Detecção Facial com Redes Neurais LVQ

André Costa

Centro de Informática – UFPE

Recife, Brasil

afc3@cin.ufpe.br

Mário Barbosa

Centro de Informática – UFPE

Recife, Brasil

mbaj@cin.ufpe.br

*Abstract*— A detecção facial é uma área importante no campo de processamento digital. Por possuir diversas aplicações práticas torna-se uma área de bastante interesse comercial. O objetivo deste trabalho é apresentar uma nova técnica de detecção facial a partir de imagens estáticas utilizando uma rede neural LVQ (Learning Vector Quantization). A implementação da técnica se inicia com o processamento da base de treinamento, extraindo suas características com um filtro Gabor, esses dados então são treinados em uma rede neural LVQ até atingir a convergência. Finalizado o treinamento da rede, o classificador é utilizado através de uma varredura piramidal para detectar as faces.

*Index Terms*—redes neurais, lvq, detecção facial, computação visual,filtro de gabor.

# Introdução

Sistemas de detecção facial atualmente são bem difundidos, e podem ser encontrado facilmente em sistemas embarcados como câmeras digitais e smartphones. Um sistema de detecção facial para obter taxas relevantes de acerto precisa lidar com diversos problemas como múltiplas poses, iluminação não controlada e oclusão parcial por exemplo.

Para o uso em sistemas embarcados e de tempo real, é necessário uma técnica de classificação rápida, sendo uma das principais características de redes neurais artificiais, apesar de possuir um processo custoso para o treinamento, porem com uma rápida classificação.

# Descrição da Técnica

As imagens são inicialmente processadas para extrair suas características. A técnica se baseia no filtro de Gabor, que consiste em um filtro linear bastante utilizado para a detecção de bordas. A frequência e a representação da orientação dos filtros de Gabor são similares ao sistema visual humano, e tem sido uma excelente ferramenta para representação de textura e discriminação [4].

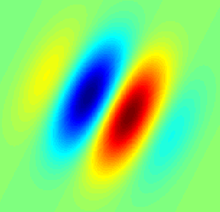


Fig 1- Filtro de Gabor de 2 dimensões

Em seguida, essas características são passadas como entrada de uma rede neural LVQ (*Learning Vector Quantization*).

A LVQ é um método de classificação de padrões que utiliza uma rede de camada única de neurônios artificiais posicionadas em duas ou mais dimensões para classificar as informações apresentadas em padrões ou classes distintas [6]. A LVQ utiliza aprendizagem do tipo supervisionado utilizando um conjunto de dados de entrada conhecidos [3].

Em uma rede LVQ cada vetor de entrada é apresentado a todos os neurônios. Um processamento de comparação é realizado em cada neurônio a fim de encontrar o neurônio que melhor responde o padrão, então o neurônio vencedor tem seus pesos atualizados seguindo as seguintes regras, caso o padrão que o neurônio representa seja da mesma classe do padrão de entrada, os pesos são atualizados para melhorar a resposta ao padrão, caso contrário são atualizados pra aumentar a diferença entre eles. Esse processo é descrito como uma aprendizagem por processo de competição, onde se sobressai o neurônio vencedor. Esse processo é repetido por diversos ciclos, e em cada ciclo a taxa de aprendizagem é reduzida para controlar a taxa de esquecimento da rede [3].

Após realizado o treinamento da rede neural, o sistema encontra-se pronto para ser utilizado, recebendo como entrada uma imagem que antes de tudo é processada para intensidades em escala de cinza. Então a imagem e varrida por uma algoritmo de multiscale piramidal que processa cada sub-imagem, extraindo as suas características com o filtro de Gabor descrito anteriormente e inserindo as características na rede neural para obter a classificação da sub-imagem. A rede então apenas responde se a sub-imagem dada na entrada é ou não uma face [12].

