



PROCESSAMENTO DE IMAGENS INTELIGENTE USANDO REDES NEURAIIS ARTIFICIAIS

João Ricardo Bittencourt
Fernando Osório

RESUMO

O processamento de imagens é uma operação bastante comum na computação gráfica. Este processo tradicionalmente é implementado através de funções matemáticas, que transformam uma imagem original em uma imagem tratada através de equações conhecidas, usando algoritmos predeterminados. Seria interessante dotar o sistema da capacidade de executar o processamento de imagens sem conhecer a função transformadora, ou seja, a própria aplicação deveria ser capaz de aprender, automaticamente, como realizar o tratamento da imagem desejado pelo usuário. Sendo assim, o objetivo deste trabalho constitui-se em analisar o treinamento de redes neurais artificiais, do tipo *backpropagation*, na criação de filtros que possibilitam este processamento de imagens inteligente. Para viabilizar nosso objetivo foi desenvolvida uma ferramenta denominada Neuron Color capaz de: criar bases de exemplos; configurar filtros utilizando uma série de atributos, como por exemplo, o tamanho da matriz de contexto; armazenar em disco as imagens transformadas; aplicar os "filtros inteligentes"; e efetuar uma comparação estatística entre imagens, obtendo assim uma medida quantitativa da qualidade da imagem tratada. Além disso, durante o desenvolvimento do Neuron Color, criou-se um *framework*, denominado ANNeF, que pode ser utilizado no desenvolvimento de qualquer aplicação que necessite da implementação de redes neurais artificiais. Este *framework* permitiu a realização de alguns experimentos com resultados bastante satisfatórios. Em função destes resultados, pretendemos realizar novas experimentações, com o objetivo de aperfeiçoar ainda mais o processamento de imagens inteligente visando à melhoria da qualidade das imagens tratadas.

ABSTRACT

Image processing is a common operation in computer graphics. Traditional image processing methods use equations expressed through well defined algorithms in order to transform an original image into a new treated image. An interesting alternative approach should provide to users a system with the ability to create image processing tools without knowing the precise equations of each transformation function. This system should be able to learn automatically from examples, how to do an image transformation accordingly to the user's needs. The main goal of this work was to analyze artificial neural networks training, specifically based on *back-propagation* algorithm, used to create filters for image transformation. These filters allow us to obtain intelligent image processing tools. We developed an image processing software called Neuron Color. The main functions of this software are: create training examples sets; create neural filters based on different image attributes, as for example the size of context matrix; apply intelligent filters to the images; store treated images in hard disk; and to do statistical comparisons between two images (quantitative measure of treated image quality). Besides that, one more tool was created during Neuron Color development: ANNeF - A framework for artificial neural networks applications. We made practical experiments using Neuron Color and ANNeF, and the results we obtained were very interesting from the user's point of view. Based upon these initial results, we hope to extend our experiments aiming to improve our intelligent image processing approach in order to create better image processing filters.

1. INTRODUÇÃO

O Processamento de imagens é uma operação bastante comum na área da informática. Este processo tradicionalmente é implementado através de funções matemáticas, que transformam uma imagem original em uma imagem tratada através de equações conhecidas, usando algoritmos predeterminados. Uma série de tarefas podem ser executadas através desta área de estudo, das quais podemos citar: a aquisição e transmissão de imagens digitais; tratamento de imagens usando filtros, compressões e transformações; e extração e/ou identificação de seus componentes (reconhecimento de padrões) [\[OSO 00\]](#).

Como já foi dito, os métodos de processamento de imagens tradicionalmente requerem uma representação matemática e em alguns casos esta representação é bastante difícil de ser modelada. Desta forma outras técnicas alternativas devem ser estudadas e as redes neurais artificiais (RNA), uma abordagem moderna de inteligência artificial, podem ser uma destas alternativas. Pode-se dizer de forma sumária que as RNAs são capazes de classificar, generalizar e aprender funções desconhecidas. Um exemplo de uso desta técnica no processamento de imagens é a segmentação de imagens ecocardiográficas [\[DAH 99\]](#).

A principal motivação deste trabalho é resultante do interesse em estudarmos o comportamento do aprendizado de uma RNA usando o algoritmo de *backpropagation* [\[RUM 86\]](#) aplicada no processamento de Imagens. Espera-se assim obter filtros inteligentes capazes de executar um processamento de imagens sem conhecer a função transformadora, onde a própria aplicação é capaz de tratar uma imagem obtendo o efeito desejado pelo usuário, através de um processo de aprendizado a partir de exemplos.

O uso desta abordagem está sendo bastante satisfatório e tem-se mostrado bastante eficaz na resolução de problemas complexos de serem representados algorítmicamente. Realiza aproximações de funções de forma bastante satisfatória, criando resultados mais suavizados, considerando as características visuais da imagem. Desta forma, o resultado final é mais valorizado do que o conhecimento detalhado da função-alvo.

Portanto, o objetivo deste trabalho é demonstrar a utilização destes filtros neurais inteligentes no processamento de imagens, apresentando uma série de experimentos que foram efetuados e que podem demonstrar a eficiência desta abordagem.

Para isto o trabalho está organizado da seguinte forma: na seção 2 serão apresentados os principais conceitos e a metodologia tradicional de processamento de imagens; na seção 3 as redes neurais artificiais serão conceituadas, enfocando o algoritmo de *backpropagation*; na seção 4 será apresentado o ambiente *Neuron Color*, uma ferramenta livremente distribuída que permitiu a realização dos experimentos neurais; na seção 5 será enfocada a metodologia usada no processamento de imagens baseado nas RNAs; na seção 6 é apresentada uma métrica de avaliação das imagens processadas; na seção 7 alguns experimentos serão descritos; e por último, na seção 8, serão feitas as considerações finais.

2. PROCESSAMENTOS DE IMAGENS

A necessidade do processamento de imagens é oriunda dos anos 20 e seu principal objetivo é a melhoria da informação pictográfica para análise humana e processamento de dados pelas máquinas [\[GON 93\]](#).

Processar uma imagem consiste em modificar, analisar e manipular uma imagem digitalizada. E para o cumprimento destas ações, em geral, utilizam-se algoritmos baseados em princípios

matemáticos [FAC 93] e [GON 93].

Na [figura 1](#) está esquematizado o uso do processamento gráfico (computação gráfica e processamento de imagens). Quando temos um conjunto de dados, imagens sintéticas podem ser criadas através de técnicas de computação gráfica. Por outro lado, podemos ter imagens e através da visão computacional dados podem ser obtidos a partir destas imagens. Em ambos os casos é necessário efetuar algum tipo de processamento, incluindo o tratamento de imagens. Desta forma é possível concluir que o processamento de imagens pode ser aplicado após uma imagem ter sido criada por técnicas de computação gráfica ou pode ser usado inicialmente para facilitar o processo de Visão Computacional.

