Pontifícia Universidade Católica
do Rio de Janeiro

Leonardo Cardia da Cruz

Habilitando Anotações de Dados Autônomos: Uma Abordagem de Aprendizado por Reforço com Humano no Loop

Tese de Doutorado

Tese apresentada como requisito parcial para a obtenção do grau de Doutor pelo Programa de Pós–graduação em Informática do Departamento de Informática da PUC-Rio.

Orientador : Prof. Alberto Barbosa Raposo
Co-orientador: Dr. Cesar Augusto Sierra Franco

Rio de Janeiro
Setembro de 2022

Pontifícia Universidade Católica
do Rio de Janeiro

Leonardo Cardia da Cruz

Habilitando Anotações de Dados Autônomos: Uma Abordagem de Aprendizado por Reforço com Humano no Loop

Tese apresentada como requisito parcial para a obtenção do grau de Doutor pelo Programa de Pós–graduação em Informática do Departamento de Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo.

Prof. Alberto Barbosa Raposo
Orientador
Departamento de Informática – PUC-Rio

Dr. Cesar Augusto Sierra Franco
Co-orientador
Departamento de Informática – PUC-Rio

Dr. Luiz Jose Schirmer Silva
Departamento de Informática – UC

Dr. Jan Jose Hurtado Jauregui
Departamento de Informática – PUC-Rio

Profª. Sandra Eliza Fontes de Avila
Departamento de Sistemas de Informação – Unicamp

Prof. Anselmo Cardoso de Paiva
Departamento de Informática – UFMA

Rio de Janeiro, 30 de Setembro de 2022

Leonardo Cardia da Cruz

Graduou-se em Tecnologia da Informação pelo Instituto Superior de Tecnologia de Petrópolis (2014) e obteve o grau de Mestre em Ciências e Sistemas da Computação pelo Instituto Militar de Engenharia - IME (2016).

Ficha Catalográfica

da Cruz, Leonardo Cardia

Habilitando Anotações de Dados Autônomos: Uma Abordagem de Aprendizado por Reforço com Humano no Loop / Leonardo Cardia da Cruz; orientador: Alberto Barbosa Raposo; co-orientador: Cesar Augusto Sierra Franco. – 2022.
109 f.: il. color. ; 30 cm

Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2022.
Inclui bibliografia

1. Informática – Teses. 2. Aprendizado por Reforço Profundo;. 3. Caixa Delimitadora;. 4. Conjunto de Dados;. 5. Aconselhamento;. 6. Deep Q-Network;. 7. Agente Virtual;. 8. Anotações;.
I. Raposo, Alberto Barbosa. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Agradecimentos

A elaboração de uma tese de doutorado foi uma das tarefas mais desafiadoras que particularmente já enfrentei. Algumas pessoas me ajudaram durante esta caminhada, e gostaria de deixar registrado nesse trabalho os meus agradecimentos.

Agradeço primeiramente a Deus, por ser o condutor de todos os meus passos.

À minha esposa Mêire, sempre muito sábia, me encorajou nos momentos de desânimos, me compreendeu nas ausências devido aos estudos e pesquisas. Mesmo quando estava passando por um milagre em sua vida, suas preocupações estavam dirigidas ao nosso sonho. Portanto, o meu muito obrigado meu Amor! Você foi, é e sempre será a minha fonte de inspiração. Como o significado de seu nome já diz, você é aquela que ilumina a minha vida. Te Amo Muito!

Aos meus pais José Luiz Alves e Maria Lúcia, o meu muito obrigado. Ter vocês como pais, é uma das maiores alegrias da minha vida. Agradeço por nunca terem medido esforços para que juntamente com meus irmãos, pudessemos sonhar e correr atrás de todos os nossos sonhos. Vocês são e sempre serão a minha base!

Aos meus irmãos Elizabeth e José Luiz da Cruz, que sempre me apoiaram, compreenderam e incentivaram em toda minha trajetória. Me faltam palavras para externar o quanto sou grato, por tê-los como irmãos e por suas orações.

Aos meus sobrinhos Jonathan e Isaac, os momentos de descontração foram importantíssimos para minha firmeza até aqui.

Aos professores da PUC-Rio e secretária do DI, em especial meu orientador prof. Alberto Raposo, por me atender e orientar sempre que precisei. Sua amizade e paciência contribuiram de forma decisiva para o meu crescimento acadêmico.

Ao casal de amigos Dr. Cesar Franco e Dra. Greis, pelas valiosas sugestões que me direcionaram nas etapas desta pesquisa.

Aos casais de amigos, Dr. Douglas e Msc. Juliana, pelos conselhos e trocas de experiência que me ajudaram em momentos de dificuldade. Sempre se mostraram disponíveis nos mais diversos momentos. Aos casais de amigos Eduardo e Quézia, Junior e Thalia, pelo apoio e incentivo durante essa jornada. Vocês são especiais para mim e para minha família!

À todas as pessoas que de alguma forma me ajudaram ou me incentivaram, em especial minha cunhada Rafaela e minha sogra Maria.

Por último, mas não menos importante, gostaria de agradecer a CAPES pelo apoio financeiro e à PUC-Rio pela bolsa de isenção de mensalidades do doutorado.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Resumo

da Cruz, Leonardo Cardia; Raposo, Alberto Barbosa; . **Habilitando Anotações de Dados Autônomos: Uma Abordagem de Aprendizado por Reforço com Humano no Loop**. Rio de Janeiro, 2022. 109p. Tese de Doutorado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

As técnicas de aprendizado profundo têm mostrado contribuições significativas em vários campos, incluindo a análise de imagens. A grande maioria dos trabalhos em visão computacional concentra-se em propor e aplicar novos modelos e algoritmos de aprendizado de máquina. Para tarefas de aprendizado supervisionado, o desempenho dessas técnicas depende de uma grande quantidade de dados de treinamento, bem como de dados rotulados. No entanto, a rotulagem é um processo caro e demorado. Uma recente área de exploração são as reduções dos esforços na preparação de dados, deixando-os sem inconsistências, ruídos, para que os modelos atuais possam obter um maior desempenho. Esse novo campo de estudo é chamado de *Data-Centric IA*. Apresentamos uma nova abordagem baseada em Deep Reinforcement Learning (DRL), cujo trabalho é voltado para a preparação de um conjunto de dados em problemas de detecção de objetos, onde as anotações de caixas delimitadoras são feitas de modo autônomo e econômico. Nossa abordagem consiste na criação de uma metodologia para treinamento de um agente virtual a fim de rotular automaticamente os dados, a partir do auxílio humano como professor desse agente. Implementamos o algoritmo *Deep Q-Network* para criar o agente virtual e desenvolvemos uma abordagem de aconselhamento para facilitar a comunicação do humano professor com o agente virtual estudante. Para completar nossa implementação, utilizamos o método de aprendizado ativo para selecionar casos onde o agente possui uma maior incerteza, necessitando da intervenção humana no processo de anotação durante o treinamento. Nossa abordagem foi avaliada e comparada com outros métodos de aprendizado por reforço e interação humano-computador, em diversos conjuntos de dados, onde o agente virtual precisou criar novas anotações na forma de caixas delimitadoras. Os resultados mostram que o emprego da nossa metodologia impacta positivamente para obtenção de novas anotações a partir de um conjunto de dados com rótulos escassos, superando métodos existentes. Desse modo, apresentamos a contribuição no campo de Data-Centric IA, com o desenvolvimento de uma metodologia de ensino para criação de uma abordagem autônoma com aconselhamento humano para criar anotações econômicas a partir de anotações escassas.

Palavras-chave

Aprendizado por Reforço Profundo; Caixa Delimitadora; Conjunto de Dados; Aconselhamento; Deep Q-Network; Agente Virtual; Anotações;

Abstract

da Cruz, Leonardo Cardia; Raposo, Alberto Barbosa (Advisor); (Co-Advisor). **Enabling Autonomous Data Annotation: A human-in-the-loop Reinforcement Learning Approach**. Rio de Janeiro, 2022. 109p. Tese de doutorado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Deep learning techniques have shown significant contributions in various fields, including image analysis. The vast majority of work in computer vision focuses on proposing and applying new machine learning models and algorithms. For supervised learning tasks, the performance of these techniques depends on a large amount of training data and labeled data. However, labeling is an expensive and time-consuming process.

A recent area of exploration is the reduction of efforts in data preparation, leaving it without inconsistencies and noise so that current models can obtain greater performance. This new field of study is called Data-Centric AI. We present a new approach based on Deep Reinforcement Learning (DRL), whose work is focused on preparing a dataset, in object detection problems where the bounding box annotations are done autonomously and economically. Our approach consists of creating a methodology for training a virtual agent in order to automatically label the data, using human assistance as a teacher of this agent.

We implemented the Deep Q-Network algorithm to create the virtual agent and developed a counseling approach to facilitate the communication of the human teacher with the virtual agent student. We used the active learning method to select cases where the agent has more significant uncertainty, requiring human intervention in the annotation process during training to complete our implementation. Our approach was evaluated and compared with other reinforcement learning methods and human-computer interaction in different datasets, where the virtual agent had to create new annotations in the form of bounding boxes. The results show that the use of our methodology has a positive impact on obtaining new annotations from a dataset with scarce labels, surpassing existing methods. In this way, we present the contribution in the field of Data-Centric AI, with the development of a teaching methodology to create an autonomous approach with human advice to create economic annotations from scarce annotations.

Keywords

Deep Reinforcement Learning; Bounding Box Datasets; Advices; Deep Q-Network; Virtual Agent; Annotations;

Sumário

1 Introdução **18**
1.1 Contexto 18
1.2 Motivação 19
1.3 Definição do problema 20
1.4 Justificativa 20
1.5 Hipótese 21
1.6 Objetivos 21
1.7 Contribuições 21
1.8 Organização 22

2 Fundamentação Teórica **23**
2.1 Aprendizado por Reforço 23
2.1.1 Processo de decisão de Markov 26
2.1.2 Algoritmos de aprendizagem 28
2.2 Aprendizado por Reforço Profundo 31
2.2.1 Deep Q-Network (DQN) 36
2.3 Aprendizagem com dados parcialmente anotados 41
2.4 Aprendizagem Ativa 43

3 Trabalhos Relacionados **46**
3.1 Aprendizado por Reforço com Humano no Loop 46
3.2 Redução de custo em anotações 51
3.3 Análise Comparativa dos Estudos 54

4 Desenvolvimento de um agente virtual para geração de anotações em imagens de forma autônoma **57**
4.1 Visão Geral 57
4.2 Solução proposta 58
4.2.1 Implementação 58
4.3 Experimentos 63
4.3.1 Caso de uso 1: Banco de dados de exames de tórax 64
4.3.2 Caso de uso 2: banco de dados de exames de mamografia 64
4.4 Resultados 64
4.4.1 Caso de uso 1: Banco de dados de exames de tórax 65
4.4.2 Caso de uso 2: banco de dados de exames de mamografia 67
4.5 Conclusão 70

5 Evolução na geração autônoma de anotações a partir de uma metodologia de ensino para um agente RL com Active Learning **71**
5.1 Método de aconselhamento: TLM (Try a Little More) 72
5.1.1 Visão Geral 72
5.1.2 Implementação do TLM 73
5.2 Metodologia para ensino de um agente RL 74
5.2.1 Visão Geral 74

5.2.2 Implementação de Aprendizado Ativo 76
5.3 Experimentos 81
5.3.1 Conjuntos de dados 82
5.3.1.1 Caso de uso 1: Base de dados com exames de mamografia 82
5.3.1.2 Caso de uso 2: Base de dados com imagens de aeronaves 82
5.3.2 Protocolo de avaliação 83
5.3.3 Técnicas comparadas 84
5.4 Resultados 85
5.4.1 Caso de uso 1: Exames de mamografia 85
5.4.2 Caso de uso 2: Imagens de aeronaves 89
5.5 Conclusão 92

6 Conclusões gerais e Trabalhos futuros 94
6.1 Conclusões 94
6.2 Contribuições 94
6.3 Publicações 95
6.4 Limitações e Trabalhos Futuros 95

Referências bibliográficas 98

Lista de figuras

Figura 2.1 Visão geral dos elementos básicos de RL e suas interações. Um agente, representado pelo personagem PAC-MAN, aplica ações em um ambiente, representado pelo labirinto. A cada ação, uma recompensa é atribuída e o próximo estado é apresentado para o agente. 25
Figura 2.2 Diagrama de estado representando uma cadeia de Markov e sua transição. 26
Figura 2.3 Fluxograma do algoritmo Q-Learning. 30
Figura 2.4 No campo de IA, estão as abordagens de aprendizado de máquina, as quais podem ser trabalhadas com aprendizagem profunda. O DRL é a interseção entre aprendizado profundo e aprendizado por reforço. Imagem adaptada de [56]. 32
Figura 2.5 A imagem (a) apresenta uma visão geral de um modelo ML. Na imagem (b) é apresentado a visão geral de uma programação clássica. Fonte [12]. 32
Figura 2.6 Representação do modelo de neurônio artificial (perceptron). 34
Figura 2.7 Representação de uma rede neural artificial com suas respectivas camadas de entrada, oculta e saída. 35
Figura 2.8 Imagem de um cachorro e sua representação em uma matriz de pixels. 35
Figura 2.9 Processo de construção do filtro de recursos na etapa da convolução. 36
Figura 2.10 Ilustração do processo de Max Pooling. 36
Figura 2.11 A imagem (a) representa o fluxo do algoritmo Q-Learning. A imagem (b) representa o fluxo geral do algoritmo DQN. 37
Figura 2.12 Ilustração esquemática da rede neural convolucional. A entrada para a rede neural consiste em uma imagem qualquer de dimensão 84x84x4, a qual segue por três camadas convolucionais e duas camadas totalmente conectadas com uma única saída para cada ação. Fonte [52]. 38
Figura 2.13 Diagrama de interação entre uma rede de política Q e uma rede Alvo. 39
Figura 2.14 Ilustração de armazenamento do Experienciece replay através do algoritmo DQN. 40

Figura 2.15 Ilustração esquemática do processo de trabalho do algoritmo de DQN. A cada transição (tupla formada pelo estado atual, ação, recompensa e próximo estado) é armazenada em uma memória de experiências (experience replay) e amostras são selecionadas aleatoriamente para treinamento da Rede Q. Através da estratégia de seleção exploração (valor randômico) ou exploitation (estratégia gulosa), uma ação é selecionada para que o agente RL possa aplicá-la no ambiente. Imagem adaptada [56]. 40
Figura 2.16 Diagrama de loop de aprendizado ativo. Em cada iteração, a função de pontuação e a estratégia de amostragem na etapa de consulta decidem quais imagens devem ser enviadas para anotação e adicionadas ao conjunto de dados de treinamento para treinamento adicional. Imagem adaptada de [31]. 43
Figura 2.17 Diagrama de loop de aprendizado ativo baseado em Pool. Imagem adaptada de [89]. 44

Figura 4.1 Ilustração das ações que o agente de RL realiza nos Estados. 59
Figura 4.2 Imagem ilustrativa do agente RL criando uma anotação em forma de caixa delimitadora da papila em um exame de mamografia. 60
Figura 4.3 Arquitetura usada para o algoritmo DQN. A entrada é uma imagem com 256 x 256 pixels e processada por camadas convolucionais. A camada de saída prevê o valor para as nove ações possíveis a serem tomadas pelo agente. 62
Figura 4.4 A imagem ilustra uma caixa delimitadora de verdade (em verde) e uma caixa delimitadora gerada por um modelo (em vermelho). Fonte [68] 63
Figura 4.5 Resultado do treinamento do agente RL para detectar a estrutura da cardiomegalia. Cada experimento é representado por uma cor diferente (indicado na legenda da imagem). As curvas dos experimentos são suavizadas produzindo uma curva mais consistente na tendência das recompensas. O experimento denominado **exp4** que utiliza ajuda humana e o modelo pré-treinado, apresentou o melhor resultado com o maior acúmulo de recompensas positivas. 66
Figura 4.6 Imagem de cardiomegalia sendo detectada. Em azul a verdade absoluta e em vermelho a caixa delimitadora gerada pelo agente. A cruz na imagem é para indicar essa é a caixa final especificada pelo agente RL. 67
Figura 4.7 Resultado do treinamento do agente RL para detectar a estrutura da papila. Cada experimento é representado por uma cor diferente (como indicado na legenda da imagem). As curvas dos experimentos são suavizadas produzindo uma curva mais consistente na tendência das recompensas. O experimento denominado **exp4** que utiliza ajuda humana e o modelo pré-treinado, apresentou o melhor resultado, com o maior acúmulo de recompensas positivas. 69

Figura 4.8 Imagem do mamilo sendo detectado. Em azul a verdade absoluta e em vermelho a caixa delimitadora gerada pelo agente. 70

Figura 5.1 Diagrama da abordagem de aconselhamento com o método TLM. Quando o agente aprendiz está com dúvida, ele solicita ao professor humano um conselho sobre qual ação realizar em um determinado estado. O professor humano, através de uma interface, é capaz de inserir uma ação dentre as quais o agente RL consegue interpretar e assim aplicá-la no ambiente. Quando o aprendiz decidir terminar de explorar o ambiente, seja porque encontrou seu objetivo ou por não ter conseguido, o professor humano poderá visualizar a ação aplicada do agente RL no ambiente e assim decidir se de fato ele pode terminar ou se ele deve tentar um pouco mais. 74

Figura 5.2 Pipeline para treinamento de um agente RL através da metodologia proposta. O agente RL adquire um aprendizado progressivo, aprendendo a realizar anotações sobre as imagens mais fáceis e, ao longo do tempo, passando para as imagens mais difíceis. Os componentes em amarelo, representam nosso método de aconselhamento apresentado na Figura 5.1. Os componentes em azul representam o ciclo do AL apresentado na Figura 5.3. 77

Figura 5.3 O ciclo da abordagem de AL baseada em pools. 78

Figura 5.4 Ilustração de uma abordagem tradicional de redes neurais e a com DQN. 79

Figura 5.5 Arquitetura de uma DQN regular e outro com a adição de *heads*. Imagem adaptada [15] 79

Figura 5.6 Imagens de aviões da base de dados Pascal VOC. 83

Figura 5.7 Resultado dos modelos sendo aplicados em um conjunto de teste para criação de novas anotações na forma de caixa delimitadora. Embora nosso método tenha ficado um pouco abaixo do limite superior no quesito quantidade de imagens anotadas, entre os métodos de aconselhamento, nossa abordagem apresenta uma considerável vantagem. 86

Figura 5.8 Análise da frequência de conselhos emitidas pelo humano durante o treinamento de um agente RL. Uma barra de cores é apresentada ao lado direito da figura, indicando que cores fracas representam baixas interações humanas. Em contrapartida, cores fortes indicam maiores interações humanas. Como não existe uma única ação correta a ser aplicada pelo agente em seu treinamento, é preciso analisar o quadro geral de interações. Nossa abordagem apresentou uma coluna com cores mais claras (baixas interações) ao comparar com os outros métodos. 88

Figura 5.9 Resultado dos modelos sendo aplicados em um conjunto de teste para criação de novas anotações na forma de caixa delimitadora em imagens com aeronaves. Nosso método apresentou o melhor resultado em comparação com limite superior, baseline e os outros métodos de aconselhamento. Além de apresentar o melhor resultado na quantidade de gerar novas imagens, também apresenta um índice baixo de predições erradas (FPs e FNs). 90

Figura 5.10 Análise da frequência de conselhos emitidas pelo humano durante o treinamento de um agente RL. Uma barra de cores é apresentada ao lado direito da figura, indicando que cores fracas representam baixas interações humanas. Em contrapartida, cores fortes indicam maiores interações humanas. Como não existe uma única ação correta a ser aplicada pelo agente em seu treinamento, é preciso analisar o quadro geral de interações. Embora nosso método apresente uma coluna com cores mais fortes (média e alta interação), a participação humana é apenas para aconselhamento e não para gerar novos rótulos de imagens, portanto, não a torna mais cara e sim essencial para o ensino e aprendizado do agente RL. 91

Lista de tabelas

Tabela 3.1 Estudos com suas respectivas estratégias. 55

Tabela 4.1 Hiperparâmetros de aprendizagem 61
Tabela 4.2 Dados de treinamento de cardiomegalia 65
Tabela 4.3 Dados de teste de cardiomegalia 65
Tabela 4.4 Dados de treinamento do mamilo 68
Tabela 4.5 Dados de teste do mamilo 68

Tabela 5.1 Hiperparâmetros de aprendizagem 85
Tabela 5.2 Comparação dos resultados da detecção de papila sobre o conjunto de testes. 87
Tabela 5.3 Avaliação das novas anotações da papila em um modelo supervisionado. 87
Tabela 5.4 Comparação dos resultados na criação de caixas delimitadoras sobre imagens de aeronaves no conjunto de testes. 92
Tabela 5.5 Avaliação das novas anotações de aviões em um modelo supervisionado. 92

Confia no Senhor os teus projetos e eles serão estabelecidos.

Provérbios, *16:3.*

Lista de Abreviaturas

A2C – Advantage Actor Critic
AL – *Active Learning* (Aprendizado Ativo)
ANN – *Artificial Neural Network* (Rede Neural Artificial)
CNN – *Convolutional Neural Network* (Rede Neural Convolucional)
DL – *Deep Learning* (Aprendizado Profundo)
RL – *Reinforcement Learning* (Aprendizado por Reforço)
DRL – *Deep Reinforcement Learning* (Aprendizado por Reforço Profundo)
DQN – *Deep Q-Network*
DQL – *Deep Q-Learning*
FP – *False Positive* (Falso Positivo)
FN – *False Negative* (Falso Negativo)
FAA – *Feedback Arbitration Agent*
IA – *Inteligência Artificial*
IOU – *Intersection Over Union* (Interseção sobre União)
ML – *Machine Learning* (Aprendizado de Máquina)
MDP – *Markov Decision Process* (Processo de Decisão de Markov)
mAP – *mean Average Precision*
NAA – *Newtonian Action Advice*
SARSA – State-Action-Reward-State-Action (Estado-Ação-Recompensa-Estado-Ação)
TLM – *Try a Little More* (Tente um Pouco Mais)
TP – *True Positive* (Verdadeiro Positivo)
YOLO – *You Only Look Once*
γ – *Discount Factor* (Fator de Desconto)
α – *Learning rate* (Taxa de Aprendizagem)
ϵ – (Probabilidade de Realizar Exploração)

1
Introdução

Nesta seção é apresentada uma breve descrição sobre o contexto deste trabalho, as abordagens motivadoras, a definição do problema, seguido pela justificativa e hipótese a ser avaliada. Também são apresentados os objetivos e as contribuições, bem como a organização das demais seções.

1.1
Contexto

Um sistema de Inteligência Artificial (IA) é composto pelos seguintes elementos: *Sistema IA = Código + Dados*, onde código compreende-se como modelos/algoritmos. Para ser alcançada uma satisfatória solução em IA, deve-se ter um equilíbrio entre trabalhos centrados no modelo (*Model-Centric*) e centrados nos dados (*Data-Centric*) [9, 86].

Trabalhos centrados nos modelos procuram realizar ajustes no algoritmo e em sua arquitetura para torná-los mais eficientes. Adotam um conjunto de dados fixos e realizam-se os esforços sobre o modelo. Esses modelos geralmente dependem da disponibilidade, volume e qualidade dos dados anotados para treinamento [3]. Realizar os esforços nos dados implica em melhorar as informações para que os modelos consigam aumentar seu desempenho. Sendo assim, adota-se um modelo que será fixo e os esforços passam a realizados sobre os dados. Esse foco nos dados de IA é um novo campo de pesquisa que vem sendo explorado por pesquisadores, cujo objetivo é construir conjuntos de dados de alta qualidade para aprendizado de máquina [73].

Muitos trabalhos na literatura ainda estão voltados para criação de métodos e aperfeiçoamento dos modelos, inclusive trabalhos voltados para computação visual [107]. Porém, é importante fornecer dados limpos e devidamente anotados para treinar modelos de aprendizado supervisionado [108]. Portanto, há um real problema dos dados serem ruidosos, possuindo anotações escassas e inconsistentes.

Essas limitações implicam diretamente na utilização de técnicas de IA, principalmente aquelas baseadas no aprendizado supervisionado, as quais requerem uma grande quantidade de dados anotados para o treinamento de um modelo. Para a área da saúde, por exemplo, o uso dessas técnicas tem

contribuído para o processamento e análise de imagens médicas [46] [20]; no entanto, a ausência de dados anotados é uma limitação para a implementação dessas soluções.

Os dados anotados são necessários para permitir que a rede aprenda a relação entre uma entrada e saída desejada durante o treinamento de um modelo de aprendizado de máquina. Com dados limpos e anotações suficientes, a precisão de um modelo frequentemente corresponde ou excede o seu propósito [21]. No entanto, obter novas anotações é uma tarefa cara e demorada. Esse processo de anotações é geralmente executado manualmente por especialistas humanos [3].

