Using Deep Learning to Recognize People by Face and Voice

Lucas Gomes da Silva
TeleMídia - PUC-Rio
lucas@telemidia.puc-rio.br

Alan L. V. Guedes
TeleMídia - PUC-Rio
alan@telemidia.puc-rio.br

Sergio Colcher
Informatics Departament - PUC-Rio
colcher@inf.puc-rio.br

ABSTRACT

There are many ways to build a person identification system and those systems can be used for authentication and security. The latest phones for example, bring fingerprint readers to enhance the user experience. From our perspective on Neural Networks we determine that Machine Learning is enough to guarantee someone's identity without the need of any specific sensors other than a camera and a microphone. It is achievable with pictures of their face, sounds of their voice and Deep Learning. This work presents a study to build an application to allow biometric authentication using only multimedia and Deep Learning.

KEYWORDS

convolutional neural network, authentication, YOLO, FaceNet, VGGNet

1 INTRODUÇÃO

A segurança de sistemas é essencial. Uma forma de segurança é por autenticação pessoal. E existem diferentes abordagens para autenticação. Dentre elas temos com o uso de sensores específicos para autenticação por atributos físicos individuais. Por exemplo, citamos celulares e laptops equipados com sensores de impressão digital, infravermelho e de distância. Exemplos desses sensores são ilustrados na figura 1 a seguir.

O objetivo deste trabalho é realizar a autenticação pessoal sem utilizar esses sensores específicos citados acima. Ou seja, utilizamos imagem e voz de um pessoas capturados por equipamentos comuns de câmeras e microfones. Como cenário de uso dessa autenticação, citamos o contexto de vigilância e segurança. Por exemplo, um porteiro eletrônico que automatiza o processo de reconhecer pessoas e autorizar a passagem.

Os métodos baseados em *Machine Learning* se tornaram o Estado da Arte em vários segmentos relacionados à análise automática de vídeo. Em particular, citamos o uso de técnicas de *Deep Learning* (DL) para o reconhecimento de padrões audiovisuais. Dentre essas técnicas, o principal método usado é o de *Convolutional Neural Networks* (CNNs), ou ConvNets. Para detecção de objetos em tempo real, por exemplo, a CNN YOLO (You Only Look Once) [10] é o estado da arte. Citamos a seguir as técnicas de CNN utilizadas para nossa autenticação por imagem e voz.

Para identificar pessoas por uma imagem de seu rosto, utilizamos CNN Facenet[12]. Ele é considerada o estado da arte nesta tarefa. Ela gera representações numéricas da face, chamadas *embeddings* . Dessa maneira, o problema de reconhecer uma pessoa por sua imagem pode ser simplificado a distância euclidiana entre *embeddings*

In: XVI Workshop de Trabalhos de Iniciação Científica (WTIC 2019), Rio de Janeiro, Brasil. Anais Estendidos do Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia). Porto Alegre: Sociedade Brasileira de Computação, 2019.
ISSN 2596-1683

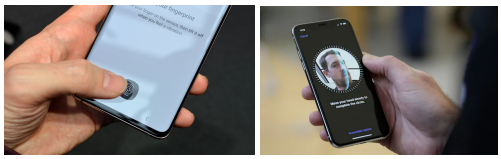

Figure 1: Uso de sensores de impressão digital e mapeamento facial em smartphones

ou um algoritmo de kNN (*k-Nearest Neighbors*) que classifica de acordo com os vizinhos mais próximos.

Para identificar uma pessoa por sua voz, utilizamos a CNN recente vencedora do *challenge* de classificação de áudio DCASE 2018 [11]. Ela é inspirada na CNN VGGNet. Nosso experimento tem objetivo de identificar pessoas que falam língua portuguesa. Logo, criamos uma modelo dessa CNN treinada com amostras dois tipos de pessoas falando português: pessoas consideradas não cadastradas; e pessoas consideradas cadastradas. Dessa maneira, reconhecer uma pessoa pela voz consiste em classificar uma amostra de voz como uma das pessoas cadastradas.

