UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

Nicolas Hiroaki Shitara

Avaliação de desempenho de uma CNN para o reconhecimento da região periocular utilizando *transfer learning* e descritores locais de textura

São Carlos

2019

Nicolas Hiroaki Shitara

Avaliação de desempenho de uma CNN para o reconhecimento da região periocular utilizando *transfer learning* e descritores locais de textura

Monografia apresentada ao Curso de Engenharia Elétrica com Ênfase em Eletrônica, da Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Adilson Gonzaga

São Carlos

2019

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da EESC/USP com os dados inseridos pelo(a) autor(a).

S555a

Shitara, Nicolas Hiroaki
Avaliação de desempenho de uma cnn para o reconhecimento da região periocular utilizando transfer learning e descritores locais de textura / Nicolas Hiroaki Shitara; orientador Adilson Gonzaga. São Carlos, 2019.

Monografia (Graduação em Engenharia Elétrica com ênfase em Eletrônica) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2019.

1. redes neurais convolucionais. 2. transfer learning. 3. descritores locais de textura. 4. região periocular. I. Título.

Eduardo Graziosi Silva - CRB - 8/8907

FOLHA DE APROVAÇÃO

Nome: Nícolas Hiroaki Shitara

Título: "Avaliação de desempenho de uma CNN para o reconhecimento da região periocular utilizando transfer learning e descritores locais de textura"

Trabalho de Conclusão de Curso defendido e aprovado em 27 / 11 / 2019,

com NOTA 9,7 (NOVE, SETE), **pela Comissão Julgadora:**

Prof. Associado Adilson Gonzaga - Orientador - SEL/EESC/USP (docente aposentado)

Prof. Associado Evandro Luis Linhari Rodrigues - SEL/EESC/USP (docente aposentado)

Dra. Carolina Toledo Ferraz - Pós-Doutorado/Faculdade Campo Limpo Paulista

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado Rogério Andrade Flauzino

AGRADECIMENTOS

Agradeço primeiramente, o Prof. Dr. Adilson Gonzaga e o pessoal do LAVI, em especial Carolina Toledo Ferraz e Osmando Pereira Jr. por todo apoio, paciência e orientações oferecidos durante a elaboração deste trabalho.

A minha família, Mônica, Toshiaki, Aya e Aline, que sempre me auxiliaram e motivaram em momentos difíceis, compartilharam momentos felizes e me inspiram mais do que tudo.

Os colegas da faculdade que me ajudaram e contribuiram, diretamente e indiretamente, durante toda a graduação.

E um agradecimento especial aos amigos que levo para a vida, pelas risadas e por preservarem minha sanidade: Gabriel "Palhaço" Laureano, Eduardo "duduzinho" Nishi, Éric "Mestre" Pizzini, Lucas "lulu" Neiro, André "DÉÉ" Ribeiro e André "responsável" Melzi.

“Nós somos uma forma do cosmos se autoconhecer.”

Carl Sagan

RESUMO

SHITARA, N. H. **Avaliação de desempenho de uma CNN para o reconhecimento da região periocular utilizando *transfer learning* e descritores locais de textura**. 2019. 58p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2019.

Este trabalho propõe a avaliação de desempenho de uma rede neural convolucional pré-treinada, originalmente com a base de imagens ImageNet, na tarefa de classificação de imagens da região periocular utilizando o método de ajuste fino, assim como uma análise de viabilidade do pré-processamento das imagens de entrada usando os descritores locais de textura LBP e LMP. A rede utilizada foi o modelo pré-treinado da AlexNet, para classificação de imagens da base *ND-CrossSensor-Iris-2013 Data Set*. Foram criados conjuntos de treinamento e teste, utilizando as imagens originais, aplicando os descritores LBP e LMP, e realizando uma combinação dos três conjuntos diferentes em cada uma das três camadas da imagem. Os treinos e testes realizados só apresentaram valores siginificativos (acima de 90%) quando os conjuntos de testes correspondiam ao tipo de imagem utilizado no treinamento, revelando que as características extraídas pela rede no treinamento não eram discriminantes para os diferentes conjuntos de teste. Além disso, foi verificado que o maior valor de acurácia (98,96%) é obtido quando se realiza o treino e teste com o conjunto de imagens originais, todos os conjuntos pré-processados com os descritores de textura apresentaram resultados um pouco inferiores, levando a conclusão que apesar do desempenho geral da classificação apresentar acurácias acima de 97%, a realização do pré-processamento de imagens não é viável quando é levado em conta o tempo e custo computacional.

Palavras-chave: Redes neurais convolucionais. *Transfer learning*. Descritores locais de textura.

ABSTRACT

SHITARA, N. H. **Performance evaluation of a CNN for recognition of the periocular region using transfer learning and local texture descriptors**. 2019. 58p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2019.

This study proposes a performance evaluation of a pre-trained convolutional neural network, originally with the ImageNet dataset, for the task of recognition of the periocular region, using fine-tuning. Also, it analises the viability of pre-processing the input images using the local texture descriptors LBP and LMP. The chosen network was a pre-trained model of AlexNet, to classify images of the ND-CrossSensor-Iris-2013 Data Set. A few training e test sets were created, using the original images of the dataset, aplying the the LBP and LMP descriptors and combining the three different sets in each of the three layers of the image. The results of the training and tests were only significant (with accuracy above 90%) when the training and test sets were composed of the same type of image, showing that the features extracted by the network were not discriminant for the different test sets. Moreover, the highest accuracy value was obtained training and testing with the original image set, all of the texture descriptors pre-processed sets showed lower accuracy values. In conclusion, even though the overall performance of the method described showed accuracies above 97%, when accounting the time spent and computational cost, it's not worth it to pre-process the images with local texture descriptors.

Keywords: Convolutional neural networks. Transfer learning. Local texture descriptors.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CNN	*Convolutional Neural Networks*
ILSVRC	*ImageNet Large Scale Visual Recognition Challenge*
LBP	*Local Binary Pattern*
LMP	*Local Mapped Pattern*
ReLU	*Rectified Linear Unit*
TU	*Texture Unit*

SUMÁRIO

1 INTRODUÇÃO

A utilização de redes neurais convolucionais, do inglês *Convolutional Neural Network* (CNN), tem contribuído para grandes avanços na área de visão computacional e classificação de imagens. Essas redes podem ser treinadas utilizando um número grande de dados, a fim de classificar acuradamente objetos presentes em imagens (KRIZHEVSKY; SUTSKEVER; HINTON, 2012; ZEILER; FERGUS, 2014). Em particular, a identificação e classificação de diferentes tipos de peculiaridades biométricas, como as da íris e região periocular, é um tópico estudado nesse campo com aplicações que podem inovar e auxiliar áreas como o de acesso e segurança, identificação criminal e saúde (ZHAO; KUMAR, 2016; NGUYEN et al., 2017).

A primeira CNN foi proposta em 1998 por LeCun et al., que tinha como objetivo identificar dígitos manuscritos. Porém, a grande disseminação da utilização de redes neurais convolucionais na área de visão computacional aconteceu apenas em 2012, quando a rede AlexNet proposta por Krizhevsky, Sutskever e Hinton (2012), venceu o ILSVRC (*ImageNet Large Scale Visual Recognition Challenge*), competição que avalia o desempenho de algoritmos na tarefa de classificação para a base ImageNet (DENG et al., 2009) que contém mais de 3 milhões de imagens. A partir desse marco obtido pela AlexNet, a aplicação mais popular das CNNs é no reconhecimento de padrões em imagens.

1.1 MOTIVAÇÃO

A região periocular se refere a região ao redor do olho, incluindo o globo ocular, pálpebra, cílios, sombrancelha e pele, e pode ser considerada a região com mais características discriminantes da face humana (ALONSO-FERNANDEZ; BIGUN, 2016). Além disso, apresenta uma maior "resistência"a variações devido ao envelhecimento (JUEFEI-XU et al., 2011) e cirurgia plástica (JILLELA; ROSS, 2012), quando comparada a face inteira. Por ter essas características, estudos da região periocular vem obtendo destaque nos últimos anos (HERNANDEZ-DIAZ; ALONSO-FERNANDEZ; BIGUN, 2018).

Mesmo com a popularidade das CNNs nas tarefas de visão computacional, suas aplicações envolvendo biometria são limitadas, com estudos recentes na área de reconhecimento e detecção facial (PARKHI et al., 2015; LI et al., 2015), reconhecimento de íris (NGUYEN et al., 2017) e segmentação de imagens (BHANU; KUMAR, 2017). Uma das razões para a limitação de pesquisas é a quantidade de dados de treinamento necessária para métodos de aprendizado profundo. Porém, com a utilização de redes pré-treinadas e o método de *transfer learning*, com os quais é possível obter valores altos de acurácia (RAZAVIAN et al., 2014; SHIN et al., 2016), é possível obter redes capazes de realizar o reconhecimento de imagens da região periocular sem a necessidade de desenvolver e treinar uma nova do início.

A utilização de descritores de textura LBP em tarefas de classificação foi disseminada pelo trabalho de Ojala, Pietikäinen e Mäenpää (2002), e recentemente, o trabalho de Zhang et al. (2017) propôs um método de identificação facial utilizando o mapa de características LBP como entrada de uma rede neural convolucional, obtendo um ganho de 3,5% na acurácia, quando comparado a utilização das imagens sem o pré-processamento. Porém, o problema de identificação facial proposto é simples quando comparado aos de classificação do ILSVRC (a rede apenas indica a se a imagem corresponde ou não, a de uma face).

Devido ao pequeno número de estudos realizados utilizando os descritores de textura locais LBP e LMP em conjunto com as redes neurais convolucionais, é motivação deste trabalho estudar esses métodos na tarefa de classificação e investigar os resultados obtidos por eles utilizando imagens da região periocular.

1.2 OBJETIVOS

O objetivo deste trabalho é avaliar o desempenho de uma rede neural convolucional na tarefa de classificação de imagens da região periocular, quando estas são pré-processadas utilizando descritores locais de textura LBP e LMP. Adicionalmente, verificar o comportamento das redes quando estas são treinadas utilizando as imagens sem pré-processamento e testadas com as imagens pré-processadas e vice-versa. Por fim, analisar a viabilidade da realização de tal pré-processamento no problema de classificação proposto.

1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho está dividido em 5 capítulos: O primeiro trata da introdução ao assunto abordado neste trabalho, incluindo a sua motivação e objetivos; o segundo apresenta a revisão bibliográfica necessária para o entendimento do trabalho; o terceiro contém o método proposto para a avaliação do desempenho de uma CNN na tarefa de classificação de imagens da região periocular aplicados descritores locais de textura; o quarto apresenta os resultados obtidos e suas respectivas discussões; e por fim, o quinto capítulo apresenta a conclusão do trabalho.

2 REVISÃO BIBLIOGRÁFICA

2.1 ARQUITETURA DAS REDES NEURAIS CONVOLUCIONAIS

Uma rede neural convolucional simples é composta por uma sequência de camadas, na qual cada camada possui uma função distinta a fim de transformar volumes de dados. Os três tipos principais de camadas necessárias para construir uma CNN são: camada de convolução, camada de *pooling* e camada totalmente conectada.

Além das camadas citadas, é comum a inclusão de outros tipos de camadas como as de ativação e normalização, a fim de melhorar o desempenho geral da rede. A arquitetura de uma rede neural convolucional completa é formada combinando esses diferentes tipos de camadas (LI; KARPATHY; JOHNSON, 2017a).

2.1.1 Camada de entrada

A camada de entrada em uma CNN recebe as imagens de entrada que são interpretadas como uma matriz de intensidades (ou um volume de ativações) de três dimensões: largura, altura e profundidade. A largura e altura se referem às dimensões da imagem e a profundidade à sua quantidade de canais, que pode assumir o valor 1 para imagens na escala de cinza e 3 para imagens coloridas (R, G e B) (ZHENG et al., 2016; LI; KARPATHY; JOHNSON, 2017a).

