



TRABALHO DE GRADUAÇÃO

DETECÇÃO E RECONHECIMENTO FACIAL POR MEIO DE APRENDIZADO DE MÁQUINA

Hugo Leite Florenço Maia

Brasília, Junho de 2016

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE REDES DE
COMUNICAÇÃO

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Curso de Graduação em Engenharia de Redes de Comunicação

TRABALHO DE GRADUAÇÃO

DETECÇÃO E RECONHECIMENTO FACIAL POR MEIO DE APRENDIZADO
DE MÁQUINA

Hugo Leite Florenço Maia

Relatório submetido como requisito parcial para obtenção
do grau de Engenheiro de Redes de Comunicação.

Banca Examinadora

Prof. Daniel Guerreiro e Silva,

UnB/ ENE (Orientador)

Prof. Eduardo Peixoto Fernandes da Silva

UnB/ ENE

Prof. Alexandre Ricardo Soares Romariz

UnB/ ENE

Brasília, Junho de 2016

FICHA CATALOGRÁFICA

HUGO LEITE FLORENÇO MAIA

Detecção e reconhecimento facial por meio de aprendizado de máquina

Distrito Federal, 2016.

xxx, xxp., xxx mm (FT/UnB, Engenheiro, Redes de Comunicação, 2016. Trabalho de Graduação – Universidade de Brasília.Faculdade de Tecnologia.

I. Redes de Comunicação/FT/UnB

REFERÊNCIA BIBLIOGRÁFICA

T. Leite Maia, 2016. Detecção e reconhecimento facial por meio de aprendizado de máquina. Trabalho de Graduação em Engenharia de Redes de Comunicação, Publicação FT.TG-nº , Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, xxp.

CESSÃO DE DIREITOS

AUTORES: Hugo Leite Florenço Maia.

TÍTULO DO TRABALHO DE GRADUAÇÃO: Detecção e reconhecimento facial por meio de aprendizado de máquina.

GRAU: Engenheiro ANO: 2016

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor

(Hugo Leite Florenço Maia)

Rua 9 sul, lote 11, apt 1102-A, Aguas Claras..

71938-360 Brasília – DF – Brasil.

AGRADECIMENTOS

Acima de tudo, agradeço a Deus por me permitir estudar e me graduar como engenheiro em uma instituição de ensino de excelência reconhecida nacionalmente e internacionalmente. Não poderia deixar de citar um trecho que narra a história do Capitão Charles Plumb que era piloto de caça dos EUA e serviu na guerra do Vietnã. Depois de muitas missões de combate, seu avião foi derrubado por um míssil. Plumb saltou de paraquedas, foi capturado e passou seis anos numa prisão norte-vietnamita.

Ao retornar aos Estados Unidos, passou a dar palestras relatando sua odisseia e o que aprendera na prisão. Certo dia, num restaurante, foi saudado por um homem:

“Olá, você é Charles Plumb, era piloto no Vietnã e foi derrubado, não é mesmo?”

“Sim, como sabe?”, perguntou Plumb.

“Era eu quem dobrava o seu paraquedas. Parece que funcionou bem, não é verdade?” Plumb quase se afogou de surpresa e com muita gratidão respondeu:

“Claro que funcionou, caso contrário eu não estaria aqui hoje!!!”

Agradeço a minha esposa Luana Nascimento, minha mãe Maria das Graças e irmã Jéssica por terem dobrado o meu paraquedas por todos esses 7 anos de caminhada. Agradeço também a minha filha Mariana, que em sua doçura, pureza e alegria tem me inspirado a dar o melhor de mim em minhas lutas. Minha sincera gratidão ao meu mestre, professor Daniel, por ter me presenteado o tema deste trabalho e não ter medido esforços para que eu pudesse concluí-lo. Por fim, agradeço também aos colegas de trabalho, que em diversos momentos foram compreensíveis e apoiaram nesta jornada.

Hugo Leite Florenço Maia.

RESUMO

O objetivo deste trabalho foi desenvolver uma proposta de identificação facial utilizando aprendizado de máquina, por meio de técnicas de extração de características e redes neurais artificiais do tipo Perceptron de Múltiplas Camadas (MLP). A primeira etapa do trabalho tratou do estudo de composição das imagens: e como é possível extrair características faciais para construção de um banco de faces e seus respectivos atributos que serão utilizados para treinamento e teste da rede neural artificial. Buscando atingir este objetivo, o texto define conceitualmente uma rede neural e expõe as definições necessárias para o seu entendimento e aplicação no problema de identificação facial. Foram empregadas técnicas de detecção baseadas na integração dos pixels que compõem a imagem e uso de classificadores em cascata para obtenção da face pesquisada. Em função da grande dimensão dos dados gerados nesta etapa, o que consequentemente tornaria inviável o treinamento da rede MLP, considerou-se também o emprego da redução de dimensionalidade por meio de uma técnica que utiliza a Análise de Componentes Principais (PCA), a qual não trouxe perdas significativas de informação e tornou possíveis os ensaios contemplando treinamento e teste com centenas de faces. Após o ensaio com várias simulações com diversos fatores de compressão, obtivemos a média de acerto do classificador para cada índice de compressão de dados de entrada. Em suma, os índices de acerto obtidos demonstram que a rede adquiriu boa capacidade de generalização no teste com o conhecimento adquirido a partir das faces apresentadas no treinamento. Contudo, a estratégia proposta na detecção de face utilizando redes neurais artificiais apresentou razoáveis resultados, média de 70% de acerto, o que demonstra que possui aplicabilidade em problemas reais.

ABSTRACT

The objective of this study was to develop a proposal for face detection using machine learning, through feature extraction techniques and artificial neural networks – the Multilayer Perceptron (MLP). The first stage of the work dealt with the images composition: how to extract facial features to build a faces dataset along with their attributes, which will be used for training and testing the neural network. In order to achieve this goal, the text conceptually defines artificial neural networks and exposes the necessary settings for understanding and application in facial recognition. Pre-detection techniques were used based on pixels integration of the image and on cascade-based classifiers. Due to the large size of the generated data, which compromises MLP training, we also considered the use of the dimensionality reduction technique known as Principal Component Analysis (PCA), which didn't brought a significant loss of information and made possible the full implementation and analysis of the framework. After performing a series of simulations with different compression factors, we obtained the average hits score according to each input data compression ratio. In short, the accuracy results have shown that the model presented a good ability to generalize the information that was gathered during the training stage. Moreover, the proposed face detection strategy using artificial neural networks presented a 70% average success rate, which indicates an applicability to real problems.

SUMÁRIO

CAPITULO 1	9
1. INTRODUÇÃO	9
1.1.MOTIVAÇÃO.....	9
1.2.OBJETIVO	11
1.3.PROPOSTA DE TRABALHO	11
1.4.DESCRIÇÃO	11
CAPITULO 2	12
2. FUNDAMENTAÇÃO TEÓRICA	12
2.1.COMPOSIÇÃO DA IMAGEM	12
2.2.MATRIZ RGB TÍPICA.....	13
2.3.EXTRAÇÃO DE CARACTERÍSTICAS (<i>features</i>)	14
2.4.ALGORITMO DE DETECÇÃO DE FACES VIOLA JONES.....	15
2.5.REDUÇÃO DE DIMENSIONALIDADE UTILIZANDO ANÁLISE DE COMPONENTES PRINCIPAIS (PCA)	19
2.6.REDES NEURAIIS ARTIFICIAIS.....	20
2.7.REDE MLP	22
CAPITULO 3	26
3. IMPLEMENTAÇÃO DA PROPOSTA.....	26
3.1.PLATAFORMA DE DESENVOLVIMENTO	28
3.2.MATLAB - TOOLBOX DE REDES NEURAIIS	28
3.3.BANCO DE FACES.....	29
3.4.REDUÇÃO DE DIMENSIONALIDADE	30
3.5.TREINAMENTO	31
3.6.SIMULAÇÃO DOS ENSAIOS.....	32
CAPÍTULO 4	33
4. RESULTADOS	33
4.1.TESTE DA DETECÇÃO E EXTRAÇÃO DE FACES.....	33
4.2. ENSAIOS DE RECONHECIMENTO UTILIZANDO O CLASSIFICADOR	37
CAPÍTULO 5	42
5. CONCLUSÕES	42
5.1.ANÁLISE DOS RESULTADOS E CONCLUSÕES	42

5.2. TRABALHOS FUTUROS	43
REFERÊNCIAS BIBLIOGRÁFICAS.....	45
APÊNDICE A – CÓDIGO DE SIMULAÇÃO	47
APÊNDICE B – SCRIPT DE EXECUÇÃO DO ENSAIO E RESULTADOS	49
APÊNDICE C – EXEMPLO DE DETECÇÃO DE FACE COM O MÉTODO VIOLA JONES	50

CAPITULO 1

1. INTRODUÇÃO

O propósito da engenharia é engendrar, inventar, desenvolver tecnologias que atendam às pessoas, é criar soluções. [1]

Os engenheiros devem ter a clara noção da importância do diálogo entre eles e a sociedade, para que a tecnologia não seja vista como vilã que vitimiza seus usuários, mas uma tecnologia inclusiva e harmonizada com o contexto social em que está inserida.

Da Vinci concebeu um futuro em que a máquina se tornava, como se tornou, parte do homem, não num sentido utópico, porém funcional. E sob a direção da inteligência humana[1].

A engenharia física deve ser desenvolvida das ciências puras, ao mesmo passo que a engenharia humana, visando a integrar cientificamente as relações entre as pessoas e suas necessidades. [1].

Neste sentido, o propósito deste trabalho surge da facilidade que a visão computacional (ou visão por máquina) pode proporcionar às pessoas, e, por conseguinte influenciar positivamente as relações sociais e humanas.

1.1. MOTIVAÇÃO

Em princípio, as demandas de segurança da informação exigem soluções que buscam identificar um padrão único e imutável no âmbito do físico humano, seja a assinatura ou medidas biométricas como leitura da íris, reconhecimento de voz, DNA e reconhecimento facial. São fontes de padrões que, sozinhos ou em associação, podem caracterizar uma pessoa, concedendo a ela a autenticidade.

Neste contexto, o reconhecimento de faces será o tema alvo deste trabalho, em função da praticidade e disponibilidade que esta técnica permite. Uma vez que o usuário não precisa de nenhuma ação para validar suas credencias, bastando por exemplo somente expor a face a um dispositivo de captura de imagens. É um método menos invasivo e requer em geral poucos recursos para captura da face, como uma simples webcam.

São diversas as áreas que se beneficiam de sistemas de reconhecimento de face, incluindo entretenimento, tais como na interação humano/computador em vídeo games e dispositivos pessoais [2].

A aplicação das técnicas de visão computacional não se restringe tão somente a questões de segurança, mas envereda também em outras áreas do conhecimento, tais como a cartografia no controle e ocupação do ambiente. É um instrumento que pode fornecer dados a serem posteriormente interpretados por algoritmos que detectam desmatamento florestal e contribuem aos mais diversos tipos de pesquisas ambientais, auxiliando o gestor ambiental na tomada de decisões [3].

No monitoramento e controle da concentração de contaminantes no solo o uso de imagens permite utilizar métodos de detecção de elementos químicos que torna possível a identificação de similaridade entre o espectro de um elemento da imagem e um espectro de referência [4].

Na indústria, a visão por máquina pode ser utilizada para reconhecimento de objetos defeituosos em linhas de produção automatizada, substituindo de forma eficaz o olho humano em tarefas repetitivas e sujeitas a erros em função do stress por repetição a que o trabalhador é submetido [5].