As áreas de maior aplicabilidade do processamento de imagens são a visão robótica, processos industriais que requerem inspeção visual, reconhecimento de padrões e reconstruções tridimensionais. Gráficas, editoras e a indústria cinematográfica também utilizam o tratamento de imagens para a melhoria do seu produto final.

Antes de iniciar a descrição do processamento de imagens propriamente dito, alguns conceitos básicos devem ser apresentados.

Primeiramente, uma imagem pode ser definida por uma função $f(x,y)$ sendo que seu valor representa uma intensidade de brilho naquela coordenada x e y . Na escala de cinza utiliza-se uma função para descrever a tonalidade (intensidade) de cinza, conforme pode ser visto na [figura 2](#). Os valores estão representados em um conjunto $[0;1]$, sendo que 0 representa a cor preta e o 1 é a cor branca, pode-se dizer que valor do ponto $p(1,1)$ da [figura 2](#) é 0,847.

Para as imagens coloridas a técnica empregada é similar. Geralmente utiliza-se o padrão RGB para descrever imagens coloridas [FOL 00], onde este modelo é baseado na capacidade natural dos seres humanos perceberem as cores. Ao contrário das imagens em tons de cinza, as coloridas necessitam de três funções para descrevê-las, uma para cada banda de frequência do espectro de cores que correspondem às bandas vermelha (R), verde (G) e azul (B). Nas imagens coloridas cada ponto é representado por estes três valores de intensidade – $r(x,y)$, $g(x,y)$ e $b(x,y)$.

Outro conceito importante é que cada ponto da imagem é chamado de *pixel*. O conjunto destes pontos organizados em uma matriz bidimensional representa uma imagem. Uma propriedade fundamental desta organização é a vizinhança. A [figura 3\(a\)](#) é um exemplo da vizinhança V_4 , dado um ponto qualquer, considera-se os quatro *pixels* vizinhos sem considerar as diagonais (*pixels* identificados pela cor escura). Já na [figura 3\(b\)](#) todos os oito *pixels* vizinhos ao ponto central são considerados. Esta forma de vizinhança é chamada de V_8 . O conceito de vizinhança é de suma importância para determinados filtros de processamento de imagem que necessitam utilizar o contexto em que está inserido certo *pixel*.

O processamento convencional de imagens é realizado por uma função $F(x)$ que irá realizar uma transformação de cada um dos *pixels* da imagem em novos *pixels*, onde estes sofrerão uma transformação segundo um algoritmo pré-especificado (veja [figura 4](#)). A principal desvantagem desta abordagem é que a realização de uma transformação sobre uma imagem deve ser feita de forma equacionada, baseando-se em algoritmos pré-definidos que nem sempre são capazes de reproduzir o efeito visual desejado pelo usuário. Alguns aplicativos encontrados no mercado utilizam um recurso de alto nível como as macros, que caracterizam-se pela definição de uma seqüência de comandos que permitem automatizar a reprodução de um determinado efeito, mas nem sempre o resultado final é completamente satisfatório. O nível de automatização também fica comprometido, pois em diversos momentos solicita-se a interação do operador para alterar ou incluir novos parâmetros. Muitas vezes o processo de tratamento de uma imagem é realizado através de um processo de "tentativa e erro", até que se encontre um efeito próximo ao desejado.

Nesta abordagem, a função de transformação deve ser conhecida e obviamente estar disponível e implementada pela aplicação. Na [figura 5](#) é apresentado um exemplo de processamento de

imagens convencional. Vamos supor um filtro que foi modelado conforme a equação 1, onde $o(x,y)$ é um *pixel* da imagem original e $t(x,y)$ é um *pixel* da imagem transformada resultante da divisão do valor da intensidade do *pixel* $o(x,y)$ por 2.

$$\text{Equação 1} \quad t(x,y) = \frac{o(x,y)}{2}$$

É possível perceber que a imagem transformada é mais escura que a imagem original, pois todos os valores de intensidade dos *pixels* originais foram reduzidos pela metade tendendo a 0 (cor preta). O exemplo, apresentado é bastante simples, mas auxilia no entendimento do mecanismo tradicional de processamento de imagens.

O que se deseja obter é a capacidade de executar este processamento de imagens sem conhecer a função transformadora, onde a própria aplicação fosse capaz de gerar a imagem desejada pelo usuário (veja figura 6). O usuário irá fornecer apenas exemplos de imagens originais e do resultado desejado do tratamento destas, deixando para o sistema a tarefa de "descobrir" uma maneira de realizar esta transformação (permitindo também que possamos repetir novamente esta transformação sobre outras imagens). [OSO 00]

Desta forma existe uma valorização dos atos criativos do autor, que pode criar um efeito baseado no seu gosto ou necessidade sem depender de uma equação pré-definida. Podemos citar, como exemplo, o caso de um cardiologista que gostaria de ressaltar, com cores específicas, detalhes existentes em uma foto ecocardiográfica. O médico pode realizar esta "operação de pintura" da imagem com naturalidade, o problema estaria em encontrar uma função que fosse capaz de colorir uma imagem, originalmente em tons de cinza, da mesma forma que um médico faria.

O propósito deste trabalho é apresentar um método de criação de filtros inteligentes capazes de efetuar tal processamento de imagens sem conhecer a função transformadora, baseando-se somente em exemplos. Podemos dizer que este processamento de imagens é feito por exemplos, através da analogia entre imagens [HER 01].

Na próxima seção será apresentada uma técnica de Inteligência Artificial, as redes neurais artificiais que permitem efetuar este processamento de imagens idealizado.

3. REDES NEURAIS ARTIFICIAIS

Os primeiros estudos e modelos bastante rudimentares sobre redes neurais artificiais (RNA) foram criados na década de 40 por McCulloch e Pitts. Foi somente na década de 80 que ocorreu um grande desenvolvimento científico na área, quando surgiram modelos modernos bastante discutidos [OSO 00], tais como *backpropagation* proposto por Rumelhart e Hinton [RUM 86] e os mapas de Kohonen (SOM) de Teuvo Kohonen [KOH 87].

Através de um modelo abstrato e simplificado dos neurônios humanos podemos desenvolver um simulador que seja capaz de classificar, generalizar e aprender funções desconhecidas [BIT 01]. Na figura 7 está esquematizado uma RNA formada por duas camadas, cinco neurônios de entrada (E1, E2, E3, E4 e E5) e três neurônios de saída (S1, S2 e S3).

