Reconhecimento facial utilizando transferência de aprendizado e redes neurais

Julio Cesar Ribeiro Silva
Universidade Tecnológica Federal do Paraná - UTFPR
Medianeira, Brasil
juliocrsilva11@gmail.com

Pedro Luiz de Paula Filho
Universidade Tecnológica Federal do Paraná - UTFPR
Medianeira, Brasil
plpf2004@gmail.com

***Abstract*—In recent decades, the field of machine learning has advanced rapidly due to increased research efforts and the emergence of new neural network architectures. This document aims to present a model that utilizes neural networks for facial recognition to provide individuals with access to systems. To achieve this, publicly available datasets will be combined with a dataset containing images of new individuals who need to be grouped with the existing datasets and identified individually.**

Initially, two deep learning neural networks, VGG and ResNet, will be employed for training. Using these two neural networks, transfer learning will be applied to perform additional training on the dataset adapted for the problem.

***Keywords*—deep learning; machine learning; model training.**

***Resumo*—Nas últimas décadas a área de aprendizado de máquina tem avançado rapidamente devido aumento nas pesquisas e no surgimento de novas arquiteturas de redes neurais. Esse documento tem como objetivo apresentar um modelo que utiliza redes neurais para reconhecimento facial para fornecer acesso de indivíduos à sistemas. Para isso serão utilizadas bases públicas misturadas à um conjunto de dados que contenha imagens de novos indivíduos que devem ser agrupados aos outros *datasets* e identificados individualmente. Inicialmente serão utilizadas duas redes neurais que utilizam *Deep Learning* para realizar o treinamento VGG e ResNet. Utilizando essas duas redes neurais, será utilizado transferência de aprendizado para realizar um novo treinamento utilizando o banco de imagens adaptadas para o problema.**

***Palavras-chave*—aprendizado profundo; aprendizado de máquina; treinamento de modelos.**

I. Introdução

O avanço em pesquisas na área de reconhecimento facial fomentou a criação de um leque de aplicações que utilizam essa ferramenta como solução em diversas situações por exemplo sistemas de segurança. Com o aumento na demanda por ferramentas de segurança da informação a quantidade de pesquisas em áreas como controle de acesso, biometria e verificação de identidades também cresceu significativamente. Em termos de controle de acesso, a biometria possui características únicas que são distintas entre os indivíduos, por esse motivo, vários sistemas que utilizam padrões de biometria foram criados nas últimas décadas, por exemplo: reconhecimento por características da íris, reconhecimento facial, reconhecimento por fala, identificação através da palma da mão, identificação por padrão de escrita, entre outros padrões biométricos [1].

O reconhecimento facial é uma tarefa muito complicada que exige uma taxa de acerto muito elevada, principalmente em sistemas de controle de acesso. Esse processo utiliza como ferramenta técnicas de aprendizado de máquina em conjunto com redes neurais artificiais para realizar a tarefa de reconhecimento. Após a detecção de faces e a extração de características, os dados são enviados ao modelo pré treinado para que ocorra a comparação com as faces pré cadastradas [2].

Atualmente uma das ferramentas mais eficazes para reconhecimento facial utiliza aprendizado de máquina e aplica técnicas de aprendizado profundo (*deep learning*). Redes neurais convolucionais utilizam múltiplas camadas em cascata para extrair características e realizar transformações nos dados. Como resultado essas redes conseguem aprender múltiplos níveis de abstração em imagens e conseguem manter a consistência nas classificações independente da posição da face, da expressão e da iluminação na cena. Modelos tradicionais não conseguem se adaptar muito bem as mesmas condições [2].

II. Metodologia

Foram utilizados diferentes modelos para tentar realizar a tarefa de classificação. Cada modelo possui características diferentes que podem facilitar a classificação dos exemplos e tornar a tarefa de treinamento mais rápida. Foram utilizados os modelos VGG e ResNet.

A. VGG

A rede neural VGG(*Visual Geometry Group*) é um dos modelos clássicos e é amplamente utilizado em diversas pesquisas relacionadas a visão computacional. Essa rede é bastante utilizada devido a sua simplicidade aliada às suas altas taxas de acerto. O desenvolvimento da rede foi realizado pelo grupo de

visão geométrica da universidade de *Oxford* em conjunto com pesquisadores do *Google Deepmind*. A rede utiliza filtros de 3×3 nas camadas de convolução[1] e filtros 2×2 na camada de pooling[2]. Foram desenvolvidos dois modelos, um com 16 camadas(VGG16) e outro modelo com 19 camadas(VGG19). Para essa pesquisa foi utilizado o modelo com 19 camadas [3].