As técnicas de reconhecimento facial também podem ser empregadas na área de segurança na Internet e em aeroportos; em áreas de vigilância, como na análise de vídeos de câmeras de segurança [6], ou até mesmo em veículos. Por exemplo, já existem veículos vendidos comercialmente que possuem câmeras de identificação de pessoas parte dianteira, as quais mesmo que fora do alcance de visão do condutor, são câmeras que associadas a outros sistemas possuem a capacidade de alertá-lo do risco de colisão frontal, ou até mesmo interferir proativamente na condução e ativar os freios para que não ocorra o acidente, caso o condutor não tome a decisão em tempo hábil [7].

Cabe destacar que há uma notória distinção entre os conceitos de visão computacional e processamento de imagens. O primeiro aborda uma sistemática com foco na teoria de Aprendizado de Máquina, ou em inglês *Machine Learning*, que é contexto deste trabalho; já o segundo consiste em aperfeiçoar os dados visuais disponíveis para enriquecer em termos de informações a visão humana [8].

1.2. OBJETIVO

Dada a devida contextualização, este trabalho propõe estudar técnicas básicas de processamento de imagens para extração de características, com a finalidade de implementar uma solução de reconhecimento facial, sob a ótica das abordagens clássicas em aprendizado de máquina.

1.3. PROPOSTA DE TRABALHO

Para atingir o objetivo geral proposto, as etapas do trabalho então se dividem nos seguintes conjuntos de objetivos específicos:

- Pesquisar técnicas elementares de processamento de imagens;
- Estudar técnicas tradicionais de extração de características de imagens;
- Compreender o funcionamento de redes neurais artificiais;
- Propor uma técnica de detecção de faces utilizando redes neurais artificiais e extração de características;
- Implementar a proposta: compor a base de dados, i.e. gerar o conjunto de vetores contendo os padrões (*FEATURES*) extraídos das imagens; executar o treinamento do classificador;
- Realizar ensaios numéricos da etapa de treinamento do classificador e em cenários de teste da metodologia, a fim de avaliar sua aplicabilidade.

1.4. DESCRIÇÃO

Nos próximos capítulos deste trabalho serão desenvolvidos os objetivos desta pesquisa. No Capítulo 2 será percorrida a fundamentação teórica do assunto; neste capítulo, o foco é o entendimento da composição da imagem digital bem como ela pode ser manipulada, também apresentar informações sobre redes neurais – biológicas e artificiais, e compreender os algoritmos de detecção facial. Ademais, elucidar técnicas de redução de dimensionalidade para que se torne viável a construção de classificadores. No Capítulo 3, será discutida a implementação da proposta utilizando a plataforma Matlab e sua *toolbox* de redes neurais artificiais, além de versar sobre os pontos críticos em trabalhos relacionados. Ainda neste capítulo, ensaios são propostos para que, a partir dos seus resultados, seja possível a análise numérica e qualitativa da metodologia. No Capítulo 4 tais resultados são apresentados e discutidos, e finalmente o Capítulo 5 tece as considerações finais e as perspectivas de trabalhos futuros.

CAPITULO 2

2. FUNDAMENTAÇÃO TEÓRICA

O objeto de estudo, a saber: processamento de imagens e reconhecimento de padrões utilizando redes neurais artificiais, deste trabalho possui atributos técnicos muito específicos, o que requer uma prévia discussão teórica para que uma vez que as definições estejam apresentadas, haja conforto no entendimento, de modo que permita compreensão do que será proposto e posterior análise dos resultados obtidos.

2.1. COMPOSIÇÃO DA IMAGEM

Este tópico tem por objetivo caracterizar a composição das imagens digitais. Nesse sentido, a imagem digital é composta precipuamente por bits e há uma abordagem matemática que detalha a organização e composição desta.

As imagens digitais podem ser definidas por uma função discreta bidimensional, na qual x e y representam as coordenadas espaciais com valores inteiros e o valor de $f(x,y)$ no espaço bidimensional $[x,y]$ corresponde à intensidade da imagem em função do par cartesiano x e y .

A menor unidade de composição da imagem é o pixel, e pode assumir valores compreendidos entre $[0, 2^k - 1]$ sendo que k um valor inteiro e geralmente indica o numero de bits utilizados para mapear a matriz de cores [9]. Por exemplo, em imagens RGB são utilizados $8+8+8 = 24$ bits para representar $255 \times 255 \times 255 = 16.581.375$ possíveis tons de cores. Em imagens monocromáticas, cada pixel pode apresentar vários tons de uma única cor e cada tom é representado por um campo único de 8 bits que corresponderá a uma variação de tom, e essa variação é representada por um valor compreendido entre $[0, 255]$. RGB é uma tipo de espaço de cores, ainda há outros disponíveis.

O referencial do sistema de coordenadas na imagem adotado neste projeto obedece ao padrão utilizado no Matlab (software detalhado na subseção 3 deste trabalho), plataforma que permitirá a implementação do trabalho. Neste sentido o marco $(1,1)$ do plano cartesiano referente à imagem encontra-se na parte superior esquerda e os valores XY podem estar no intervalo $[1, \infty]$, como ilustrado na Figura 1.1.

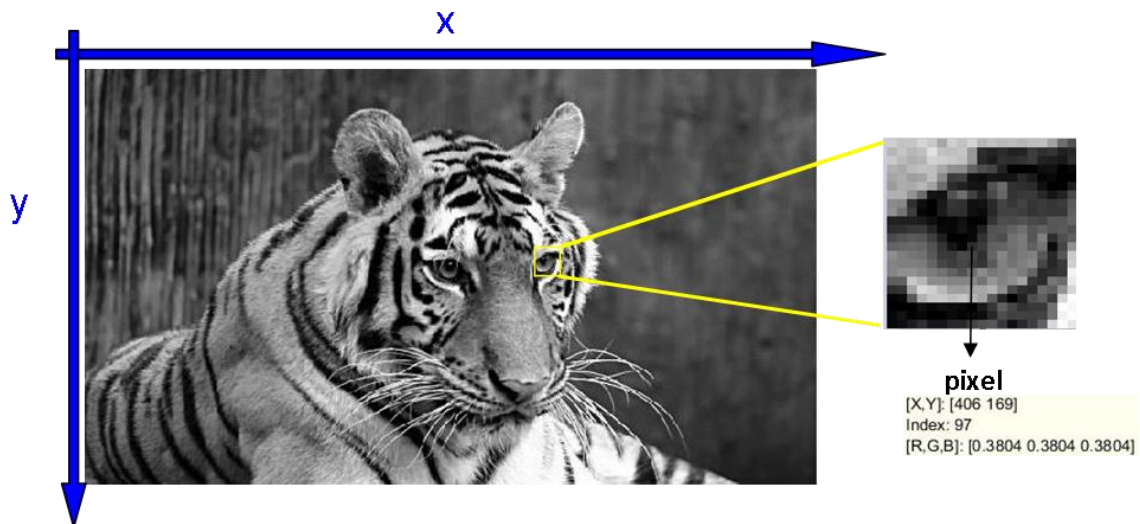


Figura 1.1. Exemplo de imagem monocromática.

A imagem colorida é apresentada de modo similar que a imagem monocromática, exceto por ser constituída pela superposição de três planos, constituindo assim a matriz RGB, detalhada melhor na subseção 2.2 a seguir. Em imagens coloridas como a da Figura 1.2, a menor unidade da imagem, o pixel, é representado por uma tupla ternária sendo que os valores variam de acordo com o espaço de cor utilizado.

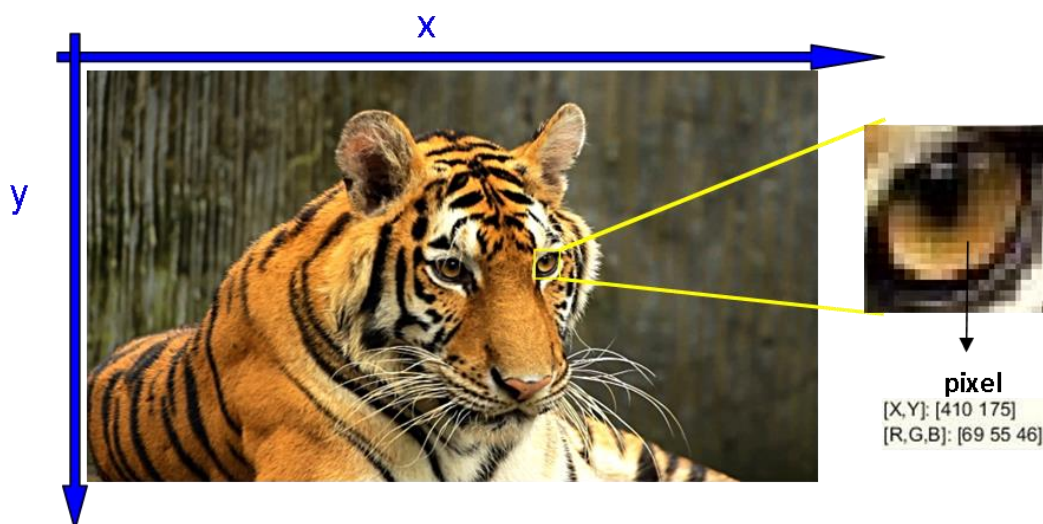


Figura 1.2. Exemplo de imagem monocromática.

2.2. MATRIZ RGB TÍPICA

Diversos programas de edição de imagens trabalham com espaços de cores diferentes. Espaço de cores corresponde a todas as possibilidades de cores possíveis

obtidas a partir da combinação das bases de dimensão usadas. Em se tratando padrão RGB, o qual foi utilizado nas simulações. O espaço de cores RGB é constituído por um cubo de dimensões: lado R, comprimento G e altura B [9], ou seja, possui base de dimensão 3. A Figura 1.3 ilustra a representação geométrica espacial do formato RGB, em forma de um cubo. Nesta figura é possível observarmos que cores parecidas podem ser representada por diversos valores de RGB e as cores básicas conhecidas são obtidas da seguinte relação:

- Branco - RGB (255,255,255);
- Azul - RGB (0,0,255);
- Vermelho - RGB (255,0,0);
- Verde - RGB (0,255,0);
- Amarelo - RGB (255,255,0);
- Magenta - RGB (255,0,255);
- Ciano - RGB (0,255,255);
- Preto - RGB (0,0,0).

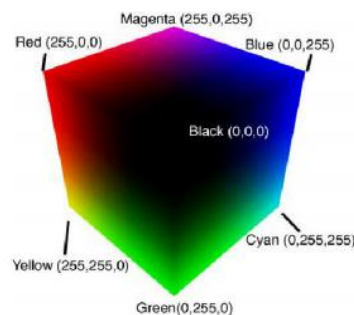


Figura 1.3. Espaço de cores no cubo RGB.

2.3. EXTRAÇÃO DE CARACTERÍSTICAS (*features*)

Um dos assuntos de maior relevância deste trabalho e que possibilita, em termos práticos, a aplicação da metodologia de classificação proposta é a extração de “*FEATURES*” ou características/atributos das imagens. O reconhecimento facial por meio de redes neurais, requer que a formato da informação e a técnica de extração de face sejam padronizados. Vez que a rede uma vez montada não pode ter o numero de entradas ou saída alteradas.

Entre essas características pode-se incluir os dados relativos ao brilho de uma determinada região, a textura de uma região, a amplitude do histograma, entre outras.

De uma forma geral, a extração de atributos é um processo usualmente associado à análise das regiões de uma imagem. Os métodos para criar novas características baseadas na combinação ou na transformação de características já existentes são chamados métodos de extração de atributos.

