

# RECONHECIMENTO DE CADEIAS DE NÚMEROS NAS ETIQUETAS IDENTIFICADORAS DOS BLOCOS DE AÇO, UTILIZANDO PROCESSAMENTO DIGITAL DE IMAGENS E REDES NEURAS ARTIFICIAIS

RENATO DE FREITAS LARA

Departamento de Ciência da Computação - Universidade Presidente Antônio Carlos  
(UNIPAC) - Barbacena - MG – Brasil

renato.lara@gerdau.com.br

**Resumo.** Este artigo tem como objetivo mostrar o desenvolvimento de um protótipo de sistema de visão, que faz leitura e identificação da ordem de produção de tarugos de aço da Gerdau Açominas. É função desse sistema, evitar que materiais de ordens de produção diferentes sejam inseridos erroneamente no forno do fio máquina, alertando o operador da cabine de comando. São usadas técnicas de Inteligência Computacional, especificamente redes neurais artificiais. Também são usadas ferramentas/rotinas de processamento de imagem, para melhor leitura dos números contidos nas etiquetas através das imagens coletadas por câmera de vídeo.

## 1. Informações gerais

Nesta seção, serão descritas as principais características do ambiente industrial, bem como os motivos de se desenvolver um sistema para reconhecimento de números de ordem de produção.

### 1.1. O ambiente industrial

O ambiente industrial que é referenciado durante o artigo, é uma linha de produção que tem o objetivo de transformar blocos de aço em bobinas de fio – máquina, (arame de aço com diâmetro específico). A primeira etapa do processo dessa área, consiste no reaquecimento de tarugos, (blocos de aço de dimensões específicas), para posterior transformação em bobinas. Esse reaquecimento ocorre no forno da laminação de fio – máquina.

Existe um sistema de rastreamento de linha de produção. Tal rastreamento se inicia na entrada do forno citado no parágrafo anterior. Existe um operador responsável por fazer a leitura da ordem de produção de cada peça que entrará no forno. O mesmo usa um leitor de código de barras para isso. A ordem de produção a ser lida se encontra em uma etiqueta fixada nos blocos de aço, tal número, após lido, é alimentado no sistema de rastreamento.

O ambiente industrial em questão é bastante inóspito. O operador fica exposto a poeira e calor para executar a tarefa. Outro agravante é a dificuldade de leitura. As etiquetas podem vir com defeitos, sujeiras e faltando partes. Quando o leitor de código de barras não consegue fazer a leitura, o operador alimenta o número de ordem de produção manualmente no sistema.

Devido aos motivos descritos acima, sentiu-se a necessidade de criação de um ambiente computacional para identificação dessa ordem de produção. O objetivo principal do mesmo é, além de evitar a exposição desse operador ao ambiente agressivo, resolver o problema da ineficiência dos leitores de códigos de barras.

Nesse artigo é mostrado o desenvolvimento de tal ambiente. O mesmo se assemelha a um “olho artificial”, que fará a leitura e o reconhecimento da ordem de produção contida nas etiquetas dos tarugos. Serão usadas nesse sistema, inteligência computacional, (Redes Neurais Artificiais), e Processamento Digital de Imagens.

Esse artigo está organizado da seguinte maneira: Na seção 2, será abordado o processo de reconhecimento da ordem de produção com seu respectivo fluxograma. Na seção 3, serão abordados os aspectos básicos de Redes Neurais Artificiais. Na seção 4, serão mostrados alguns fundamentos de Processamento Digital de Imagens. Na seção 5, será descrito o funcionamento dos módulos desenvolvidos, e as principais telas do software de reconhecimento dos números da ordem de produção.

## **2. Processo de reconhecimento**

A entrada do processo, será uma foto coletada por câmera de vídeo, que deverá ser aberta de um diretório específico. A saída do processo será a cadeia de números da ordem de produção salva em um arquivo TXT, no mesmo diretório que a imagem foi aberta.

O processo de PDI, (processamento digital de imagens) está sendo dividido em três etapas. A parte de treinamento e teste da rede neural também foi feito separadamente. Foram criados até o presente momento três módulos do sistema. Um contendo a parte de rede neural, outro contendo as etapas do processo de PDI, e um último sendo a junção de tudo de forma mais enxuta e funcional possível, para ser testado no ambiente industrial. A Figura 1 mostra as fases do reconhecimento:

