de uma imagem ou cena com o auxílio de computador.

Através do trabalho com imagens é possível desenvolver e obter muitos dados, que usados de forma correta, podem trazer benefícios dentro de vários contextos.

No início da década de 60 surgiu o termo reconhecimento de padrões. No principio significava a detecção de formas simples. Segundo Hugo (1995), Duda e Hart (1973) deram uma definição interessante para o termo, "Reconhecimento de padrões é um campo interessado no reconhecimento por máquinas de regularidades significativas em ambientes ruidosos ou complexos", ou a procura por uma estrutura nos dados.

Na visão computacional, juntamente com redes neurais, tem-se estudos sobre reconhecimento de padrões. O que para os seres humanos é uma habilidade simples e extremamente desenvolvida por se tratar de reconhecer objetos do dia a dia, como rostos, fisionomias, caligrafias e até mesmo estado de humor das pessoas que conhecem no ambiente de computadores, já se torna um pouco mais complexo, por se tratar de algoritmos e técnicas que permitam que um computador aprenda, ou seja, que ele melhore o seu desempenho em determinada tarefa.

Alguns padrões podem ser identificados como bem estruturados ou, ao menos, estruturados adequadamente. Outros dados podem ser de difícil identificação/classificação. Por este motivo o reconhecimento de padrões é, muitas vezes, chamado de "ciência não exata".

É possível interpretar padrões como sendo meios pelos quais o mundo é interpretado e, a partir desta interpretação, elaboradas atitudes e tomadas decisões.

Para classificar imagens e obter um padrão de reconhecimento, deve-se saber antes para qual finalidade se quer obter informações. Segundo Mossmann (2010), algumas áreas se destacam na aplicação do reconhecimento de padrões, como: identificação através de impressões digitais (PEDRINI; SCHWARTZ, 2008) e análise da íris, diagnósticos médicos (STEINER, 1995), análise de imagens aeroespaciais, visão computacional (PERELMUTER al., 1995), diagnósticos pré e pós-natal e certos diagnósticos de câncer (AGUIAR, 2000), investigação da qualidade do papel industrial (STEINER, 1995), processamento de imagens (ALBUQUERQUE, 2000), análise de peças para manutenção preventiva, análise de caracteres manuscritos (PRADO, 1975), análise de eletrocardiogramas (MASCARENHAS, 1987), reconhecimento e identificação de cromossomos (TODESCO, 1995), dentre outras.

Entretanto, as dificuldades de extrair estas informações, classificar a imagem e definir um padrão, estão na definição de quais técnicas e parâmetros serão utilizados, já que, para cada objetivo de uso final da imagem tem-se grande

variedade de técnicas existentes, podendo os resultados serem totalmente diferentes dependendo da técnica que é aplicada.

No contexto onde tem-se a imagem de uma chapa de granito polida obtida através de uma câmera fotográfica, o objetivo é segmentar essa imagem identificar se essa imagem possui ou não defeitos, caso possua classificar esse defeito. Principais defeitos: as rochas ornamentais e de revestimento sofrem solicitações naturais e artificiais, que provocam desgaste, perda de resistência mecânica, fissuração, manchamento, formação de crostas (eflorescência de sais) e mudança de coloração. As solicitações naturais estão relacionadas ao intemperismo geológico, deformação (tectônica e atectônica) e erosão. Já as solicitações artificiais estão ligadas á lavra, beneficiamento, manuseio e uso/aplicações (ABIROCHAS, 2001).

Para esta identificação é necessário aplicar várias técnicas na imagem, como se fosse um roteiro até chegar ao resultado final. Primeiro deve-se eliminar os ruídos da imagem, para isto é possível usar algoritmos como: filtro da média, filtro de Gauss, entre outros. Em seguida, é importante detectar as bordas da imagem, podendo então usar filtros como, Detecção de bordas de Sobel, Roberts, Marr, Hildreth, etc. Com as bordas definidas e devidamente delimitadas, a chapa deve ser segmentada para ter-se cada defeito separados. Para atingir isto podem ser usados algoritmos como: Watershed, Snacks, dentre outros. Em todo esse processo, uma técnica pode se sobressair a outra, podendo que o resultado final, a leitura da chapa, não seja igual dependendo da ordem e ou das técnicas utilizadas na mesma.

