UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA

GABRIEL MORAES DA CRUZ

Relatório Final da Atividade de Iniciação Científica: Captura, processamento e análise de grandes massas de dados usando Algoritmos de Deep Learning.

São Paulo
29 de Abril de 2021
GABRIEL MORAES DA CRUZ

Relatório Final da Atividade de Iniciação Científica: Captura, processamento e análise de grandes massas de dados usando Algoritmos de Deep Learning.

Versão Original

Relatório Final da atividade apresentado à Fundação da Universidade de São Paulo no projeto de iniciação científica oferecido através do Laboratório de Arquitetura e Redes de Computadores da Escola Politécnica da Universidade de São Paulo.

Coordenador do Projeto:
Prof. Dr. Wilson Vicente Ruggiero

Docente Responsável:
Profª. Drª. Graça Bressan

Mentor Responsável:
M. José Carlos Gutiérrez Menéndez

São Paulo
2021

FOLHA DE APROVAÇÃO

GABRIEL MORAES DA CRUZ

Relatório Final da Atividade de Iniciação Científica: Captura, processamento e análise de grandes massas de dados usando Algoritmos de Deep Learning.

Relatório final da atividade apresentado à Fundação da Universidade de São Paulo no projeto de iniciação científica oferecido através do Laboratório de Arquitetura e Redes de Computadores da Escola Politécnica da Universidade de São Paulo.

Aprovada em: __/__/2021

Examinador:

_____

RESUMO

A pesquisa foi conduzida de forma a entender o comportamento de estruturas de redes neurais profundas aplicada a reconhecimento facial em diversos tipos de etnias humanas. Para melhor entender esse comportamento foi utilizado seis tipos de bases de dados e dois métodos de aprendizado para comparar não só o desempenho entre os dados mas entre os métodos. Os resultados obtidos mostram que utilizando uma base pré-treinada somada a um treinamento em um banco de dados suficientemente grande é possível eliminar grandes variações de desempenho, diferentemente de estruturas neurais não inicializadas que não são previsíveis tampouco eficientes em ambientes onde há grande diversidade étnica.

Palavras-chave: Rede Neural; Reconhecimento Facial; Etnias; Base Pré-treinada.

Sumário

Introdução

Com a evolução dos métodos de Machine Learning um número maior de problemas vêm sendo resolvidos por meio dessa ferramenta que se mostra tão eficiente para lidar com desafios que apresentam um número grande de dados.

O reconhecimento facial é um dos problemas que é mais atacado pelo método de Machine Learning pois uma imagem digitalizada é um grande conjunto de dados que precisa de um método bastante robusto para poder se adaptar aos mais variados cenários como quantidade de luz, ângulo de fotografia, apetrechos (tais como bonés, óculos, chapéus, etc.), tentativas de fraude, distância, qualidade de imagem, etc. São esses alguns dos problemas que devem ser resolvidos para conseguir-se alguma eficiência nesse tipo de problema.

O problema que é proposto neste trabalho é entender como a variedade de etnias influencia na eficiência de um algoritmo de Machine Learning que pretende classificar indivíduos e reconhecê-los posteriormente com uma taxa operacional, isto é, possível de implementação em um ambiente real para fins de identificação. O trabalho propõe não apenas entender o problema mas também apontar direções para que ele possa ser resolvido, seja por tratamento dos dados seja por estrutura neural ou algum outro caminho.

Utilizar algoritmos Deep Learning para classificar grupos étnicos é um desafio grande, como mostra o trabalho *Ahmed et al, N. 2020 Race estimation with deep networks. Journal of King Saud University - Computer and Information Science*. Por algum motivo ainda não bem delineado essas estruturas têm grande dificuldade em lidar com o conceito de raça, que para um ser humano é de tão fácil reconhecimento. Entretanto a abordagem desse trabalho não é a classificação racial em si mas o comportamento de uma mesma estrutura dentro de diversos grupos raciais.

Metodologia

a) Pesquisa Bibliográfica sobre reconhecimento facial
b) Estudo das técnicas de Machine Learning e Deep Learning
c) Escolha do ambiente de trabalho (linguagem de programação e bibliotecas)
d) Escolha de um banco de dados
e) Tratamento do banco de dados
f) Escolha das estruturas de redes neurais
g) Treinamento e teste da redes neurais
h) Recolhimento dos dados obtidos nos treinamentos e nos resultados
i) Interpretação dos dados e documentação
j) Apresentação do trabalho publicamente.

Desenvolvimento

O banco de dados

O primeiro passo para resolver um problema de Machine Learning é saber que tipo de dado o algoritmo irá trabalhar. Para objetivo deste trabalho será necessário possuir um grande número de fotografias faciais de pessoas de diferentes sexos e etnias para que o algoritmo seja capaz de classificá-las conforme sua identidade. Essa base de dados foi retirada de https://www.kaggle.com/rezkyputri/faceplace, uma base compartilhada em 2019 na comunidade Kaggle, subsidiária da Google LLC.

O treinamento será feito com uma base de dados que possui cinco etnias: Afro Americanos, asiáticos, caucasianos e mestiços; ainda foi feito um sexto tipo em que misturou-se todas essas etnias em uma outra base que será chamada multiracial. Na *Figura 1* temos alguns exemplos de como são a qualidade e a diferença entre os dados de cada uma das bases. Na *Tabela 1* temos a quantidade de imagens e pessoas de cada uma das etnias.