O fluxo de nosso experimento, no momento de identificar uma pessoa, está ilustrado na figura 2. No momento que uma face é detectada pela CNN YOLO, então verificamos se aquele rosto está entre os cadastrados utilizando a CNN Facenet. Em caso positivo, a pessoa detectada também deve falar um frase pré-cadastrada. Esse áudio é analisado utilizando a CNN do *challenge* DCASE 2018 para verificar se é da mesma identidade. Se os resultados condizem o sistema libera o indivíduo a passar. Caso contrário o sistema pode manter a tentativa num registro de atividades suspeitas.

Figure 2: Imagem e som classificados separados

O restante desse artigo está organizado como segue. A seção 2 apresenta nosso método de classificação de face. Já a seção 3 detalha nosso método de classificação de voz e, em especial, o seu dataset e experimentação. Por fim seção 4 apresenta nossas considerações finais e trabalhos futuros.

2 CLASSIFICAÇÃO DE FACE

Este trabalho utiliza as CNNs YOLO e a Facenet pré-treinadas. Cada CNN foi treinada com seu *dataset*, ou seja, usou o conjunto de dados apropriado e devidamente classificado. A primeira foi treinada para

encontrar a face em um vídeo e a segunda para identificar uma face cadastrada.

A CNN YOLO foi treinado com o dataset Wider Face [14]. Esse dataset é composto por 32,203 imagens e 393,703 faces marcadas nelas. Com isso, o modelo para detectar objetos é um especificamente treinado para encontrar rostos. Quanto ao reconhecimento facial, a CNN Facenet foi treinada com o dataset VGGFace2 [1]. Esse dataset consiste em aproximadamente 3.3M faces e 9000 classes.

Com relação à acurácia, o modelo do YOLOv3 treinado em seu artigo alcança um COCO mAP (*mean Average Precision*, uma forma de medir acurácia de detectores de objeto) de 51.5 com um tempo de inferência de 22 milissegundos. Já o Facenet pré-treinado tem 0.9965 de acurácia segundo medida com o *Labeled Faces in the Wild* (LFW) [5].

3 CLASSIFICAÇÃO DE VOZ

Para detalharmos nossa classificação, discutimos a seguir: *dataset* utilizado (subseção 3.1); nosso processo de *feature extraction* (subseção 3.1); arquitetura da CNN (subseção 3.1); e a experimentação (subseção 3.1).

3.1 Dataset

Para classificação de voz, propomos treinar um modelo para a CNN do DCASE 2018 com amostras de voz em português. Nosso *dataset* foi construído com dois tipos de pessoas falando: pessoas consideradas não cadastradas; e pessoas consideradas cadastradas.

Para pessoas não cadastradas, criamos um *dataset* formado com amostras disponibilizadas pelo Grupo Fala Brasil (Grupo de Pesquisa em Processamento de Fala e Linguagem Natural da UFPA) [1]. Mais precisamente, utilizamos conjuntos de amostras LaPS Mail e LaPS Benchmark com diferentes pessoas. O primeiro possui 25 pessoas com 86 sentenças por pessoa (cada uma reproduzindo as 86 mesmas falas) e outro com 35 pessoas com 20 amostras por indivíduo (com frases diferentes).

Para pessoas cadastradas, criamos uma *dataset* com três pessoas com pelo menos 30 amostras de voz por individuo (cada uma repetindo sua frase que representa a senha). Essas pessoas foram membros do laboratório de pesquisa onde este trabalho foi realizado.

3.2 Feature extraction

Para realizar a classificação, uma CNN precisa realizar um processo de obter dados informativos da entrada chamado *feature extraction*. A *feature extraction* utilizada pela CNN do DCASE 2018 utiliza como entrada um do sinal de áudio no formato de *mel-spectrogram* [4]. O processo de extrair o formato *mel-spectrogram* de um sinal de audio é ilustrado na figura 3 a seguir.