2.1.2 Camada de convolução

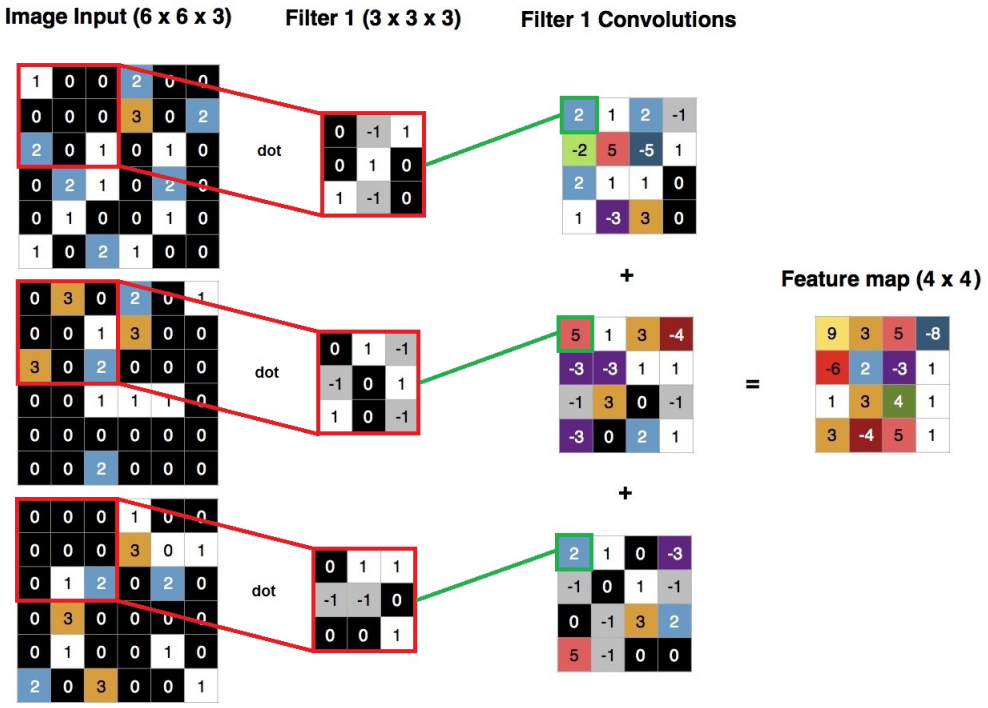
A camada de convolução é um dos componentes principais de uma CNN, e é nela que ocorre a maior parte do processamento computacional. Nesta camada são aplicados filtros que possuem dimensões (largura e altura) reduzidas, porém que percorrem todo o volume da entrada, produzindo mapas de características que representam a resposta desse filtro para toda posição espacial da entrada. A combinação desses mapas de características gerados pelos diferentes filtros, representam a saída da camada de convolução. A Figura 1 mostra um exemplo simplificado da geração de mapas de características.

A rede irá intuitivamente, aprender filtros que gerem mapas de características que possuem informações de características e padrões visuais mais simples, como uma borda orientada ou um aglomerado de cores únicas, em camadas próximas a entrada da rede. Já em camadas mais profundas, os mapas contém informações de padrões mais complexos.

Os hiperparâmetros (parâmetros definidos pelo desenvolvedor da rede) que controlam o tamanho da saída são: o número (K) e o tamanho (F) dos filtros, o passo (S) (corresponde a quantos pixels o filtro irá "andar"no processo de convolução) e o *padding* (P) (um preenchimento nas bordas da imagem, geralmente com valor igual a 0). Para uma entrada com dimensões (W_1 x H_1 x D_1) e considerando os hiperparâmetros mencionados anteriormente, a saída pode

ser representada por uma matriz de dimensões (W_2 x H_2 x D_2), determinada pelas equações 2.1, 2.2, 2.3 (LI; KARPATHY; JOHNSON, 2017a).

Figura 1: Processo de geração dos mapas de características



Fonte: (ANIEMEKA, 2017)

$$W_2 = \frac{W_1 - F + 2P}{S} + 1 \tag{2.1}$$

$$H_2 = \frac{H_1 - F + 2P}{S} + 1 \tag{2.2}$$

$$D_2 = K \tag{2.3}$$

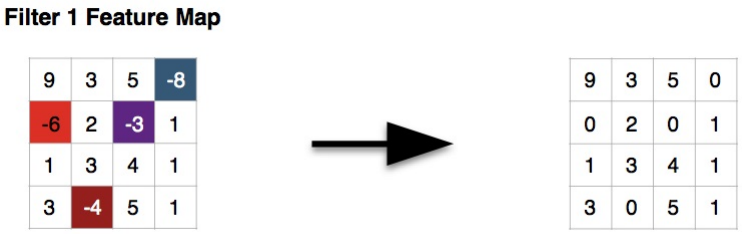
2.1.3 Camada de ativação

A camada de ativação (ou de não-linearidade) geralmente é encontrada logo após uma camada de convolução, e tem como função decidir se cada região da imagem foi acionada ou não. Essa camada, ao contrário da de convolução, não possui hiperparâmetros e entre as funções de ativação não-lineares que podem ser utilizadas nessa camada, estão a Sigmoid, Tanh, *Rectified Linear Units* (ReLU) (NAIR; HINTON, 2010) e suas derivações. Atualmente a função ReLU é amplamente utilizada por apresentar taxas de convergência mais rápidas do que as funções Sigmoid e Tanh, como exposto no trabalho de Krizhevsky, Sutskever e Hinton (2012).

A função ReLU é definida pela Equação 2.4, onde os valores acima de 0 seguem uma função linear, enquanto valores abaixo de 0 assumem valor 0. A Figura 2 exemplifica o processo da aplicação da função ReLU no mapa de características da Figura 1 (LI; KARPATHY; JOHNSON, 2017a; LI; KARPATHY; JOHNSON, 2017c).

$$f(x) = max(0, x) \tag{2.4}$$

Figura 2: Processo de aplicação da função ReLU



Fonte: (ANIEMEKA, 2017)

2.1.4 Camada de *Pooling*

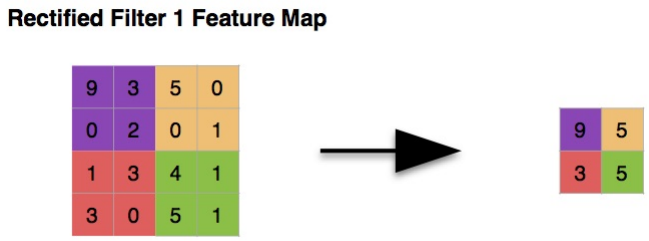
A camada de *pooling* é comumente utilizada em CNNs entre camadas de convolução com o propósito de reduzir progressivamente o tamanho dos mapas de características (*downsampling*), o número de parâmetros, custo computacional e, consequentemente, tornando a rede mais robusta a sobreajuste (quando a rede apresenta resultados ótimos com o conjunto de treinamento e validação, porém obtém desempenho insatisfatório com novas amostras).

A operação de *pooling* pode ser executada por algumas diferentes funções, como a *average pooling* e *L2-norm pooling*, porém a mais utilizada atualmente é a de *MAX pooling*, que vem apresentando melhor desempenho na prática (SCHERER; MÜLLER; BEHNKE, 2010).

A função de *MAX pooling* é aplicada independentemente em cada canal da matriz de entrada, reduzindo sua largura e altura, mas mantendo a profundidade inicial. O procedimento mais comum, é a aplicação de um filtro de máximo com dimensão 2x2 e um passo de valor 2, no qual o valor máximo dentro da janela é preservado e o resto descartado. A Figura 3 exemplifica a operação no mapa de características obtido na Figura 2.

Sendo a entrada dessa camada uma matriz de dimensões (W_1 x H_1 x D_1) e considerando os dois hiperparâmetros necessários da camada (o passo 'S' e a dimensão do filtro 'F'), a saída será representada por uma matriz de dimensões (W_2 x H_2 x D_2), determinada pelas Equações

Figura 3: Processo de aplicação da função *MAX Pooling*



Fonte: (ANIEMEKA, 2017)

2.5, 2.6, 2.7 (LI; KARPATHY; JOHNSON, 2017a).

$$W_2 = \frac{W_1 - F}{S} + 1 \tag{2.5}$$

$$H_2 = \frac{H_1 - F}{S} + 1 \tag{2.6}$$

$$D_2 = D_1 \tag{2.7}$$

2.1.5 Camada totalmente conectada

A camada totalmente conectada é a última que possui pesos em uma CNN, e recebe esse nome por ter cada neurônio da camada, conectado a todos os neurônios da camada anterior. Assim, a quantidade de neurônios na entrada corresponde ao número de elementos da camada anterior, e recebe como entrada, os pixels dispostos em forma vetorial. O número de neurônios na saída equivale a quantidade de classes, e sua saída apresenta a pontuação de cada um desses neurônios para as classes correspondentes.

Nesta camada, ocorre o processamento das informações do mapa de características recebido da camada anterior, e avalia a probabilidade dessa imagem pertencer a uma determinada classe. Os pesos são ajustados ao longo do treinamento, de modo que a pontuação seja alta quando a imagem pertencer a mesma classe que o neurônio representa, ou seja baixa caso contrário. Como exemplo, é considerada uma camada possuindo 2 neurônios na saída, representando 2 classes denominadas "cachorro"e "gato". Caso a imagem avaliada apresente características pertinentes a classe "cachorro", os pesos da camada serão atualizados de forma que a pontuação para a classe "cachorro"tenha uma valor alto, enquanto a pontuação para a classe "gato"tenha um valor baixo (LI; KARPATHY; JOHNSON, 2017a; LI; KARPATHY; JOHNSON, 2017c).

Um sistema de *dropout* é comumente implementado nessa camada a fim de reduzir um possível sobreajuste da rede. O processo faz com que alguns neurônios sejam "desligados"pela duração de uma iteração do treinamento, de forma a impedir a coadaptação excessiva desses neurônios (SRIVASTAVA et al., 2014).

2.1.6 Camada de classificação

A camada de classificação tem a função de classificar as imagens de entrada entre as classes estabelecidas pelo número de neurônios na saída da camada totalmente conectada. Um dos tipos de classificadores mais utilizados atualmente é o *Softmax*, o qual normaliza a pontuação para cada classe da camada anterior em valores entre 0 e 1, no qual a soma totaliza 1. A Equação 2.8 apresenta a função *softmax*, sendo z_j o elemento do vetor de pontuações a ser normalizado:

$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \tag{2.8}$$

O classificador *Softmax* utiliza a função *cross-entropy loss*, que tem a forma:

$$L_i = -log(\frac{e^{f_{yi}}}{\sum_j e^{f_j}}) \tag{2.9}$$

Na qual a média de L_i por todo o processo de treinamento juntamente a um termo de regularização, representa a perda total. Como a entropia cruzada entre duas distribuições de probabilidade, uma "verdadeira"(p) e outra estimada (q) é dada pela Equação 2.10, o classificador *Softmax* tenta minimizar a entropia cruzada entre as probabilidades das classes estimadas (representada pela Equação 2.8) e a distribuição "verdadeira", que nesse caso seria o valor 1 para a classe correta e 0 para as outras (LI; KARPATHY; JOHNSON, 2017b).

$$H(p,q) = -\sum_x p(x) log q(x) \tag{2.10}$$

2.2 TREINAMENTO

O treinamento de uma rede neural é o processo matemático pelo qual a rede irá ajustar e atualizar os parâmetros de forma a minimizar o erro. Esse processo possui duas fases principais: a primeira é chamada *forward propagation* e a segunda *backpropagation*.

Durante a fase de *forward propagation*, as imagens de treinamento passam por todas as camadas descritas anteriormente, gerando o vetor de pontuação na saída da camada totalmente conectada (subseção 2.1.5) que é utilizado na Equação 2.9 para calcular a perda total.

A fase de *backpropagation* utiliza o algoritmo de gradiente descendente, que se mostra como um método eficaz para atualizar os parâmetros da rede (LECUN et al., 1989). O processo

ocorre começando no final da rede e indo em direção ao seu início, com o intuito de minimizar o erro, os pesos são atualizados por uma redução escalar negativa da derivada do erro em relação a estes mesmos pesos.