A possibilidade de automatização deste processo de definição da ordem das técnicas e métodos que são aplicadas em cada imagem para obter um objetivo final e uso da mesma é que justifica e motiva a realização deste projeto. Através do desenvolvimento de um sistema para segmentação dessas imagens para futuramente ser utilizadas em um algoritmo de redes neurais acredita-se que é possível obter sucesso para o objetivo final do projeto.

### **3: BIBLIOTECA OPENCV**

O OpenCV é uma biblioteca open source de visão computadorizada, desenvolvida para alto desempenho computacional e para utilização em aplicações

de tempo real. O seu objetivo é disponibilizar uma infraestrutura que ajude as pessoas a construir aplicações bastante sofisticadas em relativamente pouco tempo. Esta biblioteca contém mais de 500 funções que se espalham por muitas áreas possíveis de aplicação de visão computadorizada, entre as quais: inspeção de produtos industriais, imagem médica, segurança, interfaces de utilizador, calibração de câmaras, visão estereoscópica e robótica.

Esta biblioteca existe desde 1999, altura em que foi lançada a sua versão experimental. Desde então tem sido a base para o desenvolvimento de inúmeras aplicações nas áreas anteriormente mencionadas. É uma biblioteca compatível com Windows, Linux e Mac OS X, desenvolvida em C e C++, mas já com suporte a ser desenvolvido também para outras linguagens tais como Java, Python, Ruby, Matlab e outras. Estas características fazem com que o OpenCV se tenha tornado uma biblioteca de referência no desenvolvimento de aplicações de visão computadorizada (BRADSKI, KAEHLER, 2008).

O OpenCV suporta vários formatos de imagem e vídeo assim como a aquisição direta de imagens de uma câmara, sendo ainda capaz de fazer a conversão entre vários formatos. Oferece também uma coleção de funções essenciais para processamento e manipulação de imagem que permitem executar desde os processos mais simples até processos mais complexos. Suporta os algoritmos de extração de contornos, Canny e Sobel, que são normalmente muito utilizados em processos de análise de imagem. Oferece também suporte para as operações morfológicas, erosão e dilatação de píxeis. Além disto contém funções para suporte de tarefas mais complexas, tal como é o caso da aplicação de filtros de Kalman, cálculo dos vectores de optical flow ou cálculo da transformada de Hough.

#### **4: ROCHAS ORNAMENTAIS.**

As rochas ornamentais brasileiras têm se firmado como um elemento promotor de desenvolvimento econômico e social de extrema relevância para muitos estados brasileiros, pois, sem dúvida, gera emprego e renda na região onde são extraídas ou beneficiadas, auxiliando na fixação do homem e sua família no interior. Do ponto de vista de comércio exterior, o setor tem gerado divisas para o Brasil em escala crescente nas exportações, mesmo com flutuações cambiais desfavoráveis

aos exportadores.

Há um crescente consumo desse tipo de material, à medida que aumentam o conhecimento da grande variedade de opções já existentes no mercado (mais de 1.000 tipos diferentes de produtos nacionais), por parte de arquitetos, designers, decoradores, construtores e consumidores.

Além disso, com a evolução tecnológica em todas as etapas do processo produtivo (extração, beneficiamento e logística), surgiram novas possibilidades de aplicação das rochas naturais, tanto na forma de revestimentos externos e internos, como na aplicação isolada, tais como, tampos (mesa, cozinha e banheiros), esculturas, arte funerária e túmulos, pés de mesa, balaustres, etc.

Segundo Chiodi Filho (2007), cerca de 60 a 70% das exportações brasileiras do setor de rochas ornamentais são representadas pelas chapas polidas de granito e materiais naturais similares, com 2 e 3 cm, apesar desse estágio ser considerado intermediário quando nos referimos à evolução dos produtos acabados e da agregação de valor comercial.

#### 4.1: MERCADOS: MUNDIAL – BRASILEIRO – CAPIXABA.

O consumo mundial de rochas ornamentais gira entorno de 600 milhões de metros quadrados por ano, que, comparando-se com a última década, resulta numa taxa média de crescimento de 7% ao ano, muito além da taxa de crescimento médio da economia mundial, conferindo a este segmento um caráter economicamente dinâmico e internacionalizado (SAMPAIO, 2007).