1.2 Motivação

As técnicas de apoio a decisões baseadas em IA podem ser utilizadas para facilitar a realização de diagnósticos eficientes. Nesse sentido, o campo de visão computacional tem se beneficiado consideravelmente pelos desenvolvimentos através do aprendizado de máquina, como os algoritmos de Aprendizado por Reforço (*Reinforcement Learning - RL*) [101], cuja solução para um determinado problema é encontrada através de um agente virtual que explora as interações no ambiente, e por técnicas chamadas Redes Neurais Convolucionais (*Convolutional Neural Networks-CNN*).

Mnih et al. [51] propuseram uma combinação desses algoritmos. Combinaram RL com CNN para propor uma nova abordagem denominada *Deep Reinforcement Learning (DRL)*. Nos últimos anos, os modelos DRL alcançaram avanços que ultrapassam o desempenho humano em jogos como Atari [52], e também demonstraram avanços promissores em permitir que robôs físicos aprendam habilidades complexas no mundo real [32], na implantação de carros autônomos [33] e na detecção de partes anatômicas [25].

No entanto, essas técnicas dependem de uma grande quantidade e qualidade dos dados para treinamento de um modelo. Para aplicações reais, muitas bases de dados não apresentam um conjunto de dados anotados que possam ser treinados por um modelo de IA, sendo necessário antes gerar novas anotações ou eliminar ruídos existentes para construir uma nova base de dados com qualidade e anotações consistentes. Porém, a construção e correção de um novo conjunto de dados têm custos elevados, além de exigir bastante esforço humano.

Nesse sentido, a principal motivação deste trabalho é apresentar uma abordagem centrada nos dados, focada em criar anotações de maneira autônoma e consistente a partir de um conjunto de dados com anotações escassas.

Desse modo, contribui-se para redução do custo e esforço na obtenção de novos dados.

1.3 Definição do problema

Na área de visão computacional, para o sucesso dos algoritmos de aprendizagem profunda, é indispensável anotações apropriadas para treinamento de um modelo [107]. Para certas tarefas em visão computacional na área médica, como classificação de patologias e detecção de estruturas anatômicas, as anotações das imagens são fundamentais. Porém, a criação de um conjunto de dados com qualidade, através de anotações manuais, consome muito tempo e é uma operação cara. Além disso, algumas anotações podem apresentar inconsistências causadas pelas diferentes interpretações dos anotadores, implicando diretamente no desempenho do modelo.

Portanto, a questão que este trabalho está concentrado em responder é a seguinte:

Como desenvolver uma abordagem que contribua na criação de novas anotações autônomas e redução dos esforços humanos em gerá-las para o treinamento de modelos supervisionados em detecção de objetos?

1.4 Justificativa

O sucesso dos métodos de aprendizagem de máquina supervisionados depende da viabilidade da qualidade e da quantidade de imagens com dados anotados para a realização do treinamento [118]. A falta de dados anotados é um real problema, sendo o custo para realizar novas anotações muito alto. Investir em abordagens que visam propor soluções que geram novas anotações com baixo custo e esforço mínimo é um caminho importante e está sendo explorada por pesquisadores [54, 63, 81, 107, 108].

Para reduzir os esforços nas anotações, os pesquisadores exploraram abordagens com anotações de dados de forma eficiente com custos menores [116]. Um exemplo dessa abordagem são os algoritmos de Aprendizado Ativo (*Active Learning - AL*). Esses algoritmos visam reduzir o custo de anotação, selecionando apenas as imagens mais informativas a serem anotadas pelo ser humano, para melhorar a precisão de um modelo [10].

Entretanto, AL geralmente trabalha com métodos de seleção heurística, limitando a eficiência do algoritmo [17, 80, 125]. Recentemente, modelos de CNN são estudados para melhorar a eficiência desse algoritmo [70, 88, 112],

bem como os algoritmos de aprendizado por reforço para automatizar o processo de seleção dos dados não anotados [49, 98].

1.5 Hipótese

A utilização de um algoritmo de aprendizado por reforço com redes neurais (DRL), através da inclusão humana no loop de treinamento do algoritmo para ensiná-lo e avaliá-lo, contribui na geração de anotações na forma de caixas delimitadoras de modo autônoma, mesmo em posse de anotações escassas.

1.6 Objetivos

Considerando o contexto atual, este trabalho tem como objetivo empregar uma abordagem centrada nos dados, de modo que a preparação dos dados de imagens, na etapa de geração de caixas delimitadores (*bounding box*), sejam feitas automaticamente, mesmo em posse de poucos dados anotados. Para atingir esse objetivo, será adotado:

- Abordagem DRL na criação de um agente virtual para realizar anotações autônomas;
- Inserção humana no processo de treinamento para apoiar o processo de aprendizado de um agente na realização da tarefa corretamente, mesmo com um conjunto de dados escasso.
- Utilização do AL e integração de métodos para mensurar a incerteza para guiar o processo de aprendizado do agente virtual.

1.7 Contribuições

A partir desse cenário, será proposta uma abordagem centrada nos dados, com foco na geração de caixas delimitadoras para treinamento de modelos de aprendizado de máquina. Este também contribuirá na criação de novas anotações automaticamente, reduzindo tempo e esforços humanos.

Diversos trabalhos na literatura estão voltados para abordagens centradas no modelo, seja na criação ou aperfeiçoamento de algoritmos e modelos. Entretanto, existe muito trabalho na parte de preparação de dados e que não são explorados com frequência. Com o intuito de apresentar uma abordagem para redução do tempo, esforços e custos na preparação dos dados, na etapa de geração das anotações, esse estudo visa contribuir para:

1. **Apresentar um algoritmo de DRL que integra o auxílio humano no loop de treinamento de um agente RL para criação de anotações automaticamente.** Na literatura alguns trabalhos adotam o humano no processo de aprendizagem do agente RL, como mostra a pesquisa feita por *Naja e Chetouani* [58]. No entanto, esses trabalhos não investigam a capacidade de um agente em criar novas anotações a partir de uma quantidade escassa de dados. A contribuição que será apresentada nesse trabalho cria uma abordagem capaz de gerar novas anotações automaticamente a partir da inserção do humano no processo de treinamento de um agente RL, proporcionando anotações do tipo caixa delimitadora.

2. **Criar uma metodologia de treinamento para um agente RL.** Embora nos últimos anos a utilização de RL esteja sendo adotada em diversas áreas, em aplicações reais onde há limitações de dados para treinamento do agente, é uma área em desenvolvimento com oportunidades de pesquisa para serem exploradas. Nesse aspecto, contribuímos com uma metodologia de ensino a um agente RL, o qual terá a sua evolução de aprendizado semelhante a um "aluno", gradualmente, sendo o humano o seu "professor". Nessa metodologia, empregamos a utilização da técnica de AL para selecionar os dados que impactarão positivamente no aprendizado do "aluno" virtual.

1.8 Organização

Este trabalho está organizado da seguinte forma. O capítulo 2 apresenta os fundamentos teóricos para entendimento dessa tese. O capítulo 3 apresenta a situação atual da literatura sobre métodos para construção de novas anotações e também são apresentados métodos para inserção do humano no loop de treinamento em algoritmos de RL. No capítulo 4, os avanços dos estudos são apresentados, com a inclusão do método de DRL, a interação humana e a criação de um novo método de aconselhamento, bem como os resultados obtidos na geração de anotações. O capítulo 5 apresenta a evolução do método para geração autônoma de anotações, com a criação de uma metodologia de ensino para treinamento de agentes virtuais e os resultados finais. Finalmente, o capítulo 6 apresenta as conclusões e trabalhos futuros.

2
Fundamentação Teórica

Este capítulo apresenta uma visão geral dividida em tópicos, que abordam conceitos fundamentais para entendimento do restante desta tese. Inicialmente, será explicado sobre os conceitos fundamentais da abordagem de aprendizado por reforço; em seguida, será apresentada conceitualmente a definição e algoritmos de Aprendizado por Reforço Profundo. Depois disso, serão apresentados os estudos a partir de dados parcialmente anotados. Por fim, será apresentada a abordagem de aprendizagem ativa para geração de um conjunto de dados.

2.1
Aprendizado por Reforço

Aprendizado por Reforço é uma das áreas que compõem o campo de aprendizado de máquina. Em RL, o aprendizado é realizado por interações em um ambiente. O aprendiz não sabe qual o melhor caminho a ser seguido para atingir o seu objetivo. Portanto, vai explorando através de tentativa e erro as melhores ações a serem realizadas em um determinado ambiente. Para estimular o processo do aprendizado, recompensas são emitidas para o agente virtual.

Considere um professor ensinando um aluno uma determinada tarefa. Este não deve mostrar a resposta para o seu aluno, ao invés disso, deve estimulá-lo a raciocinar para encontrar uma solução autossuficientemente. A cada vez que o aluno demonstrar um avanço em seu conhecimento, o professor atribui um ponto na nota do aluno, caso contrário, não o receberá. Desse modo, o aluno consegue perceber quais ações o levaram a receber os pontos e, assim, seguirá por essa linha de raciocínio de modo a obter o máximo de pontos possíveis.

Da mesma forma, em um ambiente de RL, não é ensinado ao agente como realizar, em vez disso, recompensas são atribuídas a ele a cada ação que ele fizer. Essas recompensas podem ser positivas ou negativas, dependendo da ação realizada pelo agente. A cada ação que o aproxima do objetivo final, recompensas positivas são dadas ao agente, caso contrário, recompensas negativas. Outros subelementos compõem um algoritmo em RL, tais como:

agente, ambiente, política, função de recompensa e função de valor [77, 101]. A próxima seção apresenta algumas definições que contribuem para uma maior compreensão.

Um agente é o elemento responsável por observar um ambiente e explorá-lo através de ações. A partir dessas interações, recompensas são atribuídas ao agente, de modo a estimulá-lo ao sucesso de uma determinada tarefa [38].

Para uma maior compreensão, a ilustração em um cenário de jogos é um bom auxiliador. Pegaremos o cenário de um jogo como, por exemplo, Pac-Man, um dos clássicos jogos digitais, criado por Toru Iwatani, licenciado pela *MIDWAY* e desenvolvido pela *NAMCO* [16]. Nesse jogo, o personagem principal, que é o Pac-Man, precisa comer todas as pastilhas espalhadas pelo cenário que é um labirinto, sem ser alcançado pelos inimigos, os fantasmas.

Trazendo no contexto de RL, o Pac-Man seria o nosso agente, onde através dele, as interações no cenário são realizadas com a finalidade de ser alcançado o objetivo final no cenário (ambiente).

Um ambiente é o espaço onde um agente, através de suas ações, explora (através de tentativa e erro) tudo quanto for observável pelo sistema RL [30].

Seguindo o exemplo do jogo Pac-Man, o ambiente é o próprio cenário do jogo, que nesse caso é o labirinto por onde o personagem "agente" explora suas possibilidades de ações, visando amontoar para si, recompensas positivas.

Dependendo da aplicação, um ambiente distingue-se do outro, com observações e possibilidades de interações diferentes. Existem quatro categorias de ambientes mais comuns: os **determinísticos, estocásticos, discretos e contínuos** .

Os ambientes são determinísticos, quando existe previsibilidade em qual ação a ser tomada. Ou seja, não há incerteza no ambiente. Um jogo de xadrez, por exemplo, o qual é formado por um tabuleiro contendo 64 casas e 16 peças. Os movimentos das peças podem ser previsíveis. Os ambientes são estocásticos, quando existe imprevisibilidade nas ações. Os carros autônomos, por exemplo, possuem variações em suas ações. Quando um ambiente possui um número finito de estados, a qual um agente pode se movimentar através de ações, esse ambiente é considerado Discreto. Caso contrário, é considerado um ambiente continuo.

Uma política é uma função que pode ser representada por $\pi(s) : S- > A$. É o elemento central da técnica de RL, pois ela define o comportamento do agente RL, orientando qual ação deverá ser tomada pelo agente em um determinado estado [100]. O objetivo final é modelar uma política que permita ao agente tomar ações corretas em cada estado. Portanto, uma política ótima é definida como a política que maximiza a recompensa ao longo da movimentação

pelo ambiente [5] e é esta que, ao implementar um algoritmo de RL, deseja-se alcançar.

Conforme apresentado anteriormente, um agente, através de suas ações, interage com o ambiente, movimentando de um estado para outro, por meio de sua política. Para cada ação realizada pelo agente, uma recompensa é emitida, a qual por muitas vezes é na forma de numeral. Por exemplo, ações que o aproximam do objetivo, +1 (recompensa positiva) é atribuído, caso contrário, -1 (recompensa negativa) é emitida para o agente. A Figura 2.1 apresenta uma visão geral dos elementos básicos que compõem a abordagem RL.

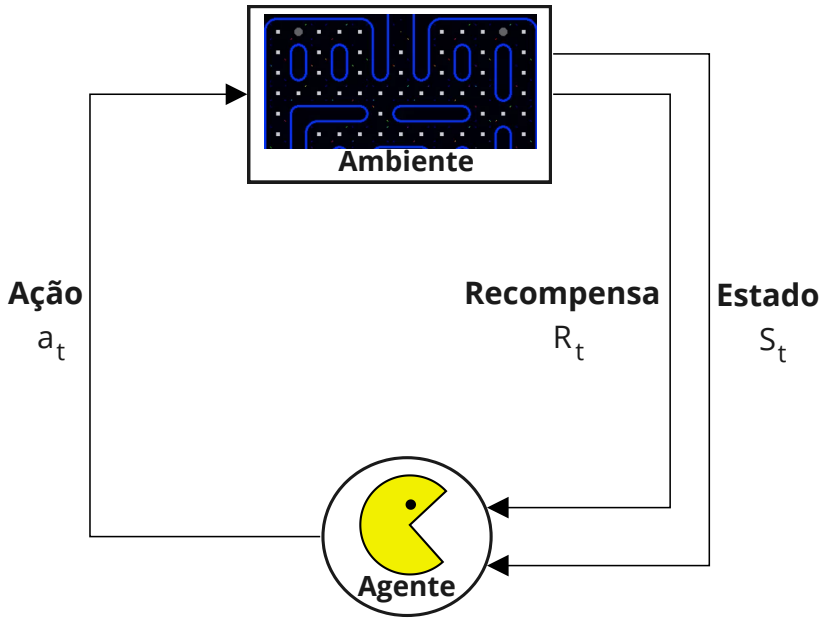

Figura 2.1: Visão geral dos elementos básicos de RL e suas interações. Um agente, representado pelo personagem PAC-MAN, aplica ações em um ambiente, representado pelo labirinto. A cada ação, uma recompensa é atribuída e o próximo estado é apresentado para o agente.

Em RL, é preciso especificar quantas vezes o agente poderá interagir com o ambiente a partir do estado inicial até o estado final. Essa quantidade de interações realizadas pelo agente no ambiente são chamadas **episódios**. A cada episódio, o agente possuirá uma determinada quantidade de passos para interagir com o ambiente ou alcançar um estado especial, chamado estado terminal, o que sinalizará o fim de um episódio, ocasionando assim na reinicialização para o estado inicial. As recompensas podem ser modeladas para serem emitidas ao fim de cada passo do agente.

Uma função valor (*value function*) denota quão bom é para um agente estar em um estado particular. Ela informa ao agente o quanto de recompensa

receberá ao tomar uma determinada ação em um específico estado. É dependente da política e é geralmente denotado por $v(s)$. O valor de um estado é a quantidade total de recompensa que um agente pode esperar acumular no futuro, a partir desse estado. Enquanto as recompensas determinam a conveniência imediata, os valores indicam a conveniência a longo prazo dos estados após considerar os que provavelmente se seguirão e as recompensas disponíveis neles.

2.1.1 Processo de decisão de Markov

Um passo para entender como funciona um processo de decisão de *Markov*, é começar pela compreensão de como funciona a cadeia de *Markov*.

Regida sobre a propriedade Markoviana, a qual afirma que o futuro depende apenas do presente e não do passado [27], uma cadeia de Markov é um modelo probabilístico cuja dependência é exclusivamente do estado atual para saber o estado futuro. Ou seja, não é preciso saber o passado para avançar ao caminho do futuro.

A movimentação de um estado (presente) para outro (futuro) é chamada *transição* e sua probabilidade é definida pela *probabilidade de transição*. Adotemos como exemplo, um aluno em uma determinada série. Podemos realizar uma predição a partir de suas notas de provas e assim prever o próximo estado que esse aluno terá, ou seja, aprovado ou reprovado. Essa predição é realizada a partir da série atual do aluno e não das anteriores. A Figura 2.2 mostra uma representação do exemplo através de um diagrama de estados.

Figura 2.2: Diagrama de estado representando uma cadeia de Markov e sua transição.

O processo de decisão de Markov (MDP), é uma extensão da cadeia de Markov, sendo uma estrutura matemática para modelar situações de tomada de decisão onde as transições entre estados são probabilísticas [71]. Um MDP é definido em uma tupla $(S, A, P^a_{ss'}, R, \gamma)$, sendo:

- S: representa o conjunto de estados onde o agente pode se deslocar.
- A: representa um conjunto de ações que podem ser realizadas pelo agente para se deslocar de um estado para o outro.
- $P^a_{ss'}$: representa a probabilidade de transição de o agente passar de um estado $s \epsilon S$ para outro $s' \epsilon S$ por realização de alguma ação $a \epsilon A$.
- R: representa uma função de recompensa que emitirá um custo a cada ação realizada no estado pelo agente.
- γ: um número entre $[0, 1]$ que controla a importância de recompensas imediatas ou futuras.

Um MDP, se enquadra muito bem em sistemas de RL, por isso que a maioria dos algoritmos adotam esse processo como base em sua estrutura. Um processo com MDP possui um estado inicial s_a, onde um agente RL toma suas primeiras observações do ambiente. A cada passo t, o agente escolhe uma ação a_t segundo a probabilidade de transição $P^a_{ss'}$ e assim move para o próximo estado $s^{'}$ e recebe uma recompensa por isso. Isso se repete até o estado final ser alcançado. Em RL, o principal objetivo é encontrar uma maneira para escolher uma $a \epsilon A$ que visa maximizar a recompensa, como mostra a Equação 2-1 [62].

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{2-1}$$

Nesse ponto, o fator desconto (*discount factor* - γ) tem um papel importante, pois através dele que será decidido o quanto de importância será dado para as recompensas imediatas ou futuras. Quanto γ mais próximo a 0, significa que recompensas imediatas são as mais importantes. Quanto γ mais próximo a 1, significa que recompensas futuras são as mais importantes [67]. Deve ser encontrado um valor para esse hiperparâmetro que evite o imediatismo, ou seja, o agente deve escolher as ações que o levem à melhor solução global possível, não apenas à melhor solução imediata.

Com as propriedades de MDP definidas, é possível expandir o entendimento sobre alguns elementos básicos que compõem algoritmos de RL. Como visto anteriormente, uma função valor $(V(s))$, especifica para um agente, o quanto é bom para ele estar em um estado particular seguindo a política (π). Uma função valor sempre depende de uma política. Portanto, uma função valor é definida pela seguinte Equação 2-2:

$$V^{\pi}(s) = E_{\pi}[\sum_{k=0}^{\infty} \gamma^{k} r_{t+k+1} | s_t = s] \tag{2-2}$$

Assim como existe uma função para verificar a importância de um estado para o agente, há também uma função que permite calcular o quanto é bom para o agente realizar uma particular ação em um estado, seguindo uma política π. Chamada função de valor de ação de estado *State-action value function* ou simplesmente *Q-function*, é definida pela seguinte Equação 2-3

$$Q^{\pi}(s,a) = E_{\pi}[\sum_{k=0}^{\infty} \gamma^{k} r_{t+k+1} | s_t = s, a_t = a] \tag{2-3}$$

Também podemos definir uma **política ótima**, a qual resulta em uma função de valor ótimo, bem como uma **função Q ótima** como mostra as Equações 2-4 e 2-5

$$V^{*}(s) = max_{\pi} V^{\pi}(s) \tag{2-4}$$

$$V^{*}(s) = max_{a} Q^{*}(s,a) \tag{2-5}$$

Em um sistema de RL, deseja-se que o agente aprenda uma política ótima, que o permita executar as melhores ações possíveis em um ambiente, acumulando o máximo de recompensas positivas. Para tornar essas equações viáveis, se utiliza a equação de *Bellman* [4], que indica recursivamente a relação entre o estado atual e o próximo estado. Portanto, a equação de *Bellman* para a *Q-function* e para função valor, pode ser representada pela Equação 2-6 e 2-7.

$$V^{\pi}(s) = \sum \pi(s,a) \sum P^{a}_{ss'} [R(s,a) + \gamma V^{\pi}(s')] \tag{2-6}$$

$$Q^{\pi}(s,a) = \sum P^{a}_{ss'} [R(s,a) + \gamma \sum Q^{\pi}(s',a')] \tag{2-7}$$

2.1.2 Algoritmos de aprendizagem

O grande foco na técnica de RL é criar um agente que consiga encontrar soluções para um problema através da exploração do ambiente. Portanto, moldar um algoritmo para ensinar um agente é desafiador. Quando um problema apresenta muito bem uma quantidade de estados e a quantidade de ações que esse agente poderá realizar, é possível armazenar essas informações, bem como as recompensas para as ações realizadas no ambiente em uma tabela e assim, a cada passo do treinamento, essa tabela é preenchida e consultada para avanço do aprendizado do agente. Para esses casos, existem dois algoritmos clássicos em RL, *Q-Learning* [114] e *SARSA (State-Action-Reward-State-Action)* [47]

O algoritmo *Q-Learning*, é o algoritmo mais popular de aprendizado por reforço. Criado por *Watkins e Dayan* [114], este algoritmo se aplica em muitos problemas. Ele se concentra no par de estado e ação para realizar seu aprendizado. Uma tabela Q (*Q-table*) é preenchida iterativamente para cada par de ação e estado em cada interação com o ambiente. Assim, essa tabela acaba se tornando o "cérebro" desse agente, pois as suas próximas decisões são aplicadas a partir dos valores armazenados nessa tabela. A atualização dessa tabela é realizada através da Equação (2-8).

$$Q(s,a) = Q(s,a) + \alpha[r + \gamma.maxQ(s',a') - Q(s,a)], \tag{2-8}$$

onde:

- s (estado): é o estado atual.
- a (ação): é uma ação realizada no atual estado.
- s'(próximo estado): a cada ação realizada, um novo estado é selecionado.
- r (recompensa): para o par (s, a), uma recompensa é emitida ao agente pelas suas movimentações em um determinado estado.
- a' (próxima ação): para um novo estado selecionado, uma nova ação também será tomada.
- γ (fator de desconto): valor predefinido que mostra a importância das recompensas presentes ou futuras.
- α (taxa de aprendizagem): implica na importância dada a novos eventos durante a fase de aprendizagem.

Existem duas maneiras com as quais um agente RL pode interagir com o ambiente, através de *Exploit* e *Explore*. Quando o agente realiza uma ação a partir de uma escolha aleatória, esse comportamento é denominado *Explore*. A ação aleatória é importante para permitir que novos estados possam ser explorados. A outra maneira de interação é a partir do conhecimento prévio obtido para tomar uma decisão. Uma ação é escolhida selecionando na tabela Q a que tenha o valor máximo para aquele estado, esse comportamento é denominado *Exploit*. O modo de interação com o ambiente escolhido pelo agente, é controlada a partir da taxa de exploração ϵ. Esta taxa é atualizada ao longo dos episódios, sendo reduzida linearmente. Com intervalos entre [0,1], $\epsilon > 0$ equivale ao comportamento *explore*, caso contrário, *exploit*.

