

INSTITUTO FEDERAL DO ESPÍRITO SANTO

CURSO DE ENGENHARIA ELÉTRICA

WANDERCLEYSON MARCHIORI SCHEIDEGGER

**CONTROLE DE ACESSO DE PESSOAS POR RECONHECIMENTO FACIAL
UTILIZANDO REDES NEURAIS ARTIFICIAIS**

VITÓRIA

2017

WANDERCLEYSON MARCHIORI SCHEIDECKER

**CONTROLE DE ACESSO DE PESSOAS POR RECONHECIMENTO FACIAL
UTILIZANDO REDES NEURAIS ARTIFICIAIS**

Trabalho de Conclusão de Curso apresentado à
Coordenadoria do Curso de Engenharia Elétrica do
Instituto Federal do Espírito Santo, como requisito parcial
para a obtenção do título de Bacharel em Engenharia
Elétrica.

Orientadora: Prof.^a. Dra. Mariana Rampinelli Fernandes

VITÓRIA
2017

(Biblioteca Nilo Peçanha do Instituto Federal do Espírito Santo)

S318c Scheidegger, Wandercleyson Marchiori.

Controle de acesso de pessoas por reconhecimento facial
utilizando redes neurais artificiais / Wandercleyson Marchiori
Scheidegger. – 2017.

62 f. : il. ; 30 cm

Orientador: Mariana Rampinelli Fernandes.

Monografia (graduação) – Instituto Federal do Espírito Santo,
Coordenadoria de Engenharia Elétrica, Curso Superior de Engenharia
Elétrica, 2017.

1. Sistemas de reconhecimento de padrões. 2. Redes neurais
(computação). 3. Visão por computador. 4. Engenharia elétrica. I.
Fernandes, Mariana Rampinelli. II. Instituto Federal do Espírito Santo.
III. Título.

CDD 21 – 006.4

WANDERCLEYSON MARCHIORI SCHEIDEGGER

**CONTROLE DE ACESSO DE PESSOAS POR RECONHECIMENTO FACIAL
UTILIZANDO REDES NEURAIS ARTIFICIAIS**

Trabalho de Conclusão de Curso apresentado à
Coordenadoria do Curso de Engenharia Elétrica do
Instituto Federal do Espírito Santo, como requisito parcial
para a obtenção do título de Bacharel em Engenharia
Elétrica.

Aprovado em 22 de dezembro de 2017.

COMISSÃO EXAMINADORA

Mariana Rampinelli Fernandes

Prof.ª Dra. Mariana Rampinelli Fernandes

Instituto Federal do Espírito Santo

Orientadora

Mário Mestria

Prof. Dr. Mário Mestria

Instituto Federal do Espírito Santo

Clebeson Canuto dos Santos

Me. Clebeson Canuto dos Santos

Universidade Federal do Espírito Santo

DECLARAÇÃO DO AUTOR

Declaro, para fins de pesquisa acadêmica, didática e técnico-científica, que este Trabalho de Conclusão de Curso pode ser parcialmente utilizado, desde que se faça referência à fonte e ao autor.

Vitória, 22 de dezembro de 2017.



Wandercleyson Marchiori Scheidegger

AGRADECIMENTOS

Agradeço primeiramente a Deus, pois sem Ele eu não seria nada. Em seguida à minha família, em especial a minha esposa Vanessa Dias Scheidegger, que apesar das dificuldades, acreditou em mim e investiu na minha educação. Sou grato a minha orientadora, Mariana Rampinelli Fernandes, que me guiou com muita atenção e presteza no desenvolvimento e melhoramento deste trabalho. Aos meus amigos de curso pela companhia, ajuda e pela vivência maravilhosa que me proporcionaram. Também sou grato a todos os professores que passaram pela minha vida e compartilharam seus conhecimentos comigo, me dando a oportunidade de me tornar uma pessoa melhor e mais crítica.

RESUMO

Por sofrer mudanças de cena nas imagens como: variação de pose, de expressão facial ou de iluminação, o reconhecimento facial em aplicações do cotidiano, não possui solução trivial e ainda desperta o interesse de pesquisadores da área de aprendizado de máquina. Este trabalho propõe uma arquitetura de rede neural convolucional para reconhecimento facial e sua utilização em uma aplicação que simula o controle de acesso a ambientes. A arquitetura da rede foi desenvolvida, treinada e validada por meio de dois *datasets* de imagens faciais disponíveis na literatura. Para a criação de um novo banco de imagens faciais, que foi utilizado na aplicação de simulação de controle de acessos, um aplicativo na plataforma Android foi desenvolvido para que voluntários enviassem suas fotos e vídeos por meio dele, e de forma automatizada as imagens e vídeos fossem processadas e destinadas ao treinamento da rede. A rede precisou de autoaprendizado nos períodos iniciais de sua operação. O auto aprendizado realizado com as amostras utilizadas na validação, conferiu à rede 97,5% de precisão já no terceiro treinamento de autoaprendizado. Para eliminar falsos positivos na validação, uma nova rede de reconhecimento de apenas uma pequena porção da face, a região compreendida entre os olhos, nariz e boca foi treinada com o objetivo de tornar a saída binária do programa de validação. Dessa forma, o acesso só é validado com o reconhecimento correto das duas redes. Essa estratégia eliminou a incidência de falsos positivos.

Palavras-chave: Reconhecimento facial. Aprendizado de máquina. Rede neural convolucional. *Deep learning*. Controle de acesso.

ABSTRACT

Pictures of a same person may suffer variations such as change of pose, facial expression or lighting. For this reason, the facial recognition in everyday applications has no trivial solution and still arouses the interest of researchers in the area of machine learning. This paper proposes a Convolutional Neural Network architecture for facial recognition and its use in an application that simulates access control for environments. The network architecture was designed, trained and validated through two datasets of facial images available in the literature. An application for Android platform was used to create a new facial images database. From this application, volunteers would send their photos and videos. In an automated way, the images and videos were processed to the network training. The network needed self-learning in the initial periods of operation. The self-learning performed with the samples used in the validation provided to the network 97.5% accuracy already in the third self-learning training. In order to eliminate false positives in validation, a new recognition network of only a small portion of the face, the region between the eyes, nose and mouth, was trained with the goal of making the binary output of the validation program. Thus, the access is just validated with the recognition of both networks. This strategy eliminated the incidence of false positives.

Keywords: Facial recognition. Machine learning. Convolutional neural network. Deep learning. Access control.

LISTA DE SIGLAS

CNN - *Convolutional Neural Networks*

HOG - Histograma de Gradientes Orientados

MLP - *Multi-Layer Perceptron*

MRC - *Maximal Rejection Classifier*

PCA - Análise de componentes principais

RPCA - Análise robusta de componentes principais

SUMÁRIO

1	INTRODUÇÃO	10
1.1	MOTIVAÇÃO	10
1.2	OBJETIVOS	11
1.3	ORGANIZAÇÃO DA MONOGRAFIA.....	11
2	REVISÃO TEÓRICA	13
2.1	RECONHECIMENTO FACIAL	13
2.2	REDES NEURAIS	14
2.2.1	Neurônio Biológico	15
2.2.2	Neurônio Artificial	15
2.2.3	Reconhecimento de objetos com redes neurais artificiais	17
2.3	DEEP LEARNING	19
2.4	REDES NEURAIS CONVOLUCIONAIS.....	19
2.4.1	Visão Geral.....	19
2.4.2	Camada convolucional	22
2.4.3	Pooling layer.....	23
2.4.4	Fully connected	23
2.5	TENSORFLOW	24
2.6	APLICATIVOS MÓVEIS	25
3	REDE NEURAL ARTIFICIAL PARA RECONHECIMENTO FACIAL	26
3.1	IMPLEMENTAÇÃO DA REDE	26
3.2	ESCOLHA DE BANCOS DE IMAGENS FACIAIS DISPONÍVEIS NA LITERATURA PARA A VALIDAÇÃO DA REDE	29
3.3	TREINAMENTO DA REDE E OS RESULTADOS OBTIDOS.....	32
3.3.1	Rede treinada com as imagens do ORL Database	32
3.3.2	Rede treinada com as imagens do FEI Database	37
4	REDE PARA RECONHECIMENTO FACIAL	39
4.1	APLICATIVO	39
4.1.1	Cadastro do Usuário	39
4.1.2	Captura e armazenamento de fotos	40
4.1.3	Captura e armazenamento de vídeos	41
4.2	PROCESSAMENTO DAS AMOSTRAS DE TREINAMENTO	43
4.2.1	Visão geral	43

4.2.2	Processamento das fotos cadastradas	44
4.2.3	Processamento dos vídeos cadastrados	45
4.3	CRIAÇÃO DO BANCO DE IMAGENS FACIAIS DE CELEBRIDADES	46
4.4	TREINAMENTO DA REDE	48
5	SIMULAÇÃO DA VALIDAÇÃO DE ACESSO EM TEMPO REAL	50
5.1	IMPLEMENTAÇÃO DO PROGRAMA	50
5.2	TESTES DE ACESSO EM TEMPO REAL	51
5.3	ESTRATÉGIA PARA RESOLUÇÃO DO PROBLEMA DE FALSOS POSITIVOS	53
6	CONCLUSÃO	56
6.1	CONCLUSÕES GERAIS	56
6.2	TRABALHOS FUTUROS	57
	REFERÊNCIAS	59

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

O problema do reconhecimento automático de rostos humanos motiva pesquisadores de áreas como processamento de imagens, reconhecimento de padrões, redes neurais, visão computacional e computação gráfica. A necessidade de sistemas de fácil utilização que possam proteger dados e a privacidade sem perder a identidade entre uma grande quantidade de números é cada vez maior (ZHAO et al., 2003).

Apesar dos esforços dos pesquisadores sobre o tema, o reconhecimento facial em aplicações do cotidiano, como por exemplo a validação de identidade, ainda é uma tarefa de solução difícil, já que as imagens faciais sofrem mudanças na cena, como variação de pose, de expressão facial ou de iluminação (ARYA; ADARSH, 2015).

Até meados da década de 1990, as pesquisas sobre reconhecimento facial concentraram-se na segmentação da face a partir de técnicas simples ou complexas. Essas técnicas incluíam o uso de um modelo de face inteira e um modelo baseado em características deformáveis, como cor de pele. Esses métodos geraram problemas de detecção ocasionados, em sua maioria, por variações de pose, iluminação precária entre outros (ZHAO et al., 2003).

Desde os anos 90, redes neurais artificiais têm sido amplamente utilizadas com bom desempenho em aplicações relacionadas ao reconhecimento de face e o algoritmo *backpropagation*, que é utilizado no *Multilayer Perceptron* (Perceptron Multi Camadas - MLP), é um dos métodos mais amplamente utilizados nesse domínio (BOUGHRARA et al., 2014).

Outra técnica que ganha destaque nos dias atuais, é o *deep learning* (aprendizado profundo), que tem sido utilizado no reconhecimento de padrões e visão computacional principalmente com a abordagem de redes neurais convolucionais (*Convolutional Neural Networks* - CNN) (LIU et al., 2015).

Como será visto na revisão literária deste trabalho, inúmeras técnicas foram pesquisadas e utilizadas a fim de resolver o problema do reconhecimento facial, mas a solução de problemas relacionados a variações ocorridas nos dados de entrada da

rede, como nas imagens a serem analisadas neste trabalho, pesquisas atuais direcionam para a utilização das CNN (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

Vale ressaltar também que a escolha da CNN para o desenvolvimento deste trabalho apoia-se na premissa de que uma rede neural convolucional é um MLP desenhado especificamente para reconhecer formas bidimensionais com um alto grau de invariância à tradução, escalonamento e distorção. Estas são exatamente as características de imagens faciais adquiridas de forma não supervisionada, como as que foram adquiridas e utilizadas no escopo deste trabalho. (LECUN; BENGIO, 2003).

1.2 OBJETIVOS

O objetivo deste trabalho é a implementação e treinamento de uma rede neural convolucional de reconhecimento facial para controle de acesso de pessoas. Esse sistema de reconhecimento pode ser utilizado para validação de ponto de trabalho, presença em eventos entre outros. Para isso, será desenvolvido um banco de imagens faciais de celebridades e um aplicativo de celular utilizando a plataforma Android para a aquisição das imagens de treinamento. A escolha da referida plataforma visa diminuir o tempo no processo de cadastramento dos usuários e na captura das imagens para o treinamento da rede. Também será possível comparar a efetividade da aquisição de imagens por fotografia e por captura de *frames* de vídeo, bem como mostrar a confiabilidade na validação do acesso em tempo real utilizando uma câmera de baixa qualidade e poucas amostras de treinamento.

1.3 ORGANIZAÇÃO DA MONOGRAFIA

O trabalho está dividido em seis capítulos, os quais estão estruturados da seguinte maneira:

- Capítulo 1: apresenta como o projeto está estruturado, bem como os aspectos motivadores e os objetivos a serem alcançados com o trabalho.
- Capítulo 2: apresenta um capítulo de uma breve revisão teórica sobre os principais assuntos abordados durante o projeto. Destes, podem ser citados: um apanhado das principais técnicas de reconhecimento facial utilizadas na literatura recente, conceitos importantes sobre Redes Neurais Artificiais e

aprendizagem profunda, bem como a principais tecnologias em pauta neste trabalho, as Redes neurais Convolucionais e biblioteca *TensorFlow*.

- Capítulo 3: descreve todas as etapas de implementação, configuração, treinamento da rede neural convolucional arquitetada neste trabalho, assim como explicita o processo de validação por meio de dois bancos de imagens faciais da literatura.
- Capítulo 4: versa sobre a rede convolucional desenvolvida, começando com o processo de criação do aplicativo de cadastro, captura e envio de fotos e vídeos, mostrando todas as etapas desde a formação do cadastro até os métodos e tecnologias utilizadas na captura das amostras por fotografia e por vídeo. Além disso, explica a metodologia adotada na criação do banco de imagens faciais a partir de imagens disponíveis na internet. Demonstra também, todas as etapas do processamento automatizado das imagens finalizando com os treinamentos efetuados e a apresentação de seus resultados.
- Capítulo 5: aborda as etapas de desenvolvimento do sistema de reconhecimento que simula o acesso em tempo real, assim como os testes realizados e seus resultados. Por fim, foi apresentada a estratégia para lidar com o problema de falsos positivos na validação, assim como apresenta os resultados dos testes realizados com a versão final do sistema de acesso criado.
- Capítulo 6: apresenta as conclusões obtidas e os trabalhos futuros a serem desenvolvidos a partir do sistema criado neste trabalho.

2 REVISÃO TEÓRICA

2.1 RECONHECIMENTO FACIAL

Detecção e reconhecimento facial, há muito tempo, são temas de pesquisas (HAN et al., 1997). Técnicas de detecção de face são apresentadas na literatura, tanto para reconhecimento facial quanto para outras aplicações. Han et al. (1997) apresentam um método de detecção ocular no qual os pixels dos olhos são localizados e preservados, e os pixels restantes são apagados. Assim a região dos dois olhos é identificada com mais facilidade e precisão.

Técnicas de reconhecimento de padrões têm sido aplicadas no reconhecimento facial, entre elas está a análise de componentes principais (PCA), que é uma técnica amplamente utilizada para identificar padrões em dados de dimensões elevadas (BORADE; DESHMUKH; RAMU, 2016). O PCA tem como objetivo reduzir um grande conjunto de variáveis para um pequeno conjunto de novas variáveis compostas que melhor representam o espaço de grandes características com uma perda mínima de informações (JOLLIFFE, 2002).