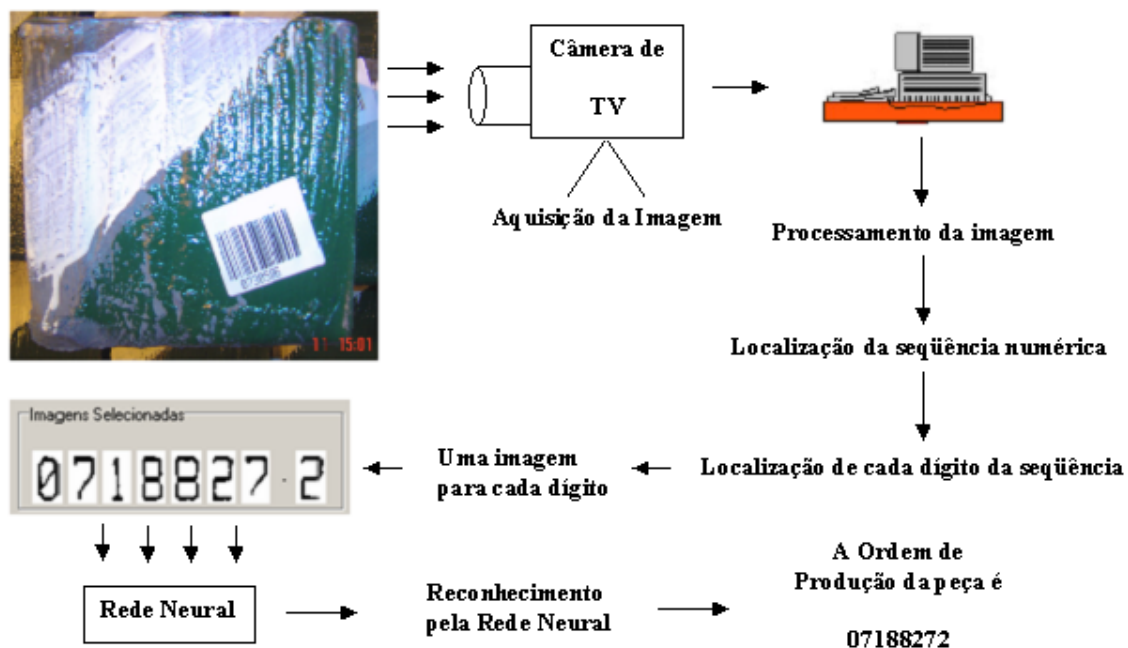


Figura 1: Fluxograma do proceso de reconhecimento

## 2.1. Dificuldades no reconhecimento

Existe uma mesa de rolos, onde o material passa para entrar no forno. Essa mesa é desligada para que o operador faça a leitura do material parado. Com o novo sistema, deve existir um sensor de proximidade que posicionará o material no local exato para coleta da imagem pela câmera. Variações de distância do tarugo em relação a aproximação frontal da câmera pode influenciar no ajuste de foco.

Variações de posicionamento do tarugo no eixo lateral, podendo não parar no ponto exato do ângulo de visão da câmera, bem como a proximidade dos mesmos, pode influenciar na qualidade da imagem.

Rotação dos tarugos, podendo deixar os números da ordem de produção de ponta cabeça, de lado ou na posição correta, também pode influenciar na qualidade da imagem, como mostra a Figura 2:



**Figura 2: Possíveis posições das etiquetas**

### **3. Redes Neurais Artificiais (RNAs)**

Uma das principais áreas da Inteligência Artificial pode ser considerada a que trata de Redes Neurais Artificiais, (RNAs). Redes Neurais Artificiais, tentam imitar o funcionamento das redes neurais biológicas que formam o cérebro humano. Nessa unidade serão abordados os fundamentos de uma RNA, será abordado também a arquitetura, funcionamento e treinamento de uma RNA modelo MLP.

#### **3.1. Fundamentos**

Redes Neurais são estruturas de dados e algoritmos projetados para simular o processamento de informações ocorrido nas estruturas neurais biológicas, principalmente nos organismos humanos, fazendo um processamento paralelo e distribuído das informações.

De acordo com [Kovács, 1996] Redes Neurais são sistemas complexos constituídos por elementos representando algumas das características dos neurônios que constituem o sistema nervoso de seres vivos e permitindo sua interação com o ambiente que os cerca. Trata-se de assunto tipicamente interdisciplinar interessando à engenheiros, matemáticos, fisiologistas, psicólogos, etc. Conseqüentemente não existe uma normalização relativa à nomenclatura e notação, entre outras. São construídas com base no funcionamento de redes de neurônios humanos. Esta estrutura

é composta de ligações entre neurônios, sendo que estas ligações formam diversos tipos de topologias. A Rede Neural é capaz de realizar aprendizado e fazer generalizações a partir de uma base de dados de exemplos.

### 3.2. Redes MLP (Perceptron Multi-camada)

Redes de uma só camada resolvem somente problemas linearmente separáveis. Problemas que não são linearmente separáveis requerem o uso de redes multi-camadas que dispõe de uma ou mais camadas intermediárias.

Redes MLP apresentam um poder computacional muito maior do que aquele apresentado pelas redes sem camadas intermediárias. Ao contrário destas redes, MLPs podem tratar com dados que não são linearmente separáveis. Teoricamente, redes com duas camadas intermediárias podem implementar qualquer função, linearmente separável ou não [Braga, Ludemir e Carvalho, 2000]. Segundo [Haykin, 2001] as redes MLP têm sido aplicadas com sucesso para resolver diversos problemas complexos.

#### 3.2.1. Arquitetura de uma RNA modelo MLP

Basicamente a rede é composta de um conjunto de unidades sensoriais (nós de fonte) que constituem a camada de entrada, uma ou mais camadas ocultas de nós computacionais e uma camada de saída. A Figura 3 exibe a estrutura de uma rede MLP com duas camadas intermediárias. Pode-se notar a separação entre a camada de entrada, as camadas intermediárias e a camada de saída. A Figura 3 foi extraída de [WWW01].

