



TRABALHO DE GRADUAÇÃO

**REDE NEURAL CONVOLUCIONAL
APLICADA AO RECONHECIMENTO DAS
LINHAS DAS PALMAS DA MÃO
COMO FORMA DE AUTENTICAÇÃO BIOMÉTRICA**

Alberto Junqueira Carvalho Branquinho

Brasília, Novembro de 2021

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

TRABALHO DE GRADUAÇÃO

**REDE NEURAL CONVOLUCIONAL
APLICADA AO RECONHECIMENTO DAS
LINHAS DAS PALMAS DA MÃO
COMO FORMA DE AUTENTICAÇÃO BIOMÉTRICA**

Alberto Junqueira Carvalho Branquinho

*Relatório submetido ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Engenheiro Eletricista*

Banca Examinadora

Prof. Alexandre Ricardo Soares Romariz
Orientador

Prof. João Paulo Leite
Examinador Interno

Prof. Ricardo Zelenovsky
Examinador Interno

Agradecimentos

Gostaria de dedicar este trabalho à minha família, meu pai Alberto Carvalho, minha mãe Maria Emília, minha irmã Daniela Junqueira e minha segunda mãe Silvia Tereza, a quem tudo devo por estar onde estou hoje, pelo apoio nestes longos e desafiadores anos de estudo, e por todo o amor que um filho pode precisar. A meus colegas de curso, que com o companheirismo e apoio nestes anos tornaram a caminhada uma experiência memorável, Guilherme Kfourri, Fausto Brenner, João Pedro, Matheus Costa e em especial Pedro Carrara, meu veterano que me guiou por esta trajetória. À minha namorada Sofia Prado pelo nosso companheirismo e amor, por me lembrar todos os dias o motivo pelo qual vivemos. Também aos professores e funcionários de todo o Departamento de Engenharia Elétrica da UnB pelo bom trabalho realizado. Agradecer em especial grandes mestres que possuem o dom de lecionar e o fazem como tal, primeiramente professor Alexandre Romariz por me guiar nesta caminhada final do curso, ao professor João Paulo Leite pelas excelentes aulas e empenho em sua profissão como professor e coordenador, ao professor Daniel Chaves Café, também pelo grande dom de lecionar e por sua empatia com os alunos. Agradecer também ao professor Lineu da C. Araújo Neto, por suas excelentes aulas e carisma com os alunos e pelas motivações verbais que muitas vezes falta aos ouvidos dos alunos. Ao Pedro Ferreira pela enorme sabedoria e auxílio na elaboração da rede.

Seria inviável agradecer a todos que fizeram parte desta jornada e injusto só agradecer aos mencionados acima. A importância de nossos amigos nesta jornada é inquestionável. Gostaria de agradecer especialmente Gabriel Arantes, meu melhor amigo com quem dividi 8 anos de uma jornada similar e esteve comigo nos bons e maus momentos. Ao Vitor Dias pela amizade de 14 anos e por viver muitos momentos desta jornada comigo.

Por fim, gostaria de agradecer a mim, pela minha dedicação no curso, pelas noites mal dormidas, pela abdicação de momentos por noites de estudo, por ter superado todo o estresse, pressão, evolução que um curso de Engenharia Elétrica exige. Foi uma longa jornada mas muito gratificante. Obrigado à matemática, à física e ao conhecimento, e que estejamos sempre evoluindo.

Alberto Junqueira Carvalho Branquinho

RESUMO

O objetivo do trabalho é avaliar a acurácia de uma Rede Neural Convolutacional na autenticação biométrica por meio do padrão das linhas das mãos. A base de dados utilizada conta com 1301 imagens de palmas de mãos (lado esquerdo) de 230 indivíduos diferentes. Por meio da utilização de redes pré treinadas como vgg16 e googlenet, a transferência de aprendizado da rede é feita e adaptada ao problema desejado.

Palavras Chave: CNN, Transfer Learning, Linhas das mãos, Autenticação Biométrica, Principal Resultado, Perspectiva Futura

ABSTRACT

The goal of this study is to evaluate which accuracy a Convolutional Neural Network can achieve in biometric authentication through the pattern Palm Prints. The database used has 1301 images of palms (left side) of 230 different individuals. Through the use of pre-trained networks such as vgg16 and googlenet, the transfer of learning from the network is made and adapted to the desired problem.

Keywords: CNN, Transfer Learning, Palm Prints, Biometric Authentication, Best Result, Future Development

SUMÁRIO

1	INTRODUÇÃO	1
1.1	OBJETIVOS DO TRABALHO	2
1.2	UNICIDADE DAS IMPRESSÕES DA PALMA DAS MÃOS	2
1.3	EVOLUÇÃO DO PODER COMPUTACIONAL	4
1.4	COMPUTAÇÃO EM NUVEM	4
1.5	APRESENTAÇÃO DO MANUSCRITO	6
2	PRINCÍPIOS DO DESENVOLVIMENTO	7
2.1	REDES NEURAIS CONVOLUCIONAIS	7
2.2	VGG-16	10
2.3	INFORMAÇÕES DA BASE DE DADOS	11
2.4	ESPAÇO DE DESENVOLVIMENTO INTEGRADO	13
3	PROCEDIMENTOS E RESULTADOS	14
3.1	IMPLEMENTAÇÃO	14
3.2	RESULTADOS	15
3.3	COMPARAÇÕES/BENCHMARKS	17
4	CONCLUSÕES	18
4.1	USO DE OUTRAS REDES PRÉ TREINADAS	18
4.2	TAMANHO DA BASE E PRÉ PROCESSAMENTO	18
4.3	TRABALHOS FUTUROS	19
4.4	CONCLUSÃO	19
	REFERÊNCIAS BIBLIOGRÁFICAS	20

LISTA DE FIGURAS

1.1	Autenticação por Análise da Palma da Mão.....	2
1.2	Impressão de Palma.....	3
1.3	Lei de Moore - Evolução temporal do poder computacional.....	4
1.4	Evolução Cloud Computing - Geeksforgeeks.....	5
2.1	Exemplo de Estrutura do Problema	7
2.2	Estrutura da VGG-16	10
2.3	Exemplo de imagem do banco de dados	12
2.4	Comparação bases de dados	13
3.1	Acompanhamento de resultados a cada época	15
3.2	Tentativas e Acurácias - Gráfico	16
3.3	Exemplo de resultado final da rede	17
3.4	Benchmarks GoogleNet e Autoral 11kHands	17

Capítulo 1

Introdução

As Redes Neurais Convolucionais foram introduzidas em 1980 por Yann LeCun. As primeiras aplicações deste modelo de rede neural foram de reconhecimento de dígitos escritos à mão. A base de dados MNIST foi trabalhada pela rede para identificar qual era o dígito escrito à mão nas imagens da base de dados. Excelentes resultados foram obtidos nesta primeira aplicação, com níveis de acurácia superiores a 90% e foi então utilizada por diversos serviços de correios e bancos, por exemplo. Esta solução até hoje está presente em diversos setores que se beneficiam desta visão computacional, mas apesar do bom desempenho na aplicação realizada, as Redes Neurais Convolucionais foram negligenciadas por algum tempo nos anos que se seguiram. O motivo para isso foi por não haver, na época, alternativas de escalabilidade, ou seja, não havia poder computacional e nem a quantidade de dados exigidas para performances boas em aplicações mais robustas.