Um hiperparâmetro muito relevante a esse processo é a taxa de aprendizagem, fator que representa o quanto os pesos serão alterados em cada atualização. Quanto menor a taxa, mais lentamente a rede irá convergir. Por outro lado, quanto maior for a taxa, maior a velocidade de aprendizado, porém existe o risco da rede não convergir devido a uma oscilação do modelo na superfície de erro. Outros hiperparâmetros também importantes para o treinamento são: o *batch size*, que indica o número de amostras de treinamento que serão utilizados para atualizar os parâmetros; e o número de épocas, que representa quantas vezes a rede irá classificar todos os dados de treinamento (LI; KARPATHY; JOHNSON, 2017d).

2.3 DESCRITORES DE TEXTURA LOCAL

Os descritores de textura local descrevem uma metodologia que visa obter propriedades da superfície de um objeto, analisando vizinhanças predefinidas da imagem e codificando cada uma delas com valores denominados *Texture Unit* (TU), computando as relações das intensidades relativas entre os pixels em uma pequena vizinhança (VIEIRA, 2013; SOUZA, 2017). A seguir são apresentados os descritores relevantes a este trabalho.

2.3.1 *Local Binary Pattern* (LBP)

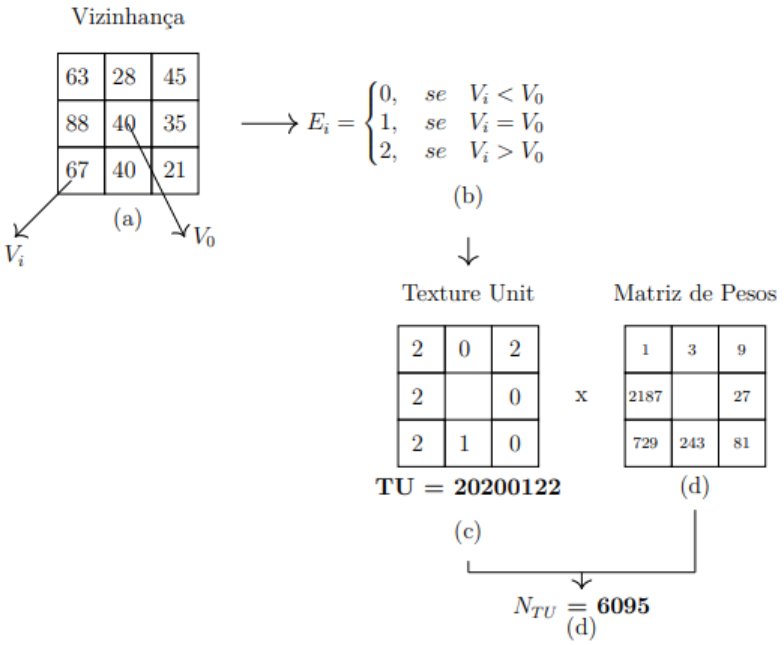
O método denominado *Texture Unit* (TU) foi primeiramente abordado por Wang e He (1990) para analisar texturas locais em imagens digitais. Este modelo é aplicado a uma vizinhança 3x3 da imagem, que representa a menor unidade de textura local, com 8 vizinhos V_i em torno de um pixel central V_0 (Figura 4(a)). Os valores deste micropadrão são divididos em três níveis (0, 1 e 2), dependendo se esse é menor, igual ou maior que o valor do pixel central, respectivamente (Figura 4(b)), podendo assim gerar até 3^8 (6561) códigos.

O código TU (Equação 2.11) é obtido pela somatória do resultado das multiplicações dos valores limiarizados (Figura 4(c)) pelos pesos atribuídos aos pixels correspondentes (Figura 4(d)), que caracteriza um determinado padrão local da imagem (WANG; HE, 1990).

$$N_{TU} = \sum_{8}^{i=1} E_i \cdot 3^{i-1} \tag{2.11}$$

Ojala, Pietikäinen e Harwood (1996), utilizando a metodologia do *Texture Unit* como base, apresentaram o *Local Binary Pattern* (LBP), uma proposta que simplifica os níveis de representação de uma vizinhança 3x3 para valores binários (0 ou 1), possibilitando gerar códigos de 2^8 (256) níveis. Assim como no método TU, o LBP utiliza o pixel central V_0 como parâmetro de limiarização (Figura 5(a)), fazendo com que E_i seja igual a 0 se V_i tiver valor inferior a V_0, e igual 1 se V_i for maior ou igual a V_0 (Figura 5(b)).

Figura 4: Cálculo do *Texture Unit* em vizinhança 3 x 3



Fonte: (VIEIRA, 2013)

O código LBP (Equação 2.12) é então obtido pela somatória do resultado das multiplicações dos valores limiarizados (Figura 5(c)) pelos pesos atribuídos aos pixels correspondentes (Figura 5(d)) (VIEIRA, 2013; SOUZA, 2017).

$$N_{TU} = \sum_{8}^{i=1} E_i \cdot 2^{i-1} \tag{2.12}$$

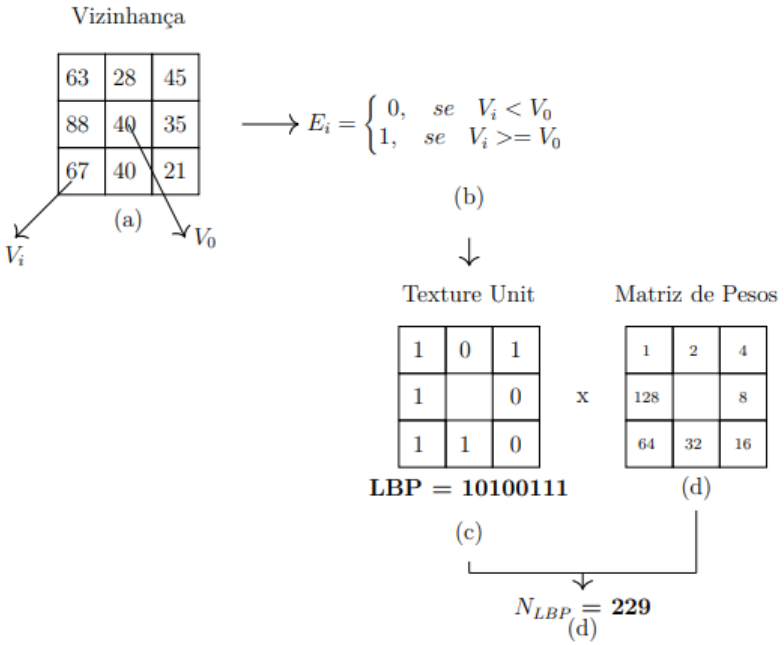
Após o cálculo dos códigos LBP de todos os pixels, um histograma de toda a imagem é gerado para computar a frequência dos diferentes códigos, formando um vetor de características LBP, que caracteriza a imagem de textura (VIEIRA, 2013).

2.3.2 *Local Mapped Pattern* (LMP)

Em 2014, Ferraz, Jr e Gonzaga propuseram a abordagem *Local Mapped Pattern* (LMP). Essa metodologia considera, como um padrão local, a diferença dos níveis de cinza em uma dada vizinhança WxW em relação ao pixel central. A Figura 6 mostra um exemplo de uma vizinhança 3 × 3 e um gráfico 3D do respectivo padrão local.

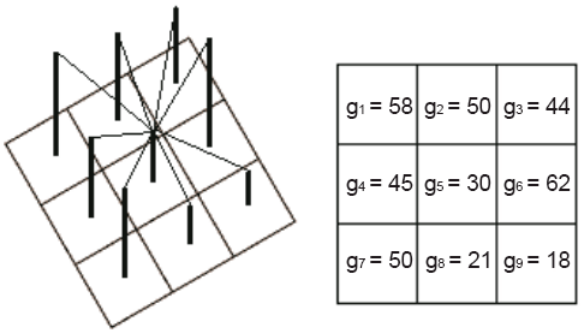
Considerando uma vizinhança como a da Figura 6, seu padrão pode ser mapeado para um bin h_b de acordo com a Equação 2.13. Essa equação representa a soma do peso de cada diferença do nível de cinza entre os pixels da vizinhança e o pixel central, mapeado no intervalo

Figura 5: Cálculo do LBP em vizinhança 3 x 3



Fonte: (VIEIRA, 2013)

Figura 6: Diferenças de níveis de cinza em uma vizinhança 3 x 3



Fonte: Adaptado de (FERRAZ; JR; GONZAGA, 2014)

[0, 1] por uma função de mapeamento e arredondado para B bins possíveis (FERRAZ, 2016; NEGRI, 2017)

$$h_b = round(\frac{\sum_v^{i=1} f(g_i)m_i}{\sum_v^{i=1} m_i}(B-1)) \tag{2.13}$$

O método *Local Binary Pattern* (LBP) pode ser derivado da Equação 2.13 utilizando a função degrau (Equação 2.14) como função de mapeamento.

$$f(g_i) = H[g_i - g_c] = \begin{cases} 1, & \text{se } g_i - g_c \geq 0 \\ 0, & \text{se } g_i - g_c > 0 \end{cases} \tag{2.14}$$

Assim, pode-se observar que o LBP é um caso particular da abordagem LMP.

Para análise de texturas, os autores sugerem uma aproximação suave da função degrau por uma curva logística ou uma curva sigmóide, como apresentado na Equação 2.15, onde β é a inclinação da curva e $[g_i - g_c]$ são as diferenças de níveis de cinza dentro de uma vizinhança centrada em g_c.

$$f(g_i) = \frac{1}{1 + e^{\frac{-[g_i - g_c]}{\beta}}} \tag{2.15}$$

A matriz de pesos proposta é:

$$M = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{2.16}$$

A utilização da função sigmóide pode ser vista como uma aproximação suave, quando comparada com o LBP, da função degrau a fim de captar características que a função degrau não é capaz. A Figura 7 apresenta uma comparação entre os códigos LBP e LMP gerados para a mesma vizinhança de uma amostra de textura, considerando B = 256.

A Função sigmóide tem sua curva com a forma de um "S", e são utilizadas em uma grande variedade de contextos, tais como funções de transferências de redes neurais. Estas curvas não-lineares são consideradas simples, encontrando um equilíbrio entre o comportamento linear e não-linear. A Figura 8 mostra as funções de mapeamento sigmóide e degrau, utilizadas na Equação 2.13 para as abordagens LMP e LBP, respectivamente. (FERRAZ, 2016; NEGRI, 2017)

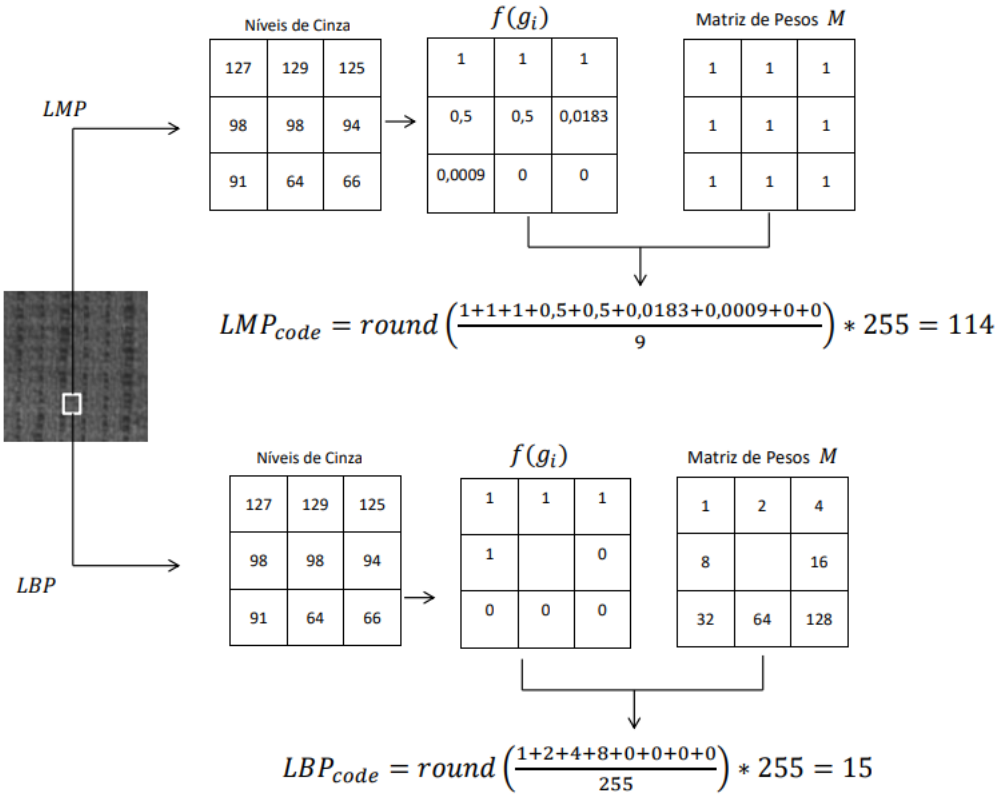
2.4 TRABALHOS RELACIONADOS

Nesta seção são apresentados alguns trabalhos que envolvem a utilização de CNNs para reconhecimento da região periocular e métodos que combinam descritores locais de textura com as redes neurais convolucionais.