Craw e Cameron (1992) obtiveram códigos por mapeamento da textura da imagem das faces para uma forma padrão e, em seguida, as formas das texturas foram gravadas. A análise de componentes principais reduziu os dados a serem armazenados e também melhorou sua eficácia na descrição dos rostos.

A análise robusta de componentes principais (RPCA) é uma variação do PCA, e tem muitas aplicações reais no reconhecimento facial (ZHAO et al., 2014). Também oferece na área de reconhecimento facial, uma maneira de remover sombras em imagens de rostos (CANDÉS et al., 2009).

Outra técnica utilizada no processamento de imagens é o *Histogram of Oriented Gradients* (histograma de gradientes orientados - HOG), que é um descritor de recurso utilizado em visão computacional e tarefas de processamento de imagem para a detecção de objetos representando padrões locais em cada pixel por meio de orientações de gradiente (DALAL; TRIGGS, 2005).

Em (BOUGHRARA et al., 2014), foi proposto um detector de face usando uma arquitetura eficiente baseada em uma rede neuronal MLP e *Maximal Rejection Classifier* (classificador de rejeição máxima - MRC). A abordagem proposta melhorou significativamente a eficiência e a precisão da detecção em comparação com as técnicas tradicionais, como o HOG. Para reduzir o custo total de computação, a rede neural foi organizada em um pré-estágio, que é capaz de rejeitar a maioria dos padrões de não-face nos fundos da imagem, melhorando significativamente a eficiência de detecção global enquanto mantém a precisão de detecção.

Além dos inúmeros métodos e técnicas de detecção e reconhecimento de objetos e faces que não foram citados anteriormente, um método em especial tem sido bastante pesquisado e desenvolvido. Esse método utiliza o conceito de aprendizagem profunda com a utilização das redes neurais convolucionais (ZEILER; FERGUS, 2014).

2.2 REDES NEURAIS

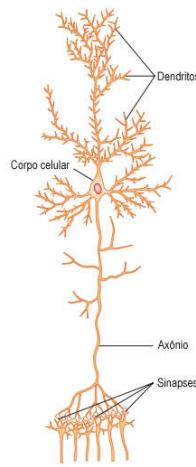
O trabalho com as redes neurais artificiais, referido comumente como redes neurais, foi motivado desde o seu início pelo reconhecimento de que o cérebro humano computa de uma maneira diferente do computador convencional. O cérebro é um computador altamente complexo, não linear e paralelo. Ele tem a capacidade de organizar seus constituintes estruturais, os neurônios, e executar cálculos, como reconhecimento de padrões, muitas vezes mais rápido do que o computador digital. Para ser específico, o cérebro realiza rotineiramente tarefas de reconhecimento perceptivo, por exemplo o reconhecer de uma face familiar embutida em uma cena desconhecida, em aproximadamente 100 a 200 ms, enquanto tarefas de complexidade muito menor levam muito mais tempo em um computador de alta configuração de *hardware* (HAYKIN, 2009).

Redes neurais artificiais são modelos computacionais que se baseiam no sistema nervoso dos seres viventes. Possuem a capacidade de adquirir assim como manter o conhecimento e podem ser definidas como um conjunto de unidades de processamento (neurônios artificiais) que são interligados por conexões, que na maioria das vezes é representada por matrizes de pesos sinápticos (SILVA; SPATTI; FLAUZINO, 2010).

2.2.1 Neurônio Biológico

O sistema nervoso central contém mais de 100 bilhões de neurônios. A Figura 1 mostra um neurônio típico encontrado no córtex motor cerebral. Os sinais entram nesse neurônio por sinapses localizadas principalmente nos dendritos neuronais, mas também no corpo celular. No cérebro humano pode haver algumas centenas ou até 200.000 conexões sinápticas. Por outro lado, o sinal de saída viaja através de um único axônio deixando o neurônio. Esse axônio pode ter muitos ramos separados para outras partes do sistema nervoso ou corpo periférico (GUYTON et al., 2014).

Figura 1 - Estrutura de um grande neurônio e suas partes funcionais



Fonte: Adaptado de Guyton et al. (2014).

2.2.2 Neurônio Artificial

Um neurônio artificial é uma unidade de processamento de informação que é fundamental para o funcionamento de uma rede neural. O diagrama de blocos da Figura 2 mostra o modelo de um neurônio artificial.

Na Figura 2, é possível notar um conjunto de sinapses ou ligações, cada um dos quais é caracterizado por um peso. Um sinal x_j na entrada da sinapse j conectada ao neurônio k é multiplicado pelo peso sináptico w_{kj} . O primeiro subscrito em w_{kj} refere-se ao neurônio em questão, e o segundo subscrito refere-se ao final de entrada da sinapse a que se refere o peso. Ao contrário de uma sinapse no cérebro, o peso sináptico de um neurônio artificial pode situar-se em um intervalo que inclui

valores negativos e positivos. Pode-se notar um somador, que soma por combinação linear os sinais de entrada, ponderado pelos respectivos pesos sinápticos do neurônio. Existe também uma função de ativação para limitar a amplitude da saída de um neurônio, logo, a amplitude permissível do sinal de saída é modificada para algum valor finito. Além disso, o neurônio artificial inclui também um *bias* aplicado externamente, denotado por b_k . O *bias* b_k , que conforme a Figura 2 também é considerado w_{k0} , tem o efeito de aumentar ou diminuir a entrada líquida da função de ativação, dependendo se ela é positiva ou negativa (HAYKIN, 2009).

Em termos matemáticos o neurônio k da Figura 2 pode ser representado por

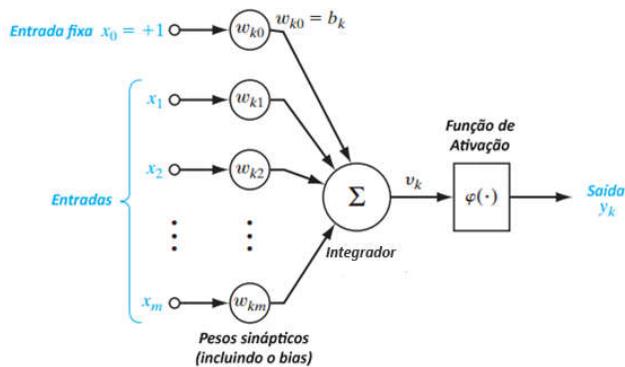
$$v_k = \sum_{j=0}^m w_{kj} \cdot x_j \quad (1)$$

e

$$y_k = \varphi(v_k) \quad (2)$$

onde x_j é o sinal da j -ésima entrada; m é o número de entradas ao neurônio; w_{kj} é o respectivo peso sináptico do neurônio k ; sendo $w_{k0} = b_k$ que é o bias; v_k é a saída do combinador linear devido aos sinais de entrada; $\varphi(\cdot)$ é a função de ativação; e y_k é o sinal de saída do neurônio (HAYKIN, 2009).

Figura 2 - Modelo não linear de um neurônio artificial



Fonte: Adaptado de Haykin (2009).

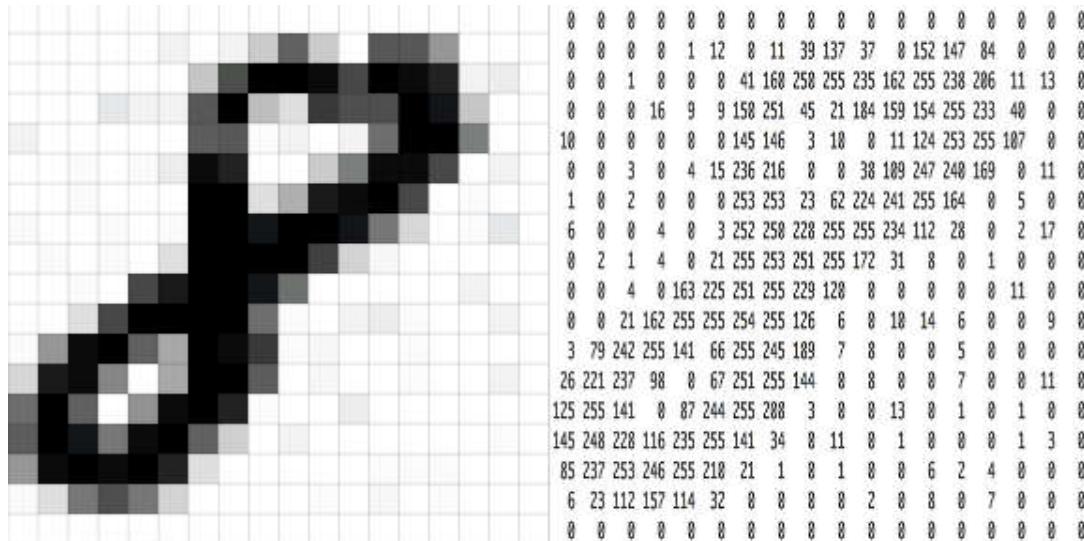
2.2.3 Reconhecimento de objetos com redes neurais artificiais

O reconhecimento de objetos é uma tarefa bastante simples para os seres humanos, mas a reprodução por computadores vem instigando cientistas da computação há mais de cinquenta anos. As redes neurais artificiais são comumente utilizadas no reconhecimento de objetos e têm resultados bastante satisfatórios como no caso de reconhecimento de escrita cursiva (LECUN et al., 1989).

Uma rede neural pode tomar números como entrada. Para um computador, uma imagem é apenas uma grade de números que representa o quanto escuro cada pixel é, como é possível observar na Figura 3.

Em uma rede neural, a imagem da Figura 3, é simplesmente tratada como uma matriz de 324 números. A partir de uma base de treinamento, como a da Figura 4, uma rede neural com apenas uma camada oculta é capaz de reconhecer o número “8” nas imagens apresentadas (LECUN et al., 1989).

Figura 3 – Representação numérica em escala de cinza



Fonte: Adaptado de Geitgey (2016).

Figura 4 – Exemplo de amostras de imagem utilizadas no treinamento da rede

```

8 2 7 7 5 7 7 2 8 8 5 7 0 7 1 7 5 9 3 1 0 2 7 9 9 6 9 4 7 4 1 1 4 4 8 8 0 2 6 3
0 0 7 6 3 4 4 4 3 4 2 3 2 8 0 8 2 9 7 6 7 9 0 0 4 2 0 6 6 4 3 3 9 0 4 7 3 2 2 0
2 6 4 6 4 7 5 9 8 7 1 9 0 6 8 7 7 1 9 8 6 5 2 1 0 1 0 8 3 4 7 7 1 3 0 9 6 0 3 8
0 2 8 3 6 5 7 6 0 7 2 6 1 0 2 6 9 7 1 9 5 8 7 0 0 6 1 6 4 4 8 6 2 3 3 1 3 9 4
$ 1 0 2 1 4 2 2 0 9 9 9 3 1 3 4 1 9 5 5 4 3 9 3 3 5 8 5 0 6 5 1 8 2 6 8 9 2 2 8
L 7 2 7 5 5 0 7 2 2 1 3 5 8 4 8 5 2 5 7 1 6 1 8 3 8 0 0 1 0 3 0 2 4 0 8 6 6 2
1 3 3 9 0 4 9 7 5 6 9 5 5 2 6 9 5 3 9 7 3 0 4 6 2 9 4 0 0 2 7 1 0 3 9 1 2 6 0 6
3 4 1 1 9 0 8 2 1 1 9 0 7 5 7 4 2 3 9 9 0 2 5 2 1 3 8 2 3 1 6 7 6 0 7 2 0 0 5
7 1 3 1 2 8 8 2 9 4 4 2 4 7 9 8 4 8 0 3 0 7 8 8 3 9 4 7 3 3 1 0 0 8 7 2 1 1 6 2
6 0 1 7 0 3 6 1 6 5 0 7 8 7 8 6 9 2 3 8 8 6 5 1 1 3 2 6 0 5 9 9 1 0 2 2 1 9

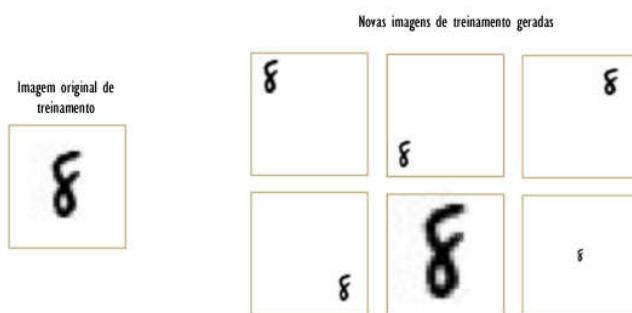
```

Fonte: Geitgey (2016).

Uma rede tomando números como entrada de dados tem um elevado grau de acertos, desde que as imagens apresentadas para a validação estejam perfeitamente centralizadas. Essa condição é complicada levando-se em consideração que o objetivo da criação da rede neural artificial seja uma aplicação para resolução de problemas reais, como o caso reconhecimento de códigos postais em cartas (LECN et al., 1989).

Para melhor adaptação às realidades que o problema suscita, faz-se necessário a incorporação de outras técnicas que permitam apresentar as imagens para validação de forma centralizada ou gerar, de forma aleatória, mais imagens na base de treinamento, conforme mostrado na Figura 5. Ao inserir grande quantidade de amostras, a complexidade do problema aumenta, necessitando de maior complexidade na rede e, consequentemente, sendo necessário a introdução de técnicas de *deep learning* (HAYKIN, 2009).

Figura 5 - Novas imagens de treinamento geradas a partir da imagem original



Fonte: Adaptado de Geitgey (2016).

2.3 DEEP LEARNING

O *deep learning* refere-se a um subcampo de aprendizagem de máquina que se baseia em níveis de aprendizagem, correspondendo a uma hierarquia de características, fatores ou conceitos, no qual conceitos superiores são definidos a partir de conceitos inferiores que ajudam a definir novos conceitos. *Deep learning* é aprender múltiplos níveis de representação e abstração, ajudando a entender os dados. O conceito de *deep learning* vem do estudo da rede neural artificial MLP que contém várias camadas escondidas (LIU et al., 2015).

Mesmo sendo possível a resolução do problema utilizando novas amostras de treinamento geradas aleatoriamente e melhorar a quantidade de acertos introduzindo mais camadas ocultas à rede, existe outra técnica que analisa a imagem mesmo que o objeto a ser reconhecido esteja fora do centro da imagem. Essa configuração de rede neural é chamada de rede neural convolucional (ZEILER; FERGUS, 2014).