A partir de 2010, Aprendizado Profundo voltou a ser estudado e comentado no âmbito científico. Dez anos atrás, o poder computacional já tinha atingido proporções suficientes para reintegrar o Aprendizado Profundo ao âmbito científico. O acesso a formas de armazenamento de dados também foi revolucionado. De 1970 com disquetes que armazenavam dados da ordem de kilo bytes, até o século XXI, avanços tremendos aconteceram nesta área, possibilitando o armazenamento de giga bytes e tera bytes de dados que cabem no seu bolso, ou melhor, estão disponíveis espaços de mesma magnitude em nuvem. O surgimento de grandes bases de dados como a ImageNet, que contém milhões de imagens rotuladas, apresentam excelentes resultados em uma rede neural quando treinada em computadores que tenham porte para tal treinamento. Desde então, as Redes Neurais Convolucionais têm sido aplicadas a diversos problemas modernos relacionados à visão computacional, como reconhecimento facial, análise de documentos, análises climáticas, controle de conteúdo em mídias sociais. Todos estes desafios são resolvidos por redes neurais convolucionais com níveis de acurácia excelentes em seus âmbitos.

1.1 Objetivos do Trabalho

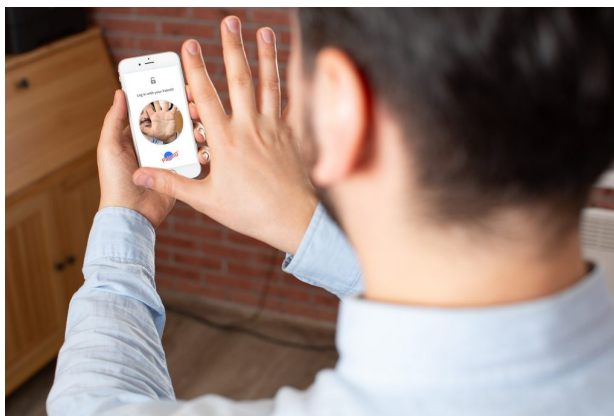


Figura 1.1: Autenticação por Análise da Palma da Mão

Este trabalho teve origem ao se avaliar a necessidade de formas mais simples e baratas de se realizar autenticações biométricas em problemas reais. O uso de câmeras já é amplamente diluído e seus custos já se tornaram acessíveis, assim, uma solução de autenticação por meio de imagens obtidas por estas câmeras seria de fácil implementação. O uso de inteligência artificial com imagens e aprendizado de padrões, há anos tem sido trabalhado e permite o atingimento de níveis excelentes de acurácia, permitindo uma autenticação confiável e barata.

Aplicações de autenticação biométrica requerem um determinado nível de acurácia para serem confiáveis a identificação de indivíduos. O objetivo é obter uma rede que possua acurácia acima de 80 por cento, alguns estudos atingiram níveis acima de 90 por cento, sendo razoável, portanto, estabelecer esta meta de acurácia. [1] O sistema aqui desenvolvido, atua sobre uma base de dados de 2601 imagens de fotos de palmas de mãos. Uma rede treinada sobre esta base deverá ser capaz de, ao dar como entrada uma foto de uma palma de mão, predizer a qual classe esta mão pertence. Os resultados encontrados neste estudo mostram a viabilidade da autenticação biométrica por fotos das palmas das mãos. Este documento abordará a parte teórica das redes neurais convolucionais, a base de dados utilizada neste trabalho e por fim, os resultados obtidos na rede de melhor performance testada.

1.2 Unicidade das Impressões da Palma das mãos

Um indivíduo se exercitando em 1997 encontrou uma marca de palma de mão com sangue na rua. [2] Anos após o crime, o então governador da Califórnia Arnold Schwarzenegger ofereceu 50 mil dólares de recompensa, e o órgão governamental utilizou sua base de dados de imagens de impressões de palmas de mãos, e houve correspondência entre as imagens, sendo possível identificar o autor do crime com um nível de confiança científico razoável para incriminá-lo. Abaixo é ilustrado diversos pontos cruciais da biometria da palma da mão utilizados na análise de cenas de crimes.

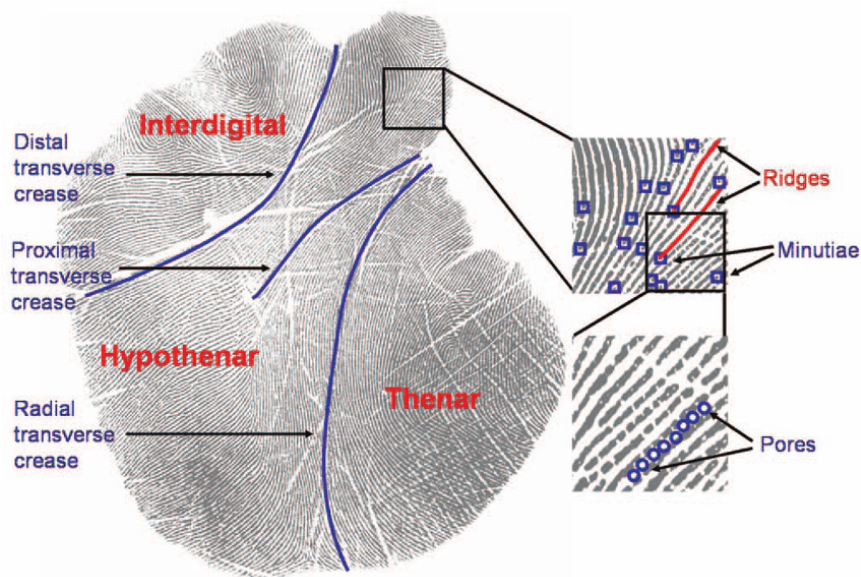


Figura 1.2: Impressão de Palma

Oficiais americanos estimam que um terço das marcas nas cenas de crimes são de palmas de mãos. As bordas e linhas da palma das mãos e seus padrões são únicos e são tão acurados quanto impressões digitais de dedos. [3] Impressões das palmas têm em média mil características passíveis de identificação, enquanto as impressões de dedos possuem em média cem. [4] Um maior número de características, embora seja bom no quesito de fornecer mais informação passível de aprendizado, implica numa maior demanda de poder computacional e de base de dados para o treinamento. O estudo das linhas das palmas das mãos é compartilhado por diversas áreas, possibilitando dessa forma, o acesso à base de dados prontas de imagens de palmas de mãos rotuladas por pessoa. Estas bases podem variar em tamanho, diversidade de indivíduos, informações adquiridas junto a imagem, etc.