O estudo de extração de atributos em reconhecimento de padrões requer a definição de duas entidades:

Atributo (*feature*): é qualquer aspecto, qualidade ou característica de um objeto. Os atributos podem ser simbólicos, numéricos ou ambos [10].

Padrão: é um conjunto de atributos que definem um objeto ou um grupo de objetos [10].

Neste trabalho as características das faces (região de pixels em que se encontra a face) identificadas nas imagens foram extraídas por meio da técnica de detecção de faces Viola-Jones, a ser apresentada na seção subsequente. O formato resultante padronizado é um conjunto *box-frame* padrão de tamanho 60x60x60x3 (padrão RGB), que equivale a um vetor de atributos de dimensão 10800, como está ilustrado na Figura 2.1:

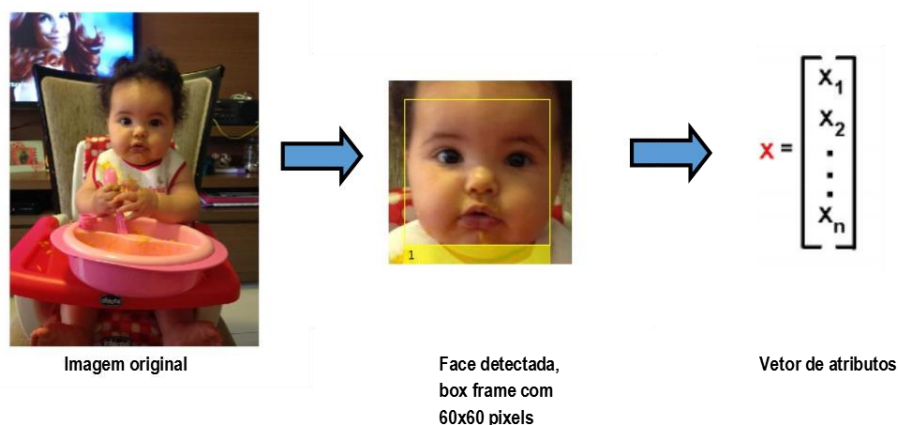


Figura 2.1: extração de atributos e características das imagens.

2.4. ALGORITMO DE DETECÇÃO DE FACES VIOLA JONES

O algoritmo de detecção de faces utilizado nesse trabalho foi proposto por Paul Viola e Michael Jones em 2001 [11]. Trata-se de uma abordagem para detecção de objetos em imagens que se baseia em três conceitos: integral de imagem,

treinamento de classificadores usando *boosting* e seu posterior uso em cascata. Embora o algoritmo possa ser treinado para reconhecer qualquer objeto, a motivação principal da abordagem de Viola e Jones foi o reconhecimento facial. O ponto forte deste algoritmo é a rapidez com que é executado. A integral de imagem, também conhecida como tabela de soma de áreas, é um algoritmo proposto por Frank Crow em 1984 [22] que permite avaliar eficientemente a soma dos valores dos pixels (intensidade dos níveis de cinza) de uma área retangular em uma sub-região da imagem. A Equação 2.1 indica como calcular a integral de imagem em uma determinada coordenada:

$$ii(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} i(x', y') \quad (2.1)$$

Onde $ii(x, y)$ é a integral da imagem nas coordenadas do pixel (x, y) e $i(x, y)$ é a imagem original. Pode-se ver que a integral da imagem na coordenada (x, y) é a soma dos valores dos pixels acima de y e à esquerda de x , inclusive x e y (supondo que a origem do sistema de coordenadas está localizada no canto superior esquerdo da imagem). A tabela de soma pode então ser computada para todos os pixels em uma única varredura, como mostra a Equação 2.2:

$$ii(x, y) = i(x, y) + ii(x - 1, y) + ii(x, y - 1) - ii(x - 1, y - 1) \quad (2.2)$$

Define-se $(x - 1) = 0$ e $(y - 1) = 0$ para contornar os casos nos quais as coordenadas dos pixels estão fora dos limites da imagem.

Desse modo, encontra-se facilmente a soma de área em qualquer região retangular da imagem. Portanto, dada uma região retangular ABCD de uma imagem (Figura 2.2), a soma das intensidades dos pixels nessa área pode ser calculada como:

$$\sum_{(x,y) \in ABCD} i(x, y) = ii(A) + ii(D) - ii(B) - ii(C) \quad (2.3)$$

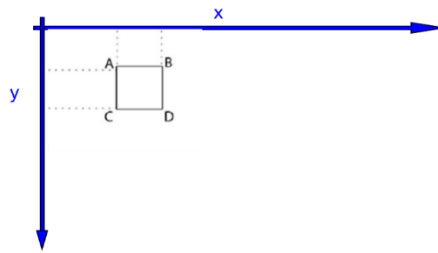


Figura 2.2: Região ABCD em uma matriz de pixels.

A partir da integral de imagem é possível identificar padrões utilizando características Haar-like, que são máscaras retangulares nas quais os valores dos pixels de uma região são subtraídos dos valores dos pixels de outra região, representando uma diferença de intensidade luminosa entre áreas da imagem [11]. A Figura 2.3 mostra quatro possíveis tipos de características-base que podem ser usadas, e que são calculadas subtraindo-se a soma dos valores dos pixels da região branca, da soma dos valores dos pixels da região preta. Para calcular a característica A da figura 2-3, são necessárias oito consultas à tabela de soma de áreas (integral de imagem), respectivamente para os oito pontos indicados.

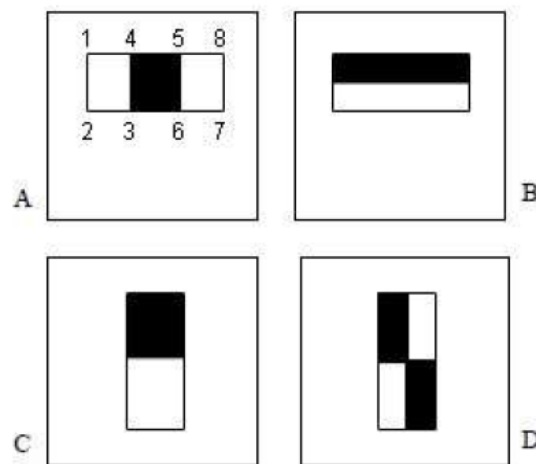


Figura 2.3: Representação visual de quatro tipos de características Haar-like. Fonte: extraído de [11].

Cada tipo de característica pode ajudar a reconhecer um determinado padrão, principalmente quando combinados em cascata. Por exemplo, a característica B da figura 2-2 permite identificar uma área na imagem onde há uma diferença de intensidade significativa entre a parte superior e a parte inferior de uma região. Essa característica pode ser aplicada no processo de detecção de faces, uma vez que frequentemente a região dos olhos é mais escura do que a região das bochechas. Obviamente, existem outros padrões com o mesmo perfil e que não são faces, daí a

necessidade de combinar várias características para refinar a busca. A resolução base da máscara usada no algoritmo é de 24x24 pixels. Existem mais de 100.000 possíveis características Haar-like distintas se considerarmos diferentes posições e tamanhos destas dentro da janela.

O segundo passo no algoritmo de Viola-Jones é o treinamento de classificadores. Dado um conjunto de características deve-se treinar o sistema com imagens positivas (faces) e imagens negativas (tudo menos faces). Para isso, deve-se usar um algoritmo de treinamento que aprenda funções de classificação. Uma opção é utilizar um algoritmo de aprendizagem que empregue a estratégia de *Boosting*, que consiste em encontrar um classificador de alta precisão combinando-se muitos classificadores “fracos”, onde cada um desses classificadores fracos possui uma precisão média com uma taxa de acertos de pelo menos 51% [11].

O cascadeamento de classificadores, ilustrado na Figura 2.4, é a ultima etapa do processo em que, para se conseguir altas taxas de detecção e, determina-se que a avaliação por um segundo classificador só será invocada caso a avaliação do primeiro seja positiva. Caso contrário, o procedimento é interrompido e a sub-janela que engloba o suposto atributo a detectar é rejeitada. Portanto, é necessário um resultado positivo em todos os classificadores para que a detecção do padrão em uma sub-janela tenha êxito.

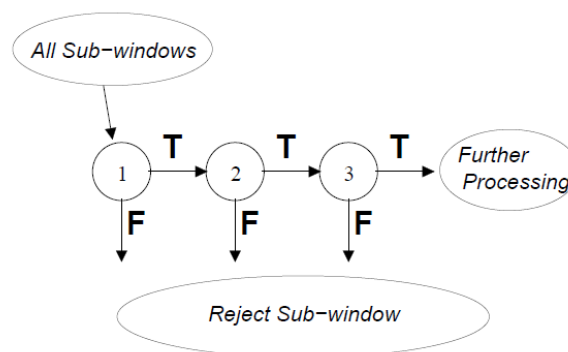


Figura 2.4: Cascadeamento de classificadores. Fonte: extraído de [11].

Os autores de [11] afirmam que a principal característica do método que apresentam é o bom desempenho propiciado pelo conjunto de passos do algoritmo, permitindo que ele apresente uma taxa de detecção tão boa quanto outras apresentadas na literatura, porém, com tempo de processamento menor.

2.5. REDUÇÃO DE DIMENSIONALIDADE UTILIZANDO ANÁLISE DE COMPONENTES PRINCIPAIS (PCA)

Ao se tentar o treinamento da rede neural artificial com o número original (10.800) de atributos gerados pelo algoritmo de detecção Viola-Jones, claramente mostrou-se ser inviável prosseguir com um vetor de atributos de tal dimensão, pois em termos de tempo de execução e custo de memória, tal tentativa consumiu todo o recurso computacional disponível e portanto não pôde ser concluída.

Consequentemente, o vetor de atributos para o classificador precisa ter a sua dimensão reduzida para que o projeto seja exequível. Neste sentido faz-se necessário explicar a aplicação do procedimento de redução de dimensionalidade.

A Análise de Componentes Principais (PCA, do inglês *Principal Component Analysis*) está relacionada à obtenção de uma transformação linear que mapeia dados N -dimensionais num espaço M -dimensional, tal que $M \ll N$. E pode propiciar a solução para o problema de dimensionalidade detectado.

Neste sentido, há dois pontos cruciais, portanto:

- Como conseguir uma projeção adequada, ou seja, representativa?
- Quanta “informação” será perdida quando os dados forem levados para uma dimensão menor?

Emana então a ideia de construir uma matriz de projeção e analisar qual seria o erro quadrático médio (EQM) entre os dados projetados e os dados originais. Começa-se então removendo a média dos dados:

$$x_k \leftarrow x_k - \frac{1}{N_{\text{dados}}} \sum_{i=1}^{N_{\text{dados}}} x_i \quad (2.4)$$

Sendo N_{dados} o número total de padrões, representados pelo vetor x_k . Perceba que a Equação 2.4 significa simplesmente que é subtraída de cada dado a média amostral, forçando, assim, uma situação de média zero. Nesse caso, a ideia exposta

anteriormente leva à seguinte função custo para que se obtenham as direções de projeção ótimas:

$$J_{PCA} = \frac{1}{N_{dados}} \sum_{i=1}^{N_{dados}} \left\| \sum_{c=1}^M z_i a_c - x_i \right\|^2 \quad (2.5)$$

Pode-se mostrar que o conjunto de direções a_c , $c = 1, \dots, M$, que minimiza essa função custo corresponderá ao conjunto dos M autovetores associados aos M maiores autovalores da matriz de autocorrelação amostral R_x dos dados [21]. Essa matriz, que é simétrica, tem dimensão $N_{dados} \times N_{dados}$ e é definida por elementos r_{ij} :

$$r_{ij} = \frac{1}{N_{dados}} \sum_{l=1}^{N_{dados}} x_{l,i} x_{l,j}^T \quad (2.6)$$

sendo $x_{l,i}$ o i -ésimo elemento do vetor x_k . É possível definir essa matriz de modo ainda mais simples como:

$$R_x = \frac{1}{N_{dados}} \sum_{l=1}^{N_{dados}} x_l x_l^T \quad (2.7)$$

Como os autovetores da matriz de autocorrelação são ortogonais, as direções de projeção a_c possuem essa propriedade também. Ademais, a matriz R_x é definida não negativa, sendo, aliás, muitas vezes, definida positiva. Portanto, seus autovalores serão sempre não-negativos e comumente positivos [12].