2.4.1 Reconhecimento da região periocular

Em 2018, Hernandez-Diaz, Alonso-Fernandez e Bigun propuseram um problema de classificação com imagens da região periocular obtidas com uma câmera digital, utilizando redes

Figura 7: Comparação entre os códigos LBP e LMP



Fonte: (NEGRI; GONZAGA, 2014)

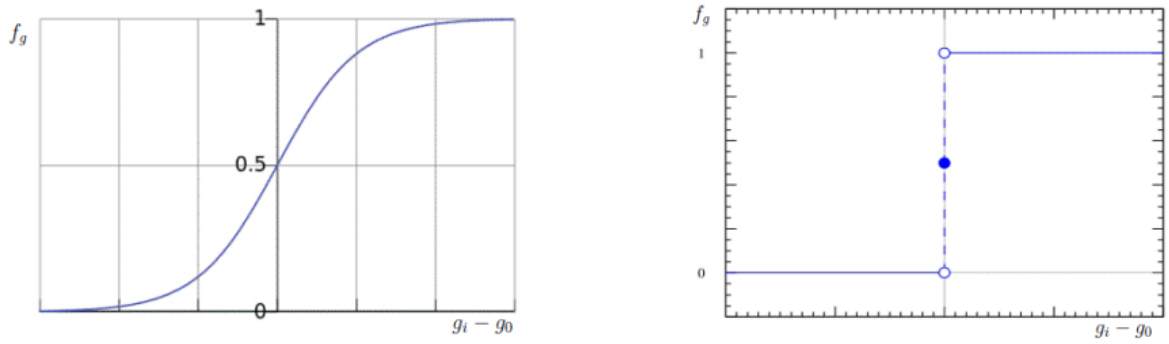
neurais convolucionais pré-treinadas com a base de dados ImageNet, entre elas a AlexNet e a GoogLeNet. O trabalho comparou ainda os resultados com métodos de referência de obtenção de características da região periocular, entre eles o LBP e histogramas de gradientes orientados. Os resultados obtidos mostraram uma redução de até 40% da taxa de erro pelos modelos de CNN utilizados em relação aos outros métodos populares.

Já o trabalho de Zhao e Kumar (2016), propõe uma nova estrutura de redes neurais convolucionais denominada *semantics-assisted convolutional neural networks* (SCNN) para conseguir reconhecimento periocular em condições não-ideais. Essa estrutura incorpora informações semânticas explícitas a fim de conseguir resultados eficientes e acurados das características da região periocular. Os experimentos obtidos mostraram que o método proposto é eficiente e pode ser utilizado não só para região periocular mas também para tarefas gerais de classificação de imagens.

2.4.2 Métodos de classificação por CNN utilizando o descritor LBP

O trabalho de Juefei-Xu, Boddeti e Savvides (2017) propõe uma uma camada de convolução denominada *local binary convolution*, na qual substitui a camada de convolução convencional

Figura 8: Diferença do mapeamento dos níveis de cinza utilizando as funções (a) sigmoidal (b) degrau



(a) Função de mapeamento sigmoidal (LMP)

(b) Função de mapeamento degrau (LBP)

Fonte: (NEGRI, 2017)

em CNNs. Essa camada proposta foi inspirada pelo descritor LBP, e consiste em um número fixo e escasso de filtros convolucionais binários que não são atualizados durante o treinamento, uma função de ativação não linear e um conjunto de pesos lineares. Essa configuração reduz o número de parâmetros quando comparada a uma camada de convolução padrão, e devido à natureza binária e escassa dos pesos, o modelo também é menor quando comparado a camadas de convolução convencionais. O trabalho obteve desempenho com as bases de dados CIFAR-10 e ImageNet semelhante a de CNNs normais, reduzindo significativamente o custo computacional.

Diferentemente do trabalho citado anteriormente, Zhang et al. (2017), realizaram um pré-processamento das imagens de entrada utlizando o descritor LBP, e executaram o treinamento utilizando os mapas de características obtidos. A rede projetada pelos autores conta com a camada de entrada, 2 camadas de convolução, 2 camadas de *pooling*, uma camada totalmente conectada a dois neurônios, que são os vetores de saída. A rede então, apenas identifica se a imagem apresenta ou não uma face humana. Os experimentos indicaram uma acurácia de 95,33% utilizando o descritor LBP, e 91,83% sem a utilização do descritor, demonstrando um ganho de 3,50% quando as imagens são pré-processadas.

3 MATERIAIS E MÉTODOS

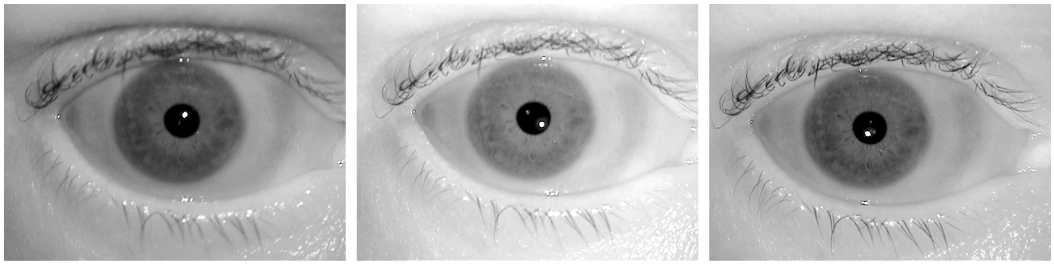
3.1 FERRAMENTAS UTILIZADAS

Para o desenvolvimento deste trabalho foi utilizado o *software* MATLAB R2018a em conjunto com as ferramentas: *Deep Learning Toolbox*, *Deep Learning Toolbox Model for AlexNet Network* e *Global Optimization Toolbox*. Os treinamentos e testes foram realizados em um microcomputador com sistema operacional Ubuntu, processador Intel Core i7-6700 (8 núcleos), 32 Gb de RAM e uma placa de vídeo GeForce Titan XP.

3.2 BASE DE IMAGENS

A base de imagens escolhida para realização dos experimentos propostos nesse trabalho foi parte da base *ND-CrossSensor-Iris-2013 Data Set* (CVRL. . . , ; BOWYER; FLYNN, 2016), que é formada por imagens da região periocular obtidas pelos sistemas LG2200 e LG400. A porção da base utilizada, foi a obtida com o sistema LG2200, que possui um total de 116.564 imagens divididas entre 676 classes únicas, com cada classe correspondendo a um indivíduo diferente. As imagens tem dimensão 640 x 480 pixels e estão na escala de cinza. A Figura 9 mostra um exemplo de imagens encontradas na base.

Figura 9: Imagens adquiridas por uma captura no sistema LG2200



Fonte: *ND-CrossSensor-Iris-2013 Data Set*

O sistema LG2200 utiliza uma iluminação no espectro do infravermelho próximo (*near infra-red*) para a aquisição de imagens. A iluminação é feita por três LEDs infravermelhos, posicionados conforme ilustrado na Figura 10 (indicados por pontos vermelhos), que são ativados em sequência conforme a aquisição é feita. Assim, o sistema obtém três imagens com cada LED iluminando em exatamente uma das imagens, como é possível observar na Figura 9. Originalmente, apenas uma das três imagens era escolhida após passar por um *software* de controle de qualidade da camera, porém com a motivação de ter uma base composta não apenas de imagens "perfeitas", mas também uma que pudesse ser utilizada para pesquisa de métodos

robustos a utilização de imagens com qualidade inferior, as três imagens passaram a ser utilizadas para compor a base.

A base de imagens obtida com o sistema LG2200 foi originalmente proposta para algoritmos de reconhecimento de íris, mas as aquisições próximas ao olho captam também informações da região periocular, motivo de desenvolvimento deste trabalho.

Figura 10: Sistema de aquisição de imagens de íris LG2200

Fonte: (BOWYER; FLYNN, 2016)

3.3 ALEXNET

A rede neural convolucional escolhida para realização dos experimentos, foi a AlexNet desenvolvida por Krizhevsky, Sutskever e Hinton (2012), vencedora do ILSVRC em 2012 e que contribuiu para a popularização da utilização de CNNs para aplicações em visão computacional (LI; KARPATHY; JOHNSON, 2017a).

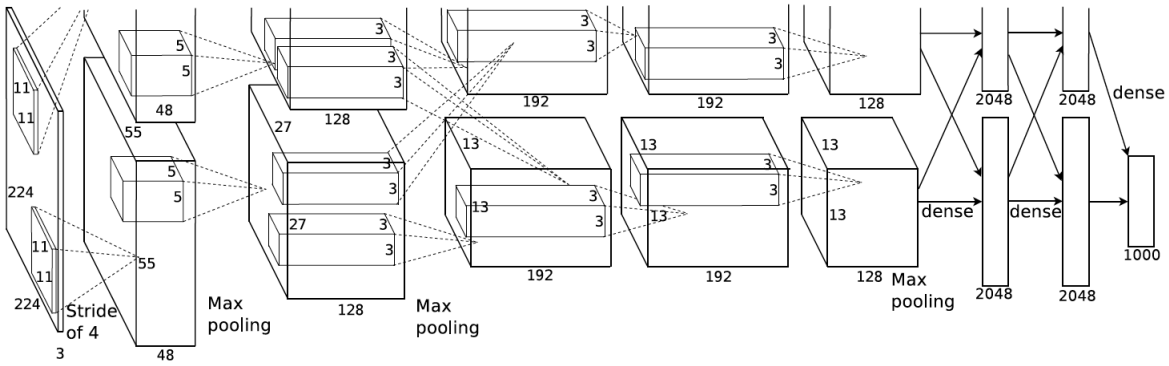
A escolha da rede foi feita após a realização de testes preliminares com a AlexNet e a GoogLeNet (SZEGEDY et al., 2015). Nesses testes, foram feitos treinamentos utilizando as duas redes e as imagens da base citada anteriormente (seção 3.2), com os mesmos hiperparâmetros. Após esses testes, foi verificado que a acurácia obtida era equivalente para as duas redes, porém o tempo de treinamento com a AlexNet era significativamente menor do que com a GoogleNet (cerca de 4 horas a menos). Assim, utilizou-se a AlexNet para realizar os experimentos descritos no trabalho.

3.3.1 Arquitetura

A AlexNet tem uma arquitetura similar à LeNet ((LECUN et al., 1998)), porém maior, mais profunda e com as camadas de convolução agrupada, que funcionam como se fossem 2 camadas de convolução empilhadas (LI; KARPATHY; JOHNSON, 2017a).

A arquitetura da rede AlexNet é composta por 8 camadas com pesos, sendo 5 camadas de convolução e 3 camadas totalmente conectadas. A rede possui além das camadas de entrada e saída, 3 camadas de *pooling*, 7 camadas de ativação com a função ReLU, 2 camadas de normalização, 2 camadas de regularização utilizando a função *dropout* e 1 camada de classificação. A Figura 11 ilustra a estrutura básica da rede e a Tabela 1 detalha a ordem e funções das camadas.

Figura 11: Arquitetura da AlexNet



Fonte: (KRIZHEVSKY; SUTSKEVER; HINTON, 2012)

3.3.2 *Transfer Learning*

A rede AlexNet pode ser treinada do zero (com inicialização dos pesos de forma aleatória), ou pode-se utilizar um modelo pré-treinado com uma base de dados grande como ponto de partida. Esse último, corresponde a um processo chamado *transfer learning*, e foi o método escolhido para executar os treinamentos, devido a baixa quantidade de imagens da base utilizada para realização do treinamento do zero.