1.3 Evolução do Poder Computacional

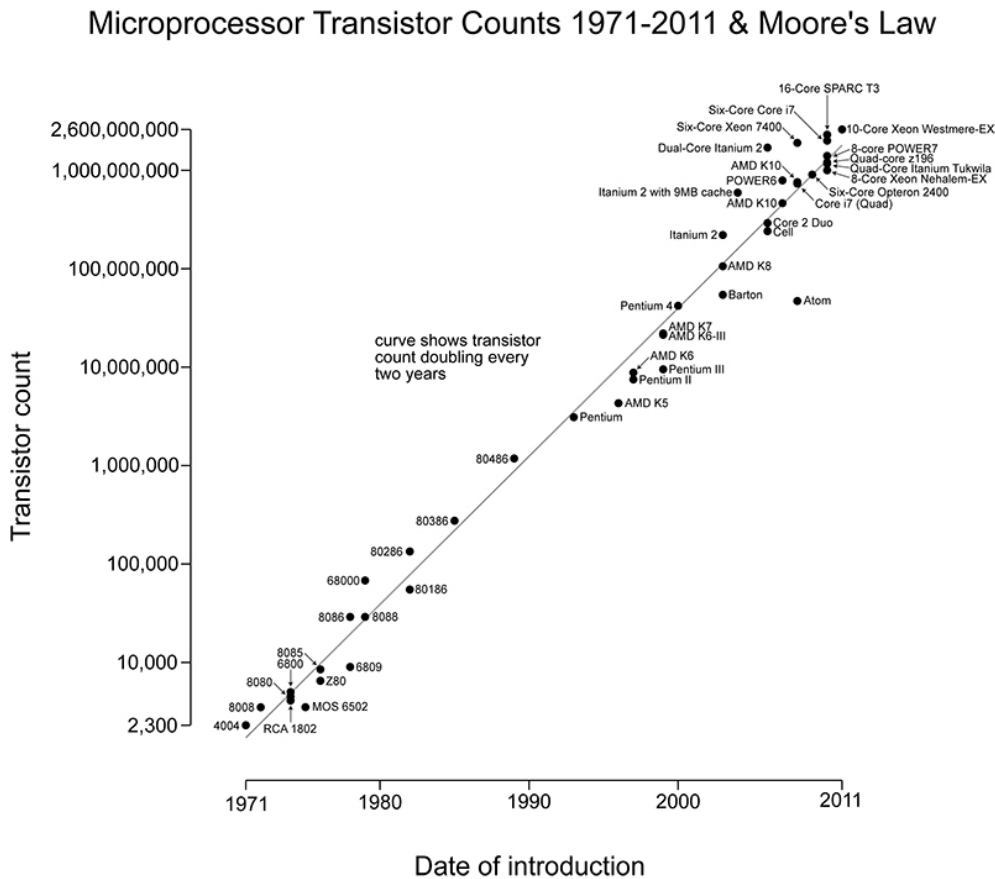


Figura 1.3: Lei de Moore - Evolução temporal do poder computacional

A evolução da densidade de poder computacional por volume tem sido constante há mais de 7 décadas. A lei de Moore é a observação de que o número de transistores em um circuito integrado (CI) denso dobra a cada dois anos. É uma observação e projeção de uma tendência histórica. [5] Inúmeras inovações de cientistas e engenheiros sustentaram a lei de Moore desde o início da era do circuito integrado. A fabricação de dispositivos semicondutores e circuitos integrados, em que a contagem de transistores cresce em mais de sete ordens de magnitude em menos de cinco décadas, tem sido cada vez mais ampliada em escala e reduzida em seus custos, favorecendo o desenvolvimento de importantes tecnologias. Dessa forma, a humanidade tem hoje à disposição, dispositivos que portam um poder computacional enorme e viabilizam o treinamento de redes neurais complexas e bases de dados de grande porte.

1.4 Computação em Nuvem

Computação em nuvem é uma tecnologia que permite a um usuário o acesso remoto a recursos computacionais, permitindo com isso que recursos sejam alocados sob demanda e dispo-

nibilizando dispositivos poderosos. Desde muito tempo há a demanda de utilizar recursos computacionais remotamente, inicialmente isto se devia ao grande espaço requerido pelos computadores antigamente. Nos dias atuais, a demanda por computação em nuvem continua alta, com pesquisadores das mais diversas áreas que necessitam de recursos de hardware para seus estudos e não possuem vantagem em comprar dispositivos caros que podem ser alugados de maneira fácil.

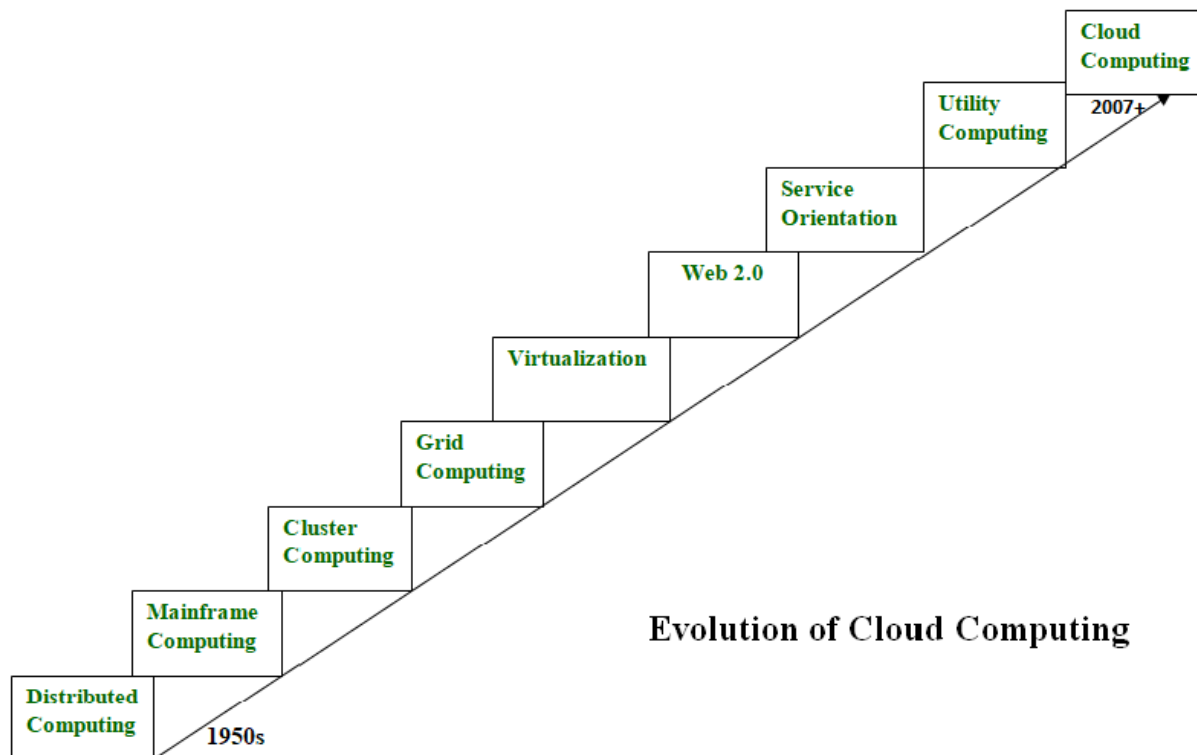


Figura 1.4: Evolução Cloud Computing - Geeksforgeeks

A evolução ao longo dos anos se deu por alguns pontos principais. A introdução dos sistemas distribuídos, uma composição de sistemas independentes com abstração para o usuário final que percebe os sistemas como apenas um. [6] Dessa forma, recursos computacionais são compartilhados com eficiência entre os sistemas, fornecendo grandes vantagens de processamento. Posteriormente, foi também introduzida a conexão dos sistemas independente de regiões geográficas distintas. Esta integração é feita pela internet, com uma conexão de alta banda é possível fazer um uso eficaz desta tecnologia.