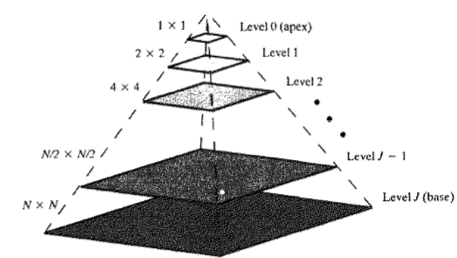


Fig 2 – Multiscale Piramidal

# Preparação da base de treinamento

Como já foi descrito, as imagens de treinamento não podem ser diretamente colocada para treinamento, o processo requer um pré-processamento antes.

Para deixar o sistema invariante a iluminação, as imagens de treinamento são processados por um método de filtragem baseado nos filtros de Gabor [4]. Inicialmente é gerado um conjunto de filtros de Gabor adequados para as frequências espaciais capaz de extrair características a partir de imagens afetada por diversas formas de iluminação. Os parâmetros utilizados por cada filtro é estimado a partir de *templates* de faces, empregando-se a transformada de Fourier. Cada filtro gera uma imagem filtrada que quantifica a frequência espacial definida no filtro.

A base de treinamento é processada sendo convertida para tons de cinza e reduzidas para o tamanho de 27 por 18 pixels. Em seguida é aplicado o filtro de Gabor a fim de extrair características invariante a iluminação. Com essas características, cada imagem é representada por um vetor de tamanho 2160.

Após serem processadas, a saída do processamento do filtro de Gabor estão preparadas para serem inseridas na rede para o treinamento.

A base de dados de treinamento e testes foram oriundas da união entre três fonte de base de dados, duas com faces para amostras positivas e uma sem faces para amostra negativa [7, 8, 9]. Todas as faces positivas eram frontais, com diversas amostras de diferentes iluminações.

Para a conversão das imagens de treinamento e teste para níveis de cinza, e posteriormente a redução para os tamanhos especificados, foram utilizadas duas ferramentas, o Automator [11], que se trata de uma ferramenta de automação nativa do sistema operacional Mac que foi utilizada para selecionar as imagens e reduzi-las para o tamanho especificado e renomea-las de forma sequencial para a leitura do Matlab, depois foi utilizada a ferramenta ImageMagick [10] para converte-las para intensidades em tons de cinza.

# Implementação da Técnica

Para a implementação foi utilizado o software matemático Matlab em conjunto com o framework fornecido pela própria MathWorks, o Neural Networls Toolbox [1], e um algoritmo de filtro de Gabor [4].

Foram testados vários algoritmos de redes auto-organizáveis provenientes da Toolbox [2], mas foi preferível trabalhar com a Neural Networks Toolbox nativa do Matlab pela estabilidade e confiabilidade.

Após diversas tentativas, a definição final do esquema de pré-processamento utilizando foi o Filtros de Gabor que conseguiu a convergência das características da base deixando-as invariante a iluminação. Por fim o algoritmo LVQ foi reimplementado em C, melhorando notavelmente o tempo de processamento necessário e incluindo uma série de otimização que serão descritas a seguir.

# Algoritmo

## Filtro de Gabor

Para o pré-processamento das imagens, os filtros de Gabor são inicialmente criados utilizando a ferramenta Matlab. Em nossa implementação foram utilizados 5 escalas em 8 ângulos cada, totalizando 40 filtros. Após criados, esses filtros são transformados para o domínio de frequência utilizando a Transformada de Fourier e armazenados. Os filtros são criados utilizando o tamanho de 32 x 32 pixels.

## Pré Processamento

Todas as imagens da base de treinamento e testes são inicialmente pré-processadas da seguinte forma:

### As imagens são convertidas para tons de cinza.

### Cada imagem é reduzida para a resolução de 27 x 18.

### Cada imagem é transformada para o domínio de frequência com o tamanho 32 x 32.

### As imagens no domínio de frequência é multiplicada por cada um dos 40 filtros de Gabor.

### O resultado é convertido para o domínio do espaço pela transformada de Fourier inversa no tamanho de 27 x 18.

### O resultado de cada vetor é normalizado para o intervalo [-1, 1]

### Uma matriz de 135 x 144 é criada a partir das imagens multiplicada pelos filtros de Gabor com 5 escalas diferentes com 8 ângulos cada.