Um dos modelos de aprendizado neural mais difundido na atualidade é o modelo denominado *backpropagation* [RUM 86]. Para que ocorra o aprendizado é necessário uma base de exemplos, conjunto de dados que descrevam os casos que serão apresentados para a RNA. O aprendizado do modelo *backpropagation* é conhecido como um aprendizado do tipo supervisionado, onde a rede aprende a responder de modo similar as respostas esperadas que são apresentadas junto com os casos. Os casos formam um conjunto de dados composto por um conjunto de entradas

mais as suas respectivas saídas desejadas.

Usando um processo iterativo, apresentamos a rede neural com os exemplos contidos na base de dados de aprendizado para que ocorra a adaptação dos pesos (simulam o reforço e a inibição das conexões sinápticas existentes nos neurônios biológicos). Desta adaptação de pesos que surge o aprendizado, e por consequência surge também a adaptação do comportamento da rede, que deverá aprender a responder aos estímulos de entrada de acordo com as respostas esperadas contidas nos exemplos que lhe foram apresentados.

4. NEURON COLOR

Antes de explicar como funciona a criação dos filtros neurais é necessário apresentar o software denominado *Neuron Color* [BIT 00], que foi construído para viabilizar o processamento de imagens inteligente usando redes neurais artificiais, conforme poderá ser visto na seção 5.

O *Neuron Color* é uma ferramenta desenvolvida utilizando a linguagem Java da Sun Microsystems [SUN 02] e é livremente distribuída para comunidade através da licença GNU GPL [FRE 91]. O sistema está dividido em três aplicações: o *Neuron Color Factory*, que permite a criação dos filtros neurais e a geração de bases de exemplos para efetuar o treinamento das RNAs; o *Neuron Color Viewer* responsável pela visualização de imagens, aplicação dos filtros neurais e armazenamento das imagens transformadas; e por último o *Neuron Color Statistical*, que permite comparar estatisticamente duas imagens através de uma métrica de avaliação, que está descrita na seção 6.

É importante destacar que durante a implementação do *Neuron Color* surgiu a necessidade de lidarmos com algoritmos de RNA. Para isto foi construído o ANNeF (*artificial neural networks framework*) [BIT 01], um conjunto de classes genéricas que oferece a implementação de algoritmos de RNA, incluindo o *backpropagation*. Desta forma ganhou-se flexibilidade pelo fato da reusabilidade das classes em outras aplicações de inteligência artificial. Desta forma o *Neuron Color* utiliza o ANNeF para modelar as RNAs dos filtros neurais.

5. CRIAÇÃO DOS FILTROS NEURAIS

Evidentemente que para a construção destes filtros é necessário a existência de um método. O objetivo desta seção é apresentar como esses filtros neurais são criados.

Dado um determinado problema de processamento de imagens, por exemplo, converter uma imagem colorida para tons de cinza, cria-se em qualquer ferramenta de edição gráfica quatro imagens, sendo que duas são originais ($train_0$ e $test_0$) e as outras duas são imagens transformadas através da manipulação do usuário que buscou aplicar um tratamento nestas imagens correspondente ao filtro que está sendo considerado. Com isto obtêm-se $train_T$ e $test_T$ a partir das imagens originais $train_0$ e $test_0$ respectivamente. No caso deste exemplo, existem duas imagens coloridas e duas em tons de cinza. Um par (original, transformado) é usado para treino e o outro para teste de generalização da RNA, como mostra a [figura 8](#).

Após as imagens terem sido criadas utiliza-se o *Neuron Color Factory*, para a criação de um novo filtro neural. Nesta etapa alguns parâmetros devem ser definidos.

Uso de Contexto (vizinhança): alguns algoritmos consideram um único *pixel*, outros utilizam uma vizinhança, para efetuar a transformação das imagens. O tamanho do contexto é definido pelo usuário, mas em geral é definido como uma matriz 3x3. Se o contexto está sendo usado, alguns parâmetros adicionais podem ser definidos, tais como média, variação, entropia e homogeneidade [DAH 99].

Entradas e Saídas: estas definições dependem das imagens de entrada e das imagens transformadas e determinam o número de neurônios de entrada e de saída de uma rede neural. Deve ser definido se o filtro utiliza imagens coloridas ou em tons de cinza e se as imagens geradas pelo filtro são coloridas ou em tons de cinza. A [tabela 1](#) apresenta o número de neurônios de entrada e de saída de uma RNA conforme o tipo de entrada e de saída. Por exemplo, se um filtro utiliza imagens coloridas, contexto e gera imagens em tons de cinza, a RNA teria 27 neurônios de entrada e 1 neurônio de saída.

Após o filtro ter sido parametrizado, as bases de exemplos podem ser criadas, ou seja, cada *pixel* das imagens será expresso em valores numéricos possíveis de serem processados. Duas bases devem ser geradas, uma para o treinamento da RNA e outra para o teste de generalização. O *Neuron Color Factory* também é usado para a geração das bases.

Ao término da criação dos arquivos, deve-se usar uma outra ferramenta para realizar o aprendizado da RNA. Nesta etapa podemos usar o *NevProp* [\[GOO 93\]](#), ou o ANNeF [\[BIT 01\]](#) que possui o BackpropSimulator, como sendo o simulador de redes neurais artificiais. Caso seja adotado o NevProp, usando qualquer editor de texto é possível fazer a configuração do arquivo de parâmetros usado por este simulador e a inserção do conteúdo dos dois arquivos de exemplos neste arquivo de configurações.

Inicia-se então o processo de aprendizagem, seja com o NevProp ou com o BackpropSimulator. Durante esta fase de treinamento o usuário deve intervir diretamente no que diz respeito ao término da simulação (adição de épocas) e verificação da queda do erro global.

Terminado este processo, salva-se o arquivo de pesos, predições e de resultados. Este último é analisado, caso a generalização não esteja suficientemente boa, os parâmetros do simulador de RNA são reconfigurados e se executa uma nova simulação. Este processo repete-se até obter uma boa generalização. É importante esclarecer que este processo de treinamento de RNAs é bastante empírico e as definições de parâmetros e da topologia da rede são o estado da arte das pesquisas de RNAs.

Com posse do arquivo de pesos, gerado pelo NevProp ao final do aprendizado, utiliza-se mais uma vez o *Neuron Color Factory*, para configurar a topologia da rede neural que será usada pelo filtro.