Segundo Chiodi Filho (2001), cerca de 80% da produção mundial é transformada em chapas e ladrilhos para revestimentos, 15% é desdobrada em peças para arte funerária e 5% para outros campos de aplicação. Os mármore extraídos participam com cerca de 45% da produção mundial, os granitos com 40%, os quartzitos e similares com 5% e 5% com as ardósias.

Segundo publicação da Revista Rochas (novembro/dezembro de 2006), cerca de 80% do mercado de manufaturados brasileiros vai para os EUA, nosso maior importador atualmente, correspondendo a US\$ 517 milhões e US\$ 150 milhões em blocos para a Itália (34%) e China (30%) principalmente, somando-se um total de US\$ 781 milhões em rochas ornamentais exportadas no período de

janeiro a setembro de 2006.

No 1º bimestre de 2007 as exportações brasileiras de rochas ornamentais alcançaram o valor de US\$ 138,53 milhões, pela comercialização de 329.838,85 toneladas de rochas brutas e processadas. Se compararmos esses dados com o mesmo período de 2006, houve uma variação positiva de 14,79% no faturamento e de 12,10% no volume físico das exportações. Nesse mesmo contexto as chapas polidas de granito, da posição 6802.93.90, continuam apresentando tendência de queda nos preços médios com variação positiva de apenas 4,9% no primeiro bimestre de 2007. (CHIODI FILHO; ABIROCHAS, 2007).

O Estado do Espírito Santo aparece com mais de US\$ 568 milhões em exportações durante o período de janeiro a outubro de 2006, um aumento de 48,42% em dólares e 36,06% em peso em um comparativo ao mesmo período do ano de 2005. Foram exportadas em 2006 1.225.354 toneladas entre blocos, manufaturados e outras rochas (INFOROCHAS, 2006).

#### 4.2: CLASSIFICANDO AS CHAPAS

O padrão cromático é o principal atributo considerado para a classificação comercial de uma rocha. Em função das características cromáticas, os materiais são enquadrados como clássicos, comuns ou exóticos. Os materiais clássicos não sofrem influência de modismos, incluindo mármore vermelhos, brancos, amarelos e negros, bem como granitos negros e vermelhos. Os materiais comuns, de largo emprego em obras de revestimento, incluem mármore bege e acinzentados, além de granitos acinzentados, rosados e amarronzados. Os materiais exóticos são normalmente utilizados para peças isoladas e pequenos revestimentos, abrangendo mármore azuis, violeta e verdes, além de granitos azuis, amarelos, multicores e brancos (CHIODI FILHO, 2001).

Para se analisar uma chapa, além das informações básicas do material a ser analisado, o classificador deve estar munido de alguns acessórios para assegurar a qualidade e imprimir agilidade no trabalho, tais como: câmera fotográfica, trena de pelo menos 3 metros, paquímetro, etiqueta de certificação e calculadora, observando os seguintes itens: espessura, área líquida, planicidade, rugosidade, trincas, impurezas e defeito do material.

#### 4.3: CLASSIFICAÇÃO DO MATERIAL

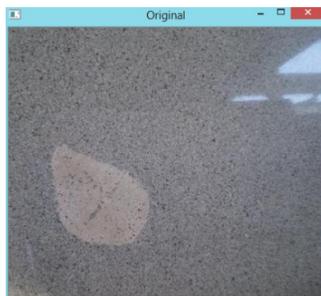
Na etiqueta deve aparecer a classificação do material, como sendo tipo exportação, padrão ou comercial, ou seja, respectivamente, material sem defeitos aparentes, material com algum defeito e material que não permite classificação para exportação, geralmente destinado ao mercado interno que é menos exigente. Essa classificação varia de empresa para empresa, pois alguns produtores subdividem as categorias padrões, como por exemplo, um material tipo exportação poderia ser subdividido em top premium, extra e primeira. Ocorre que, às vezes, aquilo que o produtor classifica como primeira qualidade, o comprador pode não concordar, pelo fato de se existirem defeitos subjetivos que se submetem segundo cada critério pessoal, salvo aqueles defeitos que são problemas físicos claros, como manchas por concentração de minerais máficos ou trincas abertas. Aplicando-se o sistema de subdivisões, procura-se então, aumentar o leque de possibilidades de aceite pelo cliente, mesmo porque há uma pequena variação de preços dentro desta faixa de categoria.