### A matriz resultante é processada extraindo um terço dos dados removendo linhas e colunas de forma alternada resultando em uma matrix 45 x 48

### Essa matriz é convertida em um vetor com 2160 características

A base de treinamento possui 906 entradas, onde 507 são padrões positivos e 399 são padrões negativos.

A base de testes possui 324 entradas, sendo 183 padrões positivos e 141 padrões negativos.

## Inicialização da Rede Neural LVQ

A rede neural LVQ é inicializada com 30 neurônios na camada de processamento, 2160 na camada de entrada e apenas um na camada de saída, respondendo apenas sim ou não para o candidato a face inserido na rede.

Para melhorar a resposta da rede duas otimizações foram realizadas:

A primeira consiste em dividir a quantidade de neurônios da camada de processamento pela proporção das classes dos padrões da base de treinamento. Como a base de treinamento tem a proporção de 507 faces positivas para 399 faces negativas, a divisão dos neurônios que responde a cada classe ficou 16 para o padrão positivo e 14 para o padrão negativo.

A segunda otimização foi inicializar o pesos da rede neural de forma aleatória porém controlada, o valor máximo e o valor mínimo de cada propriedade dos padrões foram extraídos, então o valor aleatório era calculado de forma a estar inserido neste intervalo, impedindo que valores muito distantes do espaço de características seja eleito demasiadamente para representar os padrões por ser um dos poucos a representa-los, ou nunca vença uma competição por estar muito distante.

## Fase de Treinamento

Os 906 padrões de treinamento com 2160 características cada são inseridos na rede neural. Utilizando uma taxa de aprendizagem inicial de 0.2 e final de 0.01, a rede é interada por 1000 épocas modificando os 30 pesos afim de aproxima-los aos padrões que desejam representar.

O arquivo com os dados de peso da rede é armazenado para a fase de testes e uso da rede neural.

## Fase de Testes

A base de testes com 324 padrões já pré-processados são inseridas na rede neurais. A rede neural retorna a classificação de cada padrão baseado na classe do neurônio mais próximo do padrão. As classes resultantes dos padrões são comparadas com a real classe de cada padrão resultando na taxa de acerto da rede neural. Com os parâmetros da rede neural descrita conseguimos a taxa de acerto de 86,11%.

## Uso da Rede

Para utilizar o algoritmo em base de dados conhecidas, é necessário a capacidade do algoritmo de dado uma imagem, apontar a área quadrada de onde as faces se encontram. Para isso foi implementado o algoritmo de Piramidal Scale, escalando a imagem e 5 tamanhos diferentes e fazendo a varredura a cada 5 pixels no sentido horizontal e vertical.

O algoritmo implementado não possui uma série de otimização utilizadas no estado da arte, como só realizar a varredura em regiões de alta probabilidade de ocorrência da face, e grandes pulos da janela de varredura quando em janelas próximas uma face já foi encontrada, resultando em uma péssima performance, e diversas falhas na definição das regiões de faces, inserindo mais de um quadrado por face.

# Experimentos de Avaliação Técnica

O Neural Networks Toolbox possui várias ferramentas de avaliação. O gráfico de performance de erro por época de treinamento foi utilizando para acompanhar a evolução da convergência da Rede Neural.

