**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO**
**CENTRO TECNOLÓGICO**
**DEPARTAMENTO DE ENGENHARIA ELÉTRICA**
**PROJETO DE GRADUAÇÃO**

**JUNIA GABRIELA LUBE PIMENTEL**

# CLASSIFICAÇÃO DE ATLETAS DA MODALIDADE MEN'S PHYSIQUE PELA APLICAÇÃO DE REDES NEURAIS PROFUNDAS EM IMAGENS DIGITAIS

VITÓRIA
2019

JUNIA GABRELA LUBE PIMENTEL

# CLASSIFICAÇÃO DE ATLETAS DA MODALIDADE MEN'S PHYSIQUE PELA APLICAÇÃO DE REDES NEURAIS PROFUNDAS EM IMAGENS DIGITAIS

Parte manuscrita do Projeto de Graduação da aluna **Junia Gabriela Lube Pimentel**, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Orientador: Prof. Dr. Jorge Leonid Aching Samatelo
Coorientador: Prof. Dr. Klaus Fabian Coco

VITÓRIA
2019

Junia Gabriela Lube Pimentel

# CLASSIFICAÇÃO DE ATLETAS DA MODALIDADE MEN'S PHYSIQUE PELA APLICAÇÃO DE REDES NEURAIS PROFUNDAS EM IMAGENS DIGITAIS

Parte manuscrita do Projeto de Graduação da aluna Junia Gabriela Lube Pimentel, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheira Eletricista.

Aprovado em 6 de dezembro de 2019.

**COMISSÃO EXAMINADORA:**

**Prof. Dr. Jorge Leonid Aching Samatelo**
Universidade Federal do Espírito Santo
Orientador

**Prof. Dr. Klaus Fabian Coco**
Universidade Federal do Espírito Santo
Coorientador

**Prof. Dr. Patrick Marques Ciarelli**
Universidade Federal do Espírito Santo
Examinador

**Ms. Clebeson Canuto dos Santos**
Universidade Federal do Espírito Santo
Examinador

Vitória-ES

2019

*Aos meus amigos e familiares por todo apoio e principalmente pelo amor.*

# AGRADECIMENTOS

Após 5 anos de muito esforço e dedicação é com imensa felicidade que escrevo esses agradecimentos. Assim, eu gostaria de agradecer a todas essas pessoas especiais que apareceram (e estão) na minha vida e que permitem que eu continue com os meus estudos. A todos vocês, minha eterna gratidão.

Aos meus pais, Roberto Pimentel e Carla Pimentel, pelo apoio e pela paciência desde o início dos meus estudos. Vocês sempre serão o meu maior exemplo. Ao meu irmão Roberto Junior e sua esposa Karina Costa, que sempre me apoiaram e demonstraram admiração pelo meu esforço.

Ao meu namorado, Rodolfo Picoreti, agradeço por toda compreensão e companheirismo nesses longos finais de semana de estudos. Sem você essa reta final teria sido muito mais difícil.

A todos os meus companheiros de graduação, mas principalmente Lucas Malacarne, Sofía Simonetti e Gustavo Glasgio. Obrigada pela amizade, pelas risadas e principalmente pela ajuda nos momentos de maior dificuldade. Sempre guardarei vocês em meu coração.

Ao meu orientador Jorge e coorientador Klaus por despertarem meu interesse por esse tema fascinante e por toda ajuda, orientação e dedicação durante o desenvolvimento deste trabalho. Obrigada pelos artigos enviados nas noites de sábado, pelas conversas, por me acalmarem nos dias de desespero e principalmente por acreditarem no meu potencial. Muito obrigada!

À IFBB - Espírito Santo por terem apoiado o meu trabalho desde o início e autorizado o uso da imagem dos atletas.

Ao GPU *workstation* Cheetara por sua velocidade.

À banca examinadora pela aceitação do convite e pelo tempo investido para leitura e avaliação desse trabalho.

Agradeço à Universidade Federal do Espirito Santo pela minha formação.

# RESUMO

É notório que nos últimos anos o fisiculturismo ganhou maior visibilidade no cenário esportivo nacional e mundial, haja vista que de forma inédita, a modalidade esteve presente nos Jogos Pan-Americanos de 2019. No Brasil, atualmente, as maiores competições de fisiculturismo são realizadas pela Confederação Brasileira de Musculação, Fisiculturismo e *Fitness* (CBMFF ou IFBB Brasil, filiada à IFBB - *International Federation of Bodybuilding and Fitness*). Além disso, novas modalidades competitivas foram criadas pela IFBB a fim de englobar as diferentes constituições musculares entre os atletas do esporte, tornando a competição mais justa, como no caso das modalidades *Men's Physique* e *Muscular Men's Physique*. Entretanto, por melhor que seja a qualificação dos árbitros, existe uma grande dificuldade em diferenciar os competidores dessas duas modalidades citadas, uma vez que o limiar que separa o *Men's Physique* de um *Muscular Men's Physique* não é claramente definido. Diante disso, visando contribuir para solucionar essa problemática de classificação binária, e atender a uma demanda existente, este trabalho faz uso de abordagem baseada em inteligência de máquina para buscar classificar os indivíduos dessas duas modalidades. Mais especificamente, este estudo faz uso de uma Rede Neural Convolucional (CNN, do inglês *Convolutional Neural Network*) para separar, por características antropométricas, os atletas das duas modalidades a partir de fotografias digitais. Para isso, utiliza-se de um banco de dados de imagens dos atletas com anotações dos árbitros da IFBB-ES (Unidade da IFBB no Estado do Espírito Santo). Esse estudo abordou técnicas de Aprendizado Profundo (DL, do inglês *Deep Learning*), como a transferência de aprendizagem (do inglês, *transfer learning*) e *fine tuning* para desenvolver um preditor de classes binário. Além disso, foram abordados métodos para que seja possível lidar com pequenos bancos de dados com distribuição de classes desbalanceadas.

**Palavras-chave**: *Deep learning*. Fisiculturismo. *Men's Physique*. *Muscular Men's Physique*. Redes neurais convolucionais. *Transfer learning*.

# ABSTRACT

It is notorious that in recent years bodybuilding has gained greater visibility in the national and world sports scene, given that in a new way, the sport was present at the Pan American Games of 2019. In Brazil, currently, the largest bodybuilding competitions are held by Brazilian Confederation of Bodybuilding, Bodybuilding and Fitness (CBMFF or IFBB Brasil, affiliated to IFBB - International Federation of Bodybuilding and Fitness). In addition, new competitive modalities were created by IFBB to encompass the different muscle constitutions among sport athletes, making the competition fairer, as in the case of Men's Physique and Muscle Men's Physique modalities. However, no matter how good the referees qualify, there is great difficulty in differentiating competitors from these two modalities, as the threshold that separates Men's Physique from a Muscular Men's Physique is not clearly defined. Therefore, aiming to contribute to solve this binary classification problem, and to meet an existing demand, this work used a machine intelligence based approach to try to classify individuals from these two modalities. More specifically, this study used a Convolutional Neural Network (CNN) to separate, by anthropometric characteristics, athletes from both modalities from digital photographs. This was done by using a database of athlete images with notes from the IFBB-ES referees (IFBB Unit in the State of Espírito Santo). This study addressed Deep Learning - DL techniques, such as transfer learning and fine-tuning to develop a binary class predictor. In addition, methods were discussed to deal with small databases with unbalanced class distribution.

**Keywords**: Deep Learning. Bodybuiding. Men's Physique. Muscular Men's Physique. Convolutional Neural Network. Transfer learning.

# LISTA DE FIGURAS


Figura 1 – Atletas das modalidades foco do estudo, sendo: (a) Atleta *Men's Physique* e (b) Atleta *Muscular Men's Physique* . . . . . 15
Figura 2 – Modelo matemático do neurônio . . . . . 19
Figura 3 – Exemplo básico de estrutura de RNA . . . . . 20
Figura 4 – Arquitetura da VGG16 . . . . . 23
Figura 5 – Arquitetura da VGG19 . . . . . 24
Figura 6 – Módulo Inception V1 . . . . . 25
Figura 7 – Conexão de Salto . . . . . 26
Figura 8 – Arquitetura básica de uma DenseNet . . . . . 27
Figura 9 – Estimativa de pose de atleta *Men's Physique* . . . . . 28
Figura 10 – Silhuetas base de atletas do fisiculturismo . . . . . 30
Figura 11 – Comparativo entre imagem com esqueleto desenhado e resultado do algoritmo de *bounding box*, no qual (a) é a imagem com esqueleto desenhado e (b) é a *Bounding Box* . . . . . 33
Figura 12 – Base de dados *Men's Physique* . . . . . 36
Figura 13 – Base de dados *Muscular Men's Physique* . . . . . 37
Figura 14 – Exemplo de aplicação das operações utilizadas do Augmentor . . . . . 39
Figura A1 – Processo de aprendizagem . . . . . 58
Figura A2 – Exemplo de modelo de validação cruzada com *K-Fold* . . . . . 65
Figura A3 – Exemplos de imagens de cada aquisição . . . . . 67

# LISTA DE GRÁFICOS


Gráfico 1 – Curva ROC para $K = 3$ . . . . . 42

Gráfico 2 – Curva P-S . . . . . 43

Gráfico 3 – Curva ROC - Melhor resultado utilizando DenseNet121 como modelo base 45

Gráfico 4 – Curva P-S - Melhor resultado utilizando DenseNet121 como modelo base . 46

Gráfico 5 – Curva ROC - Resultado de experimento com a VGG19, usando o conjunto de dados completo - três camadas densas de 256 neurônios e duas de *dropout* 47

Gráfico 6 – Curva P-S - Resultado de experimento com a VGG19, usando o conjunto de dados completo - três camadas densas de 256 neurônios e duas de *dropout* . 48

Gráfico 7 – Curva ROC - Resultado de experimento com VGG19, usando o conjunto de dados sem *outliers* - três camada densas de 256 neurônios e duas de *dropout* (0,3), (0,15) . . . . . 48

Gráfico 8 – Curva P-S - Resultado de experimento com VGG19, usando o conjunto de dados sem *outliers* - três camada densas de 256 neurônios e duas de *dropout* (0,3), (0,15) . . . . . 49

Gráfico 9 – Curva ROC - Resultado de experimento com VGG16 - Três camadas densas de (256, 128, 64) neurônios e duas de *dropout* (0,5) . . . . . 50

Gráfico 10 – Curva P-S - Resultado de experimento com VGG16 - Três camadas densas de (256, 128, 64) neurônios e duas de *dropout* (0,5) . . . . . 50

Gráfico 11 – Curva ROC - Resultado de experimento com VGG16 - Quatro camadas densas de (512, 256, 128, 64) neurônios e três de *Dropout* $1 \times (0,6)$, $2 \times (0,5)$ 51

Gráfico 12 – Curva P-S - Resultado de experimento com VGG16 - Quatro camadas densas de (512, 256, 128, 64) neurônios e três de *Dropout* $1 \times (0,6)$, $2 \times (0,5)$ . . 51

# LISTA DE QUADROS

Quadro 1 – Melhores resultados obtidos nos experimentos com redes densas . . . . . 45
Quadro 2 – Melhores resultados obtidos com VGG19 . . . . . 49
Quadro 3 – Melhores resultados obtidos com VGG16 . . . . . 52

# LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| AUC | *Area Under the Curve* |
| CNN | *Convolutional Neural Networks* |
| DL | *Deep Learning* |
| FN | Falso Negativo |
| FP | Falso Positivo |
| GPU | *Graphics Processing Unit* |
| IFBB | *International Federation of Bodybuilding* |
| ILSVRC | *Imagenet Large Scale Visual Recognition Challenge* |
| ML | *Machine Learning* |
| RNA | Redes Neurais Artificiais |
| ROC | *Receiver Operating Characteristic* |
| SGD | *Stochastic Gradient Descent* |
| UFES | Universidade Federal do Espírito Santo |
| VN | Verdadeiro Negativo |
| VP | Verdadeiro Positivo |

# SUMÁRIO

**1 INTRODUÇÃO** ..... **13**
**1.1 Apresentação e Objeto de Pesquisa** ..... **13**
**1.2 Trabalhos Relacionados** ..... **16**
**1.3 Objetivos** ..... **17**
**1.4 Estrutura do Texto** ..... **17**
**2 EMBASAMENTO TEÓRICO** ..... **18**
**2.1 Introdução** ..... **18**
**2.2 Redes Neurais Artificiais** ..... **18**
**2.3 *Convolucional Neural Networks*** ..... **20**
**2.4 Modelos Pré-Treinados** ..... **22**
2.4.1 VGG16 e VGG19 ..... 22
2.4.2 Inception ..... 24
2.4.3 ResNets ..... 25
2.4.4 DenseNet ..... 26
**2.5 *Human Pose Estimation* - Estimativa de Pose Humana** ..... **27**
2.5.1 *OpenPose* ..... 27
**2.6 Fisiculturismo** ..... **28**
2.6.1 *Men's physique* ..... 29
2.6.2 *Muscular men's physique* ..... 29
**3 SOLUÇÃO PROPOSTA** ..... **31**
**3.1 Introdução** ..... **31**
**3.2 Arquitetura do Modelo** ..... **31**
3.2.1 Registro das imagens ..... 31
3.2.2 Etapa de classificação ..... 32
**4 RESULTADOS** ..... **35**
**4.1 Introdução** ..... **35**
**4.2 Conjunto de Dados** ..... **35**
**4.3 Recursos Computacionais** ..... **35**
**4.4 Detalhes de Implementação** ..... **37**
**4.5 Métricas de Avaliação** ..... **39**
**4.6 Curvas de Avaliação** ..... **41**
4.6.1 Curva ROC ..... 41

4.6.2 Curva P-S.....42
**4.7 Experimentos.....43**
4.7.1 Protocolo de avaliação.....44
4.7.2 Primeiro experimento.....44
4.7.3 Segundo experimento.....46
4.7.3.1 Usando o modelo pré-treinado VGG19.....46
4.7.3.2 Usando o modelo pré-treinado VGG16.....49
**5 CONCLUSÕES E PROJETOS FUTUROS.....53**
**5.1 Conclusões.....53**
**5.2 Temas a Serem Pesquisados.....53**
**REFERÊNCIAS BIBLIOGRÁFICAS.....55**
**APÊNDICE A – INTRODUÇÃO AO APRENDIZADO DE MÁQUINAS.....58**
**APÊNDICE B – VALIDAÇÃO CRUZADA – K-FOLD.....64**
**APÊNDICE C – AQUISIÇÃO DAS IMAGENS DO CONJUNTO DE DADOS.....66**
**ANEXO A – AUTORIZAÇÃO DO USO DAS IMAGENS DOS ATLELTAS DA IFBB-ES.....68**

# 1 INTRODUÇÃO

## 1.1 Apresentação e Objeto de Pesquisa

A busca por hábitos saudáveis e o culto ao corpo são características intrínsecas da sociedade contemporânea, todavia, o conhecimento acerca da relação direta entre desenvolvimento físico e saúde está evidenciado desde o século XX (SCHWARZENEGGER; DOBBINS, 1998). Além disso, sabe-se de antemão que o treinamento de peso é um grande aliado na busca por definição e crescimento muscular. Dessa forma, nota-se um crescimento exacerbado do mercado *fitness*, tanto no ramo de alimentação, com nutricionistas e a venda de alimentos saudáveis, quanto no ramo da atividade física, com academias e centros de treinamento. A eclosão do número de academias é tanto que, no Brasil, no ano de 2016 já passava de 32.000 unidades, ficando apenas atrás dos Estados Unidos (BERTÃO, 2016). Assim, em conjunto com a ascensão da busca pela perfeição corpórea, novas vertentes de treinamento surgiram e práticas esportivas já conhecidas ganharam maior notoriedade, como é o caso do fisiculturismo. Ao fazer uma breve alusão histórica acerca do comportamento humano, nota-se que o culto ao corpo sempre esteve presente em sua idiossincrasia. No século XIX, por exemplo, o retorno do ideal de beleza grego fez com que um novo interesse pelo fisiculturismo surgisse, ultrapassando a ideia de que a funcionalidade muscular estava restrita aos métodos de autodefesa e sobrevivência. O modo de vida boêmio do europeu da época já não se encaixava aos ideais fisiculturistas de saúde geral e condicionamento físico (SCHWARZENEGGER; DOBBINS, 1998).