2.4 REDES NEURAIS CONVOLUCIONAIS

2.4.1 Visão Geral

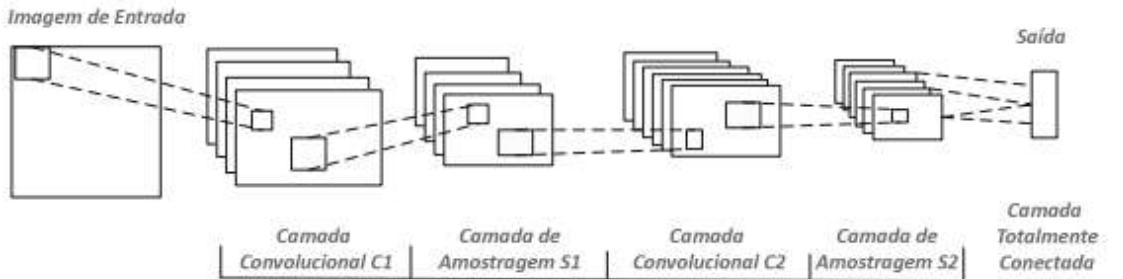
As redes neurais convolucionais têm melhorado significativamente o desempenho na classificação de imagens e outras tarefas de reconhecimento (ZEILER; FERGUS, 2014). Desde a sua introdução no início dos anos 90, as redes neurais convolucionais têm sido consistentemente competitivas com outras técnicas de classificação e reconhecimento de imagens (LECUN et al., 1989).

A CNN pode extrair a característica das imagens 2D diretamente, de modo que seja aplicada à classificação da imagem gradualmente. Lecun et al. (1989), formularam a primeira CNN concebida e treinada com base no gradiente de erro. Foi usada uma CNN para classificar os dígitos manuscritos que alcançou resultados melhores dos que haviam sido obtidos até aquele momento utilizando qualquer outra técnica anteriormente testada. Em (GARCIA; DELAKIS, 2004), foi realizada a detecção de rosto usando CNN com três camadas, incluindo uma camada convolucional, uma camada de amostragem e uma camada MLP. Em (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), foi treinada uma grande e profunda CNN que obteve sucesso sem precedentes no concurso *ImageNet*, concurso que avalia algoritmos para localização

e detecção de objetos a partir de imagens e vídeos (LAB, 2016). A rede implementada classificou 1,2 milhões de imagens naturais de alta resolução em 1000 classes diferentes pela profunda CNN, o que aumentou a confiança dos pesquisadores sobre a CNN.

A CNN é uma rede neural de várias camadas e é projetada especialmente para a classificação de imagens 2D. Cada camada de CNN é composta de múltiplos planos 2D e cada plano 2D consiste em muitos neurônios independentes. A Figura 6 mostra a arquitetura da CNN tradicional, que envolve três operações: convolução, amostragem e saída (ZHANG et al., 2016).

Figura 6 - Arquitetura tradicional de uma CNN



Fonte: Adaptado de Zhang et al. (2016).

Conforme ilustrado na Figura 6, a imagem de entrada está envolvida em quatro filtros treináveis para produzir quatro *features maps* (mapas de características) na camada convolucional C1. Cada bloco 2×2 nos mapas de características são adicionados, ponderados, combinados com um *bias* e passados através de uma função de ativação para produzir quatro *features maps* na camada de amostragem S1. Estes são filtrados novamente para produzir a camada convolucional C2. A hierarquia então produz S2 de uma maneira análoga a S1. Finalmente, a *fully connected* (camada totalmente conectada) combina todos os pixels de S2 em um vetor de saída 1D (ZHANG et al., 2016). O *feature map* da camada convolucional é definido como

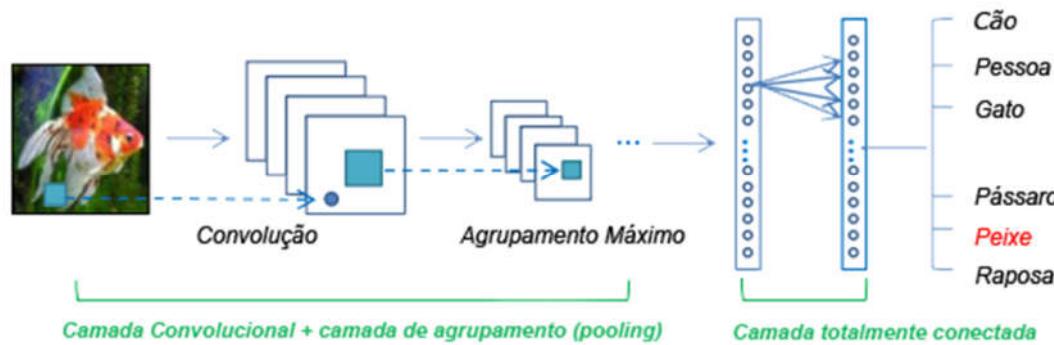
$$x_j^l = f(\sum_{i \in M} x_i^{l-1} * k_{ij}^l + b_j^l), \quad (3)$$

onde x_j^l é o j -ésimo mapa de característica na l -ésima camada, f é a função de ativação, M é o número de mapas de característica na camada $l-1$, x_i^{l-1} é o mapa de

característica i -ésimo na camada $l-1$, "*" significa a operação de convolução, k_{ij}^l é o *kernel* (núcleo) entre o i -ésimo *feature map* na camada $l-1$ e o j -ésimo *feature map* na l -ésima camada, e b_j^l é o *bias* para o j -ésimo *feature map* na l -ésima camada.

Outra maneira de expressar mais facilmente uma Rede Neural Convolucional é conforme a Figura 7.

Figura 7 - Outra demonstração gráfica de uma Rede neural Convolucional



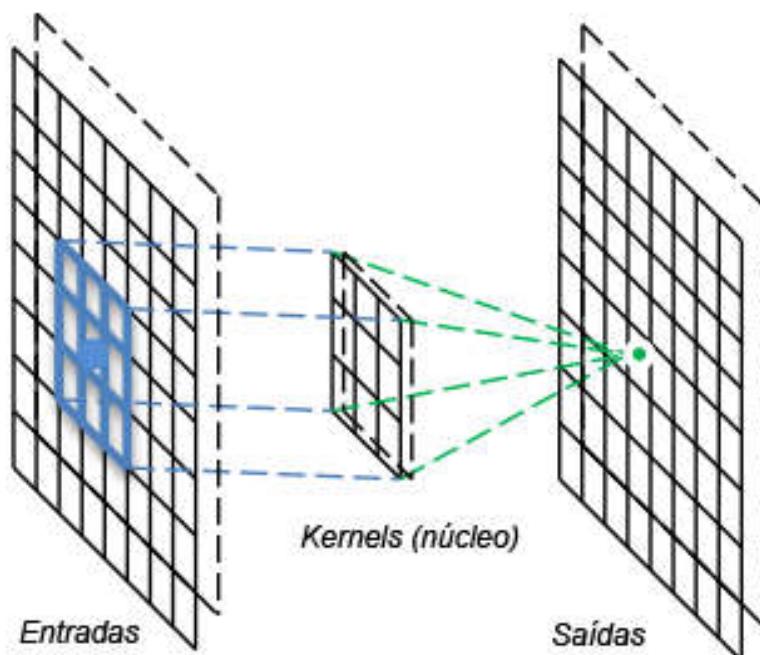
Fonte: Adaptado de GUO et al. (2016).

A Figura 7 mostra uma rede composta de três camadas neurais principais, que são a camada convolucional, *max pooling* (camada de agrupamento) e a *fully connected*. Existem duas etapas para treinar a rede: uma *forward* (para frente), no sentido mostrado na Figura 7, e um estágio *backward* (para trás). Após as etapas de convolução e de agrupamento as principais características da imagem serão extraídas gerando uma previsão de saída, que será comparada com o que deveria ser a real saída. Essa previsão de saída é utilizada para calcular o erro da previsão, que seria o *loss cost* (custo da perda). Agora, já com base no *loss cost*, o estágio de *backward* calcula os gradientes de cada parâmetro com regras de cadeia. Todos os parâmetros são atualizados com base nos gradientes e estão preparados para novamente iniciar o *forward*. Após suficientes iterações de *forward* e *backward*, a aprendizagem pode ser interrompida, geralmente quando se atinge um custo de perda pré-determinado (GUO et al., 2016).

2.4.2 Camada convolucional

A Figura 8 é uma representação da operação em uma camada convolucional. A partir de vários *kernels*, a imagem como um todo passa por convolução assim como os *features maps*, gerando outros inúmeros *features maps*. As vantagens principais da operação de convolução são: o mecanismo de compartilhamento de peso no mesmo *feature map*, que reduz o número de parâmetros a serem processados pela rede; a conectividade local “aprende” correlações entre pixels vizinhos melhorando a invariância à localização do objeto. Este último é um recurso importante no reconhecimento de padrões, já que um objeto é o mesmo na imagem não importa onde nela esteja (ZEILER; FERGUS, 2014).

Figura 8 - Operação da camada convolucional



Fonte: Adaptado de GUO et al. (2016).

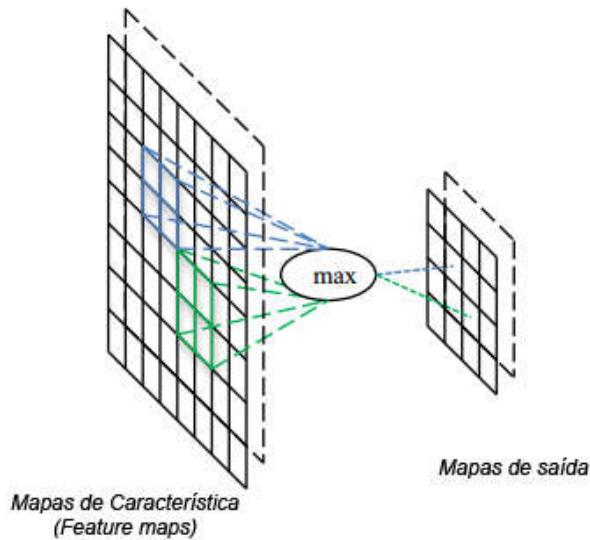
2.4.3 Pooling layer

As *pooling layers*, por levarem em conta nos cálculos os pixels vizinhos, são consideradas invariantes de tradução. São usadas para diminuir o tamanho dos *feature maps* e o número de parâmetros, sendo o *average pooling* (agrupamento médio) e o *max pooling* (agrupamento máximo) os mais comumente utilizados (GUO et al., 2016). Na Figura 9 está uma representação de uma camada de agrupamento usando o conceito de *max pooling*.

2.4.4 Fully connected

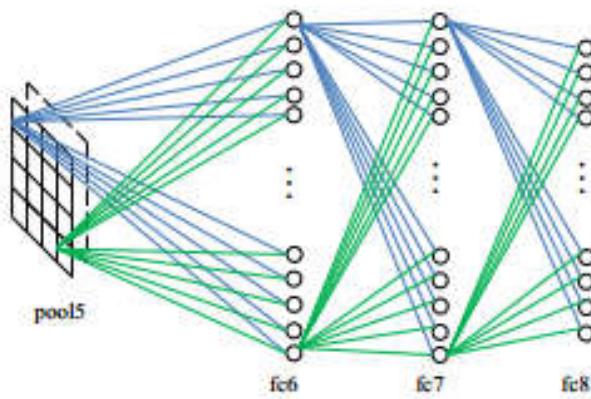
A camada *fully connected* geralmente vem após a última camada de *pooling* e converte, por meio de várias conexões, um *feature map* 2D para um *feature map* 1D. Essas camadas funcionam como uma rede neural tradicional e possuem a maioria dos parâmetros totais da rede, cerca de 90% deles (GUO et al., 2016). A Figura 10 exemplifica uma rede com três *fully connected*.

Figura 9 - Operação de uma camada *max pooling*



Fonte: Adaptado de GUO et al. (2016).

Figura 10 - A operação de uma camada totalmente conectada



Fonte: GUO et al., 2016.

2.5 TENSORFLOW

O *TensorFlow* (GOOGLE, 2016) foi desenvolvido por pesquisadores que trabalham no Google *Brain Team* dentro da organização de pesquisa da *Machine Intelligence* da Google para fins de pesquisa em máquinas e pesquisa em *deep learning*.

Ele é uma biblioteca de código aberto para computação numérica usando gráficos de fluxo de dados. Os nós no gráfico representam operações matemáticas, enquanto as bordas do gráfico representam matrizes de dados multidimensionais (tensores) comunicados entre eles. A arquitetura flexível permite implantar computação para uma ou mais CPUs ou GPUs em *desktops*, servidores ou dispositivos móveis (TensorFlow, 2016).

A partir do *TensorFlow*, foi criado pelo MIT a biblioteca de aprendizagem profunda TFLEARN, a fim de acelerar e facilitar os processos de forma transparente e compatível com o *TensorFlow*, já que o utiliza como plano de fundo. Além disso, a visualização gráfica fica fácil e intuitiva, com detalhes sobre pesos, gradientes e ativações. Assim, a API é fácil de usar e entender podendo ser utilizada para implementar redes profundas de alto nível (MIT, 2016).

2.6 APLICATIVOS MÓVEIS

Desde o advento do iPhone no início de 2007, pôde-se experimentar a funcionalidade de computadores pessoais em dispositivos de bolso. Os chamados *smartphones* e seus associados, os aplicativos, estão se tornando cada vez mais presentes no cotidiano das pessoas.

A escolha desse tipo de plataforma para a criação de um aplicativo de cadastro vem da premissa que a maioria das pessoas no mundo atual possuem aparelho celular e fazem uso constante de suas tecnologias. Ademais, recursos como a câmera e o acesso à internet são essenciais para o escopo deste trabalho, já que o objetivo é deslocar a ação de coletar imagens do banco de faces, para o usuário do sistema, reduzindo custos e tempo. Dessa forma, foi escolhida a plataforma Android por ser a que possui o maior número de usuários aumentando mais o universo de possíveis usuários (GOOGLE, 2012). Assim, foi desenvolvido o aplicativo FotoFace, com o intuito de cadastrar os usuários de forma que os mesmos enviem suas fotos e vídeos.

3 REDE NEURAL ARTIFICIAL PARA RECONHECIMENTO FACIAL

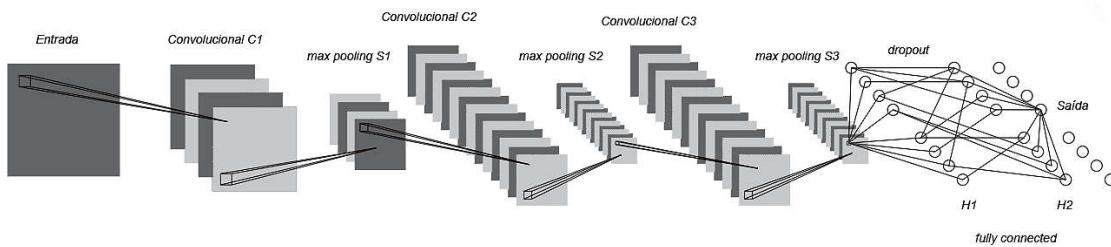
Duas importantes etapas para este trabalho são as etapas de implementação e treinamento da rede neural para o reconhecimento facial. Em um primeiro momento, serão descritas todas as etapas de implementação da rede. Após, será explicitado o processo de treinamento da rede que foi utilizada para a validação, rede essa que utilizou as imagens de *datasets* disponíveis na literatura. E, por último, serão apresentados os resultados obtidos nos referidos treinamentos.

3.1 IMPLEMENTAÇÃO DA REDE

Aparentemente, a CNN tem uma vantagem no reconhecimento de imagem. No entanto, não há nenhuma teoria comprovada sobre a construção de redes CNN, como o número de camadas ou o número de *features maps* de cada camada. Os pesquisadores constroem vários candidatos da CNN baseados na experimentação (tentativas) e determinam no final com base na comparação do melhor desempenho (ZHANG et al., 2016).