Finalizada esta etapa, podemos visualizar qualquer imagem no formato GIF, JPEG, JPG, TIF, TIFF, PNG e BMP [\[MUR 96\]](#) utilizando o *Neuron Color Viewer*. Através desta aplicação é possível utilizar os filtros neurais e armazenar em disco as imagens transformadas. Nesta aplicação é possível verificar visualmente a qualidade do resultado obtido pela aprendizagem. Para uma avaliação quantitativa é necessário utilizar alguma métrica de comparação de similaridade. Uma forma de medida numérica de avaliação será descrita na seção seguinte.

6. MÉTRICA DE AVALIAÇÃO DAS IMAGENS TRANSFORMADAS

O processo de avaliação das imagens geradas pelos filtros neurais é bastante complicado devido ao fator de subjetividade de um processo como este. Muitas vezes o uso de uma métrica para avaliação quantitativa da qualidade de um filtro pode significar um resultado negativo, mas através de uma inspeção visual este mesmo resultado pode ser extremamente positivo, pois a qualidade da imagem transformada depende do observador e do objetivo do processamento de imagens.

A métrica de avaliação proposta baseia-se na saturação, brilho e matiz dos *pixels* da imagem. Conforme foi dito na seção 2, cada *pixel* é representado por três funções RGB, mas esses *pixels* também podem ser expressos no modelo HSB (Hue, Saturation e Brightness) [\[FOL 00\]](#). Este modo de representação é mais intuitivo e está relacionado com a forma dos artistas definirem cores.

Cada cor pertence a um grupo, a matiz (H) indica a coloração do *pixel*, por exemplo, verde ou azul. A saturação (S) é a pureza desta cor, quanto menor a saturação mais próxima esta cor será do branco. Quanto ao brilho (B), quanto menor o valor mais próximo do preto será a cor.

Tendo como base duas imagens com as mesmas dimensões, uma gerada pelo método neural In e a outra imagem de referência Ir , calcula-se para cada um dos pontos a diferença absoluta dos valores de matiz, saturação e brilho, efetuando um somatório para cada um dos valores obtidos nesta diferença. Após esta etapa é possível calcular a diferença média de cada um dos valores. As equações para o cálculo da média das diferenças de matiz, saturação e brilho podem ser vistas nas equações 2, 3 e 4 respectivamente, onde h é a altura e w é a largura da imagem em *pixels*.

$$d_H = \frac{\sum_{y=0}^h \sum_{x=0}^w |In_H(x,y) - Ir_H(x,y)|}{h \cdot w} \quad \text{Equação 2}$$

$$d_S = \frac{\sum_{y=0}^h \sum_{x=0}^w |In_S(x,y) - Ir_S(x,y)|}{h \cdot w} \quad \text{Equação 3}$$

$$d_B = \frac{\sum_{y=0}^h \sum_{x=0}^w |In_B(x,y) - Ir_B(x,y)|}{h \cdot w} \quad \text{Equação 4}$$

Como os valores HSB são expressos no intervalo [0;1], logo as médias das diferenças também serão expressas neste mesmo intervalo. É importante destacar que as três medidas devem ser sempre consideradas no caso de imagens coloridas. Se a imagem está representada em tons de cinza, somente o brilho deve ser considerado.

Esta métrica é bastante flexível e pode se ajustar à questão da subjetividade. O usuário pode definir a priori quais os valores máximos que a matiz, a saturação e o brilho podem variar. Caso d_H , d_S e d_B sejam inferiores aos valores máximos aceitos, que foram preestabelecidos, pode ser dito que a imagem neural (In) assemelha-se bastante à imagem de referência (Ir).

Para efetuar estes cálculos utiliza-se o *Neuron Color Statistical* que permite efetuar esta comparação entre duas imagens.

Usando este procedimento, alguns experimentos foram efetuados e serão descritos na próxima seção.

7. EXPERIMENTOS

O objetivo desta seção é demonstrar alguns resultados práticos da aplicação dos filtros neurais. Serão demonstrados a conversão de uma imagem colorida para tons de cinza, detecção de contornos e correção de imagens desfocadas.

Nestes exemplos considerou-se um erro médio máximo de 0,019531 nos treinamentos. Este valor representa uma variação de 5 pontos em cada função do modelo RGB. Para os valores máximos das médias das diferenças, os seguintes valores foram definidos: $d_H=0,00278$; $d_S=0,01$; $d_B=0,01$.

7.1 Transformação de imagens coloridas para tons de cinza

Consiste em representar uma imagem colorida em tons de cinza. Em uma imagem colorida, usando o modelo RGB, cada *pixel* pode assumir 16 milhões de cores. No caso das imagens em tons de cinza, cada *pixel* pode assumir 256 níveis de cinza. Portanto, a função de transformação deve mapear estas 16 milhões de cores para os 256 níveis de cinza.

Em geral esta função é do tipo ponto-a-ponto, ou seja, não precisa considerar a vizinhança para efetuar a transformação. Neste experimento foram criados um filtro neural sem usar contexto e uma outra variante que considera uma matriz de contexto 3x3.

O filtro neural para efetuar a transformação sem o uso de contexto utilizou uma Rede Neural Artificial com 3 camadas, 3 neurônios de entrada, 6 neurônios na camada oculta (intermediária) e 1 neurônio na camada de saída. Foram usados 2.500 exemplos para treino e 2.500 exemplos para o teste de generalização. Com 17.500 épocas foi possível obter uma rede neural com acerto de 73,16% dos exemplos da base de teste de generalização e um erro médio de 0,01058.

No caso do filtro neural que utiliza contexto, este usou uma RNA com a seguinte topologia: 3 camadas, 27 neurônios de entrada, 3 neurônios na camada oculta e 1 neurônio na camada de saída. Foram utilizados 2.304 exemplos para o treinamento e o mesmo número de exemplos para o teste de generalização. A RNA teve um erro médio de 0,00667 com 89,89% de acertos dos exemplos para generalização e foram necessárias 21.300 épocas.

A [figura 9](#) é uma imagem obtida pelos métodos tradicionais usando funções matemáticas predeterminadas, já as [figuras 10](#) e [11](#) foram obtidas através dos filtros neurais, sendo que o filtro que gerou a [figura 10](#) não usou contexto e o filtro que gerou a [figura 11](#) utilizou. Perceba que a [figura 11](#) possui uma linha de contorno na imagem, isto ocorre em função do uso de contexto. É importante destacar que estas imagens evidentemente não são as mesmas imagens que foram usadas para o treinamento, desta forma estes resultados são oriundos da generalização da RNA. A imagem original usada neste experimento foi obtida no livro *The big book of painting nature in watercolor*, de Ferdinand Petrie & John Shaw, apud Hertzmann, Jacobs, Oliver et al [\[HER 01\]](#).