O processo de treinamento por *transfer learning* é comumente feito de duas maneiras diferentes: considerando a CNN como um extrator de características fixo ou realizando um ajuste fino na rede, sendo esse último, o método utilizado nos experimentos. O processo de ajuste fino (ou *fine-tuning*) consiste em substituir o classificador no final da rede, adaptando-o para a nova base de dados, e também em ajustar os pesos da rede pré-treinada continuando o *backpropagation*.

Tabela 1: Descrição da arquitetura da AlexNet

Camada	Função	Descrição
1	Entrada	Imagens de dimensões 227x227x3
2	Convolução	96 convoluções 11x11x3 com passo [4 4] e *padding* [0 0 0 0]
3	Ativação	função ReLU
4	Normalização	normalização cruzada com 5 canais por elemento
5	*Pooling*	*max pooling* 3x3 com passo [2 2] e *padding* [0 0 0 0]
6	Convolução agrupada	2 grupos de 128 convoluções 5x5x48 com passo [1 1] e *padding* [2 2 2 2]
7	Ativação	função ReLU
8	Normalização	normalização cruzada com 5 canais por elemento
9	*Pooling*	*max pooling* 3x3 com passo [2 2] e *padding* [0 0 0 0]
10	Convolução	384 convoluções 3x3x256 com passo [1 1] e *padding* [1 1 1 1]
11	Ativação	função ReLU
12	Convolução agrupada	2 grupos de 192 convoluções 3x3x192 com passo [1 1] e *padding* [1 1 1 1]
13	Ativação	função ReLU
14	Convolução agrupada	2 grupos de 128 convoluções 3x3x192 com passo [1 1] e *padding* [1 1 1 1]
15	Ativação	função ReLU
16	*Pooling*	*max pooling* 3x3 com passo [2 2] e *padding* [0 0 0 0]
17	Totalmente conectada	4096 neurônios na saída
18	Ativação	função ReLU
19	Regularização	dropout com probabilidade de 50%
20	Totalmente conectada	4096 neurônios na saída
21	Ativação	função ReLU
22	Regularização	dropout com probabilidade de 50%
23	Totalmente conectada	1000 neurônios na saída
24	Classificação	classificador softmax
25	Saída	crossentropyex with 'tench' and 999 other classes

Fonte: Adaptado de *Deep Learning Toolbox Model for AlexNet Network*

Os trabalhos de Razavian et al. (2014) e Zhou et al. (2014) demonstram um desempenho superior de modelos de CNN treinados por *transfer learning* da base ImageNet (RUSSAKOVSKY et al., 2015) do que outras bases de menor escala. Além disso, o treinamento por *fine-tuning* também tem a vantagem de ser um processo mais rápido, já que os pesos são iniciados com valores que permitem a identificação de certas características genéricas, que são úteis para várias tarefas diferentes (LI; KARPATHY; JOHNSON, 2017e; SHIN et al., 2016).

A ferramenta *Deep Learning Toolbox Model for AlexNet Network* (seção 3.1) consiste de um modelo pré-treinado da rede AlexNet com a base ImageNet (DENG et al., 2009), a qual possui mais de 1 milhão de imagens distribuídas em 1000 classes diferentes. A fim de se realizar o ajuste fino, as três últimas camadas da rede (camada totalmente conectada, classificador softmax e saída da classificação) foram substituídas para adaptar às 676 classes da base de imagens da região periocular.

3.4 CONJUNTOS GERADOS PARA TREINO E TESTE

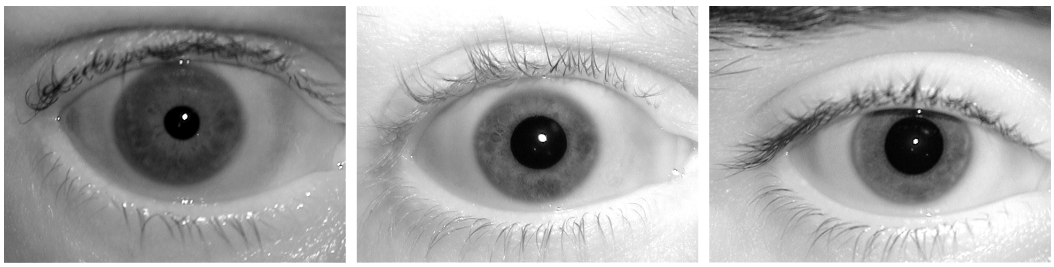
A realização dos experimentos, como descrito previamente, foi feita utilizando o modelo pré-treinado da arquitetura AlexNet que recebe como entrada imagens com dimensões fixas de 227 x 227 x 3 (largura, altura e profundidade, respectivamente).

Para todos os conjuntos, foram realizados os seguintes procedimentos: redimensionamento das imagens para adequação à entrada da rede; pequenas alterações nas imagens para prevenir *overfitting* (espelhamento em torno do eixo vertical e pequenas translações realizados de maneira aleatória) (KRIZHEVSKY; SUTSKEVER; HINTON, 2012); separação aleatória dos conjuntos em: 70% das imagens para treinamento, 15% para validação e 15% para teste, mantendo a proporção entre classes.

3.4.1 Imagens originais

Este conjunto foi formado por imagens originais da base que estão na escala de cinza. A fim de adequar à entrada da rede, foi necessária a conversão para RGB, inserindo imagens iguais em cada um dos três canais. A Figura 12 mostra algumas imagens deste conjunto.

Figura 12: Exemplo das imagens no conjunto original



Fonte: Elaborada pelo autor

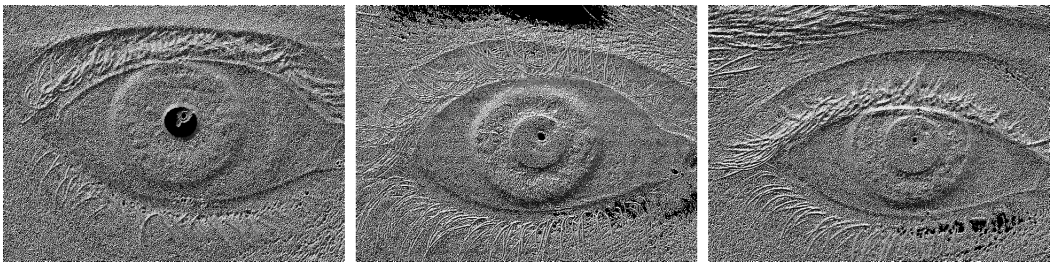
3.4.2 Imagens LBP

Este conjunto de imagens foi gerado aplicando o descritor LBP nas imagens da base, como as imagens geradas continuam na escala de cinza, também foi necessária a conversão para RGB, inserindo imagens iguais em cada um dos três canais. A Figura 13 mostra algumas imagens deste conjunto.

3.4.3 Imagens LMP

Este conjunto de imagens foi gerado aplicando o descritor LMP nas imagens da base, utilizando a matriz de pesos da Equação 2.16 e o número de bins igual a 256. O parâmetro β foi obtido por um processo de otimização com um algoritmo genético, utilizando a ferramenta

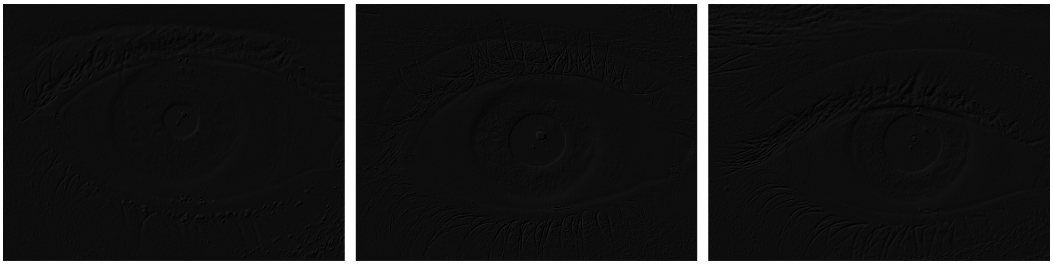
Figura 13: Exemplo de imagens após aplicar o descritor LBP



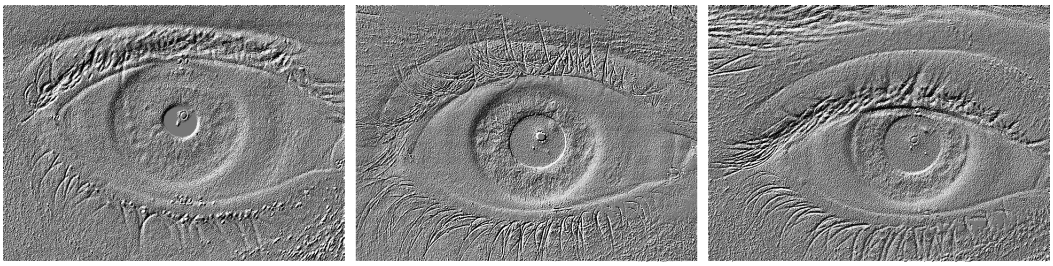
Fonte: Elaborada pelo autor

Global Optimization Toolbox disponível no MATLAB. Assim como no conjunto LBP, as imagens geradas continuam na escala de cinza, sendo necessária a conversão para RGB, inserindo imagens iguais em cada um dos três canais. A Figura 14a mostra algumas imagens deste conjunto.

Figura 14: Exemplo de imagens dos conjuntos LMP e LMP equalizadas



(a) Imagens após aplicar o descritor LMP



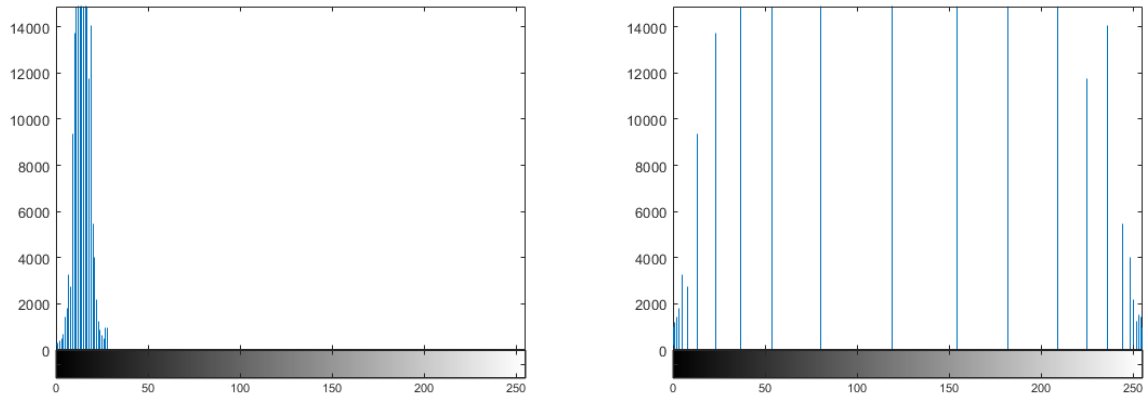
(b) Imagens após equalizar o conjunto LMP

Fonte: Elaborada pelo autor

É possível observar que as imagens do conjunto LMP ficaram com uma visibilidade particularmente ruim, devido a concentração de níveis de cinza em valores próximos a zero, como pode ser verificado no histograma da Figura 15a.

A fim de espaçar os níveis de cinza, foi gerado um outro conjunto equalizando as imagens após a aplicação do descritor LMP (Figura 15a), melhorando a visibilidade. Assim como nos conjuntos anteriores, foi realizada a conversão para RGB, inserindo imagens iguais em cada um

Figura 15: Histogramas das imagens dos conjuntos LMP



(a) histograma das imagens LMP

(b) histograma das imagens LMP equalizadas

Fonte: Elaborada pelo autor

dos três canais. A Figura 14b mostra alguns exemplos do conjunto formado pelas imagens LMP equalizadas.

3.4.4 Imagens combinadas

Este conjunto de imagens foi gerado combinando, em cada um dos três canais, as imagens do conjunto original, LMP e LBP, respectivamente, como mostra o esquema da Figura 16. A combinação foi realizada utilizando um algoritmo no *software* MATLAB, e como foram combinadas imagens diferentes em cada um dos canais, as imagens obtidas são coloridas como pode ser observado na Figura 17.