Um algoritmo que representa o funcionamento do *Q-Learning* é apresentado pelo fluxograma ilustrado na Figura 2.3:

De acordo c om *Sutton e Barto* [100], a prova de convergência do Q-learning para uma política ótima, se dá se as seguintes restrições forem satisfeitas:

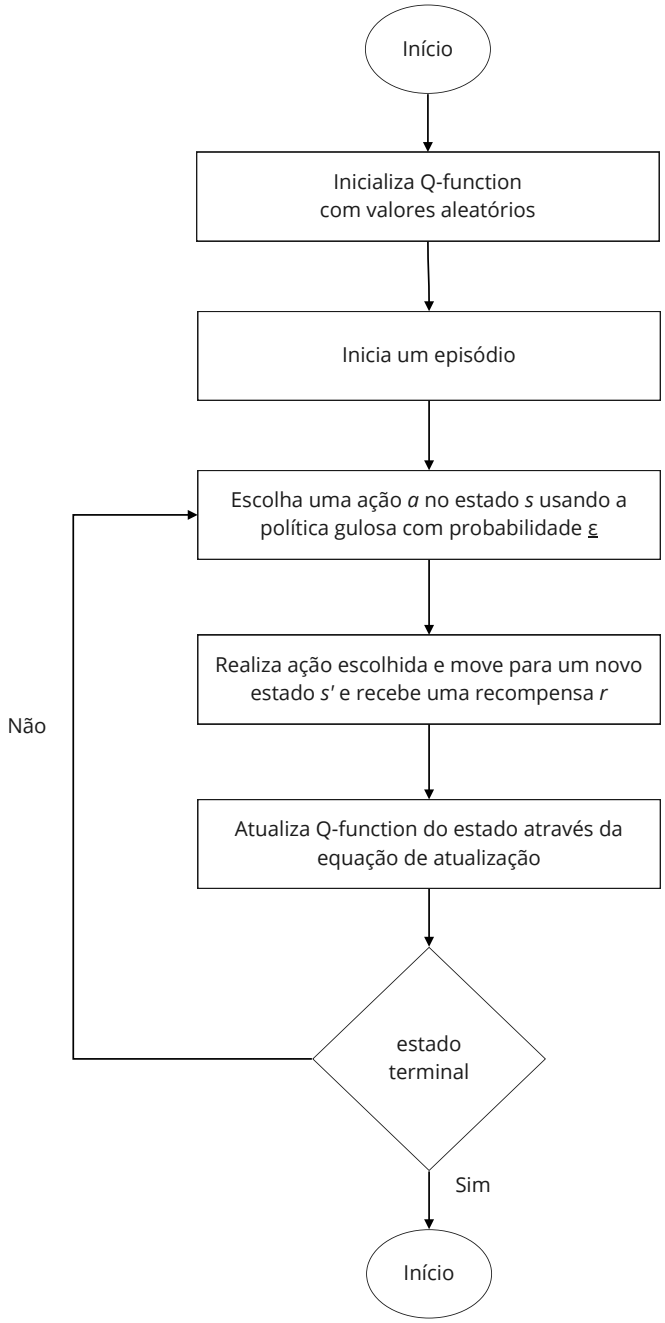

Figura 2.3: Fluxograma do algoritmo Q-Learning.

1. A propriedade de Markov seja satisfeita;
2. Todos os pares estado-ação são visitados por um infinito número de vezes;
3. A taxa de exploração (ϵ) seja reduzida a zero;
4. A taxa de aprendizagem (α) seja reduzida a zero;

Denominado *SARSA* é outro algoritmo clássico de RL, muito semelhante ao algoritmo *Q-Learning*. *SARSA*, é um algoritmo *on-policy*, ou seja, a atualização entre um par de estados e ação dependerá da política. A implementação é muito parecida com o algoritmo *Q-Learning*, a única mudança a ser realizada, é na equação de atualização dos *Q-function*, dada pela Equação 2-9:

$$Q(s,a) = Q(s,a) + \alpha(r + \gamma Q(s',a') - Q(s,a)), \tag{2-9}$$

O algoritmo que representa o funcionamento do SARSA é semelhante ao apresentado no Q-Learning, ilustrado pelo fluxograma na Figura 2.3. A parte que difere é na seleção do par de estados e ações, onde um é feito pela estratégia gulosa e outro pela próxima política.

2.2 Aprendizado por Reforço Profundo

Aprendizado por Reforço Profundo é uma abordagem de aprendizado de máquina voltada para a criação de programas de computador que podem resolver problemas que exigem inteligência [56]. DRL é uma ampliação da abordagem clássica de RL, onde agora o método utiliza aprendizado profundo para aproximar alguns componentes de aprendizado de um agente RL, como *Q-function*. A Figura 2.4 mostra a relação entre as áreas de aprendizagem com a DRL.

Ao contrário de uma programação clássica, onde cada regra é mapeada manualmente e árvores de decisões são construídas, um sistema de ML é treinado com muitos dados de entrada como exemplos, visando encontrar estruturas estatísticas nesses exemplos que permitirão ao sistema criar padrões sobre os dados e assim automatizar a tarefa, como ilustra a Figura 2.5.

Em um sistema de ML o aprendizado pode ser supervisionado, semi-supervisionado ou não-supervisionado. De modo geral, todo o aprendizado é construído a partir de cinco etapas.

1. A primeira é a *coleta dos dados*, trata-se da obtenção dos dados de interesse, para permitir que o modelo consiga aprender sobre eles.

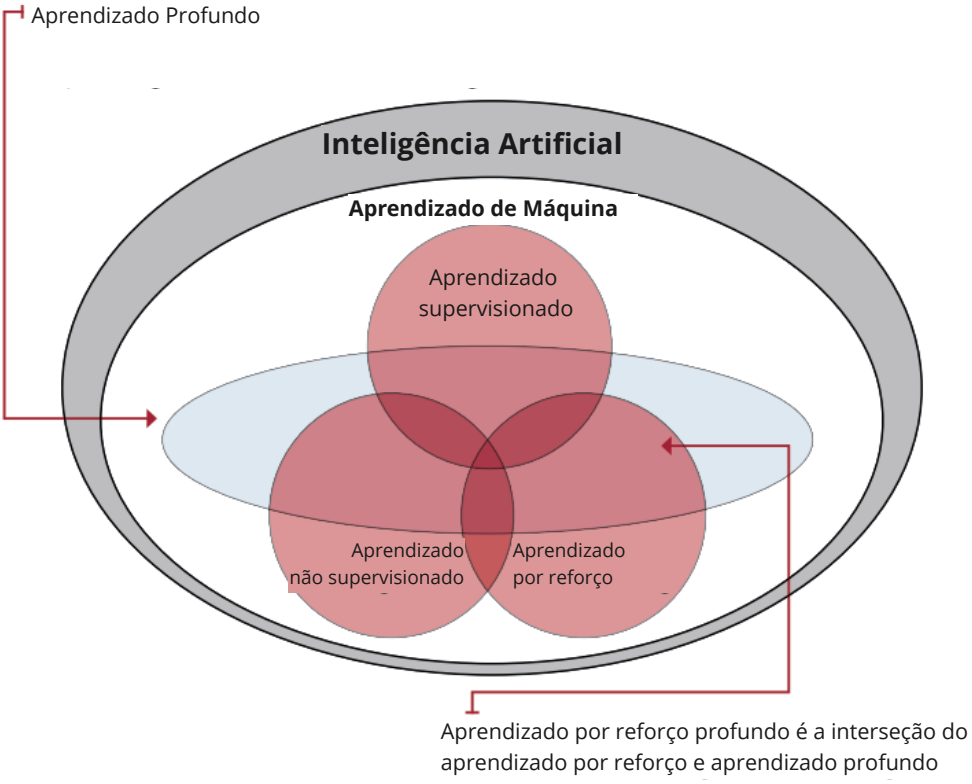

Figura 2.4: No campo de IA, estão as abordagens de aprendizado de máquina, as quais podem ser trabalhadas com aprendizagem profunda. O DRL é a interseção entre aprendizado profundo e aprendizado por reforço. Imagem adaptada de [56].

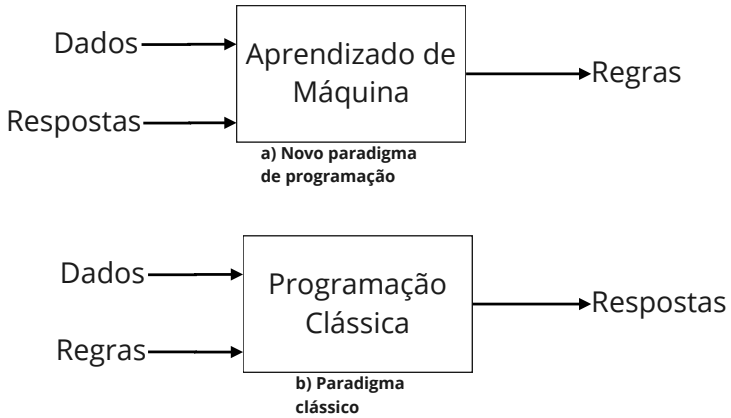

Figura 2.5: A imagem (a) apresenta uma visão geral de um modelo ML. Na imagem (b) é apresentado a visão geral de uma programação clássica. Fonte [12].

2. A segunda etapa é a *preparação dos dados coletados*, nessa etapa, visa analisar as informações coletadas, aferindo a qualidade dos dados. Os dados são divididos em duas amostras, uma para treinamento, onde os dados de entrada são analisados pelo algoritmo durante a etapa de treinamento. Outra parte é o teste, onde os dados são utilizados para avaliar o desempenho do modelo treinado. O algoritmo aplicará as regras aprendidas na fase de treinamento para realizar as predições esperadas. Esta é uma etapa crucial cujo impacto está relacionado diretamente com o resultado do aprendizado.

3. A terceira etapa é a *escolha e treinamento do modelo*. Ao longo dos anos, grandes esforços foram realizados para construir modelos de ML. Atualmente, diversos modelos estão disponíveis. A escolha de um modelo se dá pelo objetivo proposto a ser alcançado. Com a escolha do modelo certo, os dados de treinamento são passados em busca da aprendizagem com os padrões e evolução a cada treinamento.

4. Na quarta etapa é realizado o *monitoramento e aprimoramento dos parâmetros*. O conjunto de teste é utilizado para analisar o desempenho do modelo, verificando se a máquina conseguiu aprender corretamente. Esse é um ponto onde é possível monitorar as possíveis falhas e ajustar o modelo para conseguir melhorar a sua qualidade e eficiência em seu aprendizado.

5. Por fim, na quinta e última etapa, a fase de predição, onde o modelo após ser treinado, é utilizado efetivamente para solucionar os problemas pelos quais foi treinado para fazer.

Um dos sucessos do aprendizado de máquina, está ligado ao modelo de aprendizado com Redes Neurais Artificiais (ANN - *Artificial Neural Network*). ANN é um modelo computacional que apresenta características de funcionamento semelhantes às encontradas em redes neurais biológicas [22, 92]. Assim como o cérebro humano é composto por neurônios que recebem informações do ambiente externo e dividem entre os demais neurônios, uma rede neural também é composta por neurônios (*perceptron*) [83], onde processa uma informação de entrada e gera uma saída. A Figura 2.6 apresenta uma representação de um modelo simples de um neurônio artificial.

x_1, x_2 e x_3 representam um dado de entrada na rede. Cada entrada é multiplicada por pesos w_1, w_2 e w_3, para que cada entrada tenha sua importância ao influenciar no cálculo de saída da rede. Após multiplicar as entradas pelos pesos, somamos e adicionamos um valor chamado bias, sendo

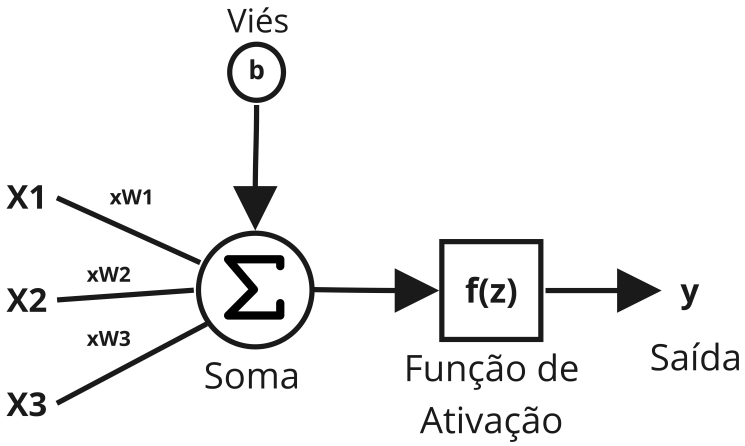

Figura 2.6: Representação do modelo de neurônio artificial (perceptron).

um elemento que serve para aumentar o grau de liberdade dos ajustes dos pesos e evitando valores nulos na saída da rede. Portanto, $z = (x_1.w_1 + x_2.w_2 + x_3.w_3) + b$ é dado pela Equação 2-10.

$$z = \sum(x * w) + b \tag{2-10}$$

A saída de cada neurônio artificial antes passa pela função de ativação $f(z)$ [121], a qual é utilizada para introduzir a não linearidade à rede. Existem diferentes funções de ativação, como *Sigmoide, tangente hiperbólica, ReLu, Softmax, etc.*

Uma ANN é constituída por camadas, onde cada uma possui uma coleção de neurônios artificiais [83]. Os neurônios de uma camada não interagem entre si, mas entre os neurônios das demais camadas. Essas comunicações são divididas entre camadas de entrada, onde através dessa a rede é alimentada com informações; nas camadas ocultas, são realizadas as identificações de padrões de uma entrada e a camada de saída recebe o resultado processado pela camada oculta e emite a predição realizada pela rede. A Figura 2.7, mostra uma representação de uma rede perceptron.

Outro modelo de aprendizado semelhante a uma ANN, são as Redes Neurais Convolucionais (CNN - *Convolutional Neural Network*). CNNs possuem características próprias como as operações matemáticas de convoluções, que a tornam uma ferramenta muito apropriada na área de computação visual, visto que possui uma arquitetura projetada para processamento de dados organizados em topologia de grade [26].

Em uma CNN, existem três principais camadas: convolucional, pooling e completamente conectada.

Na camada convolucional é realizada a operação chamada convolução (Equação 2-11), onde são extraídas importantes características da imagem,

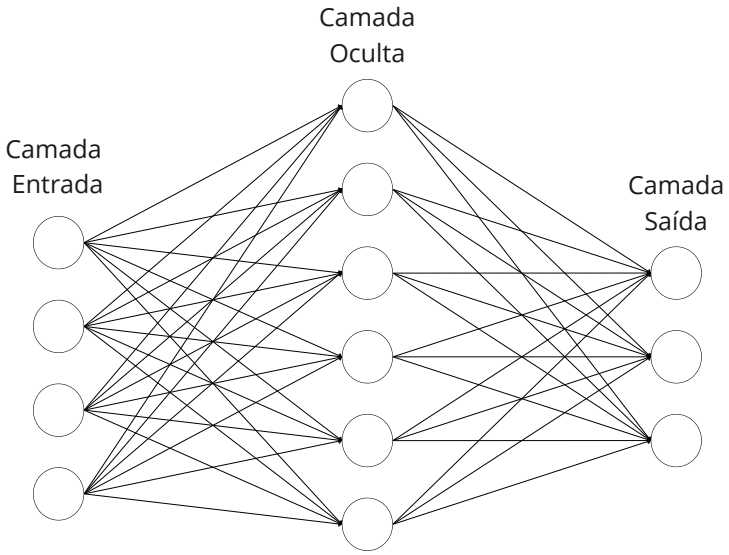

Figura 2.7: Representação de uma rede neural artificial com suas respectivas camadas de entrada, oculta e saída.

possibilitando maior entendimento sobre ela.

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n) \tag{2-11}$$

Onde S é o mapa de características e K é o filtro também bidimensional. Conforme a matriz do filtro K se desloca sobre a imagem I, a soma ponderada para cada posição i, j é calculada para a saída $S[i,j]$.

Para ilustrar o processo da camada de convolução, suponhamos uma imagem como entrada da rede. O primeiro passo é obter a matriz de valor em pixels dessa imagem, como mostra a Figura 2.8.

Figura 2.8: Imagem de um cachorro e sua representação em uma matriz de pixels.

Para essa matriz de pixels da imagem, outra matriz é considerada, o filtro. Essa matriz de filtro vai se deslizar sobre a matriz de pixels da imagem e realizará a multiplicação por elementos e resultará em uma nova matriz

chamada mapa de características ou mapa de ativação (*activation map*). Essa operação é realizada pela Equação 2-11 e ilustrada pela Figura 2.9.

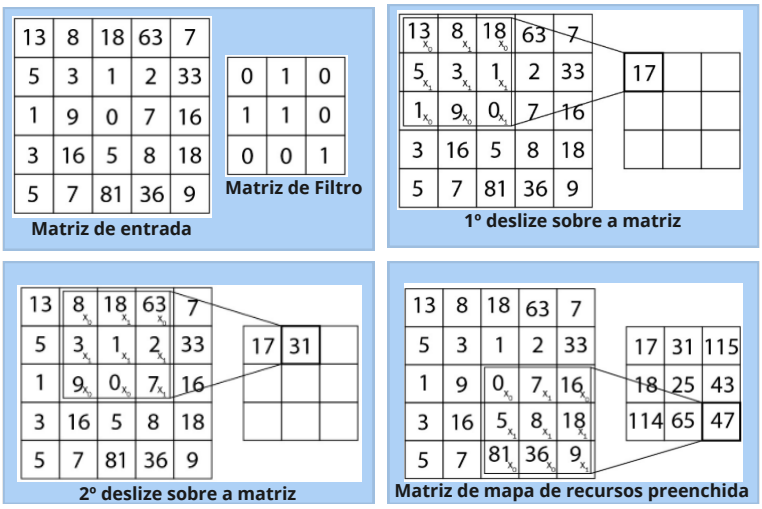

Figura 2.9: Processo de construção do filtro de recursos na etapa da convolução.

A saída da camada convolucional é uma imagem convolucionada. Essa imagem é então passada para a camada de Pooling, a qual reduzirá as dimensões do mapa de recursos a manter apenas os detalhes necessários. Uma operação mais comum nessa camada é aplicação do *Max Pooling*, onde apenas os valores máximos do mapa de recursos são pegos por janela, como mostra a Figura 2.10

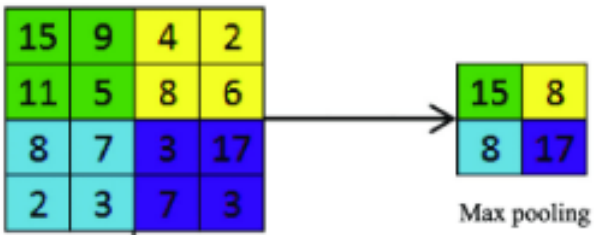

Figura 2.10: Ilustração do processo de Max Pooling.

A camada completamente conectada recebe como entrada a saída da camada de Pooling e realiza o processo como uma rede neural artificial normal, onde é aplicada a função de ativação, produzindo uma saída como resultado. Cada Neurônio em uma camada totalmente conectada tem conexões completas com todas as ativações na camada anterior.

2.2.1 Deep Q-Network (DQN)

O algoritmo clássico de RL, o Q-Learning, se concentra no par de estado e ação para realizar seu aprendizado. Uma tabela Q (também chamada Q-table)

é preenchida iterativamente para cada par de ação e estado em cada interação com o ambiente. Assim, essa tabela acaba tornando-se o "cérebro" desse agente. Ele toma suas próximas decisões a partir dos valores armazenados nessa tabela, pegando a ação que tem o maior valor em um estado. Esse algoritmo é eficiente em ambientes onde os estados são relativamente pequenos.

Em aplicações reais, existem ambientes com numerosos estados e para cada um é possível ter diversas ações, o que aumentaria expressivamente o consumo para preencher a Q-table.

Para resolver essa limitação, pesquisadores do *Deep Mind* da *Google* [91] propuseram usar uma rede neural com pesos θ ao invés de uma tabela para mapear o Q-value. O algoritmo DQN ou Deep Q-Learning [52] utiliza redes neurais convolucionais para aproximar o valor Q de todas as ações possíveis em cada estado [5].

A Figura 2.11 apresenta a diferença entre o algoritmo Q-Learning e o DQN.

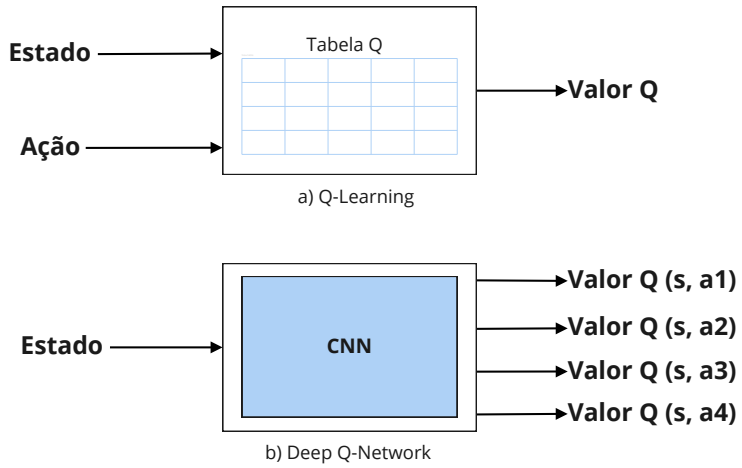

Figura 2.11: A imagem (a) representa o fluxo do algoritmo Q-Learning. A imagem (b) representa o fluxo geral do algoritmo DQN.

A arquitetura DQN é formada por duas redes neurais, a **rede Q** e uma **rede Alvo** (*target network*). Além disso, um componente chamado *Experience Replay* é inserido para interagir com o ambiente e gerar dados para treinamento da rede Q.

A **rede Q** é uma arquitetura de rede neural convolucional padrão. Essa rede é o agente treinado para aproximar o valor Q de todas as ações possíveis em cada estado. A Figura 2.12 apresenta uma ilustração esquemática da rede neural convolucional.

Uma **rede alvo** foi proposta por [52] para tornar o treinamento do algoritmo DQN mais estável. Para encontrar a relação entre um estado atual e seu próximo, é utilizada a equação de Bellman, a qual fornecerá o valor de

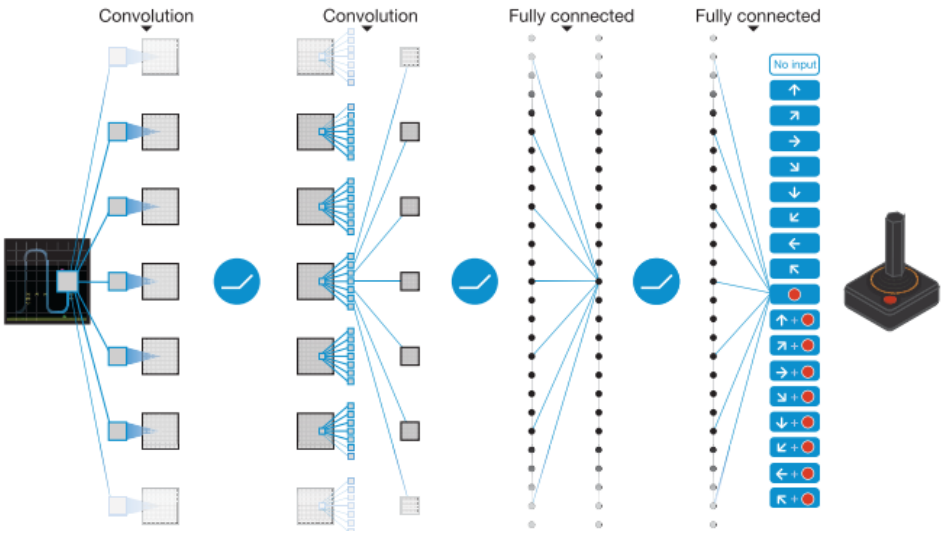

Figura 2.12: Ilustração esquemática da rede neural convolucional. A entrada para a rede neural consiste em uma imagem qualquer de dimensão 84x84x4, a qual segue por três camadas convolucionais e duas camadas totalmente conectadas com uma única saída para cada ação. Fonte [52].

$Q(s, a)$ por meio de $Q(s', a')$. Porém, a diferença entre o estado s e o estado s' é apenas de uma única etapa. Isso os torna muito semelhantes, dificultando uma rede neural de diferenciá-los. Isso pode causar um treinamento instável.

A rede alvo é um clone da rede Q e é usada para retropropagar a rede Q. Inicialmente, seus pesos são congelados com os pesos da rede Q original, sendo atualizados com os novos pesos da rede de política para um determinado período. A equação de perda que calcula a diferença entre a rede alvo e a rede Q é apresentada pela Equação 2-12

$$Loss = (r + \gamma max_{a'} Q(s', a'; \Theta') - Q(s, a; \Theta))^2 \quad (2\text{-}12)$$

onde:

- r = recompensa
- γ = fator de desconto
- Θ' = São os pesos atualizados uma vez a cada passo do *alvo*.
- Θ = Aprende os pesos corretos usando gradiente descendente

A Figura 2.13 ilustra o diagrama de interação entre uma rede Q e a rede Alvo.