Aplicando a métrica descrita na seção 6, considerando somente o brilho pelo fato da imagem ser em tons de cinza, no filtro neural sem contexto e com contexto obteve-se os seguintes valores respectivamente: $d_B = 0,0094$ e $d_B = 0,0723$. Conforme os valores máximos das diferenças das médias foram pré-definidos, é possível afirmar que o filtro neural que não utiliza contexto produziu uma imagem extremamente satisfatória, ao contrário do filtro que utiliza contexto. É possível perceber que mesmo se o resultado não foi plenamente satisfatório do ponto de vista matemático, a aparência visual da imagem obtida pelo filtro que utiliza contexto é bastante interessante e adequada para diversas aplicações.

7.2 Detecção de Contornos

Baseia-se em encontrar um conjunto de *pixels* que supostamente sejam o contorno dos objetos presentes na imagem. Um algoritmo eficiente na detecção de contornos é a codificação de Freeman [\[FAC 93\]](#), que leva em conta sete direções distintas em relação a um *pixel* e através de uma busca iterativa tenta encontrar pontos pertencentes ao contorno. Além da codificação de Freeman existe o método da transformada de Hough [\[FAC 93\]](#), que detecta curvas analíticas, como elipses e circunferências. Ambos os métodos envolvem contexto.

O filtro neural configurado utiliza uma RNA com 2 camadas, formada por 27 neurônios de entrada

e 1 neurônio de saída. As bases para o treinamento e para o teste de generalização possuem 2.304 exemplos. O treinamento consumiu 22.700 épocas e a RNA obteve 94,92% de acertos da base de exemplos com erro médio de 0,06249.

Este experimento é bastante interessante pela questão relativa à subjetividade e ao resultado da avaliação, usando métodos matemáticos. Em relação à média das diferenças, o valor obtido é 0,11477; portanto, a imagem gerada pelo filtro neural não é muito semelhante à imagem gerada pelos métodos tradicionais. Através da inspeção visual também é possível perceber estas diferenças, onde constatamos que o resultado da aplicação da função aprendida difere de certa forma do resultado presente na base de exemplos de aprendizado.

A [figura 12\(b\)](#), gerada por métodos matemáticos, é bastante simplificada, praticamente destaca as partes principais do rosto como a forma da face, cabelo, olhos, nariz e boca. Já a [figura 12\(c\)](#) é uma imagem gerada pelo filtro neural e possui mais detalhes, principalmente os olhos. O que vai determinar o nível de qualidade de cada método são os objetivos da aplicação onde está inserido este processamento de imagens. Se em uma aplicação real prioriza-se um maior detalhamento, evidentemente que o método neural é melhor que o tradicional. Ao contrário, se a aplicação necessita de resultados mais simplificados, o outro filtro poderia ser mais adequado.

7.3 Correção de Imagens Distorcidas

Muitas vezes é possível encontrar imagens que perderam informação pictográfica por estarem desfocadas. Seria interessante construir um filtro que fosse capaz de restaurar esta imagem desfocada, melhorando a sua nitidez.

Para isto foi construído um filtro neural que utiliza uma matriz de contexto 3x3 e uma RNA com a seguinte topologia: 3 camadas, 27 neurônios de entrada, 6 neurônios ocultos e 3 neurônios de saída. Foram necessárias 23.920 épocas e duas bases de exemplos com 2.304 casos cada uma. Ao término do aprendizado, a RNA teve 35,14% de acertos com um erro médio de 0,04555.

A [figura 13\(a\)](#) é a imagem desfocada, percebe a falta de detalhamento dos ponteiros do relógio, das torres e dos detalhes da construção. A [figura 13\(b\)](#) é a imagem original e também o alvo do filtro neural. A [figura 13\(c\)](#) é a imagem restaurada pelo filtro neural. É importante destacar que originalmente estas imagens eram coloridas [\[PRE 02\]](#) e as médias das diferenças são as seguintes: $d_H = 0,12036$; $d_S = 0,0777$; $d_B = 0,03425$. Considerando a imagem em tons de cinza, a média das diferenças de brilho é 0,02858. Em ambos os casos, matematicamente os resultados são insatisfatórios em relação aos valores preestabelecidos, mas através da inspeção visual é possível verificar que existe uma boa restauração da imagem desfocada, conforme podemos constatar na [figura 13\(c\)](#).

8. CONSIDERAÇÕES FINAIS

Ao término deste trabalho é possível perceber que os filtros neurais geram imagens bastante satisfatórias no aspecto visual, sendo que muitas vezes os resultados de uma avaliação quantitativa da qualidade são insatisfatórios. É bastante difícil avaliar a qualidade do efeito visual, pois lidamos com a questão da subjetividade, para um indivíduo a imagem resultante pode ser excelente e para outros o efeito pode ser simplesmente razoável.

Mas este problema da subjetividade da avaliação dos resultados pode ser considerado por outro ângulo. O principal objetivo deve ser o de respeitar a individualidade do usuário, possibilitando que este crie um processamento de imagens de acordo com sua preferência pessoal ou a necessidade da aplicação. Esta é justamente a motivação deste trabalho, que destaca a questão do usuário poder produzir um efeito de acordo com seus desejos sem preocupar-se com a definição algorítmica da função transformadora.

A métrica da média das diferenças (matiz, saturação e brilho) pode ser uma tentativa de mensurar quantitativamente o quanto que uma imagem assemelha-se a outra. Outra propriedade importante desta métrica é a flexibilidade na determinação dos valores máximos de diferença entre imagens que são aceitáveis, e desta forma, considerar uma imagem insatisfatória dependerá principalmente da aplicabilidade da imagem gerada pelo método neural. A desvantagem desta métrica é a questão relativa à determinação destes valores máximos, que é feita de modo totalmente empírico.

Também é possível constatar a eficiência do modelo neural do tipo *backpropagation*, no que diz respeito à aprendizagem de funções matemáticas desconhecidas. Sendo que os maiores problemas desta técnica são: a determinação da topologia da RNA e a fase de aprendizado, que pode consumir um longo tempo.

As possibilidades de trabalhos futuros referem-se ao constante aperfeiçoamento da ferramenta *Neuron Color* e o método de criação dos filtros neurais, principalmente no que se refere à definição da topologia das RNAs. Além disso, seria de extrema importância validar os filtros neurais utilizando um domínio de aplicação, como por exemplo, em aplicações de visão artificial e de segmentação de imagens ecocardiográficas.