Os elementos que compõem cada vetor resultante da transformação $z_i = Ax_i$ são chamados de componentes principais do vetor x_i a ele associado, daí advém o nome da técnica. PCA, portanto, permite realizar redução de dimensionalidade por meio de projeções que minimizam o erro quadrático médio de “compressão” [13].

2.6. REDES NEURAIS ARTIFICIAIS

O reconhecimento de padrões é uma área da ciência que identifica e estuda as interações cognitivas que reproduzem características únicas na natureza ou

identidades. São elas: digitais humanas, voz, sinais, assinaturas manuscritas, íris dos olhos, reconhecimento facial [14].

O tema não é restrito a questões de segurança, mas possui aplicabilidade em técnicas de controle de qualidade de pintura de produto na indústria automotiva, por exemplo; também possui aplicação na geologia, na identificação de movimento de placas tectônicas por meio de imagens de satélites ou na identificação de minerais e substâncias; no monitoramento e controle de desmatamento e monitoramento de ocupação do ambiente [15].

O padrão ou a característica para ser tipificado como tal deve preencher alguns pré-requisitos:

Universalidade: significa que todos membros desta classe devem possuir o atributo; por exemplo: todo ser humano possui face isso é uma regra universal [16] ou seja um, a face é um atributo e possui um padrão associado.

Mensurabilidade: indica que a característica pode ser quantizada e medida [16].

Entre as diversas técnicas existentes para reconhecimento de padrões, uma das mais tradicionais é a Rede Neural Artificial (RNA). Em termos simples, ela é a associação de uma grande quantidade de unidades simples de processamento inspiradas no sistema nervoso de seres humanos [17].

Redes neurais artificiais são processadores capazes de aprender através da experiência e, então, utilizar o conhecimento adquirido em situações novas no mesmo escopo de sua aprendizagem. As RNAs foram criadas com base no modelo de aprendizado humano, que inclui os neurônios e as suas transmissões sinápticas, bem como as propriedades de plasticidade e adaptabilidade. O modelo “canônico” de uma rede neural artificial é composto pela associação de múltiplos neurônios (unidades de processamento), onde cada um contém pesos sinápticos (fatores de multiplicação dos sinais de entrada), um somador para reunir os sinais de entrada e a função de ativação, que responde ao sinal de entrada apresentando o sinal de saída [12].

Quanto à arquitetura, dentre as tipificações possíveis, as redes neurais artificiais podem ser classificadas conforme o número de camadas de neurônios que possuem. Neste trabalho, serão utilizadas redes do tipo MLP (Perceptron de Múltiplas Camadas), cuja unidade básica é o Perceptron, ilustrado na Figura 2.5.

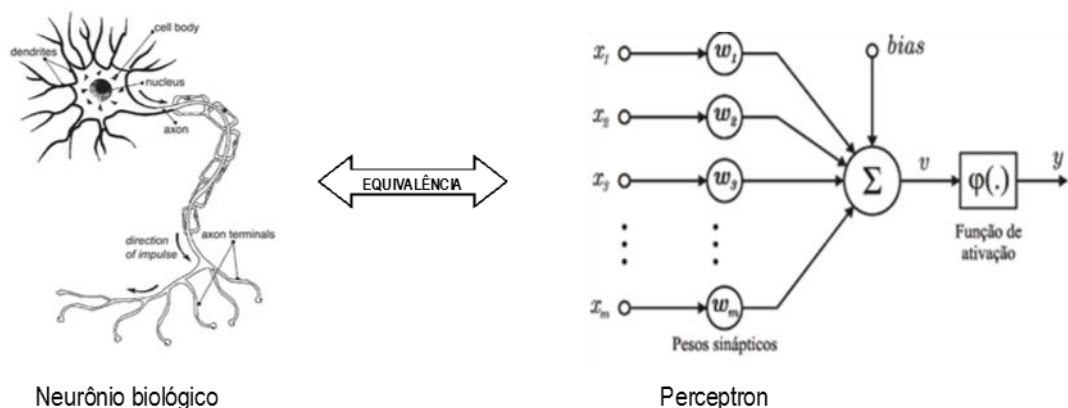


Figura 2.5: associação entre neurônio e o modelo do Perceptron.

2.7. REDE MLP

O Perceptron é uma representação computacional pelo qual é apresentada a menor unidade neural artificial, conforme ilustra-se a Figura 2.5. Ele possui a seguinte arquitetura: entradas, processamento e saída. Ao se interligar com outros Perceptrons, o que constitui uma RNA do tipo MLP, forma-se um modelo aproximador de grande capacidade adaptativa.

Pode-se dizer que é nos pesos sinápticos que se encontra todo o conhecimento adquirido pela rede. Os pesos são responsáveis pelos parâmetros ajustáveis que mudam e se adaptam à medida que o conjunto de treinamento é apresentado à rede. Assim, o processo de treinamento/aprendizado de uma RNA resulta em sucessivos ajustes dos pesos sinápticos, de tal forma que a saída da rede seja a mais próxima possível da resposta desejada (alvo ou “target”).

A função de ativação dos neurônios da rede pode ser escolhida entre dentre algumas opções, conforme a resposta desejada. Nas imagens na Figura 2.7 consta os tipos respostas disponíveis.

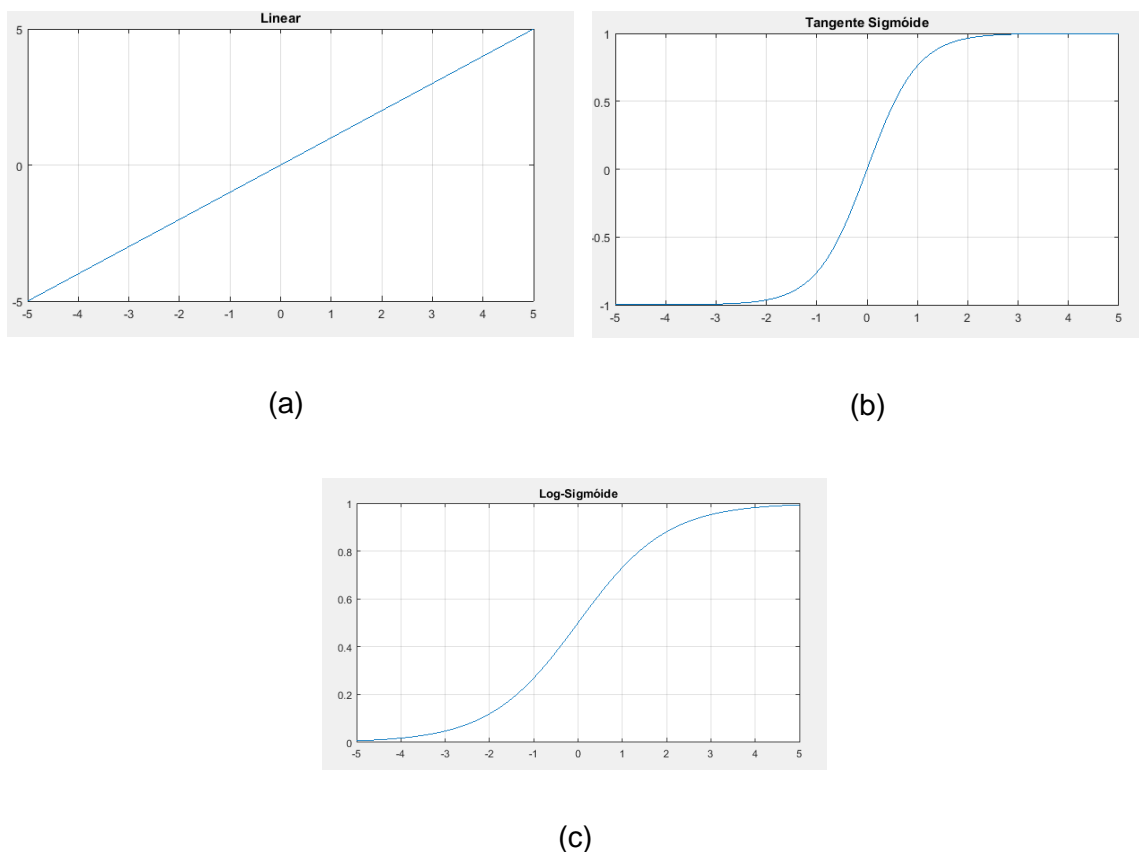


Figura 2.7: Exemplos de funções de ativação que podem ser usadas na rede neural artificial, (a) linear, (b) tangente sigmóide e (c) log-sigmóide.

Neste trabalho, com o propósito de minimizar a influência de ruído na resposta do Perceptron, utilizou-se o padrão de saída com fator de distância de fronteiras entre símbolos igual a 2, de forma que o valor das saídas está compreendido entre $[-1, 1]$. Repare que na figura 2.5 à direita, onde se encontra o modelo do Perceptron, encontra-se o termo chamado de “*bias*”, o qual aumenta os graus de liberdade e permite uma melhor adaptação, por parte da rede neural, ao conhecimento fornecido a ela.

Semelhante à capacidade de realizar diversas ligações sinápticas dos neurônios biológicos, o neurônio artificial associa-se a outros Perceptrons para constituir as redes do tipo MLP. Estas redes são exemplos clássicos de máquinas de aprendizado e são utilizadas em diversos problemas de classificação, mediante a aplicação de uma técnica de treinamento adequada. Vejamos a título de exemplo, a expressão matemática da MLP com um neurônio na camada oculta a partir da

aplicação de sinal de entrada x , de dimensão $[i \times 1]$, em que o sinal de entrada possui i linhas as quais correspondem cada uma a um *input* da MLP.

$$\sum_{i=1}^n w_i x_i + b_i = v_i \quad (2.7)$$

Em que:

w_i : é o peso sináptico da entrada do neurônio i

b_i : indica o bias de entrada do neurônio i

x_i : sinal de entrada

n : quantidade total de neurônios na camada oculta

v_i : resposta de saída

Na figura 2.8 verificamos uma MLP mais complexa que possui duas camadas ocultas em sua estrutura.

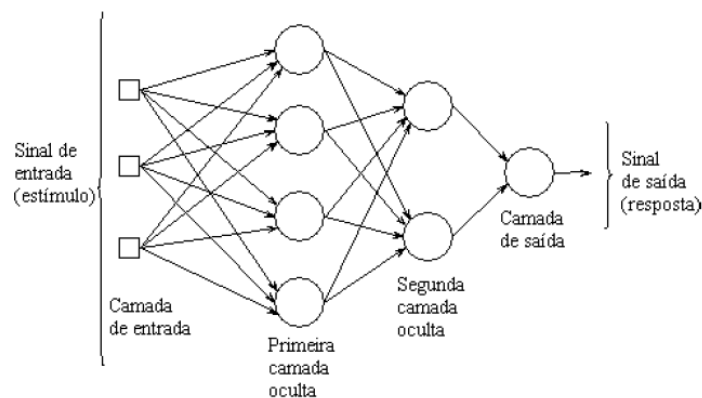


Figura 2.8: MLP com 2 camadas.