A criação de ambientes virtuais em hardwares permite que múltiplas instâncias sejam executadas simultaneamente, potencializando o processamento do problema. Com a grande demanda por estes serviços, empresas ou usuários finais podem contratar recursos virtualmente e facilmente e ainda podem escolher vários provedores deste tipo de serviço.

1.5 Apresentação do Manuscrito

O restante deste relatório está organizado da seguinte forma. O Capítulo 2 apresenta as informações sobre a Base de Dados utilizada, aborda outras opções de base que satisfazem o mesmo problema. Finaliza abordando o tema de Ampliação de Dados como alternativa para melhorar a performance da rede.

A partir daí, é realizada no Capítulo 3 uma análise detalhada sobre as Redes Convolucionais Neurais. Métodos de Transferência de Aprendizado e Redes Pré Treinadas são detalhados. A utilização da ferramenta PyTorch é detalhada e os resultados obtidos serão comparados com *benchmarks* de estudos similares.

Capítulo 2

Princípios do Desenvolvimento

Neste capítulo é abordado o problema objetivado por este trabalho, a teoria por trás das Redes Neurais Convolucionais e suas técnicas, as informações da base de dados utilizada e a metodologia utilizada.

2.1 Redes Neurais Convolucionais

As Redes Neurais Convolucionais, quando utilizadas em problemas de imagens, são modelos de Inteligência Artificial que desenvolvem seus próprios mecanismos de detecção de linhas e formas da imagem. Uma rede neural bem treinada, com uma quantidade de camadas apropriada, reconhece automaticamente as características da imagem de forma hierárquica, começando por bordas e cantos simples até objetos e formas complexas. [7] São algoritmos que, quando treinados em um grande conjunto de dados, têm como saída para cada imagem de entrada, uma representação dela em forma de vetor que captura suas características. A rede é composta por camadas: na entrada está uma camada de convolução. A seguir, o desenvolvedor tem liberdade para determinar a quantidade de camadas utilizadas e as técnicas de regularização, que são posteriormente detalhadas, adotadas. No fim, um vetor é processado de modo a interpretar a predição da rede quanto a uma imagem na entrada. A figura abaixo mostra um exemplo de estrutura destas redes bem como sua entrada e a classificação na saída.

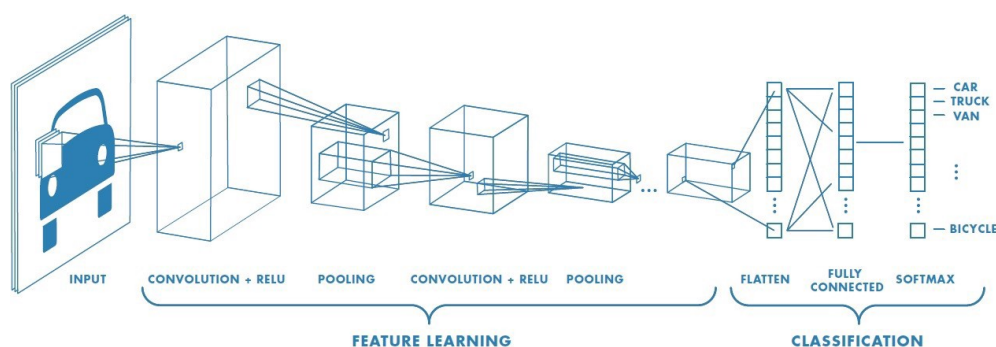


Figura 2.1: Exemplo de Estrutura do Problema

As camadas iniciais da rede são responsáveis pelo primeiro estágio de captura de características da imagem, como contornos, cores, orientação. Conforme o modelo agrega camadas sua

complexidade aumenta e possibilita a extração de outras características das imagens. A utilização da camada de *pooling* faz com que as saídas das camadas de convolução sejam reduzidas, fazendo com que a demanda de processamento computacional caia de forma significativa, favorecendo o treinamento. Esta camada varre a imagem em janelas de tamanhos fixados e armazena em uma matriz de menor tamanho do que a imagem, ou a média ou o máximo dos elementos contidos na janela durante cada iteração.

A escolha de utilização de mais ou menos camadas, com maior ou menor complexidade, é muito relativa ao problema estudado. A depender das imagens estudadas e das características que se deseja extrair destas imagens, a rede pode ter mais ou menos camadas, de maior ou menor complexidade. A saída destas camadas é então vetorizada e inserida na rede neural para o treinamento. Existem diversas técnicas intermediárias de regularização que evitam problemas de generalização da rede.

Sobre-ajuste (*Overfitting*) é um termo estatístico utilizado quando um modelo criado e treinado se ajusta ao conjunto de dados anteriormente alimentado de forma muito específica e acurada, no entanto, se mostra ineficaz para prever novos resultados quando alimentada com dados não pertencentes ao conjunto inicial. Um modelo sobre-ajustado apresenta alta precisão quando testado com seu conjunto de dados, porém tal modelo não apresenta boa generalização e por isso deve ser evitado. Algumas técnicas são utilizadas para evitar o sobre-ajuste da rede. O sobre-ajuste é uma evidência de uma rede com parâmetros demasiadamente complexos para a quantidade de dados disponível no treinamento.

Regularização é a técnica utilizada que evita o sobre-ajuste penalizando modelos complexos de redes neurais. Dessa forma, o sobre-ajuste é evitado fazendo com que os pesos da rede tenham suas magnitudes reduzidas. Com isso, a rede apresenta melhor desempenho em casos de alimentação de novos dados que não estavam presentes no conjunto de dado utilizado no treinamento. Regularização L1 adiciona uma penalidade L1 que é igual ao valor absoluto da magnitude do coeficiente da rede, ou então restringe o tamanho dos coeficientes. Regularização L2 adiciona uma penalidade L2 que é igual ao quadrado da magnitude dos coeficientes. A função de custo é uma medida do quanto a rede neural está diferindo dos resultados esperados. O resultado da rede depende dos cálculos com os parâmetros da rede de pesos. As equações abaixo descrevem as funções de custo das regularizações L1 e L2.

$$Custo = \sum_{i=0}^N (y_i - \sum_{j=0}^M x_{ij} \cdot W_j)^2 + \lambda \sum_{j=0}^M |W_j| \quad (2.1)$$

$$Custo = \sum_{i=0}^N (y_i - \sum_{j=0}^M x_{ij} \cdot W_j)^2 + \lambda \sum_{j=0}^M W_j^2 \quad (2.2)$$

Em ambas equações é calculada a diferença entre o resultado esperado e o resultado obtido. A principal diferença é o termo de penalidade de pesos complexos estar elevada ao quadrado na regularização L2, tornando ainda mais intensa a busca por redes mais simples. Outra técnica utilizada, o termo *Dropout* em inglês, que a cada iteração selecionada de modo pseudo aleatório alguns parâmetros internos da rede e os reiniciam em zero, favorecendo novamente redes mais

simples e promovendo a generalização da rede para novas alimentações de dados e contornando o sobre-ajuste.