Com o passar dos anos, essa modalidade esportiva foi se modernizando, tanto nas metodologias de treinamento de força, quanto na análise de composição corporal. Já existe na literatura, por exemplo, vertentes que utilizam conjuntos de silhuetas específicos dos praticantes do fisiculturismo para avaliação perspectiva de imagem corporal (CASTRO et al., 2011). Além disso, novas modalidades competitivas foram criadas pela Federação Internacional de *Bodybuilding* e *Fitness* (IFBB, do inglês *International Federation of Bodybuilding and Fitness*) a fim de englobar as diversidades musculares entre os atletas do esporte e tornar a competição mais justa, como é o caso da *Men's Physique* e *Muscular Men's Physique*.

A princípio, a regra definia que um atleta do *Men's Physique* não pode possuir volume muscular que o caracterize como um culturista. Entretanto, muitos atletas se apresentavam com volume muscular acima do padrão da modalidade, mas ainda não se caracterizavam como culturistas.

A IFBB tentou corrigir o problema definindo uma nova modalidade de *Men‘s Physique* com maior volume, a *Muscular Men‘s Physique*, entretanto o fato apenas redirecionou o problema agora em definir qual limiar de volume muscular deve ser utilizado para separar os atletas entre as duas modalidades *Men‘s Physique* e *Muscular Men‘s Physique*. Nesse sentido a premissa básica se estabelece na regra de que um *Men‘s Physique* não pode possuir volume muscular que o defina como um culturista, mas o atleta da *Muscular Men‘s Physique*, por outro lado, deve ter um volume mais expressivo, levando-o a se comparar a um culturista clássico (o volume muscular do atleta nesta modalidade é restrito a uma relação de peso e altura).

A questão que pode surgir dessa classificação é: Por que não utilizar relações como peso e altura para a classificação desses atletas? A resposta, entretanto, esbarra no fato de que apenas o peso e a altura não são suficientes para essa classificação, havendo muito mais da conformação antropomórfica envolvida no processo, representada de forma mais adequada nas linhas e contornos da morfologia do tronco bem como em suas proporções de massa muscular associadas à estrutura óssea do atleta.

Devido à falta de um limiar determinado de separação entre as modalidades de fisiculturismo, muitos atletas competem em categorias que não estão de acordo com sua morfologia. Além disso, o julgamento da arbitragem é feito apenas a partir de análises antropomórficas, o que evidencia a dificuldade em efetuar a classificação visual entre os competidores das duas modalidades.

O trabalho em questão consiste no estudo e na aplicação de métodos automatizados de análise de dados, como Aprendizado de Máquina (ML, do inglês *Machine Learning*) e Aprendizado Profundo (DL, do inglês *Deep Learning*), para construção de um modelo analítico que faça a separação antropométrica entre os atletas das modalidades *Men’s Physique* e *Muscular Men’s Physique*. Diante disso, visa-se o desenvolvimento de um classificador binário a partir de uma inteligência de máquina que permita reconhecer os pormenores antropomórficos presentes nas imagens dos atletas permitindo uma classificação automática entre as duas modalidades. Na Figura 1 pode-se observar atletas das modalidades foco do estudo.

Atualmente o universo acadêmico depara-se com diversas ferramentas relacionadas à área de pesquisa do *Machine Learning* que apresentam grande eficácia nas resoluções de problemas de classificação. Dentre elas, encontra-se o *Deep Learning* que possibilita que uma máquina

Figura 1 – Atletas das modalidades foco do estudo, sendo: (a) Atleta *Men's Physique* e (b) Atleta *Muscular Men's Physique*

(a) (b)

Fonte: Produção do próprio autor.

de aprendizagem seja capaz de extrair desde características simples e triviais, como bordas, até características mais complexas que sejam determinantes para efetuar a tarefa de classificação com alta precisão. O aumento da complexidade das características extraídas está relacionada ao fato do DL utilizar arquiteturas profundas. Inspiradas na estrutura córtex/visual dos mamíferos, as redes neurais buscam modelar abstrações de alto nível de dados sempre visando maior proximidade com o comportamento cerebral humano.

Para problemas de classificação de imagens digitais, redes neurais são ferramentas de alta eficácia visto que atualmente há uma maior disponibilidade de recursos computacionais e um grande número de modelos pré-treinados que podem ser facilmente adaptados à tarefa desejada. As redes de maior importância para problemas de visão computacional e processamento de imagens são as Redes Neurais Convolucionais (CNN, do inglês *Convolutional Neural Networks*). A eficiência das CNN foi comprovada em 2012 ao apresentar uma melhora significativa na taxa de erro do desafio *Imagenet Large Scale Visual Recognition Challenge* - ILSVRC, que consiste na classificação de um banco de dados de 1000 classes.

Apesar de a maior parte das aplicações de DL serem no ramo da tecnologia, atualmente nota-

se um crescimento de soluções interdisciplinares, como por exemplo o uso de imagens para estimar somatotipos de atletas de fisiculturismo (GONÇALVES et al., 2016). Contudo, diante dos resultados surpreendentes das CNN em problemas de classificação, viabiliza-se a aplicação das ferramentas de aprendizado de máquina na problemática do trabalho em questão, uma vez que trata-se da classificação de duas modalidades de fisiculturismo que dependem apenas da análise subjetiva da comissão julgadora.

## 1.2 Trabalhos Relacionados

Na literatura atual é possível encontrar trabalhos relacionados à temática do fisiculturismo. Primeiramente pode-se citar o trabalho desenvolvido por Silva, Trindade e Rose (2003) intitulado **Composição Corporal, Somatotipo e Proporcionalidade de Culturistas de Elite do Brasil**. Este trabalho possui por principal objetivo descrever características, como composição corporal e somatotipo de atletas de fisiculturismo, a fim de auxiliar no processo de treinamento de culturistas.

Uma outra vertente relacionada à estimativa de somatotipos é o trabalho elaborado por Gonçalves e outros (2016), intitulado **Uma Análise da Viabilidade de Estimar Somatotipos de Fisiculturistas Através de Medições em Imagens**. Este trabalho foi desenvolvido a partir da premissa da importância do somatotipo como ferramenta para que profissionais de educação física possam auxiliar atletas de fisiculturismo. Nesse projeto, o autor propõe um método de estimar o somatotipo de atletas a partir do uso de técnicas de medições em imagens.

Em geral, os trabalhos citados apresentam a capacidade de extração de características de atletas, contudo, até o momento, não há aplicações para problemas reais de enquadramento de atletas em diferentes modalidades de fisiculturismo. De diferente modo, a proposta deste trabalho está baseada no uso de uma rede CNN atuando como um preditor de classes binário para identificar atletas de duas modalidades distintas de fisiculturismo, de forma a auxiliar no processo de arbitragem em campeonatos oficiais. Assim, além de extrair informações características de cada modalidade, o algoritmo desenvolvido retornará o resultado final de classificação.

## 1.3 Objetivos

O objetivo geral do projeto consiste em desenvolver uma CNN que atue como um preditor de classe binário para classificar dois tipos de perfil humano através de fotografias digitais. Para isso, pretende-se aplicar o conhecimento acerca de sistemas baseados em inteligência de máquina para treinar uma rede que consiga resolver de forma eficaz o problema em questão.

Para atingir o objetivo geral, definiu-se os seguintes objetivos específicos:

- Estudar métodos automatizados de análise de dados, como *Machine Learning* e *Deep Learning*;
- Obter modelo analítico que faça a separação antropométrica entre os atletas das modalidades *Men's Physique* e *Muscular Men's Physique*;
- Implementar uma CNN que atue como um preditor de classe binário utilizando métodos como transferência de aprendizagem (do inglês *transfer learning*) e ajuste fino (do inglês *fine-tuning*);
- Validar e testar a rede.

## 1.4 Estrutura do Texto

O texto deste trabalho está dividido em 5 capítulos. O Capítulo 1 possui o objetivo de apresentar a motivação para o desenvolvimento do projeto além de expor o cenário atual acerca do tema abordado. O Capítulo 2 apresenta o embasamento teórico para o desenvolvimento do projeto enquanto que o Capítulo 3 possui o intuito de expor a metodologia de desenvolvimento do projeto pra resolver o problema proposto. A exposição e discussão dos resultados obtidos estão presentes no Capítulo 4. O texto finaliza-se com o Capítulo 5 no qual são apresentadas as conclusões e as propostas de trabalhos futuros.

# 2 EMBASAMENTO TEÓRICO

## 2.1 Introdução

Este capítulo tem por finalidade expor os conceitos teóricos que foram utilizados como base do trabalho em questão. Dessa forma, o capítulo inicia-se com uma breve alusão histórica e definições de conceitos importantes acerca da teoria de *Machine Learning* e *Deep Learning*, como por exemplo as métricas de avaliação e a aplicação de Redes Neurais Convolucionais em problemas como o de classificação binária. Em seguida, é descrito o método de estimativa de pontos de interesse do corpo humano, o *OpenPose*. Por fim, adentra-se no universo do fisiculturismo e assim são expostas as principais características das modalidades foco do estudo.

## 2.2 Redes Neurais Artificiais

As Redes Neurais Artificiais - RNA são sistemas computacionais que buscam modelar as capacidades cognitivas do cérebro humano para resolver tarefas específicas. Dessa forma, os neurônios naturais que formam as redes neurais biológicas foram a inspiração para a modelagem dos neurônios artificiais, propostos por (MCCULLOCH; PITTS, 1943), que compõem as RNA.

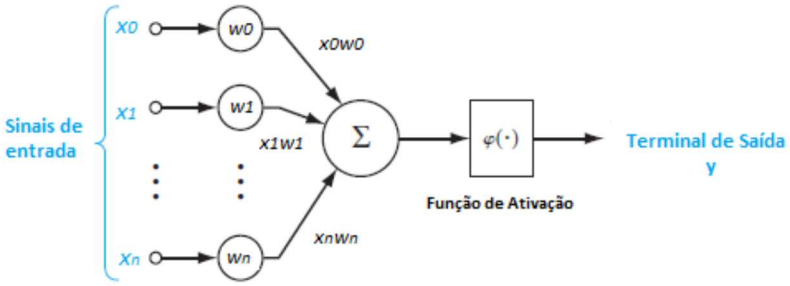
O modelo matemático resultante de um neurônio, mostrado na Figura 2, inspirado no neurônio biológico possui: $n+1$ terminais axônicos de entrada, oriundos de outros neurônios, representados por $x_0, x_1, ..., x_n$; $n+1$ pesos $w_0, w_1, ..., w_n$ que representam os conjuntos de sinapses, responsáveis por estabelecer a conexão entre os neurônios a partir de multiplicações $x_0w_0, x_1w_1, ..., x_nw_n$; um somador para os sinais de entrada que já foram multiplicados pelos respectivos pesos; uma função de ativação não linear responsável por limitar a amplitude de saída do neurônio; um terminal de saída $y$.

Portanto, o sinal de saída $y$ seguirá a forma da função de ativação escolhida, conforme exposto pela equação (2.1).

$$y = \varphi(\sum_{k=0}^{n} w_k x_k) \tag{2.1}$$

De acordo com o modelo de McCulloch e Pitts (1943), a função de ativação era dada pela função

Figura 2 – Modelo matemático do neurônio




Fonte: Haykin (2010).
Nota: Adaptado pelo autor.

de limiarização (do inglês *threshold*), onde a saída sempre assumirá os valores de zero ou um, conforme exposto pela equação (2.2), na qual $\alpha$ é o limiar de ativação do neurônio.

$$\varphi(x) = \begin{cases} 1, & x \geq \alpha \\ 0, & x < \alpha \end{cases} \tag{2.2}$$

Contudo, como derivação do modelo de neurônio proposto inicialmente, novos modelos surgiram com diferentes funções de ativação que permitem que o valor de saída seja diferente dos obtidos pela equação (2.2). Dentre elas, tem-se:

- A função linear:

$$\varphi(x) = \alpha x \tag{2.3}$$

- A função sigmoide:

$$\varphi(x) = \frac{1}{1 + e^{-\alpha x}} \tag{2.4}$$

- A função tangente hiperbólica:

$$\varphi(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.5}$$

- A função sinal (Perceptron):

$$\varphi(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases} \tag{2.6}$$

- A função rampa:

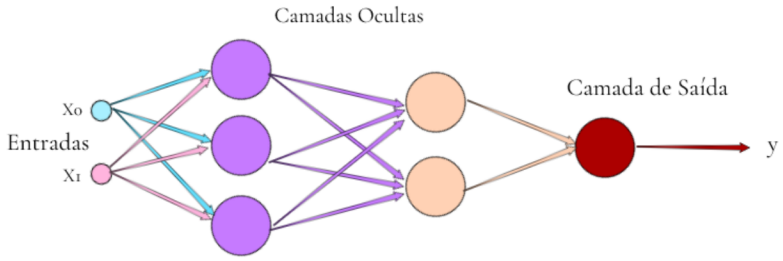
$$\varphi(x) = \begin{cases} 1, & x \geq \frac{1}{2} \\ x + \frac{1}{2}, & \frac{-1}{2} < x < \frac{1}{2} \\ 0, & x \leq \frac{-1}{2} \end{cases} \tag{2.7}$$

- A função ReLU:

$$\varphi(x) = max(0, x) \tag{2.8}$$

Os neurônios de uma RNA são organizados em uma estrutura de camadas. A primeira camada é a de entrada. Em seguida, camadas ocultas formam a estrutura responsável pelo processamento das informações da rede. Por fim, uma camada de saída para gerar o resultado final da RNA. Na Figura 3, pode-se observar uma estrutura básica de uma RNA, na qual $x_0$ e $x_1$ são as entradas enquanto $y$ é a saída gerada. O exemplo possui duas camadas ocultas sendo a primeira com três neurônios e a segunda com dois.

Figura 3 – Exemplo básico de estrutura de RNA




Fonte: Produção do próprio autor.

## 2.3 *Convolutional Neural Networks*

As redes neurais convolucionais, propostas em (LECUN et al., 1998), diferenciam-se das RNA por optarem pela utilização da operação matemática de convolução ao invés da multiplicação matricial. A operação de convolução consiste na aplicação de um filtro linear espacial de forma sucessiva por toda imagem a fim de detectar um padrão e assim gerar um mapa de características. Além disso, as CNN apresentam as seguintes características:

- **Conexões esparsas**: em camadas convolucionais, apenas alguns elementos de entrada contribuem com o cálculo de elementos de saída, enquanto que em camadas totalmente conectadas todos os elementos são utilizados;

- **Compartilhamento de parâmetros**: as vizinhanças possuem os mesmos pesos do filtro de entrada, acarretando em uma diminuição de parâmetros do modelo.
- **Equivariância**: é o resultado do compartilhamento de parâmetros. Já que os pesos de um filtro são usados sobre diferentes vizinhanças de entrada, a resposta do filtro é a mesma independente do deslocamento espacial.

A arquitetura básica de uma CNN usualmente consiste na junção de diferentes tipos de camadas. Primeiramente, tem-se uma camada opcional destinada ao processamento dos dados (CIRESAN et al., 2011) (normalmente utilizada em classificadores de imagens). Em seguida, tem-se a camada de convolução, a de *pooling* e por fim a camada totalmente conectada que também pode ser denominada por camada de classificação.

Em uma operação de convolução, o filtro é o responsável por detectar um padrão na matriz de entrada e gerar o respectivo mapa de características. Entretanto, em problemas de classificação de imagens, é preciso destacar inúmeras informações relevantes para o classificador. Assim, na camada de convolução, surge a necessidade de aplicação de vários filtros lineares, de forma que cada um será destinado para o reconhecimento de um determinado padrão. Dessa forma, para cada filtro será criado um mapa de características que em seguida é aplicado a uma função de ativação não linear acarretando em um mapa de ativação correspondente. O número de filtros utilizados define a profundidade da camada convolucional.

Cada camada de convolução comumente é seguida por uma camada de *pooling* que possui a finalidade de subamostrar a imagem de entrada para obter uma menor na saída. Assim, a quantidade de informações de cada característica obtida na camada de convolução, é reduzida mantendo sempre as mais importantes. As duas metodologias mais comuns de subamostragem são de *average pooling*, que resume a presença média de uma característica, e *max pooling* que sumariza o valor máximo no mapa de características.