O ajuste dos parâmetros livres é feito no processo de aprendizado, onde os vetores de atributos representam estímulos de entrada na rede, cujo objetivo é gerar um modelo implícito adaptado ao ambiente de operação da rede, como ilustra o diagram de estados da MLP na Figura 2.9.

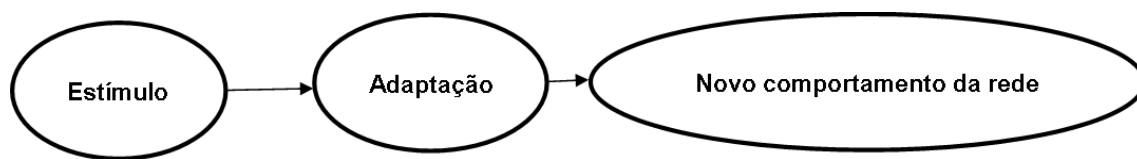


Figura 2.9. Diagrama de estados do processo de aprendizado da RNA

De forma mais clara, o treinamento da MLP é tradicionalmente feito de forma supervisionada e envolve o ajuste dos pesos sinápticos por meio do algoritmo de minimização do erro quadrático médio estimado entre a saída da rede e os valores desejados de referência das amostras de treinamento, com auxílio da técnica de “*backpropagation*” ou retropropagação.

Durante o treinamento com o algoritmo “*backpropagation*”, a rede opera em uma sequência de dois passos. Primeiro, um padrão é apresentado à camada de entrada da rede $(x_1, x_2, x_3, \dots, x_n)$. A atividade resultante flui através da rede, camada por camada, até que a resposta seja produzida pela camada de saída. Em seguida, a saída obtida é comparada à saída desejada para esse padrão particular, calculando o erro associado. O erro é propagado a partir da camada de saída até a camada de entrada, de forma que se calcule o vetor gradiente da função de erro quadrático médio e consequentemente os pesos das conexões sinápticas w_i das camadas internas são ajustados na direção minimizante da função custo. Repete-se o processo para todos os padrões, o que compreende uma época, e repetem-se as épocas até que se atinja um número máximo ou um dado limiar de qualidade no treinamento [12].

Outro critério de parada do treinamento é a avaliação da capacidade de generalização da rede após cada época de treinamento, neste caso, após cada época a rede é testada pelo conjunto de validação, e então o processo é paralisado antes que o erro médio medido em um conjunto de dados de validação seja deteriorado.

CAPITULO 3

3. IMPLEMENTAÇÃO DA PROPOSTA

Discorridos os aspectos teóricos fundamentais, passamos a partir deste capítulo a detalhar a proposta em si. Em termos gerais, o foco do trabalho é criar um classificador de imagens que, contando com as devidas etapas de pré-processamento (detecção Viola-Jones e PCA), possa realizar o reconhecimento de faces de pessoas.

Utilizando o Matlab, parte-se da análise de dois classificadores do tipo MLP com uma camada oculta, o primeiro com 10 neurônios nesta e o segundo com 20 neurônios. Para fins de organização no desenvolvimento da proposta, a implementação geral do projeto foi estruturada em um fluxo de processos, conforme ilustrado na figura 3.1.

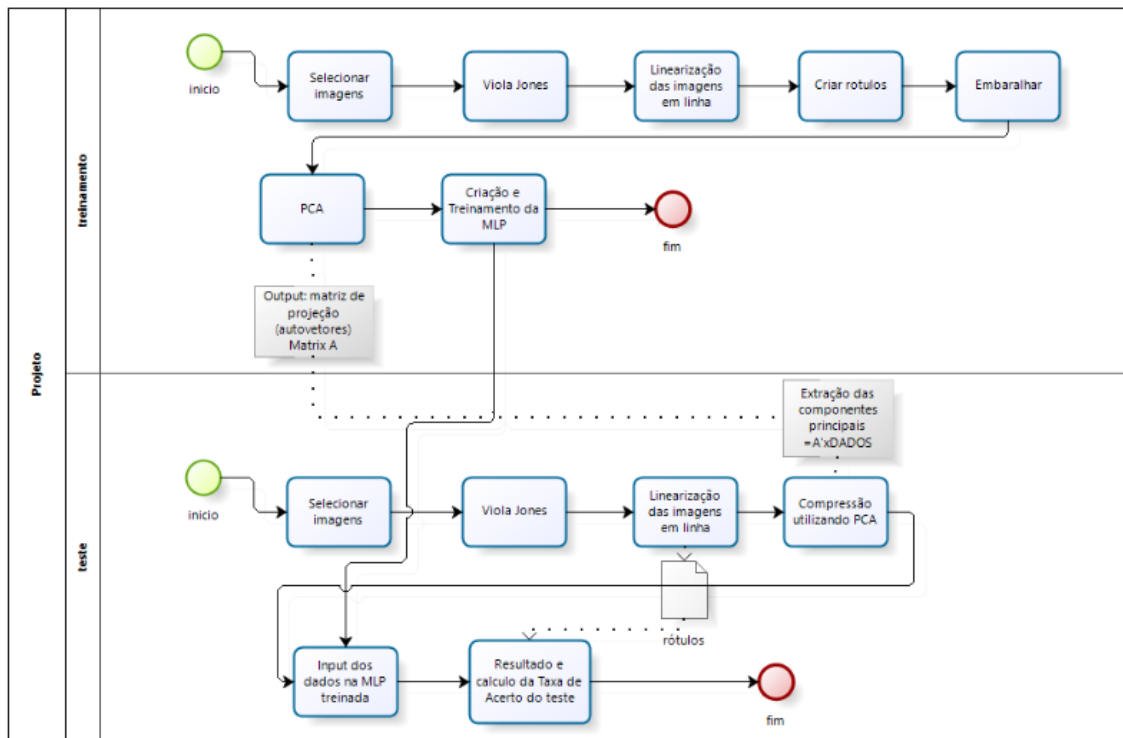


Figura 3.1: fluxo de processo do projeto.

Observe que a figura ilustra um “*dashboard*” que se divide em duas etapas: treinamento e teste. Mas por que treinar e também testar?

Devido ao caráter puramente orientado aos dados, para um bom aprendizado da RNA é necessário que a base de dados de treinamento contenha, no problema em

particular deste trabalho, a mesma face/padrão em diversos contextos para que se construa um modelo com boa capacidade de generalização .

Quanto maior for a quantidade de informações no treinamento, melhor será o aprendizado. Desta forma, planejou-se o treinamento levando em conta quatro possíveis classes de faces: pessoa1, pessoa2, pessoa3 e uma quarta classe que contém amostras de pessoas distintas das três classes anteriores. Neste sentido, espera-se que o aprendizado considere não somente as três faces que se deseja identificar, mas também os padrões diferentes dos quais se deseja uma identificação positiva. A rede também precisa identificar quanto não há o padrão desejado.

Além do aspecto de discriminar corretamente a classe presente ou não, deve-se levar em conta a capacidade de generalização, isto é, se a rede consegue associar corretamente, após o treinamento, padrões “novos” às classes. Por esta razão o projeto foi segmentado em duas etapas, treinamento e teste. Na primeira etapa a RNA aprende a partir de uma base de faces já selecionadas e, no segundo momento, é submetida à base de dados de teste para fins de verificação quanto à capacidade de generalização e viabilidade de uso em um ambiente “real”.

Recorde-se que, devido à elevada dimensionalidade gerada pela etapa de detecção de faces, os dados precisam ser em seguida compactados por meio da técnica de Análise de Componentes Principais (PCA), cuja matriz de projeção foi gerada na etapa de treinamento, transformando/compactando a base de dados até a dimensão de entrada desejada. Por esta razão verificamos que no fluxo da Figura 3.1 há dois *links* entre os processos de treino e teste, um apresenta a própria RNA treinada que também será utilizada no teste e o outro a matriz de compactação PCA que comprimiu a base de treino e também será usada para comprimir a base de teste.

Na etapa de teste e operação da rede, a matriz de compactação realiza a compressão dos dados (por meio de uma operação vetorial $A_{pca}^T \times Dado$) por meio da seguinte transformação linear de dimensionalidade:

$$A_{10800 \times PCA}^T \times Dado_{10800 \times 1_{teste}} = Input_{MLP_{PCA \times 1}} \quad (3.1)$$

Em que: $A_{10800 \times PCA}^T$ é a matriz de compactação;
 $Dado_{10800 \times 1}$ trata-se da face que será apresentada a rede para fins de teste;
 $Input_{MLP_{PCA \times 1}}$ é o sinal resultante compactado pronto para ser apresentado a MLP.

3.1. PLATAFORMA DE DESENVOLVIMENTO

O MATLAB (abreviação de 'laboratório de matrizes' - MATRIX LABoratory) é um sistema para cálculos matemáticos e matriciais que também comporta uma linguagem própria de programação. Todas as variáveis são tratadas como matrizes pelo MATLAB, com uma característica especial: são dimensionadas automaticamente, fato que facilita sobremaneira a implementação de algoritmos matriciais. Outra vantagem do uso do MATLAB é o seu extenso conjunto de rotinas de representação gráfica. É possível a criação de programas com as funções do MATLAB para implementar algoritmos mais complexos. Esses programas são conhecidos como arquivos-M ou scripts. A implementação deste trabalho foi feita na versão do R2015a do sistema.

Neste trabalho utilizou-se diversas funções associadas às etapas fundamentais tais como importação de dados, detecção de faces, extração de *FEATURES*, criação da RNA, seu treinamento e teste, bem como validação dos resultados e cálculo de acerto da simulação. A seguir destacamos alguns destes aspectos.

3.2. MATLAB - TOOLBOX DE REDES NEURAIS

Dentre diversas ferramentas intrínsecas ao Matlab; para este trabalho, merece destaque a *toolbox* específica para redes neurais, a qual é acessada por meio do comando *nntool*, digitado no prompt de comandos do Matlab. Este programa, versão R2015a, oferece ferramentas e suporte a aplicações que utilizam redes neurais artificiais. O mesmo constrói a rede de neurônios a partir do vetor atributo em formato coluna, ou ainda justaposição de vários vetores atributos em colunas adjacentes, constituindo assim uma de matriz que representa uma base de dados. Nesta toolbox de contexto é possível definir o método de treinamento. Nesta interface, figura 3.2, é possível inserir todos dados que serão utilizado para treino e teste, e por fim, os resultados e erros são apresentados e disponibilizados para análise e conclusões.

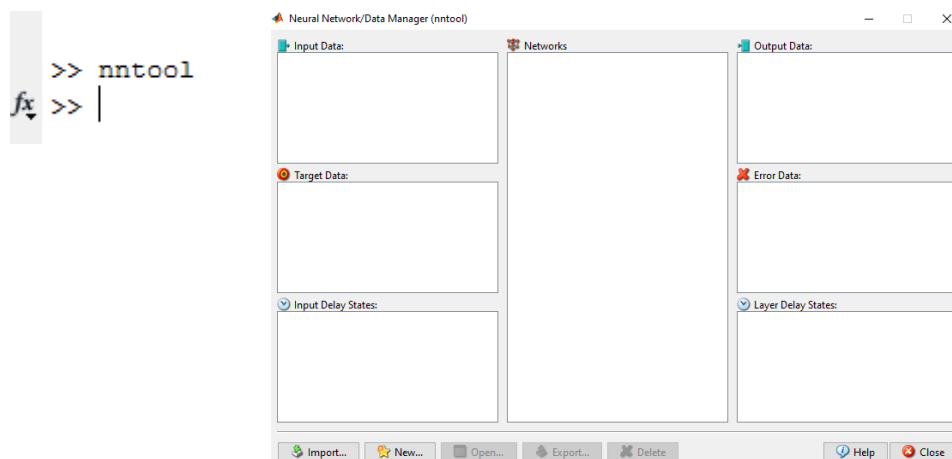


Figura 3.2. Toolbox de redes neurais do Matlab.