A técnica de ampliação de dados, do inglês *Data Augmentation*, permite que, a partir de uma base de imagens com uma quantidade limitada de exemplos, seja obtida uma base replicada e expandida com imagens similares, mas diferentes das originais, dessa forma, a rede é treinada com mais exemplos rotulados, que são percebidas pela rede como imagens distintas. Esta técnica proporciona diversas transformações às imagens: rotações, que rotacionam a imagem em ângulos de 0 a 360 graus, recortes aleatórios de seções da imagem, que obtém regiões específicas das imagens que podem contar detalhes relevantes, etc. Redes Neurais Convolucionais exigem uma quantidade elevada de dados para treinamento, a técnica de aumento de dados pode proporcionar melhores resultados de treinamento da rede e conseqüentemente melhores acurácias nos testes de validação da rede. Ao mesmo tempo que se evita o sobre-ajuste, esta técnica de ampliação de dados favorece a generalização pela rede uma vez que permite pesos mais robustos na rede sem causar sobre-ajuste.

O trabalho desenvolvido utiliza de Redes Neurais Convolucionais e de um aprendizado profundo supervisionado, ou seja, aquele em que a rede treina com exemplos que possuem rótulos do resultado esperado pela rede e ela se adequa de modo a obter os resultados mais acurados. A rede deve, ao fim, baseado em um vetor de n elementos, onde n é a quantidade de classes do problema em questão, predizer qual a classe da imagem de entrada. Para isso, é feito o uso da função Softmax. A função Softmax é responsável por receber um vetor de entrada de i elementos (Z vetor) e os normalizar em valores proporcionais, de modo que a soma de todos os i elementos do vetor seja igual a 1. Estes valores podem ser positivos ou negativos, que quando normalizados para a soma dar 1, representam a probabilidade da imagem de entrada representar a classe do elemento i . A Softmax em seguida extrai a maior probabilidade deste vetor e retorna a predição da rede quanto à imagem de entrada.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (2.3)$$

A função Softmax é descrita pela equação acima onde a probabilidade do elemento Z_i ser da respectiva classe é a razão da exponencial natural deste elemento pelo somatório das exponenciais naturais de todos os elementos individualmente. Após este cálculo, basta varrer o vetor de probabilidades e encontrar a maior probabilidade dentre eles, será então a predição da rede.

O processo de treinamento da rede passa por ajustes dos pesos é chamado de “treinamento” da rede neural. No início, a Rede Neural Convolucional começa com pesos aleatórios. Durante o treinamento, os arquivos de imagens com seus respectivos rótulos, ou seja, com suas classes correspondentes, são dados de entrada à rede neural. [2] A rede neural convolucional processa cada imagem com seus parâmetros aleatórios inicialmente e, em seguida, compara sua saída com o rótulo correto da imagem. Se a saída da rede não corresponder ao rótulo, o que provavelmente é o caso no início do processo de treinamento, ela faz um pequeno ajuste nos pesos de seus neurônios para que na próxima vez que veja a mesma imagem, sua saída seja um pouco mais perto da resposta correta.

Este processo de treinamento nestas redes possui alto custo computacional e requer uma enorme quantidade de imagens para atingir bons níveis de acurácia. Felizmente, a técnica de Transferência de Aprendizado auxilia nesse quesito. Esta técnica permite obter o aprendizado de uma rede que foi treinada com uma base de dados com quantidades de imagens na ordem de milhões. Com isso, todo o longo processo de aprendizado desta base pela rede é realizado, ao fim do treinamento, os pesos de seus neurônios são "congelados" e reutilizados em um novo problema. De posse da base de dados do novo problema, basta obter estes pesos congelados e retrainar a rede mas, agora sob a base de dados do novo problema, dessa forma, diversas características da imagem já conseguem ser extraídas com sucesso, acelerando o tempo de atingimento de altos níveis de acurácia.

2.2 VGG-16

Fazendo uso da técnica de transferência de aprendizado, ao mesmo tempo que se poupa um enorme custo computacional, tem-se acesso a redes pré treinadas muito robustas. A VGG-16, por exemplo, é uma rede neural convolucional que possui 16 camadas profundas, pré treinada com 14 milhões de imagens da ImageNet, uma base de dados. A rede classifica as imagens da base em 22 mil diferentes classes de acordo com suas características.

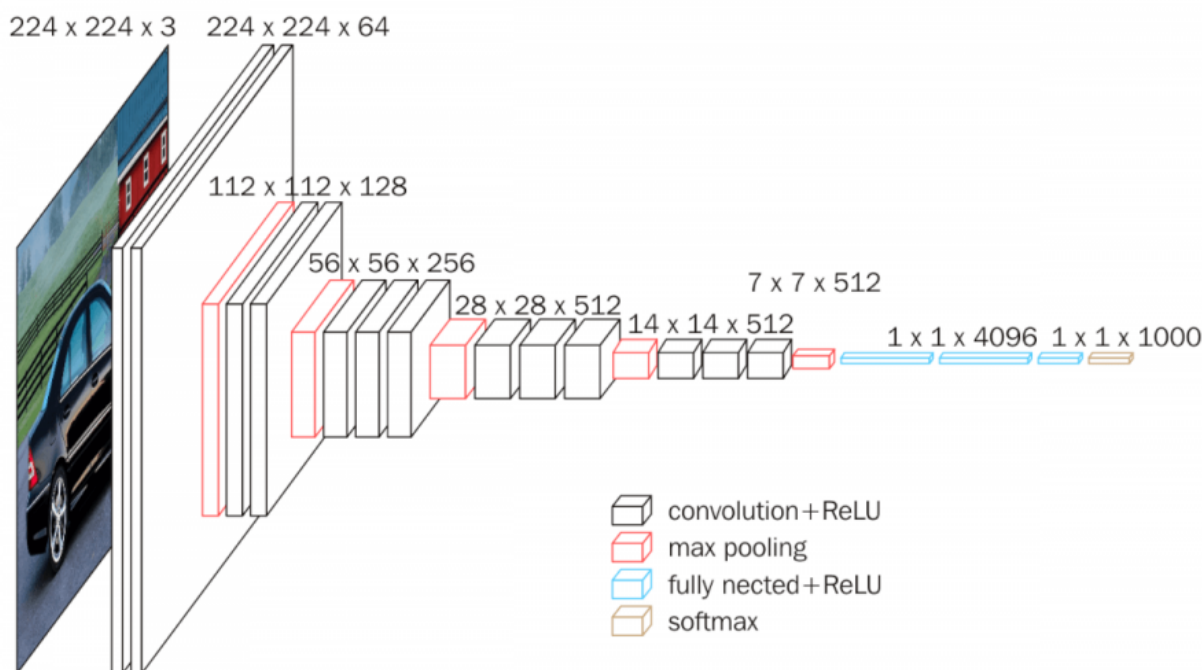


Figura 2.2: Estrutura da VGG-16

A VGG-16 atinge 92.7% de acurácia sob a base ImageNet. A rede recebe uma imagem de entrada de $224 \times 224 \times 3$, ou seja, 224 por 224 pixels em RGB. Primeiramente a imagem passa por uma camada de convolução, com função de ativação ReLU, a seguir, a rede passa por todas