Por fim, tem-se a camada totalmente conectada que tem por finalidade gerar um vetor dimensional de tamanho $N$ a partir da saída da última camada convolucional ou de *pooling*, sendo $N$ o número de classes em um problema de classificação.

## 2.4 Modelos Pré-Treinados

A tarefa de treinar uma CNN a partir de valores aleatórios de seus pesos, requer de um conjunto de dados com milhares de amostras e recursos computacionais. Na tentativa de contornar esses empecilhos, surge o método de *transfer learning*.

O *transfer learning* visa a utilização do conhecimento obtido a partir do treinamento de uma tarefa fonte, utilizando muitos dados rotulados, para tentar solucionar problemas relacionados, mesmo quando os dados de treino e teste possuem diferentes distribuições e características em uma tarefa destino (PAN; KWOK; YANG, 2008). Contudo, essa transferência de aprendizado só é possível pois as primeiras camadas de uma CNN são extratoras de características primitivas (RADHAKRISHNAN, 2017), por exemplo bordas, texturas e padrões. A complexidade das características extraídas aumenta conforme a profundidade da camada da rede. Portanto, as camadas iniciais de uma rede pré-treinada podem ser úteis para auxiliar na extração de características dos novos dados.

Quando os dados da tarefa fonte apresentam semelhança com os da tarefa destino, retira-se a camada de saída da rede e adiciona-se uma nova camada com pesos inicializados aleatoriamente. Nesse caso, apenas a nova camada é treinada, uma vez que o extrator de características primitivas da tarefa fonte pode ser reaproveitado. Entretanto, um outro cenário de aplicação consiste em retirar a camada de saída original da rede pré-treinada, adicionar uma nova e retreiná-la junto aos outros pesos das camadas remanescentes escolhidas. Esse procedimento é denominado *fine tuning*. No *fina tuning* o modelo pré-treinado atua inicializando os pesos para o treino de um novo modelo, o que normalmente ocorre de forma aleatória. Dessa maneira, ocorre uma adaptação do modelo pré-treinado ao novo banco de dados. Dentre as principais redes CNN pré-treinadas, destacam-se: VGG16 e VGG19, Inception e suas derivações, ResNet e DenseNet. Cada uma destas arquiteturas será explicada em seguida.

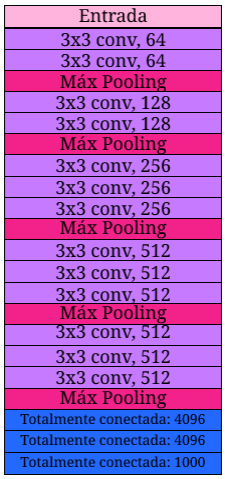
### 2.4.1 VGG16 e VGG19

As arquiteturas VGG16 e VGG19 foras criadas por Karen Simonyan e Andrew Zisserman (SIMONYAN; ZISSERMAN, 2015). Apresentaram elevada eficácia em problemas de reconhecimento de imagens em larga escala, como classificação do banco de dados ImageNet que possui

1000 classes.

A arquitetura da VGG16 apresenta 13 camadas convolucionais, 5 de *pooling* e 3 totalmente conectadas, conforme exposto pela Figura 4.

Figura 4 – Arquitetura da VGG16

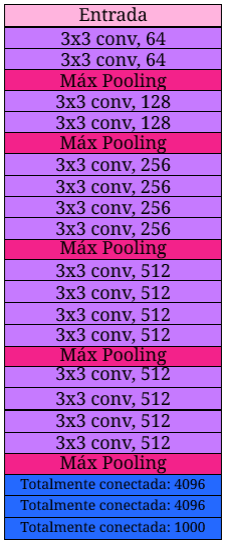



Fonte: Simonyan e Zisserman (2015).
Nota: Adaptado pelo autor.

Já a VGG19 apresenta 16 camadas convolucionais, 5 de *pooling* e 3 totalmente conectadas. A Figura 5 ilustra a arquitetura da rede.

Conforme exposto nas Figuras 4 e 5, os filtros utilizados nas camadas convolucionais são $3 \times 3$. O tamanho escolhido é o mínimo para que seja possível capturar noções de esquerda/direita e cima/baixo.

Figura 5 – Arquitetura da VGG19



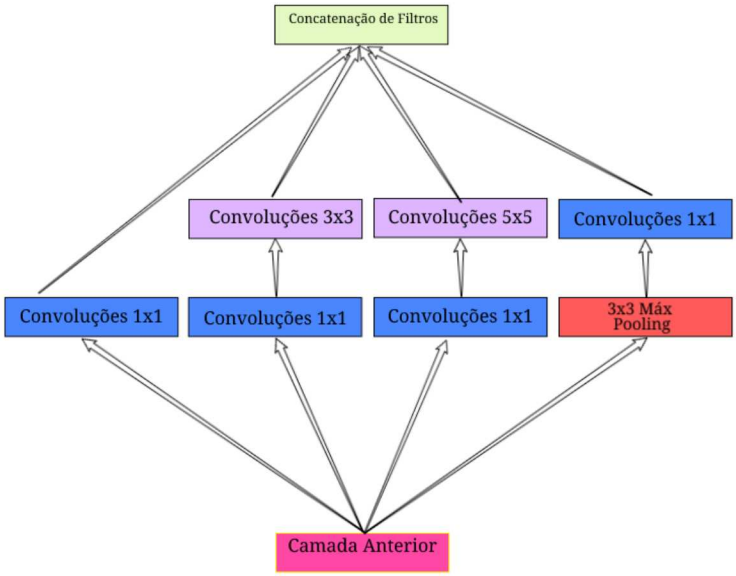

Fonte: Simonyan e Zisserman (2015).
Nota: Adaptado pelo autor.

### 2.4.2 Inception

O modelo da arquitetura Inception foi proposto a partir da premissa de que a maneira mais direta de melhorar o desempenho de uma rede neural é aumentando a sua profundidade. Contudo, o aumento da profundidade usualmente acarreta em um número maior de parâmetros e assim gera uma rede propensa ao *overfitting*, principalmente quando o número de amostras rotuladas do banco de dados é escasso (SZEGEDY et al., 2015). Além disso, nota-se um aumento significativo no custo computacional atrelado ao aumento do número de neurônios. Dessa forma, a ideia principal da Inception V1 é descobrir como uma rede esparsa pode se aproximar de uma densa aplicando múltiplos filtros de diferentes tamanhos em cada determinado nível de profundidade. Já para diminuir o custo computacional, os autores (SZEGEDY et al., 2015) propuseram a adição de uma convolução $1 \times 1$ no módulo Inception V1, antes de cada convolução $3 \times 3$ e $5 \times 5$ acarretando na limitação do número de canais de entrada. Por fim, as saídas são concatenadas e enviadas para o próximo módulo (ver Figura 6).

A Inception V2 e a Inception V3 foram propostas no mesmo artigo (SZEGEDY et al., 2016) em 2015. A Inception V2 diferencia-se da primeira versão, pois utiliza métodos de fatoração nas convoluções para incrementar a velocidade computacional. A fatoração da convolução de um filtro de tamanho $n \times n$ é feita em combinações de convoluções de tamanho $1 \times n$ e $n \times 1$. Além

Figura 6 – Módulo Inception V1




Fonte: Szegedy e outros (2015).
Nota: Adaptado pelo autor.

disso, os bancos de filtros do módulo foram expandidos a fim de diminuir a perda de informações entre camadas.

A Inception V3, além de trazer consigo as adaptações da Inception V2, propõe melhorias como a utilização de um novo otimizador, a normalização de lote (*batch)* entre outras.

Já em 2017 surge uma versão "extrema"da Inception denominada Xception (CHOLLET, 2017) que introduz a ideia de uma arquitetura baseada em camadas de convolução separáveis em profundidade com conexões residuais. As convoluções separáveis em profundidade consistem na realização de uma convolução espacial seguida por uma convolução $1 \times 1$ separadamente em cada canal da entrada. Entretanto, na Xception a ordem das operações é invertida e não existe uma não função de ativação intermediária o que gerou uma melhor acurácia nos testes.

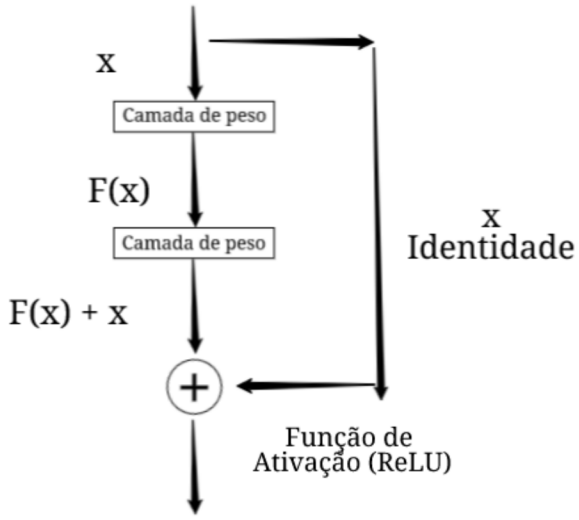
### 2.4.3 ResNets

Sabe-se de antemão que o grau de complexidade das características extraídas aumenta de acordo com a profundidade da rede. Entretanto, nota-se que a partir de um determinado grau de profundidade, a rede passa a perder a sua capacidade de generalização, haja vista que a acurácia do modelo tende a saturar degradando o desempenho. Além disso, os gradientes calculados a partir da regra da cadeia na função de perda tendem a zero, acarretando em um aprendizado nulo.

Esse problema é conhecido como o desaparecimento do gradiente (*vanishing gradient*).

Dessa forma, as ResNets (HE et al., 2016) representam a solução para o treino de redes de alta profundidade, ultrapassando a barreira das 100 camadas. As principais características da arquitetura das ResNets são: (*i*) conexões de salto que adicionam a entrada original à saída do bloco convolucional. Essas conexões são responsáveis por contornar o problema do *vanishing gradient*, haja vista que criam atalhos pelos quais o gradiente pode fluir. Além disso, efetuam um mapeamento de identidade a fim de garantir que a camada superior apresente um desempenho tão satisfatório quanto a camada anterior, conforme exposto pela Figura 7; (*ii*) normalização por lotes diretamente após cada convolução e antes da ativação.

Figura 7 – Conexão de Salto

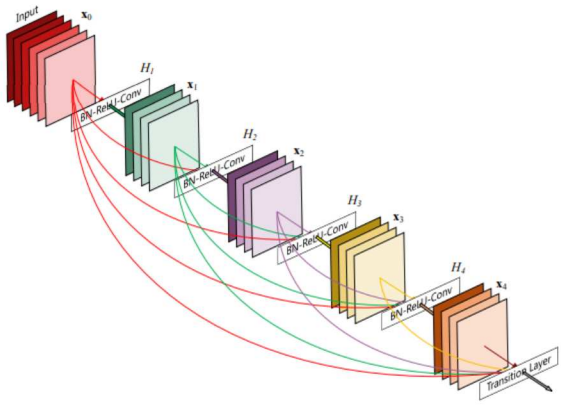



Fonte: He e outros (2016).
Nota: Adaptado pelo autor.

### 2.4.4 DenseNet

A Densenet - Rede Convolucional Densa (do inglês, *Dense Convolutional Network*), proposta por Gao Huang, é a arquitetura que surge logo após a descoberta das ResNets e a apresentam vantagens como a solução do problema do *vanishing gradient*, uma vez que o sinal de erro pode ser facilmente propagado e monitorado nas camadas finais, reutilização e diversidade de recursos, o que possibilita a extração de padrões mais complexos e redução do número de parâmetros quando comparado com uma ResNet (HUANG et al., 2017). Essas particularidades se devem ao fato de que na DenseNet cada camada recebe o mapa de características de todas as camadas anteriores, conforme ilustrado pela Figura 8.

Figura 8 – Arquitetura básica de uma DenseNet




Fonte: Huang e outros (2017).

## 2.5 *Human Pose Estimation* - Estimativa de Pose Humana

A estimativa de pose humana é definida como um problema de localização de articulações e partes do corpo, *keypoints* (pontos-chave), de um ou mais humanos em imagens e vídeos. A detecção de mais de um indivíduo em uma imagem é um problema de maior complexidade do que a detecção de apenas um, haja vista que em um determinado cenário pode-se encontrar um número desconhecido de pessoas em diferentes escalas e posições, além de diversos outros fatores que podem interferir na tarefa como oclusões de articulações. Usualmente a metodologia das soluções para a estimativa de pose de várias pessoas consiste na aplicação de um detector de pessoas na imagem e em seguida utilizam o algoritmo de estimativa de pose de uma única pessoa em cada detecção. Todavia essa solução pode apresentar falhas devido às interferências citadas anteriormente. Além disso, o tempo de execução será proporcional ao número de pessoas na imagem.

### 2.5.1 *OpenPose*

*OpenPose* (CAO et al., 2017) é a primeira biblioteca aberta para detecção de *keypoints* no corpo, pés, mãos e face humana de múltiplas pessoas em tempo real, totalizando 135 pontos de interesse. A metodologia utilizada é a *bottom-up*. Nessa nova abordagem, o algoritmo primeiramente detecta partes do corpo de baixo para cima (*bottom-up*) em toda a imagem e a análise final é utilizada para a extrair as estimativas de pose. A grande vantagem dessa metodologia é que dessa forma o tempo de execução não fica associado à quantidade de pessoas encontradas no cenário. Na Figura 9, tem-se um exemplo de aplicação da biblioteca *OpenPose* em um atleta do banco de

dados utilizado no desenvolvimento do trabalho.

Figura 9 – Estimativa de pose de atleta *Men's Physique*

Fonte: Produção do próprio autor.

## 2.6 Fisiculturismo

O fisiculturismo é uma modalidade esportiva que enfatiza o culto ao corpo e a busca pela perfeição estética, visando sempre excelentes parâmetros de antropomorfia e antropometria. Os atletas buscam simetria corporal, definição e desenvolvimento muscular por meio de dietas extremamente seletivas e árduos treinamentos de força.

Em competições oficiais promovidas pela IFBB, os principais pilares da base de julgamento são (SANTONJA, 2019):

- **Desenvolvimento muscular**: volume e densidade da musculatura do atleta. Visualização da rigidez muscular.
- **Definição muscular**: separação e profundidade da musculatura do atleta. Visualização de grupos musculares, planos e fibras.
- **Presença de palco**: personalidade, coreografia, aspecto saudável, dentre outros fatores.
- **Simetria corporal**: proporção harmônica entre volume muscular e comprimento dos ossos, silhueta e balanço de massa muscular.

Cada modalidade feminina e masculina apresenta suas respectivas exigências baseadas nesses quatro pilares de julgamento. As duas modalidades foco do trabalho desenvolvido serão abordadas nas subseções 2.6.1 e 2.6.2.

### 2.6.1 *Men's physique*

A modalidade *Men's Physique* foi oficialmente reconhecida pela IFBB *Executive Council* e IFBB *Congress* como uma nova modalidade de fisiculturismo em 11 de novembro de 2012, em Guayaquil no Equador IFBB (2017). A divisão de categorias é feita da seguinte forma (SANTONJA, 2019): (*i*) seis categorias divididas por altura no *Senior Men's Physique*; (*ii*) três categorias divididas por altura no *Junior Men's Physique*; (*iii*) três categorias divididas por idade no *Masters Men's Physique*.

No geral, os atletas dessa modalidade devem apresentar volumes musculares que não caracterizem um culturista, ou seja, devem ser portadores de baixa profundidade, grande separação e densidade muscular, além de média definição (SANTONJA, 2019). Os ombros e dorsais devem ser bem desenvolvidos em relação aos braços, trapézios e peito. A cintura dos atletas deve ser fina e pouco desenvolvida, assim como seu abdômen.