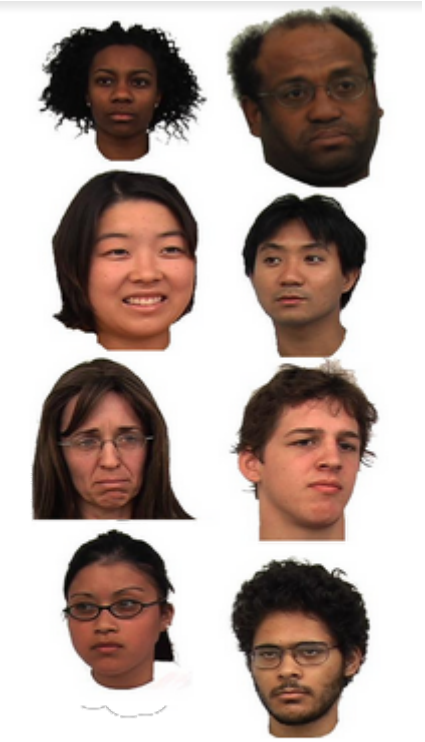

Figura 1 - Amostra de cada etnia. De cima para baixo temos: afro americanos, asiáticos, caucasianos, mestiços. Na coluna da esquerda temos uma mulher e na da direita um homem de cada etnia. (Fonte: Rezky Arisanti Putri)

Etnia	**Número de pessoas**	**Imagens totais**
Afro americanos	34	273
Asiáticos	53	487
Caucasianos	105	1522
Mestiços	18	180
Total:	210	2462

Tabela 1 - Dados do banco de dados (Fonte: Rezky Arisanti Putri)

O tratamento

As imagens apresentadas não possuem grande padronização. Elas apresentam ângulos de fotografia que variam de 0 a 45 graus à esquerda ou à direita, expressões faciais diferentes, apetrechos na cabeça ou nos olhos e rotações em relação ao eixo horizontal.

Faz-se necessário, portanto, padronizar essas imagens conforme um critério comum, e o padrão escolhido foi o alinhamento dos olhos com o eixo horizontal e o recorte da região facial. Esses critérios visam eliminar características alheias ao indivíduo, como cor da roupa, corte de cabelo, etc. O objetivo é capturar a face em uma mesma direção conservando apenas os dados essenciais de cada pessoa para que diminua-se o erro.

Ambos os procedimentos de tratamento utilizam os olhos como referência para finalizar a operação. O primeiro procedimento é a rotação. Identificando a posição dos olhos é feita uma rotação na imagem até que os olhos fiquem alinhados com o eixo horizontal. O procedimento está ilustrado na *Figura 2*. É notável a perda de qualidade da imagem mas a qualidade ainda é suficiente para os objetivos deste trabalho.

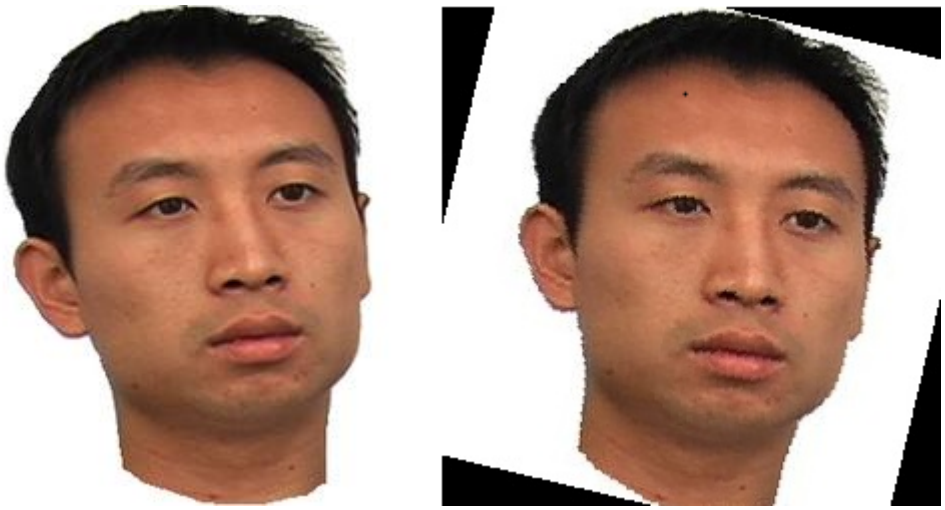

Figura 2 - Antes de depois de uma imagem rotacionada. (Fonte: Rezky Arisanti Putri)

Após o alinhamento é identificada a região facial e é feito um recorte e, por fim, um redimensionamento da imagem para o valor de 150x150. O procedimento completo está ilustrado na *Figura 3*. É importante notar que a ordem dos

procedimentos é essencial. Quando a função de recorte é aplicada às imagens não alinhadas, o recorte não é tão bem realizado como se vê abaixo.

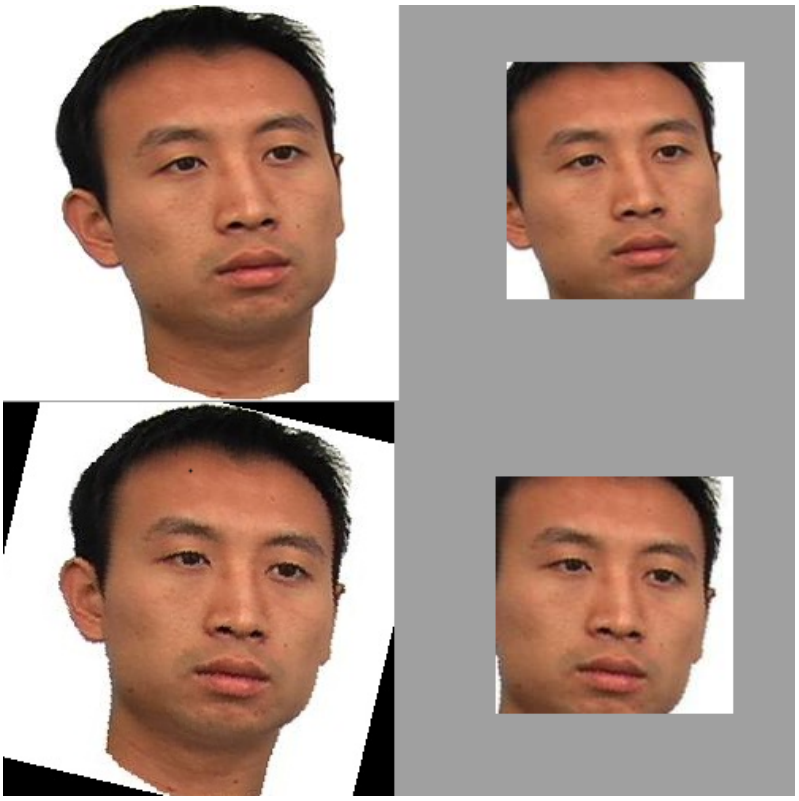

Figura 3 - Comparação dos recortes com rotação e sem rotação. (Fonte: Rezky Arisanti Putri)

Todos esses procedimentos foram aplicados por meio de algoritmos escritos em Python e utilizou-se as bibliotecas NumPy, PIL, OpenCV2 e Autocrop. Esses procedimentos foram realizados de modo a gerar os arquivos intermediários que estão expostos neste relatório, entretanto isso foi feito para fins de recolhimento de dados e não para fins práticos. Em um ambiente operacional é preferível que a imagem seja capturada e completamente processada para a forma matricial final e introduzido à rede neural diretamente, sem intermediários. Isso pode e deve ser feito.