Uma *experience replay* [40] serve para armazenar as experiências adquiridas pelo agente RL em cada etapa. Um buffer de memória é usado para armazenar uma quantidade predeterminada das experiências anteriores (batch size) e treina a rede com o mini-lote de amostras armazenadas desse buffer. Em

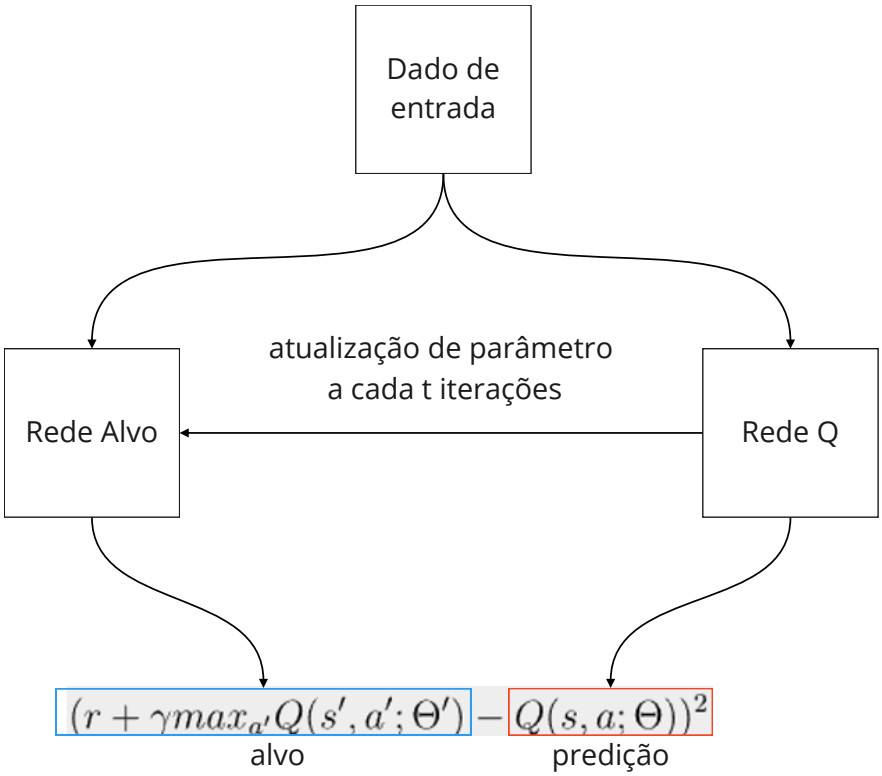

Figura 2.13: Diagrama de interação entre uma rede de política Q e uma rede Alvo.

cada etapa t, uma transição é salva e então usada para treinar a rede neural através de uma descida gradiente estocástica.

Uma transição é uma tupla formada pelo Processo de Decisão de Markov (MDP), composta por (S, A, R, S'), sendo:

- *S (Estado)*: Estado atual.
- *A (Ação)*: Ação realizada no estado atual.
- *R (Recompensa)*: Recompensa obtida por realizar uma ação em um dado estado.
- *S' (Próximo Estado)*: O próximo estado a ser observado.

A Figura 2.14 ilustra o armazenamento de transições em um buffer de memória;

A Figura 2.15 ilustra o fluxo de trabalho do algoritmo DQN.

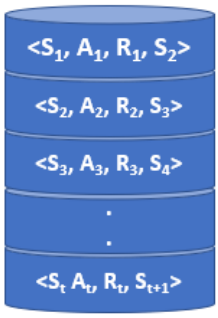

Figura 2.14: Ilustração de armazenamento do Experienciece replay através do algoritmo DQN.

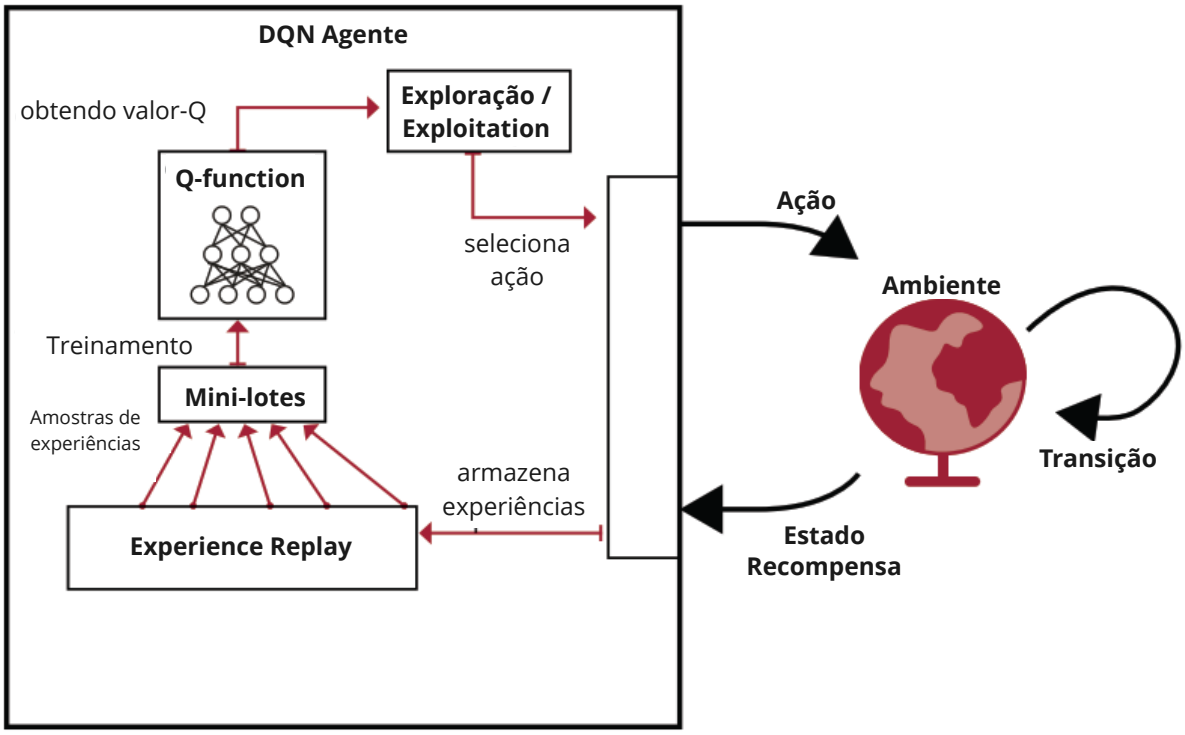

Figura 2.15: Ilustração esquemática do processo de trabalho do algoritmo de DQN. A cada transição (tupla formada pelo estado atual, ação, recompensa e próximo estado) é armazenada em uma memória de experiências (experience replay) e amostras são selecionadas aleatoriamente para treinamento da Rede Q. Através da estratégia de seleção exploração (valor randômico) ou exploitation (estratégia gulosa), uma ação é selecionada para que o agente RL possa aplicá-la no ambiente. Imagem adaptada [56].

2.3 Aprendizagem com dados parcialmente anotados

O sucesso das técnicas de IAs, principalmente aquelas baseadas no aprendizado supervisionado [54, 84], dependem muito da disponibilidade de uma grande quantidade de dados de imagens anotadas [110]. No entanto, na ausência de um conjunto de dados totalmente anotados, surge a necessidade de realizar novas anotações.

Obter novas anotações é uma tarefa cara e demorada. Para geração de novas anotações em imagens, no tipo de caixa delimitadora, geralmente é necessário criar ou encontrar uma ferramenta gráfica que tenha uma interface para possibilitar que anotadores empreguem suas anotações. Esse processo de anotação é geralmente realizado manualmente por especialistas humanos com experiência e pode levar meses nessa construção [54].

Para reduzir os esforços na aquisição de novas anotações, pesquisadores exploram novas abordagens [116] através de aprendizagens semi-supervisionadas *(semi-supervised)* [78], supervisão fraca *(Weak-Supervised)* [104], não supervisionado *(Unsupervised)* [119], auto-supervisionado *(Self-Supervised)* [95], aumento de dados *(Data augmentation)* [102] e aprendizado ativo *(active learning)*, este último descrito na Seção 2.4.

Aprendizagem Semi-Supervisionada é o ramo do aprendizado de máquina voltado para a combinação do uso de dados anotados e não anotados para executar determinadas tarefas de aprendizado [109] e assim aprender a partir dessas amostras.

Existem diversas maneiras de realizar o aprendizado semi-supervisionado, uma delas é combinar algoritmos de *clustering* para agrupar os dados com base em suas semelhanças, ajudando assim no encontro de dados relevantes e utilizando algoritmos de aprendizado supervisionado para uma tarefa convencional como classificação ou detecção. Ao treinar o modelo semi-supervisionado com os poucos dados anotados, ele é utilizado para anotar os dados não anotados e treinado novamente com o novo conjunto de dados anotados. Em seguida, o modelo retreinado é aplicado sobre um conjunto de dados de testes. No entanto, não há como verificar se o algoritmo produziu os rótulos de maneira precisa e consistente, implicando em um resultado menos confiável.

O aprendizado com supervisão fraca é uma abordagem que usa os dados com rótulos imprecisos, ruidosos [99].

O aprendizado fracamente supervisionado tem o foco em três categorias: supervisão incompleta (*incomplete supervision*), onde um subconjunto único de dados de treinamento é fornecido com rótulos; supervisão inexata (*inexact supervision*), onde os dados de treinamento são fornecidos com rótulos, mas não

tão exatos quanto o desejado; e supervisão imprecisa (*inaccurate supervision*), quando nos dados de treinamento existem alguns rótulos com erros. [124].

O aprendizado não supervisionado visa encontrar características em um conjunto de dados de interesse, sem a necessidade de dados anotados que mostrem o padrão que deve ser aprendido. Nesse método de aprendizagem, uma maior correlação dos dados de entrada é aprendida, tendo como saída um conjunto de dados anotados [12].

Um algoritmo não supervisionado funciona analisando os dados sem seus rótulos, determinando as correlações e tentando aprender alguma estrutura inerente aos dados, a partir de exemplos não anotados. A redução de dimensionalidade e o agrupamento são duas tarefas bem conhecidas de aprendizado não supervisionado.

O aprendizado auto-supervisionado é uma abordagem de aprendizado híbrido que combina esquemas de aprendizado supervisionado e não supervisionado [42]. É uma abordagem que aprende recursos semanticamente úteis para uma determinada tarefa, onde os resultados são obtidos por modelos que analisam dados, anotam e categorizam informações independentemente, eliminando a necessidade de dados anotados ou esforços humanos para obtenção de novas anotações.

O aprendizado auto-supervisionado é formado por duas tarefas distintas, chamadas pretexto e *downstream*. Na tarefa de pretexto, o modelo aprende de forma supervisionada, por meio dos dados não anotados. Os conceitos aprendidos na tarefa de pretexto, são transferidos como pesos iniciais para a tarefa de *downstream* que executa as principais tarefas, como tarefas de detecção e classificação.

O aumento de dados tem servido como uma solução eficaz particularmente na ausência de grandes conjuntos de treinamento anotados [102]. Esse método visa aumentar a diversidade de dados de treinamento, sem a necessidade de coletar mais dados diretamente ou realizar novas anotações.

A maioria das estratégias adiciona cópias ligeiramente modificadas de dados existentes ou cria dados sintéticos. Em computação visual, o aumento de dados tem sido amplamente utilizado, visto que o emprego de algumas técnicas, como recorte (*cropping*), rotação (*flipping*) e contraste (*color jittergin*), resulta em uma modificação significativa que pode ser inserida como um novo dado para treinamento do modelo [23]. No entanto, aumento de dados não está restrito apenas às aplicações de computação visual, outras aplicações podem se beneficiar desse método como, por exemplo, aplicações de reconhecimento de voz, onde a inserção de ruídos pode já produzir nova informação de treinamento para o modelo [59, 93].

2.4 Aprendizagem Ativa

Aprendizagem ativa é um importante método para resolver o problema de poucos dados anotados em abordagens de aprendizagem de máquina [120]. Quando há falta de dados na etapa de aprendizagem de um modelo, amostras são anotadas por especialistas humanos, o que é um progresso caro e demorado.

Esse framework é utilizado para obter novas anotações de verdade absoluta (*ground-truth*) sob um conjunto de dados específicos. O objetivo do AL é construir um bom modelo enquanto minimiza os esforços em realizar novas anotações pelo humano [28]. A ideia-chave é: se uma máquina pode escolher os dados com os quais aprende, então o número de instâncias de treinamento necessárias para ele evoluir será muito menor do que se um humano escolhesse os dados com os quais deveria aprender [8]. A Figura 2.16 mostra o diagrama do ciclo de aprendizado ativo clássico.

Figura 2.16: Diagrama de loop de aprendizado ativo. Em cada iteração, a função de pontuação e a estratégia de amostragem na etapa de consulta decidem quais imagens devem ser enviadas para anotação e adicionadas ao conjunto de dados de treinamento para treinamento adicional. Imagem adaptada de [31].

AL se divide em duas categorias principais: baseado em fluxo (*stream-based*) e baseado em pool (*pool-based*) [113].

A amostragem baseada em fluxo também é chamada aprendizado ativo sequencial. As instâncias não anotadas são apresentadas uma a uma da fonte de dados. Uma estratégia de aprendizagem deve decidir se a instância é informativa o suficiente para ser anotada pelo especialista (anotador) ou

descartada. A amostragem baseada em fluxo precisa definir um limite mínimo na avaliação informativa da instância não anotada. Se a avaliação da instância exceder esse limite ela será consultada e anotada.

Na amostragem seletiva baseada em pool, o método de aprendizado acessa um conjunto (pool) de instâncias não anotadas e, em seguida, usa uma espécie de critério para comparar e consultar as instâncias especiais do pool, independentemente de sua ordem individual. A amostragem baseada em pool é amplamente estudada e usada para muitas aplicações reais em aprendizado de máquina. A principal diferença entre pool-based e a amostragem baseada em fluxo é que a baseada em pool avalia todo o conjunto de instâncias não anotadas e, em seguida, seleciona a melhor, enquanto a baseada em fluxo consulta os dados um a um e toma a decisão individualmente. A Figura 2.17 ilustra o fluxo de AL baseado em Pool.

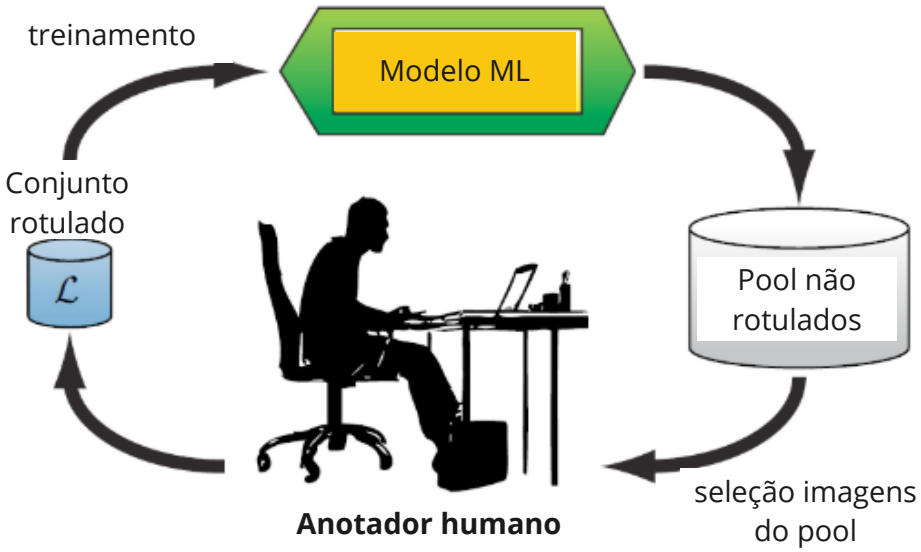

Figura 2.17: Diagrama de loop de aprendizado ativo baseado em Pool. Imagem adaptada de [89].

O principal componente em uma estrutura de aprendizagem ativa é como determinar a "informatividade" de uma amostra [117], muitas vezes referida como estratégia de seleção de amostra ou estratégia de consulta [112, 117].

A estrutura de amostragem mais comum é a amostragem de incerteza, que mostra os pontos de dados sobre os quais o modelo é mais "incerto" [90].

A amostragem de incerteza é um conjunto de técnicas para identificar itens não anotados que estão próximos de um limite de decisão em seu modelo de aprendizado de máquina atual [54]. Existem diferentes categorias de métricas de incerteza, onde as predições são uma distribuição de probabilidade

(x), y_1^* representa a predição mais confiável e y_2^* a segunda mais confiável. O total de rótulos preditos é representado por n.

- **Amostragem de menor confiança (*Least confidence sampling*)**: diferença entre a previsão mais confiável e a com 100% de confiança.

$$\Phi_{LC}(x) = (1 - P_\theta(y^*|x))\frac{n}{n-1} \tag{2-13}$$

- **Amostragem de margem de confiança (*Margin of confidence sampling*)**: diferença entre as duas previsões mais confiantes.

$$\Phi_{MC}(x) = 1 - (P_\theta(y_1^*|x) - P_\theta(y_2^*|x)) \tag{2-14}$$

- **Relação de confiança (*Ratio of confidence*)**: relação entre as duas previsões mais confiantes.

$$\Phi_{RC}(x) = (P_\theta(y_2^*|x)/P_\theta(y_1^*|x)) \tag{2-15}$$

- **Entropia (*Entropy*)**: Diferença entre todas as previsões, conforme definido pela teoria da informação.

$$\frac{-\sum_y P_\theta(y|x|)log_2 P_\theta(y|x|)}{log_2(n)} \tag{2-16}$$

3
Trabalhos Relacionados

Neste capítulo contextualizam-se alguns estudos encontrados na literatura apresentando soluções já existentes que usam algoritmos de RL e integram o humano no loop de treinamento para possibilitar um aprendizado eficaz ao agente. Esses trabalhos foram encontrados e analisados a partir de buscas por palavras-chave, as quais foram: *"reinforcement learning"; "learning with human in the loop"; "reinforcement learning advice"; "Deep Reinforcement Learning"; "Deep Q-Network"; "DQN"; "data-centric ai"; "reduction of datasets annotations"; "active learning"; "data annotations AND machine learning"*. Alguns trabalhos foram encontrados a partir de citações bibliográficas.

Destacamos algumas abordagens propostas onde a inserção de uma ação humana, seja para a tomada de decisão do agente ou para atribuir uma recompensa após uma determinada ação, influencia diretamente no seu aprendizado. Também serão evidenciados as limitações observadas nos estudos.

Ainda neste capítulo são apresentados estudos voltados para a falta de dados na etapa de aprendizagem de um modelo, em que para a obtenção de novas amostras grandes esforços são realizados por especialistas humanos, ocasionando um processo caro e demorado.

Embora na área de aprendizado de máquina haja diversos trabalhos na literatura voltados para as diversas ramificações de ML, os trabalhos apresentados nesse capítulo estão voltados para abordagens que utilizam RL. Desse modo, uma maior compreensão e entendimento é dado ao leitor sobre o que está sendo desenvolvido em trabalhos de pesquisas relevantes ao contexto desta tese, bem como suas limitações.

3.1
Aprendizado por Reforço com Humano no Loop

A inclusão humana no ciclo de treinamento de um agente RL é influenciada pela capacidade do ser humano de ensinar tarefas, avaliar o desempenho e intervir em determinados momentos para evitar ações desastrosas. Essa inclusão pode aumentar a velocidade do agente RL em aprender uma tarefa, tornando-o confiante para tomar decisões rápidas e precisas, conforme destacado por Liang et al. [43].

Uma abordagem onde um treinador humano aplica *feedbacks* enquanto observa o aprendizado do agente é apresentada por Thomas et al. [115]. Chamado pelos autores de *Deep Policy Shaping (DPS)*, este método ajusta a política de aprendizagem do agente, segundo o *feedback* enviado pelo humano.

O DPS usa uma rede neural de *feedback* que aprende a otimização das ações humanas ruidosas combinado com um algoritmo RL, o DQN. Para melhorar a qualidade das estimativas de incerteza da rede *feedback* proposta pelos autores, são utilizados *ensembles* profundos. A utilização desse método permite a exploração em aplicações onde os espaços de estados são contínuos e de alta dimensão. Através de experimentos utilizando ambiente do OpenAI [7], DPS superou ou correspondeu às linhas de base médias em várias configurações de hiperparâmetros.

Um dos principais desafios dessa abordagem é interpretar corretamente o feedback humano, visto que tal interpretação determina como essa inserção é usada para melhorar a política na estrutura do MDP.

A abordagem proposta no trabalho [115] não apresenta um estudo que inclua a adição de aprendizado ativo ao DPS, visando reduzir a quantidade total de feedback necessário no aprendizado do agente. Além disso, falta realizar experimentos em aplicações reais, para ver como se comporta esse método em cenários mais complexos.

Outra abordagem de interação humana e agente RL é apresentada por Samantha Krening [35]. A autora apresenta um novo método para que um humano consiga aconselhar verbalmente quais ações um agente RL deve realizar, a qual chama de *Newtonian Action Advice (NAA)*.

NAA permite que um agente RL aprenda sua tarefa a partir de conselhos de ação humana. Quando um conselho é fornecido, por exemplo, "vá para frente", o agente imediatamente tomará a ação aconselhada, substituindo a indicada pela sua própria política. A ação se repetirá até que o "atrito" faça com que o agente retome à exploração normal. O atrito é um parâmetro importante em NAA, pois garante que, após algum tempo, o agente retomará a política de exploração do algoritmo RL. Além disso, o algoritmo NAA, permite que o agente siga o mesmo conselho, sempre que o mesmo estado for observado no futuro. Assim, o humano só terá que fornecer conselhos uma vez para uma determinada situação.

Como limitação, este trabalho não apresenta um estudo visando reduzir os esforços humanos em gerar novas anotações na forma de caixa delimitadora para o problema de detecção de objetos.

Um dos desafios, ao criar sistemas onde o humano participa do ciclo de treinamento de um algoritmo, é saber quando e onde integrá-lo. Objetivando

encontrar respostas para esse desafio, os autores Felipe Leno et al. [15], propõem um framework denominado *Requesting Confidence-Moderated Policy advice (RCMP).*

RCMP é um algoritmo que permite um agente RL receber conselhos humanos de quais ações tomar, quando sua incerteza é alta para cada estado explorado. Mas, quando o agente RL possui uma incerteza baixa, indica que está conseguindo explorar o cenário, com uma convergência a partir do estado atual. Deste modo, os conselhos limitados podem ficar para os estados mais críticos. Essa abordagem segue a mesma apresentada por *Osband et al.*[65], chamada *Bootstrapped DQN*, onde consiste em aprender simultaneamente múltiplas estimativas da função valor de uma única rede.

A abordagem apresentada pelos autores Felipe Leno et al. [15] não apresenta um estudo que inclua a adição de aprendizado ativo, visando selecionar imagens específicas para o aprendizado do agente RL. Também seus resultados foram obtidos em um ambiente simulado, faltando experimentos em aplicações reais.

Em aprendizado por reforço profundo, uma questão em aberto é o tempo excessivo que um agente RL gasta para conseguir aprender uma tarefa a partir de imagens de entrada bruta. Pensando apresentar uma proposta para esse problema, os autores Ithan et al. [57] apresentam um estudo avaliativo de como o feedback interativo pode afetar o desempenho de um agente RL.

Os autores apresentam uma implementação de três abordagens. DeepRL, onde o agente interage de forma autônoma com o ambiente; agente-IDeepRL: em que a abordagem DeepRL é complementada com um agente artificial previamente treinado para dar conselhos; human–IDeepRL: onde a abordagem DeepRL é complementada com um treinador humano. Para comparar os algoritmos DeepRL e IDeepRL, os agentes são treinados em um cenário doméstico simulado, onde um braço robótico possui o objetivo de organizar os objetos em cena. Como resultado do estudo, a abordagem interativa de aprendizado por reforço profundo apresenta uma vantagem em ambientes domésticos. Permite acelerar o processo de aprendizagem de um agente robótico interagindo com o ambiente e permite que as pessoas transfiram conhecimento prévio sobre uma tarefa específica.

O trabalho apresentado pelos autores Ithan et al. [57] não considera as dinâmicas e variáveis de ambientes reais, já que seus experimentos foram através de um ambiente simulado.

Ainda seguindo sobre aconselhamento a agente RL em um cenário de robô doméstico, os autores Francisco et al. [14] apresentam quatro abordagens de aconselhamentos: *probabilistic advising, early advising, importance advising,*

e *mistake correcting*.

Sabendo que a tarefa de aconselhamento não deve ser com muita frequência, já que o ser humano tende a não prolongar paciência para realizar tarefas repetitivas, a ideia de conselhamento limitado é apresentada pelos autores, conforme já estudado por [105]. A inserção de um orçamento, ou seja, um número fixo que impacta na frequência com que o aconselhamento pode acontecer durante um episódio de aprendizagem do agente RL. A abordagem *early advising* simplesmente permite que o agente-treinador gaste seu orçamento o mais rápido possível, ou seja, nos primeiros n estados que o agente-aprendiz se encontra. *Importance advising* considera que dar conselhos pode ser mais crucial em algumas situações do que em outras. *Mistake correcting* difere das outras estratégias, pois considera a decisão que o agente-aprendiz toma autonomamente antes de decidir se deve ou não dar conselhos. Por fim, a abordagem *probabilistic advising* não usa um orçamento de aconselhamento, portanto, o agente-treinador pode aconselhar o aluno a qualquer momento durante o processo de aprendizagem com base em uma probabilidade de interação fixa. Os experimentos com os métodos de aconselhamentos foram realizados em um cenário onde um robô aprendia a limpar uma mesa. Os resultados mostraram que as abordagens de aconselhamento apresentam uma evolução eficiente no aprendizado de um agente RL. Para o cenário apresentado, a abordagem *mistake correcting* coletou a maior quantidade de recompensas.