3.3. BANCO DE FACES

Levando em conta o padrão de imagens em formato RGB - esclarecido na Seção 2.2, foi utilizada a ferramenta Google Picasa para auxílio à coleta das fotografias para composição do banco de dados, de forma a coletar imagens em que estivessem presentes as faces das quatro classes de interesse. A base de dados possui a distribuição de classes conforme a Tabela 3.1.

Tabela 3.1. Detalhamento global do banco de imagens.

Classes de pessoas	Imagens selecionadas pelo Picasa 3	Faces detectadas pelo método Viola Jones
Outras pessoas	594	495
Pessoa 1 (Hugo)	548	499
Pessoa 2 (Luana)	523	488
Pessoa 3 (Mariana)	503	330
totais	2168	1812

Após a seleção das fotografias originais indexada por tipo de classe através do Picasa, inicia-se o procedimento de extração de *FEATURES*. As imagens são processadas pela função de detecção de face baseada no método Viola-Jones, as faces identificadas são acondicionadas em frames de imagens do tipo RGB de dimensão 60x60x3 pixels.

A seguir, todas as faces são linearizadas em um vetor coluna unitária com 10800 linhas e realocadas em uma só matriz de dados, constituindo um banco de faces representado por uma matriz de dimensão 10800x1812. Em que cada coluna representa uma face/padrão. Nesta etapa os vetores referência com os rótulos correspondentes a cada amostra da base foram gerados também.

Finalmente a base de faces foi segmentada de forma aleatória em duas partes de tamanhos iguais, a saber, 906 faces para compor o conjunto de treino e 906 faces para compor o conjunto de teste, totalizando 1812 faces, conforme a Tabela 3.2.

Tabela 3.2. Detalhamento do quantitativo utilizando nas bases de treinamento e teste.

Descrição	Quantitativo de faces
Base de treinamento	906*
Base de teste	906*
Total	1812

*Observação: para cada base há uma matriz rótulo correspondente com o mesmo quantitativo de faces da base em referência.

3.4. REDUÇÃO DE DIMENSIONALIDADE

A etapa seguinte consiste em comprimir a matriz de dados utilizando o método de Análise de Componentes Principais – PCA, detalhado na Seção 2.5. A função de compressão (*aplicaPCA*) recebe a matriz de dados de treinamento e o parâmetro de compressão desejado (*m_PCA*), i.e. o número de componentes principais, e retorna a matriz de projeção e o nível de energia associado à compressão, que expressa o percentual da variância dos dados que se preserva após o processamento.

Para este trabalho consideramos os seguintes números de componentes principais (m_{PCA}): 50, 100, 200 e 400. Em suma, este fator define a dimensão da camada de entrada da RNA e consequentemente interfere na relação de desempenho do treinamento *versus* qualidade da classificação.

3.5. TREINAMENTO

Findado a etapa de extração e processamento da base, os dados estão prontos para iniciar o processo de aprendizado supervisionado da rede MLP, via *toolbox* de redes neurais do Matlab.

Tendo em mãos a base de dados de treino já processada pela técnica de PCA, a saber, $Input_{MLP_{PCA \times 906}}$, em que o parâmetro PCA corresponde à dimensão de cada amostra e que, no caso, há 906 faces, lembrando que paralelamente está presente o vetor alvo ou *target* com os rótulos da base usada, que será usado como referência no processo de aprendizado, resta somente especificar o número de camadas e demais parâmetros da ferramenta, conforme ilustra a Figura 3.3.

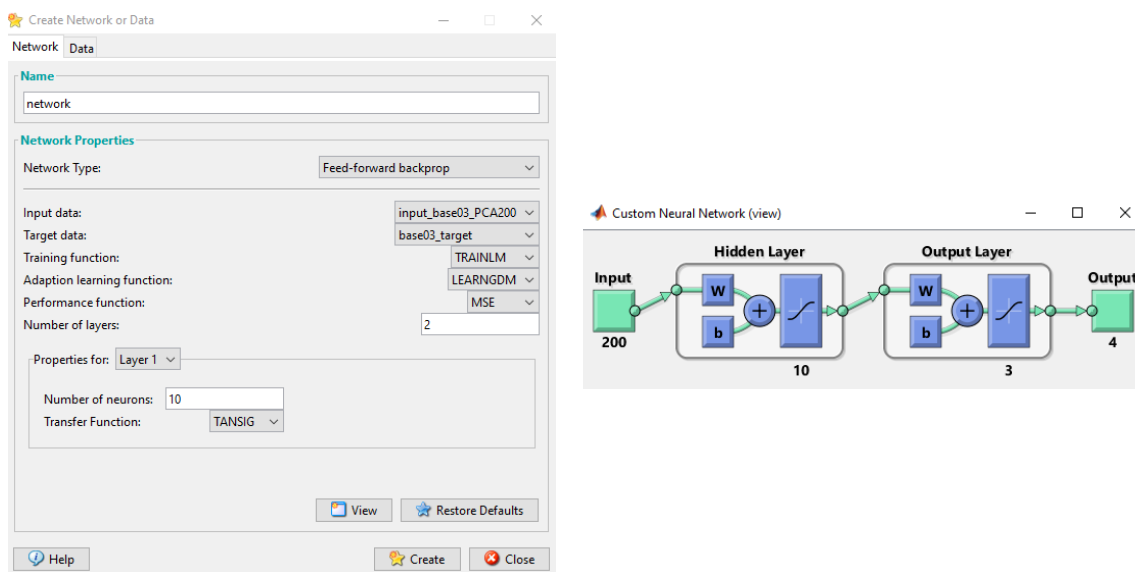


Figura 3.3. Parâmetros de criação e resultado da MLP.

Por exemplo, na Figura 3.3. observe que foram fornecidos os seguintes parâmetros para treinamento da MLP:

- rede do tipo Feed forward backprop (“*backpropagation*”);
- input: base com 906 faces com $m_{PCA}=200$;
- 200 linhas de entrada;

- TRAINLM: algoritmo de treinamento de 2ª ordem Levenberg-Marquadt;
- LEARNNGDM: método de aprendizado que utiliza o gradiente para ajuste dos pesos do neurônio a cada época;
- MSE: fator de ajuste que usa o erro quadrático médio normalizado;
- 2 camadas (sendo uma camada de entrada e uma oculta);
- 10 neurônios por camada;
- função de transferência tangente hiperbólica.

3.6. SIMULAÇÃO DOS ENSAIOS

Para de fato avaliar a metodologia em um conjunto de cenários, foi desenvolvido um script Matlab (vide Apêndice A) pelo qual foram executadas as simulações de treinamento e teste da rede neural. Repare que, no código, a base de dados aplicada na entrada da função é dividida em dois percentuais que podem ser manipulados: um destinado ao treinamento propriamente dito (75%) e outro para o conjunto de validação (25%), utilizado de base para o critério de parada.

Quanto aos cenários para estudo, realizaram-se 20 simulações independentes (treinamento e subsequente teste) para cada um dos seguintes valores de componentes principais, já mencionados anteriormente: 50, 100, 200 e 400. Além disso, os ensaios foram aplicados em dois cenários distintos, a saber:

- Rede MLP com uma camada oculta de 10 neurônios;
- Rede MLP com uma camada oculta de 20 neurônios.

O script de execução do experimento de validação dos resultados se encontra no Apêndice B.

No capítulo seguinte, os resultados gerais são finalmente discutidos.

CAPÍTULO 4

4. RESULTADOS

Neste capítulo, inicialmente, serão apresentados testes práticos de detecção de face e os problemas encontrados nesta etapa, assim como no processo de extração de faces das imagens.

Por ultimo, apresentamos os resultados das primeira tentativas de simulação e em seguida e os problemas em tempo de execução que impossibilitaram a execução das simulações, e finalmente, após o uso de técnicas de redução de dimensionalidade para resolver o problema do tamanho excessivo de entradas na rede neural, apresentaremos as simulações e resultados finais, obtidos a partir da execução dos procedimentos descritos no Capítulo 3.

4.1. TESTE DA DETECÇÃO E EXTRAÇÃO DE FACES

A tarefa de detecção de faces é executada pela função de detecção disponível no próprio Matlab, a qual detecta face utilizando um objeto que retorna as coordenadas na imagem de onde se encontra a face detectada. O código que provê esta funcionalidade está disponível no Apêndice C.

Nas Figuras 4.1 e 4.2 a seguir, verificamos que o objeto “*visio*” presente no código disponível no Apêndice C realiza a detecção de faces da figura a seguir, utilizando o algoritmo de detecção de faces Viola Jones.



Figura 4.1, imagem original.



Figura 4.2, face detectada.

O processo de detecção nem sempre consegue êxito ao que se propõe, vejamos que na Figura 4.3 que foi identificado um falso positivo ao lado direito da face detectada corretamente. Por inspeção visualmente verificamos que na imagem há apenas uma face e o algoritmo retornou indevidamente duas faces. Estas falhas se devem ao erro propagado no cascadeamento dos classificadores fracos, utilizados internamente no algoritmo Viola Jones, conforme visto na Seção 2.4.

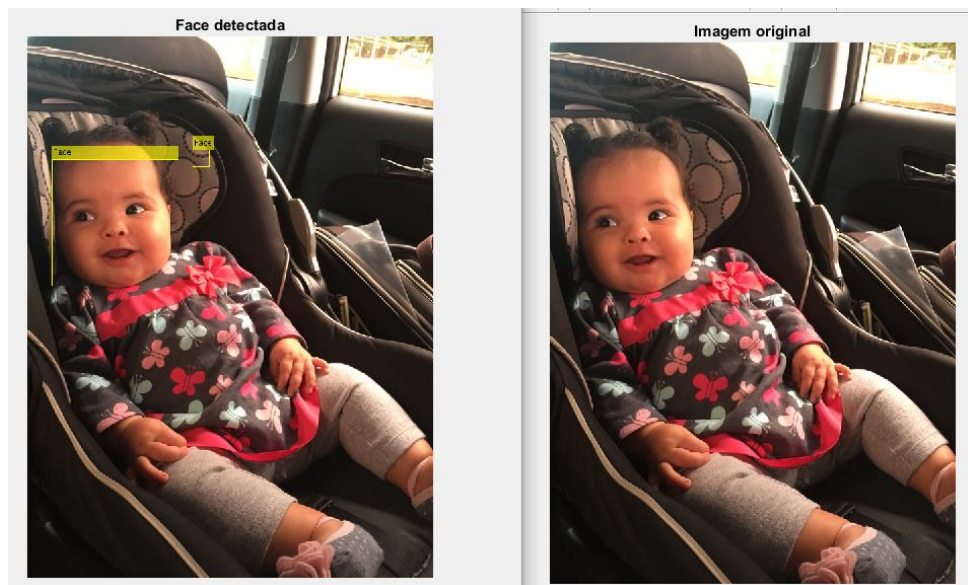


Figura 4.3. Falso positivo detectado

Vale então destacar que o bom desempenho do classificador que este trabalho propõe, está relacionada com a qualidade e variações de atributos resultantes do processo de extração de *FEATURES*, ou seja, na eficiência da tarefa de detecção e extração das faces nas imagens. Faces detectadas equivocadamente e utilizadas na subsequente etapa de aprendizado podem comprometer a acurácia da rede como um todo.