Somado a esses pré-processamentos nas imagens somou-se a utilização da técnica de *data augmentation*, que consiste em aumentar artificialmente a base de dados que possuímos mediante pequenas rotações, translações, variação no brilho e variações na ampliação das imagens. Para o caso específico deste trabalho utilizamos uma variação na rotação de 20 graus e uma translação artificial de 20% na horizontal.

A técnica de *data augmentation é* extremamente importante para os casos de *over-fitting,* isto é, os casos onde as redes neurais são muito profundas para a quantidade de dados disponíveis. Utilizando técnica conseguimos diminuir erros nesses tipos de situações.

O Pré-processamento

Uma imagem em computação é descrita no padrão RGB por uma matriz que possui as dimensões de sua quantidade de pixels horizontais, sua quantidade de pixels verticais e mais três dimensões de espessura em que cada uma delas corresponde a um valor de 0 a 255 correspondentes, respectivamente, ao grau de vermelho, verde e azul. Por exemplo, se temos uma imagem 127x255, então teremos uma matriz (127, 255, 3). Este conceito está ilustrado na *Figura 4*.

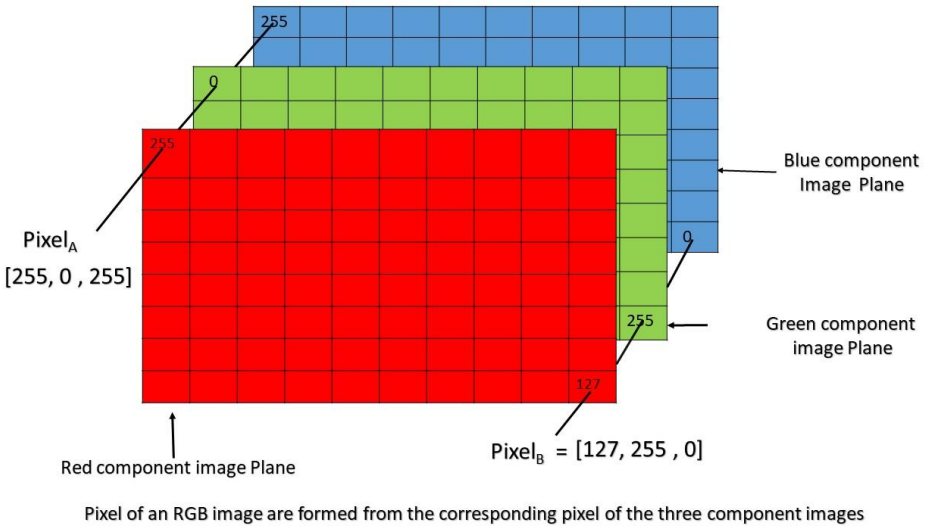

Figura 4 - Matriz RGB de uma imagem 127x255. (Fonte: Gabriel Cruz)

Para adaptar os dados para o algoritmo será feita uma normalização dos valores, dividindo-os todos por 255 (Valor máximo do padrão RGB), e, por fim, a matriz será verticalizada completamente. No exemplo teríamos, por fim, uma matriz (97155, 1), em que 97155 é o produto 127 vezes 255 vezes 3.

Eis a forma final dos dados que serão computadorizados pelos métodos de Machine Learning no procedimento de Back Propagation, o procedimento de tratamento. Por fim, os dados serão colocados em sua forma final utilizando a biblioteca de pré-processamento ImageDataGenerator que faz parte do ferramental da biblioteca Keras, famosa biblioteca de Aprendizado Profundo em Python que funciona baseada na biblioteca TensorFlow. Utilizaremos 20% das imagens para validação dos resultados e 80% para o treinamento, dessa forma a quantidade de imagens para cada fim está explicitado na *Tabela 2*.

Etnia	Número de imagens usadas no treinamento	Número de Imagens usadas na validação
Afro americanos	234	39
Asiáticos	418	69
Caucasianos	1263	259
Mestiços	158	22
Multiracial	2214	411

Tabela 2 - Relação de quantidade de imagens para treino e para validação. (Fonte: Gabriel Cruz)

A estrutura Deep Learning

Os modelos

A partir das bases construídas é necessário estruturar algoritmos de Machine Learning para o treinamento e reconhecimento dessas imagens para a obtenção de resultados operacionais.

Foram construídas sete estruturas para os treinamentos, divididas em dois grupos majoritários, os com uso de Transfer Learning e o grupo sem Transfer Learning que consiste em apenas uma rede.

A primeira rede, exposta na Figura 5, é uma rede profunda inicializada aleatoriamente e treinada desde o seu início com o banco de dados já apresentado, nesta rede não foi utilizada a técnica de Transfer Learning. A biblioteca Keras do Python foi utilizada para sua construção, treinamento e validação.

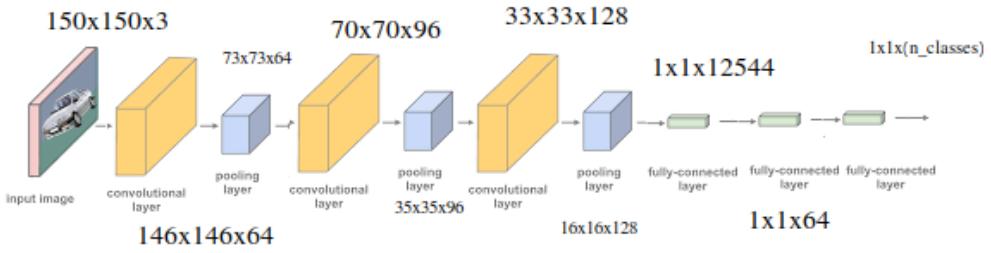

Figura 5 - Estrutura da rede inicializada aleatoriamente. (Fonte: Gabriel Cruz)

As outras estruturas de redes foram feitas utilizando o método de Transfer Learning, para sua implementação foram utilizadas as API 's da biblioteca Keras que possui todas as redes pré-treinadas ilustradas na Tabela 3.