9. REFERÊNCIAS BIBLIOGRÁFICAS

- [BIT 00] BITTENCOURT, J. R.; OSÓRIO, F. S. [Adaptive filters for image processing based on artificial neural networks](#) In: [SIBGRAPI 2000. Brazilian Symposium on Computer Graphics and Image Processing, 13. Proceedings](#). Los Alamitos: IEEE Computer Society, 2000. p.336.
- [BIT 01] BITTENCOURT, J. R. OSÓRIO, F. ANNeF – [Artificial neural networks framework: uma solução software livre para o desenvolvimento, ensino e pesquisa de aplicações de inteligência artificial Multiplataforma](#). In: [Workshop sobre Software Livre. 2. Anais...](#) Porto Alegre: Sociedade Brasileira de Computação, 2001. p.13-16.
- [DHA 99] DAHMER, A. PICCOLI, L. SIQUEIRA, M. et al. [Aplicação de duas abordagens neurais na segmentação de imagens ecocardiográficas](#). In: [CONGRESSO NACIONAL DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 19. Anais...](#) Rio de Janeiro, 1999. v.4, p.80-90.
- [FAC 93] FACON, J. *Processamento e análise de imagens*. Córdoba: Universidade Nacional de Córdoba, 1993. 198p.
- [FBI 02] FBI – [Federal Bureau of Investigation](#). Acesso em: 02 jul. 2002
- [FOL 00] FOLEY, J.A. *Computer graphics: principles and practice*. 2.ed. Reading: Addison-Wesley, 2000. 1175p.
- [FRE 91] FREE SOFTWARE FOUNDATION. GNU General Public License, 1991. Acesso em: 23 maio 2002
- [GON 93] GONZALES, R. C.; WOODS, R. E. *Digital image processing*. Reading: Addison-Wesley Publishing, 1993. 716p.
- [GOO 93] GOODMAN, P.; ROSEN, D.; PLUMMER, A. [NevProp: \(users manual\)](#) . University of Center for Biomedical Modeling Research, 1993.
- [HER 01] HERTZMANN, A.; JACOBS, C. E.; OLIVER, N. et al. [Image analogies](#). In: [SIGGRAPH 2001. Proceedings](#). Los Angeles: ACM, 2001

- [KOH 87] KOHONEN, T. *Self-organization and associative memory*. Berlin: Springer-Verlag, 1987. (Series in Information Science).
- [MUR 96] MURRAY, J. D. *Encyclopedia of graphics file formats*. 2.ed. Bonn: O'Reilly & Associates, 1996. 1116p.
- [OSO 00] OSÓRIO, F. S.; BITTENCOURT, J. R. [Sistemas inteligentes baseados em redes neurais artificiais aplicados ao processamento de imagens](#). In: WORKSHOP DE INTELIGÊNCIA ARTIFICIAL, 1., 2000, Santa Cruz do Sul.
- [PRE 02] PREFEITURA MUNICIPAL DE CANELA/RS. [Canela](#). Acesso em: 02 jul. 2002.
- [RUM 86] RUMELHART, D.; HINTON, G.; WILLIAMS, R. Learning internal representations by error propagation. In: RUMELHART, D.; MACCLELLAND, J.L. *Parallel distributed processing: explorations in the microstructure of cognition*. Cambridge: MIT Press, 1986. v.1.
- [SUN 02] [SUN MICROSYSTEMS. Java](#). Acesso em: 30 jun. 2002

AUTORES

João Ricardo Bittencourt [CURRÍCULO LATTES](#)

[EMAIL](#) | [HOMEPAGE](#)

Fernando Osório [CURRÍCULO LATTES](#) | [GRUPO DE PESQUISA](#)

[EMAIL](#) | [HOMEPAGE](#)

Mestrado em Computação Aplicada – Centro de Ciências Exatas e Tecnológicas

Universidade do Vale do Rio dos Sinos (UNISINOS)
Caixa Postal 275 – 93.022-000 – São Leopoldo – RS – Brasil

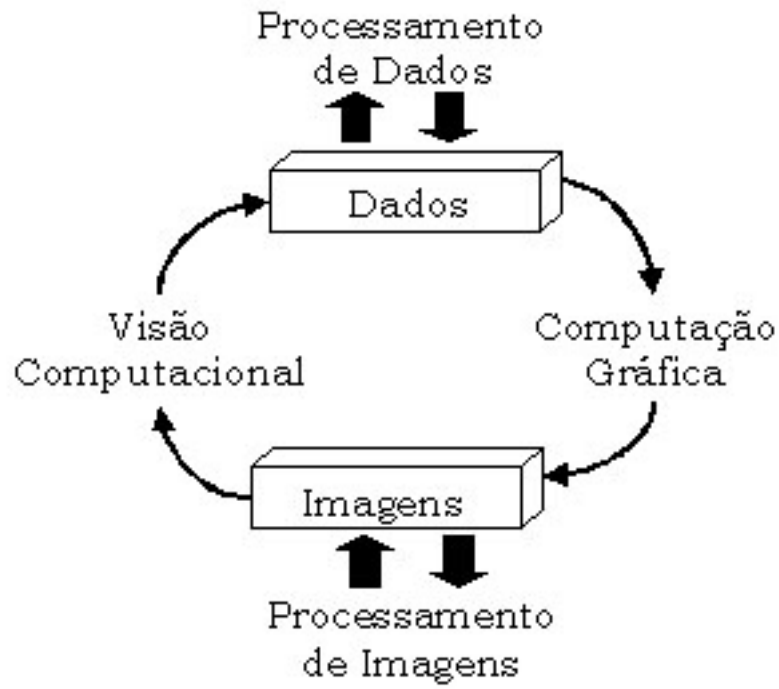


Figura 1 – Relação entre o processamento de imagens e a computação gráfica

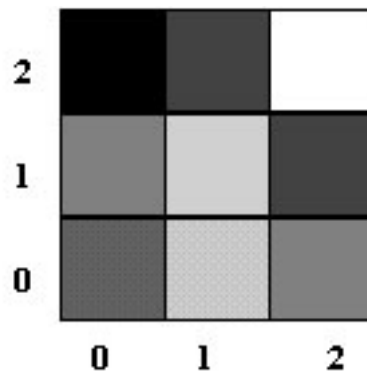
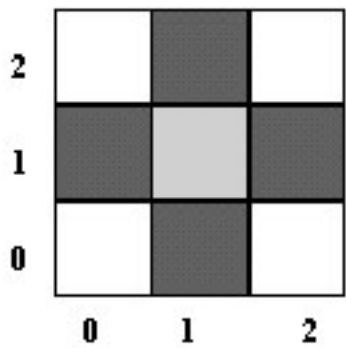
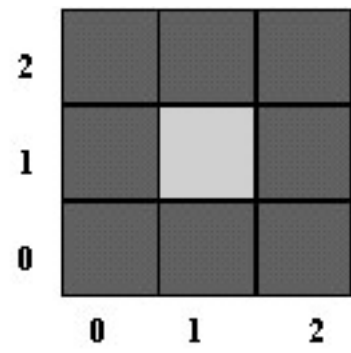