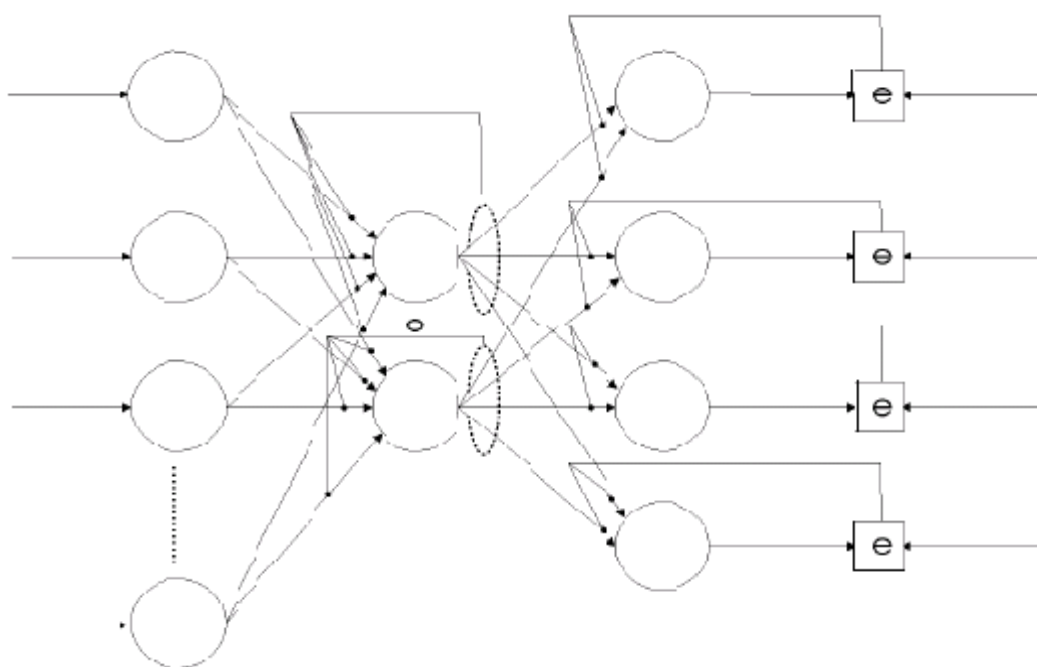


Figura 3: Estrutura de uma Rede MLP

A camada de entrada contém as unidades que recebem os estímulos, ou seja, representa a camada de nodos que receberão os sinais, ou padrões, de entrada.

As camadas ocultas, ou intermediárias, não fazem parte nem da entrada nem da saída da rede. Os neurônios destas camadas conferem a rede a capacidade de aprender tarefas complexas extraíndo progressivamente as características mais importantes dos padrões aplicados a entrada da rede. De acordo com [Yoneiama,2002] o uso de unidades escondidas torna possível a recodificação dos padrões de entrada. Cada camada escondida executa a recodificação do seu sinal de entrada. A maioria das bibliografias estudadas apontam as camadas internas como sendo unidades extratoras de características, pois criam representações internas extraíndo atributos dos dados fornecidos.

Segundo [Yoneiama,2002] dependendo do número de unidades escondidas pode ser possível encontrar uma nova representação na qual os vetores de saída da camada escondida são linearmente separáveis, mesmo que os vetores de entrada não sejam. Esta recodificação interna dos padrões de entrada é então enviada a camada de saída da rede.

Cada nodo da camada de saída ainda executa uma função computacional através do processamento dos sinais recebidos das camadas intermediárias ou diretamente da camada de entrada. Durante o processo de treinamento, o valor de saída fornecido por esta camada será comparado com um valor ideal de saída. Com a diferença, (Saída Ideal - Saída da Rede), obtém-se um erro que será utilizado para reajustar os pesos das sinapses.

O modelo estrutural de cada neurônio da rede possui uma função de ativação. Esta função de ativação deve ser uma função não linear pois uma rede com mais de uma camada cujos nodos utilizam funções de ativação lineares é equivalente a uma rede de uma só camada. Isto quer dizer que se uma rede Multi-Camada utilizar funções de ativação lineares, ela limitará sua capacidade de processamento em problemas linearmente separáveis como as redes de uma só camada. A demonstração deste fato pode ser encontrado em [Braga, Ludemir e Carvalho, 2000].

### **3.2.2. Funcionamento de uma Rede MLP**

Em uma rede multi-camadas, o processamento realizado por cada nodo é definido pela combinação dos processamentos realizados pelos nodos da camada anterior que estão conectados a ele. Quando se segue da primeira camada intermediária em direção à camada de saída, as funções implementadas se tornam cada vez mais complexas. Estas funções definem como é realizada a divisão do espaço de decisão [Braga, Ludemir e Carvalho, 2000].

Segundo Braga o funcionamento das redes MLP com pelo menos duas camadas intermediárias pode ser resumido da seguinte maneira: na primeira camada intermediária cada nodo traça retas no espaço de padrões de treinamento. Na Segunda camada intermediária cada nodo combina as retas

traçadas pelos nodos da camada anterior conectados a ele, formando regiões convexas, onde o número de lados é definido pelo número de unidades a ele conectadas. Na camada de saída cada nodo forma regiões que são combinações das regiões convexas definidas pelos nodos e ele conectados da camada anterior. Os nodos definem, desta maneira, regiões com formatos abstratos [Braga, Ludemir e Carvalho, 2000].