O processo de extração do *mel-spectrogram* envolve dividir o sinal em pequenos frames, com overlap entre frames consecutivos. Em seguida, aplicamos uma janela de Hamming em cada frame. Uma exemplo dessa janela é ilustrado na figura 4. Os dados resultantes da janela devem passar por uma FFT (*Fast Fourier Transform*). Com esses frames, vamos computar um periodograma. Nele aplicamos filtros triangulares numa escala mel (figura 5). E depois computamos

[1]Disponível em https://gitlab.com/users/falabrasil/groups

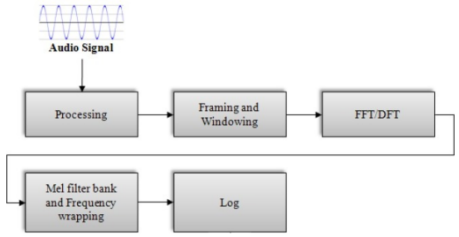

Figure 3: Ilustração do processo que chega no espectrograma mel [7]

o log em cada frequência mel. Com isso temos os espectrogramas mel [4].

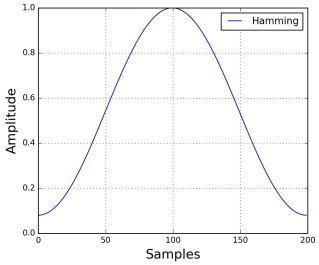

Figure 4: Janela de Hamming

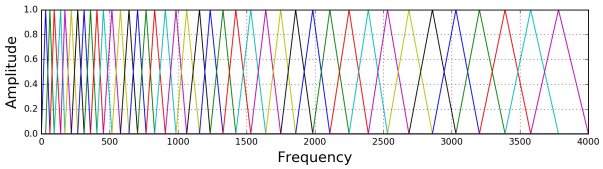

Figure 5: Filter Bank on Mel Scale

Essa extração de características é mais comum em trabalhos recentes de classificadores de áudio que usam Redes Neurais que não precisam de tanta compressão e conseguem processar mais dados com mais informação[13]

3.3 Arquitetura da CNN

Dado que o problema é classificação, a implementação é baseada em uma solução proposta no challenge DCASE 2018 (figura 6) para classificar áudio. De acordo com o autor, ela é inspirada no VGGNet.

A rede foi montada e compilada no Keras [2]. A saída é um vetor de probabilidades de pertencer a cada classe. Podemos usar um *argmax* e então obtemos a predição da classe.

Das amostras do *dataset*, 1/3 são separadas para teste (mantendo proporção das classes e 2/3 são para treino. Dessas do treino, 1/3 ficam para validação. A função de loss é a *categorical crossentropy* (já que nosso objetivo final é ter N > 2 classes: no momento da classificação, se cadastrado, o modelo especifica o falante) e a do

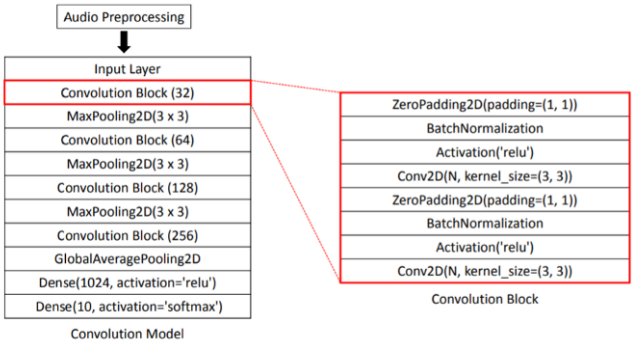

Figure 6: A rede proposta - São quatro blocos de convolução (depois de cada um fazemos um pooling) , neles se faz o processo de *Batch Normalization* [6], que ajuda a evitar *Overfitting*, que é o caso do modelo só aprender a reproduzir o treino.

otimizador é SGD (*Stochastic Gradient Descent*)usando Nesterov momentum, com learning rate=0.01, decay=0.0001 e momentum=0.9 .