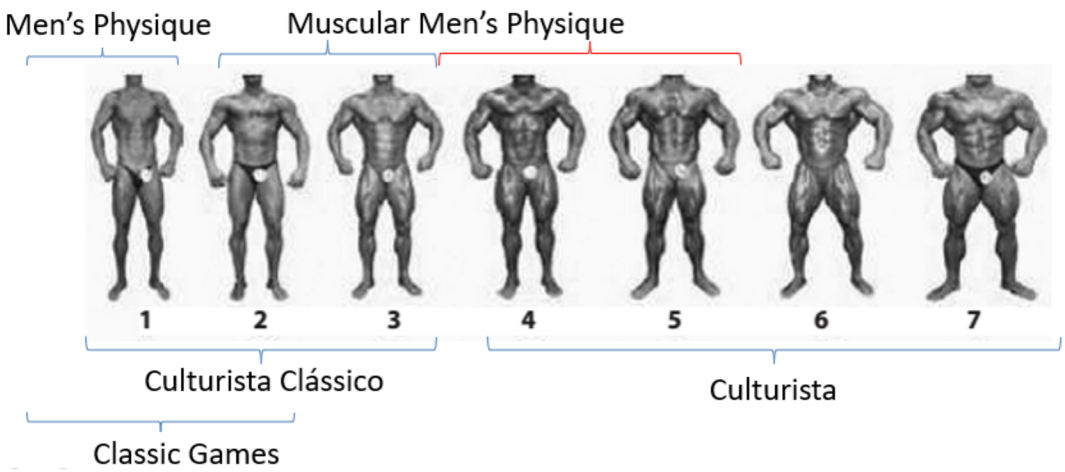
### 2.6.2 *Muscular men's physique*

Essa modalidade surgiu a partir da problemática envolvendo competidores que estavam participando da *Men's Physique* apresentando um volume muscular acima do limite definido pela regra da categoria, mas mesmo assim não se encaixavam como culturistas, nem como culturistas

clássicos, haja vista que esta última apresenta restrições que devem ser respeitadas de acordo com as relações de peso e altura de cada categoria.

Para uma melhor exemplificação, basta analisar a Figura 10. Os atletas da modalidade *Men's Physique* devem ter volume igual ou inferior à silhueta de número 1. Os atletas da modalidade *Muscular Men's Physique* se assemelham às silhuetas representadas entre a número 2 e 3. Observe que a silhueta 1 representa o limiar entre o *Men's Physique* e o Culturista Clássico. Já para o *Muscular Men's Physique* é permitido um volume muscular baixo, semelhante ao culturista clássico, porém nunca um volume de um culturista não clássico.

Figura 10 – Silhuetas base de atletas do fisiculturismo




Fonte: Castro e outros (2011).
Nota: Adaptador pelo autor.

# 3 PROPOSTA

## 3.1 Introdução

O trabalho em questão possui o intuito de desenvolver uma CNN que seja capaz de atuar como um preditor de classes binário para classificar atletas de duas modalidades de fisiculturismo, a *Men's Physique* e a *Muscular Men's Physique*. Para viabilizar a solução do problema proposto, se fez necessário de estudo de técnicas de *Deep Learning* como por exemplo o *transfer learning*, validação cruzada por *K-Fold* estratificado, *fine tuning*, redes convolucionais, entre outros. Além disso, a escolha correta da métrica de avaliação ideal foi de extrema importância, uma vez que o banco de dados disponível é numericamente limitado, desbalanceado e foi rotulado de forma subjetiva por 7 integrantes da comissão de arbitragem da IFBB - Espírito Santo.

## 3.2 Arquitetura do Modelo

Aqui é proposto um modelo de aprendizado composto de duas etapas, uma etapa para o registrado das imagens e outra para classificação:

- **Registro das imagens**. As imagens foram registradas utilizando *OpenPose*, os *keypoints* retornados por *OpenPose* foram usados para recortar as imagens em retângulos que apenas contassem com a região de interesse (ROI, do inglês *Region of Interest*) para o julgamento de cada modalidade.
- **Etapa de classificação**. Considerando como entrada a ROI, uma rede CNN é treinada para operar como classificador das duas classes, as quais correspondem às modalidades *Muscular Men's Physique* e *Men's Physique*.

### 3.2.1 Resgistro das imagens

Antes de criar uma *bounding box* (caixa delimitadora) que abrange apenas a ROI de cada modalidade para a comissão julgadora, se fez necessária a seleção dos pontos que serão alocados em seus respectivos vértices. Vale ressaltar que como exposto anteriormente na Subseção 2.6.1, ombros, dorsais, trapézios, peito e cintura são os alvos de análise dos árbitros. Diante disso, utilizou-se um serviço de detecção de esqueletos (QUEIROZ, 2018) baseado na biblioteca

*OpenPose* disponível no Espaço Inteligente (ALMONFREY et al., 2018) do Laboratório de Visão Computacional e Robótica da UFES. O serviço de detecção de esqueletos oferece a possibilidade de escolha entre dois modelos de *keypoints*, o COCO, que retorna 15 pontos, e o MPI que retorna 18. Vale ressaltar que a implementação do serviço de detecção de esqueletos não segue o padrão da implementação oficial do *OpenPose*, portanto algumas divergências entre os modelos são notáveis, como a identificação de cada ponto.

Primeiramente foi desenvolvido um algoritmo em Python cuja finalidade é enviar cada imagem para o serviço de modo a extrair os *keypoints* de acordo com o modelo escolhido. O formato de retorno fornecido pelo serviço é JSON e contém as coordenadas de cada ponto, sua identificação (*id*) e o seu respectivo índice de confiabilidade (*score*). Além disso foi elaborado um algoritmo para interligar os *keypoints* ilustrando o esqueleto criado. Em seguida, foi elaborado um outro *script* cuja finalidade é ler o arquivo JSON de todas as imagens, selecionar os pontos escolhidos a partir de seu respectivo *id* e em seguida gerar a *bounding box* de cada imagem automaticamente.

Se fez necessário utilizar o serviço duas vezes, uma vez que o modelo COCO retorna os pontos que serão utilizados como vértices da região de interesse, enquanto que o ponto máximo de altura é retornado pelo MPI. Os pontos escolhidos do modelo COCO para representar os vértices foram: ombro direito ($id = 4$), ombro esquerdo ($id = 7$), quadril direito ($id = 10$), quadril esquerdo ($id = 13$). Contudo, para estabelecer o ponto máximo do eixo $y$ de forma que as curvas dos ombros dos atletas não fossem cortadas, foi necessário extrair o ponto do pescoço ($id = 3$) modelo MPI. O resultado final do tratamento da imagem está exposto pela Figura 11.

### 3.2.2 Etapa de classificação

Para solucionar o problema de classificação de imagens de atletas nas classes *Men's Physique* e *Muscular Men's Physique*, optou-se pelo uso de modelos de CNN pré-treinados. As redes escolhidas, para serem usadas via o paradigma de *transfer learning* foram: VGG, DenseNet, ResNet e Xception. Sendo escolhidas por seus bons resultados em tarefas de visão computacional e principalmente pela diferença de profundidade das arquiteturas, incluindo redes de profundidade média (VGG), até redes de grande profundidade (DenseNet).

A metodologia usada na aplicação de *transfer learning* foi: (*i*) Um modelo pré-treinado é

Figura 11 – Comparativo entre imagem com esqueleto desenhado e resultado do algoritmo de *bounding box*, no qual (a) é a imagem com esqueleto desenhado e (b) é a *Bounding Box*

(a) (b)

Fonte: Produção do próprio autor.

selecionado e é cortado na última camada convolucional, adicionando-se algumas camadas densas, de modo que a última camada densa tenha uma função de ativação *sofmax* e dois neurônios de saída. (*ii*) Inicia-se o processo de treinamento "congelando" os pesos relacionados às camadas originais do modelo pré-treinado, e atualizando os pesos das camadas densas a partir de valores aleatórios. (*iii*) Finalizada a etapa anterior, inicia-se o treinamento em modo *fine tuning*, ou seja: todos os pesos do modelo (camadas originais e camadas adicionadas) são atualizados no treinamento, mas as camadas originais do modelo pré-treinado são "descongeladas" e os pesos das camadas densas são pré-inicializados em função dos valores assumidos no passo anterior.

O conjunto de dados em estudo está conformado por 76 imagens, onde, 56 imagens pertencem à primeira classe (*Men's Physique*) e as restantes à segunda classe (*Muscular Men's Physique*). Tal conjunto de dados caracteriza um desafio na elaboração de modelos de ML: lidar com um conjunto de dados com uma pequena quantidade de amostras e com classes desbalanceadas (HE;

GARCIA, 2009).

Modelos que trabalham com pequenos conjuntos de dados apresentam maior tendência ao *overfitting* enquanto que a distribuição desbalanceada de classes acarreta em predições ineficientes e erros de acurácia. Assim, algumas estratégias para lidar com essas peculiaridades foram de suma importância para o desenvolvimento da etapa de classificação. Foram elas:

- Aplicação de *transfer learning*.
- Utilização de métodos de regularização para manter o modelo mais conservador.
- Remoção de *outliers*.
- Escolha da métrica de avaliação adequada.
- Utilização de uma metologia de validação cruzada estratificada para garantir que os conjuntos de treino e validação sejam formados com o mesmo percentual de amostras de cada classe do banco de dados original.
- Uso de uma função de perda ponderara pelos pesos de distribuição de classes (*weighted cross entropy*.

A forma em que estas estratégias foram aplicadas é explicada no capítulo a seguir.

# 4 RESULTADOS

## 4.1 Introdução

Nesse capítulo os resultados obtidos a partir de experimentos realizados com os modelos propostos serão expostos, comparados e discutidos.

Foi dada prioridade aos testes com as redes pré-treinadas VGG16 e VGG19 por apresentarem menor quantidade de camadas convolucionais e serem de fácil entendimento. Os resultados obtidos com modelos mais densos serão expostos de maneira breve. Em todos os experimentos a classe escolhida como predição positiva foi a *Muscular Men's Physique*. Essa escolha foi feita tendo como base o fato de ser a classe minoritária, haja vista que a AUC da sua linha de base será menor e assim resultados acerca do poder de classificação do modelo serão mais conclusivos.

## 4.2 Conjunto de Dados

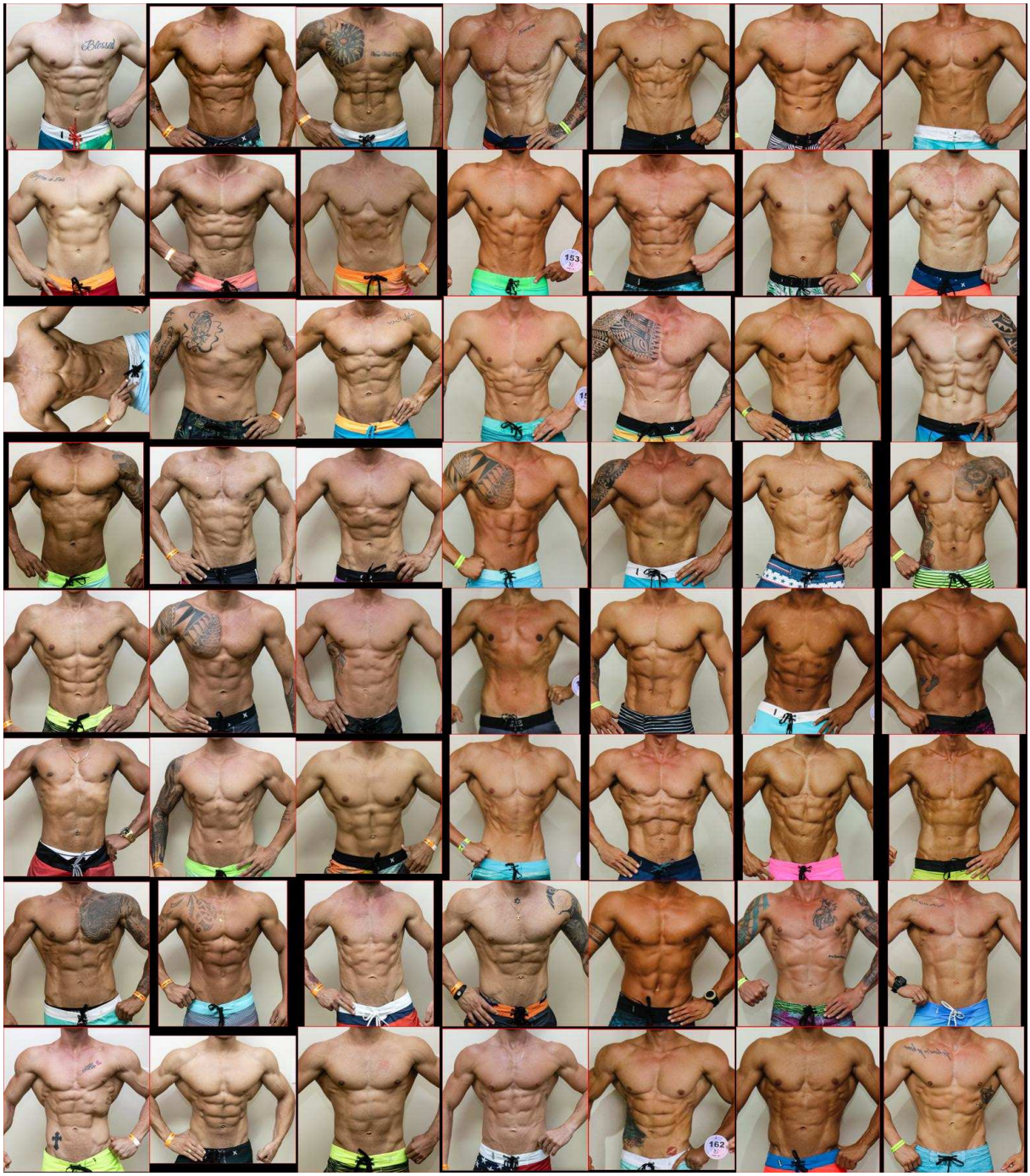
A literatura atual não dispõe de uma base de dados com objetivos semelhantes ao do trabalho proposto. Dessa forma, as imagens utilizadas para a realização do treinamento e validação dos classificadores foram adquiridas conforme exposto no Apêndice C. O conjunto de dados contém $76$ imagens RGB, sendo $20$ de atletas da modalidade *Muscular Men's Physique*, expostos na Figura 12, e $56$ *Men's Physique*, expostos na Figura 13. As imagens utilizadas foram dividias utilizando *K-fold* estratificado com $K = 3$ e possuíam dimensões de $224 \times 224 \times 3$ *pixels*.

Conforme exposto anteriormente, trata-se de um banco de dados com um pequeno número de amostras e com classes desbalanceadas.

## 4.3 Recursos Computacionais

**Recursos de *Software*** As etapas de treinamento e teste da arquitetura proposta foram realizadas utilizando Keras, *software* de código aberto implementado em Python e desenvolvido inicialmente por um engenheiro da Google. Utilizou-se *Tensorflow* como *backend* do Keras. *Tensorflow* é um *framework* de código aberto para *Machine Learning* desenvolvido pela Google. Além disso foi utilizada a biblioteca Augmentor (BLOICE, 2016) para *Data Augmentation*. Toda programação foi implementada na linguagem Python.

Figura 12 – Base de dados *Men's Physique*



Fonte: Produção do próprio autor.

**Recursos de *Hardware*** Além disso, a máquina utilizada nos experimentos possuía a seguinte configuração: (*i*) sistema operacional Linux, distribuição Ubuntu Server 16.04; (*ii*) processador Intel Core i7-7700, 3.60GHz com 8 núcleos físicos; (*iii*) memória RAM de 32 GB; (*iv*) unidade de armazenamento de 1TB (disco rígido); (*v*) placa de vídeo Titan Xp, com 12 GB de memória dedicada.

Figura 13 – Base de dados *Muscular Men's Physique*

Fonte: Produção do próprio autor.