Para objetivos de comparação foi utilizada a mesma estrutura final em cada uma dessas redes. Dessa forma utilizamos a rede pré-treinada como entrada congelando todas as camadas de treinamento já existentes de modo a manter o trabalho dos autores que as propuseram e conectamos uma estrutura treinável composta de 3 camadas de redes-neurais completamente conectadas *(fully-connected NN layers)* com a quantidade de neurônios de, respectivamente, 512, 64, e um número final de neurônios igual a quantidade de pessoas que devem ser reconhecidas de cada base de dados, ou, genericamente, classes. Em cada uma dessas camadas foi utilizada a função de ativação SoftMax e uma regularização Dropout de valor de *rate* igual a 0,5.

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	790	945	22,910,480	126	109,42	8,06
VGG16	528	713	901	138,357,544	23	69,5	4,16
VGG19	549	713	900	143,667,240	26	84,75	4,38
ResNet50	98	749	921	25,636,712	-	58,2	4,55
ResNet101	171	764	928	44,707,176	-	89,59	5,19

ResNet152	232	766	931	60,419,944	-	127,43	6,54
ResNet50V2	98	760	930	25,613,800	-	45,63	4,42
ResNet101V2	171	772	938	44,675,560	-	72,73	5,43
ResNet152V2	232	780	942	60,380,648	-	107,5	6,64
InceptionV3	92	779	937	23,851,784	159	42,25	6,86
InceptionResNetV2	215	803	953	55,873,736	572	130,19	10,02
MobileNet	16	704	895	4,253,864	88	22,6	3,44
MobileNetV2	14	713	901	3,538,984	88	25,9	3,83
DenseNet121	33	750	923	8,062,504	121	77,14	5,38
DenseNet169	57	762	932	14,307,880	169	96,4	6,28
DenseNet201	80	773	936	20,242,984	201	127,24	6,67
NASNetMobile	23	744	919	5,326,716	-	27,04	6,7
NASNetLarge	343	825	960	88,949,818	-	344,51	19,96
EfficientNetB0	29	-	-	5,330,571	-	46	4,91
EfficientNetB1	31	-	-	7,856,239	-	60,2	5,55
EfficientNetB2	36	-	-	9,177,569	-	80,79	6,5
EfficientNetB3	48	-	-	12,320,535	-	139,97	8,77
EfficientNetB4	75	-	-	19,466,823	-	308,33	15,12
EfficientNetB5	118	-	-	30,562,527	-	579,18	25,29
EfficientNetB6	166	-	-	43,265,143	-	958,12	40,45
EfficientNetB7	256	-	-	66,658,687	-	1578,9	61,62

Tabela 3 - Dados de todas as bases pré-treinadas disponíveis na biblioteca Keras
(Disponível em: https://keras.io/api/applications/)

Para ilustrar o procedimento a Figura 6 possui a estrutura da rede VGG16 proposta no artigo *Simonyan & Zisserman, N. Very Deep Convolutional Networks for scale image recognition. Apresentado no ICLR 2015, 10 de Abril, 2015, 14. Disponível em https://arxiv.org/abs/1409.1556.* Trata-se de uma rede profundíssima como mostra a arquitetura da Figura 6, treinada com uma base de dados de mais de 1.2 milhões de imagens e 1000 classes diferentes, obtendo uma taxa de acerto de 92,7% na ImageNet, uma base de dados famosa na academia para disputar resultados

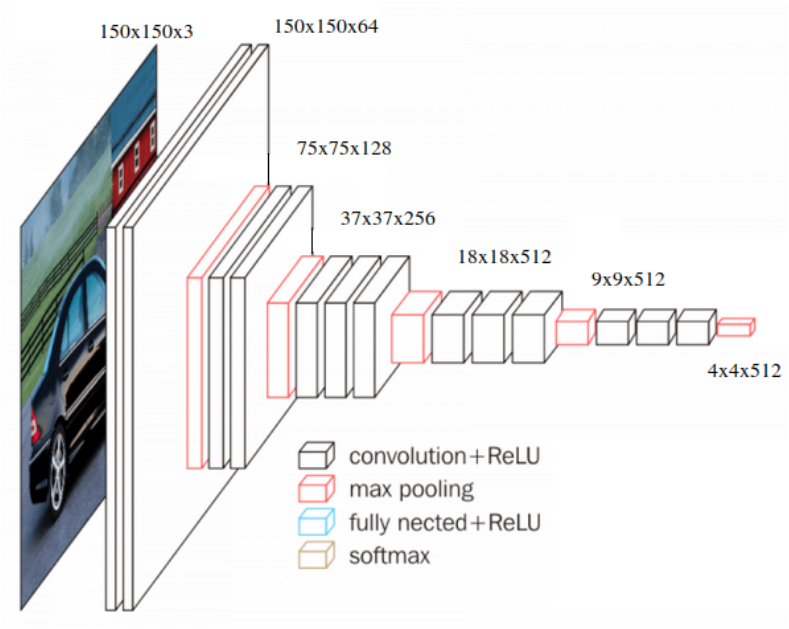

Figura 6 - Estrutura VGG16. (Fonte: Gabriel Cruz)

A preparação para sua utilização foi adaptar a entrada da rede para dados no formato que a base aqui apresentada necessita (150, 150, 3), congelar todas as camadas pré-treinadas para que os parâmetros obtidos pelos autores não sejam alterados e, por fim, acoplar 3 camadas finais treináveis para que a base seja aplicável ao problema aqui proposto. O resultado final está exposto graficamente na Figura 7.