Como os métodos foram aplicados sob um ambiente controlado, fica difícil afirmar qual método permitirá uma evolução do aprendizado do agente RL em um ambiente real.

Os autores [45] apresentam um método para usar feedback humano discreto para aprimorar o desempenho de agentes de aprendizado profundo em ambientes tridimensionais virtuais. Sabendo que o feedback humano não é uma tarefa trivial, apresentam alguns fatores que devem ser observados, tais como: *Inconsistência*, onde é improvável que um professor humano consiga dar feedback correto consistentemente; *Feedback intermitente*, também não é garantido que os humanos fornecerão feedback sobre cada ação que o agente executa. *Diferentes escalas de recompensa e feedback*, na interpretação mais simples, o feedback humano pode ser considerado parte do sinal de recompensa. Em DRL em mundos 3D, o verdadeiro estado do agente e do ambiente deve ser inferido a partir de observações ao nível de pixel que podem conter uma quantidade não trivial de ruído do sensor, torna-se difícil determinar em quais estados aplicar valores de feedback humano.

Os autores estendem as técnicas de DRL para aprender com o conselho humano, considerando que esse conselho pode ser inconsistente e intermitente.

Portanto, apresentam uma abordagem que faz o uso de um *arbiter*, o qual decide quando um agente RL executa uma ação de sua política ou segue o conselho humano. O árbitro mede a confiança na rede Q profunda à medida que aprende uma função de perda de rede. Essa medição é feita por três abordagens probabilísticas, *Exploration check*, a verificação de exploração faz com que o agente siga uma estratégia gulosa épsilon, forçando o agente a explorar uma certa porcentagem do tempo. *Confidence check*, esta verificação usa uma medida de quão confiante o agente está na sugestão do DQN. Quando a confiança no DQN for baixa, o agente-aprendiz preferirá a sugestão de ação (se houver) do humano. *Consensus Check*, esta verificação usa uma medida para alinhar a ação do humano com a da política. A verificação de consenso é usada para neutralizar a inconsistência do humano, já que o humano pode variar em suas ações. Os autores realizaram experimentos de suas abordagens em um cenário virtual do jogo *Minecraft*. Os resultados mostram que a técnica não apresenta desempenho inferior à linha de base. O agente não se beneficia do feedback do humano em mapa fácil porque é relativamente trivial encontrar a política ideal.

Como os métodos foram aplicados sob um ambiente controlado, fica difícil afirmar qual método permitirá uma evolução do aprendizado do agente RL em um ambiente real.

No trabalho dos autores Travis et al. [50], é apresentada uma estrutura que possibilita que o ser humano adiciona ações a um sistema de aprendizado por reforço ao longo do tempo, visando aumentar o desempenho do aprendizado do agente RL. Denominado *Expected Local Improvement (ELI)*, o método proposto tem como intuito selecionar os estados nos quais as ações humanas beneficiarão no aprendizado do agente RL. Assim, o ser humano não desperdiçará seus esforços com ações em lugares não muito aproveitosos para o aprendizado.

ELI, é uma heurística para selecionar o estado onde a próxima ação deve ser adicionada. Essa abordagem foi avaliada em uma variedade de domínios simulados adaptados da literatura, incluindo domínios com mais de um milhão de ações. Em cada ambiente avaliado, o método e suas variações, apresentaram resultados satisfatórios.

Por exemplo, quais UIs e visualizações fazem com que os humanos desenvolvam as ações mais úteis em resposta a uma solicitação? Ou quão restrita deve ser a linguagem das ações? Ou, como devemos orientar os humanos sobre quando (não apenas onde) eles devem adicionar ações? Trabalhos futuros estudarão essas questões.

3.2 Redução de custo em anotações

Na área de visão computacional, para o sucesso dos algoritmos de aprendizagem profunda, é indispensável anotações apropriadas para treinamento. Em certas tarefas em visão computacional, como por exemplo em aplicações voltadas para área médica, como classificação de patologias e detecção de estruturas anatômicas, as anotações das imagens são fundamentais. No entanto, a criação de um conjunto de dados com qualidade, através de anotações manuais, consome muito tempo e é uma operação cara. Além disso, algumas anotações podem apresentar inconsistências, causadas pelas diferentes interpretações dos anotadores, implicando diretamente no desempenho do modelo.

Para reduzir os esforços nas anotações, os pesquisadores exploraram abordagens de anotação de dados que não gerem muitos esforços humanos e que seja realizada mais rápida, reduzindo o custo dessa operação [116]. Um exemplo dessa abordagem são os algoritmos de aprendizado ativo. Esses algoritmos visam reduzir o custo de anotação, selecionando apenas as imagens mais informativas a serem anotadas pelo ser humano, para melhorar a precisão de um modelo [10].

Os autores Ismail et al. [18] apresentam um framework de aprendizado ativo, que considera tanto a incerteza quanto a robustez de um detector, garantindo que a rede funcione com a precisão em todas as classes. Dois componentes são os principais apresentados pelos autores: uma nova função para calcular o *score* do aprendizado ativo baseado na robustez da rede, usando um novo *score* de consistência; e outro componente é um esquema de pseudo-rotulagem para amostras fáceis.

Seja D um conjunto de dados dividido em um conjunto anotado L e um conjunto de dados não anotados U. Inicialmente é utilizada uma rede de detecção profunda de objetos e, em seguida, um subconjunto de amostras do conjunto de dados U não anotados é extraído e transferido para o conjunto anotado L, incluindo o custo de anotação.

O objetivo é selecionar as amostras sólidas que impactarão na confiança do aprendizado supervisionado durante o treinamento. No entanto, aumentar arbitrariamente o conjunto L com apenas amostras difíceis cria um desvio de distribuição dos dados de treinamento. Portanto, é proposto incluir no treinamento as amostras fáceis, ou seja, objetos para os quais a confiança da rede é alta, usando pseudo-rotulagem. O treinamento da rede é feita com esse novo conjunto de dados. Esse processo se repete até terminar o ciclo do AL. A demonstração da abordagem apresentada pelos autores foi através de experimentos em dois conjuntos de dados públicos: PASCAL VOC07+12 [18]

e MS-COCO train2014[44]. Como métrica principal é utilizada a média da precisão (mAP).

Os autores destacam como limitação de seu método a falta de possibilidade em funcionar com classes desconhecidas em um ambiente de mundo aberto. Além disso, o método proposto é específico para uma tarefa, portanto, menos adequado para adquirir conjuntos de dados para redes multitarefa. Visando reduzir os esforços para realizar novas anotações em um conjunto de dados, a autora Priyanka [96] apresenta um novo paradigma de aprendizagem chamado Aprendizagem Interativa. Esse modelo de aprendizagem é atualizado continuamente através de consultas ao humano/oráculo que verifica a anotação proposta.

Os autores exploram dois métodos para realizar a aprendizagem interativa: aprendizagem incremental ou Randômica e aprendizagem ativa. Na aprendizagem incremental, um conjunto fixo de exemplos é escolhido aleatoriamente, de um conjunto de dados não anotados, para verificar o resultado do modelo mais recente. O modelo é então atualizado com base nas anotações recém-obtidas. A outra abordagem é a aprendizagem ativa, onde um conjunto de imagens específicas, mas benéficas para o modelo, são escolhidas para a realização das anotações pelo humano. A ideia é maximizar o desempenho do modelo e reduzir os esforços humanos na anotação. Os autores avaliaram sua respectiva abordagem sobre conjuntos de dados como PASCAL VOC07+12 [18], The Weizmann Horse[6], CZHeLa e os resultados experimentais mostram que ao empregar o AL, após o treinamento de cerca de 2,5% dos dados, quase o mesmo desempenho do treinamento em todo o conjunto de dados pode ser alcançado.

Saripalli et al. [85] apresentam uma abordagem para contribuir com o processo de anotação, onde os dados de dispositivos de monitoramento de saúde precisam ser interpretados. Os autores usaram algoritmos RL para criar um agente virtual capaz de anotar dados de alarme, com base nas anotações feitas por um especialista.

A criação do agente foi através de dois algoritmos DQN e A2C (*Advantage Actor Critic*), com o objetivo de anotar dados de sinais médicos embasado no fato de representar ou não um estado de alarme. A proposta é permitir que os agentes aprendam a realizar tomadas de decisão semelhante ao especialista, onde o aprendizado é realizado sem quaisquer suposições sobre o tipo de sinal ou pré-codificação de qualquer conhecimento do domínio. Assim que o agente RL atingir um desempenho razoável, será possível substituir o especialista humano pelo agente RL para anotar os dados. Como resultado, a abordagem apresentada pelos autores criou um especialista virtual com alta sensibilidade,

capaz de distinguir um número notável de alarmes falsos.

As limitações deste trabalho são os poucos dados utilizados tanto para treinamento quanto teste, sendo restrito a uma semana de dados e a tarefa de detecção de alarme é limitada a duas classes.

Wang et al. [111] apresentam o *Deep Reinforcement Learning Active (DRLA)*, um método para classificação de imagens médicas. Este método utiliza o algoritmo DQN aplicado com o paradigma ator-crítico, para criar um agente capaz de aprender uma política de seleção de imagens mais informativas a ser anotada por um humano.

Uma rede classificadora para diagnóstico de doenças é treinada com dados anotados, incluindo amostras com rótulo antecipado e amostras selecionadas e anotadas no procedimento de aprendizado ativo. Durante o processo de aprendizado, uma rede de atores é dedicada a selecionar as amostras mais informativas de dados de treinamento não anotados, conforme o estado atual e uma política aprendida. Depois disso, um anotador é responsável por anotar as amostras selecionadas. Como resultado, a cada ciclo obtém-se cada vez mais dados de treinamento anotados para atualizar o classificador gradualmente. Por último, mas não menos importante, uma rede crítica é treinada para avaliar se a seleção da rede de atores é efetiva para melhorar o desempenho do classificador. Ao empregar uma abordagem de aprendizado por reforço profundo para treinar a rede de atores e a rede crítica, é possível selecionar e anotar as amostras mais informativas benéficas, para treinar um classificador eficaz e melhorar ainda mais o seu desempenho. A abordagem foi avaliada em um conjunto de dados de tomografia computadorizada e retinopatia. Como resultado, o método apresentou uma abordagem prática para aliviar os esforços humanos em fazer anotações, mostrando que pode reduzi-los.

Zimo et al. [49] propõem outra abordagem usando aprendizagem ativa e a denominam *Deep Reinforcement Active Learning (DRAL)*. O objetivo do estudo é minimizar os esforços humanos para obter anotações. Para isso, propõem esse método onde a seleção de amostras de treinamento são realizadas por um agente RL.

Aplicado no caso de re-identificação, o agente RL aprende a selecionar o melhor par de imagens para o anotador humano, que dará feedback binário para anotar a imagem como certa ou errada. A cada feedback do humano, uma recompensa é dada ao agente. A ação que o agente RL aplica sobre os estados determina qual candidato de um conjunto de dados não anotados será enviado para o anotador humano para consulta. A recompensa é calculada com diferentes feedbacks humanos. Uma CNN é adotada para inicialização dos estados. Para validar a abordagem, os experimentos foram realizados em 4 conjuntos

de dados, o Market-1501 [122] é amplamente adaptado em um conjunto de dados de re-identificação de larga escala, CUHK01 [41] é um conjundo de dados de re-identificação de baixa escala e DukeMTMC-ReID(Duke) [82] é um dos mais populares conjuntos de dados de re-identificação de grande escala. Extensos experimentos demonstram a superioridade do método DRAL para re-ID de pessoa humana em loop baseado em aprendizado de reforço profundo quando comparado aos modelos existentes de aprendizado não supervisionado e de transferência, bem como modelos de aprendizado ativo.

Sun e Gong [98] também apresentam um novo framework que usa aprendizado ativo para anotar imagens. Eles propuseram uma estrutura que usa o DRL como estratégia de seleção de dados. Em vez de escolher qual imagem anotar usando algoritmos heurísticos, o algoritmo RL aprende uma política de seleção.

Os autores descrevem o processo de aprendizagem ativa como um problema de aprendizado por reforço. A aprendizagem ativa pode ser vista como um procedimento iterativo de amostragem e anotação. O agente RL deve decidir sobre anotar ou não os dados conforme as observações atuais, que incluem as características extraídas pelo modelo CNN. Os autores avaliaram o método com outros estudos do estado da arte que obtiveram resultados superiores em um conjunto de dados populares. As avaliações de desempenho foram realizadas usando três conjuntos de dados de classificação de imagem: CIFAR-1O [36], CIFAR-1OO [36] e SVHN [61].

3.3 Análise Comparativa dos Estudos

Os resultados apresentados pelos estudos selecionados são baseados em abordagens que envolvem o ser humano na etapa de treinamento de um algoritmo. Em alguns trabalhos, a inserção humana contribui para o aprendizado do agente RL, já em outros, o humano é inserido como anotador de novas imagens que contribuirão no aprendizado de algoritmos supervisionados.

Uma parte dos estudos apresentados utilizam estratégias de aprendizagem a partir da abordagem de aprendizado por reforço. Dentre os algoritmos de RL, os mais utilizados pelos autores são aqueles que envolvem abordagens com redes neurais convolucionais. Esses algoritmos compõem as abordagens propostas pelos autores nos mais diversos cenários. No entanto, a maioria desses experimentos está limitada a ambientes simulados, dificultando comparações com aplicações reais. A Tabela 3.1 apresenta a relação dos cenários utilizados pelos autores para seus respectivos experimentos, bem como o algoritmo de RL adotado e o método de interação.

Tabela 3.1: Estudos com suas respectivas estratégias.

	Estratégias utilizadas					
Trabalho	Humano no loop	Aprendizagem por reforço profundo (DRL)	Aprendizagem ativa (AL)	Experimento em ambiente real	Geração autônoma de caixas delimitadoras	Metodologia para treinamento com poucos dados
Thomas et al. [115]	✓	✓	✗	✗	✗	✗
Samantha [35]	✓	✗	✗	✗	✗	✗
Felipe Leno et al. [15]	✓	✓	✓	✗	✗	✗
Ithan et al.[57]	✓	✓	✗	✗	✗	✗
Francisco et al. [14]	✓	✗	✗	✗	✗	✗
Zhiyu et al. [45]	✓	✓	✗	✗	✗	✗
Travis et al. [50]	✓	✗	✗	✗	✗	✗
Ismail et al. [18]	✗	✗	✓	✗	✗	✓
Priyanka [96]	✗	✗	✓	✗	✗	✓
Saripalli et al. [85]	✗	✓	✗	✓	✗	✗
Wang et al. [111]	✗	✓	✓	✓	✗	✓
Zimo et al. [49]	✓	✗	✓	✓	✗	✓
Sun e Gong [98]	✗	✓	✓	✓	✗	✓
Esta Tese (Seções 4 e 5)	✓	✓	✓	✓	✓	✓

Como apresentado nos estudos acima, as abordagens utilizam a soma de no máximo dois métodos, DQN com aconselhamento ou DQN com AL. Nesse ponto, propomos a combinação de três técnicas para resolver o problema de redução dos esforços na aquisição de novos dados de treinamento. Assim, temos uma abordagem centrada nos dados de modo que a preparação dos dados de aplicações reais, na etapa de geração de caixas delimitadoras, seja feita automaticamente mesmo em posse de poucos dados anotados. Será adotado o algoritmo Deep Q-Network (DQN) integrado com o humano no loop de treinamento de modo que seja um "professor" ensinando um agente RL a realizar anotações corretamente. Além disso, empregaremos a técnica de AL, para que dados específicos sejam anotados e assim contribuir na evolução do aprendizado do agente.

Na literatura existem outras propostas para integrar o humano no processo de treinamento de um algoritmo DRL, tais como por demonstração [39], imitação [60], e métodos heurísticos para selecionar um estado onde o ser humano deve adicionar ações ao sistema de RL, como mostra o trabalho [50]. Todavia, esses métodos precisam de averiguações adicionais para o treinamento do agente. A abordagem proposta nesse trabalho, visa criar um agente RL capaz de criar anotações a partir de poucas interações humanas, reduzindo assim o custo de se gerar novas anotações.

4
Desenvolvimento de um agente virtual para geração de anotações em imagens de forma autônoma

Este capítulo apresenta o desenvolvimento de aprendizagem de um agente RL para criar anotações em imagens de forma autônoma. Técnicas de aprendizado profundo têm mostrado contribuições significativas para diversas áreas, incluindo análise de imagens médicas [1, 48, 97, 106]. Para tarefas de aprendizado supervisionado, o desempenho dessas técnicas depende de uma grande quantidade de dados de treinamento e de dados anotados. No entanto, a anotação é um processo caro e demorado. Uma visão geral sobre este problema real é apresentado na Seção 4.1. A partir de uma limitação real, apresentamos uma nova abordagem baseada em DRL para redução dos esforços humanos em gerar novas anotações e consequentemente reduzir o custo dessa operação. Nossa abordagem consiste em um agente virtual que consiga aprender como realizar anotações e assim etiquetar automaticamente os dados de treinamento, e um humano no circuito para auxiliar no treinamento do agente (Seção 4.2). Nossa abordagem foi avaliada em um conjunto de dados de raios-X médicos em diferentes casos de uso, onde o agente foi instruído a criar novas anotações na forma de caixas delimitadoras de dados não anotados. (Seção 4.3). Na Seção 4.4, são apresentados os resultados encontrados durante os experimentos. Por fim, as conclusões são apresentadas e discutidas na Seção 4.5.

4.1
Visão Geral

Apresentamos uma abordagem que visa contribuir para anotações escassas. Em particular, focamos na criação de novas anotações automaticamente em exames médicos, reduzindo o tempo e o custo das anotações. Para atender a proposta, utilizamos dois objetivos: 1) Criação de um agente autônomo baseado em técnicas de aprendizado por reforço [101] com o objetivo de criar anotações. 2) inserção humana no processo de treinamento: ensinar o agente autônomo a realizar sua tarefa corretamente mesmo com escassas anotações.

Como já introduzido no capítulo de fundamentos teóricos, RL é um paradigma de aprendizado de máquina que consiste em como um agente virtual (vamos adotar o termo agente RL) encontra uma solução para um determinado

problema, explorando interações no ambiente. Mnih et al [51] propuseram *Deep Reinforcement Learning (DRL)* que combina RL e Rede Neural Convolucional (CNN). Este modelo é uma CNN treinada com uma variante do algoritmo RL chamada *Q-Learning*. Este método visa possibilitar a conexão entre um algoritmo RL e algoritmos de redes neurais profundas, operando sobre os *pixels* das imagens.

Nos últimos anos, os modelos DRL alcançaram avanços que superam o desempenho humano em jogos como Atari [52], também se mostraram promissores em permitir que robôs físicos aprendam habilidades complexas no mundo real [32] e na implantação do mundo real de condução autônoma [33]. Tradicionalmente, o DRL emprega um tipo de algoritmo que é *Deep Q-Network (DQN)* [51] [103].

Nosso estudo mostra um agente RL para anotação automática, onde incluímos o humano no loop de treinamento do algoritmo RL. Essa inclusão se deve à capacidade do ser humano de ensinar tarefas, avaliar desempenho, intervir em determinados momentos para evitar ações indesejadas e aumentar a eficiência de aprendizado do agente RL.

4.2 Solução proposta

Integramos o humano no loop de treinamento para contribuir com o processo de aprendizagem do agente RL. Com isso, o agente pode gerar um número mais significativo de anotações a partir de algumas amostras de dados anotados.

Usamos o algoritmo DQN para o processo de aprendizado do agente. Ele usa redes neurais convolucionais (CNN) para aproximar o valor da função Q de todas as ações possíveis em cada estado. Duas técnicas são a base para o sucesso deste algoritmo: replay de experiência e rede de destino, como explicado na Seção 2.2.1.

4.2.1 Implementação

Com base no estudo de Caicedo et al. [11, 66], começamos implementando o algoritmo DQN para localizar objetos em imagens bidimensionais (2D).

Em cada etapa, o agente RL observa o estado atual (região de uma imagem) e estima as recompensas potenciais com base no custo de realizar diferentes ações. Após esse cálculo, ele seleciona a ação que o levará a receber a recompensa máxima e passa para o próximo estado. Este processo é repetido

até atingir o estado terminal. Este ciclo dentro do RL é chamado de episódio. A seguir, um mapeamento do MDP para o contexto do nosso trabalho.

1. Estados: Inicialmente, a área de visualização do agente RL é do tamanho da imagem e servirá como dados de entrada para a rede (estado atual). A cada passo do algoritmo, o agente analisa pixels da imagem dentro de sua área de visualização e assim calcula a melhor ação a ser tomada. A cada ação realizada pelo agente RL, sua área de visualização será ajustada. O próximo estado é a imagem atual e a área de visualização do agente é ajustada pela última ação realizada. O estado terminal é alcançado de duas maneiras: a primeira é quando o agente RL aplica uma ação final, indicando que localizou a estrutura de interesse. Nesse caso, é criada uma caixa delimitadora sobre a região da imagem observada pelo agente; a segunda, é quando a quantidade de passos (interações) que o agente RL pode aplicar no ambiente, chega ao fim, ocasionando o insucesso. Essa quantidade é especificada como um hiperparâmetro do algoritmo.

   O estado terminal é quando o agente para de realizar ações porque já concluiu sua pesquisa. Neste caso, é criada uma caixa delimitadora caso seja encontrado um objeto.

2. Ações: Adotamos um conjunto de nove ações que o agente RL pode realizar no estado atual, oito delas aplicadas à deformação da área de visão do agente e uma para indicar o estado terminal, conforme mostrado na Figura 4.1. À medida que o agente realiza suas ações, sua área de visualização vai sendo ajustada até encontrar o objeto de interesse.

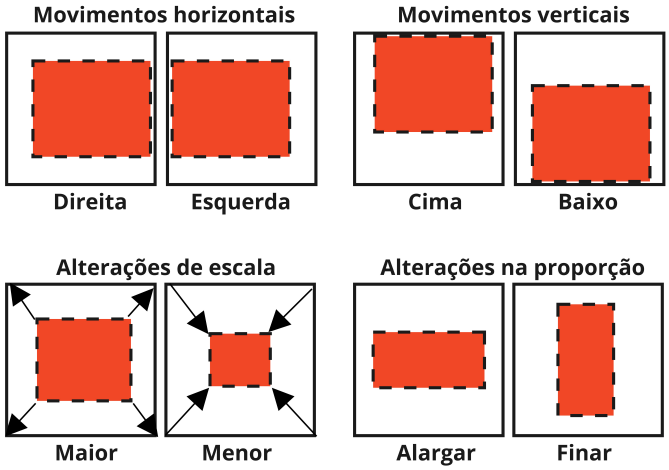

Figura 4.1: Ilustração das ações que o agente de RL realiza nos Estados.

A Figura 4.2 ilustra as ações que o agente RL realiza para detectar uma região de interesse.

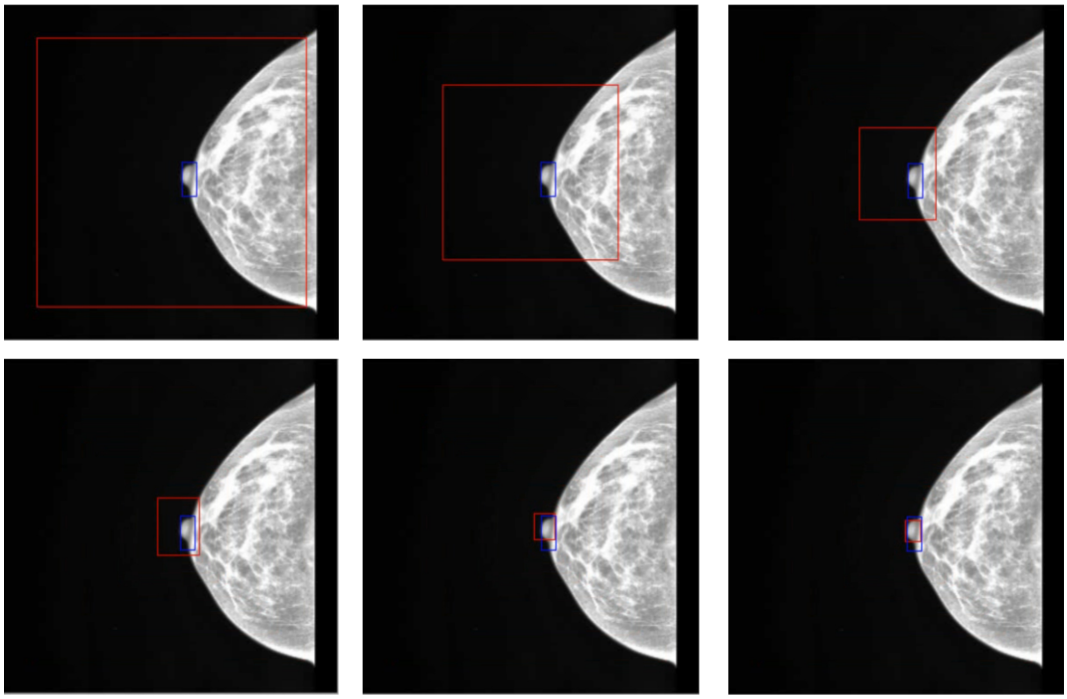

Figura 4.2: Imagem ilustrativa do agente RL criando uma anotação em forma de caixa delimitadora da papila em um exame de mamografia.