#### 5: ABORDAGEM UTILIZADO NO DESENVOLVIMENTO

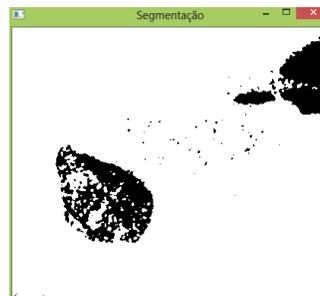
O primeiro passo para análise e entendimento da imagem consiste na segmentação (CONCI et al, 2007). A segmentação é uma tarefa básica em processamento de imagens e visa à identificação e separação de uma imagem em suas partes constituintes. Na segmentação, a imagem é dividida em regiões com propriedades comuns (intensidade, cor, textura, etc). O nível de subdivisão da imagem depende do objetivo que se pretende alcançar. A segmentação se encerra quando as partes constituintes de interesse forem isoladas, o que requer o conhecimento prévio da natureza da imagem. Assim, a segmentação é um problema orientado à aplicação, pois depende do contexto (BERDER, 2003).

A segmentação pode ser utilizada na análise e entendimento de imagens em sistemas de visão computacional e também pode auxiliar a análise feita por um ser humano. Nesta última, as áreas da imagem são identificadas e destacadas, realçando detalhes muitas vezes imperceptíveis para a visão humana. Como demonstra a Figura 2 onde lado esquerdo mostra uma imagem não processada e no

lado direito é mostrada uma imagem após a segmentação da imagem.



**Figura 1:** Imagem Original.  
Fonte: Elaborado pelo Autor (2015).



**Figura 2:** Imagem Segmentada.  
Fonte: Elaborado pelo Autor (2015).

Atualmente existem diversas técnicas de segmentação, entre elas encontra-se a utilizada por esse trabalho que é a segmentação baseada em cor e textura. Nessas técnicas as partes constituintes das imagens são separadas de acordo com a gama de cor que as compõem junto com sua saturação e luminosidade da região. Assim, nesse contexto os objetos constituintes da imagem possuem como característica uma gama de cores, saturações e luminosidade que os identificam.

Para realizar tal segmentação, inicialmente a imagem em análise que se encontra no espaço RGB (Red, Green e Blue) é transformada para o espaço HSV (Hue, Saturation e Value), o qual é mais prático para a manipulação de cores. Em seguida, a cor base do objeto que pretendesse segmentar é selecionada e armazenada junto com sua saturação e luminosidade. Como cada objeto é constituído por uma gama de cor semelhante a variação entre as cores pode ser definida por um fator de tolerância, que informa a faixa de cores, saturação e luminosidade que constituem o objeto partindo da base selecionada. O resultado desse processo pode ser visto na figura 2, a qual foi selecionada como cor base uma área da imagem do original com valor limiar H, S e V, respectivamente valor do limiar inferior para H:0, valor do limiar superior para H:180, valor limiar inferior para S:0, valor do limiar superior para S:256, valor limiar inferior para V:0, valor do limiar superior para V:147.

## 6: VISÃO TÉCNICA GERAL DO PROJETO

### 6.1: COMPONENTES ESSENCIAIS DA IMPLEMENTAÇÃO SÃO:

I: Leitura das imagens de chapas de granito polidas;

II: Conversão das imagens de RGB para HVS;

III: Análise de pixel das imagens para fazer a correta segmentação;

### 6.2: AS OPERAÇÕES IMPLEMENTADAS NO PROJETO PARA O PROCESSO DE SEGMENTAÇÃO PODEM SER RESUMIDAS PELOS PASSOS SEGUINTE:

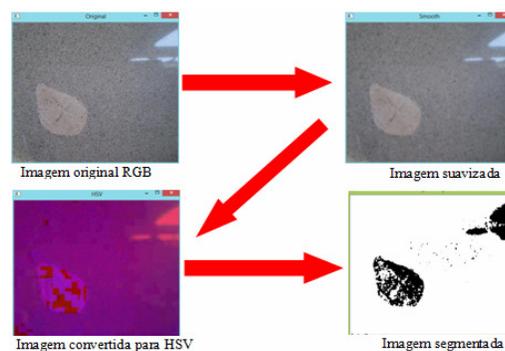
I: Leitura da imagem da chapa de granito polida no formato RGB;