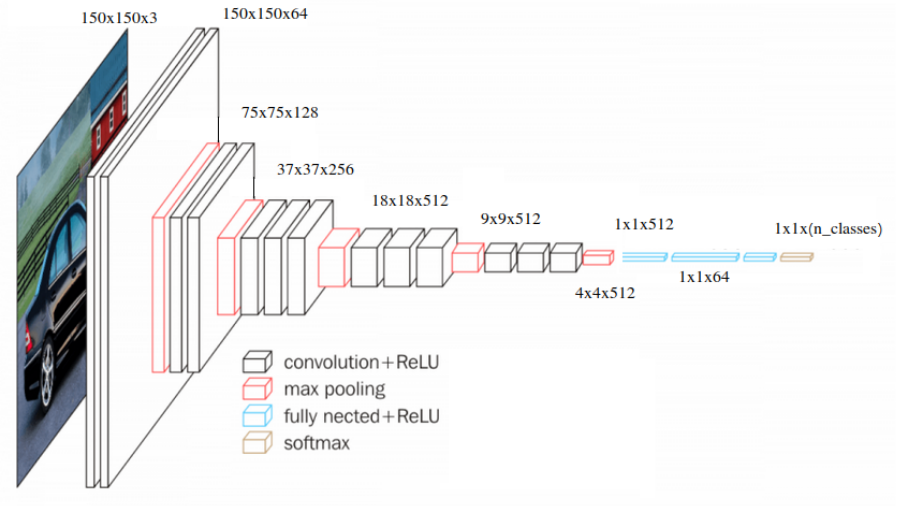

Figura 7 - Modelo com Transfer Learning. (Fonte: Gabriel Cruz)

Esses mesmos procedimentos foram realizados para todas as bases exibidas na Tabela 3, não estão aqui por mera questão de objetividade do relatório. Por fim, a quantidade de parâmetros treináveis e não treináveis de cada estrutura completa está exibido na Tabela 4.

Modelo	**Parâmetros treináveis**	**Parâmetros não-treináveis**
Sem Transfer Learning	1,328,395	1,088
Com VGG16	310,763	138,357,544
Com VGG19	310,763	143,667,240
Com Xception	310,763	22,910,480
Com ResNet50	310,763	25,636,712
Com ResNet101	310,763	44,707,176
Com ResNet152	310,763	60,419,944
Com ResNet50V2	310,763	25,613,800
Com ResNet101V2	310,763	44,675,560
Com ResNet152V2	310,763	60,380,648
Com InceptionV3	310,763	23,851,784
InceptionResNetV2	310,763	55,873,736
MobileNet	310,763	4,253,864
MobileNetV2	310,763	3,538,984
DenseNet121	310,763	8,062,504
DenseNet169	310,763	14,307,880
DenseNet201	310,763	20,242,984
NASNetMobile	310,763	5,326,716
NASNetLarge	310,763	88,949,818
EfficientNetB0	310,763	5,330,571
EfficientNetB1	310,763	7,856,239
EfficientNetB2	310,763	9,177,569
EfficientNetB3	310,763	12,320,535
EfficientNetB4	310,763	19,466,823

EfficientNetB5	310,763	30,562,527
EfficientNetB6	310,763	43,265,143
EfficientNetB7	310,763	66,658,687

Tabela 4 - Quantidade de parâmetros de cada modelo. (Fonte: Gabriel Cruz)

O treinamento

O treinamento consistiu em treinar cada uma das 5 bases de dados em cada uma das 27 estruturas construídas. Os dados recolhidos de cada um dos treinamentos foram: a perda de treinamento, taxa de acerto de treinamento, perda de validação e taxa de acerto de validação durante 600 epochs.

O ambiente utilizado para treinamento foi o Google Colab. Os motivos que levaram a essa escolha foram a capacidade de integração com o Google Drive, facilitando a versatilidade e praticidade da pesquisa; e o hardware fornecido, uma GPU NVIDIA P100s, K80s ou P4s sendo impossível saber com qual delas foi feito cada treinamento. Mas todas elas são suficientes para o tipo de processamento exigido neste trabalho.

Nas dadas condições o tempo de treinamento total foi de aproximadamente 240 horas, desconsiderando momentos de adaptação dos algoritmos de treinamento.

Tratamento dos dados

Para a visualização dos dados foi utilizada a biblioteca MatPlotLib e Pandas e por meio dela serão plotados os dados de acordo com as epochs. As epochs são a repetição do processo iterativo de aprendizado.

Foi utilizado a média móvel para análise dos dados. A escolha deste filtro foi feita por sua capacidade de suavizar curvas e analisar tendências, isso condiz com o objetivo deste trabalho de entender por quais meios podemos indicar para reduzir as diferenças de reconhecimento entre etnias.

Devido aos resultados obtidos foi escolhido ignorar os dados de perdas por não complementarem a análise de tendência na eficácia do modelo. Pelo contrário, os gráficos de accuracy (taxa de acerto, em inglês) conseguem mostrar o desempenho dos algoritmos e dos bancos de dados.

Resultados

O primeiro problema identificado foi a dificuldade de algumas bases pré-treinadas em obter resultados satisfatórios, isso ocorreu com a grande maioria delas. Foi utilizado o critério filtro de que estruturas de treinamento que não ultrapassassem 10% de acerto na validação em ao menos 100 epochs seriam descartadas para análises futuras.

Foi exatamente o que ocorreu com as bases ResNet50, ResNet101, ResNet152, ResNet50V2, ResNet101V2, ResNet152V2, MobileNetm MobileNetV2, DenseNet121, DenseNet169, DenseNet201, NASNetMobile, NASNetLarge, EfficientNetB0, EfficientNetB1, EfficientNetB2, EfficientNetB3, EfficientNetB4, EfficientNetB5, EfficientNetB6, EfficientNetB.

Leva-se a crer que o motivo do baixo desempenho das redes acima deve-se ao fato de estas redes se encontrarem em região de underfitting, onde a profundidade da rede é muito menor do que a requerida, ou overfitting, onde a profundidade é muito acima da requerida. Entretanto, se uma grande quantidade de redes obtivesse sucesso na análise dos dados seria muito mais complexo do que já foi. Portanto, crê-se que este filtro inicial é muito conveniente para os fins deste projeto.

Por fim, os dados das bases estruturadas com VGG16, VGG19, Xception, InceptionV3, InceptionResNetV2 e a rede sem uso de Transfer Learning foram coletados, filtrados com uma média móvel de tamanho 15 e plotados dividindo-os em acurácia de treino e de teste, e em cada um desses gráficos foi plotado a evolução de cada uma das bases de interesse. Vide a figura 8.