B. ResNet

Rede Residual (ResNet) é um modelo clássico amplamente utilizados em tarefas relacionadas à visão computacional e processamento de imagens. Essa arquitetura foi a vencedora da competição *ImageNet*[3] em 2015. O avanço fundamental da rede ResNet é que ela proporciona o treinamento de uma rede extremamente profunda, com 150 camadas escondidas por exemplo. Até o desenvolvimento da ResNet, conforme a rede ganha profundidade, o treinamento se torna cada vez mais complicado e a performance do treinamento decresce rapidamente [5]. Porém esse perda não ocorre por conta de *overfitting* e quanto mais camadas são adicionadas ao modelo, maior é o erro no treinamento.

Em [6] tem-se que as redes neurais seguem a mesma topologia simples, porém eficaz da rede VGG que utiliza diversos blocos de construção que realizam operações e tem o mesmo tamanho. Essa ideia implica na redução na escolha de hiperparâmetros e expõe a profundidade de uma rede neural como característica essencial. Além disso, essa regra simples também reduz o risco de *overfitting*, ou seja, reduz a possibilidade do modelo se adaptar para o conjunto de dados que está sendo utilizado para a sua construção.

O principal diferencial das redes residuais em comparação à outras arquiteturas mais clássicas de redes profundas está no bloco residual.

III. Conjunto de Dados

Para o desenvolvimento do *dataset* personalizado, foi utilizado uma combinação de imagens com as imagens existentes em conjuntos de dados que são amplamente utilizados dentro da comunidade de aprendizado de máquina. Para esse estudo foram selecionados os *datasets Milborrow / University of Cape Town* (MUCT) e a *Labeled Faces in the Wild* (LFW).

A. Labeled Faces in the Wild (LFW)

A LFW é um banco de imagens público utilizado para realizar reconhecimento facial e verificação de faces. Esse *dataset* contém mais de treze mil imagens de faces coletadas da internet em que cada exemplo tem a identificação do nome da pessoa que está na imagem, além disso estão representados nesse conjunto de dados mais de cinco mil pessoas diferentes. Dentre essa quantidade de exemplos, mais de mil seiscentos e oitenta pessoas dentro do *dataset* que possuem mais de duas imagens [7].

Cada exemplo dentro desse conjunto de dados possui dimensões 250×250 e possuem três canais de cores no padrão *Red, Blue, Green* (RGB). A Figura 1 apresenta um exemplo das imagens contidas no *dataset*.

Dentro da LFW existe uma diferença na quantidade de imagens para cada indivíduo. Para que seja possível determinar um padrão para o estudo, serão utilizados os indivíduos que possuam mais que 20 imagens dentro do seu diretório.

Fig. 1. Exemplo de imagens do conjunto LFW

B. Milborrow University of Cape Town (MUCT)

O conjunto de dados MUCT é um conjunto de dados público com aproximadamente três mil imagens de faces humanas com diversas condições de iluminação, idade e diferentes etnias. Os conjuntos de dados estão organizados por diretórios, tendo cada diretório todas as imagens do indivíduo. A Figura 2 apresenta um conjunto de faces que estão contidos dentro do *dataset* [8]

As imagens contidas dentro do conjunto de dados foram criadas dentro de um estúdio e por isso possuem iluminação e posição controlada para cada exemplo dentro do *dataset*. Foram posicionadas cinco câmeras em diferentes posições com o objetivo de capturar as faces em posições diferentes. Além

[1]Camada que realiza operações na imagem de entrada da rede extraindo valores dos pixels para auxiliar no processo de extração de características.

[2]Camada utilizada para reduzir a dimensão da saída da rede neural utilizando operações de agrupamento com o objetivo de preparar os dados para a classificação reduzindo a complexidade.

[3]Conjunto de diversas imagens de diferentes categorias organizadas de acordo com a hierarquia *WorldNet* [4]

Fig. 2. Exemplo de imagens do *dataset* MUCT

da posição das câmeras, também foram utilizadas diferentes iluminações na criação do conjunto de dados.