Dessa forma, foi proposta por meio de experimentação, a seguinte configuração de CNN, conforme mostrado na Figura 11, a fim de alcançar de forma satisfatória os objetivos deste trabalho.

Figura 11 - Esquema mostrando a rede implementada



Fonte: Autor (2017).

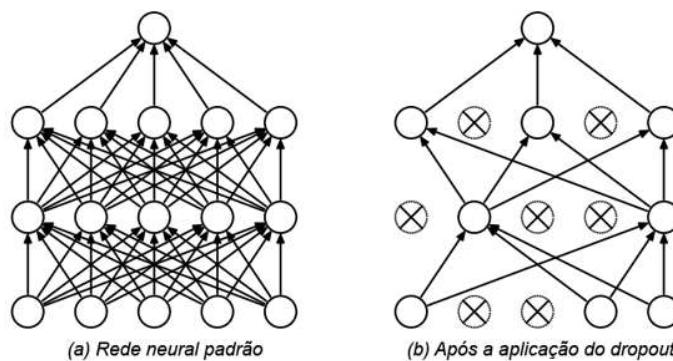
Para a implementação de uma algoritmo CNN, é necessário certa familiaridade com a arquitetura da rede e o modelo deve ser testado incessantemente na aplicação prática em que deseja empregá-lo, a fim de obter a mais adequada arquitetura para a aplicação específica (LIU et al., 2015).

A escolha em utilizar apenas escala de cinza foi tomada baseada no objetivo de minimizar o tempo de treinamento e tornar o aprendizado diário da rede possível. Testes preliminares mostraram o acréscimo de até 487% no tempo de processamento utilizando imagens coloridas e em alguns casos, não houve decréscimo da taxa de erros na validação.

Como a rede possui dados de treinamento limitados, é necessário utilizar a técnica de *dropout* a fim de evitar o *overfitting*, que é resultado de interferências das relações complicadas entre as entradas e as saídas de uma rede em *deep learning*. As interferências causadoras do *overfitting*, citadas anteriormente, vão existir no conjunto de treinamento, mas não no real conjunto de dados de teste, mesmo que os dados de teste sejam extraídos da mesma distribuição (SRIVASTAVA et al., 2014).

O termo *dropout*, dito anteriormente, refere-se a abandonar aleatoriamente um percentual de unidades que em uma rede neural são os neurônios. Remove-se temporariamente alguns neurônios da rede juntamente com todas as conexões de entrada e saída, como exemplificado na Figura 12.

Figura 12 - Modelo de uma rede utilizando o conceito de *dropout*



Fonte: Adaptado de SRIVASTAVA et al. (2014).

Foi utilizada como função de ativação, uma adaptação da função *ReLU*, denominada *Parametric Rectified Linear Unit - PReLU* (unidade linear paramétrica retificada) que em estudos recentes apresentou uma melhora de quase um ponto percentual na taxa de acertos (HE et al., 2015).

Para problemas de classificação usando técnicas de *deep learning*, é padrão usar o *SoftMax*. O *SoftMax* retorna um valor de acordo com uma distribuição de probabilidade, em que a previsão de saída é dada pela probabilidade distribuída entre todas as classes inseridas no treinamento. Esse tipo de classificação é o requerido neste trabalho (TANG, 2013). Além do *SoftMax*, foi utilizado também um método de otimização estocástica baseado em estimativas adaptativas de momentos de ordem inferior, o *ADAM*. O método calcula taxas de aprendizagem adaptativas individuais para diferentes parâmetros a partir de estimativas de primeiro e segundo momentos dos gradientes. Isso faz com que o algoritmo solicite pouca memória, diminuindo o custo computacional (KINGMA; BA, 2014).

A primeira arquitetura proposta possuía apenas duas camadas convolucionais, duas camadas de *pooling* e duas camadas *fully connected*. As funções de ativação eram a *ReLU*, com exceção da segunda camada *fully connected* que possuía função de ativação *SoftMax*. Nos testes realizados com essa arquitetura, ocorriam muitas variações nas perdas e as taxas de treinamento variavam entre 70 e 80% de precisão. Com a mudança das funções de ativação de *ReLU* para *PreLU* os valores passaram de 80 a 90% de precisão no treinamento. Embora, houvesse melhora nos valores de precisão, ainda haviam inconsistências, ou seja, muitas variações nas perdas.

Dessa forma, mais uma camada convolucional e mais uma camada de *pooling* foram adicionadas elevando as precisões para um patamar acima de 95% e com a utilização da técnica de *dropout*, descrita anteriormente, as variações nas perdas diminuíram. Assim, foram realizados testes de aumento e de decréscimo do número de *kernels*, ocorrendo pequenas variações de melhora e de piora nos índices. Dessa maneira, foi possível chegar aos números propostos, conforme Tabela 1, que obtiveram o melhor resultado para a aplicação em questão.

A rede proposta é composta de uma camada de entrada, a primeira camada convolucional (C1), uma camada de *max pooling* (S1), a segunda camada convolucional (C2), uma camada de *max pooling* (S2), uma terceira camada convolucional (C3), mais uma camada de *max pooling* (S3), uma camada *fully connected* (H1), uma segunda camada *fully connected* (H2) e a camada de saída.

Os parâmetros escolhidos para a arquitetura estão resumidos e dispostos na Tabela 1 abaixo.

Tabela 1 – Parâmetros de arquitetura da rede implementada neste trabalho

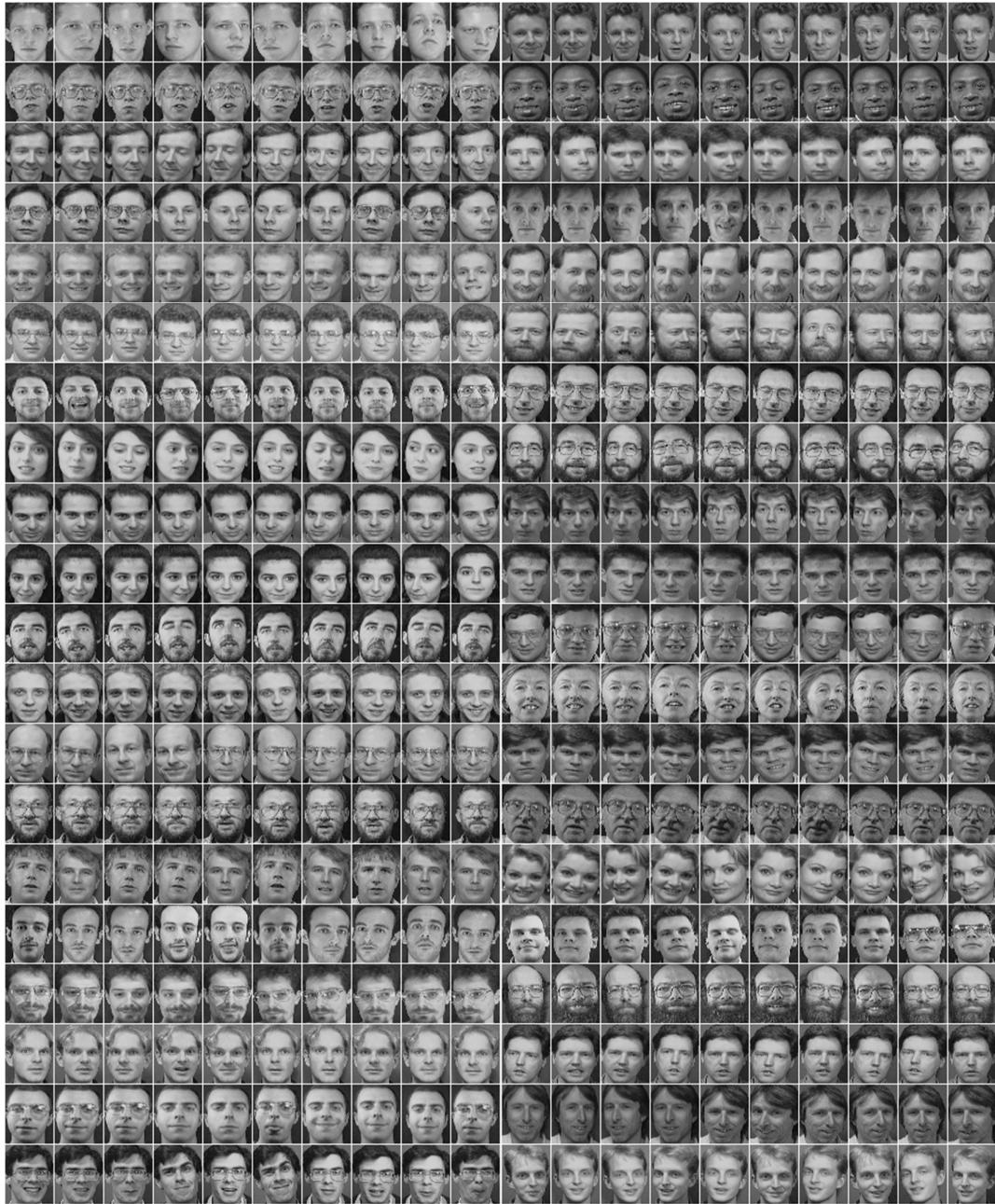
Camada	Kernels	Tamanho dos Kernels (pixels)	Função de ativação	Quantidade de neurônios
Convolução C1	20	20x20	PReLU	-
Convolução C2	25	10x10	PReLU	-
Convolução C3	30	5x5	PReLU	-
<i>Pooling S1</i>	20	10x10	-	-
<i>Pooling S2</i>	25	5x5	-	-
<i>Pooling S3</i>	30	5x5	-	-
<i>Fully Connected H1</i>	-	-	PReLU	10 x número de classes
<i>Fully Connected H2</i>	-	-	SoftMax	Número de classes

Fonte: Autor (2017).

3.2 ESCOLHA DE BANCOS DE IMAGENS FACIAIS DISPONÍVEIS NA LITERATURA PARA A VALIDAÇÃO DA REDE

Foram selecionados dois *datasets* de imagens faciais a fim validar os resultados da rede. O primeiro foi o *ORL database* (CAMBRIDGE, 2002), que contém um acervo de fotos adquiridas entre abril de 1992 e abril de 1994, no Laboratório de Pesquisas Olivetti em Cambridge. Há um conjunto de 10 diferentes imagens de 40 indivíduos distintos. Alguns indivíduos cederam as imagens em datas variadas demonstrando leve variação na aparência. Há, no *dataset*, variações quanto a sorrisos, olhos abertos e fechados e utilização ou não de óculos. Todas as imagens foram adquiridas contra um fundo escuro e homogêneo, estando todas em posição frontal e com variações de rotação máxima de 20 graus. Em algumas imagens, também são notadas variações em torno de 10% na escala (CAMBRIDGE, 2002). As imagens são em escala de cinza com resolução de 92x112 e são mostradas na Figura 13.

Figura 13 - *Dataset ORL de imagens faciais. São 40 indivíduos e 10 imagens cada*



Fonte: CAMBRIDGE (2002).

Há grande dificuldade de se encontrar trabalhos relacionados a redes neurais convolucionais que utilizem bancos de faces com poucas amostras por classe. A maioria dos trabalhos mais recentes buscam resolver problemas de classificação de imagens com bancos de milhares de imagens. Mas, para o escopo deste trabalho é necessário que os testes sejam com um número reduzido de imagens a fim de

encontrar semelhança com os parâmetros do sistema em questão, já que espera-se um pequeno número de imagens para treinamento.

O segundo banco de imagens faciais utilizado foi o *FEI Face Database* que é um banco de dados brasileiro que contém imagens adquiridas entre junho de 2005 e março de 2006 pelo Laboratório de Inteligência Artificial do Centro Universitário FEI de São Bernardo do Campo, São Paulo (THOMAZ, 2012). O banco possui imagens de 200 indivíduos, 14 imagens cada, em um total de 2800 imagens. Todas as imagens são coloridas e foram adquiridas contra um fundo branco e homogêneo. Há imagens rotacionadas em até 180 graus e com cerca de 10% de alteração de escala e uma das amostras de cada indivíduo possui baixa iluminação. Os voluntários da pesquisa estão divididos em 100 homens e 100 mulheres, entre 19 e 40 anos de idade com aparência, adornos e estilo de penteado distintos (FEI Face Database, 2012). A Figura 14 mostra o exemplo de variação nas posições das imagens.

Figura 14 - Variações nas posições das imagens do *FEI Face Database* de um voluntário



Fonte: THOMAZ (2012).

Após uma busca por publicações relacionadas ao escopo desse trabalho, constatou-se que não há qualquer publicação sobre reconhecimento facial por meio de redes neurais que utilizaram esse banco de imagens. Então, a escolha do referido banco baseia-se na semelhança das características das imagens do *FEI Face Database* com as características das imagens capturadas pelo aplicativo proposto neste trabalho, como: variações nas posições faciais, alteração de feições dos participantes, poucas imagens por indivíduo e interferências de iluminação, como também há ainda a possibilidade de avaliar o desempenho da rede na classificação de um número elevado de classes, no caso para 100 indivíduos.

3.3 TREINAMENTO DA REDE E OS RESULTADOS OBTIDOS

3.3.1 Rede treinada com as imagens do *ORL Database*

Para validar a arquitetura de rede proposta, foram realizados dois treinamentos utilizando o banco de imagens faciais *ORL Database*. Em todos os treinamentos a arquitetura descrita na seção 3.1 foi integralmente mantida.

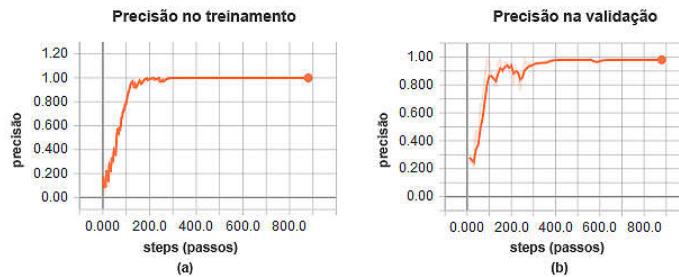
Primeiro as imagens foram reduzidas para 46x56 pixels e foram aleatoriamente sortidas para que as amostras da mesma classe não se encontrassem em sequência na ordem de processamento. Para os treinamentos foi utilizada uma taxa de aprendizagem de 0,001. Foi realizado um treinamento para classificar 10 classes, ou seja, 10 indivíduos distintos e utilizou-se 50% das amostras para treinamento e 50% das amostras para validação.

Todos os treinamentos foram executados em um notebook *Lenovo* arquitetura 64 bits, processador INTEL® Core™ i7, de 2,4 GHz e 8 GB de memória RAM e placa de vídeo dedicada AMD Radeon R5 M230 de 2 GB.

Cada etapa de treinamento processa uma quantidade preestabelecida de imagens por vez. Esta quantidade é denominada de *batch size* (tamanho do lote). O *batch size* determina quantos passos cada época de treinamento terá que realizar para processar todas as imagens. Após cada passo de treinamento, são gravadas as precisões de treinamento e de validação, assim como as perdas de treinamento e de validação. Dessa forma, é possível visualizar o andamento do treinamento após cada passo realizado.