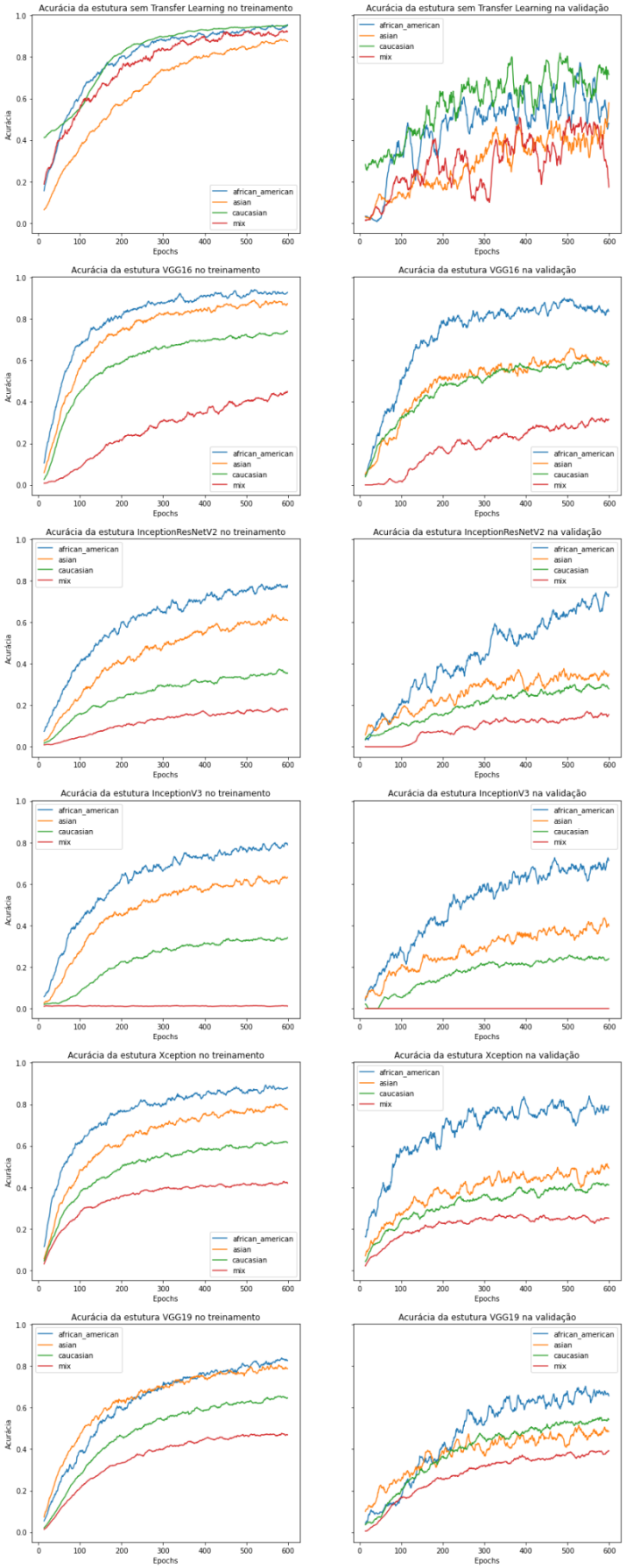

Figura 8 - Evolução dos treinamentos (Fonte: Gabriel Cruz)

Para efeitos de comparação decidiu-se plotar um gráfico com a média de desempenho das redes como se apresenta na figura 9, este gráfico é interessante principalmente quando compara-se os resultados de treino e de validação.

Também tem-se um gráfico em barra que analisa o desempenho final comparativo em cada uma das bases e que deve ser tomado como resultado final líquido na Figura 10 e a comparação geral entre os valores de validação e de treinamento nas figuras 11 e 12.

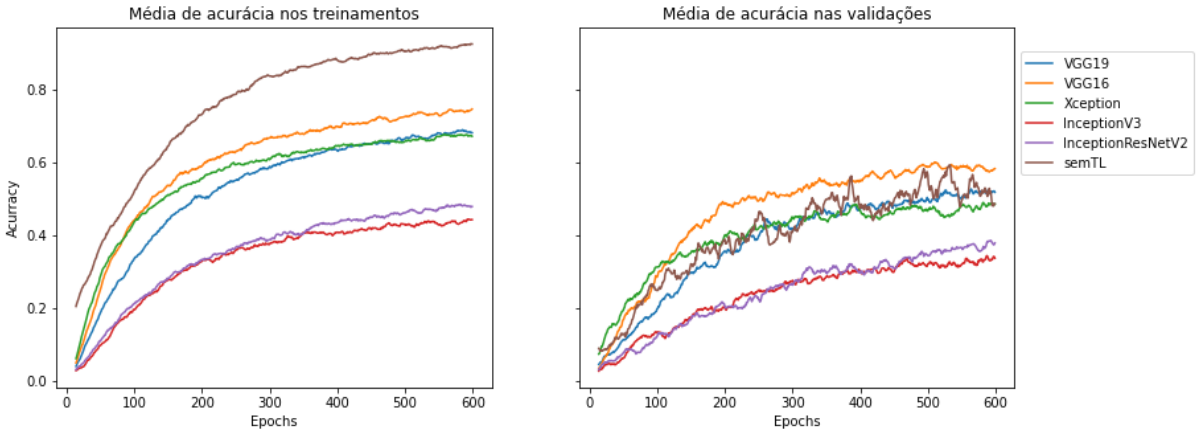

Figura 9 - Evolução Média das redes (Fonte: Gabriel Cruz)