As imagens possuem dimensões de 480×640 e estão organizados em 278 diretórios em que cada diretório contém entre 10 a 15 imagens de cada exemplo. Cada exemplo possui três canais de cores utilizando o padrão RGB.

IV. Treinamento

Para realizar o treinamento, as imagens foram redimensionadas para serem inseridas na rede. Foram utilizadas imagens com dimensões 300×300. Além da dimensão, foi utilizado um parâmetro chamado *batch size* que determina a quantidade de imagens que serão utilizadas pelo modelo para realizar o treinamento, ou seja, no treinamento o conjunto de dados é dividido de acordo com o valor do *batch size* e cada uma dessas partes é enviada para treinamento separadamente para ser realizado o ajuste dos pesos dos neurônios. A quantidade de épocas de treinamento foi sendo ajustada até o valor padrão de 40 épocas para o treinamento de todas as redes utilizadas no trabalho.

Foi utilizado transferência de aprendizado para realizar as tarefas desse trabalho e para isso houve o congelamento dos pesos das camadas originais de cada rede para que o aprendizado ocorresse apenas nas camadas adicionadas para realizar as novas classificações. Além do congelamento dos pesos das camadas originais, também foi utilizada uma camada de *dropout* com 30% de neurônios removidos com o objetivo de reduzir a possibilidade de *overfitting*. Essa camada está localizada logo após as camadas originais e está ligada diretamente à camada totalmente conectada.

Também com o objetivo de melhorar os resultados de treinamento, foi utilizada uma configuração de *callback* durante a etapa de validação do modelo para que caso a rede mantenha a mesma taxa de precisão por mais de três épocas, o treinamento deveria ser interrompido. Esse comportamento é determinado *Early Stopping* e está disponível na biblioteca keras [9]. Junto ao *callback* para interromper o treinamento, foi adicionado um novo para salvar a precisão do modelo no conjunto de testes e no conjunto de validação a cada época do treinamento.

V. Resultados

O treinamento foi realizado inicialmente utilizando o conjunto de dados LFW com o objetivo de refinar os parâmetros da rede neural e em seguida foi aplicado o mesmo conjunto de parâmetros da rede no conjunto de dados MUCT. Os resultados serão apresentados separadamente apresentado os gráficos gerados durante o treinamento e o gráfico gerado durante a etapa de testes. Além disso também será apresentado o mapa de calor gerado pela rede durante a classificação das duas classes que foram inseridas no conjunto de testes.

A. LFW VGG

A rede neural VGG levou aproximadamente 26 horas para realizar o treinamento e obteve uma taxa de acerto de 21,57% e um valor de *loss* de 11,86. Para o subconjunto de validação a rede obteve uma taxa de acerto de 15,33% e um valor de *loss* de 12,1. Já para o conjunto de teste a rede neural obteve uma precisão de 16,60% e um valor de *loss* de 11,8284.

A Figura 3 apresenta os gráficos gerados pela ferramenta Tensorboard [10] durante o treinamento da rede. Em (a) tem-se o gráfico gerado com os valores de precisão obtidos pelo modelo para os subconjuntos de treinamento e validação a cada época. É possível verificar que com um tempo maior de treinamento a rede VGG poderia obter um resultado um pouco melhor porque tanto a linha que representa validação quanto a linha que representa o conjunto de treinamento não haviam iniciado a estabilização dos resultados como foi verificado em outras redes. Em (b) tem-se representado o gráfico gerado com os valores de *loss* obtidos a cada época do treinamento.

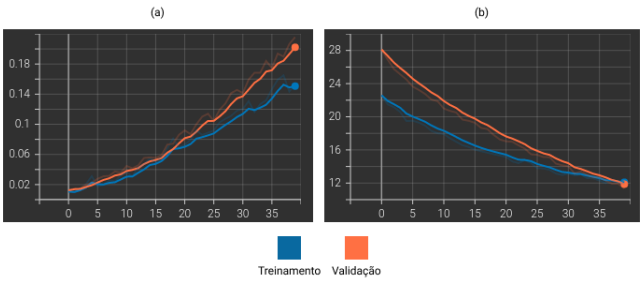

Fig. 3. Gráficos gerados pelo Tensorboard durante o treinamento da rede VGG. Em (a) gráfico de precisão do modelo por época de treinamento e (b) gráfico que representa o valor *loss* por época de treinamento.