II: Foi aplicado um filtro de suavização gaussiano na imagem original no formato RGB;

III: A imagem foi convertida de RGB para HSV;

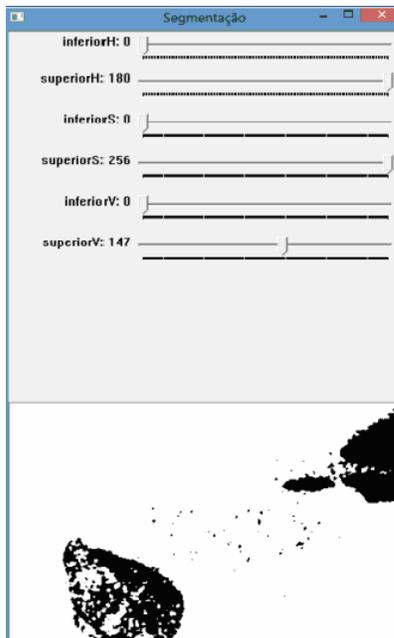
IV: Foi utilizado uma função limiar (threshold) para ajustar os valores limiares inferiores e superiores para H, S e V.

V: Após segmentadas as imagens foram salvas com os padrões.



**Figura 3:** Processo de segmentação da imagem  
Fonte: Elaborado pelo Autor (2015).

### 6.3: TELA PRINCIPAL DO SOFTWARE DESENVOLVIDO PARA A SEGMENTAÇÃO DAS IMAGENS:



**Figura 4:** Tela principal do software desenvolvido com imagen com os defeitos já segmentados.  
Fonte: Elaborado pelo Autor (2015).

## 7: CONCLUSÃO

Este trabalho apresentou uma abordagem de segmentação de cor que tem como principal contexto imagens de chapas de granito polida com objetos que são formados por cores, saturação e luminosidades próximas. Nessas técnicas áreas de regiões defeituosas são identificadas de forma semi-automáticas, para diminuir diferenças entre identificação feita caso exista pode ser feito ajuste manual dos limiares para uma melhor segmentação.

A abordagem apresentada nesse trabalho procura delimitar uma área apontada com características não desejada em uma chapa de granito polida de acordo seu padrão de coloração com um fator de tolerância escolhido. Esse fator irá dizer a variação de cores, saturação e luminosidade que o objeto apontado é formado, sendo com isso, a identificação do defeito na imagem feita de forma semi-automática.

Devido ao fator de tolerância e a identificação da área ainda ser feito de forma

semi-automática, a técnica aqui utilizada ainda é sujeita a erros, menores que o da técnica manual descrita nas seções anteriores. Desse modo um aperfeiçoamento desse trabalho, seria utilizar-se de técnicas para calcular tal fator de tolerância utilizado, proporcionando assim, uma análise feita de forma totalmente automática.

Como trabalho futuro essas imagens segmentadas serão utilizadas em uma rede neural com o objetivo de reconhecer e classificar esses defeitos de forma automática.

## 8: REFERÊNCIAS BIBLIOGRÁFICAS

AGUIAR, R. G.; **Segmentação de Imagens em cores de Imunohistoquímica**. 2000. Dissertação (Mestrado em Ciências da Computação) – Universidade Federal de São Carlos. São Carlos, 2000.

ALBUQUERQUE, M. P. **Processamento de imagens**: métodos e análises. Disponível em: <<http://www.cbpf.br/cat/lpdsi/pdf/ProcessamentoImagens.PDF>>. Acesso em: 6 jun. 2013.

ARTUR, J.S. **Reconhecimento de padrões usando indexação recursiva**. 1999. Tese (Doutorado em Engenharia de Produção). Universidade Federal de Santa Catarina, Florianópolis, 1999.