## 4.4 Detalhes de Implementação

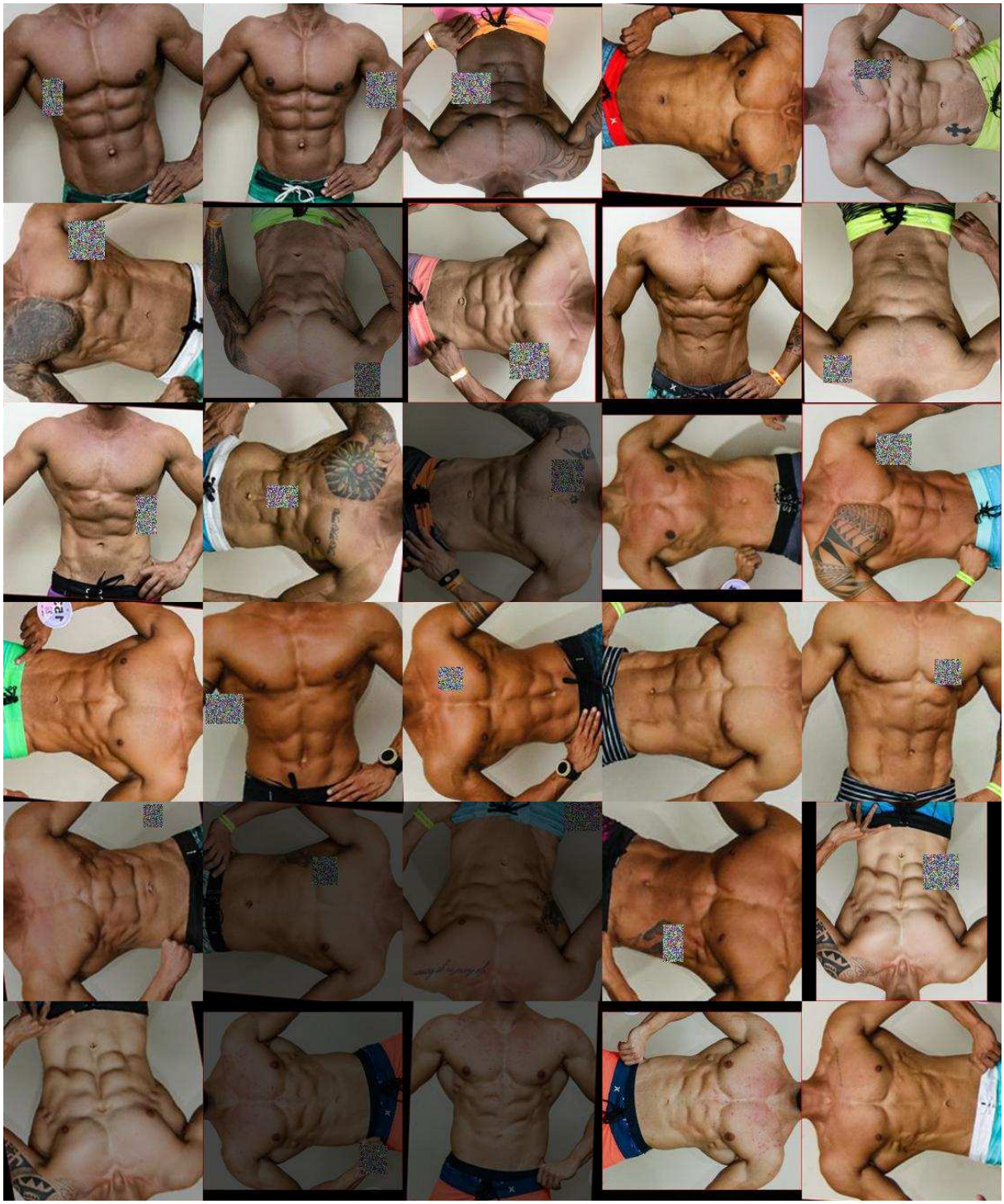
Os hiperparâmetros selecionados para os modelos foram obtidos de maneira experimental. A função de perda escolhida foi a Entropia Cruzada Binária Ponderada (APÊNDICE A). O

otimizador escolhido para o modelo foi o *Stochastic Gradient Descent* - SGD (Gradiente Descendente Estocástico). A taxa de aprendizado de $0,0001$. Vale ressaltar que para a aplicação de *Fine-Tuning* a taxa de aprendizado foi $10$ vezes menor. Já o tamanho dos lotes foi de $20$ e o número de épocas foi $200$ para o treinamento com as camadas da rede pré-treinada "congeladas". Ao aplicar *fine-tuning* o número de épocas diminuiu para $100$.

Um preditor de classes de bom desempenho deve apresentar um elevado poder de generalização, ou seja, grande capacidade de classificar corretamente entradas previamente não observadas. Entretanto, quando o número de amostras do banco de dados é muito inferior quando comparada ao número de parâmetros a serem otimizados pela rede, o modelo tende a sofrer *overfitting* e assim não alcançar a generalidade necessária. Diante disso, é viável a aplicação de uma técnica de regularização denominada *Data Augmentation*, cujo objetivo consiste em gerar novos exemplares de dados de treinamento a partir de transformações como rotações, espelhamento, inserção de ruídos, entre outros. Para implementar o *Data Augmentation* optou-se por utilizar uma biblioteca desenvolvida em Python, denominada Augmentor (BLOICE, 2016). O *script* implementado para o processo de *Data Augmentation* inicia-se com a criação de um *pipeline* que é apontado para o diretório onde encontra-se as imagens que serão utilizadas como base das transformações. Em seguida, foram escolhidas as seguintes operações: espelhamento vertical, espelhamento horizontal, rotação $90°$, pequenas rotações de $10°$ para esquerda e para a direita, remoção de regiões retangulares aleatórias da imagem, e mudança de brilho aleatória. Além da vasta quantidade de operações, o Augmentor apresenta a grande vantagem de gerar as imagens em tempo de execução. Na Figura 14 pode-se observar um exemplo das transformações que foram utilizadas para a expansão do banco de dados.

Além do *Data Augmentation*, outro método de regularização utilizado foi *dropout* (SRIVASTAVA et al., 2014), que é uma das técnicas de maior uso em DL, orientada a reduzir o *overfitting* de modelos baseados em redes neurais. *Dropout* funciona de maneira probabilística, removendo aleatoriamente elementos de saída das camadas densas na etapa de treinamento. Dessa forma, incrementa-se a dificuldade do treino e evita que as camadas das redes se adaptem ao erros das camadas anteriores. Esse procedimento assemelha-se ao treino de redes com diferentes arquiteturas em paralelo. A taxa de *dropout* é responsável por especificar a probabilidade de desligamento dos elementos das camadas. As camadas de *dropout* foram inseridas entre as camadas densas adicionadas na saída do modelo pré-treinado.

Figura 14 – Exemplo de aplicação das operações utilizadas do Augmentor



Fonte: Produção do próprio autor.

## 4.5 Métricas de Avaliação

Para avaliação do classificador binário desenvolvido, o seguinte conjunto de métricas foi utlizado:

- **Acurácia**: Mede o número de amostras preditas corretamente em relação ao número total

de amostras. Assim, a acurácia pode ser calculada de acordo com a equação (4.1).

$$Ac = \frac{VP + VN}{VP + VN + FP + FN} \times 100\% \tag{4.1}$$

Onde $VP$ (verdadeiros positivos) é o número de amostras positivas que foram preditas corretamente; $VN$ (verdadeiros negativos) é o número de amostras negativas que foram preditas corretamente; $FP$ (falsos positivos) é o número de amostras negativas que foram classificadas como positivas; $FN$ (falsos negativos) é número de amostras positivas que foram classificadas como negativas.

Apesar de ser uma métrica comumente utilizada em problemas de classificação, quando o banco de dados apresenta classes desbalanceadas, a acurácia tende a ser tendenciosa e assim não confiável[1]. Diante disso, a métrica de acurácia não é suficiente para avaliar um classificador. Outras métricas se fazem necessárias.

- **Precisão**: definida pela equação (4.2), é uma métrica que tem por objetivo avaliar a exatidão do modelo, ou seja, dentro de todas as amostras classificadas como positivas, identifica quantas foram preditas corretamente.

$$Pr = \frac{VP}{VP + FP} \times 100\% \tag{4.2}$$

- **Sensibilidade**. também chamada de taxa de verdadeiros positivos é descrita pela equação (4.3). Mensura a quantidade de entradas classificadas como positivas corretamente dentro de todas as amostras positivas.

$$Se = \frac{VP}{VP + FN} \times 100\% \tag{4.3}$$

- **Especifidade**: também chamada de taxa de verdadeiros negativos, tem por finalidade medir a quantidade de entradas classificadas como negativas corretamente dentro de todas as amostras negativas. A especifidade é calculada de acordo com a equação (4.4).

$$Esp = \frac{VN}{VN + FP} \tag{4.4}$$

[1] Um conjunto de dados que possui, por exemplo, $85\%$ dos dados pertencentes à primeira classe enquanto que apenas $15\%$ pertencentes à segunda é caracterizado como um conjunto com classes desbalanceadas. Portanto, se o classificador predizer todas as novas amostras que não foram observadas anteriormente como pertencentes à primeira classe, a acurácia será de $85\%$. Apesar der ser um valor de acurácia alto, não é válido dizer que o modelo foi validado, haja vista que apenas amostras de uma classe foram submetidas ao método de classificação.

- **F1**: é a média harmônica entre a sensibilidade e a precisão, conforme exposto pela equação (4.5).

$$F1 = 2 \times \frac{Se \times Pr}{Se + Pr} \tag{4.5}$$

## 4.6 Curvas de Avaliação

A partir das métricas de avaliação expostas na Seção 4.5 duas curvas de avaliação foram geradas.

### 4.6.1 Curva ROC

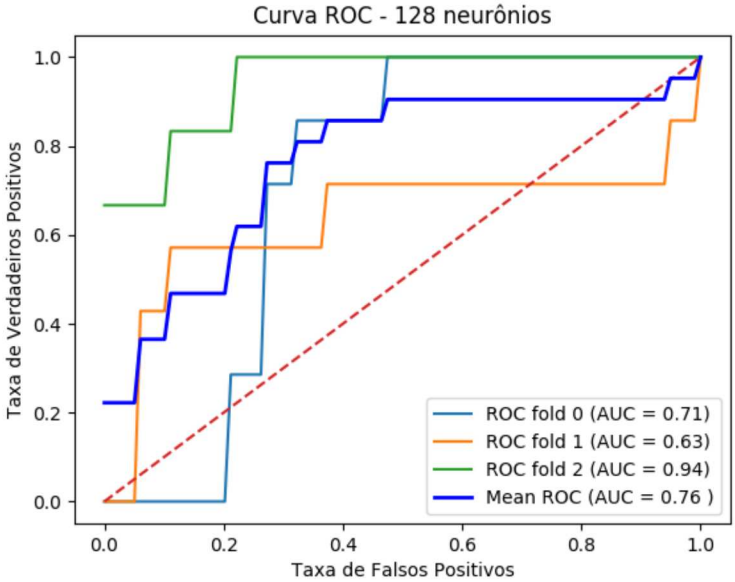
ROC vem do acrônimo *Receiver Operating Characteristic* (Características Operacionais do Receptor). Esta curva consiste em um gráfico da taxa de falsos positivos (eixo $x$) versus a taxa de verdadeiros positivos (eixo dos $y$) para um número de limiares diferentes entre $0,0$ e $1,0$. Assim, busca mensurar a desempenho do modelo de classificação a partir dos limiares de predição de probabilidades de cada classe. A razão para utilizar tal forma de predição e não a usual (predição direta para cada classe) é fornecer a capacidade de escolher e até calibrar o limite de como interpretar as probabilidades previstas (BROWNLEE, 2018).

A taxa de verdadeiros positivos é calculada de acordo com a equação (4.3). Já a taxa de falsos positivos é calculada de acordo com a equação (4.6). Note que a taxa de falsos positivos é igual a $1 - Esp$.

$$FPR = 1 - \frac{VN}{VN + FP} \tag{4.6}$$

Resumidamente, com a curva ROC é possível mensurar taxa de verdadeiros positivos e de falsos positivos para cada limiar de probabilidade de predição de cada classes. Além disso, uma medida comparativa entre as curvas se faz necessária. A área sob a curva (AUC, do inglês *Area Under the Curve*) é a representação do poder de separação do classificador. Assim, quanto maior o valor da AUC, melhor o desempenho do classificador. Quando um classificador não consegue distinguir as classes (classificador randômico) o seu AUC é igual a $0,5$. Esse valor é determinado pela área sob a curva da linha de base (do inglês, *baseline*) da curva ROC, representado pela reta $y = x$. Nos casos em que o modelo é validado a partir da metodologia de *K-fold* (APÊNDICE B), uma curva ROC é gerada para cada *Fold* e o resultado final é a média das AUC, conforme exposto pelo Gráfico 1.

Gráfico 1 – Curva ROC para $K = 3$




Fonte: Produção do próprio autor.

### 4.6.2 Curva de P-S

P-S vem das siglas Precisão-Sensibilidade (também conhecido na literatura pelos termos em inglês *Precision-Recall*). Esta curva relaciona a taxa de verdadeiros positivos com todos os valores preditos como positivos para diferentes limiares de probabilidades. A precisão (eixo $y$) é dada pela equação (4.2) e a Sensibilidade (eixo $x$) é dado pela equação (4.3). Valores altos de precisão representam baixas taxas de falsos positivos enquanto que uma alta sensibilidade representa uma taxa baixa de falsos negativos. Dessa forma, um valor alto de AUC representa que o classificador está retornando a maior parte das predições corretamente como positivas.

Enquanto que na curva ROC a linha de base é sempre fixada em $y = x$ e possui AUC = $0,5$, na Curva P-S ela muda de acordo com a variação do número de amostras positivas e negativas (SAITO; REHMSMEIER, 2015). Diante disso a curva P-S apresenta um grande diferencial para avaliar modelos de classificação binária com dados desbalanceados, ou seja, com diferentes números de amostras em cada classe. Outra característica importante é que quando há um grande número de amostras de uma determinada classe e ela não está sendo utilizada para predição, a quantidade de Verdadeiros Negativos ($VN$) tende a ser alta o que interfere nos resultados da curva ROC. Entretanto, quando trata-se da curva P-S, $VN$ não são utilizados para gerar o gráfico.

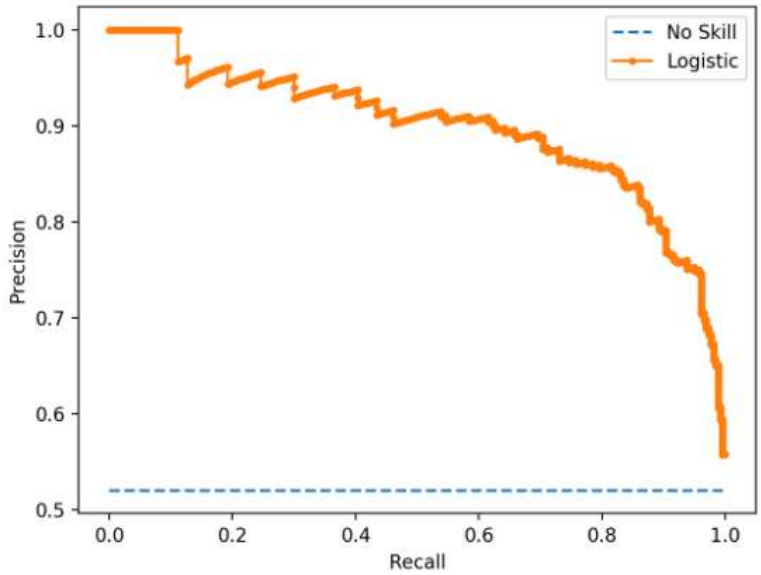
A linha de base é dada pela equação (4.7), onde $P$ é o número de amostras pertencentes à

classe que foi determinada como predição positiva, enquanto que $N$ é o número de amostras pertencentes à classe remanescente.

$$y_{P-S} = \frac{P}{P+N} \tag{4.7}$$

O Gráfico 2 representa um bom exemplo de uma curva P-S de um modelo de regressão logística.

Gráfico 2 – Curva P-S




Fonte: Brownlee (2018).

## 4.7 Experimentos

No primeiro experimento várias redes pré-treinadas com arquiteturas profundas foram usadas como extratores de características. Basicamente aqui foi analisada a dificuldade de treinar modelos com maior número de camadas convolucionais.

No segundo experimento foram usadas as redes VGG, uma vez que possuem arquiteturas de média profundidade. Portanto, o experimento foi proposto a fim de comparar os resultados com os obtidos utilizando os modelos do primeiro experimento.