Figura 2 – Representação de uma imagem



(a)



(b)

Figura 3 – Conceito de vizinhança V_4 e V_8

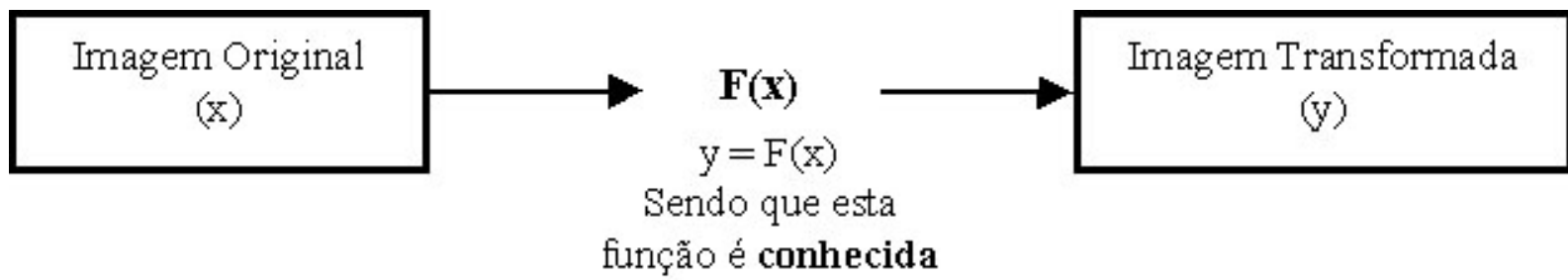
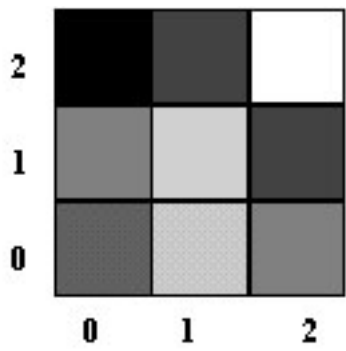
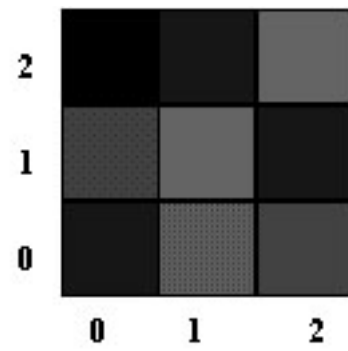


Figura 4 – Processamento de imagens convencional



$o(x,y)$

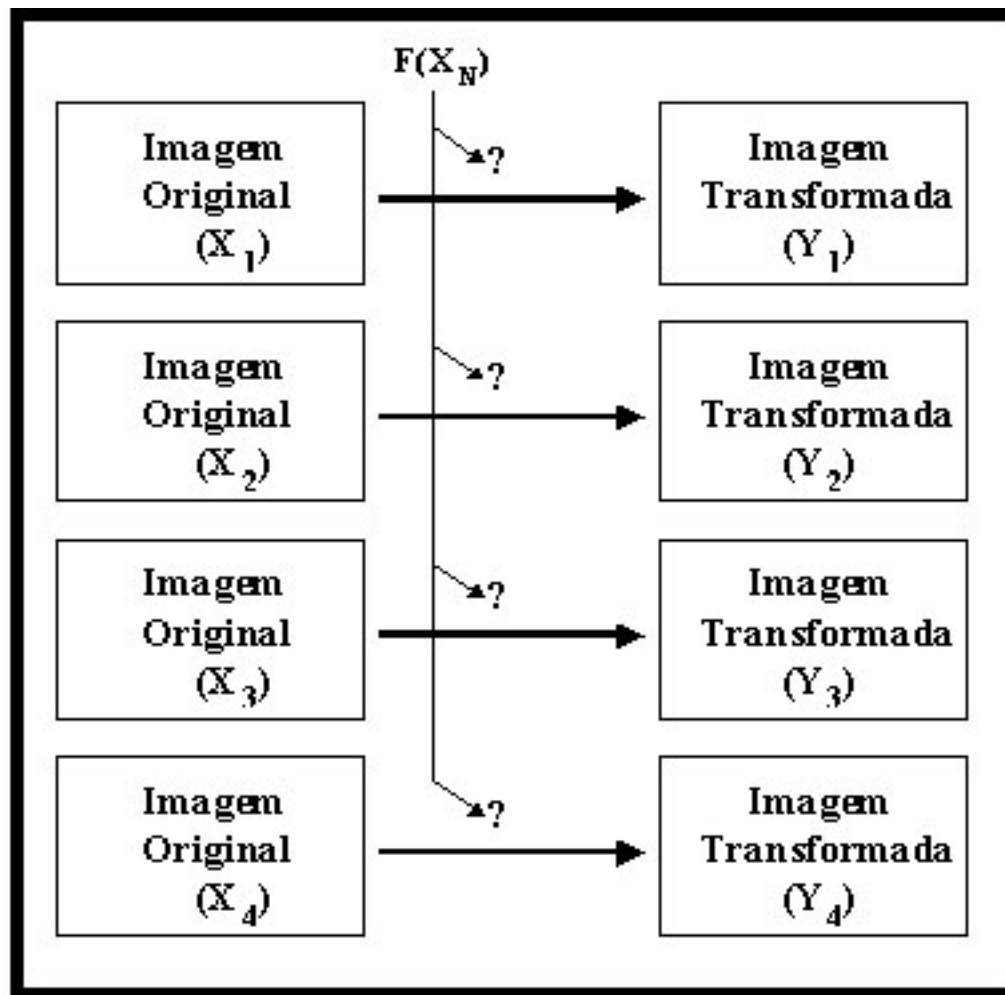
Imagem Original



$t(x,y)$

Imagem Transformada

Figura 5 – Exemplo de um processamento de imagens convencional



X_N e Y_N são fornecidas pelo usuário

Figura 6 – Processamento de imagens baseado em exemplos, sem conhecer a função transformadora [OSO 00]