Foram necessárias 88 épocas de treinamento para treinar a rede com 10 classes. No treinamento, cada passo processou 5 imagens, totalizando nas 88 épocas, 880 passos de treinamento em um tempo de 180,96 segundos. A precisão no treinamento foi de 100% e a precisão de validação de 98%, conforme mostrado na Figura 15.a e na Figura 15.b, em que o valor 1.00 na imagem corresponde a 100%. Nota-se que o comportamento após o passo 250 foi de estabilidade, ocorrendo apenas uma breve oscilação próximo ao passo 600.

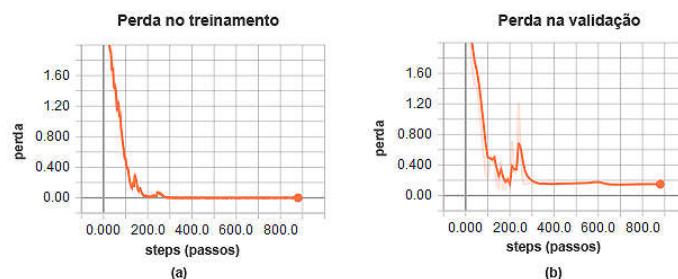
Figura 15- Precisão de treinamento e de validação para reconhecimento de 10 classes



Fonte: Autor (2017).

Ainda sobre a análise do treinamento de 10 classes, pode-se notar na Figura 16.a e na Figura 16.b, a perda no decorrer do treinamento e da validação da rede. Conforme já dito anteriormente, o valor de perda é o erro da previsão, uma mensuração da diferença da saída alcançada, para a saída real. Para problemas de classificação de mais de duas classes, é comumente utilizada para o cálculo das perdas, a *categorical cross entropy* (entropia cruzada categórica) (RUBINSTEIN et al., 2004). A perda no treinamento foi de 0,0001 e na validação foi de 0,151. É possível notar também que a perda se mantém com pouca ou nenhuma variação no momento em que a precisão se encontra estável, caracterizando que não houve *overfitting* na rede (HU et al., 2015).

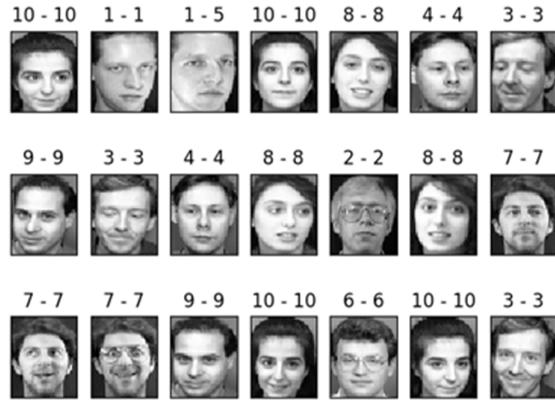
Figura 16 - Perda de treinamento e de validação para o reconhecimento de 10 classes



Fonte: Autor (2017).

Algumas amostras das faces validadas são mostradas na Figura 17, em que o primeiro número sobre a face corresponde à classe real e o segundo à classe reconhecida.

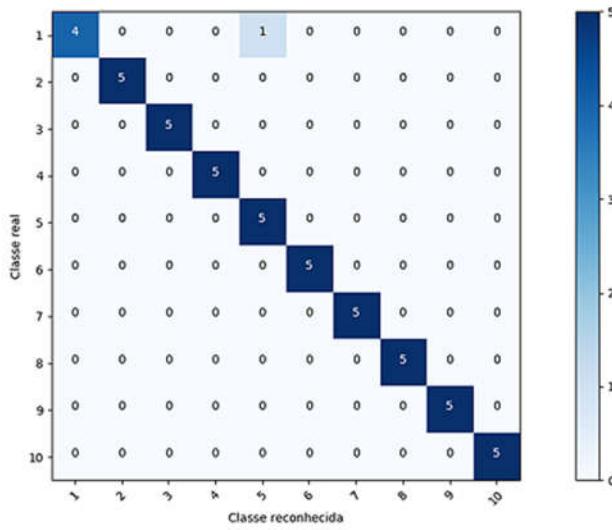
Figura 17 - Algumas amostras utilizadas na validação



Fonte: Autor (2017).

A matriz confusão para a validação das amostras de teste demonstrou um resultado condizente com os resultados apresentados no treinamento. Conforme pode ser visualizado na matriz da Figura 18, apenas uma classe não obteve a totalidade nos acertos.

Figura 18 - Matriz confusão para as 10 classes



Fonte: Autor (2017).

Em testes preliminares utilizando 20 classes com 5 amostras de treinamento e 5 amostras de validação para cada classe, houve muita oscilação nas perdas de

treinamento e de validação com altas ocorrências de *overfitting* que impossibilitaram a convergência, por se tratar de várias classes e poucas amostras de treinamento. Dessa forma, para melhorar a apresentação dos resultados com 20 classes, foi utilizado uma quantidade mais condizente com as usualmente utilizadas para treinamentos, 70% das imagens para treinamento e 30% para a validação. Dessa forma, para o *dataset* em questão foram 7 imagens para o treinamento e 3 para a validação.

Foram necessárias 155 épocas de treinamento para treinar a rede com 20 classes. No treinamento, cada passo processou 5 imagens, totalizando nas 155 épocas, 4340 passos de treinamento em um tempo de 565,04 segundos. A precisão no treinamento foi de 100% e a precisão de validação de 96,67%, conforme mostrado na Figura 19.a e 19.b.

A perda no treinamento e na validação estão expressas na Figura 20.a e na Figura 20.b, sendo que a perda no treinamento foi de 0,00001 e na validação em 0,36. Pode-se notar instabilidade no início do treinamento e algum tempo depois da estabilização, sendo que após os 3000 passos houve um decréscimo na perda da validação mas a precisão na validação de manteve constante, caracterizando que houve *overfitting* na rede nesse momento. Embora, pela pouca diferença entre a precisão de treinamento e de validação, pode-se concluir que houve pouco *overfitting*, corroborados pelos resultados de (HU et al., 2015).

Em (LAWRENCE et al., 1997) foi proposta uma solução de sistema de reconhecimento neural híbrido que combinou amostragem de imagem, mapas auto organizáveis e uma rede neural convolucional. Os valores de validação apresentados, no referido trabalho híbrido, utilizando o mesmo *dataset* com 10 classes foi de 98,67% de acerto e com 20 classes foi de 95,77%, ambos utilizando 5 amostras para o treinamento e 5 para a validação. Tais resultados estão sintetizados na Tabela 2. É possível notar que para 10 classes, o valor da validação na rede proposta neste trabalho foi bem próxima ao do sistema híbrido, uma diferença de apenas 0,67%. Apesar do índice de precisão na validação alcançado nesse trabalho para 20 classes ter sido maior que no sistema híbrido, ele foi alcançado utilizando 7 amostras de cada classe no treinamento, conforme foi exposto anteriormente.

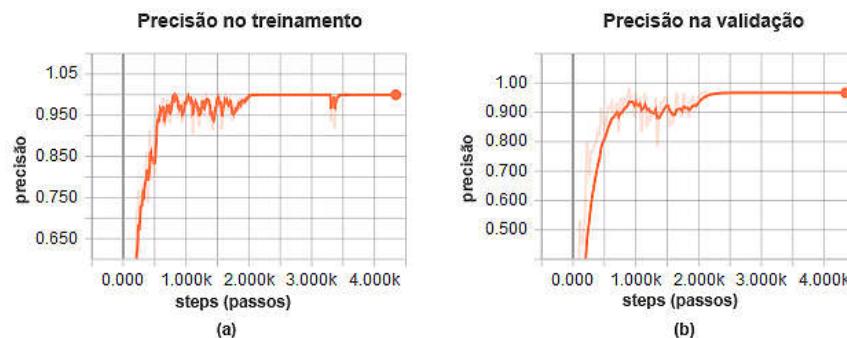
Tabela 2 – Resultados dos treinamentos com as imagens do *ORL Database*

Arquitetura	Número de classes	Precisão no treinamento	Precisão na validação
Proposta neste trabalho	10	100%	98,00%
	20	100%	96,67%
Híbrida (LAWRENCE et al., 1997)	10	-	98,67%
	20	-	95,77%

Fonte: Autor (2017).

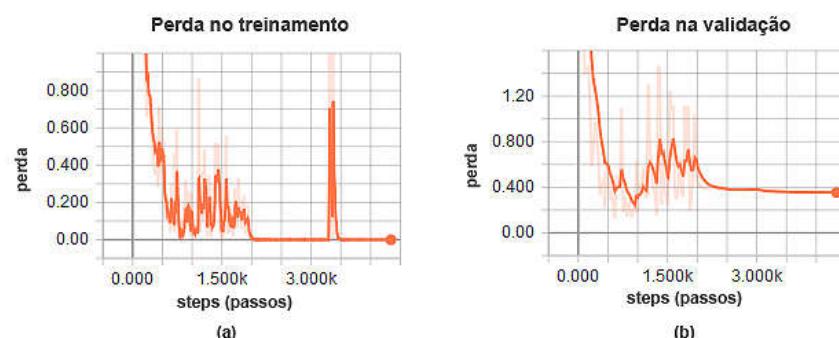
Mesmo com algumas discrepâncias, é possível atestar a validade do resultado, pois o desempenho da rede proposta neste trabalho para poucas classes obteve desempenho semelhante ao de um sistema híbrido de reconhecimento facial.

Figura 19 - Precisão de treinamento e de validação para 20 classes



Fonte: Autor (2017).

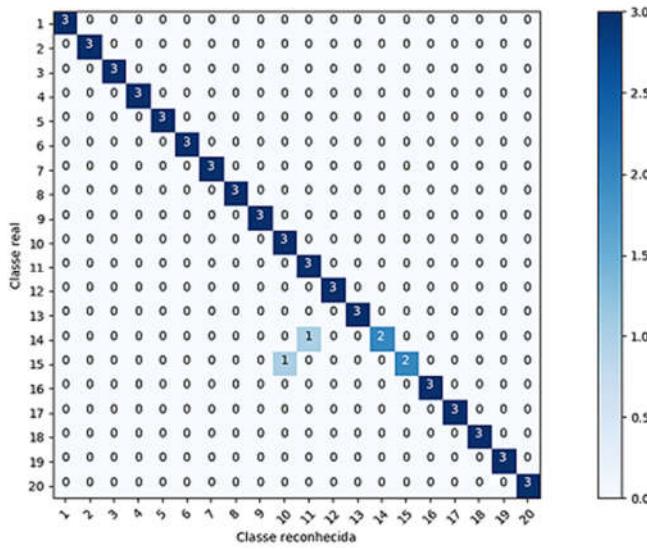
Figura 20 - Perda de treinamento e de validação para o reconhecimento de 20 classes



Fonte: Autor (2017).

Conforme pode ser visualizado na matriz confusão da Figura 21, o resultado foi condizente com a precisão na validação.

Figura 21 - Matriz confusão para a validação das 20 classes



Fonte: Autor (2017).

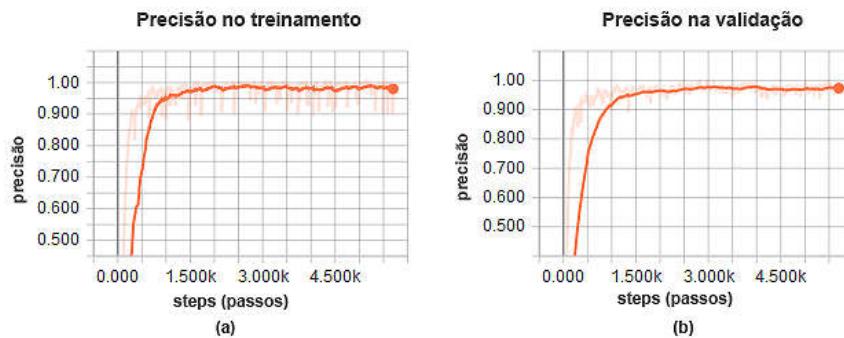
3.3.2 Rede treinada com as imagens do *FEI Database*

Como ocorrido no treinamento anterior, primeiro as imagens foram reduzidas para 80x60 pixels e foram aleatoriamente sortidas para que as amostras da mesma classe não se encontrassem em sequência na ordem de processamento. Para os treinamentos foi utilizada uma taxa de aprendizagem de 0,0005. Foi realizado 1 treinamento de 100 classes com 12 imagens de cada classe para o treinamento e 2 imagens para a validação. Vale ressaltar, que a arquitetura proposta no capítulo 3 foi integralmente mantida.

Foram necessárias 182 épocas de treinamento. O treinamento foi executado no mesmo *hardware* utilizado nos treinamentos anteriores.

No treinamento, cada passo processou 32 imagens, totalizando nas 182 épocas, 7962 passos de treinamento em um tempo de 6400 segundos. A precisão no treinamento foi de 98% e a precisão de validação de 97,3%, conforme mostrado na Figura 22.a e na Figura 22.b.

Figura 22 - Precisão de treinamento e validação no reconhecimento de 100 classes

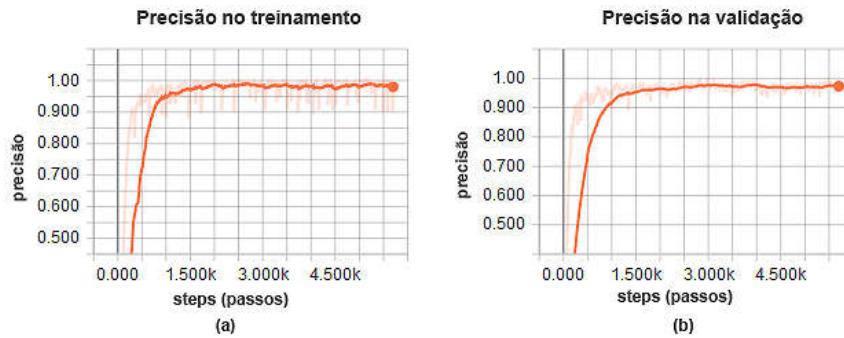


Fonte: Autor (2017).

Devido à variação no decorrer do treinamento, foi utilizado um recurso de suavização na exibição das informações do gráfico, a fim de melhorar a visualização da curva. Nota-se que, mesmo com o alto número de classes e a presença de oscilação, o modelo manteve-se estável em um intervalo entre 92 e 98% de precisão, tanto no treinamento como na validação.

A perda no treinamento foi de 0,131 e na validação foi de 0,152, conforme mostrado na Figura 23.a e na Figura 23.b. Dessa forma, com os resultados apresentados, nota-se que a arquitetura de rede proposta no trabalho tem resultados mais estáveis para a classificação de poucas classes. Mesmo assim, a instabilidade no treinamento não impediu a rede de executar bem a classificação das faces do banco de imagens em questão.

Figura 23 - Perda de treinamento e de validação no reconhecimento de 100 classes



Fonte: Autor (2017).

4 REDE PARA RECONHECIMENTO FACIAL

A configuração da rede validada, conforme descrito no capítulo anterior, foi utilizada para criar o sistema chamado de FotoFace. O sistema FotoFace é composto por um aplicativo de cadastro e captura de imagens dos usuários do sistema e uma interface para reconhecimento facial, que utiliza a rede neural convolucional descritas anteriormente. Neste capítulo serão abordadas as etapas de criação do aplicativo FotoFace. Serão mostradas também, as etapas de treinamento da rede a partir das imagens cadastradas e dos resultados obtidos com o treinamento.