### **3.2.3. Treinamento de uma rede MLP - Back-Propagation**

De acordo com [Braga, Ludemir e Carvalho, 2000] Existem vários algoritmos de treinamento utilizados em redes MLP sendo em sua maioria do tipo supervisionado. Em função do tipo de parâmetros que eles atualizam, os algoritmos para treinamento de redes MLP podem ser classificados como estáticos e dinâmicos. Enquanto os estáticos alteram somente os valores dos pesos sinápticos, os algoritmos dinâmicos alteram também o tamanho da rede (quantidade de camadas, número de nodos nas camadas intermediárias, número de conexões).

O algoritmo de treinamento de MLP mais utilizado é o Algoritmo de Retro Propagação de Erro ou simplesmente "back-propagation". Este algoritmo será tratado com maior nível de detalhes pois além de ser o mais popular foi utilizado na realização do trabalho a que se refere esse artigo.

O algoritmo de retropropagação, ou de back-propagation, para redes de neurônios de múltiplas camadas foi essencialmente inventado e popularizado por Rumelhart, Hilton e Williams, resolvendo uma das limitações fundamentais para o treinamento de redes complexas [Kovács, 1996].

Segundo [Braga, Ludemir e Carvalho, 2000] a maioria dos métodos de aprendizado para RNAs do tipo MLP utiliza variações do algoritmo back-propagation e que ele foi um dos principais responsáveis pelo ressurgimento do interesse em RNAs quando foi publicado em 1986.

O desenvolvimento do algoritmo de retropropagação representa um marco nas redes neurais, pois fornece um método computacional eficiente para o treinamento de Redes MLP. Apesar de não se poder afirmar que o algoritmo de retropropagação forneça uma solução ótima para todos os problemas resolúveis, ele acabou com pessimismo sobre a aprendizagem em máquinas de múltiplas camadas que havia sido causado pelo livro de Minsky e Papert de 1969 [Haykin, 2001].

O algoritmo back-propagation é um tipo de algoritmo supervisionado que utiliza correção de erros para ajustar os pesos da rede. O back-propagation utiliza pares "entrada / saída desejada" para efetuar o processo de treinamento da rede. Neste processo um sinal (padrão de entrada) é aplicado à entrada da rede que gera uma saída correspondente. Esta saída da rede é comparada com uma saída ideal (saída desejada). A diferença entre estas duas saídas (Saída Ideal - Saída Rede) gera um sinal de erro que será utilizado para ajustar os pesos da rede.

Segundo [Braga, Ludemir e Carvalho, 2000] o treinamento pode ocorrer em duas fases em que cada fase percorre a rede em um sentido. Um passo para frente, forward ou propagação e outro para trás. A retropropagação ou backward. A fase forward é utilizada para definir a saída da rede em função de um sinal aplicado a entrada. A fase backward utiliza a saída desejada e a saída fornecida pela rede para atualizar os pesos de suas conexões.

No passo para frente, um padrão de atividade (vetor de entrada) é aplicado aos nós sensoriais da rede e seu efeito se propaga através da rede, camada por camada. Finalmente, um conjunto de saídas é produzido como a resposta real da rede. Durante o passo de propagação, os pesos sinápticos da rede são todos fixos. Durante o passo para trás, por outro lado, os pesos sinápticos são todos ajustados de acordo com uma regra de correção de erro. Especificamente, a resposta real da rede é subtraída de uma resposta desejada (alvo) para produzir um sinal de erro. Este sinal de erro é então propagado para trás através da rede, contra a direção das conexões sinápticas - vindo daí o nome de "retropropagação de erro" (error back-propagation). Os pesos sinápticos são ajustados para fazer com que a resposta real da rede se mova para mais perto da resposta desejada, em um sentido estatístico (Haykin, 2001).

#### **4. Processamento Digital de Imagens (PDI)**

Para que se entenda o conceito de processamento de imagem vale mencionar o conceito de "imagem". Segundo [Gonzalez e Woods, 2000] o termo "imagem" refere-se a uma função de intensidade luminosa bidimensional, denotada por  $f(x,y)$ , em que o valor ou amplitude de  $f$  nas coordenadas espaciais  $(x,y)$  dá a intensidade (brilho) da imagem naquele ponto. Para melhor entender pode-se tomar a função imagem como sendo uma matriz bidimensional onde cada elemento  $x,y$ , coluna e linha, seja um ponto, (pixel), com um valor que representa a intensidade luminosa neste ponto. Para concluir pode-se dizer que cada ponto, ou "pixel", tem um valor numérico que representa uma cor.

As imagens que as pessoas percebem em atividades visuais corriqueiras consistem de luz refletida dos objetos. A natureza básica de  $f(x,y)$  pode ser caracterizada por dois componentes: (1) a quantidade de luz incidindo na cena sendo observada e (2) a quantidade de luz refletida pelos objetos na cena. Apropriadamente, esses componentes são chamados iluminação e reflectância, respectivamente, e são representados por  $i(x,y)$  e  $r(x,y)$ . O produto das funções  $i(x,y)$  e  $r(x,y)$  resulta  $f(x,y)$  [Gonzalez e Woods, 2000]. Onde cada ponto de coordenada  $(x,y)$  da imagem  $f$  tem um valor que representa a intensidade de luz (luz incidida x luz refletida) neste ponto. Tendo definido o termo "imagem" agora pode-se esclarecer processamento de imagem digital.