Para fins de ilustração do algoritmo de detecção Viola Jones na localização e reconhecimento de múltiplas faces dentro de cada fotografia, observe as Figuras 4.4, 4.5 e 4.6.



Figura 4.4. Imagem original.



Figura 4.5. Faces detectadas.



Figura 4.6. 14 faces extraídas.

Na imagem original, Figura 4.4, o algoritmo de detecção identificou quatorze faces, as quais foram extraídas e ilustradas na figura 4.6. A etapa seguinte conforme detalhado na subseção 3.2, é linearizar as faces para cada uma compor amostras da matriz de dados não compactada. Paralelamente, o respectivo vetor de rótulos de da amostra também é concebido e armazenado. Doravante, a base de dados é compactada no processo de compressão utilizando o método PCA e o resultado da operação linear está pronto e é apresentado à entrada da MLP para realização do treinamento.

4.2. ENSAIOS DE RECONHECIMENTO UTILIZANDO O CLASSIFICADOR

Finalmente, consideremos os resultados dos ensaios de treinamento e teste do classificador de faces baseado na rede MLP. Vale registrar que o vetor de rótulos tem dimensão $[4 \times 1]$, onde cada linha está associada a uma classe presente ou não no vetor de entrada, possuindo valores discretos -1 (classe não presente) ou 1 (classe presente). Dado que, naturalmente, a saída da rede compreende valores reais, os mesmos são truncados para 1 quando positivos e para -1 em caso contrário. A Figura 4.8 descreve mais o processo de teste para uma imagem.

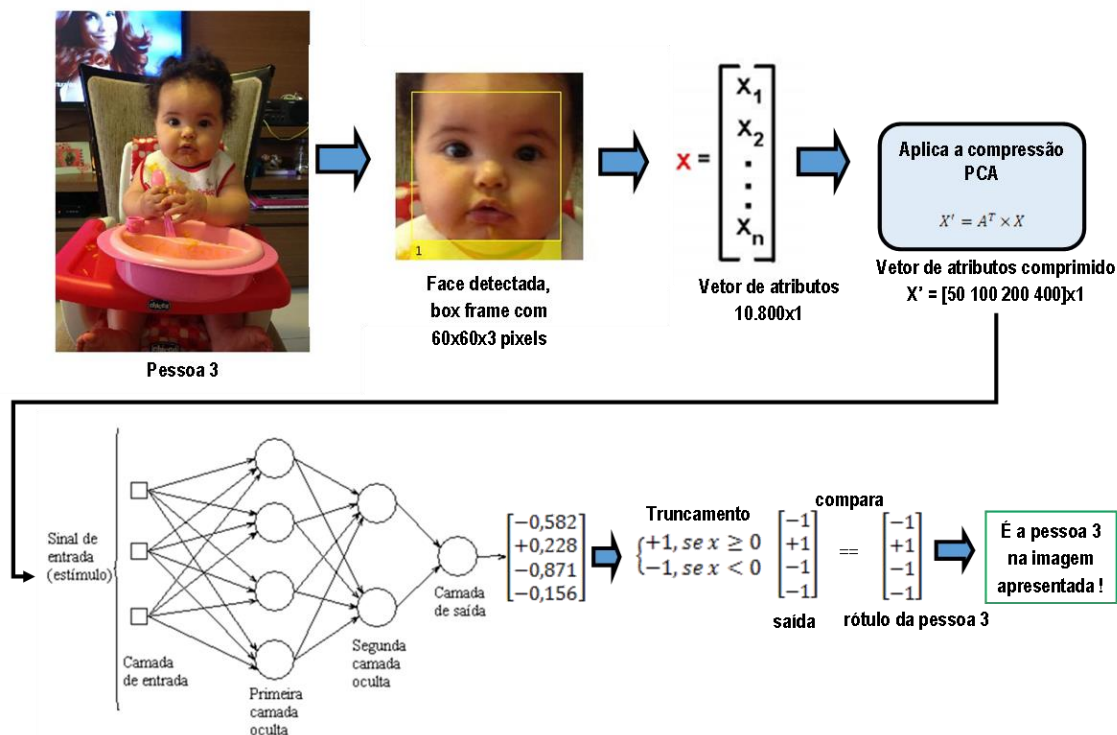


Figura 4.8. Teste de reconhecimento com uma face.

Doravante, o resultado do ensaio utilizando a base de dados composta por 906 faces no treino e 906 faces no teste, com as especificações dos ensaios configuradas conforme vistos na subsecção 3.6 **Erro! Fonte de referência não encontrada.** Após total de 20 simulações, utilizando cada fator de compressão, logrou-se os valores médios de acerto, com respeito ao conjunto de teste, observados na Tabela 4.1. A saber, o parâmetro Energia da informação preservada expressa o quanto do padrão original foi preservado após o processo de redução de dimensionalidade. Valores iguais a 1 indicam que toda informação foi preservada.

Tabela 4.1. Resultado do ensaio com MLP de uma camada oculta de dez neurônios.

MLP com 1 camada oculta de 10 neurônios		
Fator de compressão PCA	Percentual de Acerto obtido	Energia da informação preservada
50	66%	0,87
100	69%	0,92
200	70%	0,96
400	65%	0,99

O fator de compressão é o numero de *FEATURES*. O gráfico presente na Figura 4.9 foi obtido a partir dos dados da Tabela 4.1 e ilustra de outra forma a dispersão em colunas dos resultados produzidos.

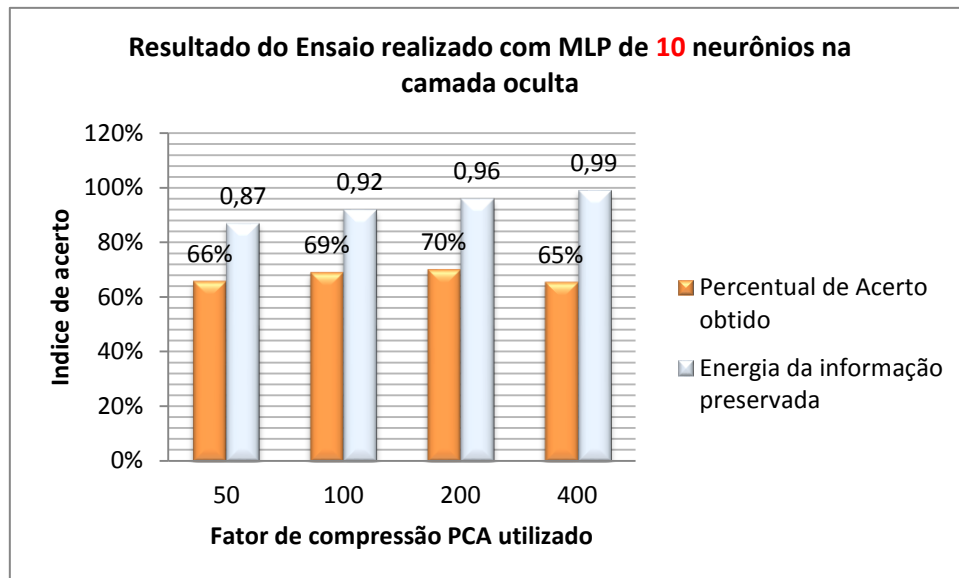


Figura 4.9. Dispersão em colunas da taxa de acerto em função do fator de compressão PCA sobre a informação original.

Observe que no gráfico a simulação contendo o fator de compressão $m_{PCA}=200$ e com energia de informação 0,96, indicou a melhor taxa de acerto com 70% de acerto no reconhecimento nas faces apresentadas. Já no segundo cenário, descrito na Tabela 4.2, resultados distintos foram obtidos.

Tabela 4.2. Resultado do ensaio com MLP de uma camada oculta de vinte neurônios.

MLP com 1 camada oculta de 20 neurônios		
Fator de compressão PCA	Percentual de	Energia de informação preservada
50	70%	0,87
100	73%	0,92
200	69%	0,96
400	67%	0,99

Por fim, o gráfico presente na Figura 4.10, foi obtido a partir dos dados da Tabela 4.2 e ilustram de outra forma a dispersão em colunas dos dados produzidos na rede com uma camada oculta de 20 neurônios.

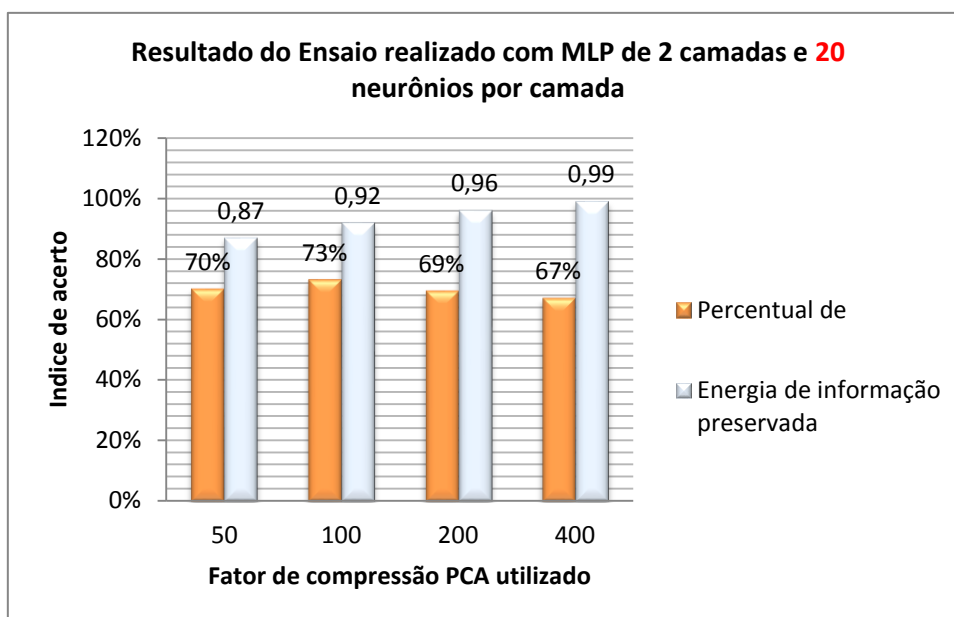


Figura 4.10. Dispersão em colunas da taxa de acerto em função do fator de compressão PCA sobre a informação original.

Neste contexto, com uma rede de 20 neurônios por camada, a melhor marca de percentual de acerto no conjunto de teste foi atingida ao utilizar o fator de compressão $m_PCA=100$, em que o índice de energia de informação preservada foi de 0,92. Ou seja, a segunda arquitetura com 20 neurônios trouxe um ganho de desempenho no acerto maior que na primeira arquitetura contendo 10 neurônio.

CAPÍTULO 5

5. CONCLUSÕES

5.1. ANÁLISE DOS RESULTADOS E CONCLUSÕES

Este trabalho estudou o emprego de redes neurais artificiais para realizar reconhecimento de faces. A metodologia empregada para solução do problema envolveu também as técnicas de extração de *FEATURES* como a detecção de faces Viola Jones e Análise de Componentes Principais, de forma a tornar a base de treinamento de dimensão compatível aos aspectos de uso adequado dos recursos computacionais.

Neste estudo, a partir dos resultados obtidos notou-se que o desempenho da rede neural artificial, em termos da capacidade de generalização, está fortemente ligada a fatores tais como: qualidade da base de treino, distribuição das classes, representatividade dos padrões, etc..