Para ambos experimentos foram desconsiderados os *outliers*. Esse procedimento foi feito com o auxílio de um integrante da comissão de arbitragem da IFBB - Espírito Santo. A base de dados passou a possuir 17 atletas *Muscular Men's Physique* e 50 atletas *Men's Physique*. Portanto a

linha de base da curva P-S foi determinada de acordo com a equação (4.7) e consequentemente seu $AUC = 0,25$.

### 4.7.1 Protocolo de avaliação

As métricas escolhida foram a área sob a curva P-S (AUC P-S), e a área sob a curva ROC. A AUC da curva ROC (AUC ROC) é de suma importância, pois o seu valor exemplifica a capacidade do classificador em separar as classes corretamente. Entretanto, por se tratar de um problema com banco de dados desbalanceado é preciso ter cuidado com a métrica de avaliação escolhida. Diante disso, a AUC da curva P-S apresenta maior confiabilidade, uma vez que seu valor final exemplifica o quão bem o modelo é capaz de predizer corretamente as amostras de uma determinada classe para todos os valores de limiares de probabilidade.

Portanto o protocolo de avaliação iniciou-se com a a divisão do banco de dados em treino e validação. Para isso utilizou-se um *K-fold* Estratificado, que permite efetuar a validação cruzada por *K-fold* (APÊNDICE B) respeitando os percentuais de amostras de cada classe. O número de partições escolhido foi $3$ devido ao pequeno número de imagens do banco de dados. Além disso, foi definido que os dados não deveriam ser embaralhados antes da separação em partições. Vale ressaltar que o uso desse procedimento foi de suma importância, haja vista que a distribuição de classes do banco de dados é desbalanceada (especificamente 25,4% para a primeira classe e 74,6% para segunda). Em seguida, uma curva ROC e uma curva P-S para cada *fold* foi gerada com sua respectiva AUC. Por fim, a AUC média do modelo foi calculada a partir das médias dos *folds*.

### 4.7.2 Primeiro experimento

Foram realizados testes com os modelos pré-treinados DenseNet121, DenseNet201, ResNet50 e Xception. Arquiteturas com diferentes configurações de camadas densas e métodos de regularização foram testados. O *fine-tuning* foi realizado em cada modelo a partir da última camada convolucional ou *pooling*. O melhor resultado obtido para cada modelo pré-treinado está exposto no Quadro 1, junto de sua configuração com a quantidade de número de neurônios por camada densa ($D$) e a probabilidade da camada de *dropout* ($p$). Nota-se que os valores de AUC obtidos para os modelos ficaram relativamente próximos dos valores de AUC das linhas de base, fato que

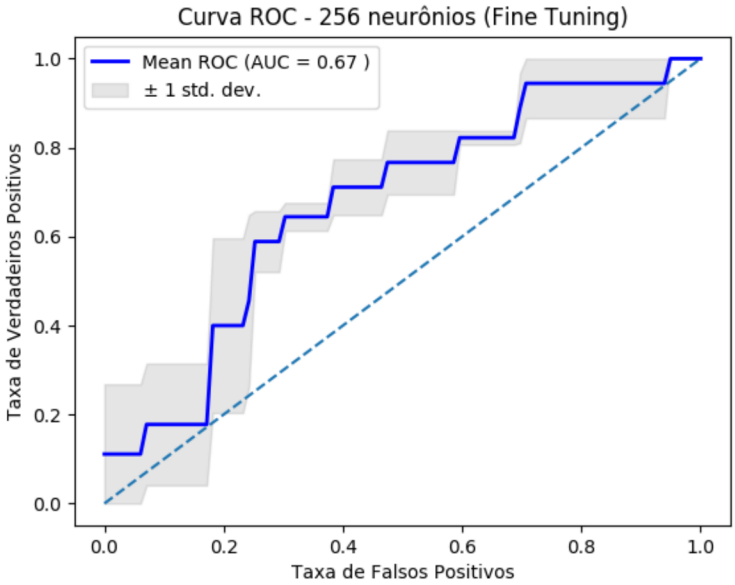
indica que os modelos não conseguiram separar as classes de maneira eficiente e que o número de predições retornadas corretamente como positivas não é satisfatório. Tal característica deve-se ao fato de que aplicar o *fine tuning* nesses modelos mais profundos não é uma tarefa trivial. Deve-se conhecer a arquitetura e saber exatamente o local ideal para "descongelar"os pesos. Os Gráficos 3 e 4 representam o melhor resultado obtido nessa etapa experimental.

Quadro 1 – Melhores resultados obtidos nos experimentos com redes densas

| Modelo Base | $D$ | $p$ | AUC ROC | AUC P-S ($y_{P-S} = 0,25$) |
|---|---|---|---|---|
| DenseNet121 | 256 | 0,5 | 0,67 | 0,40 |
| DenseNet201 | 256 | 0,5 | 0,53 | 0,24 |
| ResNet50 | 256 | 0,3 | 0,57 | 0,26 |
| Xception | 256 | 0,3 | 0,61 | 0,46 |

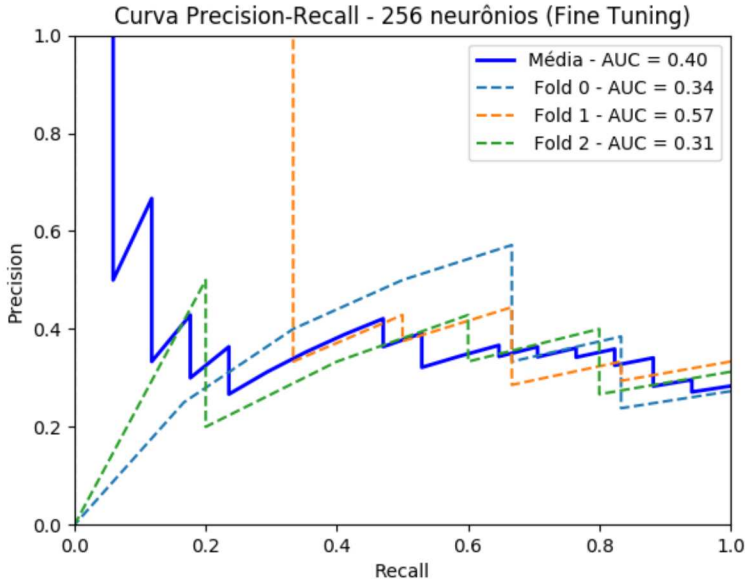
Fonte: Produção do próprio autor.

Gráfico 3 – Curva ROC - Melhor resultado utilizando DenseNet121 como modelo base




Fonte: Produção do próprio autor.

Gráfico 4 – Curva P-S - Melhor resultado utilizando DenseNet121 como modelo base




Fonte: Produção do próprio autor.

### 4.7.3 Segundo experimento

Nesse experimento foram utilizadas as redes VGG. A seguir serão expostos os melhores resultados.

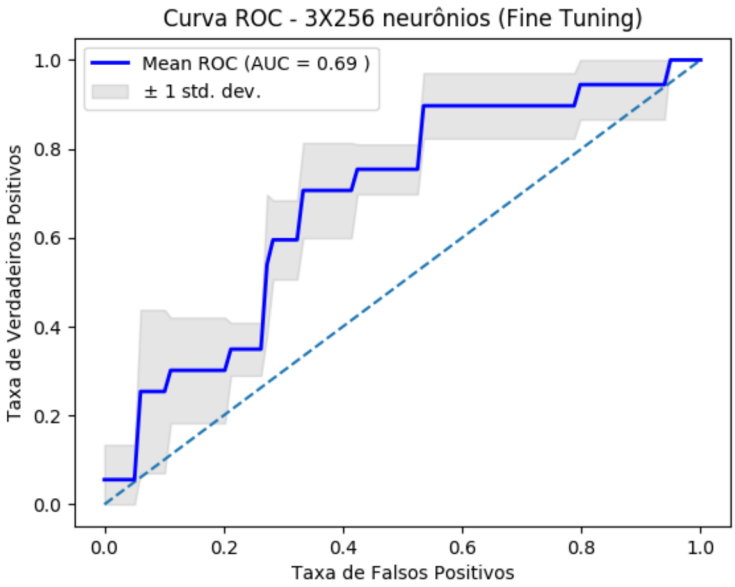
#### 4.7.3.1 Usando o modelo pré-treinado VGG19

As seguir são expostos os dois melhores resultados dos experimentos com a rede VGG19. Vários testes foram realizados a fim de testar arquiteturas com diferentes configurações de camadas densas e métodos de regularização. O corte na VGG19 foi realizado na camada 17 já que é a última camada convolucional do modelo base.

Nos Gráficos 5 e 6 encontram-se as duas curvas geradas no melhor teste utilizando o banco de dados completo. Enquanto que nos Gráficos 7 e 8 estão expostas as curvas resultantes do experimento sem os *outliers*. Note que os melhores resultados do experimento foram com a mesma configuração de camadas densas e de *dropout*. Outra análise importante a ser feita a respeito dos experimentos é que em ambos os casos a AUC da curva ROC foi a mesma $(0,69)$. Entretanto, ao comparar os valores de AUC das curvas P-S nota-se uma melhora significativa do segundo resultado quando comparado ao primeiro. Essa análise enfatiza: $(i)$ a importância da exclusão dos *outliers*; $(ii)$ a importância da aplicação do *fine tuning*.

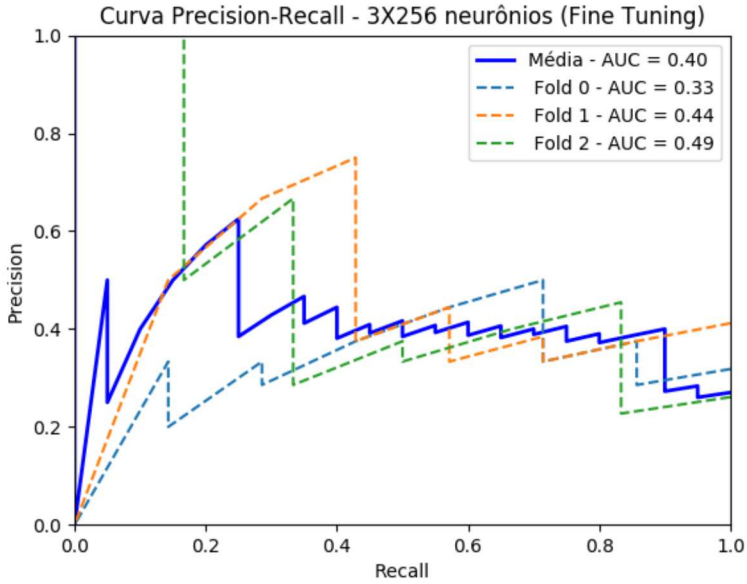
O Quadro 2 expõe resumidamente os melhores valores de AUC obtidos nessa etapa experimental. Conclui-se então que ao retirar os *outliers* o modelo apresentou um maior desempenho ao retornar predições positivas corretamente, uma vez que ocorrou um aumento significativo na AUC da curva P-S.

Gráfico 5 – Curva ROC - Resultado de experimento com a VGG19, usando o conjunto de dados completo - três camadas densas de 256 neurônios e duas de *dropout*



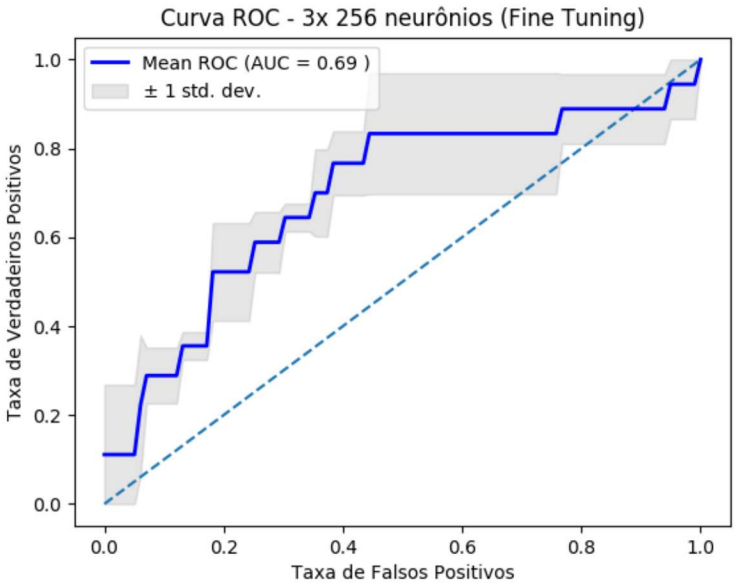

Fonte: Produção do próprio autor.

Gráfico 6 – Curva P-S - Resultado de experimento com a VGG19, usando o conjunto de dados completo - três camadas densas de 256 neurônios e duas de *dropout*



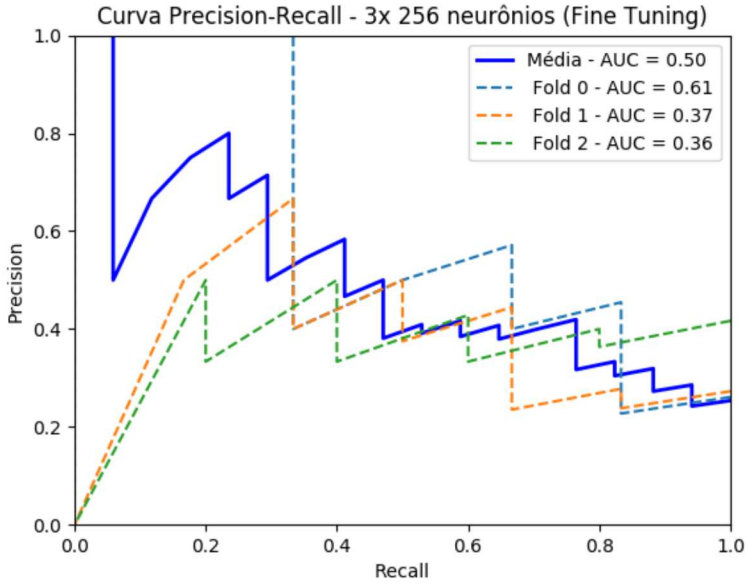

Fonte: Produção do próprio autor.

Gráfico 7 – Curva ROC - Resultado de experimento com VGG19, usando o conjunto de dados sem *outliers* - três camada densas de 256 neurônios e duas de *dropout* (0,3), (0,15)




Fonte: Produção do próprio autor.

Gráfico 8 – Curva P-S - Resultado de experimento com VGG19, usando o conjunto de dados sem *outliers* - três camada densas de 256 neurônios e duas de *dropout* (0,3), (0,15)




Fonte: Produção do próprio autor.

Quadro 2 – Melhores resultados obtidos com VGG19

| Dados | $D$ | $p$ | AUC ROC | AUC P-S |
|---|---|---|---|---|
| Completo | $3 \times (256)$ | (0,3),(0,3) | 0,69 | 0,40 |
| Sem *outliers* | $3 \times (256)$ | (0,3),(0,15) | 0,69 | 0,50 |

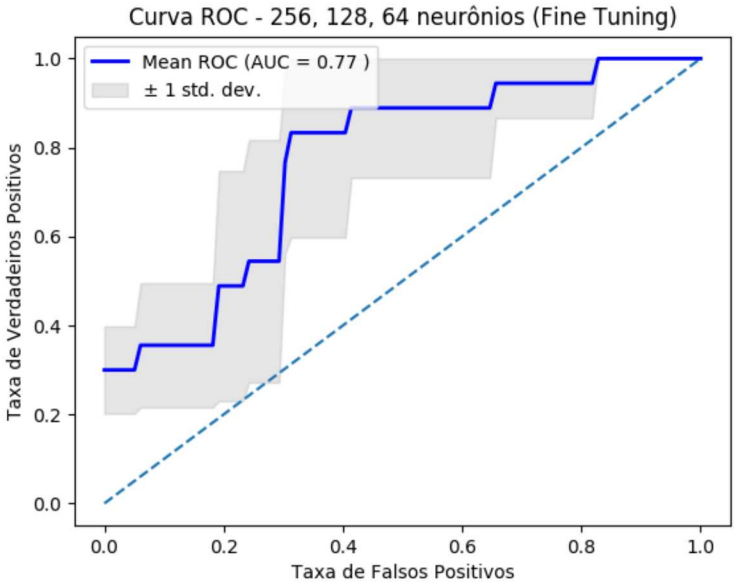
Fonte: Produção do próprio autor.