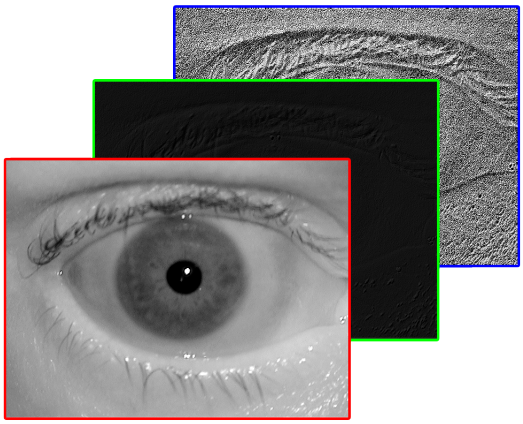
Também foi gerado um conjunto combinando da mesma forma as imagens do conjunto original, LMP equalizado e LBP, respectivamente, obtendo-se as imagens da Figura 18.

3.5 ETAPAS DE TREINAMENTO E TESTE

A fim de se analisar o desempenho e a viabilidade de se utilizar descritores de textura no treinamento de uma CNN para reconhecimento de imagens da região periocular, foram realizados 2 experimentos.

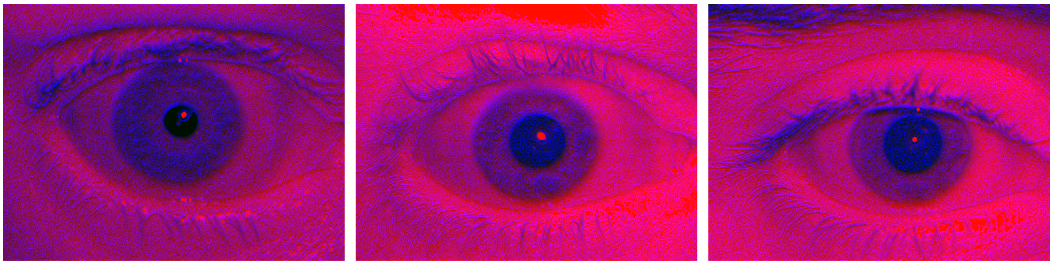
Para todos os treinamentos descritos nesta seção, foi utilizado um *batch size* de 300 amostras e 10 épocas de treinamento. A rede foi treinada através de um gradiente descendente estocástico com taxa de aprendizagem global de 0,001. A nova camada totalmente conectada, descrita na subseção 3.3.2, foi configurada com um fator de taxa de aprendizado de 20 para pesos e *bias*, ou seja, esta camada foi treinada com uma taxa de aprendizado 20 vezes maior do que a taxa global.

Figura 16: Esquema da combinação de imagens original no canal R, LMP no canal G e LBP no canal B



Fonte: Elaborada pelo autor

Figura 17: Exemplo de imagens após combinar escala de cinza, LMP e LBP, em cada uma das camadas



Fonte: Elaborada pelo autor

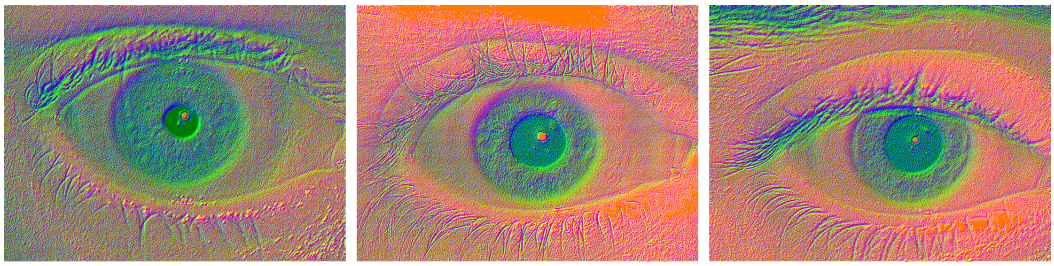
O algoritmo utilizado para realização dos treinamentos e testes pode ser verificado no Apêndice A.

3.5.1 Experimento 1

Neste experimento, foram realizados treinamentos utilizando os seguintes conjuntos de imagens:

- conjunto das imagens originais;
- conjunto LBP;

Figura 18: Exemplo de imagens após combinar escala de cinza, LMP equalizado e LBP, em cada uma das camadas



Fonte: Elaborada pelo autor

- conjunto LMP;
- conjunto das imagens combinadas.

Para a rede treinada com as imagens originais, foram realizados testes com os conjuntos das imagens originais, LBP, LMP e imagens combinadas (Figura 19a). O mesmo procedimento foi realizado para as redes treinadas com os conjuntos LBP (Figura 19b), LMP (Figura 19c) e imagens combinadas (Figura 19d).

Esse experimento foi elaborado a fim de verificar o desempenho da rede em classificar cada um dos conjuntos de teste, quando esta é treinada com os diferentes conjuntos de treinamento. Sendo possível assim, comparar os resultados obtidos utilizando as imagens originais com os obtidos pelos conjuntos pré-processados com os descritores de textura, a fim de investigar a viabilidade da realização de tal pré-processamento.

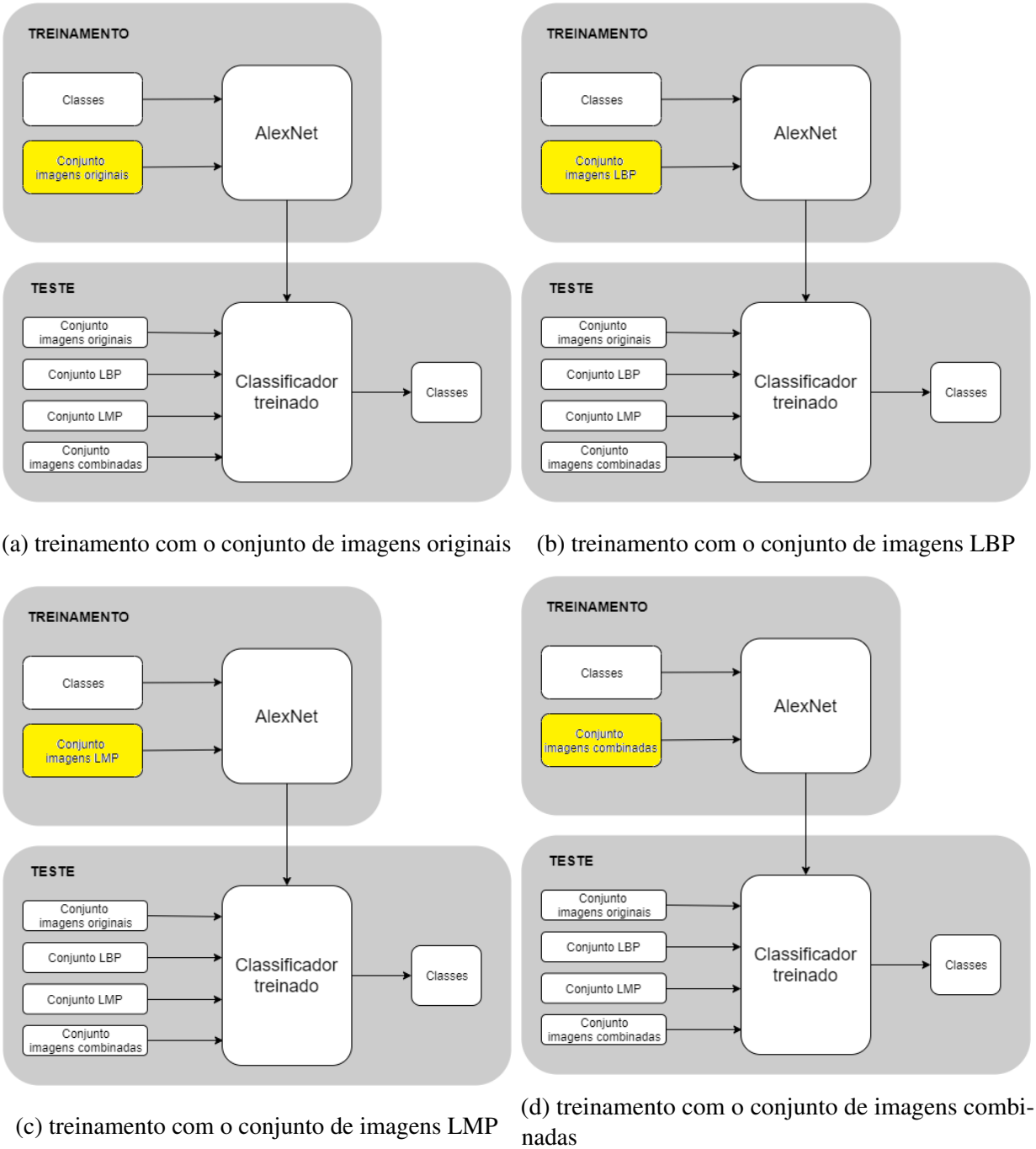
3.5.2 Experimento 2

O experimento 2 consistiu na realização de dois treinamentos adicionais, um utilizando o conjunto de imagens LMP equalizadas e outro com o conjunto de imagens combinando as imagens originais, LMP equalizado e LBP, respectivamente (indicado por "conjunto imagens combinadas EQ"nos diagramas da Figura 20).

Assim como no experimento 1, as redes treinadas foram testadas usando os conjuntos de teste com as imagens originais e LBP, porém substituindo os conjuntos LMP e imagens combinadas por suas respectivas versões equalizadas, como ilustram as Figura 20a e Figura 20b. Adicionalmente, os novos conjuntos de teste equalizados foram utilizados para testar as redes treinadas com os conjuntos de imagens originais e LBP, provenientes do experimento 1 (Figura 20c e Figura 20d).

A motivação para realização desse experimento partiu da baixa visibilidade do conjunto LMP gerado inicialmente (Figura 14). A fim de distribuir os níveis de cinza e melhorar a visibili-

Figura 19: Esquema de treinamentos e testes do experimento 1



(a) treinamento com o conjunto de imagens originais

(b) treinamento com o conjunto de imagens LBP

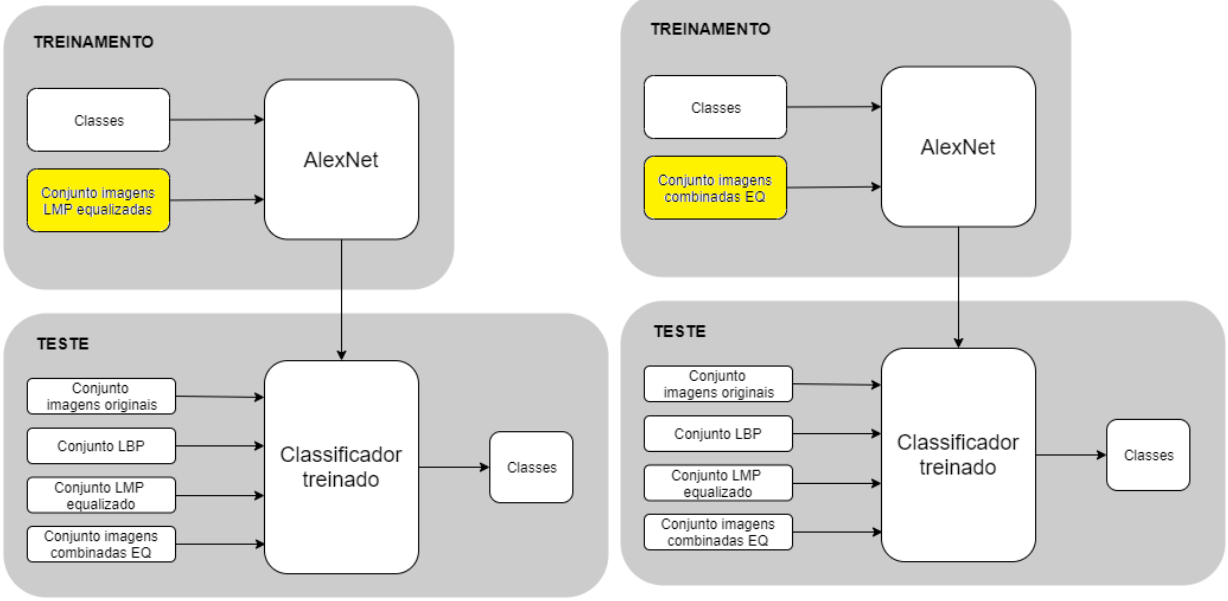
(c) treinamento com o conjunto de imagens LMP