Quanto à representatividade das classes no conjunto de treino, a capacidade de reconhecimento de faces e objetos por seres humanos é influenciada por alguns parâmetros, como a fonte de iluminação e distorções na imagem, que afetam de forma diferente o reconhecimento das faces e objetos. Em geral, o reconhecimento de faces é mais sensível às variações de direção da iluminação, rotação em profundidade, rotação no plano, polaridade de contraste e métrica de variação (proporção ou grau de curvatura). Essas mesmas variáveis devem ser consideradas no estudo de reconhecimento automático de faces

Em relação ao processo de aprendizado da rede, verificou-se que não há como se obter uma solução única determinística e fechada, pois a função de erro é calculada a cada época e os pesos sinápticos ajustados empiricamente a cada época de treino, portanto para uma mesma estrutura de rede com a mesma base de treino, os resultados não serão idênticos. Também é importante citar que o valor dos pesos sinápticos são iniciados aleatoriamente a cada vez que a rede é criada. Logo, cada treino é único quanto ao número de épocas e erro quadrático médio obtido.

Em relação ao processo de redução de dimensionalidade empregado, o uso da técnica de PCA não comprometeu aparentemente o desempenho nos resultados da rede, visto que índices de acerto foram próximos em simulações com vários fatores de

compressão distintos, portanto esta técnica pode ser empregada em aplicações práticas.

A quantidade de neurônios nas camadas de entrada e saída da rede é geralmente influenciada pela natureza do problema abordado. Entretanto, a quantidade de neurônios em cada camada é, em geral, uma variável de projeto. Aumentando-se a quantidade de neurônios nas camadas intermediárias, aumenta-se, de certa forma, a capacidade de mapeamento não-linear da rede, razão que observado nos resultados melhores utilizando rede com 20 neurônio por camada. O que leva a intuir que os padrões apresentados possuem mapeamento não linear entre si. Não obstante, é preciso tomar cuidado para que o modelo não se sobreajuste aos dados, pois, nesse caso, a rede poderá sofrer de sobre treinamento (*overfitting*).

Por outro lado, uma rede com poucos neurônios na camada intermediária pode não ser capaz de aproximar o mapeamento desejado, causando um *underfitting*: situação em que a rede possui baixa capacidade de acerto tanto no treino quanto no teste.

Contudo, a simulação apresentou uma razoável capacidade de generalização tanto com 10 ou 20 neurônios e os valores obtidos são a princípio satisfatórios. Naturalmente, perspectivas como o uso de faces com maior representatividade e uma melhor distribuição de faces podem melhorar os índices de acerto obtidos pelo método. Ou ainda, pode-se aumentar o numero de neurônio, o que certamente consumirá mais recursos de máquina e demandará maior tempo de treino e teste.

5.2. TRABALHOS FUTUROS

Apesar de os resultados obtidos na detecção de face utilizando redes neurais terem sido inicialmente satisfatórios, o índice de acerto precisaria ser maior para fins de competitividade com as estratégias estado-da-arte. Nesse sentido, o uso agregado de mais *FEATURES* como detecção de curvas, tonalidade de cor de pele e proporções nas medidas faciais, certamente proporcionaria maior representatividade de padrões na base de treinamento. Afinal a melhoria da qualidade dos atributos certamente retornaria melhores índices de acerto. Ainda sim, o aumento de números de neurônios contribuirá com resultados melhores, entretanto esta sugestão impactará diretamente no consumo de recursos computacionais.

Por último, uma aplicação real utilizando a solução discutida requer a implementação de uma interface gráfica, para que a aplicação pudesse ser testada por usuários comuns, em aplicativos móveis ou estações de trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Freire. Gilberto, 1900-1987, Homens, engenharias E Rumos Sociais, Editora É Realizações, 2010.
- [2] Zhao, W.; Chellappa, R.; Phillips, P. J. And Rosenfeld, A. Face Recognition: A Literature Survey. ACM Computing Surveys, vol. 35, no 4, p. 399-458, 2003.
- [3] III Simpósio Brasileiro de Ciências Geodésicas e Tecnologias da Geoinformação, Recife - PE, Uso de imagens do satélite cbers 2b para detecção de áreas desmatadas no norte de minas gerais, p. 001-004, 27-30 de Julho de 2010.
- [4] Anais XVII Simpósio Brasileiro de Sensoriamento Remoto - SBSR, João Pessoa-PB, Brasil, Uso de imagens hyperion e dados de espectroscopia para caracterização de poluentes traços provenientes da indústria fabril de fertilizantes na cidade de Rio Grande, RS, 25 a 29 de abril de 2015, INPE.
- [5] SEGeT – Simpósio de Excelência em Gestão e Tecnologia, Resende – RJ, Visão computacional aplicada à verificação do nível de líquido para utilização em automação industrial, 31/out a 01 de novembro de 2014.
- [6] Software de reconhecimento facial auxilia polícia em caso de tumultos, Disponível em: < <http://www.idelity.com.br/blog/?p=55> > Acesso em: 21 junho. 2016.
- [7] MVA2009 IAPR Conference on Machine Vision Applications, Yokohama, JAPAN, An Intelligent Night Vision System for Automobiles, page 15-3, May 20-22, 2009.
- [8] C. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.
- [9] Gonzales e Woods, 2002, Digital Image Processing 3rd ed, McGraw Hill Education, 2013.
- [10] de Castro, L. N., Análise e Síntese de Estratégias de Aprendizado para Redes Neurais Artificiais, Tese de Mestrado, FEEC - Unicamp, 1998.
- [11] Paul. Viola; M. Jones. Rapid object detection using a boosted cascade of simple *features*. In Proc. Of CVPR, 2001.
- [12] Haykin. Simon S. Neural networks and learning machines. 3rd ed. New York: Prentice Hall, c2009. xxx, 906 p. 2009.

- [13] Kolen J.F., & Pollack J.B., "Back Propagation is Sensitive to Initial Conditions", Technical Report TR 90-JK-BPSIC, 1990, disponível em : <ftp://ftp.cs.brandeis.edu/pub/faculty/pollack/bpsic.tar.Z>.
- [14] Instituto Federal de Educação, Ciência e Tecnologia. Visão computacional aplicada à verificação do nível de líquido para utilização em automação industrial. Disponível em: http://www.aedb.br/seget/arquivos/artigos09/191_AC%20-%20visao.pdf. Acesso em: 21 de jun. 2016.
- [15] Anais XIV Simpósio Brasileiro de Sensoriamento Remoto, Natal, Brasil, Extração de Atributos de Forma para Classificação de Imagens de Alta Resolução do Satélite HRC/CBERS-2B, p. 7015-7022, 25-30 de abril de 2009.
- [16] Marques Filho, Ogê; Vieira Neto, Hugo. Processamento Digital de Imagens, Rio de Janeiro: Brasport, 1999.
- [17] Nigrin, A. Neural Networks for Pattern Recognition, Cambridge, MA: The MIT Press, 1993.
- [18] Biederman, I.; Kalocsai, P. Neural and Psychophysical Analysis of Object and Face Recognition. Springer. 1998. Disponível em: <http://www.kalocsai.net/publications/kalo_nato2.pdf>. Acesso em: 08 abr. 2016
- [19] Aapo Hyvarinen, Juha Karhunen, and Erkki Oja, Independent Component Analysis, Final version of 7 March 2001.
- [20] Eder A. Penharbel, Ricardo C. Destro, Flavio Tonidandel, Reinaldo A.C. Bianchi . Filtro de Imagem Baseado em Matriz RGB de Cores Padrão para Futebol de Robôs, Disponível em: <http://fei.edu.br/~flaviot/pub_arquivos/Enri04.PDF>. Acesso em 30 jul. 2016.
- [21] Duda, R. O., Hart, P. E., & Stork, D. G. (2001). Pattern Classification (2o ed). New York, USA: Wiley-Interscience.
- [22] Crow, F. Summed –area tables for texture mapping. In: ACM Siggraph 84, p. 207-212, July 1984.

APÊNDICE A – CÓDIGO DE SIMULAÇÃO

```
function [ Ind_Acerto, energia ] = Ensaio(n,m_PCA)
% Ensaio(numero de neuronios por camada, fator de compressão PCA)
%Realiza ensaios dado:
% n: numero de neuronio na MLP
% Retorna: indice de acerto da MLP analisada
% Essa função só pode ser executada se houver os seguintes

% arquivos no workspace em questão:
load base03_teste;

switch m_PCA
    case 50
        load PCA50_treino;
        inputs=Xpca50_treino;
        targets=target_base03_TREINO;
        inputs_teste= A50_treino'*input_base03_TESTE;
        targets_teste=target_base03_TESTE;
        energia = alpha50_treino;
    case 100
        load PCA100_treino;
        inputs=Xpca100_treino;
        targets=target_base03_TREINO;
        inputs_teste= A100_treino'*input_base03_TESTE;
        targets_teste=target_base03_TESTE;
        energia = alpha100_treino;
    case 200
        load PCA200_treino;
        inputs=Xpca200_treino;
        targets=target_base03_TREINO;
        inputs_teste= A200_treino'*input_base03_TESTE;
        targets_teste=target_base03_TESTE;
        energia = alpha200_treino;
    case 400
        load PCA400_treino;
        inputs=Xpca400_treino;
        targets=target_base03_TREINO;
        inputs_teste= A400_treino'*input_base03_TESTE;
        targets_teste=target_base03_TESTE;
        energia = alpha400_treino;
    otherwise
        disp('Parametro PCA inválido')
end

% houseInputs - input data.
% houseTargets - target data.

% Create a Fitting Network
hiddenLayerSize = n;
net = fitnet(hiddenLayerSize);

% Set up Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 75/100;
net.divideParam.valRatio = 25/100;
net.divideParam.testRatio = 0;% 15/100;

% Train the Network
```

```
[net,tr] = train(net,inputs,targets);

% Test the Network
outputs = net(inputs_teste);
errors = gsubtract(outputs,targets_teste);
performance = perform(net,targets_teste,outputs);

%Indice de acerto no teste
Ind_Acerto = resultado( outputs, targets_teste );
End
```


APÊNDICE B – SCRIPT DE EXECUÇÃO DO ENSAIO E RESULTADOS

```
%% Produção do ensaio
%limpeza do chache e dados no Worksapce
clc;clear all;close all;

% Gera arquivo .txt para saída dos dados
arq = fopen('exemplo.txt','wt');
% Parâmetro de compressão PCA
% PCA=[50 100 200 400];
PCA=100;
% n=[10 20]; % numero de neuronios por camada
n=20;
rep=1; % numero de repetições do ensaios por parametro PCA

for neu=n
    fprintf(arq,'Ensaio usando %d por camada\n',neu);
    fprintf(arq,'\nPCA\tAcerto\tEnergia\n');
    for i=1:length(PCA);
        for j=1:rep
            [ Ind_Acerto, energia ] = Ensaio(neu,PCA(i));
%            [ Ind_Acerto ] = rand(1); energia=rand(1);
            fprintf(arq,'%d\t%.2f\t%.2f\n',PCA(i),Ind_Acerto,energia);
        end
    end
    fprintf(arq,'\n');
end
fclose(arq);
disp('fim do ensaio!');
```

APÊNDICE C – EXEMPLO DE DETECÇÃO DE FACE UTILIZANDO O MÉTODO VIOLA JONES

```
%detecção de face frontal
I = imread('12092688_881975948561069_1933499145_n.jpg');
%detecção de face frontal
faceDetector = vision.CascadeObjectDetector('FrontalFaceCART');
%retorna as coordenadas de onde se encontra a face na imagem
bboxes = step(faceDetector, I);
%% Extraíndo a face da imagem
IFaces = insertObjectAnnotation(I, 'rectangle', bboxes, [1:size(bboxes,1)]);
imshow(IFaces)
```