As imagens digitais representam imagens do mundo real. Como são imagens digitais cada ponto ou pixel possui um valor discreto para representar a cor naquele ponto. Imagens coloridas são amplamente. Porém grande parte das aplicações também



trabalham com imagens em tons de cinza onde cada ponto (x,y) da imagem possui um valor discreto entre 0 e 255. Nesta escala cada cor entre 0 e 255 representa um nível de cinza. O preto ("ausência de luz"), é representado pelo 0 e o branco pelo número 255. Sistemas de visão computacional geralmente utilizam imagens digitais em tons de cinza pois o custo computacional é menor quando se trabalha com um universo menor de cores.

O interesse em métodos de processamento de imagens digitais decorre de duas áreas principais de aplicação: melhoria de informação visual para a interpretação humana e o processamento de dados de cenas para percepção automática através de máquinas [Gonzalez e Woods, 2000].

Em processamento de imagem digital, o sistema admite como entrada uma imagem que, após processada, produz outra imagem na saída. Esta imagem pode ser observada num dispositivo gráfico. Um exemplo clássico desta área é o processamento de imagens enviadas por um satélite com o objetivo de colorir ou realçar detalhes [Gomes e Velho, 1994].

O processamento de imagens digitais visa processar uma imagem de modo que o resultado final, a "Imagem de Saída", seja de melhor qualidade para uma determinada aplicação. O termo "melhor qualidade" é bastante relativo visto que o processamento de imagem utilizado para melhorar uma informação visual aos olhos dos seres humanos pode não ser o mesmo utilizado para melhorar a visualização de uma máquina. Uma imagem de péssima qualidade, sem cor, sem forma definida aos olhos humanos pode ser uma excelente fonte de informação para um computador. Ao mesmo tempo que uma imagem bastante nítida ao olhos humanos pode não ser adequada aos olhos de um computador. Os próximos tópicos abordarão as principais técnicas de processamento de imagem, ou seja, como uma imagem pode ser manipulada, alterada, de forma que o resultado do processamento seja uma imagem que seja de melhor qualidade para uma aplicação qualquer.

## **5. Reconhecimento de caracteres em etiquetas usando Processamento Digital de Imagens(PDI) e Redes Neurais Artificiais (RNAs)**

Nesse tópico, serão abordadas as principais etapas do desenvolvimento do sistema de visão, descrito na primeira seção desse artigo. Será descrito também, o funcionamento dos módulos desenvolvidos, bem como as telas do software.

### **5.1. Atividades**

Foram desenvolvidas paralelamente atividades na área de processamento digital de imagens e Redes Neurais Artificiais.

### **5.2. Módulo da Rede Neural**

Boa parte do código foi adaptado de [Frontino, 2003] O mesmo apresenta de forma simples a parte teórica de RNAs. Logo em seguida o autor consegue fazer uma ponte entre a teoria e a prática, mostrando exemplos de aplicações reais usando a ferramenta Delphi. No livro é explicado o funcionamento de um componente genérico de construção de redes neurais. Tal componente possibilita a construção de uma rede neural. Fica a cargo do programador, a manipulação do mesmo de forma correta para a construção da arquitetura de rede desejada. A desvantagem do mesmo é que não é portátil. É mostrado também um exemplo de reconhecimento ótico de caracteres, (*Optical Rocoganized Character – OCR*), que teve parte aproveitado na implementação da rede neural.

### **5.2.1. Desenvolvimento da Rede Neural**

Foi desenvolvida a parte de treinamento e teste da rede neural. O componente MLP, que monta um modelo, *Multilayer perceptron*, (Perceptron multi-camada), foi utilizado na aplicação.

O conhecimento da rede neural é guardado em um arquivo de conhecimento que na hora do reconhecimento é carregado nas sinapses. Foram usadas 20000 interações do conjunto de treinamento. Mais detalhes serão possíveis de se demonstrar com o software.

### **5.2.2. Arquitetura da RNA**

Usados 720 neurônios na camada de entrada da rede, o que equivale ao reconhecimento de uma imagem de 20x36 pixels. Cada neurônio da camada de entrada é excitado com o valor de cada pixel da imagem. Uma camada oculta com 8 neurônios. Esse número de neurônios é definido empiricamente. Usados 10 neurônios na camada de saída. Cada neurônio da camada de saída deverá se referir a um dos dígitos da base decimal.

### **5.2.3. Comportamento do erro**

A taxa de erro com a arquitetura descrita, teve tendência a convergir para algo bem próximo de zero. É mostrado na Figura 4 o comportamento do erro após 200 interações do conjunto de teste.