as outras camadas, de *max pooling* que reduz a imagem a metade de sua proporção em pixels a cada camada. Outras camadas convolucionais extraem mais propriedades das imagens e ao fim a função Softmax realiza a predição.

PyTorch é uma biblioteca de aprendizado de máquina de código aberto baseada na biblioteca Torch. É utilizada em situações de visão computacional. [8] Uma série de softwares de Aprendizado Profundo são construídas com PyTorch, que oferece computação de tensores (como NumPy) com forte aceleração por meio de unidades de processamento gráfico (GPU). O PyTorch permite a importação da rede VGG 16 pré treinada e a construção da rede para o problema proposto.

2.3 Informações da Base de Dados

A base de dados utilizada foi a "IIT Delhi Touchless Palmprint Database (Version 1.0)". [1] A base de dados conta com 1301 imagens de palmas de mãos esquerdas de 230 indivíduos diferentes. Os indivíduos possuem idade entre 12 e 57 anos, homens e mulheres. As imagens possuem a proporção 1600x1200 pixel e é RGB. A base fornece também uma versão centralizada destas imagens que tem 150x150 pixels, em preto e branco, estas imagens estão em formato .bmp e não é apropriado para as redes neurais convolucionais aqui implementadas, uma vez que seu tamanho é menor do que o da entrada da rede e seu aumento causaria distorções na imagem. O acesso foi concedido por se tratar de um projeto de graduação.

A seguir, um exemplo de imagem da base de dados, as fotos são grandes, nítidas, coloridas e seus fundos são pretos, facilitando o reconhecimento da mão pela rede. [1] As mãos são de diversos tipos e tamanhos, de homens e mulheres, com ou sem anéis. Bastante diversa, mas com uma quantidade limitada de dados.



Figura 2.3: Exemplo de imagem do banco de dados

[1] A base escolhida possui 2601 imagens no total (1301 de mãos esquerdas e 1300 de direitas). As outras bases em geral possuem mais imagens, 2 ou 3 vezes maior do que a utilizada neste trabalho, no entanto, as bases de dados de [1] a [4] não estavam disponíveis para download e citação neste trabalho. Dessa forma a melhor base de dados disponível para o desafio proposto foi a de número [5].

Dataset	Total images	Population	Avg imgs/subj	Hand-side	Left-right	ID	Age	Skin color	Pixels	Gender	Additional details
11k Hands (ours)	11,076	190	58	Palm - dorsal	Both	Yes	Yes	Yes	1600×1200	Yes	Yes
CASIA [1]	5,502	312	8	Palm	Both	Yes	No	No	640×480	Yes	No
Bosphorus [2,3]	4,846	642 (276)	6 (3)	Palm	Both (left)	Yes	No	No	382× 525	No	No
[4]	4,000	200	20	Palm	Left	Yes	No	No	1382×1036	No	No
IITD [5]	2,601	230	11	Palm	Both	Yes	No	No	1600×1200	No	No
GPDS150hand [6]	1,500	150	10	Palm	Right	Yes	No	No	NA	NA	No

Figura 2.4: Comparação bases de dados

Acima é possível observar as comparações entre a base utilizada e outras bases de dados existentes. Possivelmente a utilização de outras bases de dados maiores do que a escolhida poderiam resultar em melhores performances da rede, no entanto, a base utilizada apresentou bons resultados na ordem de 75 a 80% e a utilização de técnicas de expansão da base de dados auxiliam a contornar a desvantagem de possuir menos quantidade de imagens. [9]

2.4 Espaço de Desenvolvimento Integrado

Este trabalho foi desenvolvido fazendo uso de computação em nuvem e dados na nuvem. Isto foi possível por meio das ferramentas Google Colab Pro+. A assinatura deste serviço disponibiliza ao usuário o acesso remoto à GPUs K80, T4 e P100, e possibilita o tempo de execução de rotinas de até 24h, oficialmente. No entanto, se treinamentos superiores a este tempo for necessários é possível simplesmente utilizar uma trava de execução que ele rodará até alguma outra interrupção.

A base de dados foi salva na plataforma do Google Drive. Dessa forma, o acesso ao banco de dados pelo IDE (Ambiente de Desenvolvimento Integrado) do Google Colab é muito facilitado. A seguir é detalhado o procedimento utilizado e os resultados obtidos com os testes.

[10]

Capítulo 3

Procedimentos e Resultados

Este capítulo aborda as técnicas utilizadas na implementação da rede final do projeto.

3.1 Implementação

A rede VGG-16 foi escolhida para o desenvolvimento deste trabalho. As redes pré treinadas mais utilizadas são a VGG, ResNet, EfficientNet e GoogleNet. A VGG 16 utilizada possui 16 camadas profundas, é um modelo de rede neural convolucional proposto por K. Simonyan e A. Zisserman da Universidade de Oxford. [11] O modelo atinge 92,7 de acurácia em um conjunto de dados de mais de 14 milhões de imagens pertencentes à 22 mil classes.

A base de dados deste trabalho foi salva no Google Drive. Por meio do Google Colab, foi feito o acesso ao drive e conseqüentemente às imagens da base. A importação de bibliotecas essenciais é feita logo no início do código. PyTorch, PILLOW, Torchvision são importados. Vale lembrar que o código desenvolvido neste projeto está disponível no fim deste documento. Algumas outras bibliotecas são necessárias: Pandas para tratar a abertura de arquivos csv, e Numpy para tratar operações numéricas.

É criada a classe em Python que representa a base de dados com seus atributos. A seguir, Torchvision é utilizado para aplicar transformações nas imagens. A rede converte as imagens para o tamanho de 256 pixels apropriados. A função ToTensor transforma as variáveis para tensores e as imagens são normalizadas.