B. LFW ResNet

Com uma taxa de acerto no conjunto de teste de 55,57% e com um valor de *loss* de 1,7073%, a rede ResNet obteve

o melhor resultado dentre as redes neurais testadas para o conjunto LFW. O treinamento levou aproximadamente 8 horas e a rede obteve precisão de 91,14% e um valor de *loss* de 0,5871 no conjunto de treinamento. Já no conjunto de validação, a rede neural obteve uma taxa de acerto de 55,96% e um valor de *loss* de 1,7860.

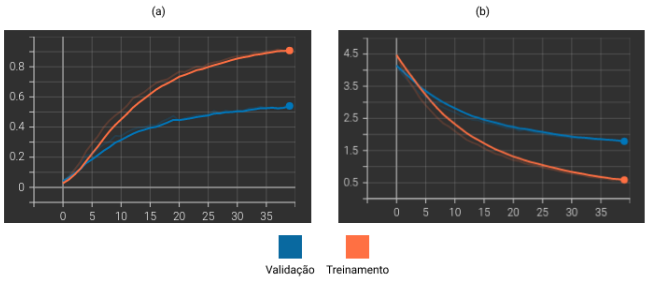

Fig. 4. Gráficos gerados pelo Tensorboard para o conjunto de treinamento e validação durante o treinamento da ResNet. Em (a) gráfico de precisão do modelo por época de treinamento e (b) gráfico que representa o valor *loss* por época de treinamento.

A Tabela I apresenta um resumo dos valores obtidos para precisão durante o treinamento dos modelos utilizando o conjunto de dados LFW.

	Treinamento	Validação	Teste
VGG	21,57%	15,33%	16,60%
ResNet	91,14%	55,96%	55,57%

TABELA I
PRECISÃO DOS MODELOS TREINADOS COM O CONJUNTO LFW

C. MUCT VGG

A rede neural VGG levou aproximadamente 61 horas para realizar o treinamento e obteve uma taxa de acerto de 54,92% e um valor de *loss* de 3,454. No conjunto de validação a rede obteve uma taxa de acerto de 63,33% e um valor de *loss* de 2,274. Por fim, no conjunto de testes a rede neural obteve uma precisão 61,80% e um valor de *loss* de 2,4572. Diferente da rede VGG treinada utilizando o conjunto de dados LFW, a mesma rede utilizando o conjunto MUCT obteve uma precisão maior nos conjuntos de validação e teste do que no conjunto de treinamento. O tempo de treinamento também foi mais elevado para esse conjunto de dados o que pode estar atrelado à quantidade de exemplos, porém não de forma proporcional.

A Figura 5 apresenta os gráficos gerados com os dados do treinamento da rede VGG. Em (a) tem-se o gráfico gerados com os valores de precisão obtidos nos conjuntos de treinamento e de validação. Em (b) está representado o gráfico gerado com os valores de *loss* obtidos no treinamento. Mesmo a rede neural VGG sendo uma rede clássica, ela ainda conseguiu obter uma precisão maior do que um dos modelos que serão apresentados na sequência. Porém, o tempo que essa rede demorou para realizar o treinamento foi o maior dentre os modelos.

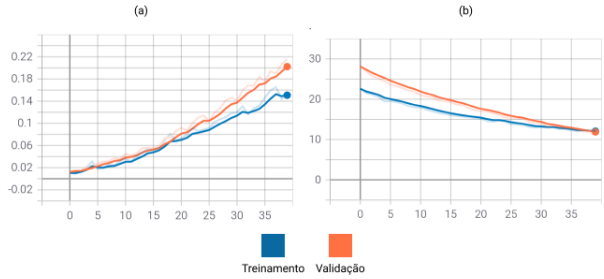

Fig. 5. Gráficos gerados pelo Tensorboard durante o treinamento da rede VGG sobre o conjunto MUCT. Em (a) gráfico de precisão do modelo por época de treinamento e (b) gráfico que representa o valor *loss* por época de treinamento.

D. MUCT ResNet

A rede ResNet foi a rede que obteve a maior precisão entre as redes neurais testadas nesse trabalho para o conjunto MUCT. Além disso, o tempo necessário para realizar o processo de treinamento foi de aproximadamente 19 horas e obteve uma taxa de acerto de 99,91% com o valor de *loss* em 0,040. Para o subconjunto de validação ela obteve uma taxa de acerto de 98,99% e um valor de *loss* de 0,1085. No conjunto de testes a ResNet conseguiu atingir uma precisão de 99,17% junto a um valor de *loss* de 0,1020. Diferente das redes anteriores, a ResNet obteve um acerto maior no conjunto de treinamento do que nos conjuntos de validação e testes o que é esperado ao se treinar uma rede neural.