Para essa CNN, os espectrogramas em escala mel são as entradas. A biblioteca librosa [9] é usada para gerá-los a partir dos sinais de áudios.

3.4 Experimentação

Nos testes iniciais algumas etapas da arquitetura proposta para o challenge ainda ficaram ausentes, como *Data Augmentation* (métodos para expandir o *dataset*) e uso de Stacked Networks.

Começamos os testes com parte do *dataset*. Primeiramente com duas classes: uma juntando diversas pessoas, os "não cadastrados", e a segunda com amostras de uma pessoa é o "primeiro cadastrado". Nessa montagem a primeira classe tem 142 arquivos e a segunda 42. A classe grande representa a diversidade das propriedades de fala e uma classificação de algum áudio que não pertence a uma das classes não cadastradas deve se aproximar dela.

Como o *dataset* é pequeno, os resultados do treinamento variam. Testamos mudando o otimizador para *Adam* e os resultados pareciam equivalentes, então o SGD foi mantido. Dentre eles, um dos modelos gerados teve a matriz de confusão do teste dada pela Tabela 1. A Tabela 2 indica acurácia desse mesmo modelo.

	Não-Cadastrado	Cadastrado
Não-Cadastrado	50	0
Cadastrado	2	12

Table 1: Matriz de confusão do test split: nas colunas ficam as predições,nas linhas ficam os esperados. A diagonal são os acertos

Em seguida aumentamos a quantidade de classes "cadastrados". Adicionamos dois, cada um com 30 arquivos. No total ficaram quatro classes, um "não cadastrado" e três "cadastrados". Também aumentamos o número de amostras de áudio não cadastrados, que ficou com 2247.

	Precision	Recall	f1-score	support
Não-Cadastrado	0.96	1.00	0.98	50
Cadastrado	1.00	0.86	0.92	14
accuracy			0.97	64
macro avg	0.98	0.93	0.95	64
weighted avg	0.97	0.97	0.97	64

Table 2: Classification Report

A matriz de confusão do teste desse modelo é dado pela tabela 3 e as estatísticas do modelo é dado pela tabela 4

	0	1	2	Não-Cadastrado
0	11	0	0	3
1	0	6	0	4
2	0	0	9	1
Não-Cadastrado	0	0	0	758

Table 3: Matriz de confusão do test split com o *dataset* maior. Interessante que o erro converge para o não-cadastrado, o erro menos indesejável visando a segurança

	Precision	Recall	f1-score	support
0	1.00	0.79	0.88	14
1	1.00	0.60	0.75	10
2	1.00	0.90	0.95	10
Não-Cadastrado	0.99	1.00	0.99	758
accuracy			0.99	792
macro avg	1.00	0.82	0.89	792
weighted avg	0.99	0.99	0.99	792

Table 4: Classification Report com mais classes

4 CONSIDERAÇÕES FINAIS

Este trabalho realizou a autenticação de pessoa sem utilizar sensores dedicados. Ou seja, utilizamos imagem e voz de um pessoas capturados por equipamentos comuns de câmeras e microfones. Para isso utilizamos: a CNN YOLO para encontrar face; a CNN facenet para classificar; e a CNN do DCASE para classificar voz. Ao final realizamos uma avaliação da proposta com os usuários cadastrados e não cadastrados e tivemos resultados satifatórios.

Este trabalho é uma pesquisa em andamento. Consideramos que o trabalho tem limitações com relação ao dataset e a arquitetura de classificação de voz utilizada

Com relação ao *dataset*, como trabalho futuro também podemos buscar os resultados de técnicas de aumentar o tamanho e a diversidade do *dataset* a partir do que já temos, ou seja, fazer *data augmentation*. Essa é uma técnica para criar novas amostras, por exemplo, adicionando ruídos nas amostras iniciais. Ou ainda verificar os resultados de classificação sem se limitar ao idioma falado. Algo interessante para todos os *datasets*, o de vozes e o de rostos, seria analisar a diversidade deles de forma sistemática [8]. A falta de diversidade no treino pode ser um motivo de falha no momento da classificação.