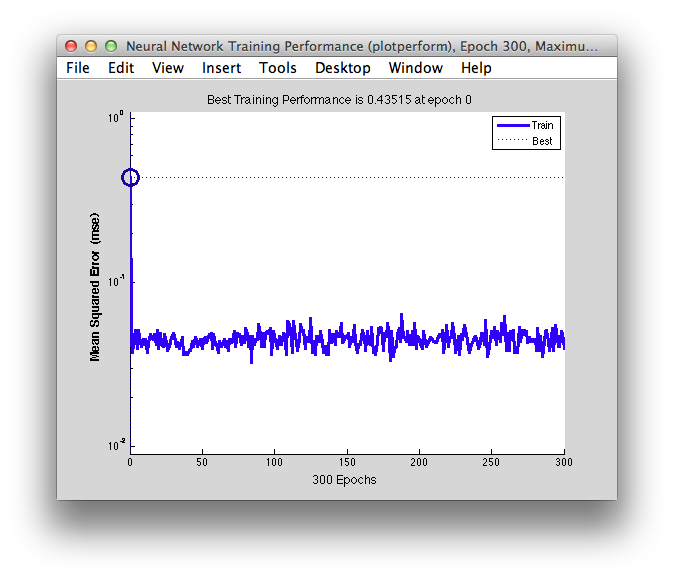


Fig 3 – Convergência da Rede Neural

A taxa de acerto da rede neural guiava as modificações dos parâmetros como a quantidade de neurônios da camada de processamento, a taxa de aprendizagem inicial e final, e a quantidade de épocas necessárias para atingir convergência.

Utilizando o sistema para detectar faces com imagens reais, ele apresentou um comportamento esperado, apresentando uma excelente invariância a iluminação porem com uma performance bastante afetada dependendo do angulo da face, só conseguindo detectar faces frontais.

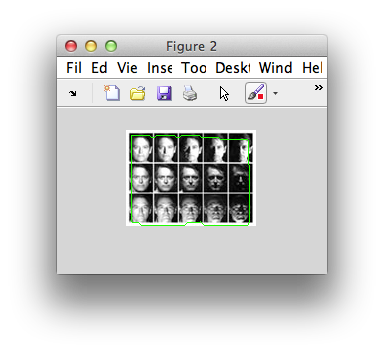


Fig 4 – Detecção das faces com variações de iluminação

# Conclusão

O projeto ainda não se encontra finalizado, diversos problemas ainda precisam ser corrigidos. Apesar do sistema ser invariante a iluminação, resolvendo um dos principais problemas na área de detecção facial, ainda apresenta uma taxa de erro alta para encontrar faces que não estejam na posição frontal. Possuindo um excelente resultado na etapas de treinamento e testes, mas com taxas relevantes de erro com imagens reais, pois as imagens de treinamento e teste possuem apenas faces frontais, falhando em imagens com outras poses, não resolvendo completamente o problema de detecção.

##### Referências

Matlab Trial Software. Disponível em: http://www.mathworks.com/products/matlab/

SOM Toolbox for Matlab5. Disponível em: <http://www.cis.hut.fi/somtoolbox/>

Wikipedia, Learning Vector Quantization. Acessado em Junho/2012. <http://en.wikipedia.org/wiki/Learning_Vector_Quantization>

Wikipedia, Gabor Filter Algorithm. Acessado em Junho/2012. <http://en.wikipedia.org/wiki/Gabor_filter>

Neide Pizzolato Angelo, Aplicação de Filtros de Gabor no PRocesso de Classificação de Imagens Digitais com Base em Atributos de Textura, Porto Alegre, Brasil

Yasuo Kono, José Demiso Simões, Rafael Duarte. Utilização de Rede Neural para Previsão do Nível do Rio Paraguai. São José dos Campos, Brasil.

Department of Informatics and Mathematicam Modeling. The IMM Face Database. Disponível em : <http://www2.imm.dtu.dk/~aam/>

Psychological Image Collection at Stiling. Diponível em: <http://pics.psych.stir.ac.uk/2D_face_sets.htm>

Negatives Database. Asian Institute of Technology. Disponível em : <http://www2.cs.ait.ac.th/vgl/irshad/newsite/data>

Image Magick. Disponível em: <http://www.imagemagick.org/script/index.php>

Software Automator. Disponível em: <http://support.apple.com/kb/HT2488?viewlocale=pt_PT>

Journal of Mathematical Imaging and Vision, Multi-Scale Image Analysis by Piramidal Algorithms, Volume 10, Number 1 (1999), 87-95