4.1 APlicATIVO

Um dos objetivos do trabalho é a criação de um aplicativo na plataforma Android, para o cadastro, captura e envio de fotos e vídeos, que será chamado de Aplicativo FotoFace. Nesta seção, serão detalhadas as etapas da sua implementação.

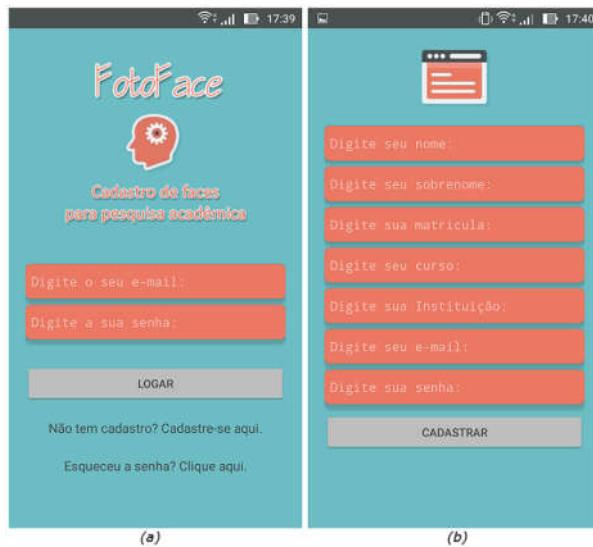
4.1.1 Cadastro do Usuário

Visando a melhor comodidade dos participantes do cadastramento, a total autonomia na coleta das informações e simular o uso de um sistema de cadastro real, foi desenvolvido para o cadastro a seguinte configuração de telas, conforme mostrado na Figura 24.

Na Figura 24.a, encontra-se a tela inicial do aplicativo FotoFace. Nela, o usuário que não tem cadastro é encaminhado para a realização do mesmo, clicando no texto “Não tem cadastro? Cadastre-se aqui”, logo abaixo do botão “LOGAR. O usuário é então encaminhado para a tela mostrada na Figura 24.b, para o preenchimento de seus dados. Ao clicar no botão “CADASTRAR” um *login* e senha serão armazenados no sistema *Authentication* (autenticação) da plataforma on-line *Firebase* da Google, plataforma que apresenta funcionalidades voltadas ao armazenamento de dados, autenticação de usuários e ferramentas de gerenciamento de aplicativos e sites (GOOGLE, 2017). Salienta-se que estratégias de validação dos campos foram implementadas a fim de evitar cadastros incorretos e incompletos.

A tela inicial do aplicativo, além de conter os campos de *login*, possui também o recurso de recuperação de senhas, caso o usuário se esqueça da senha que cadastrou ele receberá um e-mail da plataforma convidando-o a modificar sua senha.

Figura 24 - *Layout* das telas referentes ao cadastro



Fonte: Autor (2017).

Todo o aplicativo foi desenvolvido na plataforma Android Studio versão 2.1 em linguagem nativa do Android (GOOGLE, 2012), ou seja, linguagem Java (ORACLE, 2017).

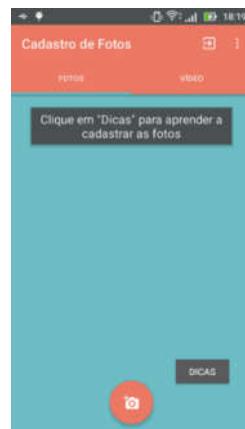
4.1.2 Captura e armazenamento de fotos

Ao efetuar o *login* com sucesso, o usuário é encaminhado para a tela principal do aplicativo em que há duas abas, uma para a captura de fotos e outra para a captura de vídeos. A aba de captura de fotos, caso o usuário não possua ainda fotos cadastradas, exibe a opção de dicas para a captura das fotos, conforme é mostrado na Figura 25.

Ao clicar no botão “DICAS”, uma animação contendo as instruções de utilização das ferramentas de edição é exibida. Nela, o usuário do aplicativo tem acesso ao passo a passo da operação a ser realizada. Após o pequeno tutorial, a tela de edição de fotos é iniciada e é possível adquirir uma foto da galeria de fotos do celular ou utilizar a câmera frontal para a captura, conforme Figura 26.a.

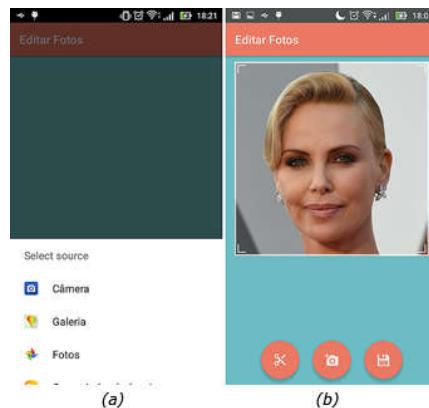
Dessa forma, é possível editar as fotos, visando a centralização da face por exemplo, e permitindo a utilização de fotos antigas armazenadas na galeria do dispositivo. Ao clicar no botão de gravação, mostrado na Figura 26.b, a fotografia é então salva em 240x240 pixels e enviada ao servidor *Firebase* vinculada ao usuário que a enviou.

Figura 25 - Tela principal do aplicativo



Fonte: Autor (2017).

Figura 26 - Tela de edição de fotos. À esquerda as opções de captura e a direita detalhe da janela de edição



Fonte: Autor (2017).

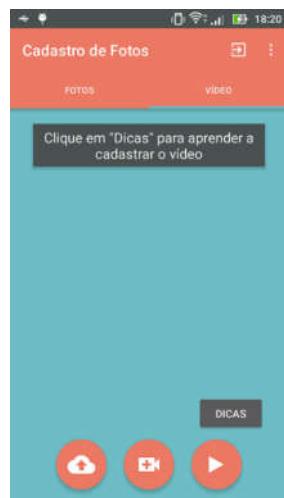
4.1.3 Captura e armazenamento de vídeos

Como estratégia para tornar o processo menos custoso para o usuário, foi desenvolvido um sistema de captura de vídeo, no qual o usuário rotaciona a câmera

do celular ao redor de seu rosto. Dessa forma, ao utilizar os *frames* (quadros) do vídeo gravado, é possível obter uma grande quantidade de imagens, além da possibilidade de capturar diferentes ângulos da face do indivíduo. A aba para a captura de vídeos é mostrada na Figura 27. Nota-se que também há a opção de dicas, em que um vídeo tutorial é exibido a fim de padronizar a captura e proporcionar a estratégia correta para a execução.

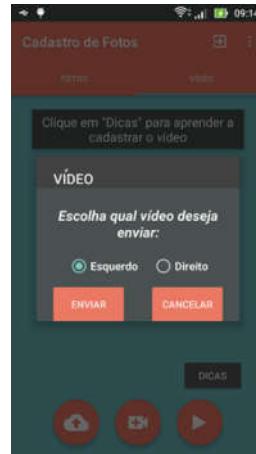
São necessários dois vídeos, um com uma panorâmica de 90 graus do lado direito e outro do lado esquerdo, cobrindo então toda a face. Após a exibição das dicas, ao clicar no botão central da tela, conforme mostrado na Figura 27, é possível capturar o vídeo da face do usuário. Se o vídeo foi corretamente salvo, uma mensagem será exibida para atestar o sucesso da operação e então, com o auxílio do botão *play* à direita, mostrado na Figura 27, pode-se visualizar a qualidade da captura antes de enviar. Um clique no botão enviar, o símbolo da esquerda exibido na Figura 27, abre a janela com a opção de qual vídeo vai ser enviado, esquerdo ou direito, conforme mostrado na Figura 28. Não há como identificar se o usuário realmente gravou o lado esquerdo ou direito do rosto.

Figura 27 - Aba para a captura e envio dos vídeos



Fonte: Autor (2017).

Figura 28 - Janela para envio de vídeos



Fonte: Autor (2017).

Vale ressaltar que o usuário pode enviar mais de um vídeo e o último vídeo enviado é o que prevalece no servidor, ou seja, o anterior é sobreescrito pelo atual. Logo, somente são armazenados dois vídeos de cada indivíduo da pesquisa. Já em relação às fotografias, o armazenamento foi pensado para acumular o maior número possível de fotos de cada indivíduo. Dessa maneira, o usuário pode mandar quantas fotos desejar para o aumento do banco de dados da pesquisa.

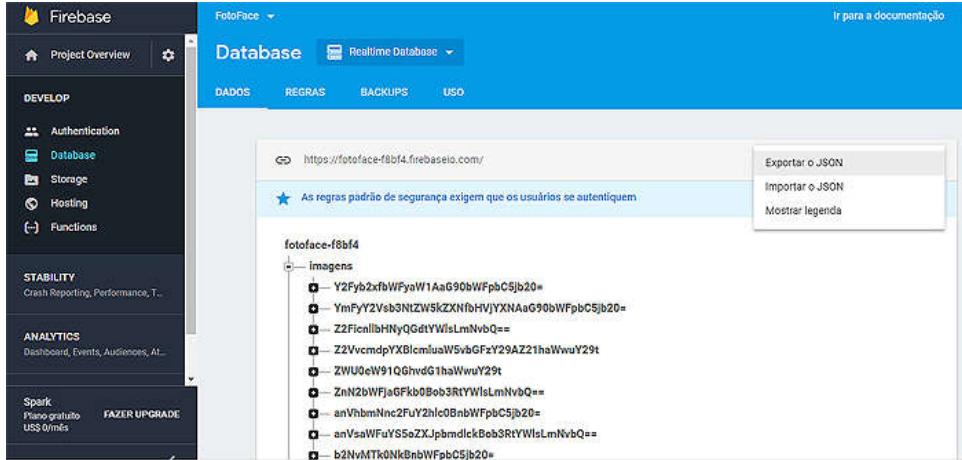
Todo processo de cadastro, captura de fotos e vídeos e o envio das amostras via *web*, foi programado para automatizar ao máximo a tarefa, e facilitar o processamento das amostras de treinamento, conforme será mostrado na próxima seção.

4.2 PROCESSAMENTO DAS AMOSTRAS DE TREINAMENTO

4.2.1 Visão geral

As etapas de processamento das imagens visando gerar amostras de treinamento, foi implementada, como dito anteriormente, de forma a automatizar o processo. A única etapa sem este escopo é o ato de baixar o arquivo do banco de dados no console do *Firebase* (GOOGLE, 2017), conforme Figura 29, de forma que deve-se fazer manualmente a exportação do referido arquivo.

Figura 29 - Console do *Firebase* (detalhe para a exportação da planilha)



Fonte: GOOGLE (2017).

Por meio da visualização da Figura 29, nota-se o padrão do banco de dados JSON, em que há “nós” de hierarquia (JSON, 2017). A tabela do banco de dados em questão possui três nós principais, sendo eles: imagens, usuários e vídeo. O “nó imagens” é onde se encontram os endereços de armazenamento de todas as imagens enviadas pelo usuário. Já no “nó usuários”, estão todas as informações inerentes ao cadastro de cada um e o “nó vídeos”, encontram-se os endereços de armazenamento dos dois vídeos de cada participante da pesquisa. O padrão *Firebase* não aceita caracteres especiais nos “nós”, como o “@” e espaços, então, visto que pode ocorrer o aparecimento de nomes de usuários iguais, para diferenciar cada usuário sem comprometer sua correta identidade, foi necessário converter os caracteres do e-mail para a base 64. Como se pode notar em cada “nó” vinculado ao “nó imagens”, há um código diferenciando cada usuário cadastrado. Essa conversão foi realizada no momento do cadastro e as informações já foram vinculadas ao novo código em questão.

4.2.2 Processamento das fotos cadastradas

A fim de automatizar o processo, foi desenvolvida uma rotina, em linguagem Python (PYTHON, 2017), para que por meio das informações contidas no banco de dados JSON, que foi exportado do console, fosse realizado o *download* das imagens. Após esse procedimento, todas as imagens foram armazenadas em uma pasta para o

treinamento da rede e então, uma nova rotina é iniciada para a contagem do número de fotos de cada usuário. Dessa forma, aleatoriamente 20% das imagens são movidas para a pasta de teste da rede. Esse procedimento visa isentar a pesquisa de possíveis manipulações. Durante o processo uma tabela associando o código em base 64 ao nome do usuário é gerada para a futura validação dos resultados da rede.

4.2.3 Processamento dos vídeos cadastrados

O processamento dos vídeos enviados demanda mais custo computacional e de tráfego, por se tratar de arquivos maiores. Um dos objetivos da captura de vídeo, além da contribuição de outras posições faciais, é a quantidade de amostras que podem ser extraídas dos quadros que foram capturados nos vídeos. Pensando nisso, foi necessário implementar uma rotina em Python para o processamento dos vídeos em questão.

Com as informações oriundas do banco de dados JSON, o *download* foi realizado de maneira automática, os vídeos salvos em um diretório e um arquivo Python foi gerado relacionando o vídeo ao usuário em questão. De cada vídeo é feita uma amostragem de 1 a cada 5 quadros para evitar o acumulo excessivo de fotos com pouca alteração, já que na maioria das câmeras a taxa de captura é de 30 quadros por segundo. Vale ressaltar que a repetição de fotos com pouca modificação de rotação da face, aumenta bastante o tempo e treinamento e não traz benefícios significativos para os resultados.

O quadro utilizado é rotacionado noventa graus no sentido anti-horário, já que a câmera no momento da gravação está na posição retrato e os vídeos de câmeras de celular são salvos na posição paisagem. Após rotacionadas, a parte superior e inferior das fotos é eliminada com o intuito de deixar a altura e a largura na mesma proporção e também, centralizar mais o rosto eliminando áreas de pouco interesse para o treinamento.

Toda edição de imagens foi desenvolvida na biblioteca OpenCV (OPENCV, 2017) e Python de forma totalmente automatizada.

4.3 CRIAÇÃO DO BANCO DE IMAGENS FACIAIS DE CELEBRIDADES

Para a criação do banco de imagens para treinamento da rede convolucional proposta, foi desenvolvida uma rotina em Python para a realização do *download* de imagens de 11 celebridades da lista do site IMDb (IMDB, 2017) escolhidas aleatoriamente a partir do buscador de imagens do Google. Desse forma, 150 imagens de cada uma das 11 celebridades foram armazenadas com o intuito de extrair a região facial destas imagens.

Para a extração da região de interesse, foi desenvolvida uma rotina em Python de localização de faces nas imagens baixadas. Para a localização da face, foi utilizado um classificador de *cluster* (aglomeração) da pele baseado no espaço de cor RGB, conforme apresentado nas equações descritas em (AHLVERS; RAJAGOPALAN; ZÖLZER, 2005)

$$R > 95 \text{ e } G > 40 > e B > 20 \quad (4)$$

$$\max(R, G, B) - \min(R, G, B) > 15 \quad (5)$$

$$|R - G| > 15 \quad (6)$$

$$R > G \text{ e } R > B \quad (7).$$

Ao identificar a região cuja cor aproxima-se à cor de pele, a nova rotina permitiu localizar o ponto central da face na imagem e a partir desse ponto, estipular o tamanho da região de interesse a ser capturada. Na Figura 30 é possível visualizar o resultado de uma segmentação correta.