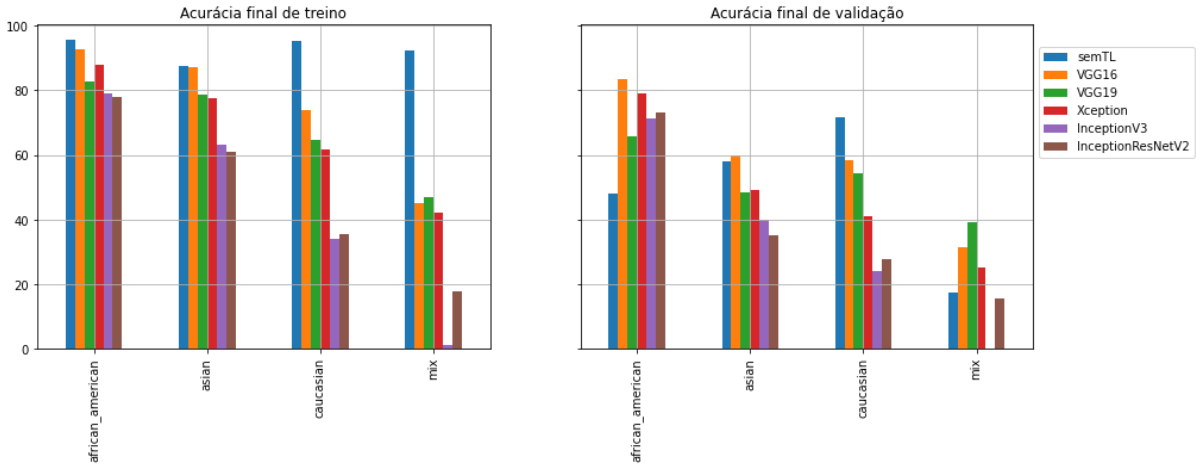

Figura 10 - Resultados Finais (Fonte: Gabriel Cruz)

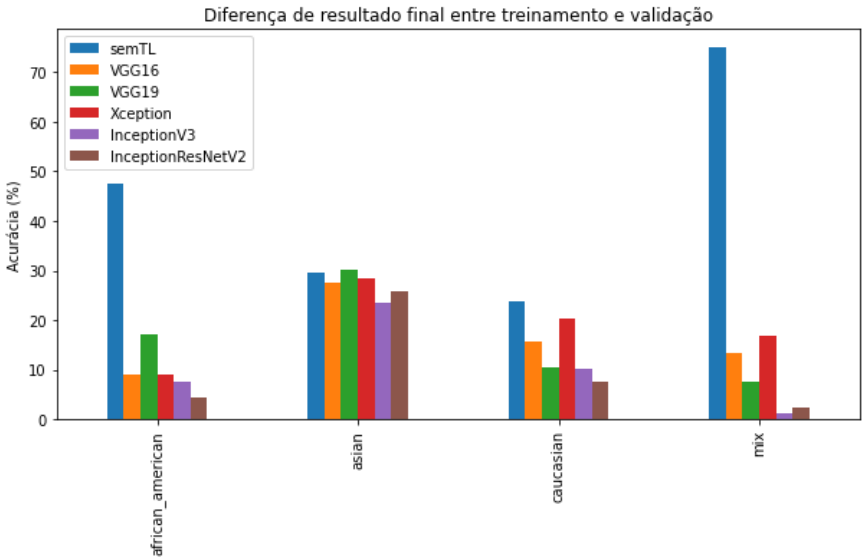

Figura 11 - Comparação entre os resultados de treinamento e validação (Fonte: Gabriel Cruz)

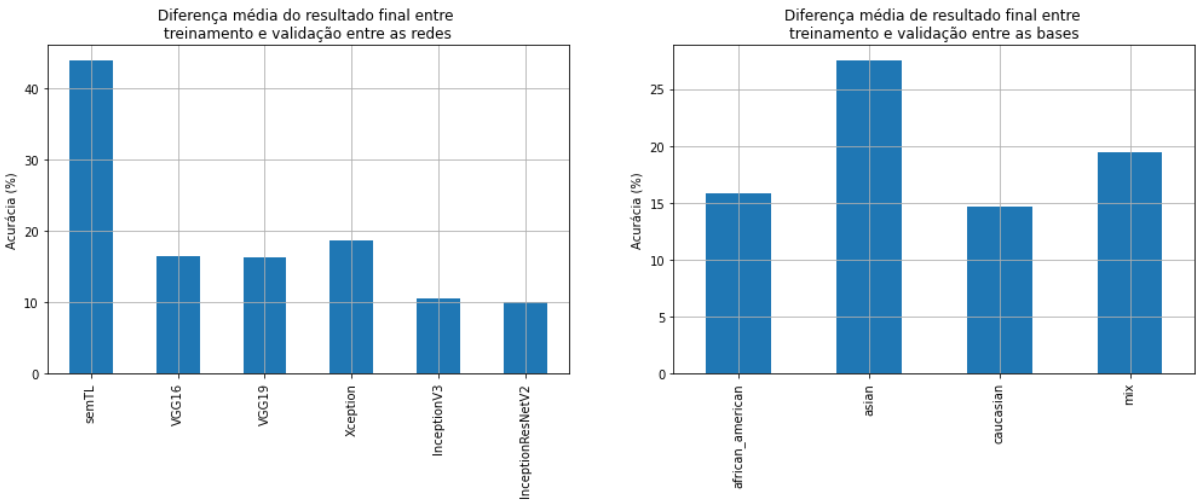

Figura 12 - Comparação entre as diferenças médias de validação e treinamento. (Fonte: Gabriel Cruz)

Análise

1. Evolução

Em primeiro lugar é necessario analisar a evolução dos treinamentos e dos testes de validação. Na grande maioria dos casos foi observada a curva característica assintótica de treinamento em Machine Learning, fato este que mostra que não há grandes problemas de adequação da base de dados à estrutura neural proposta neste trabalho.

Entretanto observamos duas exceções bastante latentes. Em primeiro lugar a rede neural baseada na rede pré-treinada InceptionV3 obteve um desempenho assombroso no treinamento e validação da base mista, devendo ser descartada completamente já que não corrobora para os fins últimos do trabalho.

Em segundo lugar observou-se uma evolução linear no treinamento e validação da base mista utilizando a rede baseada em VGG16. Esse resultado é excitante pelo motivo de haver perspectiva de crescimento caso aumente-se o número de epochs no treinamento.

É notável também, quando analisa-se a evolução, a presença de uma grande volatilidade na validação. Essa volatilidade foi reduzida em grande medida quando aplicada uma rede neural pré-treinada, este fato é suficiente para excluir a rede sem o uso de Transfer Learning já que a previsão de resultados fica comprometida, fato este que não caminha juntamente com os objetivos deste trabalho.