3. Recompensa: A função de recompensa utilizada para este trabalho é a mesma apresentada por Caicedo et al. [11]. A equação 4-1 é calculada para atribuir recompensas ao agente RL para cada ação realizada. Esta equação é formada pela área de visualização atual do agente RL b, juntamente com a verdade do objeto alvo a ser localizado g, e b' é a área de visualização na próxima etapa. Em geral, esta função atribuirá uma recompensa positiva ao agente se a ação realizada aproximar o agente de seu objetivo, caso contrário, recompensas negativas são emitidas. Para verificar essa aproximação entre essas duas áreas, é utilizada uma métrica de avaliação denominada *Intersection over Union (IoU)*. A IoU é uma métrica de avaliação usada para medir a precisão de um detector de objetos em um conjunto de dados específico. É uma medida da sobreposição entre duas áreas, a da caixa delimitadora gerada pelo algoritmo e a caixa delimitadora da verdade absoluta. [68].

   a IoU entre o estado atual e o próximo, caso contrário, a recompensa será negativa, como representa a Equação 4-2.

$$RewSign_a(s, s') = sign(IoU(b', g) - IoU(b, g)) \tag{4-1}$$

$$\begin{cases} +1, & \text{if } RewSign_a(s, s') > 0 \\ -1, & \text{Caso contrário} \end{cases} \tag{4-2}$$

A equação 4-3 recompensa o agente quando atinge o estado terminal de acordo com o resultado final. Nesse caso, verificamos se a IoU é maior ou igual ao limite t (adotamos $t = 0,3$ e $0,5$, dependendo do caso de uso). Com isso, o agente recebe uma recompensa positiva ou negativa.

$$\begin{cases} +3, & \text{if } IoU \geq t \\ -3, & \text{Caso contrário} \end{cases} \tag{4-3}$$

A Tabela 4.1 resume os hiperparâmetros utilizados para o treinamento do agente RL.

Tabela 4.1: Hiperparâmetros de aprendizagem

Parâmetros	**Valores**
Tamanho do buffer de memória	50000
Número de episódios	5
Fator desconto	0.99
Total de passos	700
Taxa de aprendizagem	0.00025
Começo do épsilon	1.0
Fim do épsilon	0.2
Batch size	32
Otimizador	RMSProp

A arquitetura DQN usa uma sequência de camadas de uma rede convolucional para extrair características da imagem em um vetor para reduzir a complexidade. A entrada para a rede são as áreas de visualização da imagem, gerada em cada estado. É comum reduzir a resolução do pixel e converter os valores RGB em valores de escala de cinza para reduzir a computação e consumir menos memória. Camadas totalmente conectadas são usadas com uma função de ativação para estimar valores Q diretamente da imagem. A última camada define o número de unidades da camada de saída de acordo com as ações possíveis no ambiente. A Figura 4.3 mostra a arquitetura DQN utilizada.

Para a implementação de um método de aconselhamento, foi preciso identificar quando aconselhar. Portanto, analisamos a incerteza do agente em realizar ações em um determinado estado. A incerteza nesse momento é adotada a partir da verificação da função de perda do próprio algoritmo DQN, como mostra a Equação 2-12. Portanto, em cada etapa o agente RL observa o estado atual (região de uma imagem) e estima as recompensas potenciais com base no custo de realizar diferentes ações naquele estado. Após

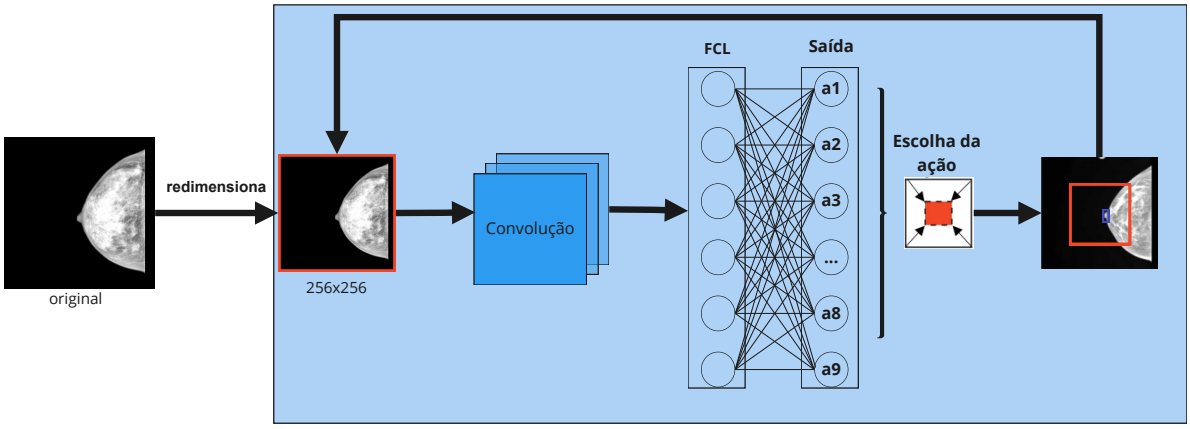

Figura 4.3: Arquitetura usada para o algoritmo DQN. A entrada é uma imagem com 256 x 256 pixels e processada por camadas convolucionais. A camada de saída prevê o valor para as nove ações possíveis a serem tomadas pelo agente.

esse cálculo, ele seleciona a ação que o levará a receber a recompensa máxima e passa para o próximo estado. Para este experimento, foi estabelecido um limite de aconselhamento para uma função de perda superior a 1.2, pois, após uma observação visual, detectamos que o agente RL tende a realizar ações adequadas abaixo desse valor. Adotamos um limite de 5 conselhos que o professor humano pode aplicar por episódio. O Algoritmo 1 representa o pseudocódigo do método implementado.

Como fase experimental, o usuário informa a ação sugerida por meio do teclado, inserindo números que correspondem às ações do agente. Este usuário precisa acompanhar o treinamento do agente RL através do monitor, para poder inserir os conselhos sempre que solicitados.

- Mover para a direita = 0
- Mover para baixo = 1
- Escala Maior = 2
- Finar = 3
- Mover para a esquerda = 4
- Mover para cima = 5
- Escala Menor = 6
- Alargar = 7
- Encerrar = 8

Algorithm 1 Algoritmo para aconselhamento

Require: Imagem
Ensure: anotação em caixa delimitadora
for cada episódio **do**
budget = 5;
for cada estado **do**
calcula incerteza;
if incerteza >= 1.2 **then**
if budget > 0 **then**
agente recebe aconselhamento humano
budget = budget - 1
else
O agente realiza uma ação gerada pela própria política.
end if
else
O agente realiza uma ação gerada pela própria política.
end if
end for
end for

4.3 Experimentos

Conforme sugerido por Poole e Mackworth [74], uma forma de medir o desempenho de um agente é analisando a recompensa acumulada por episódio. À medida que o agente RL aprende a executar as ações corretamente, ele recebe recompensas maiores.

Avaliamos também quantitativamente o desempenho do agente através do número de anotações que ele conseguiu fazer com e sem ajuda humana. Além disso, adotamos a métrica *Intersection Over Union (IoU)* para avaliar a criação da caixa delimitadora criada pelo agente RL.

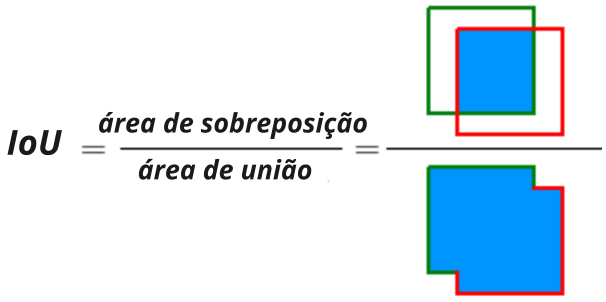

Figura 4.4: A imagem ilustra uma caixa delimitadora de verdade (em verde) e uma caixa delimitadora gerada por um modelo (em vermelho). Fonte [68]

Através de um limiar t, o IoU permite classificar se a localização do objeto está correta ($IoU >= t$) ou incorreta ($IoU < t$).

Durante todos os experimentos, os aconselhamentos foram emitidos ao agente RL por um único ser humano.

Após implementar nosso algoritmo e as definições das métricas de avaliação, aplicamos nossa abordagem em dois casos de uso diferentes.

4.3.1 Caso de uso 1: Banco de dados de exames de tórax

Iniciamos analisando a atuação do agente com e sem aconselhamento em um banco de dados com exames médicos de radiografia de tórax para detecção de cardiomegalia [2]. Para isso, usamos o banco de dados de radiografia de tórax do NIH [55]. Cardiomegalia refere-se a uma condição cardíaca aumentada. É uma das doenças hereditárias mais comuns entre as doenças cardiovasculares, com uma prevalência de pelo menos 1 em 500 na população geral [87].

Os exames de radiografia de tórax são frequentes e econômicos. No entanto, o diagnóstico clínico de uma radiografia de tórax pode ser desafiador e, às vezes mais complexo do que o diagnóstico por tomografia computadorizada de tórax. A falta de grandes conjuntos de dados disponíveis publicamente com anotações significativas é um desafio, atrasando a detecção e o diagnóstico de exames de radiografia de tórax.

4.3.2 Caso de uso 2: banco de dados de exames de mamografia

Um segundo caso de uso em que testamos nossa abordagem foi em casos de imagens de mamografia. O câncer de mama pode ser considerado um dos problemas de saúde globais mais comuns e é considerado a segunda causa de mortalidade por câncer em mulheres [37] [53].

As imagens da mama são adquiridas através de um exame de raio-x. Duas projeções são feitas durante o procedimento de exame: os planos Cranial Caudal (CC) e Médio Lateral Oblico (MLO). Na visão CC, a mama é vista de cima para baixo, enquanto na MLO, a visão é da região lateral.

O mamilo é uma estrutura de interesse a ser observada nos exames de mamografia. Essa estrutura auxilia o mamógrafo a verificar a qualidade do posicionamento de um exame, o que pode minimizar a necessidade do paciente retornar para repetir o exame devido ao mau posicionamento [29]. No entanto, detectar essa estrutura não é trivial, pois, além de ser uma estrutura pequena, nem sempre aparece com clareza nas imagens.

4.4 Resultados

4.4.1 Caso de uso 1: Banco de dados de exames de tórax

Realizamos quatro experimentos de treinamento com o agente RL para analisar seu desempenho na criação de rótulos automaticamente. A descrição dos dados utilizados para o treinamento está destacada na Tabela 4.2. Na coluna de **Ajuda Humana** é indicado qual experimento o agente RL teve auxílio humano para completar sua tarefa. O total de imagens anotadas utilizadas para treinamento como verdade absoluta é apresentada na coluna **Imagens**. A coluna **Pré-treinado** indica se o modelo salvo do agente RL treinado anteriormente está sendo reutilizado para uma nova rodada de treinamento. Portanto, o modelo salvo durante o primeiro experimento (exp1) é utilizado na execução do terceiro experimento (exp3). Já o modelo salvo durante o segundo experimento (exp2) é utilizado na execução do quarto experimento. A quantidade de anotações que o agente RL conseguiu realizar durante o treinamento é indicada pela coluna **Anotações**. Todas as imagens utilizadas para treinamento quanto para teste, possuem cardiomegalia.

Tabela 4.2: Dados de treinamento de cardiomegalia

Experimentos	**Ajuda Humana**	**Imagens**	**Pré-treinado**	**Anotações**
exp1	Não	31	Não	11
exp2	Sim	31	Não	17
exp3	Não	31	Sim	17
exp4	**Sim**	**31**	**Sim**	**19**

A Tabela 4.3 apresenta os resultados obtidos em um conjunto de testes não rotulados.

Tabela 4.3: Dados de teste de cardiomegalia

Experimentos	**Ajuda Humana**	**Imagens**	**Pre-treinado**	**Anotações**
exp1	Não	64	Não	25
exp2	**Sim**	**64**	**Não**	**38**
exp3	Não	64	Sim	37
exp4	Sim	64	Sim	32

A Figura 4.5 mostra a evolução do aprendizado do agente RL ao criar anotações da estrutura da cardiomegalia. Ao longo dos episódios (indicados pelo eixo horizontal), é mostrado o acúmulo de recompensas (eixo vertical) que o agente RL obteve. Recompensas negativas significam que o agente RL teve dificuldade em aprender a fazer anotações.

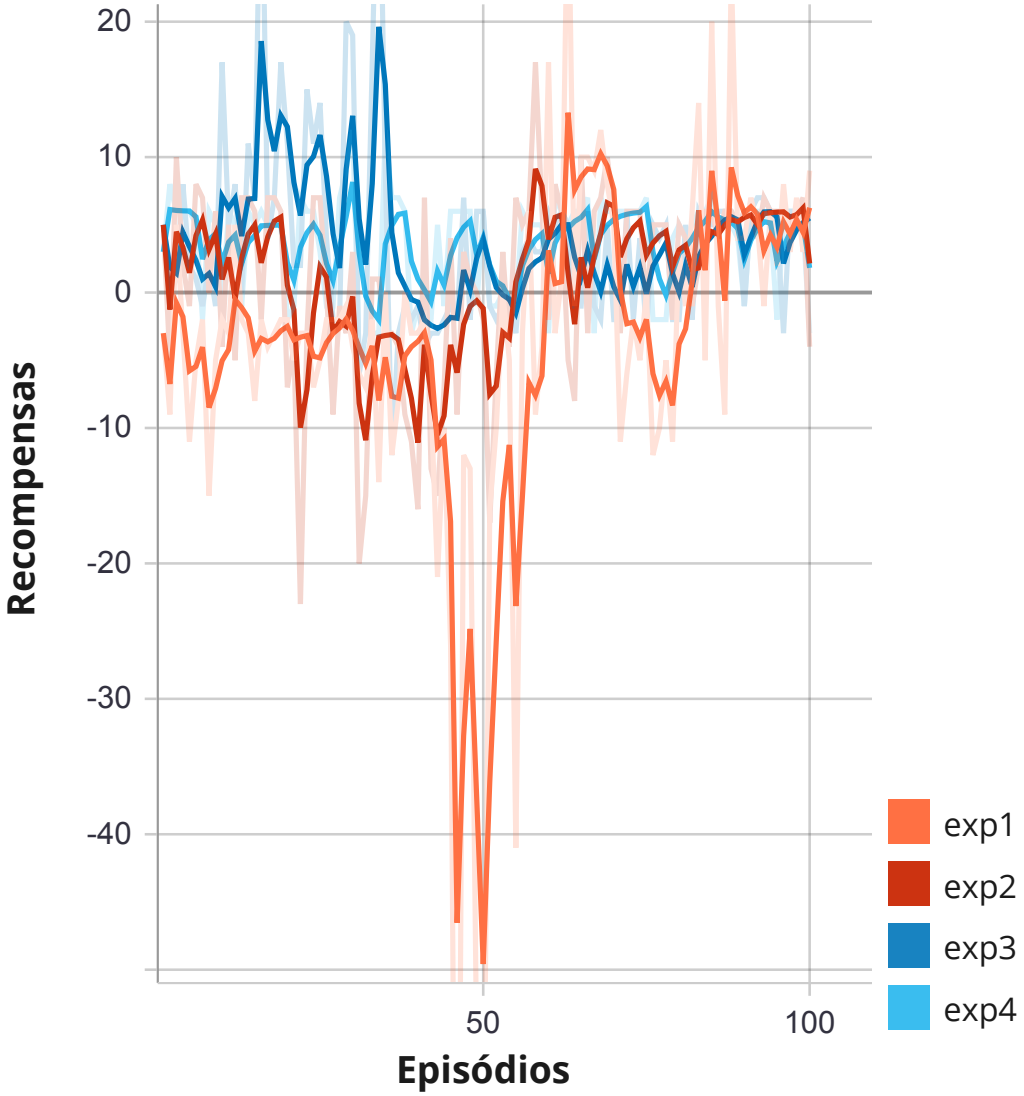

Figura 4.5: Resultado do treinamento do agente RL para detectar a estrutura da cardiomegalia. Cada experimento é representado por uma cor diferente (indicado na legenda da imagem). As curvas dos experimentos são suavizadas produzindo uma curva mais consistente na tendência das recompensas. O experimento denominado **exp4** que utiliza ajuda humana e o modelo pré-treinado, apresentou o melhor resultado com o maior acúmulo de recompensas positivas.

Conforme mostrado na Figura 4.5, e na Tabela 4.3, no **exp1** o agente RL teve dificuldade em criar anotações de caixa delimitadora. O seu baixo desempenho é notado pelo acúmulo de recompensas negativas adquiridas ao longo dos episódios; com ajuda humana, através do **exp2** o agente conseguiu obter um melhor resultado em relação ao treinamento sem aconselhamento, alcançando um total de 38 criações de caixas delimitadoras; no **exp3** que é a utilização do modelo salvo do exp1 para uma nova rodada de treinamento sem auxílio humano, o agente RL conseguiu obter maiores recompensas positivas comparado com o exp1, alcançando um total de 37 criações de caixas delimitadoras; o **exp4** apesar de possuir sutilmente o melhor resultado durante o treinamento, ao ser utilizado sobre o conjunto de testes, alcançou um resultado abaixo dos demais experimentos.

A Figura 4.6 ilustra o resultado obtido pelo agente RL ao criar uma anotação em forma de caixa delimitadora. O modelo utilizado foi o que apresentou o melhor resultado, gerado no segundo experimento.

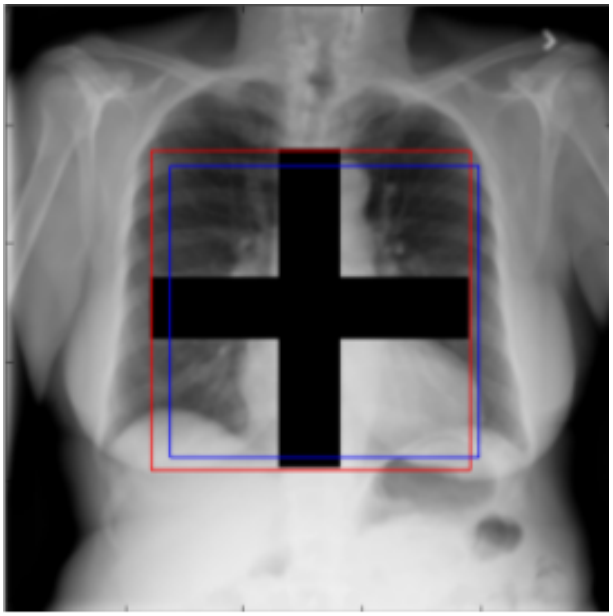

Figura 4.6: Imagem de cardiomegalia sendo detectada. Em azul a verdade absoluta e em vermelho a caixa delimitadora gerada pelo agente. A cruz na imagem é para indicar essa é a caixa final especificada pelo agente RL.

4.4.2 Caso de uso 2: banco de dados de exames de mamografia

Da mesma forma, para este caso de uso, realizamos quatro experimentos de treinamento. A descrição dos dados utilizados nesse caso de uso são semelhantes aos utilizados no caso anterior e pode ser vista na Tabela 4.4. O agente RL foi treinado para criar automaticamente novas anotações do mamilo

a partir de exames projetados no plano CC (Crânio Caudal). Todas as imagens utilizadas para treinamento quanto para teste, possuem a região do mamilo.

Tabela 4.4: Dados de treinamento do mamilo

Experimentos	**Ajuda Humana**	**Imagens**	**Pre-treinado**	**Anotações**
exp1	Não	31	Não	0
exp2	Sim	31	Não	5
exp3	Não	31	Sim	1
exp4	**Sim**	**31**	**Sim**	**15**

Após o treinamento do agente, analisamos a performance do agente RL em um conjunto de teste com imagens não anotadas. A Tabela 4.5 apresenta os resultados obtidos.

Tabela 4.5: Dados de teste do mamilo

Experimentos	**Ajuda Humana**	**Imagens**	**Pre-treinado**	**Anotações**
exp1	Não	192	Não	0
exp2	Sim	192	Não	34
exp3	Não	192	Sim	6
exp4	**Sim**	**192**	**Sim**	**60**

A Figura 4.7 mostra a evolução do aprendizado do agente RL ao criar anotações de uma região de interesse para a mama. Ao longo dos episódios (indicados pelo eixo horizontal), é mostrado o acúmulo de recompensas (eixo vertical) que o agente de RL obteve. Recompensas negativas significam que o agente RL teve dificuldade em aprender a fazer anotações. Como mostra o gráfico, o experimento que apresentou as melhores recompensas, ou seja, o agente obteve recompensas positivas, foi o experimento 4 através do aprendizado com auxílio humano.

Conforme mostrado na Figura 4.7 e na Tabela 4.5, no **exp1** o agente RL teve dificuldades em criar anotações de caixa delimitadora. Como a estrutura a ser localizada possui uma região menor, um maior "esforço" é realizado pelo agente. O seu baixo desempenho é notado pelo acúmulo de recompensas negativas adquiridas ao longo dos episódios e por não ter conseguido criar nenhuma caixa delimitadora. Mesmo com a ajuda humana, através do **exp2** o agente não obteve um comportamento de acumular recompensas positivas, pelo contrário, ao decorrer dos episódios permaneceu acumulando recompensas negativas, conseguindo alcançar algumas recompensas positivas próximo ao fim do experimento. No entanto, conseguiu absorver conhecimento e criou algumas caixas delimitadoras. O **exp3**, que é a utilização do modelo salvo do exp1 para

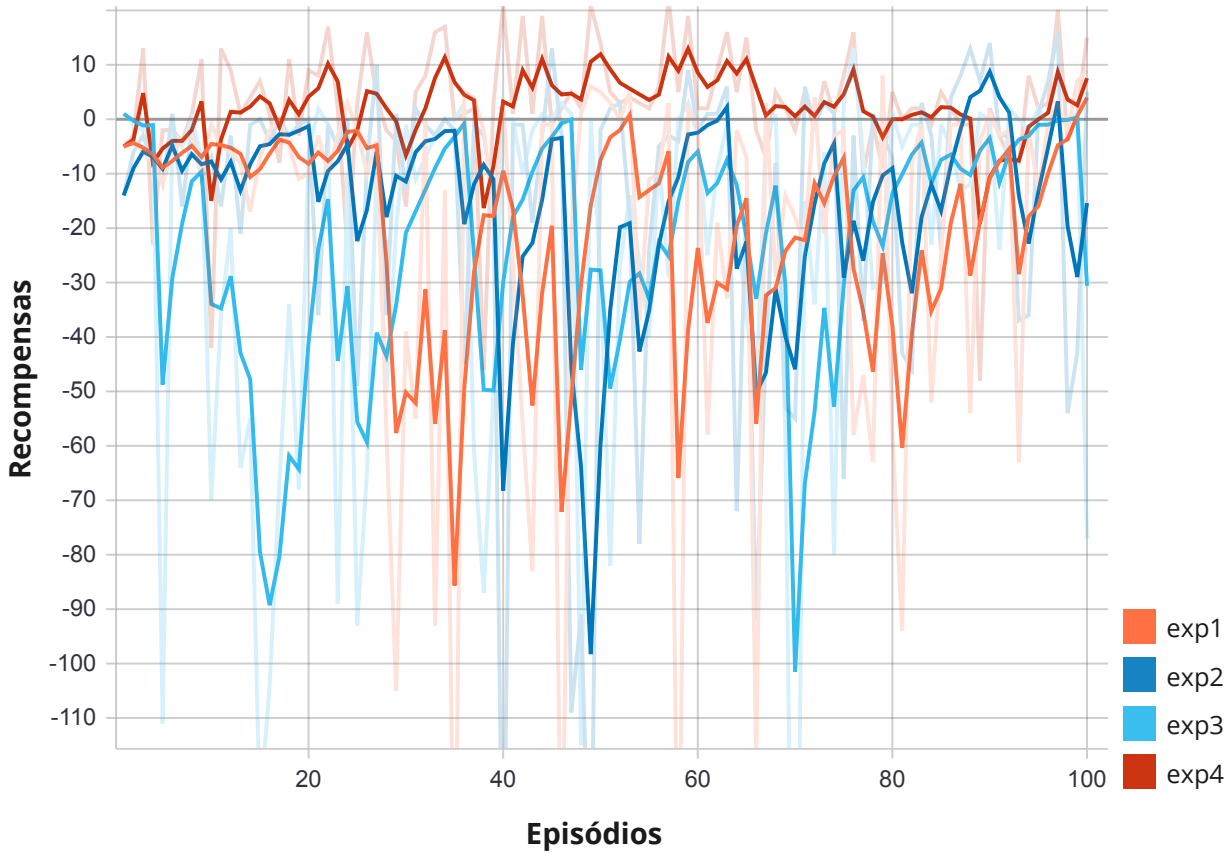

Figura 4.7: Resultado do treinamento do agente RL para detectar a estrutura da papila. Cada experimento é representado por uma cor diferente (como indicado na legenda da imagem). As curvas dos experimentos são suavizadas produzindo uma curva mais consistente na tendência das recompensas. O experimento denominado **exp4** que utiliza ajuda humana e o modelo pré-treinado, apresentou o melhor resultado, com o maior acúmulo de recompensas positivas.

uma nova rodada de treinamento sem auxílio humano, manteve o resultado insatisfatório em seu desempenho, acumulando recompensas negativas durante toda o experimento. No **exp4**, que é a utilização do modelo salvo do exp2 para uma nova rodada de treinamento com auxílio humano, o agente RL conseguiu obter maiores recompensas positivas ao decorrer dos episódios, acumulando na maior parte, valores acima de 0 e alcançando um total de 60 criações de caixas delimitadoras.