#### 4.7.3.2 Usando o modelo pré-treinado VGG16

A seguir são expostos os dois melhores resultados dos experimentos com a rede VGG16. O corte na VGG16 foi realizado na camada $15$ já que é a última camada convolucional do modelo base.

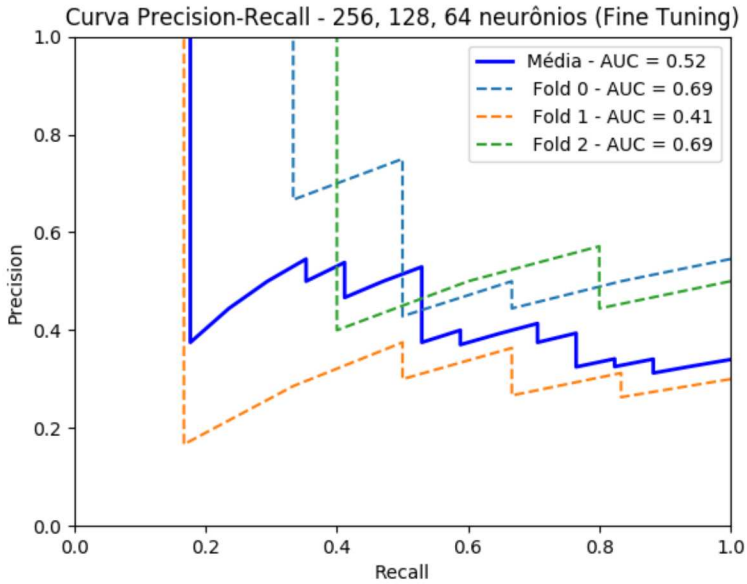
Os melhores resultados obtidos, sem os *outliers*, após diversas configurações de arquiteturas podem ser visualizados nos Gráficos 9, 10, 11 e 12. Note que ambos os resultados apresentam valores altos de AUC quando comparados com todos os outros experimentos do projeto desenvolvido, fato que enfatiza a facilidade de trabalhar com modelos de profundidade média. Além disso, nos Gráficos 11 e 12 é possível notar que no *Fold*2 a AUC da curva de P-S foi $1,0$, o que significa que naquela partição de validação todas as amostras da classe determinada como positiva (*Muscular Men's Physique*) foram preditas corretamente.

Gráfico 9 – Curva ROC - Resultado de experimento com VGG16 - Três camadas densas de (256, 128, 64) neurônios e duas de *dropout* (0,5)



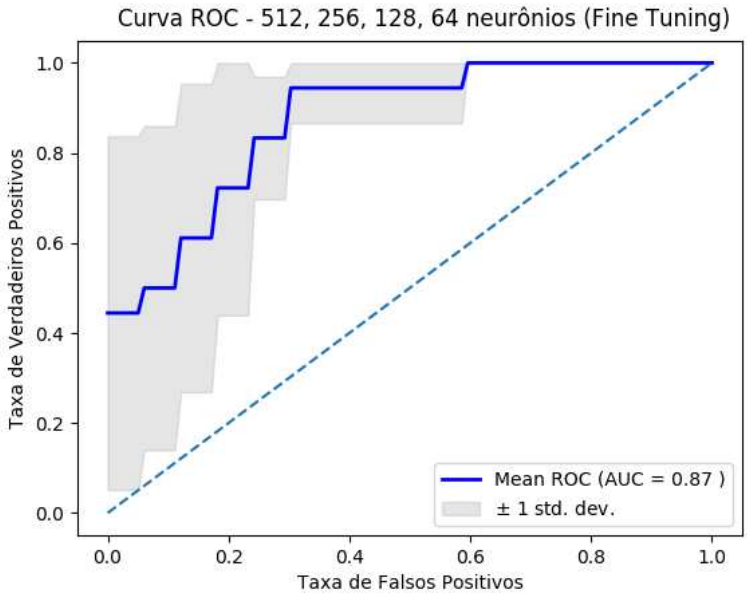

Fonte: Produção do próprio autor.

Gráfico 10 – Curva P-S - Resultado de experimento com VGG16 - Três camadas densas de (256, 128, 64) neurônios e duas de *dropout* (0,5)



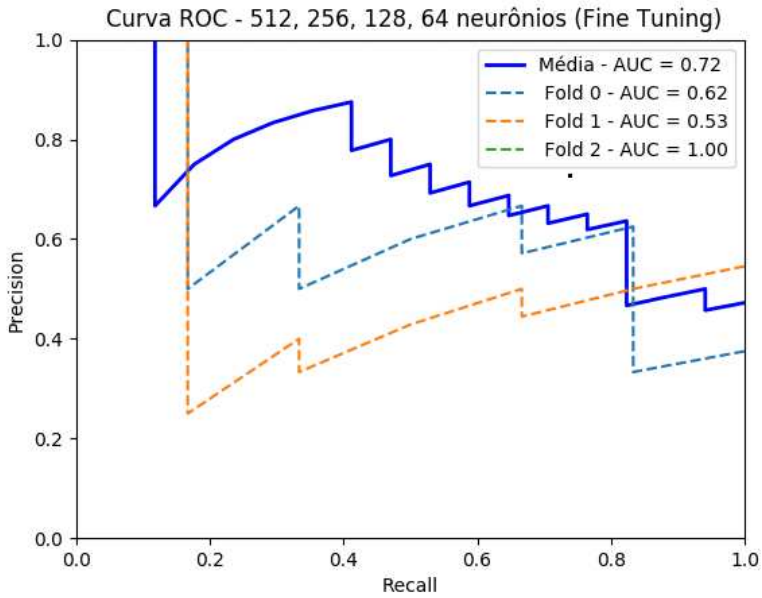

Fonte: Produção do próprio autor.

Gráfico 11 – Curva ROC - Resultado de experimento com VGG16 - Quatro camadas densas de (512, 256, 128, 64) neurônios e três de *Dropout* $1 \times (0,6)$, $2 \times (0,5)$




Fonte: Produção do próprio autor.

Gráfico 12 – Curva P-S - Resultado de experimento com VGG16 - Quatro camadas densas de (512, 256, 128, 64) neurônios e três de *Dropout* $1 \times (0,6)$, $2 \times (0,5)$




Fonte: Produção do próprio autor.

Por fim, no Quadro 3 são expostas as configurações de camadas densas e os AUC obtidos em cada curva.

Quadro 3 – Melhores resultados obtidos com VGG16

| $D$ | $p$ | AUC ROC | AUC P-S |
|---|---|---|---|
| 256, 128, 64 | (0,5), (0,5), (0,5) | 0,77 | 0,52 |
| 512, 256, 128, 64 | (0,6),(0,5),(0,5) | 0,87 | 0,72 |

Fonte: Produção do próprio autor.

# 5 CONCLUSÕES E PROJETOS FUTUROS

## 5.1 Conclusões

O objetivo principal deste trabalho foi propor um preditor de classes binário para enquadramento de atletas de fisiculturismo na modalidade *Men's Physique* e *Muscular Men's Physique*. A técnica proposta para resolução do problema é baseada fundamentalmente na teoria da utilização de modelos pré-treinados de redes neurais convolucionais.

No decorrer dos experimentos notou-se uma grande dificuldade em aplicar *fine tuning* de maneira eficaz nas redes com maior número de camadas convolucionais. Assim, a partir dos resultado, conclui-se que as redes pré-treinadas com menor número de camadas convolucionais comportaram-se como melhores preditores visto as características do banco de dados disponível. A Tabela 3 mostra que os melhores resultados alcançados ao fim do projeto em questão foram com o modelo pré-treinado VGG16. Além disso, ressalta-se a importância da escolha correta da métrica de avaliação e das técnicas de regularização para trabalhar com dados com classes desbalanceadas. Como por exemplo de técnica de regularização pode-se citar o uso do *Data Augmentation*, que possibilitou a expansão de um pequeno banco de dados aplicando transformações nas imagens originais.

Ao fim deste trabalho conclui-se que a curva P-S é realmente mais informativa para trabalhar com dados que apresentam as características do banco de dados construído, uma vez que a curva utilizada como base para análise do modelo, a linha de base, adapta-se à distribuição de classes de cada banco. Em suma, o trabalho em questão mostra que com conhecimento acerca das ferramentas e técnicas de *Deep Learning* disponíveis atualmente, é possível implementar e treinar redes de predição binária mesmo com um pequeno número de amostras distribuídas de maneira desigual entre suas classes.

## 5.2 Temas a Serem Pesquisados

Como trabalhos futuros serão pesquisadas técnicas de *ensembles* a fim de obter resultados ainda mais satisfatórios utilizando o banco de dados disponível. A implementação de um *ensemble* basicamente consiste no agrupamento de múltiplos submodelos já treinados para construir um preditor a partir dessa combinação. Assim, o intuito seria implementar um preditor que utilizasse

como submodelos os dois modelos que apresentaram os melhores resultados do projeto. O estudo de técnicas de aprimoramento do *fine-tuning* para conseguir utilizar redes mais densas com pequenos bancos de dados também é relevante.

Embora existam imagens externas de atletas das duas modalidades abordadas, outro ponto interessante a ser desenvolvido é a extensão do banco de dados. Para isso será necessário acompanhar a pesagem dos próximos campeonatos promovidos pela IFBB - Espírito Santo e assim adquirir novas imagens.

Por fim, implementação de uma metodologia de exclusão automática de *outliers* também é um importante trabalho a ser desenvolvido, haja vista que assim não será mais necessário o auxílio de integrantes na arbitragem da IFBB o que tornará o trabalho menos sujeito à incerteza humana. Para isso, a literatura já disponibiliza técnicas para que uma rede gere faixas de incerteza via *dropout*, como por exemplo o Monte-Carlo *Dropout* (GAL; GHAHRAMANI, 2016). Essas faixas são importantes para modelos que trabalham com poucos dados uma vez que é uma forma eficiente de eliminar *outliers*.

# REFERÊNCIAS BIBLIOGRÁFICAS


ALMONFREY, D.; CARMO, A. P.; QUEIROZ, F. M.; PICORETI, R.; VASSALLO, R. F.; SALLES, E. O. T. A flexible human detection service suitable for intelligent spaces based on a multi-camera network. **International Journal of Distributed Sensor Networks**, SAGE Publications Sage UK: London, England, v. 14, n. 3, p. 1550147718763550, 2018.

BERTÃO, N. **Brasil já é um dos maiores mercados "fitness" do mundo**. 2016. Disponível em: https://exame.abril.com.br/revista-exame/brasil-ja-e-um-dos-maiores-mercados-fitness-domundo/. Acesso em: 17 out. 2018.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. [S.l.]: Springer, 2006.

BLOICE, M. D. **Image augmentation library in Python for machine learning.** 2016. Disponível em: https://github.com/mdbloice/Augmentor. Acesso em: 10 nov. 2019.

BROWNLEE, J. **How to Use ROC Curves and Precision-Recall Curves for Classification in Python**. 2018. Disponível em: https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/. Acesso em: 1 nov. 2019.

BROWNLEE, J. **How to Choose Loss Functions When Training Deep Learning Neural Networks**. 2019. Disponível em: https://machinelearningmastery.com/how-tochoose-loss-functions-when-training-deep-learning-neural-networks/. Acesso em: 27 out. 2019.

CAO, Z.; SIMON, T.; WEI, S.-E.; SHEIKH, Y. Realtime multi-person 2d pose estimation using part affinity fields. *In*: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2017, Havaí. **Proceedings […]**. Havaí: 2017 p. 7291–7299.

CASTRO, A. P. A.; DAMASCENO, V. O.; MIRANDA, J. A.; LIMA, J. R. P.; VIANNA, J. M. Fotossilhuetas para avaliação da imagem corporal de fisiculturistas. **Revista Brasileira de Medicina do Esporte**, v. 17, n. 4, p. 250–253, 2011.

CHOLLET, F. Xception: Deep learning with depthwise separable convolutions. *In*: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2017, Havaí. **Proceedings […]**. Havaí: 2017 p. 1251–1258.

CIRESAN, D. C.; MEIER, U.; MASCI, J.; GAMBARDELLA, L. M.; SCHMIDHUBER, J. Flexible, high performance convolutional neural networks for image classification. *In*: TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 2011, Barcelona. **Proceedings […]**. Barcelona: 2017.

GAL, Y.; GHAHRAMANI, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *In*: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 2016, Nova Iorque. **Proceedings […]**. Nova Iorque: 2016. p. 1050–1059.

GODOY, D. **Understanding binary cross-entropy / log loss: a visual explanation**. 2018. Disponível em: https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a. Acesso em: 27 out. 2019.

GONÇALVES, T. O.; CIARELLI, P. M.; COCO, K. F.; GONÇALVES, E. C.; CASTRO, J. W. B. Uma análise da viabilidade de estimar somatotipos de fisiculturistas através de medições em imagens. *In:* CONGRESSO BRASILEIRO DE AUTOMÁTICA - CBA, 2016, Vitória. **Anais [...]**. Vitória: 2016.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. [S.l.]: MIT press, 2016.

HAYKIN, S. **Neural Networks and Learning Machines, 3/E.** [S.l.]: Pearson Education India, 2010.

HE, H.; GARCIA, E. A. Learning from imbalanced data. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, v. 21, n. 9, p. 1263–1284, 2009.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. *In*: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2016, Las Vegas. **Proceedings [...]**. Las Vegas: 2016. p. 770–778.

HUANG, G.; LIU, Z.; MAATEN, L. V. D.; WEINBERGER, K. Q. Densely connected convolutional networks. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2017, Havaí. **Proceedings [...]**. Havaí: 2017. p. 4700–4708.

IFBB. **Introdução às Modalidades da IFBB**. 2017. Disponível em: https://ifbbbrasil.com.br/modalidades/. Acesso em: 21 out. 2019.

INTERNATIONAL FEDERATION OF BODYBUILDING & FITNESS. **Rules for bodybuilding and fitness**. Madrid: IFBB, 2019.

KIM, P. **Matlab deep learning. With Machine Learning, Neural Networks and Artificial Intelligence**, Springer, v. 130, 2017.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of IEEE**, Taipei, Taiwan, v. 86, n. 11, p. 2278–2324, 1998.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The Bulletin of Mathematical Biophysics**, Springer, v. 5, n. 4, p. 115–133, 1943.

PAN, S. J.; KWOK, J. T.; YANG, Q. Transfer learning via dimensionality reduction. *In*: ASSOCIATION FOR THE ADVANCEMENT OF ARTIFICIAL INTELLIGENCE, 2008, Chicago. **Proceedings [...]**. Chicago: 2008. p. 677–682.

QUEIROZ, F. M. **Skeletons Detection Service**. 2018. Disponível em: https://github.com/labviros/is-skeletons-detector/tree/openpose. Acesso em: 9 nov. 2019.

RADHAKRISHNAN, P. **What is Transfer Learning?** 2017. Disponível em: https://towardsdatascience.com/what-is-transfer-learning-8b1a0fa42b4. Acesso em: 3 nov. 2019.

SAITO, T.; REHMSMEIER, M. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. **PloS one, Public Library of Science**, v. 10, n. 3, p. e0118432, 2015.

SCHWARZENEGGER, A.; DOBBINS, B. **The new encyclopedia of modern bodybuilding**. [S.l.]: Simon and Schuster, 1998.

SILVA, P. R. P.; TRINDADE, R. S.; ROSE, E. H. Composição corporal, somatotipo e proporcionalidade de culturistas de elite do brasil. **Revista Brasileira de Medicina do Esporte**, SciELO Brasil, v. 9, p. 403–7, 2003.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition**.** *In*: INTERNATIONAL CONFERENCE ON LEARNING REPRESENTATIONS, 2015, San Diego. **Proceedings […]**. San Diego: 2015.

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. **The Journal of Machine Learning research, JMLR**. v. 15, n. 1, p. 1929–1958, 2014.

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. Going deeper with convolutions. *In*: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2015, Boston. **Proceedings […]**. Boston: 2015. p. 1–9.

SZEGEDY, C.; VANHOUCKE, V.; IOFFE, S.; SHLENS, J.; WOJNA, Z. Rethinking the inception architecture for computer vision. *In*: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2016, Las Vegas. **Proceedings […]**. Las Vegas: 2016. p. 2818–2826.