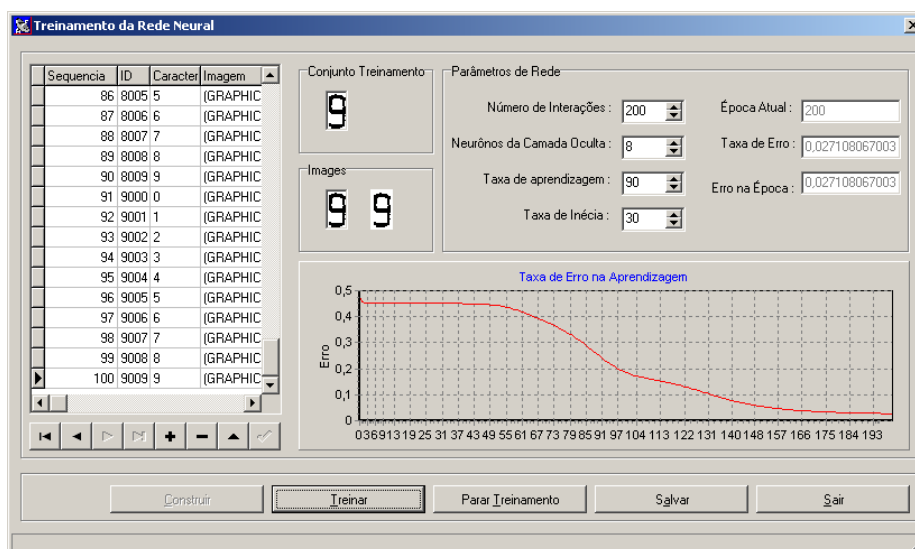


Figura 4: Comportamento do erro após 200 interações

#### 5.2.4. Média de acerto no reconhecimento

Foi verificado que essa arquitetura consegue resolver problemas que não eram linearmente separáveis. Teve bastante sucesso no reconhecimento de um conjunto de teste diferente do que a rede foi treinada. Média de 97% de acerto.

Percebeu-se que quando a rede sabia o valor, o neurônio de saída que representava o valor reconhecido estava com o valor tendendo a 1 e os demais neurônios tendendo a 0. Quando a rede errava o valor, todos os neurônios de saída ficavam com valores bem próximos de 1. Com isso foi implementada uma rotina que quando a rede estava em dúvida quanto ao reconhecimento, ela informava ao usuário do sistema que não sabia o valor. Isso trouxe mais confiabilidade ao reconhecimento.

### 5.3. O módulo de Processamento Digital de Imagem

Na parte de PDI, (processamento digital de imagens), estão sendo utilizadas ferramentas simples de programação, usadas inclusive na parte de redes neurais. Foi desenvolvido outro módulo em Delphi para a parte de PDI.

#### 5.3.1. Separação da região dos números de uma cadeia

Foi desenvolvida a parte de extração de dígitos de uma cadeia de números. A entrada desse processo é um arquivo BitMap contendo cadeia de números, a saída, são oito arquivos BitMap's no formato de entrada da rede neural, (20x36 pixels). O conteúdo dos arquivos, se referem à região onde se encontra o dígito da ordem de produção. A Figura 5 mostra a tela em questão.



Figura 5: Separação dos dígitos da ordem de produção

### 5.3.2. Separação da região da cadeia de números da etiqueta

Foi desenvolvida a parte de extração da cadeia de números contida na etiqueta. A entrada desse processo é um arquivo Bitmap contendo a foto da etiqueta. A saída é um outro arquivo BitMap contendo a região onde se encontra a cadeia de números da ordem de produção. A saída desse processo é a entrada do processo descrito no item anterior. A Figura 6 mostra a tela em questão:

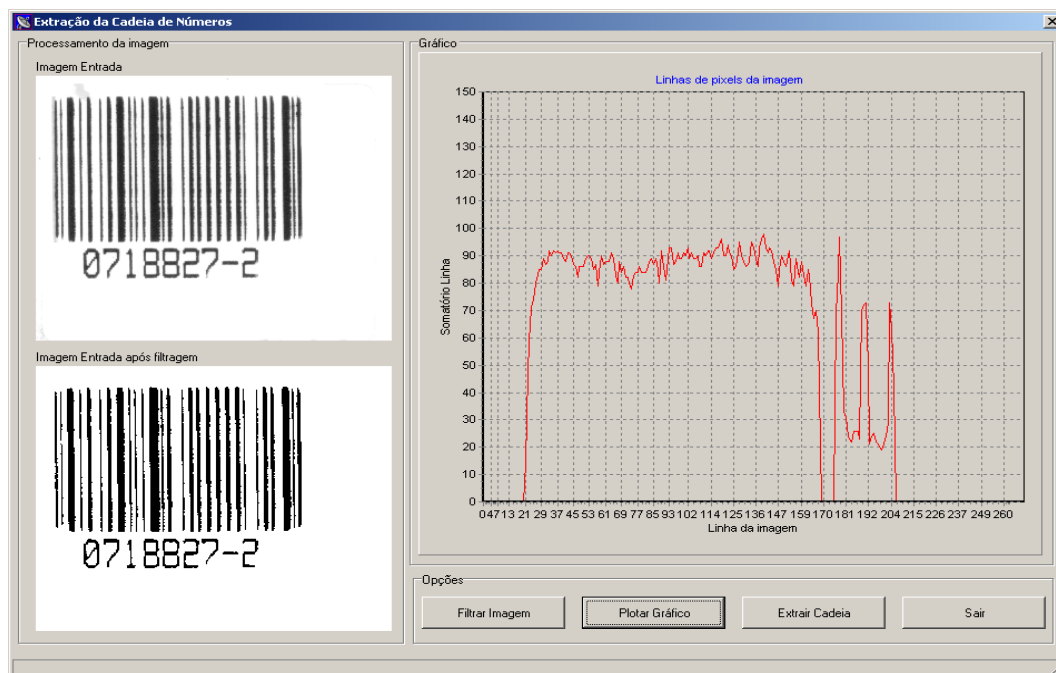


Figura 6: Separação da região da cadeia de números contida na etiqueta

### 5.3.3. Separação da região da etiqueta na foto de entrada

Na parte de extração da etiqueta, que fecharia o projeto, foram usadas várias idéias como variância de cores e filtros de sobel. Porém o custo computacional ficou muito elevado, o que inviabilizou o uso de tais práticas. Será usado em uma atividade futura, a idéia de vetorização de imagens, onde será verificada a existência de todas as possíveis linhas da imagem.