(d) treinamento com o conjunto de imagens combinadas

Fonte: Elaborada pelo autor

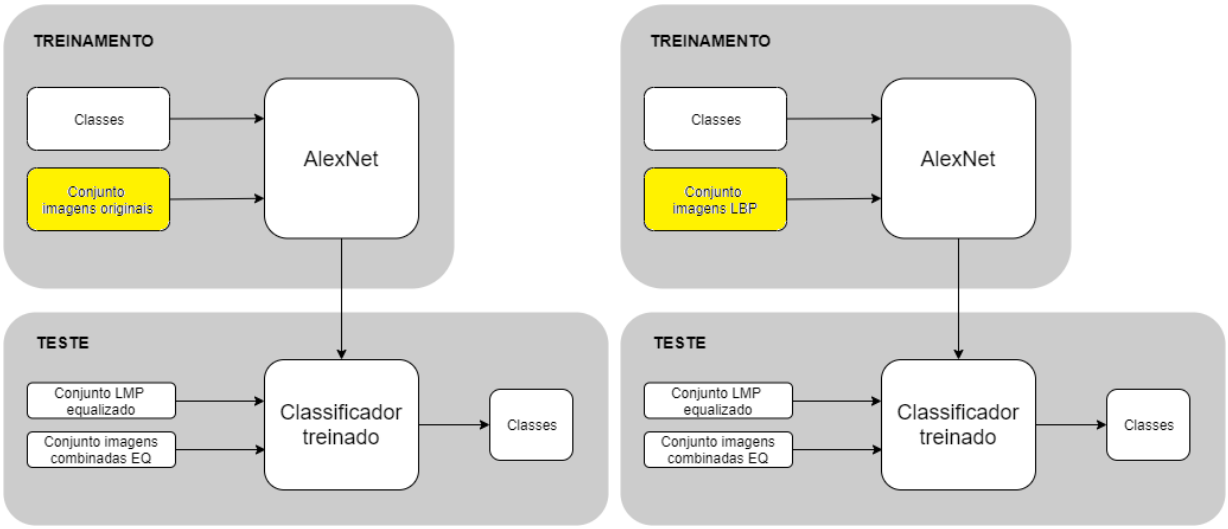
dade da imagem, foi realizada a equalização mencionada na subseção 3.4.3. Esse experimento foi realizado com o intuito de verificar o impacto da equalização das imagens após aplicar o descritor LMP, no desempenho da tarefa de classificação.

Figura 20: Esquema de treinamentos e testes do experimento 2



(a) treinamento com as imagens LMP equalizadas

(b) treinamento com as imagens combinadas equalizadas



(c) treinamento com o conjunto de imagens originais

(d) treinamento com o conjunto de imagens LBP

Fonte: Elaborada pelo autor

4 RESULTADOS

4.1 Resultados do experimento 1

No primeiro experimento, a rede foi treinada com o conjunto de imagens originais e testada com cada um dos conjuntos de teste gerados (Figura 19a). A Tabela 2 exibe os resultados obtidos.

Tabela 2: Acurácia da rede treinada com as imagens do conjunto original

Imagens do conjunto de teste	originais	LBP	LMP	Combinadas
Acurácia (%)	98,96	1,49	0,35	3,64

Fonte: Elaborada pelo autor

Os resultados indicam que após o treinamento da rede com as imagens originais, o único conjunto de teste que o classificador da rede foi capaz de classificar acuradamente com 98,96% de acerto, foi o do próprio conjunto de teste com as imagens originais. A partir desses resultados, pode-se afirmar que ao fazer o ajuste fino da rede com essas imagens, as características aprendidas pela rede no processo, não são "úteis"para a classificação das imagens pré-processadas.

Os treinamentos restantes foram feitos da mesma forma que o primeiro, porém trocando o conjunto de treinamento, assim como ilustrado na Figura 19. Os resultados constam nas Tabelas 3, 4, 5, para os treinamentos realizados com os conjuntos LBP, LMP e imagens combinadas, respectivamente.

Tabela 3: Acurácia da rede treinada com as as imagens do conjunto LBP

Imagens do conjunto de teste	originais	LBP	LMP	Combinadas
Acurácia (%)	4,48	97,12	0,42	8,29

Fonte: Elaborada pelo autor

Tabela 4: Acurácia da rede treinada com as imagens do conjunto LMP

Imagens do conjunto de teste	originais	LBP	LMP	Combinadas
Acurácia (%)	2,08	0,64	98,11	0,64

Fonte: Elaborada pelo autor

Tabela 5: Acurácia da rede treinada com as imagens, combinando em cada uma das três camadas, conjunto original, LMP e LBP, respectivamente

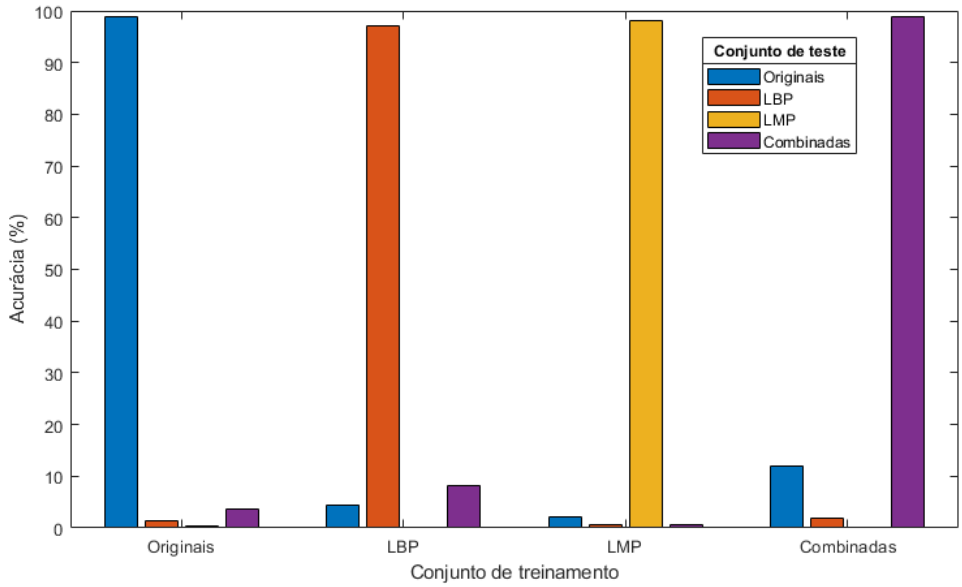
Imagens do conjunto de teste	originais	LBP	LMP	Combinadas
Acurácia (%)	12,08	1,92	0,15	98,94

Fonte: Elaborada pelo autor

Assim como no treinamento com o conjunto de imagens originais, é possível observar que a rede só foi capaz de classificar acuradamente quando o tipo das imagens do conjunto de teste corresponderam ao do conjunto de treinamento. Ou seja, para cada treinamento realizado, utilizando cada um dos conjuntos listados, as características extraídas pela rede não foram discriminantes para todos os outros conjuntos de teste utilizados.

O gráfico abaixo foi elaborado para melhor visualização dos resultados.

Figura 21: Comparação entre as acurácias obtidas por cada um dos treinamentos



Fonte: Elaborada pelo autor

Os resultados obtidos revelam também que não houve melhora na acurácia ao pré-processar as imagens utilizando os descritores LBP e LMP, e nem ao combinar as imagens em cada uma das três camadas, quando comparados a rede treinada e testada com o conjunto original de imagens. É possível observar ainda, que a melhor acurácia obtida foi com o conjunto de treinamento e teste de imagens originais.

4.2 Resultados do experimento 2

O experimento 2 foi realizado de maneira semelhante ao experimento 1, porém utilizando os conjuntos de imagens LMP equalizadas, a fim de verificar se a separação dos níveis de cinza na imagem pré-processada influenciaria no resultado da tarefa de classificação. Um novo conjunto de imagens combinadas também foi gerado utilizando as imagens LMP equalizadas, referido como "imagens combinadas equalizadas"nas tabelas abaixo.

As tabelas 6, 7, contêm os valores de acurácia obtidos para a rede treinada com o conjunto de imagens originais (Figura 20a) e LBP (Figura 20b) e testadas com os novos conjuntos, respectivamente.

Tabela 6: Acurácia da rede treinada com as imagens do conjunto original

Imagens do conjunto de teste	LMP equalizadas	Combinadas equalizadas
Acurácia (%)	2,11	7,16

Fonte: Elaborada pelo autor

Tabela 7: Acurácia da rede treinada com as as imagens do conjunto LBP

Imagens do conjunto de teste	LMP equalizadas	Combinadas equalizadas
Acurácia (%)	11,16	4,74

Fonte: Elaborada pelo autor

Assim como no experimento anterior, a utilização de conjuntos de testes diferentes dos de treinamento não gerou resultados que possam ser considerados satisfatórios para as redes treinadas com o conjunto original e LBP. Isso mostra que mesmo equalizando as imagens LMP, as características extraídas pela rede nesses treinamentos ainda assim não são discriminantes para as imagens do conjunto de teste.

Em seguida a rede foi treinada utilizando o conjunto LMP equalizado (Figura 20c) e o conjunto combinando em cada uma das três camadas, conjunto original, LMP equalizado e LBP, respectivamente (Figura 20d). As tabelas 8, 9 exibem os valores de acurácia obtidos.

Tabela 8: Acurácia da rede treinada com as imagens do conjunto LMP equalizadas

Imagens do conjunto de teste	originais	LBP	LMP equalizadas	Combinadas equalizadas
Acurácia (%)	6,14	8,01	98,23	7,84

Fonte: Elaborada pelo autor

Tabela 9: Acurácia da rede treinada com as imagens, combinando em cada uma das três camadas, conjunto original, LMP equalizado e LBP, respectivamente

Imagens do conjunto de teste	originais	LBP	LMP equalizadas	Combinadas equalizadas
Acurácia (%)	9,13	2,16	5,75	98,91

Fonte: Elaborada pelo autor

Os resultados indicam que houve um pequeno ganho na acurácia, quando a rede é treinada e testada com o conjunto LMP equalizado, em comparação à rede treinada e testada com o conjunto LMP sem equalização do experimento 1. Porém, assim como os resultados obtidos no primeiro experimento, só se obtiveram acurácias acima de 90% para os conjuntos de imagem de teste que correspondem ao tipo de conjunto do treinamento.

Os valores de acurácia obtidos no experimento 2, indicam ainda que a equalização das imagens LMP não influenciam significativamente na tarefa de classificação proposta. Isso pode ser verificado comparando os valores das tabelas 4, 8 e 5, 9, respectivamente, onde é possível observar uma variação negligenciável (menor que 0,5%) das acurácias para os mesmos conjuntos de treinamento e teste.

5 CONCLUSÃO

Este trabalho realizou uma análise da viabilidade de se pré-processar as imagens de entrada de uma rede neural convolucional para o reconhecimento de imagens da região periocular, utilizando os descritores de textura local LBP e LMP, visto que o trabalho de Zhang et al. (2017) conclui que a identificação de faces humanas tem acurácia melhorada quando essas imagens são pré-processadas com o descritor LBP. Foi avaliado o desempenho da rede AlexNet pré-treinada utilizando o método de ajuste fino, com diferentes tipos de conjuntos de treinamento e teste.

Os valores de acurácia significativos (acima de 90%) foram obtidos apenas quando a rede foi treinada e testada com os mesmo conjuntos de imagens, levando a conclusão que as características extraídas pela rede com os diferentes conjuntos de treinamento, não foram discriminativos para os diferentes conjuntos de teste utilizados. Como exemplo, a rede treinada com o conjunto de treinamento de imagens originais, só obteve uma acurácia acima de 90% para o conjunto de teste contendo imagens originais. A Tabela 10 sintetiza os valores de acurácias obtidas.