A Figura 4.8 ilustra o resultado obtido pelo agente RL ao criar uma anotação na forma de caixa delimitadora da papila. O modelo utilizado foi o que apresentou o melhor resultado, gerado no quarto experimento.

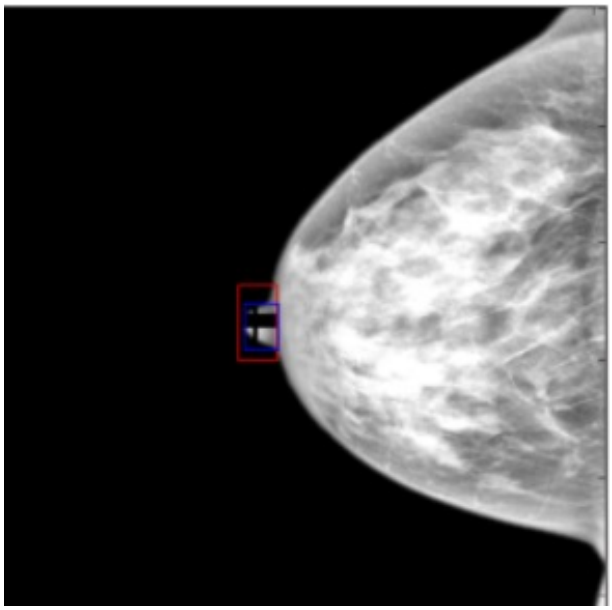

Figura 4.8: Imagem do mamilo sendo detectado. Em azul a verdade absoluta e em vermelho a caixa delimitadora gerada pelo agente.

4.5 Conclusão

Este capítulo apresentou uma nova abordagem para geração de novas anotações em um conjunto de dados de imagens, onde as anotações são realizadas por um agente virtual através de aconselhamento humano. Avaliamos nossa abordagem em conjuntos de dados médicos para radiografia de tórax e mamografia. Os resultados mostraram que o aconselhamento humano permitiu que o agente RL realizasse o aprendizado mesmo com uma pequena amostra de dados anotados. Os resultados também mostraram que o auxílio humano se torna necessário quando a tarefa do agente RL exige mais esforços de sua parte. No caso de mamografia, onde a região de interesse é o mamilo e sua estrutura é pequena, a ajuda humana impactou positivamente no aprendizado do agente RL, aumentando sua precisão ao criar anotações na forma de caixa delimitadora.

Como próximo passo, melhoraremos o método de aconselhamento, buscando uma maior influência no aprendizado do agente virtual. Além disso, será implementada uma abordagem de aprendizado ativo para aumentar a precisão do agente autônomo, aumentando sua capacidade de criar anotações adequadas para algoritmos de aprendizado de máquina supervisionado.

5
Evolução na geração autônoma de anotações a partir de uma metodologia de ensino para um agente RL com Active Learning

Este capítulo apresenta uma evolução da abordagem descrita no capítulo 4. Nesse estudo, uma metodologia de ensino para treinamento de agentes RL é apresentada, capaz de aprender a partir de uma quantidade limitada de dados. Essa proposta possibilita o treinamento desse agente para gerar novas anotações automaticamente do tipo caixa delimitadora (*bounding box*), reduzindo assim os esforços humanos na aquisição de novas anotações. Essa tese tem duas características chaves. A primeira é uma abordagem que permite a inserção humana no processo de treinamento do agente RL, a qual é abordada na Seção 5.1. A segunda característica é apresentada logo em seguida, onde introduzimos uma metodologia de ensino, a qual permitirá uma evolução de aprendizado como um "aluno", gradualmente, sendo o humano o seu "professor". Nessa metodologia, empregamos a utilização da técnica de *Active Learning* para selecionar os dados que contribuirão positivamente no aprendizado. Mais detalhes dessa aplicação são abordados na Seção 5.2. A combinação dessas duas características (Seção 5.1 e 5.2) é que leva ao êxito do processo de ensino. A abordagem aqui desenvolvida foi avaliada em um conjunto de dados de diferentes casos de uso, onde o agente RL foi ensinado a criar anotações na forma de caixas delimitadoras a partir de dados não anotados, detalhes na Seção 5.3. Na Seção 5.4, são apresentados os resultados encontrados durante os experimentos. Por fim, as conclusões são apresentadas na Seção 5.5.

Todos os conjuntos de dados utilizados para esse trabalho já possuem anotações de todas as imagens. No entanto, as informações de anotações são utilizadas apenas na etapa de treinamento do agente RL. Portanto, no decorrer desse capítulo, ao ser mencionada a utilização de um conjunto sem anotações ou não anotados, o que estamos fazendo é não trabalhar com as informações de anotações.

5.1 Método de aconselhamento: TLM (Try a Little More)

Nesta seção abordamos a criação de um método de aconselhamento capaz de contribuir no aprendizado de um agente RL.

5.1.1 Visão Geral

Existem diversas estratégias que podem ser utilizadas com a inclusão do humano no processo de aprendizagem de um sistema de aprendizado por reforço. Uma estratégia que pode influenciar no processo é o aconselhamento. Aconselhar um agente RL em seu processo de treinamento implica em um ensinamento através de interações que se assemelham aos ensinamentos mais convencionais entre o ser humano.

Em um ensino convencional do mundo real há diversos métodos pedagógicos que um professor pode adotar para transmitir os conhecimentos para seus respectivos alunos, tais como:

- **Tradicional:** Um dos métodos mais convencionais adotados. Nesse, o ensino é centrado no transmissor do conhecimento, que no caso é o professor.
- **Montessoriano [13]:** Esse método estimula o aprendiz a ter uma autoformação do conhecimento, sendo o professor, um facilitador desse processo.
- **Waldorf [75]:** Esse método introduz um processo de aprendizado com uma abordagem mais humanista, integrando elementos práticos, conceituais e artísticos.
- **Freiriano [76]:** O educador Paulo Freire criou esse método pedagógico onde o professor deve investigar o universo vocabular do aluno, seus modos de vida e costumes da região e assim trazer temas que fazem parte de sua realidade para evolução de seu aprendizado.
- **Construtivista/interacionista [19]:** Nessa metodologia o aluno possui o papel central na sua própria evolução do aprendizado. Os alunos são autônomos e produzem conhecimento de maneira contextualizada. O professor é considerado um mediador que estimula o estudante a evoluir em seu aprendizado.

  Inspirados na metodologia *interacionista ou construtivista*, desenvolvemos um método de aconselhamento que coloca o ser humano como um mediador do aprendizado de um agente RL. Denominamos nossa abordagem de *"Try a Little More" (TLM).*

O TLM consiste em um humano (professor) como um mediador que estimula o agente aprendiz a evoluir em seu aprendizado até alcançar o objetivo pretendido. No TLM, quando o agente RL possui uma dúvida sobre qual ação tomar em um determinado estado, o humano emite um conselho sobre a ação que deve ser aplicada. Essa dúvida do agente-aprendiz é medida pela sua incerteza em qual ação realizar naquele estado. Além disso, entre os conjuntos de ações que o agente RL pode realizar, em um determinado momento ele pode aplicar uma terminal, indicando que alcançou o estado terminal. No entanto, durante a fase de aprendizagem, pode acontecer que o aluno decida encerrar precocemente a tentativa de localizar um objeto de interesse, ocasionando recompensas negativas e impactando em seu aprendizado. Com o TLM, quando o aprendiz decidir terminar de explorar o ambiente, seja porque encontrou seu objetivo ou por não ter conseguido, o humano professor poderá escolher se aconselhará o agente RL aprendiz com uma nova ação, fazendo assim com que ele explore um pouco mais o ambiente de modo a encontrar o estado final corretamente ou se de fato é o estado final confirmando o sucesso da tarefa. A Figura 5.1 apresenta um diagrama com a visão geral do método.

5.1.2 Implementação do TLM

O humano aconselha o agente RL estudante quando este possui uma incerteza sobre qual ação realizar. A incerteza nesse momento é adotada a partir da verificação da função de perda do próprio algoritmo DQN. Portanto, em cada etapa o agente RL observa o estado atual (região de uma imagem) e estima as recompensas potenciais com base no custo de realizar diferentes ações naquele estado. Após esse cálculo, ele seleciona a ação que o levará a receber a recompensa máxima e passa para o próximo estado. Se tiver uma incerteza superior a 1.2, o agente RL solicita auxilio ao professor humano. Esse processo é repetido até atingir o término do episódio.

Com o método TLM o objetivo é permitir que o agente RL consiga ter uma autonomia em seu aprendizado. Sendo assim, inserimos um limite de quantas vezes o humano poderá influenciar nesse processo. A cada episódio o agente aprendiz pode perguntar até 3 vezes para o humano professor se é de fato o momento certo de finalizar sua exploração. Para a implementação do método de aconselhamento, um limite de 5 conselhos por episódio é permitido pelo humano professor. O Algoritmo 2 representa o pseudocódigo do método implementado.

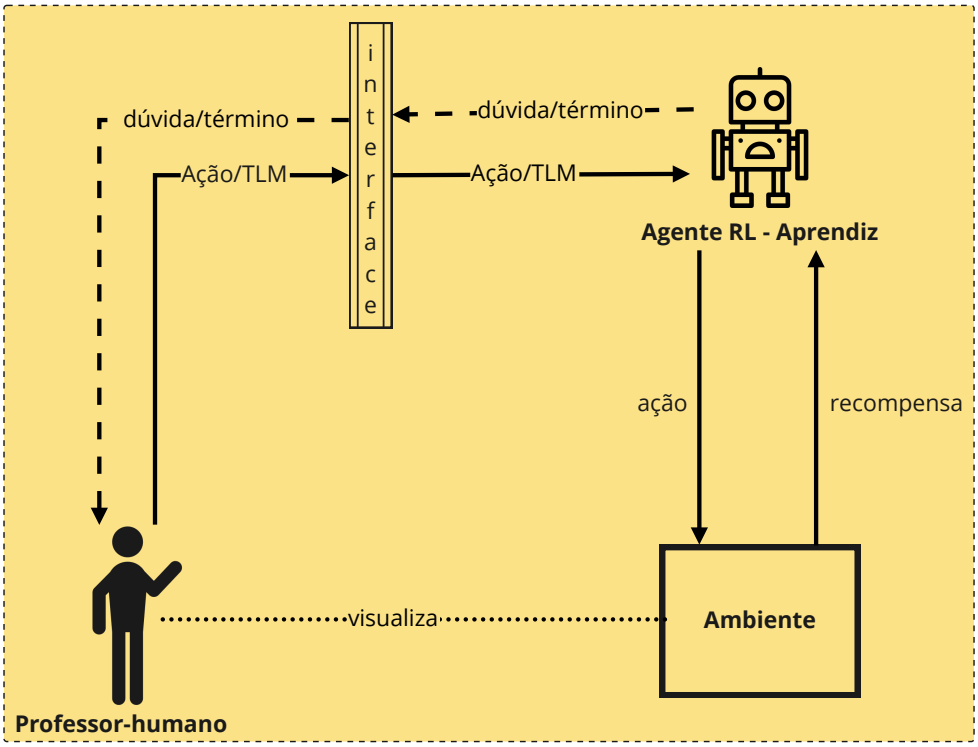

Figura 5.1: Diagrama da abordagem de aconselhamento com o método TLM. Quando o agente aprendiz está com dúvida, ele solicita ao professor humano um conselho sobre qual ação realizar em um determinado estado. O professor humano, através de uma interface, é capaz de inserir uma ação dentre as quais o agente RL consegue interpretar e assim aplicá-la no ambiente. Quando o aprendiz decidir terminar de explorar o ambiente, seja porque encontrou seu objetivo ou por não ter conseguido, o professor humano poderá visualizar a ação aplicada do agente RL no ambiente e assim decidir se de fato ele pode terminar ou se ele deve tentar um pouco mais.

5.2 Metodologia para ensino de um agente RL

Nesta seção é proposta uma metodologia para ensinar um agente RL a aprender realizar anotações autônomas a partir de auxílio humano (conforme método da Seção 5.1) na etapa de aprendizagem.

5.2.1 Visão Geral

O aconselhamento humano é uma poderosa ferramenta para auxiliar na evolução do aprendizado de um agente RL. Com base nesta informação, propomos uma metodologia que permita a inclusão humana no ciclo do aprendizado, muito além de apenas aconselhamentos, mas trazendo um passo a passo de ensino, como uma abordagem pedagógica de ensino entre professor e aluno.

Algorithm 2 Algoritmo para o método "Try a Little More"(TLM)

```
Require: Imagem
Ensure: Novas anotações
    for cada episódio do
 2:   orçamento = 5;
      TLM = 0
 4:   for cada estado S do
        Calcula a incerteza;
 6:     if incerteza >= 1.2 then
          if orçamento > 0 then
 8:         Agente recebe aconselhamento humano
            orçamento = orçamento - 1
10:       else
            O agente realiza a ação gerada pela sua política
12:       end if
        else
14:       O agente realiza a ação gerada pela sua política
        end if
16:     if ação == 8 then
          if TLM < 3 then
18:         TLM = TLM + 1
            Agente recebe aconselhamento humano e tenta um pouco mais
20:       end if
        end if
22:   end for
    end for
```

Criamos uma metodologia baseada nessa forma de interação humana em ensinar um aluno. Desse modo, ao fim do processo um agente terá o conhecimento necessário para resolver os problemas de forma autônoma. Nossa metodologia é baseada em 5 passos que se complementam, implicando diretamente na evolução da aprendizagem do agente RL.

O primeiro passo é a aprendizagem inicial do agente. A partir de um conjunto de dados anotados e limitado, o agente RL é treinado com o auxílio humano sobre imagens aleatórias do conjunto de dados. Desse modo, o humano com o papel de professor, pode aconselhar sobre as ações que o agente aprendiz deve realizar sobre um determinado estado. Essa primeira etapa é realizada através do método TLM, como apresentado na Figura 5.1.

No segundo passo, o agente RL já terá passado por uma etapa de aprendizado, portanto terá um "conhecimento" prévio. Ele usará esse conhecimento para realizar uma "avaliação" buscando diagnosticar quais são as imagens que possuem uma maior dificuldade em analisá-las para alcançar o seu objetivo (que em nosso caso é encontrar uma região de interesse e criar um caixa delimitadora) e quais são as imagens que possuem mais facilidade em completar

sua tarefa. Essa verificação é feita sob um conjunto de dados não anotados. Nesse ponto é adotada a técnica de aprendizado ativo.

Após o agente RL passar por essa avaliação e encontrar as imagens mais informativas (capacidade de uma instância em reduzir a incerteza de um modelo [89]), o próximo passo é passar as imagens não anotadas escolhidas pelo agente RL como sendo as mais fáceis para um anotador humano realizar as anotações dessas imagens específicas. Após anotadas, essas imagens são integradas para re-treinamento do agente RL. Assim, o agente RL passa por um novo treinamento com o auxílio humano sobre um conjunto de imagens específicas. Desse modo, o humano com o papel de professor, pode aconselhar para evolução do aprendizado.

Na quarta etapa, após uma nova etapa de aprendizado, o agente passa por uma nova "avaliação" buscando diagnosticar as imagens mais informativas. Buscando um aprendizado progressivo, nessa etapa o agente seleciona o conjunto de imagens que possui mais incerteza, ou seja, as imagens mais difíceis. Com isso, o terceiro passo de nossa metodologia é repetido.

No quinto e último passo, o agente já passou por um aprendizado evolutivo, aprendendo a realizar anotações inicialmente com imagens fáceis e depois com as imagens mais difíceis. Portanto, nesse momento se torna capaz de realizar anotações em um conjunto de dados de maneira autônoma. A Figura 5.2 apresenta o pipeline para treinamento de um agente RL através da metodologia proposta.

5.2.2 Implementação de Aprendizado Ativo

O TLM consiste em utilizar o humano como professor, de modo a ensinar o agente RL (aprendiz) a continuar sua tarefa de modo a alcançar o objetivo pretendido. No entanto, para a completude da metodologia proposta, combinamos AL e RL para completar a estratégia de seleção das imagens mais fáceis e mais difíceis e assim reduzir a quantidade de esforço humano em realizar anotações.

Adotamos o AL baseado em pool, onde o método de aprendizagem acessa um conjunto (pools) de instâncias não anotadas e, em seguida, usa um critério específico para comparar e consultar as instâncias, independentemente de sua ordem individual.

O ciclo da abordagem de aprendizado ativo baseado em pools consiste em 4 passos: *treinamento do modelo, verificação por predição, seleção de pool e anotação* , como mostra a Figura 5.3.

O **treinamento do modelo** é a primeira etapa desse ciclo. Esse treina-

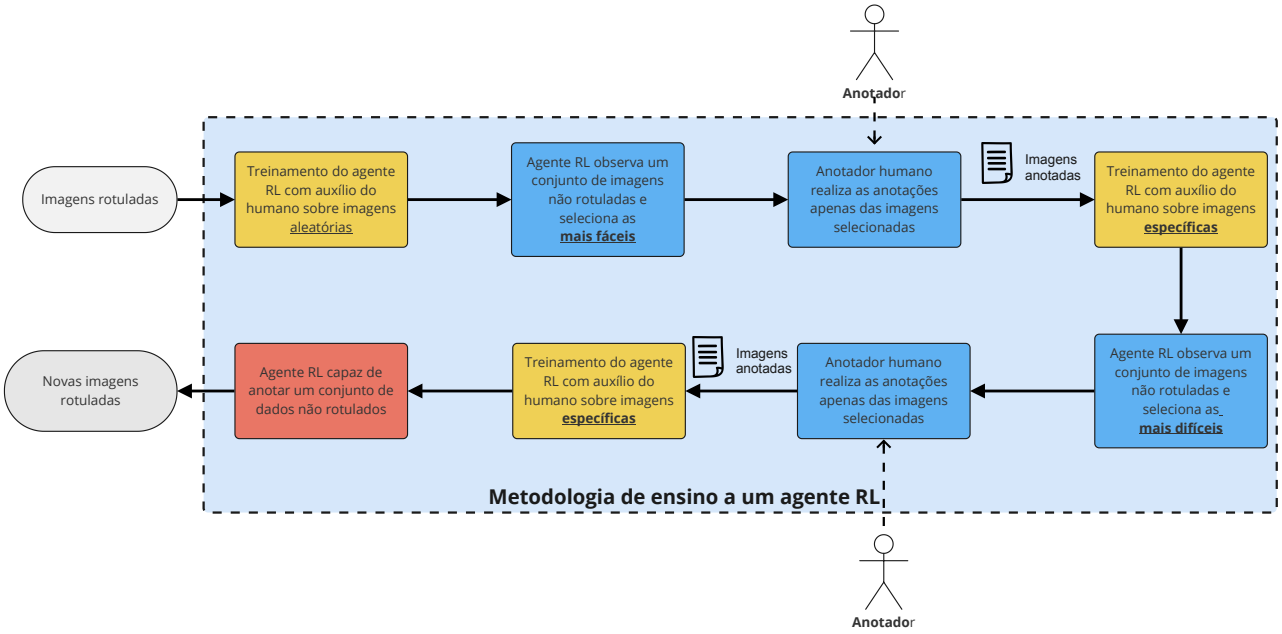

Figura 5.2: Pipeline para treinamento de um agente RL através da metodologia proposta. O agente RL adquire um aprendizado progressivo, aprendendo a realizar anotações sobre as imagens mais fáceis e, ao longo do tempo, passando para as imagens mais difíceis. Os componentes em amarelo, representam nosso método de aconselhamento apresentado na Figura 5.1. Os componentes em azul representam o ciclo do AL apresentado na Figura 5.3.

mento consiste em utilizar um conjunto de dados já anotados para treinar um modelo de aprendizado de máquina. Nessa etapa do ciclo, utilizamos o nosso modelo de aprendizado por reforço, com a abordagem TLM (Figura 5.1). Dado um conjunto de imagens limitadas com anotações, o treinamento inicial do agente RL é realizado com o auxílio humano e assim obtém um aprendizado inicial para realizar as predições. O Algoritmo 2 apresenta a implementação para essa etapa.

Após o treinamento do modelo, a próxima etapa do ciclo de AL é realizar a **predição** sobre os conjuntos de imagens separadas em pool. O objetivo dessa etapa é o modelo detectar as imagens mais informativas, pois se anotadas e utilizadas para treinamento, aumentarão a acurácia desse modelo. Ou seja, o algoritmo de aprendizagem escolhe os dados com os quais deseja aprender. Existem várias abordagens que visam atender a esse aspecto, dentre elas, a mais comum é o cálculo da incerteza. Embora haja diversos métodos para calcular a incerteza sobre um modelo de aprendizado de máquina, aplicar esses métodos em uma abordagem de aprendizado por reforço é bem desafiador.

Os meios de calcular a incerteza mais comuns, seja por margem de confiança, entropia, entre outros, requerem um comparativo entre todas as

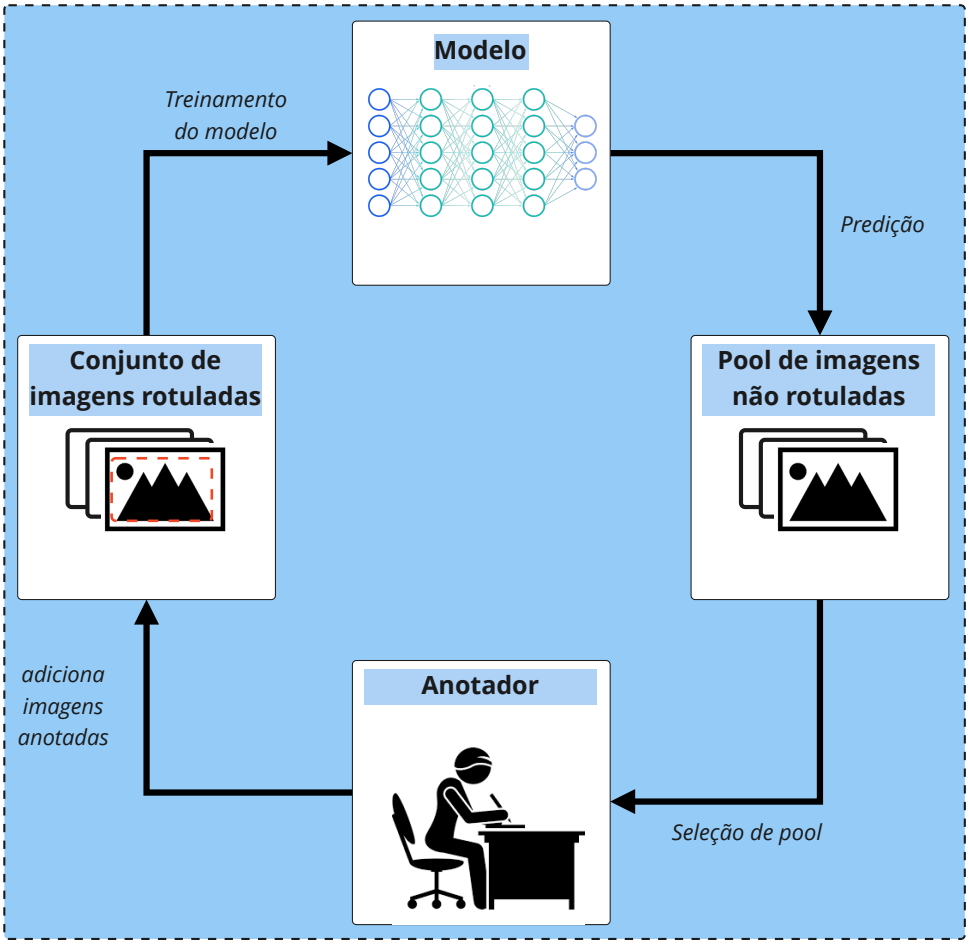

Figura 5.3: O ciclo da abordagem de AL baseada em pools.

predições do modelo. Suponha que escolhemos adotar o cálculo de incerteza através da margem de confiança. Esse cálculo utiliza a diferença entre as duas maiores predições do modelo. Quando se utiliza um modelo convencional de detecção de objetos, a saída do modelo tem a probabilidade para as classes de interesse. Em um modelo de aprendizado por reforço, a saída do modelo é a probabilidade de qual ação deve ser realizada em um determinado estado de um cenário. Essa diferença dificulta aplicar os métodos tradicionais em uma abordagem de RL. A Figura 5.4 apresenta uma comparação entre abordagens tradicionais de aprendizado de máquina e aprendizado por reforço.