# APÊNDICE A – INTRODUÇÃO AO APRENDIZADO DE MÁQUINAS

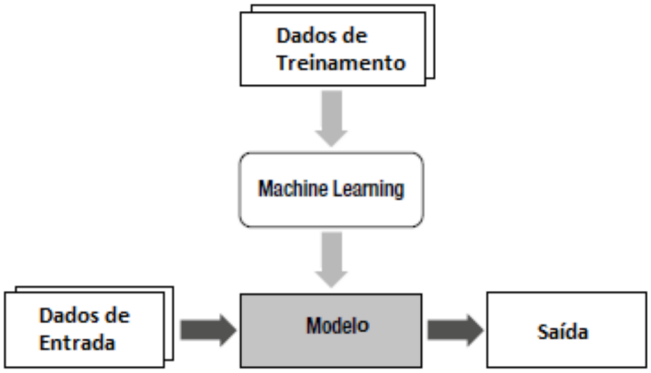
## A.1 *Machine Learning*

O *Machine Learning* consiste em um específico método de análise de dados pertencente à área de Inteligência Artificial, no qual o objetivo é a busca pela automatização da construção de modelos analíticos que não estão facilmente disponíveis.

O processo de aprendizagem, ilustrado pelo sentido vertical da Figura A1, acontece quando um sistema computacional processa determinada entrada, chamada de Dados de Treinamento, e identifica características medidas quantitativamente que são utilizadas na construção de um modelo para tomada de decisão, com o mínimo de intervenção humana.

O sentido horizontal da Figura A1 representa o modelo já treinado, no qual um dado de entrada, diferente dos utilizados para aprendizagem, é fornecido ao sistema que o analisa e gera uma saída. Vale ressaltar que por se tratar de um método iterativo, a partir do momento em que o modelo é exposto a novos dados, adaptações são feitas a fim de produzir resultados confiáveis. Dessa forma, o ML consiste essencialmente em uma forma computacional aplicada de métodos estatísticos que visa a obtenção de funções com modelagens matematicamente complicadas (GOODFELLOW; BENGIO; COURVILLE, 2016). A maior parte dos algoritmos de ML podem ser divididos em duas categorias: Aprendizagem supervisionada e aprendizagem não supervisionada.

Figura A1 – Processo de aprendizagem




Fonte: Kim (2017).
Nota: Adaptado pelo autor.

### A.1.1 Aprendizagem supervisionada

Esse algoritmo de *Machine Learning* busca o aprendizado a partir de um conjunto rotulado de dados que permite a construção de um novo modelo. Na aprendizagem supervisionada cada dado de treino deve apresentar um par, equação (A.1), correspondente, no qual a $sadacorreta$ consiste no resultado que o modelo deveria gerar, ou seja, a solução esperada do problema. Assim, uma série de revisões do modelo é feita a fim de reduzir as diferenças entre a saída gerada pelo modelo e a $sadacorreta$ esperada para uma mesma entrada.

$$z(x) = (entrada, sadacorreta) \tag{A.1}$$

### A.1.2 Aprendizagem não supervisionada

Baseia-se na descoberta de padrões intrínsecos nos dados a partir de conjuntos não rotulados. Cada elemento do conjunto de dados de treino apresenta apenas a entrada, ou seja, não se sabe de antemão qual deve ser a solução gerada pelo modelo.

## A.2 *Deep Learning*

De acordo com (HAYKIN, 2010), o cérebro é um computador paralelo, não linear, altamente complexo que possui a capacidade de processar informações a partir de uma organização estrutural muito mais veloz do que a dos computadores mais modernos da atualidade. Pertencente à família de métodos de *Machine Learning*, o *Deep Learning*, ou Aprendizado Profundo, tem por inspiração o funcionamento hierarquizado do córtex áudio/visual do cérebro humano.

A hierarquização é feita a partir de uma divisão por camadas, na qual quanto maior a profundidade, maior é a complexidade das informações processadas. Assim, o *Deep Learning* utiliza um conjunto de algoritmos que tentam modelar abstrações de alto nível de dados usando um grafo profundo com várias camadas de processamento, compostas de várias transformações lineares e não lineares. Cada camada sucessiva utiliza a saída da anterior como entrada da mais profunda. Em suma, níveis mais profundos são oriundos dos mais superficiais, fato que origina a representação hierárquica semelhante a do córtex áudio/visual.

Assim, divergindo da maior parte dos métodos de ML, o DL não faz uso de extratores de características específicos, projetados manualmente, uma vez que apresenta uma estrutura flexível de aprendizagem.

## A.3 Classificação Binária

Basta uma breve pesquisa acerca de abordagens das técnicas de ML para perceber que existem muitas aplicações envolvendo os tarefas de classificação. Assim, como o projeto trata-se da classificação de atletas em duas distintas modalidades de fisiculturismo, uma abordagem mais detalhada acerca dos modelos para resolução de problemas de classificação binária se faz necessária. Neste tipo de tarefa, o programa computacional busca especificar em qual das $k$ categorias um dado de entrada pertence (GOODFELLOW; BENGIO; COURVILLE, 2016).

Um classificador pode ser definido de maneira simplista, equação (A.2), como a composição de duas funções: função discriminante e função de ativação. A **função discriminante**, representada por $h(\mathbf{x}; \mathbf{w})$ onde $\mathbf{x}$ é a entrada e $\mathbf{w}$ são os parâmetros, permite que o algoritmo decida a que classe $k$ o vetor $\mathbf{x}$ de entrada pertence (BISHOP, 2006) enquanto que a **função de ativação**, representada por $f(x)$, é responsável por representar a regra de decisão do classificador.

$$y = f(h(\mathbf{x}; \mathbf{w})) \tag{A.2}$$

Nota-se a partir da equação (A.2) que a função discriminante é dependente de um parâmetro desconhecido $\mathbf{w}$. Diante disso, cabe ao algoritmo de aprendizagem determinar o valor desse parâmetro que minimize a função de custo, equação (A.4), na qual $N$ representa o tamanho do conjunto de dados. A equação (A.3) é a representação do conjunto de dados de treinamento, no qual $\mathbf{x}$ é o ponto observado e $\mathbf{y}$ é o rótulo correspondente.

$$\tau = (\mathbf{x_i}; \mathbf{y_i})_{(\mathbf{i}=1,\ldots,N)} \tag{A.3}$$

$$L(\tau, \mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} loss(\hat{h}_i, y_i) \tag{A.4}$$

### A.3.1 Função discriminante

Conforme o exposto na Seção A.3, uma função discriminante é responsável por atribuir um vetor de entrada a uma determinada classe. Uma representação simplista de uma função **discriminante linear** é dada pela equação A.5.

$$h(\mathbf{x}; \mathbf{w}) = \mathbf{w}^T \mathbf{x} \tag{A.5}$$

$$h(\mathbf{x}; \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + ... + w_n x_n \tag{A.6}$$

**w** é chamado de vetor peso e é ortogonal a todos os vetores que estão na superfície de separação. Consequentemente seu valor determina a orientação do hiperplano. Já $w_0$ é um valor escalar denominado bias do hiperplano e determina a localização da superfície de decisão. Além disso, evita que todos os hiperplanos passem pela origem (BISHOP, 2006).

Entretanto, quando a função discriminante trata-se de uma **função discriminante logística**, a representação baseia-se em uma sigmoid, conforme pode-se averiguar na equação (A.7).

$$h(\mathbf{x}; \mathbf{w}) = \frac{e^{\mathbf{w}^T \mathbf{x}}}{1 + e^{\mathbf{w}^T \mathbf{x}}} \tag{A.7}$$

Como a equação (A.7) é normalizada, essa função caracteriza-se pela transformação dos valores da função linear em valores de probabilidades. Tal característica é de suma importância, uma vez que em certos casos não é possível determinar uma fronteira de decisão que separe de maneira exata os dados de entrada. Diante disso, se faz de grande valia o retorno da classificação em forma de probabilidade de acordo com a respectiva proximidade de cada entrada com a fronteira de decisão.

### A.3.2 Função de perda

Conforme exposto na Seção A.3, o objetivo da aprendizagem em um problema de classificação consiste em minimizar a função de custo, equação (A.4). Contudo, para que seja possível atender ao objetivo proposto, deve-se ponderar o erro entre o rótulo de treinamento e o rótulo correspondente predito para todas as amostras. Para isso, se faz necessária a escolha de uma função *loss*, responsável por estimar a perda de um modelo a fim de reduzir os erros no decorrer das iterações de uma rede neural.

Redes neurais usualmente caracterizam-se pelo aprendizado a partir do mapeamento de exemplos de entradas com suas respectivas saídas (BROWNLEE, 2019). Dessa forma, a escolha da função de perda que esteja de acordo com a modelagem do problema de predição específico se faz de grande valia.

#### A.3.2.1 Entropia cruzada binária

Para problemas de classificação binária, a função de perda padrão é a de entropia cruzada. Essa função, equação (A.8), tem por objetivo o cálculo de um fator que representa a diferença entre as distribuições de probabilidades reais e preditas para prever determinada classe (GODOY, 2018). Assim, pode-se definir a entropia cruzada como uma maneira de quantificar a diferença entre as medidas de probabilidades das duas classes.

$$loss_{ent}(h(\mathbf{x}_i;\mathbf{w}), y_i) = \begin{cases} -log(h(\mathbf{x}_i;\mathbf{w})), & y_i = 1 \\ -log(1 - h(\mathbf{x}_i;\mathbf{w})), & y_i = 0 \end{cases} \tag{A.8}$$

#### A.3.2.2 Entropia cruzada binária ponderada

Para problemas com dados desbalanceados, adicionam-se os pesos das classes na função de custo. Intuitivamente, deseja-se atribuir maior peso à classe minoritária e menor peso à classe majoritária.

A equação (A.9) representa a perda por entropia cruzada binária ponderada, na qual $w_0$ e $w_1$ são

os pesos das classes 0 e 1 respectivamente.

$$loss_{entpod}(h(\mathbf{x}_i;\mathbf{w}), y_i) == \begin{cases} -w_1 log(h(\mathbf{x}_i;\mathbf{w})), & y_i = 1 \\ -w_0 log(1 - h(\mathbf{x}_i;\mathbf{w})), & y_i = 0 \end{cases} \tag{A.9}$$

# APÊNDICE B – VALIDAÇÃO CRUZADA - *K-FOLD*

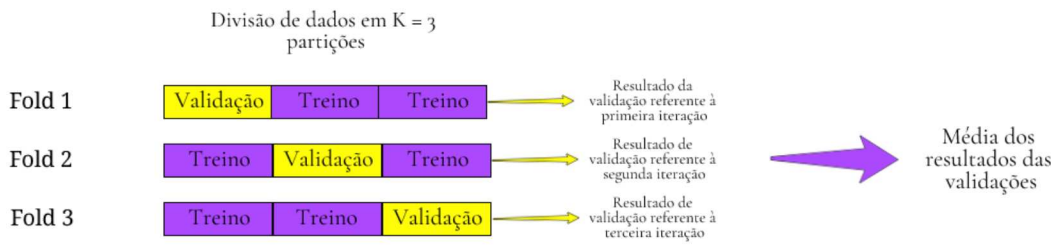
A validação cruzada consiste em um método de análise acerca do poder de generalização de um modelo a um banco de dados independente. Usualmente a construção de um algoritmo de ML inicia-se a partir da partição dos dados em um conjunto de treinamento e um de validação. Entretanto, no decorrer da validação pode-se notar alguns problemas como *underfitting* e *overfitting*. O *underfitting* ocorre quando tem-se um modelo demasiado simples e diante disso não é capaz de identificar padrões no banco de dados. Modelos com *underfitting* apresentam alto erro de treino e alto erro de validação. Já o *overfitting* é basicamente o oposto, ou seja, ocorre quando o modelo apresenta alta complexidade e assim tende a se ajustar do forma exagerada aos detalhes e ruídos do banco de dados. Assim, o modelo tende a retornar predições corretas para o banco de dados original do problema, todavia quando exposto a novos dados, não se comporta de maneira eficaz e apresenta alto erro de validação e baixo erro de treino.

Problemas como esses tendem a ocorrer principalmente quando não há grande disponibilidade de dados, uma vez que os resultados de validação podem apresentar divergências significativas mediante à escolha do do conjunto de treino e de validação. Todavia, ao utilizar a validação cruzada por *K-fold* essa problemática pode ser facilmente contornada.

A validação cruzada por *K-fold* é um procedimento de divisão do conjunto de amostra em treino e validação em $K$ partições criadas randomicamente. Em seguida, instancia-se $K$ iterações idênticas nas quais $K - 1$ partições são utilizadas para treinamento enquanto que a remanescente é utilizada para validação. O resultado final de validação é obtido a partir da média dos resultados de validação de cada iteração.

No exemplo exposto pela Figura A2 tem-se $K = 3$, portanto o banco de dados será dividido em 3 partições, sendo 2 para treino e 1 para validação. Além disso, o modelo é instanciado 3 vezes, ou seja, ocorrerá uma iteração em cada *Fold*, o que acarreta em $K$ diferentes resultados. Por fim, tem-se o valor final de validação do modelo a partir da média os resultados obtidos anteriormente.

Figura A2 – Exemplo de modelo de validação cruzada com *K-Fold*




Fonte: Produção do próprio autor.

# APÊNDICE C – AQUISIÇÃO DAS IMAGENS DO CONJUNTO DE DADOS

O primeiro passo para o desenvolvimento do projeto proposto foi a aquisição das imagens. Por se tratar de uma demanda apresentada pela IFBB, as fotos foram capturadas nas pesagens de dois campeonatos organizados pela IFBB - Espírito Santo:

- 3ª Copa Capixaba de *Bodybuilding* e *Fitness*, realizada no dia 10 de novembro de 2018.
- Campeonato Estadual de Estreantes, realizado no dia 11 de maio de 2019.

A pesagem dos atletas ocorre no dia que antecede o campeonato. Nesse momento os atletas são medidos e alocados nas respectivas categorias de cada modalidade, seguindo os critérios da IFBB. A primeira aquisição de imagens ocorreu no dia 9 de novembro de 2018 enquanto que a segunda ocorreu no dia 10 de maio de 2019, ambas no Centro de Convenções de Vitória.

Como forma de padronizar o protocolo de aquisição de imagens, foi solicitado que os atletas reproduzissem a pose oficial "De Frente"para os árbitros. Além disso, optou-se por sempre utilizar um fundo branco com iluminação superior frontal, a fim de evitar a formação de sombras. A câmera utilizada é de domínio do Diretor Administrativo da IFBB - Espírito Santo, que também foi o responsável pela aquisição das imagens.

As imagens da primeira aquisição, Figura A3, apresentavam resolução 1638x2048 *pixels*, enquanto que as da segunda, Figura A3, possuíam resolução 2400x3000 *pixels*, ambas com formato JPG. Essa divergência de resoluções se deve ao fato de que as imagens da primeira aquisição foram recortadas pelo fotógrafo/Diretor Administrativo antes de serem fornecidas para o desenvolvimento do trabalho.

Figura A3 – Exemplos de imagens de cada aquisição

(a) Primeira aquisição

(b) Segunda aquisição

Fonte: Produção do próprio autor.

# ANEXO A – AUTORIZAÇÃO DO USO DAS IMAGENS DOS ATLETAS DA IFBB - ES

FEDERAÇÃO CAPIXABA DE BODYBUILDING E FITNESS
IFBB-ES

## AUTORIZAÇÃO

Vitória, 03 de dezembro de 2019.

A IFBB-ES no uso de suas atribuições definidas no Estatuto da IFBB-ES e no que estabelece o Livro de Regras da IFBB Internacional, edição de 2018, na seção 1, artigo 7, autoriza o uso das imagens de seus atletas na pesquisa de **Junia Gabriela Lube Pimentel** no curso de Engenharia Elétrica da Universidade Federal do Espírito Santo.

A IFBB-ES ressalta que tal autorização se estabelece mediante a devida comunicação dos resultados do estudo a esta Federação bem como a menção desta nas publicações resultantes deste trabalho.

Atenciosamente,

Andreia Borges
PRESIDENTE DA IFBB-ES