A Figura 6 apresenta os gráficos gerados utilizando os valores de precisão e os valores de *loss*. Em (a) está localizado o gráfico gerado com as taxas de acerto obtidas para o conjunto de treinamento (azul) e para o conjunto de validação (laranja). Em (b) tem-se o gráfico que foi gerado com os valores de *loss* para os mesmos conjuntos de dados.

A Tabela II apresenta um resumo dos valores obtidos para precisão durante o treinamento dos modelos utilizando o conjunto de dados MUCT.

	Treinamento	Validação	Teste
VGG	54,92%	63,33%	61,80%
ResNet	99,91%	98,99%	99,17%

TABELA II
PRECISÃO DOS MODELOS TREINADOS COM O CONJUNTO MUCT

VI. CONCLUSÃO

Por fim, foram realizados treinamentos das duas redes neurais (ResNet e VGG) em cada um dos conjuntos de dados ap-

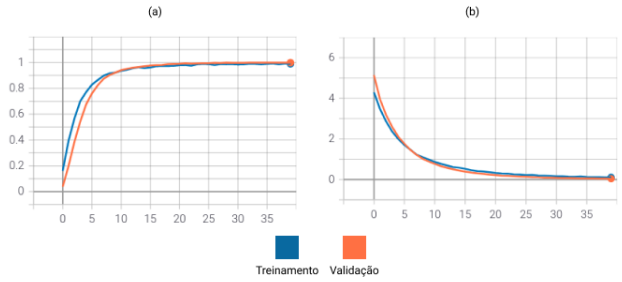

Fig. 6. Gráficos gerados pelo Tensorboard durante o treinamento da rede ResNet sobre o conjunto MUCT. Em (a) gráfico de precisão do modelo por época de treinamento e (b) gráfico que representa o valor *loss* por época de treinamento.

resentados. A rede neural VGG obteve resultado menor, porém ainda é possível realizar um novo treinamento estendendo o número de épocas com o objetivo de melhorar os resultados obtidos para os dois *datasets*, isso porque a rede não apresentou um inicio de estabilização no gráfico de precisão e além disso obteve valores de precisão similares tanto para o conjunto de treinamento quanto para o de validação.

A rede ResNet para o conjunto LFW obteve valores mais altos de precisão tanto no conjunto de treinamento quanto no conjunto de validação quando comparada a rede anterior. Entretanto o valor da precisão para o conjunto de treinamento difere muito do valor obtido no conjunto de validação e esse comportamento pode caracterizar *overfitting*. Portanto essa rede obteve resultados maiores de precisão, porém esses valores não são estáveis.

Para o conjunto de dados MUCT a rede ResNet obteve resultados relativamente melhores e sem a grande diferença entre os conjuntos de treinamento e validação atingindo uma precisão de 99.17% no conjunto de teste.

Referências

[1] K. Okokpujie, E. Noma-Osaghae, S. John, K.-A. Grace, and I. Okokpujie, "A face recognition attendance system with gsm notification," in *2017 IEEE 3rd International Conference on Electro-Technology for National Development (NIGERCON)*, 2017, pp. 239–244.

[2] M. Wang and W. Deng, "Deep face recognition: A survey," *CoRR*, vol. abs/1804.06655, 2018. [Online]. Available: http://arxiv.org/abs/1804.06655

[3] W. Tan, P. Liu, X. Li, Y. Liu, Q. Zhou, C. Chen, Z. Gong, X. Yin, and Y. Zhang, "Classification of COVID-19 pneumonia from chest CT images based on reconstructed super-resolution images and VGG neural network," *Health Inf. Sci. Syst.*, vol. 9, no. 1, p. 10, 2021. [Online]. Available: https://doi.org/10.1007/s13755-021-00140-0

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[5] P. Dwivedi, "Understanding and coding a resnet in keras," Mar 2019. [Online]. Available: https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33

[6] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," 2017.

[7] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[8] S. Milborrow, J. Morkel, and F. Nicolls, "The MUCT Landmarked Face Database," *Pattern Recognition Association of South Africa*, 2010, http://www.milbo.org/muct.

[9] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[10] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/