Com relação à rede para identificar faces, uma mais recente é a ArcFace [3] que promete resultados melhores na geração dos embeddings. Com relação a arquitetura de classificação de voz, consideramos limitações na maneira como a rede foi projetada, em que para cadastrar uma nova pessoa o processo de treino deve ser todo refeito. Outro entrave foi com o *dataset* especificamente em português, que acabou possuindo um tamanho reduzido. Como próximos passos podemos buscar outras arquiteturas para classificar som, averiguar uma mais específica para voz. O ideal seria alguma que se aproxime à solução que temos da identificação facial, com um modelo treinado para que a saída seja uma representação da entrada e simplifique a classificação (o sistema pode treinar para criar embeddings, com isso para adicionar mais classes ao classificador não precisa treinar um novo modelo [15]). Deste modo então, poderíamos comparar os novos resultados com os vistos anteriormente.

REFERENCES

[1] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. 2018. VGGFace2: A dataset for recognising faces across pose and age. In *International Conference on Automatic Face and Gesture Recognition*.

[2] François Chollet et al. 2015. Keras. https://keras.io

[3] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. 2019. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[4] Haytham Fayek. 2016. Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between. https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html

[5] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. 2007. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments.* Technical Report 07-49. University of Massachusetts, Amherst.

[6] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167 [cs]* (Feb. 2015). http://arxiv.org/abs/1502.03167 arXiv: 1502.03167.

[7] Gaganpreet Kaur, Deepika Bharti Singh, and Gagandeep. 2015. - 9359 (Volume-4 , Issue-5) A Survey on Speech Recognition Algorithms.

[8] Ludmila I. Kuncheva and Christopher J. Whitaker. 2003. Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy. *Machine Learning* 51, 2 (01 May 2003), 181–207. https://doi.org/10.1023/A:1022859003006

[9] Brian McFee, Vincent Lostanlen, Matt McVicar, Alexandros Metsai, Stefan Balke, Carl Thomé, Colin Raffel, Dana Lee, Kyungyun Lee, Oriol Nieto, Jack Mason, Frank Zalkow, Dan Ellis, Eric Battenberg, , Ryuichi Yamamoto, Rachel Bittner, Keunwoo Choi, Josh Moore, Ziyao Wei, nullmightybofo, Pius Friesch, Fabian-Robert Stöter, Darío Hereñú, Thassilo, Taewoon Kim, Matt Vollrath, Adam Weiss, CJ Carr, and ajweiss dd. 2019. librosa/librosa: 0.7.0. https://doi.org/10.5281/zenodo.3270922

[10] Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An Incremental Improvement. *CoRR* abs/1804.02767 (2018). arXiv:1804.02767 http://arxiv.org/abs/1804.02767

[11] Yuma Sakashita and Masaki Aono. 2018. *Acoustic Scene Classification by Ensemble of Spectrograms Based on Adaptive Temporal Divisions.* Technical Report. DCASE2018 Challenge.

[12] F. Schroff, D. Kalenichenko, and J. Philbin. [n. d.]. FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[13] Amirsina Torfi, Nasser M. Nasrabadi, and Jeremy M. Dawson. 2017. Text-Independent Speaker Verification Using 3D Convolutional Neural Networks. *CoRR* abs/1705.09422 (2017). arXiv:1705.09422 http://arxiv.org/abs/1705.09422

[14] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. 2016. WIDER FACE: A Face Detection Benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[15] Chunlei Zhang and Kazuhito Koishida. 2017. End-to-End Text-Independent Speaker Verification with Triplet Loss on Short Utterances. In *Interspeech 2017*. ISCA, 1487–1491. https://doi.org/10.21437/Interspeech.2017-1608