ASSOCIAÇÃO BRASILEIRA DA INDÚSTRIA DE ROCHAS ORNAMENTAIS. (ABIROCHAS ). **Balanco das exportações de rochas ornamentais brasileiras**: no 1º. bimestre de 2007: Disponível em: <[http://www.abirochas.com.br/noticias.php?not\\_id=&pagNews=10](http://www.abirochas.com.br/noticias.php?not_id=&pagNews=10)>. Acesso em: 12 mar. 2014.

\_\_\_\_\_. **Exportações batem recorde em agosto**: síntese das exportações e importações brasileiras de rochas ornamentais e de revestimento no período janeiro. Disponível em: <[http://www.ivolution.com.br/mais/fotos/6/17/2027/Informe\\_14\\_2013.pdf](http://www.ivolution.com.br/mais/fotos/6/17/2027/Informe_14_2013.pdf)> . Acesso em: 20 set. 2013.

\_\_\_\_\_. **Feira de Vitória**: quartzitos são os novos materiais exóticos brasileiros. São Paulo: Associação Brasileira da Indústria de Rochas Ornamentais, 2007. Disponível em: <[http://www.abirochas.com.br/noticias.php?not\\_id=&pagNews=10](http://www.abirochas.com.br/noticias.php?not_id=&pagNews=10)>. Acesso em: 12 mar. 2014..

\_\_\_\_\_. **A importância da feira de dubai para o setor brasileiro de rochas ornamentais e de revestimento.** Disponível em: <[http://www.abirochas.com.br/noticias.php?not\\_id=&pagNews=10](http://www.abirochas.com.br/noticias.php?not_id=&pagNews=10)>. Acesso em: 12 mar. 2014..

\_\_\_\_\_. **Relatório das feiras de Beijing e Xiamen.** São Paulo: Associação Brasileira da Indústria de Rochas Ornamentais, 2007. Disponível em: <[http://www.abirochas.com.br/noticias.php?not\\_id=&pagNews=10](http://www.abirochas.com.br/noticias.php?not_id=&pagNews=10)>. Acesso em: 12 mar. 2014..

ASSOCIAÇÃO NORDESTE DE PRODUTORES DE PEDRAS ORNAMENTAIS DO ESPÍTIRO SANTO (ANPO). Exportações capixabas ultrapassam resultados do ano de 2005. **Revista Rochas de Qualidade.** V.31, n.191, nov-dez., 2006. p 127. Disponível em: < <http://anpo.com.br/main.asp?link=noticia&id=116>>. Acesso em: 12 jan.2014.

BALLARD, D. H.; C.M. BROWN. **Computer vision.** New Jersey: Prentice-Hall,1982.

BERDER, T.C. **Classificação e recuperação de imagens por cor utilizando técnicas de inteligência artificial.** Dissertação (Mestrado em Computação Aplicada) - Universidade do Vale do Rio dos Sinos, São Leopoldo, 2003.

BRADSKI, G.; KAEHLER, A. **Learning opencv:** computer vision with the opencv library. New York-US: O'Reilly Media, 2008.

BISHOP,C.M. **Neural networks for pattern recognition.** Oxford: Claren-donPress. 1995.

CONCI, A; AZEVEDO, E; LETA, F. **Computação gráfica:** teoria e prática. São Paulo: Elsevier, 2007. V.2

CHIODI FILHO, Cid. Kistemann. **Rochas ornamentais no século XXI.** São Paulo: Associação Brasileira da Indústria de Rochas, 2001. Disponível em: <<http://www.abirochas.com.br>>. Acesso em: 12 jan.2014.

GONZALEZ, R. C.; WOODS, R.E. **Processamento digital de imagens.** São Paulo: Pearson, 2009.

HAYKIN, S. **Redes neurais :** princípios e prática. 2. ed. Porto Alegre: Bookman, 2001

HENRIQUE, J. B. R. **Desenvolvimento de uma técnica de reconhecimento de padrões baseada em distância**. 2003. Dissertação. (Mestrado em Engenharia Elétrica) - Universidade Federal de Santa Catarina, 2003. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/84988>>. Acesso em: 12 jan. 2014.

HUGO, MARCEL. **Uma interface de reconhecimento de voz para o sistema de gerenciamento de central de informações de fretes**. Monografia (Pós-Graduação em Engenharia de Produção) - Universidade Federal de Santa Catarina, Florianópolis, 1995.