Uma vez estabelecidas as 230 classes presentes nas imagens da base de dados e os parâmetros de melhores resultados, a rede de melhor desempenho foi configurada com uma taxa de aprendizado de $3E-4$, tamanho do *batch* de 32 e 200 épocas. A base de dados é então importada como instância da classe previamente criada. Funções do Torch.utils separam a base de dados em treinamento e teste aleatoriamente por meio da função `data.randomsplit()`. [8] Foram separadas 1000 imagens para treinamento e 300 imagens para teste. Em seqüência, é instanciado o modelo classificador, por meio da função `Sequential` do Torch.nn, instancia-se a rede VGG-16 pré treinada, com 4096 características de entrada e 256 de saída. Instancia-se a função de ativação ReLU, a seguir, com os 256 de saída anterior agora como entrada, saem 230 classes de saída. Ao fim, a função Softmax realiza sua parte de representar a predição da rede. A cada época o algoritmo

mostra a função de perda do modelo onde é possível se ter noção do aprendizado da rede naquele instante de tempo do treinamento.

```
30it [03:43, 7.26s/it]
31it [03:50, 7.25s/it]
32it [03:52, 7.25s/it]
 1% |          | 1/120 [03:52<7:40:13, 232.04s/it]It's epoch 0 and the loss is 5.49296860396862
0it [00:00, ?it/s]
1it [00:01, 1.79s/it]
```

Figura 3.1: Acompanhamento de resultados a cada época

Acima é possível observar o acompanhamento da rede por meio da saída no console. Para cada época é determinado o tempo de cada iteração e época, bem como o valor da função de custo e valor de acurácia ao fim do treinamento. Com um número de épocas configurado para 120, com cada época levando aproximadamente 4 minutos, o tempo de execução do algoritmo gira em torno de 7 a 8 horas. Ao longo do treinamento é possível validar a eficácia do treinamento ao se observar a função de perda decaindo conforme as épocas passam.

O acesso ao Código Fonte se dá pelo endereço a seguir: <https://colab.research.google.com/drive/19gcm0G896EBxOCQ4SiLWblSwg6O8e9T?usp=sharing>

3.2 Resultados

Diversos tipos de configurações de redes foram testados. Variações entre o tipo de técnica da Ampliação de Dados, variações na taxa de aprendizado e no número de épocas bem como os valores de normalização das imagens. Os resultados obtidos são resumidos na tabela abaixo que mostra resultados da função de erro e da acurácia no teste.

O melhor resultado foi também o de maior número de épocas, no entanto, a relação acurácia/número de épocas com a configuração de seus parâmetros de rede apresentou o melhor custo benefício almejando a acurácia final. O melhor desempenho veio da tentativa 9. Possivelmente pelo elevado número de épocas e a taxa de aprendizado reduzida. A tentativa 9 não fez uso de Ampliação de Dados, teve taxa de aprendizado de $7E-4$ e 80 épocas. A tentativa final foi feita baseada em uma rede similar, com taxa de aprendizado semelhante e um número ainda maior de épocas na expectativa de obter melhores resultados no teste quando comparado a outros estudos relacionados comentados previamente.

A rede final obteve na última época uma acurácia no treinamento de 99 e uma acurácia no teste de 80. Foi possível testar as diversas técnicas aplicáveis ao problema e avaliar quais implicaram melhora de performance. O resultado de 80 de acurácia obtido é satisfatório quando comparado às redes dos estudos mencionados e mostra a viabilidade de aplicação desta rede na autenticação biométrica.

Tentativa	Acurácia %	Número de Épocas	Taxa de Aprendizado
15	79,3	200	3e-4
9	77,3	80	7e-4
14	77,0	120	5e-4
11	74,7	14	1e+3
10	74,0	40	9e+3
3	71,0	20	1e+3
7	70,3	40	9e+2
12	68,0	15	1e+3
8	67,0	10	9e+2
1	66,7	10	9e+2
2	65,0	10	6e+2
13	62,3	15	1e+3
4	62,3	10	1e+3
6	51,0	30	9e+2
5	35,3	10	1e+3

Tabela 3.1: Tentativas e Resultados

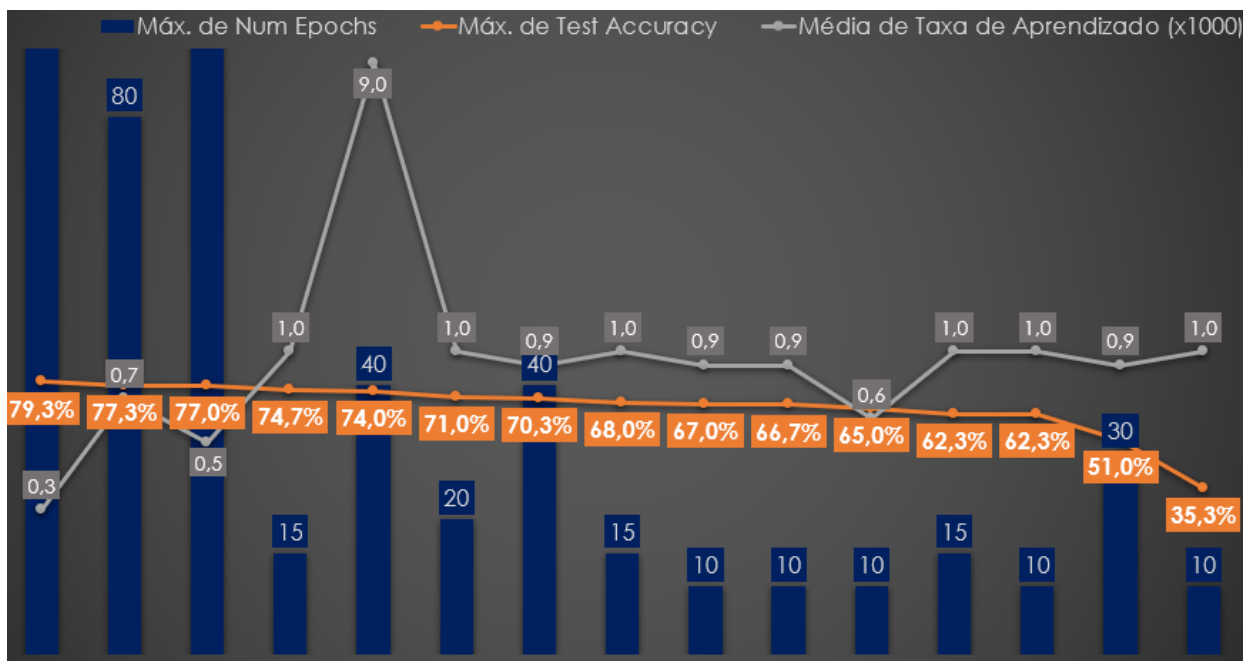


Figura 3.2: Tentativas e Acurácias - Gráfico

```

32it [00:54, 1.69s/it]
100%|██████████| 120/120 [1:50:42<00:00, 55.36s/it]
It's epoch 119 and the loss is 1.1488600820302963
Train accuracy: 0.712
Test accuracy: 0.47

```

Figura 3.3: Exemplo de resultado final da rede

Acima está novamente o acompanhamento da rede, que em seu estágio final de treinamento e teste apresenta este tipo de saída no console, com os valores da função de custo, da acurácia de treinamento e da acurácia de teste bem definidos ao fim da execução.