Tabela 10: Acurácia da rede treinada e testada com os conjuntos de mesmo tipo de imagem

Imagens do conjunto de treinamento e teste	**Acurácia (%)**
originais	98,96
LBP	97,12
LMP	98,11
combinadas	98,94
LMP equalizadas	98,23
combinadas equalizadas	98,91

Fonte: Elaborada pelo autor

Os resultados obtidos revelam ainda que o maior valor de acurácia foi alcançado utilizando o conjunto de imagens originais, e que todas as formas de pré-processamento usando os descritores locais de textura forneceram resultados levemente inferiores. Isso ocorre, pois quando os descritores são aplicados sobre as imagens originais, apesar de definirem características da textura da imagem, também causam perdas de informações relacionados aos diferentes níveis de cinza que existem na imagem original, influenciando de maneira negativa na tarefa de classificação da rede. Diferentemente dos resultados obtidos no trabalho de reconhecimento facial de Zhang et al. (2017), a utilização dos mapas de características como entrada da CNN, não impactou de forma positiva nos resultados.

Assim, apesar do desempenho geral da rede ter sido positivo para os diferentes tipos de conjuntos utilizados (com acurácias acima de 97%), levando em conta o tempo e custo

computacional para a aplicação dos descritores de textura e os valores levemente inferiores de acurácia obtidos pelo método de classificação proposto, é possível concluir que a realização de um pré-processamento das imagens da região periocular utilizando os descritores de textura local descritos neste trabalho, não é viável.

REFERÊNCIAS

ALONSO-FERNANDEZ, F.; BIGUN, J. A survey on periocular biometrics research. **Pattern Recognition Letters**, Elsevier, v. 82, p. 92–105, 2016.

ANIEMEKA, I. A friendly introduction to convolutional neural networks. In: . [S.l.]: Hashrocket Blog, 2017. Disponível em: <https://hashrocket.com/blog/posts/a-friendly-introduction-to-convolutional-neural-networks#introduction>. Accesso em: 15-10-2019.

BHANU, B.; KUMAR, A. **Deep learning for biometrics**. [S.l.]: Springer, 2017.

BOWYER, K. W.; FLYNN, P. J. The nd-iris-0405 iris image dataset. **arXiv preprint arXiv:1606.04853**, 2016.

CVRL: Datasets. [S.l.]: University of Notre Dame Computer Vision Research Lab. Disponível em: <https://cvrl.nd.edu/projects/data/#nd-crosssensor-iris-2012-data-set> Acesso em: 20-10-2019.

DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: IEEE. **2009 IEEE conference on computer vision and pattern recognition**. [S.l.], 2009. p. 248–255.

FERRAZ, C. T. **Novos descritores de textura para localização e identificação de objetos em imagens usando Bag-of-Features**. 2016. Tese (Doutorado) — Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2016. Doi: 10.11606/T.18.2016.tde-28092016-141219. Acesso em: 15-10-2019.

FERRAZ, C. T.; JR, O. P.; GONZAGA, A. Feature description based on center-symmetric local mapped patterns. In: ACM. **Proceedings of the 29th Annual ACM Symposium on Applied Computing**. [S.l.], 2014. p. 39–44.

HERNANDEZ-DIAZ, K.; ALONSO-FERNANDEZ, F.; BIGUN, J. Periocular recognition using cnn features off-the-shelf. In: IEEE. **2018 International Conference of the Biometrics Special Interest Group (BIOSIG)**. [S.l.], 2018. p. 1–5.

JILLELA, R.; ROSS, A. Mitigating effects of plastic surgery: Fusing face and ocular biometrics. In: IEEE. **2012 IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS)**. [S.l.], 2012. p. 402–411.

JUEFEI-XU, F.; BODDETI, V. N.; SAVVIDES, M. Local binary convolutional neural networks. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2017. p. 19–28.

JUEFEI-XU, F. et al. Investigating age invariant face recognition based on periocular biometrics. In: IEEE. **2011 International Joint Conference on Biometrics (IJCB)**. [S.l.], 2011. p. 1–7.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2012. p. 1097–1105.

LECUN, Y. et al. Backpropagation applied to handwritten zip code recognition. **Neural computation**, MIT Press, v. 1, n. 4, p. 541–551, 1989.

____. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, Taipei, Taiwan, v. 86, n. 11, p. 2278–2324, 1998.

LI, F.-F.; KARPATHY, A.; JOHNSON, J. Convolutional neural networks: Architectures, convolution / pooling layers. In: **Convolutional Neural Networks for Visual Recognition Course Notes**. [S.l.]: Stanford University, 2017. Disponível em: <http://cs231n.github.io/convolutional-networks/>. Accesso em: 15-10-2019.

____. Linear classification: Support vector machine, softmax. In: **Convolutional Neural Networks for Visual Recognition Course Notes**. [S.l.]: Stanford University, 2017. Disponível em: <http://cs231n.github.io/linear-classify/>. Accesso em: 15-10-2019.

____. Neural networks part 1: Setting up the architecture. In: **Convolutional Neural Networks for Visual Recognition Course Notes**. [S.l.]: Stanford University, 2017. Disponível em: <http://cs231n.github.io/neural-networks-1/>. Accesso em: 15-10-2019.

____. Optimization: Stochastic gradient descent. In: **Convolutional Neural Networks for Visual Recognition Course Notes**. [S.l.]: Stanford University, 2017. Disponível em: <http://cs231n.github.io/optimization-1/>. Accesso em: 15-10-2019.

____. Transfer learning and fine-tuning convolutional neural networks. In: **Convolutional Neural Networks for Visual Recognition Course Notes**. [S.l.]: Stanford University, 2017. Disponível em: <http://cs231n.github.io/transfer-learning/>. Accesso em: 15-10-2019.

LI, H. et al. A convolutional neural network cascade for face detection. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 5325–5334.

NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: **Proceedings of the 27th international conference on machine learning (ICML-10)**. [S.l.: s.n.], 2010. p. 807–814.

NEGRI, T. T. **Descritores locais de textura para classificação de imagens coloridas sob variação de iluminação**. 2017. Tese (Doutorado) — Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2017. Doi: 10.11606/T.18.2018.tde-02032018-112555. Acesso em: 15-10-2019.

NEGRI, T. T.; GONZAGA, A. Color texture classification under varying illumination. **Worshop de visao computacional**, p. 61–66, 2014.

NGUYEN, K. et al. Iris recognition with off-the-shelf cnn features: A deep learning perspective. **IEEE Access**, IEEE, v. 6, p. 18848–18855, 2017.

OJALA, T.; PIETIKÄINEN, M.; HARWOOD, D. A comparative study of texture measures with classification based on featured distributions. **Pattern recognition**, Elsevier, v. 29, n. 1, p. 51–59, 1996.

OJALA, T.; PIETIKÄINEN, M.; MÄENPÄÄ, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. **IEEE Transactions on Pattern Analysis & Machine Intelligence**, IEEE, n. 7, p. 971–987, 2002.

PARKHI, O. M. et al. Deep face recognition. In: **bmvc**. [S.l.: s.n.], 2015. v. 1, n. 3, p. 6.

RAZAVIAN, A. S. et al. Cnn features off-the-shelf: an astounding baseline for recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition workshops**. [S.l.: s.n.], 2014. p. 806–813.

RUSSAKOVSKY, O. et al. Imagenet large scale visual recognition challenge. **International journal of computer vision**, Springer, v. 115, n. 3, p. 211–252, 2015.

SCHERER, D.; MÜLLER, A.; BEHNKE, S. Evaluation of pooling operations in convolutional architectures for object recognition. In: SPRINGER. **International conference on artificial neural networks**. [S.l.], 2010. p. 92–101.

SHIN, H.-C. et al. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. **IEEE transactions on medical imaging**, IEEE, v. 35, n. 5, p. 1285–1298, 2016.

SOUZA, J. M. de. **Reconhecimento de textura de íris sob variação do tamanho da pupila**. 2017. Tese (Doutorado) — Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2017. Doi: 10.11606/T.18.2017.tde-30062017-091537. Acesso em: 15-10-2019.

SRIVASTAVA, N. et al. Dropout: a simple way to prevent neural networks from overfitting. **The journal of machine learning research**, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014.

SZEGEDY, C. et al. Going deeper with convolutions. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 1–9.

VIEIRA, R. T. **Análise de micropadrões em imagens digitais baseada em números fuzzy**. 2013. Dissertação (Mestrado) — Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2013. Doi:10.11606/D.18.2013.tde-29042013-154729. Acesso em: 15-10-2019.

WANG, L.; HE, D.-C. Texture classification using texture spectrum. **Pattern Recognition**, Elsevier, v. 23, n. 8, p. 905–910, 1990.

ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. In: SPRINGER. **European conference on computer vision**. [S.l.], 2014. p. 818–833.

ZHANG, H. et al. A face recognition method based on lbp feature for cnn. In: IEEE. **2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)**. [S.l.], 2017. p. 544–547.

ZHAO, Z.; KUMAR, A. Accurate periocular recognition under less constrained environment using semantics-assisted convolutional neural network. **IEEE Transactions on Information Forensics and Security**, IEEE, v. 12, n. 5, p. 1017–1030, 2016.

ZHENG, L. et al. Good practice in cnn feature transfer. **arXiv preprint arXiv:1604.00133**, 2016.

ZHOU, B. et al. Learning deep features for scene recognition using places database. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2014. p. 487–495.

APÊNDICE A – ALGORITMO PARA TREINAMENTO E TESTE DA ALEXNET

```
%Cria um datastore com a base de imagens, sendo as subpastas as
    classes
imds = imageDatastore('LG2200_Comb_2', ...
    'IncludeSubfolders',true, ...
    'LabelSource','foldernames');

%Divide a base em 70% Treinamento, 15% Validacao e 15% Teste
[imdsTrain, imdsMedium] = splitEachLabel(imds,0.7,'randomized');
[imdsValidation, imdsTest] = splitEachLabel(imdsMedium,0.5,'
    randomized');
numTrainImages = numel(imdsTrain.Labels);

%Carrega a AlexNet
net = alexnet;

inputSize = net.Layers(1).InputSize

%Extrai a rede excluindo as ultimas tres camadas
layersTransfer = net.Layers(1:end-3);

%Adiciona 3 camadas no final da rede, para adaptar a rede a nova
    quantidade de classes
numClasses = numel(categories(imdsTrain.Labels))

layers = [
    layersTransfer
    fullyConnectedLayer(numClasses, ...
    'WeightLearnRateFactor',20,...
    'BiasLearnRateFactor',20)
    softmaxLayer
    classificationLayer];

%Adequa o tamanho das imagens da base e para as imagens de
    treinamento, espelha as imagens em torno do eixo vertical e
    translada por 30pixels  horizontalmente e verticalmente, de
    maneira randomica.
pixelRange = [-30 30];
```

```
imageAugmenter = imageDataAugmenter( ...
    'RandXReflection',true, ...
    'RandXTranslation',pixelRange, ...
    'RandYTranslation',pixelRange);

augimdsTrain = augmentedImageDatastore( ...
inputSize(1:2), imdsTrain,'DataAugmentation',imageAugmenter,'
   ColorPreprocessing', 'gray2rgb');
augimdsValidation = augmentedImageDatastore(...
inputSize(1:2),imdsValidation,'ColorPreprocessing', 'gray2rgb');
augimdsTest = augmentedImageDatastore(...
inputSize(1:2),imdsTest,'ColorPreprocessing', 'gray2rgb');

%'ColorPreprocessing', 'gray2rgb'

%Definindo os paramentros de treinamento
options = trainingOptions('sgdm', ...
    'MiniBatchSize',300, ...
    'MaxEpochs',10, ...
    'InitialLearnRate',1e-3, ...
    'Shuffle','every-epoch', ...
    'ValidationData',augimdsValidation, ...
    'ValidationFrequency',20, ...
    'ValidationPatience', Inf, ...
    'Verbose',true, ...
    'VerboseFrequency', 500, ...
    'Plots','training-progress');

%Treinando a rede
netTransfer = trainNetwork(augimdsTrain,layers,options);

%Classificando as imagens de teste utilizando a rede treinada
[YPred,scores] = classify(netTransfer,augimdsTest);

%Calcula a acuracia da rede em relacao as imagens de teste
YValidation = imdsTest.Labels;
accuracy = mean(YPred == YValidation)
```