2. Resultados Gerais

As figuras 9 e 10 são absolutamente essenciais para entender os resultados finais de cada uma das redes neurais propostas. Em primeiro lugar, é necessário dizer que as redes neurais InceptionV3 e InceptionResNetV2 obtiveram um desempenho de cerca de 10% abaixo dos de suas concorrentes. O valor de 10% é expressivo nestes cenários pois pode ser a diferença entre o limiar de 50% que dirá se a rede mais acerta ou mais erra em suas previsões.

As redes com melhor desempenho, descartando a rede sem Transfer Learning por motivos supracitados, são as redes VGG16, VGG19 e Xception, onde a rede Xception é a que mais sofre com a questão da volatilidade.

Ainda nessas três redes, a Xception é a rede que sofre com a maior diferença média entre valores de validação e valores de treinamento o que a coloca abaixo de suas concorrentes (vide figura 12). As outras duas do tipo VGG possuem volatilidade e diferença de acurácia treino-teste bem parecidas, embora VGG19 obtenha leve vantagem nestes quesitos.

3. Etnias

Também é necessário entender o desempenho de todas as redes em seu conjunto quando treinadas em cada uma das bases que por sua vez tem a proposta de representar uma etnia.

É de fácil visualização mediante a figura 10 que a base de afro-americanos é a com melhor desempenho geral e a base mista é a com o pior desempenho, esta última estando dentro do previsto. As bases de asiáticos e caucasianos ficaram em uma posição intermediária e muito parecidas entre si.

É interessante notar, entretanto, que a base com a maior diferença entre acurácia de treinamento e de validação é a base de asiáticos e logo em seguida está a base mista. Isso pode indicar que há algum problema na correspondência entre as imagens de teste e de validação nessas duas bases, visto que essas diferenças são, respectivamente, de 27% e 19%.

Conclusão

Fica claro com os resultados obtidos a necessidade de uma rede neural profundíssima para lidar com o problema. A rede inicializada aleatoriamente não alcançou resultados satisfatórios na validação em nenhum dos casos, seja pelo valor final de acerto ou pela instabilidade desse mesmo valor.

Entre as redes com melhor desempenho temos as redes VGG16 e VGG19 onde esta última possui uma quantidade menor de pontos fracos como menor volatilidade e menor diferença entre valores de treinamento e validação, entretanto também possui uma menor acurácia.

Pelos motivos citados quando falou-se da evolução das acurácias, este trabalho conclui que a melhor rede para os fins deste trabalho é a rede construída em cima de VGG16. O motivo com maior peso é a acurácia média já alcançada e a curva linear no treinamento da base mista que mostra uma perspectiva de aprimoramento de resultados nesta base que ofereceu tantas dificuldades às outras redes aqui testadas.

Um modelo ainda mais profundo pode ser um caminho interessante para resolver isso. No modelo VGG16 acoplamos apenas mais 3 camadas não-lineares, talvez acoplar algumas de convolução antes dessas pode ser uma boa alternativa.

Quanto às avaliações referentes às bases de dados é necessário checar o grande valor de diferença entre acurácia entre treinamento e teste nas bases asiática e mista. É possível, como já foi dito, que haja algum problema de correspondência entre a fração de validação e de treinamento, embora não tenha sido identificada durante o desenvolvimento deste trabalho.

Por fim, crê-se que o trabalho contribui para o entender o comportamento de Redes Neurais Profundas para identificação facial em diversos cenários diferentes. Foi mostrado as vantagens de utilizar bases pré-treinadas para acelerar o treinamento e obter melhores resultados e também foram indicados problemas que devem ser superados.

Agradecimentos

Agradeço neste trabalho os colegas que comigo desenvolveram essas atividades: André Devay Torres Gomes e Juliana Marques da Cruz. Agradeço ao José Carlos Gutiérrez Menéndez, doutorando do LARC e principalmente a professora Graça Bressan por ter me orientado durante este projeto e ao LARC (Laboratório de Arquitetura de Redes e Computadores) laboratório junto ao qual desenvolvi a pesquisa.

Agradeço a oportunidade de ter apresentado este projeto no 29º Simpósio Internacional de Iniciação Científica e Tecnológica da USP - SIICUSP e pelos professores que avaliaram e fizeram questionamentos.

Por fim agradeço a FUSP pelo financiamento da bolsa e a Universidade de São Paulo que juntamente com a Escola Politécnica da Universidade de São Paulo me proporcionaram a experiência de desenvolver essa pesquisa.

Referências

Valiente, Rodolfo; Gutiérrez, José C; Sadaike, Marcelo T; Bressan, Graça; "Automatic Text Recognition in Web Images", Proceedings of the 23rd Brazilian Symposium on Multimedia and the Web, 241-244,2017.

Gutiérrez, José C; Valiente, Rodolfo; Sadaike, Marcelo T; Soriano, Daniel F; Bressan, Graça; Ruggiero, Wilson V; "Mechanism for Structuring the Data from a Generic Identity Document Image using Semantic Analysis",Proceedings of the 23rd Brazilian Symposium on Multimedia and the Web,,,213-216,2017.

Gutiérrez, Armando M; Pacheco, Patricia A; Gutiérrez, José C; Bressan, Graça; "Development of a naive bayes classifier for image quality assessment in biometric face images", Proceedings of the 25th Brazilian Symposium on Multimedia and the Web,,,177-180,2019.

R. Szeliski, Computer Vision: algorithms and applications. London; New York: Springer, 2011

O. Kahm and N. Damer, "2D face liveness detection: An overview," BIOSIG-Proceedings IEEE Int. Conf. the. Biometrics Spec. Interes. Gr. (BIOSIG)., pp. 171–182, 2012.

M. Saini and C. Kant, "Liveness Detection for Face Recognition in Biometrics : A Review," pp. 31–36.

M. Singh and A. S. Arora, "A Novel Face Liveness Detection Algorithm with Multiple Liveness Indicators," Wirel. Pers. Commun., vol. 100, no. 4, pp. 1677–1687, 2018.

Simonyan & Zisserman, N. Very Deep Convolutional Networks for scale image recognition. Apresentado no ICLR 2015, 10 de Abril, 2015, 14. Disponível em https://arxiv.org/abs/1409.1556.