MASCARENHAS, N. D. A. Breve introdução ao reconhecimento estatístico de padrões. In: SIMPÓSIO DE INTELIGÊNCIA ARTIFICIAL E REUNIÃO ANUAL DA SOCIEDADE BRASILEIRA PARA O PROGRESSO DA CIÊNCIA, 39, 1987, Brasília. **Anais....** São Paulo: SBC, 1988. p.12. Disponível em: <<http://mtc-m12.sid.inpe.br/col/sid.inpe.br/iris@1912/2005/07.18.23.52.36/doc/INPE%204206.pdf?metadatarpository=&mirror=sid.inpe.br/banon/2001/04.06.10.52.39>>. Acesso em: 05 de Fev. 2015.

MOSSMANN, S. R. **Extração e análise de características de imagens digitais para construção de um modelo de reconhecimento de padrões**. 2010. 64f. Monografia (Graduação em Sistemas de Informação) - Universidade Feevale, Novo Hamburgo, 2010.

OSÓRIO, F. S. **Um estudo sobre o reconhecimento visual de caracteres através de redes neurais**. 1991. Dissertação (Mestrado em Computação) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 1991.

PAL, U.; CHAUDHURI, B. B. Indian script character recognition: a survey. **Pattern Recognition**, v. 37, p. 1887-1899, 2004. Disponível em: <[https://www.researchgate.net/publication/222197898\\_Chaudhuri\\_BB\\_Indian\\_script\\_character\\_recognition-A\\_survey\\_Pattern\\_Recognition\\_37\\_1887-1899](https://www.researchgate.net/publication/222197898_Chaudhuri_BB_Indian_script_character_recognition-A_survey_Pattern_Recognition_37_1887-1899)>. Acesso em: 05 de Fev. 2015.

PEDRINI, H. **Processamento e Análise de Imagens (MC940)**, 2015. Disponível em <[http://www.ic.unicamp.br/~helio/disciplinas/MO445/aula\\_fundamentos.pdf](http://www.ic.unicamp.br/~helio/disciplinas/MO445/aula_fundamentos.pdf)>. Acesso em: 05 fev 2015.

PEDRINI, H; SCHWARTZ, W. R. **Análise de imagens digitais: princípios, algoritmos, e aplicações**. São Paulo: Thompson Learning, 2008.

PERELMUTER, G., CARRERA, E. V., VELLASCO, M., PACHECO, A. reconhecimento de imagens bidimensionais utilizando redes neurais artificiais. In: SIMPÓSIO BRASILEIRO DE COMPUTAÇÃO GRÁFICA E PROCESSAMENTO DE IMAGENS, 8, 1995, 31 out. – 30 dez; Campinas-SP. **Anais...** Campinas-SP: UNICAMP, 1995. p. 197-203. Disponível em: <<http://www.bv.fapesp.br/pt/auxilios/36051/anais-do-sibgrapi-95-viii-simposio-brasileiro-de-computacao-grafica-e-processamento-de-imagens/>>. Acesso em: 12 jan 2015.

PRADO JÚNIOR, A, ELFES, A. **Um projeto em reconhecimento de padrões de forma**. 1975. 70 f. Monografia (Graduação em Ciência da Computação) - Instituto Tecnológico da Aeronáutica, São José dos Campos- SP, 1975.

SAMPAIO, Reinaldo Dantas. Balanço de importações e exportações de rochas. **Informativo do SIMAGRAN**, Salvador-Bahia, v.1, n.1, maio, 2007. Disponível em: <>. Acesso em: 12 jan. 2014.

STEINER , M.T.A. Uma **Metodologia de para o reconhecimento de Padrões multivariados com resposta dicotômica**. Tese (Doutorado em Engenharia de Produção) - Universidade Federal de Santa Catarina, Florianópolis-SC, 1995

TODESCO, J. L. **Reconhecimento de padrões usando rede neuronal artificial com uma função de base radial**: uma aplicação na classificação de cromossomos humanos. 1995. Tese. (Doutorado em Engenharia de Produção) - Universidade Federal de Santa Catarina, Florianópolis, 1995