### 5.4. Junção dos módulos PDI e RNA

Foi desenvolvida uma junção contendo todos os módulos desenvolvidos. Com isso, é alimentada uma foto de etiqueta e o mesmo retorna o valor reconhecido, e escreve o mesmo em um arquivo TXT que é salvo no mesmo diretório da foto de entrada. Foram incluídas rotinas para reconhecimento de fotos defeituosas, com sujeiras, riscos e outros. A separação e o reconhecimento ocorreram em praticamente 100% dos casos. É mostrado na Figura 7 a tela principal do programa ilustrando o reconhecimento de uma imagem defeituosa:

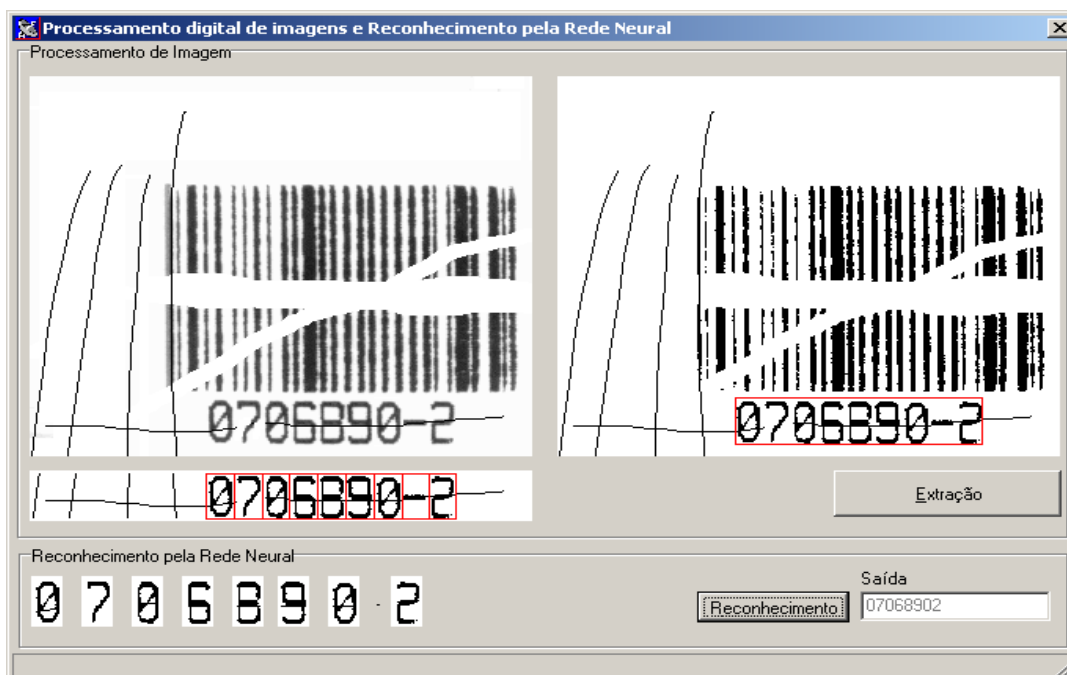


Figura 7: Tela principal da Junção entre módulos com reconhecimento de imagens defeituosas