3.3 Comparações/Benchmarks

Os benchmarks para o desafio deste trabalho são estabelecidos por redes importantes como AlexNet, GoogleNet, VGG16, VGG19. A AlexNet obteve resultados na base de palmas de 0,839 de acurácia no momento do teste. VGG 16, a mesma que utilizada neste trabalho possuiu acurácia de 0,862. A GoogleNet obteve resultados ligeiramente melhores que a VGG16.

GoogleNet (palmar)	Results	0.863	0.894	0.851	0.867	0.835	0.886	0.907	0.918	0.844	0.925	0.879
	Model**	G_p_1	G_p_2	G_p_3	G_p_4	G_p_5	G_p_6	G_p_7	G_p_8	G_p_9	G_p_10	-
GoogleNet (dorsal)	Results	0.878	0.845	0.876	0.908	0.914	0.859	0.908	0.914	0.859	0.932	0.8893
	Model**	G_d_1	G_d_2	G_d_3	G_d_4	G_d_5	G_d_6	G_d_7	G_d_8	G_d_9	G_d_10	-
Ours (palmar)	Results	0.853	0.828	0.866	0.885	0.885	0.885	0.891	0.860	0.899	0.885	0.874
	Model**	O_p_1	O_p_2	O_p_3	O_p_4	O_p_5	O_p_6	O_p_7	O_p_8	O_p_9	O_p_10	-
Ours (dorsal)	Results	0.940	0.947	0.906	0.953	0.909	0.900	0.900	0.908	0.876	0.860	0.910
	Model**	O_d_1	O_d_2	O_d_3	O_d_4	O_d_5	O_d_6	O_d_7	O_d_8	O_d_9	O_d_10	-

Figura 3.4: Benchmarks GoogleNet e Autorial 11kHands

Os resultados obtidos neste trabalho não estão longe dos resultados obtidos em outros estudos do mesmo desafio, estando a menos de 10% de diferença de acurácia dos melhores resultados. Possivelmente a adição de recursos computacionais ou maior tempo de desenvolvimento da rede façam com que seja alcançada a mesma taxa de acurácia. Após todos os testes uma rede final foi testada. Uma baixa taxa de aprendizagem com um grande número de épocas indicou resultar em boas acurácias, foi estabelecido uma taxa de aprendizagem de 3E-4 com 200 épocas. A rede foi capaz de alcançar uma acurácia no teste de 80 %, menos de 10 pontos percentuais atrás de benchmarks excelentes relacionados. [1]

Capítulo 4

Conclusões

O detalhamento das ferramentas utilizadas bem como a explicação das técnicas adotadas já foram discutidas, os resultados obtidos foram mostrados e discutidos. Este capítulo aborda as conclusões do trabalho, as ideias para prosseguimento do desafio e outros fatores que poderiam ser alterados.

4.1 Uso de Outras Redes Pré Treinadas

Como discutido, há uma ampla variedade de redes pré treinadas em base de dados de grande porte que possibilitam o alcance de grandes resultados com menos custo computacional do que começar o treinamento do zero. A autenticação por reconhecimento das linhas das palmas das mãos é uma maneira válida. Além disso, é interessante notar como nossas linhas das palmas das mãos são suficientes para nos identificar e são únicas.

O uso da VGG 16 neste trabalho se mostrou uma boa escolha ao se obter resultados satisfatórios para o problema proposto. Este trabalho foi desenvolvido em Python, no entanto, alguns outros estudos relacionados também fazem uso do software MatLab que oferece diversas ferramentas que auxiliam na construção de Redes Neurais Convolucionais dentro de seu ambiente. A utilização de outras redes pré treinadas com certeza acarretaria mudanças nas acurácias obtidas com um determinado número de épocas, podendo atingir resultados ainda melhores.

4.2 Tamanho da Base e Pré Processamento

A base utilizada neste trabalho não é de grande porte. Como discutido, a base possui 2601 imagens de palmas de mãos de 230 indivíduos diferentes, um número insuficiente de dados mas que com o uso de redes pré treinadas e algumas técnicas de Ampliação de Dados foi possível contornar o problema e obter boas performances da rede. O acesso às outras bases mencionadas não foi autorizado, dessa forma não foi possível avaliar o desempenho do modelo na classificação de outras bases de dados que possuem um volume maior do que a base utilizada.

O pré processamento das imagens também possui grande impacto nos resultados finais de performance. Esta etapa do desenvolvimento é a que normalmente consome a maior quantidade de tempo e de recursos computacionais. Pré processamentos utilizados mais comuns são transfor-

mações morfológicas da imagem envolvendo tamanho e forma. Normalização da imagem ou filtros por limites são utilizados na tentativa de evoluir os resultados da rede.

4.3 Trabalhos Futuros

Os resultados obtidos neste trabalho, embora satisfatórios, podem ser melhorados. A utilização de outras bases de dados que possuam as fotos das impressões de palma das mãos obtidas de maneira mais nítida podem favorecer uma melhor performance da rede. A utilização de outras redes pre treinadas como a GoogleNet ou a ResNet também podem apresentar resultados interessantes neste problema. As configurações possíveis para esta rede são infinitas e podem apresentar resultados dos mais diversos. A busca por configurações que atinjam melhor acurácia exige meses de teste e muito poder computacional, ficando como plano para o futuro.

4.4 Conclusão

O problema proposto vem sendo estudado por pesquisadores há anos e o fruto da boa performance destas redes já estão sendo colhidos. De autenticações biométricas à busca por criminosos, as Redes Neurais Convolucionais têm seu espaço estabelecido na resolução destes desafios trazendo excelentes resultados.

Os resultados obtidos neste trabalho estão na mesma ordem de grandeza de importantes benchmarks da área, comprovando que as linhas das palmas das mãos carregam em si uma identificação única de cada ser humano, restando à inteligência artificial classificar estes padrões que ao olho humano não são suficientes para uma identificação.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] M. Afifi, “11k hands,” <https://sites.google.com/>, 2019.
- [2] L. Pascu, “Q5id chooses redrock biometrics’ palm print and vein biometrics for enrollment, authentication,” <https://www.biometricupdate.com/>, 2020.
- [3] P. Waldron, “Catching criminals red-palmed,” *Biometrics CSE MSU*, 2013.
- [4] S. K. Dewan, “Elementary watson: Scan a palm find a clue,” *New York Times*, 2003.
- [5] J. Feng, “palmprint,” <https://www.researchgate.net/>, 2009.
- [6] “How do i use google colab for cloud computation?” *NC STATE University Libraries*, 2020.
- [7] B. Dickson, “What are convolutional neural networks (cnn),” <https://bdtechtalks.com/>, 2020.
- [8] “Vgg16 – convolutional network for classification and detection,” <https://neurohive.io>, 2018.
- [9] S. S. Ajay Kumar, “Personal identification using rank-level fusion,” *IEEE Trans.*
- [10] A. Gandhi, “Data augmentation | how to use deep learning when you have limited data,” <https://nanonets.com/>, 2021.
- [11] O. G. Yalçın, “4 pre-trained cnn models to use for computer vision with transfer learning,” <https://towardsdatascience.com/>, 2020.