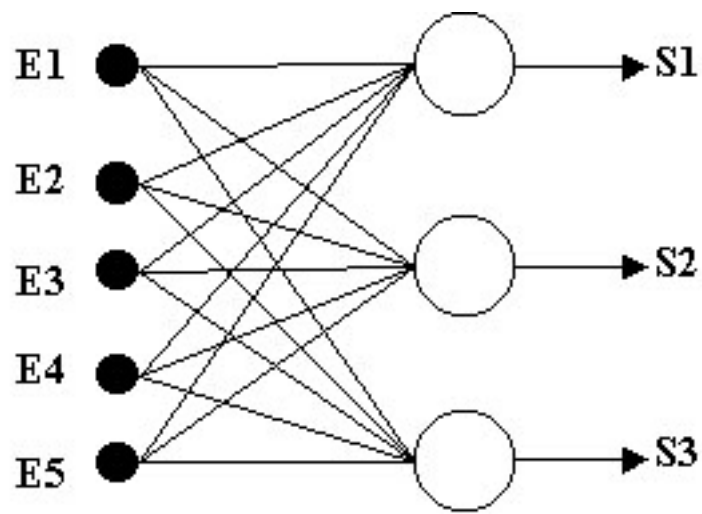


Figura 7 – Exemplo de uma rede neural artificial

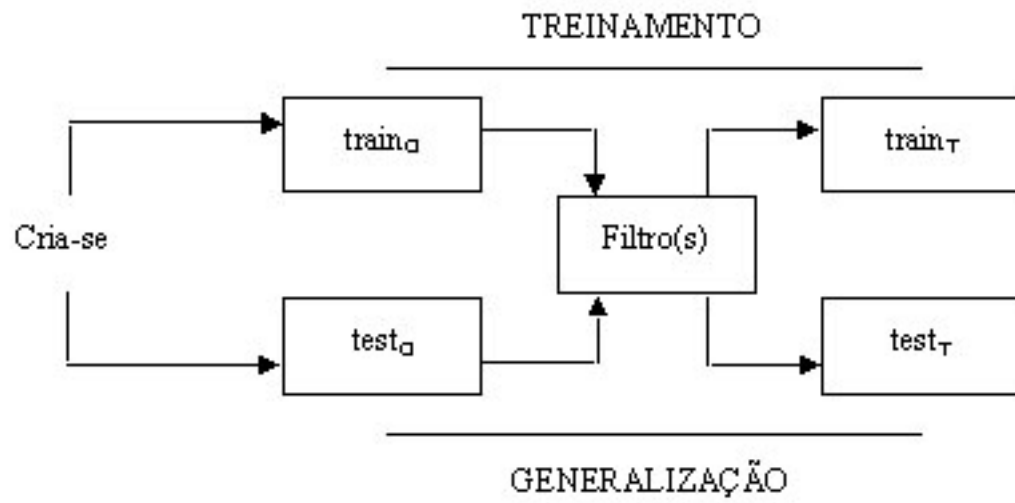


Figura 8 – Esquema da criação de imagens para o treinamento das RNAs

Tabela 1 – Número de entradas e saída conforme o tipo de entrada e saída dos filtros

Tipo de entrada/saída	Número de entradas		Número de saídas
	<i>Com contexto</i>	<i>Sem contexto</i>	
RGB / Cinza	27	3	1
RGB / RGB	27	3	3
Cinza / RGB	9	1	3
Cinza / Cinza	9	1	1



Figura 9 – Imagem em tons de cinza obtida pelos métodos tradicionais



Figura 10 – Imagem em tons de cinza obtida usando um filtro neural *sem* contexto



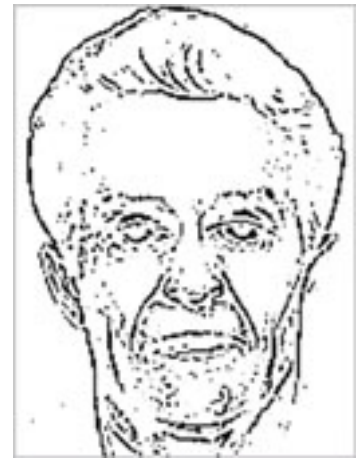
Figura 11 – Imagem em tons de cinza obtida usando um filtro neural *com* contexto



(a)

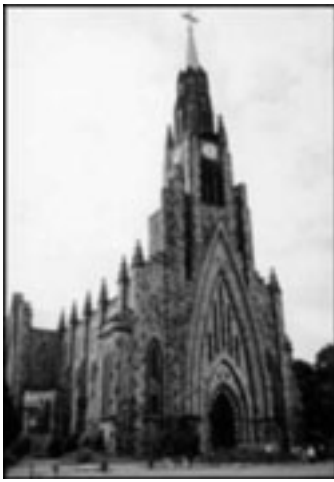


(b)



(c)

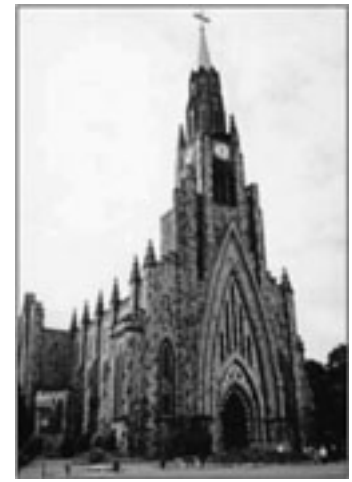
Figura 12 – Detecção de bordas. (a) Imagem original [FBI 02]; (b) Imagem obtida com métodos matemáticos; (c) Imagem obtida usando um filtro neural com contexto



(a)



(b)



(c)

Figura 13 – Correção de imagens. (a) Imagem desfocada (b) Imagem original; (c) Imagem restaurada com método neural.

[XIII Brazilian Symposium on Computer Graphics and Image Processing \(SIBGRAPI'00\)](#)

October 17 - 20, 2000

Gramado, Brazil

p. 336 • [Adaptive Filters for Image Processing Based on Artificial Neural Networks](#)



J.R. Bittencourt, F.S. Osorio , Univ. do Vale do Rio dos Sinos, Sao Leopoldo, Brazil

The paper presents our research in image processing (filters used for edge detection, color transformation, and distortion correction) using artificial neural networks. We have developed a tool, Neuron Color, in order to carry out these experiments and improving our adaptive methods.

Index Terms- adaptive filters; adaptive filters; image processing; artificial neural networks; edge detection; color transformation; distortion correction; Neuron Color

Copyright (c) 2000 Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

The full text of sibgraphi is available to members of the IEEE Computer Society who have an [online subscription](#) and an [web account](#).

This site and all contents (unless otherwise noted) are Copyright ©2002 , Institute of Electrical and Electronics Engineers, Inc. All rights reserved