## 5.5. Diagrama de fluxo de dados

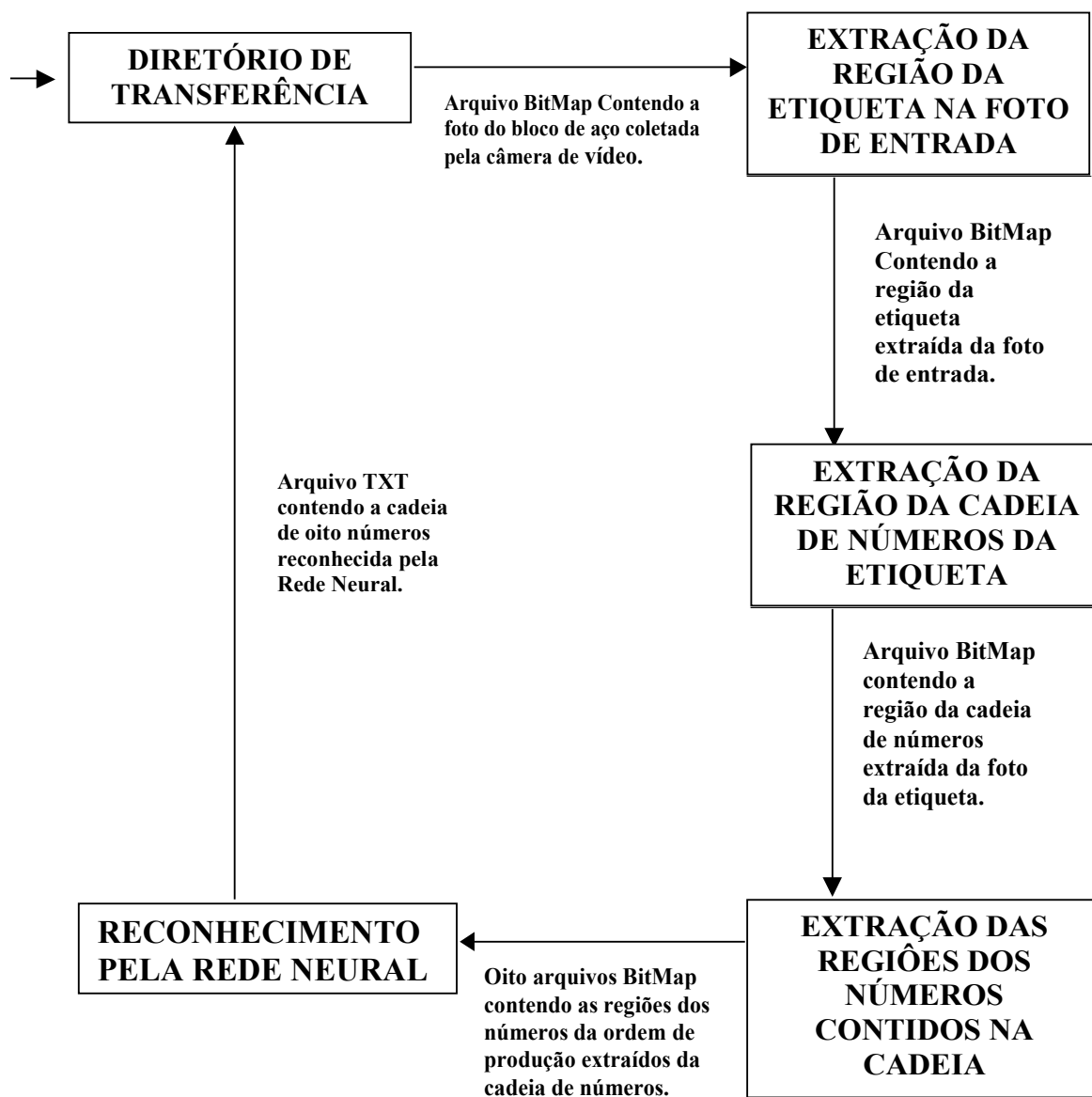


Figura 8: Diagrama de Fluxo de dados da aplicação

Como é mostrado na Figura 8, o processamento de informações foi dividido em 4 etapas. A primeira consiste em localizar e extrair a etiqueta da foto de entrada. A segunda consiste em localizar e extrair a cadeia de números da foto da etiqueta. A terceira consiste em localizar e separar cada um dos dígitos da etiqueta para reconhecimento pela Rede Neural. A quarta e última etapa consiste no reconhecimento pela rede neural e salvamento da cadeia de números em um arquivo TXT, no mesmo diretório que foi aberta a foto de entrada.

## 6. Conclusão

Foram implementadas as etapas 2, 3 e 4 descritas no digrama da Figura 8. A etapa 1 que consiste em localizar a etiqueta na foto de entrada está ficando como atividade futura.

Como sugestão para implementação da etapa 1, pode ser usado um filtro de bordas, (*Sobel Filter*), e logo em seguida uma vetorização da imagem submetida ao mesmo a procura de segmentos de reta, para que possam ser encontrados os extremos da etiqueta. Com isso pode-se varrer a parte da etiqueta gerando a entrada da etapa 2.

Como atividade futura pode-se reescrever o código do componente MLP utilizado para construção da Rede Neural. O mesmo foi escrito em Delphi e tem código fonte aberto. Seria interessante reescrever o mesmo em Java ou em C++, visando portabilidade.

Como atividade futura, pode-se também adicionar à etapa 4, rotinas relacionadas à idéia de “Raciocínio Baseado em Casos”, (RBC). Essa idéia, no projeto, consistiria em guardar na base de dados imagens não reconhecidas e retreinar a Rede com elas. Terminada a etapa 1, o projeto estará pronto para ser testado no ambiente industrial.

## 7. Bibliografia

[Frontino, 2003], FRONTINO, L. et al. **Redes Neurais em Delphi**. Rio de Janeiro: Visual Books, 2003.

[Braga, Ludemire Carvalho, 2000], BRAGA, A. et al. **Redes Neurais Artificiais- Teoria e Aplicações**. Rio de Janeiro: LTC, 2000.

[Kovács, 1996], KOVÁCS, Z.L. **Redes Neurais Artificiais- Fundamentos e Aplicações**. São Paulo: Edição Acadêmica, 1996.

[Haikin, 2001], HAIKIN, S. **Redes Neurais – Princípios e Práticas**. 2 ed. Porto Alegre: Bookman, 2001.

[Nascimento e Yoneyama, 2002], NASCIMENTO JR, C.L. & YONEIAMA T. **Inteligência Artificial em Controle e Automação**. São Paulo: Edgard Blucher, 2000.

[Gonzales e Woods, 2000], GONZALES R.C. & WOODS R.E. **Processamento de Imagens Digitais**. São Paulo: Edgar Blucher, 2000.

[Gomes e Velho, 1994], GOMES J. & VELHO, L. **Computação Gráfica: Imagem**. Rio de Janeiro: SBM, 1994.

[WWW01], <http://www.din.uem.br/ia/neurais> acesso em 26/07/2005 às 2:55Hs