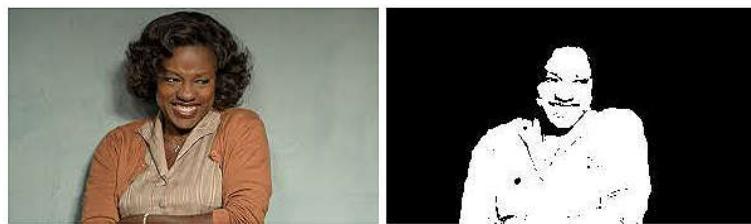
Figura 30 - Resultado de uma segmentação efetuada com sucesso



Fonte: Adaptado de IMDB (2017).

Em algumas imagens, a segmentação não foi como esperado e não foi possível localizar corretamente o centro da face, resultando na captura de uma área diferente da área de interesse. Por esse motivo, apenas 80 imagens de cada celebridade foram utilizadas para o treinamento da rede. A Figura 31 mostra o resultado de uma segmentação em que não foi possível localizar com exatidão o centro da face.

Figura 31 - Resultado de uma segmentação efetuada sem sucesso



Fonte: Adaptado de IMDB (2017).

O processo de identificação das imagens com problemas não foi automatizado e coube ao autor deste trabalho a seleção das melhores imagens. Ao final do processo, foram selecionadas ao todo, 880 imagens, ou seja, 80 imagens de cada celebridade. Na Figura 32 encontra-se uma amostra das imagens do banco de imagens faciais criado a partir das imagens processadas da internet.

Figura 32 - Exemplo de algumas imagens do banco criado



Fonte: Autor (2017).

Foram incorporadas ao banco de imagens faciais de celebridades criado, 20 imagens faciais do autor deste trabalho, que foram adquiridas por meio do aplicativo FotoFace, a fim de avaliar a possibilidade do uso deste aplicativo em pesquisas futuras. Também foram adicionadas ao banco, 30 imagens capturadas do vídeo esquerdo e 30 imagens

capturadas do vídeo direito, totalizando 80 imagens faciais adicionadas ao treinamento. Os referidos vídeos também foram capturados e enviados por meio do aplicativo FotoFace. Dessa forma, a rede foi treinada com 960 imagens a fim de classificar 12 indivíduos distintos, sendo que 20% destas imagens foram utilizadas para o processo de validação da rede.

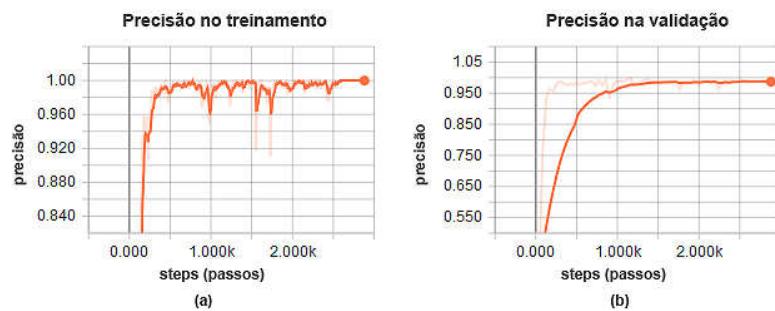
4.4 TREINAMENTO DA REDE

No início da rotina de treinamento, as imagens foram reduzidas para 100x100 pixels e foram, de maneira semelhante aos treinamentos anteriores, aleatoriamente sortidas para que as amostras da mesma classe não se encontrassem em sequência na ordem de processamento. Para o treinamento, foi utilizada uma taxa de aprendizagem de 0,0005, sendo utilizadas 80% das amostras para treinamento e 20% das imagens para validação, com o intuito de classificar 12 indivíduos diferentes a partir das 960 imagens adquiridas do banco de imagens faciais implementado e por meio do aplicativo FotoFace.

O treinamento foi executado no mesmo *hardware* utilizado nos treinamentos anteriores e foram necessárias 61 épocas de treinamento.

Assim, cada passo processou 16 imagens totalizando nas 61 épocas, 2928 passos de treinamento em um tempo de 3004,45 segundos. A precisão no treinamento foi de 99,45% e a precisão de validação de 98,16%, conforme mostrado na Figura 33.a e na Figura 33.b.

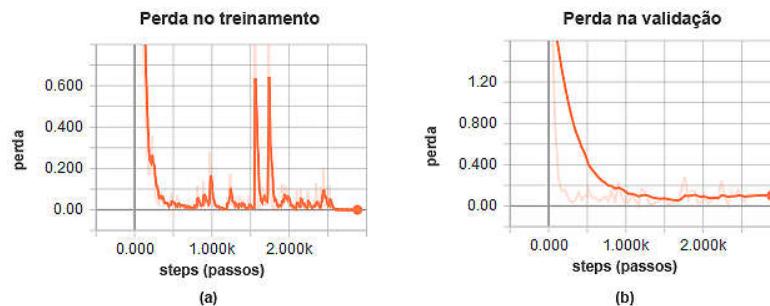
Figura 33- Precisão de treinamento e de validação da rede



Fonte: Autor (2017).

Sobre as perdas, mostradas na Figura 34.a e na Figura 34.b, no treinamento foi de 0,000043 e na validação foi de 0,081. É possível notar também que a perda se mantém com pouca ou nenhuma variação no momento em que a precisão se encontra estável, caracterizando que não houve *overfitting* na rede (HU et al., 2015).

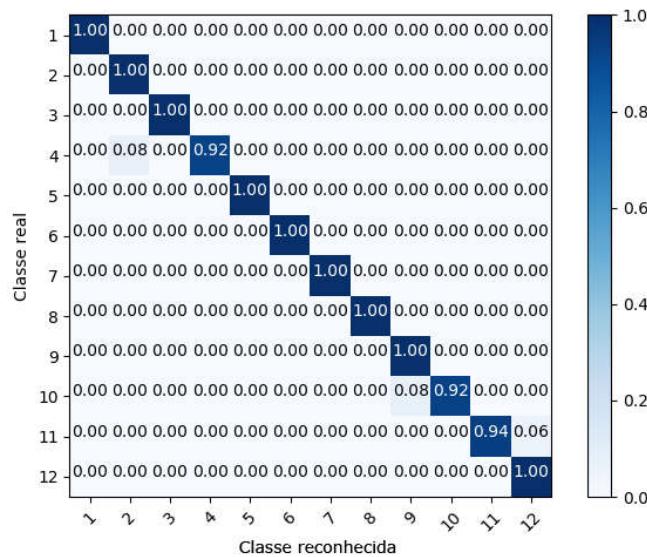
Figura 34 - Perda de treinamento e de validação da rede



Fonte: Autor.

Para uma melhor visualização do resultado, a matriz confusão mostrada na Figura 35, foi normalizada e seu resultado foi condizente com a precisão na validação.

Figura 35 – Matriz confusão normalizada da rede



Fonte: Autor (2017).

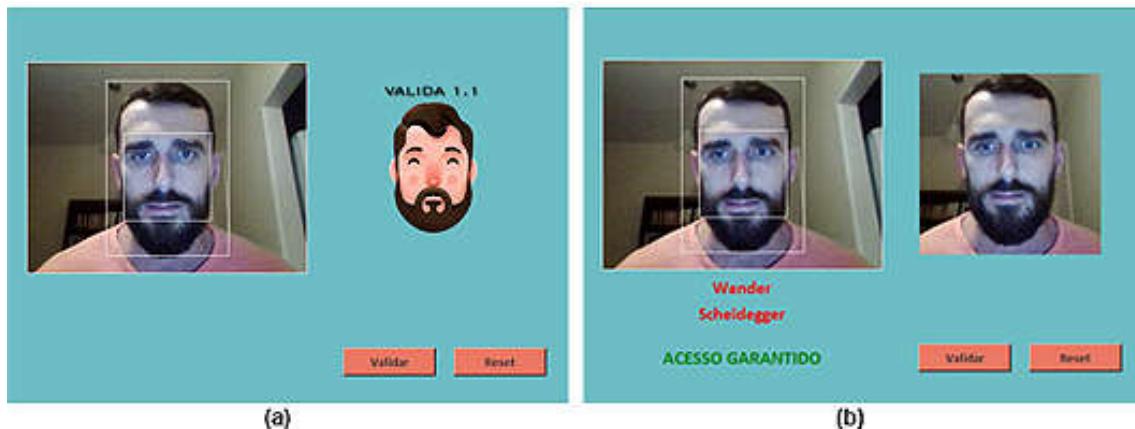
5 SIMULAÇÃO DA VALIDAÇÃO DE ACESSO EM TEMPO REAL

Um dos objetivos do trabalho é a criação de um programa, que utiliza a rede neural convolucional desenvolvida, treinada e validada, para simular o controle de acessos a ambientes em tempo real. Este capítulo descreve as etapas de criação do sistema de simulação de acessos em tempo real, que será chamado de VALIDA 1.0, além de apresentar os resultados da simulação de controle de acessos. Também será mostrada a estratégia utilizada para minimizar o problema de falsos positivos no acesso.

5.1 IMPLEMENTAÇÃO DO PROGRAMA

O referido programa Valida 1.0 foi implementado em linguagem *Python* utilizando a sua interface gráfica nativa, o módulo TKInter (PYTHON, 2016). A interface gráfica implementada pode ser visualizada na Figura 36.a.

Figura 36 – Interface gráfica do programa



Fonte: Autor (2017).

Por meio da *webcam* do computador, a imagem do indivíduo é capturada com o objetivo de reconhecer sua face validando a entrada. Ao clicar sobre o botão “Validar” a imagem capturada é validada na rede neural implementada. Caso o usuário tenha sido previamente cadastrado pelo aplicativo FotoFace e sua imagem conste no banco de imagens utilizado para treinar a rede neural, o programa exibe a mensagem “ACESSO GARANTIDO”, indicando que a rede reconheceu o usuário, como mostra a figura 36.b. Caso contrário, é exibida a mensagem “NÃO LOCALIZADO”, “ACESSO

NEGADO” e “Tente outra vez”, indicando que o usuário não foi identificado como cadastrado no sistema. O botão “Reset” prepara a rede para uma nova operação assim como retorna o *layout* da tela para o padrão inicial, conforme mostrado na figura 36.a.

A cada captura de imagem para validação, uma cópia é armazenada e passa a fazer parte do banco de imagens de treinamento da rede. Assim ao ser acionado um novo treinamento, a rede poderá alcançar mais precisão. Os referidos treinamentos podem ser agendados para os períodos em que a rede não atua. Caso não haja momentos como esses, a rede pode ser treinada paralelamente à sua operação e ao término, incorporada ao programa.

Vale ressaltar que ao ser validada, a foto salva é relacionada ao usuário validado. As fotos não validadas, não tem a possibilidade de serem associadas a algum usuário, já que é necessário reconhecer o indivíduo corretamente para efetuar a associação. Dessa forma, caso o indivíduo cadastrado não seja reconhecido pela rede, é possível que haja interferência humana, a fim de associar as imagens que não foram validadas ao usuário correto e assim, melhorar o aprendizado da rede. Dessa forma, de todas as etapas do processo, essa é uma das que necessitariam de interferência humana para a sua realização.

5.2 TESTES DE ACESSO EM TEMPO REAL

As imagens utilizadas para a validação do treinamento da rede, conforme já mencionado anteriormente, não são utilizadas no treinamento propriamente dito. Mesmo assim, por terem sido adquiridas muitas vezes no mesmo contexto, possuem certa similaridade. Essa similaridade é encontrada principalmente nas imagens processadas dos vídeos obtidos no cadastro do usuário com o aplicativo FotoFace. Mesmo que no processamento 4 de cada 5 imagens sejam eliminadas, ainda assim, a alta taxa de quadros por segundo das câmeras acaba imprimindo certa similaridade às imagens sequenciais capturadas desses vídeos. Esse fato, aliado à pouca quantidade de imagens para treinamento de cada indivíduo, faz com que o resultado inicial na aplicação prática não seja o esperado do treinamento.

Como esta pesquisa não envolveu voluntários em sua realização, para testar o sistema de acesso em tempo real desenvolvido neste projeto, o VALIDA 1.0, o próprio autor realizou seu cadastro por meio do aplicativo FotoFace. Vale ressaltar que o banco de dados da rede já contava com o cadastro de imagens de 11 celebridades.

Para isso, foram executados 20 testes de acesso. Em seis testes a identidade atribuída foi incorreta, totalizando uma precisão de 70%.

A fim de que a rede adquirisse mais precisão, um novo treinamento foi executado ao final dos testes, com as 20 imagens capturadas pelo programa de validação no momento dos testes de validação de acesso. Para atribuir a identidade correta às seis imagens não reconhecidas anteriormente, foi necessária interferência humana. Os parâmetros do treinamento foram os mesmos já descritos na seção 4.3, apenas foram adicionadas as novas imagens capturadas no teste às amostras de treinamento. Vale destacar que o modelo de treinamento anterior salvo foi continuado nos treinamentos subsequentes, a partir da situação final treinada, ou seja, o treinamento continuou a partir da época de treinamento que havia parado, sem reiniciar todo o processo de treinamento desde o início.

Nos novos testes, das 20 tentativas de acesso realizadas, em três, a identidade do autor foi atribuída erroneamente. Desse modo, como resultado no segundo teste, a rede alcançou uma precisão de 85%.

Após o término dos testes, mais um treinamento foi realizado inserindo as novas imagens adquiridas. O treinamento em questão também foi executado de acordo com os parâmetros anteriores de treinamento, e de forma análoga, a única variação foi o acréscimo das novas amostras ao treinamento.

Novamente após o treinamento, novos testes de validação de acesso foram executados, desta vez 40. Com o advento desse último, apenas em um teste o indivíduo não foi reconhecido contabilizando uma taxa de precisão de 97,5%.

Analizando a tendência de crescimento da taxa de acerto da rede, proporcionada pelo aprendizado constante, pode-se concluir que o autoaprendizado aparentemente melhorou a eficiência do sistema em questão. No entanto, embora tenha ocorrido uma melhora substancial na taxa de acertos, devido à impossibilidade de testar o acesso

dos outros indivíduos classificados pela rede, não é possível afirmar que este seja o padrão da rede com o acréscimo de novas amostras de diferentes classes, ou mesmo o acréscimo de novas classes. Além disso, também não foi possível mensurar se a inserção de novas amostras de apenas uma classe no treinamento, influenciou na classificação das demais classes que mantiveram suas quantidades de imagens de treinamento iguais.

Os testes descritos anteriormente foram realizados apenas com um indivíduo inseridos no universo do treinamento da rede. Na próxima seção, será descrita a alternativa utilizada para não validar as classes não inseridas no universo do treinamento, ou seja, a estratégia utilizada para evitar os falsos positivos.

5.3 ESTRATÉGIA PARA RESOLUÇÃO DO PROBLEMA DE FALSOS POSITIVOS

Na classificação, assume-se que existam um determinado conjunto de classes entre as quais deve-se classificar. Já no reconhecimento, assume-se que existam algumas classes que se pode reconhecer em um espaço muito maior de coisas em que não se pode reconhecer (BENDALE; BOULT, 2015a). Esse fato ocorre com o reconhecimento facial, objeto de estudo deste trabalho, pois o treinamento é realizado com um universo conhecido de classes e na validação um grande número de classes oriundas de um universo desconhecido são testadas.

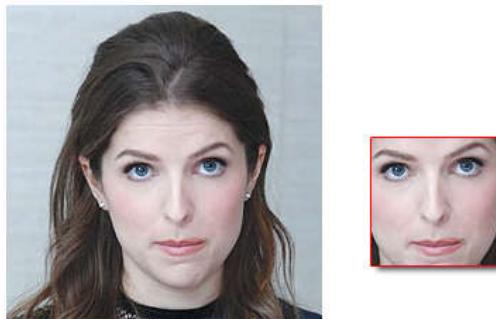
Na rede deste trabalho, como dito anteriormente, a última camada *fully connected* utiliza como ativação a função *SoftMax*, que produz uma distribuição de probabilidade sobre as classes do universo de treinamento. Nesse contexto, espera-se que a saída da rede sempre tenha uma classe mais provável, podendo esperar que, por uma entrada desconhecida, todas as classes teriam baixa probabilidade e esse limiar de incerteza rejeitaria essas classes desconhecidas. Mas, o limiar de incerteza não é suficiente para determinar com segurança o que é desconhecido (BENDALE; BOULT, 2015b).

Pensando em eliminar os riscos de acesso de pessoas não cadastradas, característica essencial de segurança de um sistema de acesso, conforme o proposto neste trabalho, a estratégia utilizada foi tornar a saída da rede binária, em que outra rede, com a mesma arquitetura descrita na seção 3.1, com o foco de reconhecimento

na região compreendida somente entre os olhos, nariz e boca, a fim de validar a entrada somente se nas duas predições o mesmo resultado de escolha da classe fosse alcançado. Essa estratégia visa eliminar o risco de que uma classe estranha obtenha um limiar alto na classificação e burle o sistema e também tem o objetivo de ser mais precisa na classificação das classes inseridas no treinamento.

Dessa forma, uma nova rede foi treinada a partir das mesmas imagens utilizadas para o treinamento da rede anterior. Foi extraída a região de interesse das referidas fotos, ou seja, a região que comprehende os olhos, nariz e boca, conforme visualizado na Figura 37.

Figura 37 - Exemplo da região de interesse extraída



Fonte: Adaptado de IMDB (2017).

Para a localização da face, foi utilizado o mesmo classificador de *cluster* da pele baseado no espaço de cor RGB, utilizado anteriormente na criação do banco de imagens faciais da internet.

Após o treinamento da rede com as imagens criadas a partir da região de interesse mencionada anteriormente, o programa VALIDA 1.0 foi alterado de forma que, além de salvar a foto da face em sua totalidade, também salva a região de interesse mostrada na Figura 37. De posse das duas regiões, o programa faz a predição das duas redes e analisa o resultado. Se as saídas das duas redes indicarem a mesma classe, e o limiar de 99% de precisão for alcançado em ambas, a validação é confirmada e o acesso é então liberado.

Para capturar corretamente a região de interesse, é necessário que no momento da validação o usuário esteja com a região de interesse ocupando a área do quadrado menor, conforme mostrado anteriormente na Figura 36.a.

A fim de comprovar se o que foi proposto para diminuir a ocorrência de falsos positivos realmente seria eficaz, cerca de 40 acessos com um número de 20 indivíduos distintos, selecionados aleatoriamente e que estavam fora do universo de treinamento e portanto, não poderiam ter seu acesso validado no sistema, foram realizados. Em nenhuma das 40 tentativas o acesso foi validado, comprovando o aumento de eficiência na estratégia adotada.

Para que a nova estratégia não influenciasse o tempo de processamento, já que o sistema passou a utilizar duas redes na validação, foi preciso carregar os dois modelos da rede na inicialização do programa, tornando seu carregamento um pouco mais demorado. O tempo de carregamento do programa passou de 5 segundos para 9,5 segundos, já a diferença no tempo de validação foi quase imperceptível, culminando em um tempo de menos de 0,3 segundo para cada foto validada.

6 CONCLUSÃO

6.1 CONCLUSÕES GERAIS

Neste trabalho, foi desenvolvido um sistema de controle de acesso por reconhecimento facial. Para isso, uma rede neural convolucional foi implementada, treinada e validada. Além da rede neural convolucional, também foram implementadas duas interfaces com o usuário: a primeira, um aplicativo para Android, que possibilita o cadastro e captura de fotos do usuário do sistema; a segunda, a interface para o controle de acesso, que, a partir da rede neural implementada e treinada, libera o acesso do usuário caso sua face seja reconhecida dentre as imagens utilizadas para o treinamento anterior da rede. Além disso, foi proposta uma estratégia para reduzir falsos positivos. Como esses aplicativos, foi possível simular o controle de acesso, aplicando na prática a rede desenvolvida.

Os testes preliminares da rede neural convolucional proposta mostraram que a mesma obteve desempenho satisfatório, com resultados comparáveis aos encontrados na literatura.

O teste do sistema completo foi realizado com 12 diferentes indivíduos. Onze desses indivíduos eram celebridades e compuseram um banco de dados para o teste de acesso. O 12º indivíduo do teste foi cadastrado pelo sistema FotoFace. A rede neural desenvolvida obteve bons resultados, chegando a um acerto de 70%.

Como forma de melhorar o desempenho da rede, as imagens capturadas durante o teste de acesso foram utilizadas em um processo de autoaprendizado da rede. Desse modo, ao adicionar novas imagens do indivíduo cadastrado, a rede alcançou uma precisão de 85%, o que sugere que o acréscimo de novos dados melhora significativamente seu desempenho. Desse modo, o sistema deve implementar sistematicamente a captura de novos dados a cada reconhecimento. Vale ressaltar que, para testar o sistema proposto de modo global, é necessário realizar novos testes com voluntários, a fim de simular o uso real do aplicativo e do sistema de reconhecimento de face.

Além disso, foi possível perceber que a rede que foi treinada com as imagens de uma pequena região da face e sem angulação obteve resultados de precisão similares aos

da rede que utilizou em seu treinamento as imagens de toda a região da face e com angulação. Logo, mais testes devem ser feitos a fim de avaliar a eficiência da utilização de ângulos maiores que 30 graus de rotação nas faces, já que na validação, somente é capturada a foto na posição frontal ao indivíduo.

Em relação ao número de classes para treinamento, o resultado dos testes realizados mostram que a rede pode ser utilizada para um maior número de classes. No treinamento pra 100 classes, utilizando as imagens do FEI *Database*, a rede obteve bom desempenho para reconhecimento. No entanto, foi necessário utilizar um maior número de amostras de treinamento para cada classe. Esse fato também foi comprovado com os resultados de treinamento para 20 classes, utilizando as imagens do ORL *Database*, em que foi preciso aumentar a quantidade de amostras de treinamento para um resultado condizente ao resultado do sistema híbrido apresentado na literatura. Ademais, devido a algumas instabilidades no treinamento, para um número acima de 100 classes a arquitetura deve ser modificada.

6.2 TRABALHOS FUTUROS

Alguns trabalhos futuros podem ser propostos a partir dos resultados deste projeto. O primeiro deles é realizar testes com o aplicativo FotoFace, para cadastrar e capturar imagens de voluntários. Desse modo, seria possível testar o comportamento do aplicativo FotoFace na tarefa de cadastro, captura e envio de fotos e vídeos de uma grande quantidade de indivíduos a fim de validar sua eficiência na execução desta tarefa. Tais testes também permitiriam analisar o comportamento de todo o sistema de acesso utilizando fotos mais variadas e em maior número.

Outro trabalho que pode ser desenvolvido é o de um sistema de captura de imagens em ambientes de trabalho a fim de adquirir um extenso banco de imagens faciais dos funcionários de uma empresa para o treinamento de uma rede neural de reconhecimento facial mais robusta. Além disso, o sistema deveria registrar de horário de entrada e saída dos funcionários identificados. Dessa forma, automaticamente, a rede aprenderia diariamente podendo, por exemplo, validar o ponto dos funcionários e controlar o acesso aos vários ambientes da empresa.

Também é possível propor o reconhecimento facial de indivíduos a partir de câmeras de vídeo, em que um algoritmo de localização de faces mais robusto seja implementado para automatizar o processo, utilizando algoritmos de identificação de face robustos e amplamente utilizados na literatura como (JONES; VIOLA, 2003). Nesse caso, deve-se efetuar testes com mais indivíduos para testar efetivamente se a arquitetura em questão comporta-se bem para um número acima de 100 classes, focando os estudos em processamento paralelo para que várias faces do enquadramento sejam avaliadas ao mesmo tempo.

REFERÊNCIAS

AHLVERS, U.; RAJAGOPALAN, R.; ZÖLZER, U. **Model-free face detection and head tracking with morphological hole mapping**. Signal Processing Conference. Anais...Antalya, Turkey: 2005

ARYA, K. V.; ADARSH, A. An Efficient Face Detection and Recognition Method for Surveillance. **2015 International Conference on Computational Intelligence and Communication Networks (CICN)**, p. 262–267, 2015.

BENDALE, A.; BOULT, T. Towards Open World Recognition. **Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, p. 1893–1902, 2015a.

BENDALE, A.; BOULT, T. Towards Open Set Deep Networks. **ArXiv e-prints**, 2015b.

BORADE, S. N.; DESHMUKH, R. R.; RAMU, S. **Face recognition using fusion of PCA and LDA: Borda count approach**. 24th Mediterranean Conference on Control and Automation, MED 2016. Anais...2016

BOUGHRARA, H. et al. Facial expression recognition based on a mlp neural network using constructive training algorithm. **Multimedia Tools and Applications**, v. 75, p. 729–739, 2014.

CAMBRIDGE, A. L. **ORL Face Database**. Disponível em: <<http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>>. Acesso em: 11 nov. 2017.

CANDÉS, E. J. et al. Robust Principal Component Analysis. **Journal of the ACM**, v. 58, n. 3, p. 1–37, 2009.

CRAW, I.; CAMERON, P. Face Recognition by Computer. **Procedings of the British Machine Vision Conference**, p. 52.1-52.10, 1992.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. **Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005**, v. 1, p. 886–893, 2005.

GARCIA, C.; DELAKIS, M. Convolutional face finder: A neural architecture for fast and robust face detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 26, n. 11, p. 1408–1423, nov. 2004.

GEITGEY, A. **Deep Learning and Convolutional Neural Networks**. Disponível em: <<https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721#.ahj0r09q1>>. Acesso em: 15 nov. 2016.

GOOGLE. **Android Developers**. Disponível em: <<https://developer.android.com/develop/index.html>>. Acesso em: 20 nov. 2017.

GOOGLE. TensorFlow — an Open Source Software Library for Machine Intelligence. Disponível em: <<https://www.tensorflow.org/>>. Acesso em: 19 nov. 2016.

GOOGLE. Firebase. Disponível em:
<https://console.firebaseio.google.com/u/1/project/fotoface-f8bf4/database/data>.
 Acesso em: 10 nov. 2017.

GUO, Y. et al. Deep learning for visual understanding: A review. **Neurocomputing**, v. 187, p. 27–48, 2016.

GUYTON, L. et al. Guyton and Hall Textbook of Medical Physiology. **Igarss 2014**, n. 1, p. 577–578, 2014.

HAN, C. C. et al. Fast face detection via morphology-based pre-processing1. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 1311, p. 469–476, 1997.

HAYKIN, S. **Neural Networks and Learning Machines**. 3. ed. New Jersey: Pearson, 2009.

HE, K. et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. **Proceedings of the IEEE International Conference on Computer Vision**, v. 2015 Inter, p. 1026–1034, 2015.

HU, W. et al. Deep convolutional neural networks for hyperspectral image classification. **Journal of Sensors**, v. 2015, p. 1–12, 2015.

IMDB. **IMDb - Movies, TV and Celebrities - IMDb**. Disponível em:
<http://www.imdb.com/>. Acesso em: 9 dez. 2017.

JOLLIFFE, I. T. Principal Component Analysis, Second Edition. **Encyclopedia of Statistics in Behavioral Science**, v. 30, n. 3, p. 487, 2002.

JONES, M.; VIOLA, P. Fast Multi-view Face Detection. **Mitsubishi Electric Research Lab TR2000396**, 2003.

JSON. **JavaScript Object Notation**. Disponível em: <<https://www.json.org/>>. Acesso em: 19 nov. 2017.

KINGMA, D. P.; BA, J. Adam: A Method for Stochastic Optimization. **ArXiv e-prints**, p. 1–15, 2014.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet Classification with Deep Convolutional Neural Networks. **Advances In Neural Information Processing Systems**, p. 1–9, 2012.

LAB, S. V. **ImageNet**. Disponível em: <<http://image-net.org/>>. Acesso em: 18 nov. 2017.

LAWRENCE, S. et al. Face Recognition : A Convolutional Neural Network Approach.

Neural Networks, v. 8, n. 1, p. 98–113, 1997.

LECUN, Y. et al. Backpropagation Applied to Handwritten Zip Code Recognition. **Neural Computation**, v. 1, n. 4, p. 541–551, 1989.

LECUN, Y.; BENGIO, Y. Convolutional networks for images, speech, and time series. **The handbook of brain theory and neural networks**, v. 3361, n. April 2016, p. 255–258, 1995.

LIU, T. et al. Implementation of Training Convolutional Neural Networks. **ArXiv e-prints**, 2015.

MIT. **TFLearn**. Disponível em: <<http://tflearn.org/>>. Acesso em: 8 nov. 2017.

OPENCV. **OpenCV library**. Disponível em: <<https://opencv.org/>>. Acesso em: 19 nov. 2017.

ORACLE. **JavaDoc**. Disponível em:
[<https://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html>](https://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html).
Acesso em: 20 nov. 2017.

PYTHON. **TkInter - Python Wiki**. Disponível em:
[<https://wiki.python.org/moin/TkInter>](https://wiki.python.org/moin/TkInter). Acesso em: 24 nov. 2017.

PYTHON. **Python Documentation**. Disponível em: <<https://www.python.org/>>. Acesso em: 19 nov. 2017.

RUBINSTEIN, R. Y. et al. **The Cross-Entropy Method**. 1. ed. Nova York: Springer-Verlag, 2004.

SILVA, I. N.; SPATTI, D. H.; FLAUZINO, R. A. **Redes Neurais Artificiais Para Engenharia e Ciências Aplicadas**. São Paulo: Artliber Editora Ltda, 2010.

SRIVASTAVA, N. et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. **Journal of Machine Learning Research**, v. 15, p. 1929–1958, 2014.

TANG, Y. Deep Learning using Linear Support Vector Machines. **ArXiv e-prints**, 2013.

THOMAZ, C. E. **FEI Face Database**. Disponível em:
[<http://fei.edu.br/~cet/facedatabase.html>](http://fei.edu.br/~cet/facedatabase.html). Acesso em: 11 nov. 2017.

ZEILER, M. D.; FERGUS, R. **Visualizing and understanding convolutional networks**. Lecture Notes in Computer Science. **Anais...** Springer Verlag, 2014

ZHANG, Y. et al. Adaptive Convolutional Neural Network and Its Application in Face Recognition. **Neural Processing Letters**, v. 43, n. 2, p. 389–399, 2016.

ZHAO, Q. et al. **Robust Principal Component Analysis with Complex Noise**. Proceedings of the 31st International Conference on Machine Learning (ICML-14). **Anais...** 2014Disponível em: <<http://jmlr.org/proceedings/papers/v32/zhao14.pdf>>

ZHAO, W. et al. Face recognition: A literature survey. **Acm Computing Surveys**, v. 35, n. 4, p. 399–458, 2003.