A partir dessa limitação, decidimos estimar a incerteza do agente RL adotando a estratégia encontrada nos trabalhos de [15, 65]. Sabendo que a última camada gera uma estimativa do valor esperado para cada ação que o agente RL pode realizar, adicionamos na última camada, *heads*, que realizam a estimativa dos valores esperados para cada ação, separadamente. A Figura 5.5 apresenta a arquitetura de uma DQN regular e outro com a adição de *heads*.

Criamos a mesma quantidade de *heads* conforme a quantidade de ações que o agente pode realizar. Para cada um, uma estimativa diferente dos valores

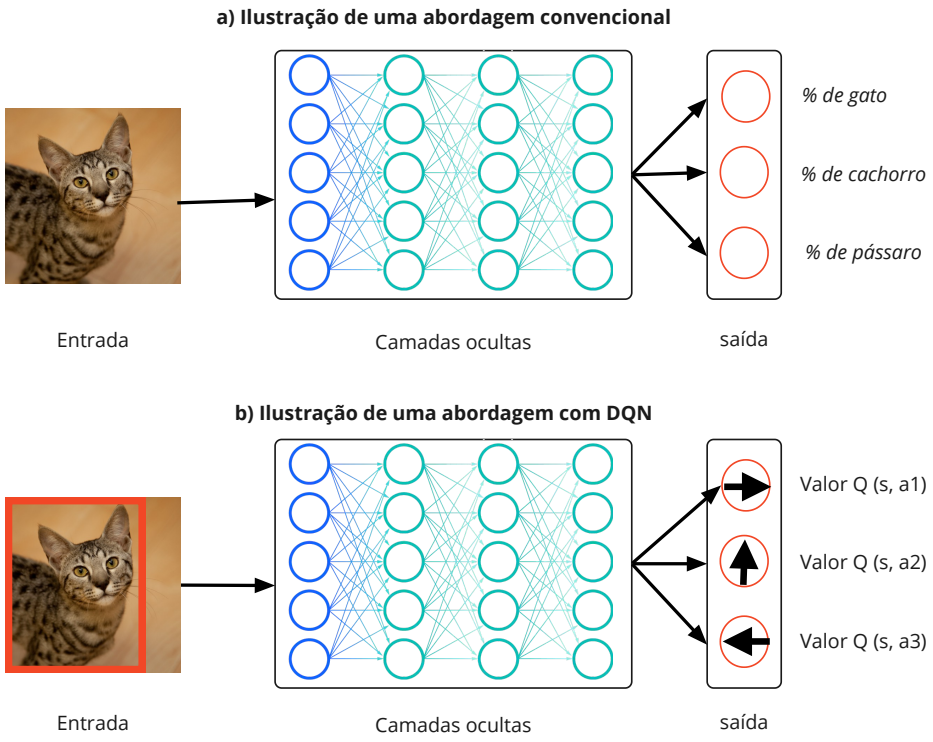

Figura 5.4: Ilustração de uma abordagem tradicional de redes neurais e a com DQN.

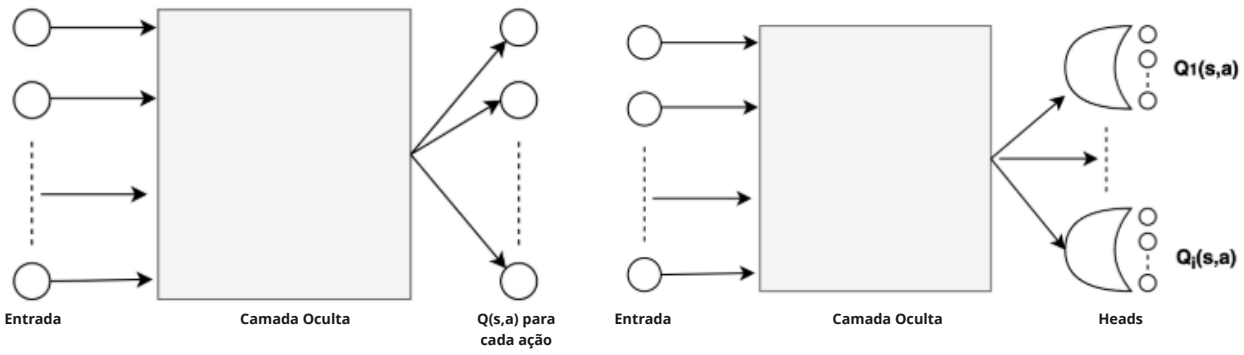

Figura 5.5: Arquitetura de uma DQN regular e outro com a adição de *heads*. Imagem adaptada [15]

de ação são gerados. Desse modo, ao longo da evolução do aprendizado do algoritmo as variâncias das predições são reduzidas. Logo, para estimar a incerteza do agente RL, utilizamos a variância das previsões dos heads. A Equação 5-1 mostra o cálculo para essa estimativa.

$$\mu(s) = \frac{\sum_{\forall a \epsilon A} var(Q(s,a))}{|A|} \tag{5-1}$$

Onde s representa o estado atual, a representa uma ação e A o conjunto total das possíveis ações, var é a variância. $Q(s,a)$ é o valor obtido nas funções Q em cada *head*, $Q(s,a) = \begin{bmatrix} Q_1(s,a) \\ ... \\ Q_h(s,a) \end{bmatrix}$

Cada *head* possui sua própria função de perda, dada pela Equação 5-2.

$$Loss = (r + \gamma max_{a'} Q(s', a'; \Theta') - Q(s, a; \Theta))^2 \tag{5-2}$$

O Algoritmo 3 apresenta a implementação do DQN com heads, bem como a etapa de *seleção dos pools*. O ponto de partida para essa implementação é a criação das pools, ou seja, dado um conjunto de imagens não anotadas $\mathbb{U}$, é preciso dividi-las em sub-conjuntos, $\mathbb{U} = \{u_1, u_2, u_3, ..., u_n\}$. Em cada subconjunto, adotamos utilizar um total de 20 imagens. Como posteriormente dois desses subconjuntos serão passados para o ser humano realizar as anotações das respectivas imagens contidas nele, passar imagens demais influencia diretamente no esforço em realizar essas anotações. Após a separação das pools e o modelo treinado com um conjunto de L imagens anotadas, a predição é realizada sobre cada um dos pools. Logo, sobre cada um será feita a estimativa da incerteza sobre as respectivas imagens contidas nele. Ao percorrer todas as imagens de um pool, é realizada a soma das incertezas obtidas em cada uma das imagens e adicionada em uma lista de incertezas, cuja referência aponta para o seu específico pool. Desse modo, após analisar todos os pools, nesta lista constará as pontuações de quais pools possuem as imagens mais fáceis (onde a incerteza é menor) e quais as imagens mais difíceis (onde a incerteza é maior). Seguindo a nossa metodologia de ensino a um agente RL, primeiro as imagens mais fáceis são selecionadas e então são passadas para um anotador humano realizar as respectivas anotações dessas imagens. Em seguida, essas imagens anotadas são utilizadas para um novo treinamento do agente RL. Com o novo conhecimento adquirido pelo agente, um novo conjunto de pools é gerado e o ciclo descrito acima se repete para seleção do pool contendo as imagens mais difíceis. Desse modo, durante todo o processo, apenas duas pools são escolhidas para serem anotadas pelo humano e utilizadas para treinamento

do agente, um pool contendo as imagens mais fáceis e outro com as mais difíceis. Os componentes em azul apresentados na Figura 5.2, exemplificam essa dinâmica.

Algorithm 3 Algoritmo para o nosso método de active learning

Require: $\mathbb{L}$: um conjunto limitado de imagens anotadas
Require: $\mathbb{U}$: um conjunto de imagens não anotadas divididas em pools
Ensure: Novas anotações
while epoca != 2 **do**
  Treina o agente RL com auxilio humano através do TLM
  Utiliza o agente RL e prediz as amostras $\mathbb{U}$
  for cada $\{u_1, u_2, u_3, ..., u_n\}$ **do**
    for cada imagem **do**
      calcula os pontos de confidência $\mu(s) = \frac{\sum_{\forall a \epsilon A} var(Q(s,a))}{|A|}$
    end for
    τ = soma de todos os pontos de confidencia por pool
    $lista = \tau$
  end for
  if epoca == par **then**
    Seleciona na *lista*, o pool com menor incerteza.
    Realização das anotações do pool selecionado pelo humano.
  else
    seleciona em *lista*, o pool com maior incerteza
    Realização das anotações do pool selecionado pelo humano.
  end if
  $\mathbb{L}$ = novas imagens anotadas
  $\mathbb{U}$ = remove o pool selecionado
end while
Treina o agente RL com auxilio humano através do TLM

A última etapa do ciclo é realizar as anotações das imagens selecionadas e adicioná-las para treinamento. Vale apena destacar aqui que todos os conjuntos de dados utilizados para esse trabalho já possuem anotações de todas as imagens. No entanto, as informações de anotações não são utilizadas durante a predição e criação dos pools. Quando um conjunto de imagens é selecionado para que um anotador humano realize suas anotações, o que fazemos nessa etapa é ativar as informações dos rótulos novamente e adicionar a imagem com seu respectivo rótulo para treinamento do modelo.

5.3 Experimentos

Durante todos os experimentos, os aconselhamentos foram emitidos ao agente RL por um único ser humano. Este usuário precisou acompanhar o

treinamento do agente RL através do monitor, para poder inserir os conselhos sempre que solicitados.

5.3.1 Conjuntos de dados

Foram adotadas duas bases de dados para avaliação experimental do método proposto. A base de dados de mamografia contém imagens cuja estrutura de interesse é pequena, dificultando sua identificação e exigindo uma maior precisão do método. Já a segunda base de dados utilizada, com imagens de aeronaves, é formada por imagens com o objeto de interesse em dimensões maiores, facilitando assim sua identificação. Em ambas as bases de dados, apenas uma região de interesse é processada por imagem.

5.3.1.1 Caso de uso 1: Base de dados com exames de mamografia

O conjunto de dados é formado por 1065 imagens, sendo utilizadas apenas 40 imagens anotadas com caixa delimitadora para treinamento do agente RL. As demais foram utilizadas sem anotações para composição do conjunto de teste e dos pools. Esse conjunto de dados segue a estruturação do PASCAL VOC, onde os arquivos de anotações estão no formato XML.

Assim como no experimento do capítulo anterior, a região de interesse é a estrutura do mamilo. Detectar essa estrutura, como já falado, não é trivial, pois, além de ser uma estrutura pequena, nem sempre aparece com clareza nas imagens.

5.3.1.2 Caso de uso 2: Base de dados com imagens de aeronaves

As imagens de aeronaves utilizadas foram extraídas da base de dados do Pascal VOC 2012 + 2007. Pascal VOC consiste em 500 mil imagens com 20 categorias de objetos [34, 72, 123]. A Figura 5.6 apresenta exemplos de imagens utilizadas.

Para esse experimento, as imagens passaram por uma filtragem, sendo selecionadas apenas aquelas com apenas um objeto em cena. Sendo assim, para esse experimento é utilizado um total de 650 imagens, sendo apenas 40 com anotações na forma de caixa delimitadora e utilizadas para treinamento dos agentes RL. As demais foram utilizadas sem anotações para composição do conjunto de teste e dos pools.

Figura 5.6: Imagens de aviões da base de dados Pascal VOC.

5.3.2 Protocolo de avaliação

Quatro métricas de avaliação são adotadas para avaliar o desempenho da tarefa de anotação de novos dados. A primeira é através da *Intersection Over Union (IoU)*. Mais detalhes sobre essa métrica podem ser encontradas na Seção 4.3. A equação 5-3 apresenta a definição dO IoU.

$$IoU = \frac{TP}{TP + FP + FN} \tag{5-3}$$

O segundo conjunto de medidas de desempenho usado para avaliação da precisão são através de *True Positive (TP), False Positive (FP) e FN (False Negative)*. TP indica o número de imagens corretamente identificadas, ou seja, o IoU é superior ao limite estabelecido. FP indica o número de imagens detectadas incorretamente como positivas, mas não são, ou seja, o IoU é inferior ao limite. FN indica o número de imagens não detectadas. Esses parâmetros permitem calcular a Precisão (dentre todas as classificações de classe Positivo que o modelo fez, quantas estão corretas); Revocação (entre todas as situações de classe Positivo como valor esperado, quantas estão corretas) e F1-Score (média harmônica entre precisão e revocação). As Equações 5-4, 5-5, 5-6 apresentam a definição para essas métricas:

$$Precisão = \frac{TP}{TP + FP} \tag{5-4}$$

$$Revocação = \frac{TP}{TP + FN} \tag{5-5}$$

$$F1Score = 2 * \frac{precisao * revocacao}{precisao + revocacao} \tag{5-6}$$

Outra medida que avaliamos é a frequência de conselhos por ações que o humano enviou para o agente. Para isso, verificamos através de um mapa de interações de ações inseridas.

Por fim, avaliamos o conjunto de dados gerados pela nossa abordagem em um algoritmo de detecção de objetos supervisionado consolidado no estado da arte chamado *You Only Look Once (YOLO)* [79]. Esse algoritmo realiza a detecção de vários objetos em tempo real. A detecção é feita como um problema de regressão e fornece as probabilidades de classe das imagens detectadas. O algoritmo YOLO emprega redes neurais convolucionais (CNN) para detectar objetos em tempo real. A métrica utilizada para essa verificação é a *mean Average Precision (mAP)*. Essa métrica é usada para medir a capacidade do modelo de identificar apenas o objeto de interesse. O resultado varia de 0 a 1, sendo que quanto mais próximo de 1, mais preciso está o modelo em identificar um objeto.

5.3.3 Técnicas comparadas

Para validação da metodologia descrita nessa tese, foram realizados experimentos com algoritmos de aconselhamentos encontrados na literatura e comparados com o proposto.

Com base nos estudos de Caicedo & Lazebnik e de Otoofi [11, 66], foi implementado um algoritmo de DQN sem aconselhamentos para localizar objetos em imagens bidimensionais (2D). Essa implementação é referenciada como **baseline**. A partir desse algoritmo foi realizado um experimento utilizando todas as informações de anotações das imagens, apresentando um resultado do **limite superior**.

Como algoritmo de aconselhamento, foi implementado o algoritmo proposto por Lin et al. [45] e estudado por Frazier S. e Riedl M. [24], denominado **Feedback Arbitration Agent (FAA)**. Esse algoritmo permite ao humano aplicar conselhos de forma intermitente. Durante o treinamento, o agente escolhe uma ação aleatória de acordo com sua política ou opta por consultar o humano. A pontuação de confiança da rede Q é calculada como uma função de custo, de modo que a baixa confiança incorre em um alto custo. A Equação 5-7 é utilizada para calcular o custo de confiança do agente:

$$custoRelativo = \frac{-1}{ln\sqrt{(\frac{min_{a}\epsilon A(s)L_{a})}{L_{m}ax}) - 1}} \tag{5-7}$$

O segundo método de aconselhamento utilizado é o proposto por Samantha Krening [35]. Denominado **Newtonian Action Advice (NAA)**, permite que um agente RL aprenda sua tarefa a partir de conselhos de ação humana. O atrito é um parâmetro importante em NAA, pois garante que, após algum tempo, o agente retomará a política de exploração do algoritmo RL. Uma ação se repetirá até que o "atrito" faça com que o agente retome a exploração normal. Além disso, o algoritmo NAA, permite que o agente siga o mesmo conselho, sempre que o mesmo estado for observado no futuro. Assim, o humano só terá que fornecer conselhos uma vez para uma determinada situação.

5.4 Resultados

Todos os métodos foram treinados a partir do mesmo conjunto de testes e com os mesmos valores de hiperparâmetros, representados na Tabela 5.4

Tabela 5.1: Hiperparâmetros de aprendizagem

Parâmetros	**Valores**
Tamanho do buffer de memória	50000
Número de episódios	5
Fator desconto	0.99
Total de passos	100
Taxa de aprendizagem	0.00025
Começo do épsilon	1.0
Fim do épsilon	0.2
Batch size	20
Otimizador	RMSProp

5.4.1 Caso de uso 1: Exames de mamografia

Após implementação dos algoritmos, foram treinados com uma base dados limitada, sendo 40 imagens no total (exceto para o resultado do limite superior, onde foram utilizadas 585 imagens totalmente anotadas). A nossa abordagem segue uma metodologia original que inclui a seleção de imagens específicas para serem anotadas pelo ser humano e posteriormente utilizadas para treinamento do agente RL. Portanto, inicialmente o agente RL é treinado com as 40 imagens anotadas e ao fim do ciclo da metodologia, outras 40 imagens são selecionadas e anotadas pelo humano (sendo 20 imagens as mais fáceis e outras 20 as mais difíceis). Após treinamento, os algoritmos foram analisados em um conjunto de teste contento 480 imagens não anotadas, de modo a gerar

anotações automáticas da estrutura de interesse. Todas as imagens utilizadas para treinamento quanto para teste, possuem a estrutura do mamilo.

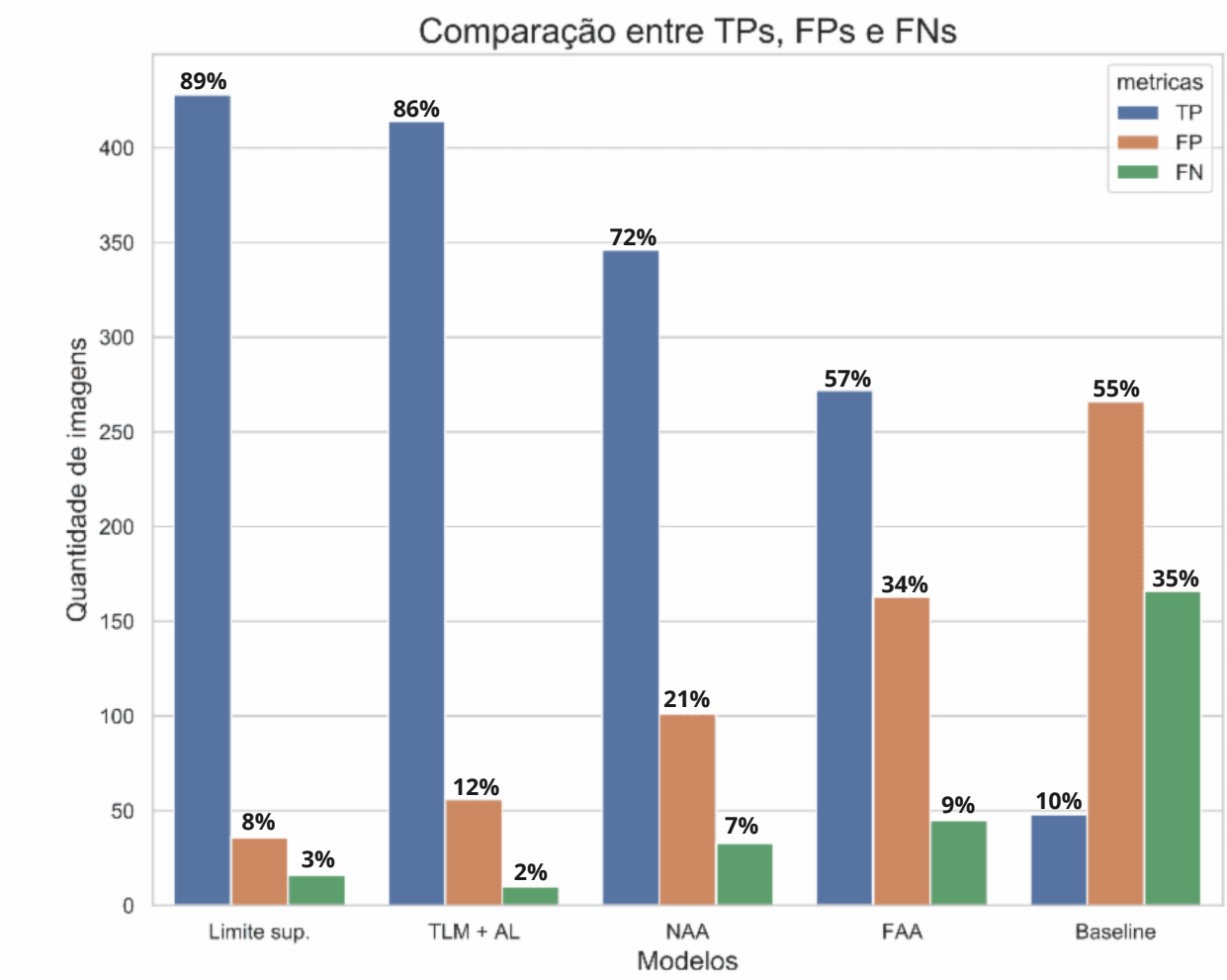

Figura 5.7: Resultado dos modelos sendo aplicados em um conjunto de teste para criação de novas anotações na forma de caixa delimitadora. Embora nosso método tenha ficado um pouco abaixo do limite superior no quesito quantidade de imagens anotadas, entre os métodos de aconselhamento, nossa abordagem apresenta uma considerável vantagem.

Como mostra a Figura 5.7, o limite superior apresenta o melhor resultado, conseguindo gerar 428 novas anotações. No entanto, esse resultado foi obtido a partir de 545 imagens utilizadas para treinamento do modelo. Em contrapartida, entre os métodos de aconselhamentos, nossa proposta é a que apresenta melhores resultados, conseguindo gerar 414 novas anotações a partir de um conjunto de dados com apenas 40 imagens anotadas inicialmente.

Os resultados listados na Tabela 5.2, mostram a comparação entre os modelos, na detecção correta da estrutura (TP), detecções erradas (FP), a precisão, recall e a média F1-score, sobre o conjunto de testes com 480 imagens não anotadas. As linhas com fundo cinza da tabela, destacam as abordagens

de aconselhamento. A nossa abordagem apresenta o melhor resultado entre as abordagens.

Tabela 5.2: Comparação dos resultados da detecção de papila sobre o conjunto de testes.

Método	**TP**	**FP**	**FN**	**Prec.**	**Rec.**	**F1-Score**	**IoU**
Limite sup	**428**	**36**	**16**	**0.92**	**0.96**	**0.94**	**0.89**
TLM + AL	414	56	10	0.88	0.97	0.92	0.86
NAA	346	101	33	0.73	0.89	0.80	0.72
FAA	272	163	45	0.62	0.85	0.72	0.56
Baseline	48	266	166	0.15	0.22	0.18	0.10

A Figura 5.8 mostra o resultado de uma análise da frequência de conselhos emitidos pelo humano durante o treinamento de um agente RL. A partir de um mapa de interação é possível analisar as frequências de intervenções humanas por ações em cada método ao decorrer do treinamento. Olhando como um todo, a nossa abordagem conseguiu alcançar resultados superiores, como apresentado na Tabela 5.2, com uma menor frequência de aconselhamento humano.

Com o foco de gerar novas anotações para que modelos supervisionados possam ser treinados, selecionamos os dados anotados pela nossa abordagem e utilizamos para treinar o algoritmo YOLO. Realizamos dois experimentos, o primeiro foi treinar o modelo com os dados escassos e o segundo a partir dos novos dados gerados.

Tabela 5.3: Avaliação das novas anotações da papila em um modelo supervisionado.

Método	**Num. Imagens**	**mAP**
YOLO	40	0.52
YOLO + TLM	**414**	**0.91**

Conforme é apresentado na Tabela 5.3, o agente RL conseguiu gerar dados que podem contribuir no desempenho do modelo.

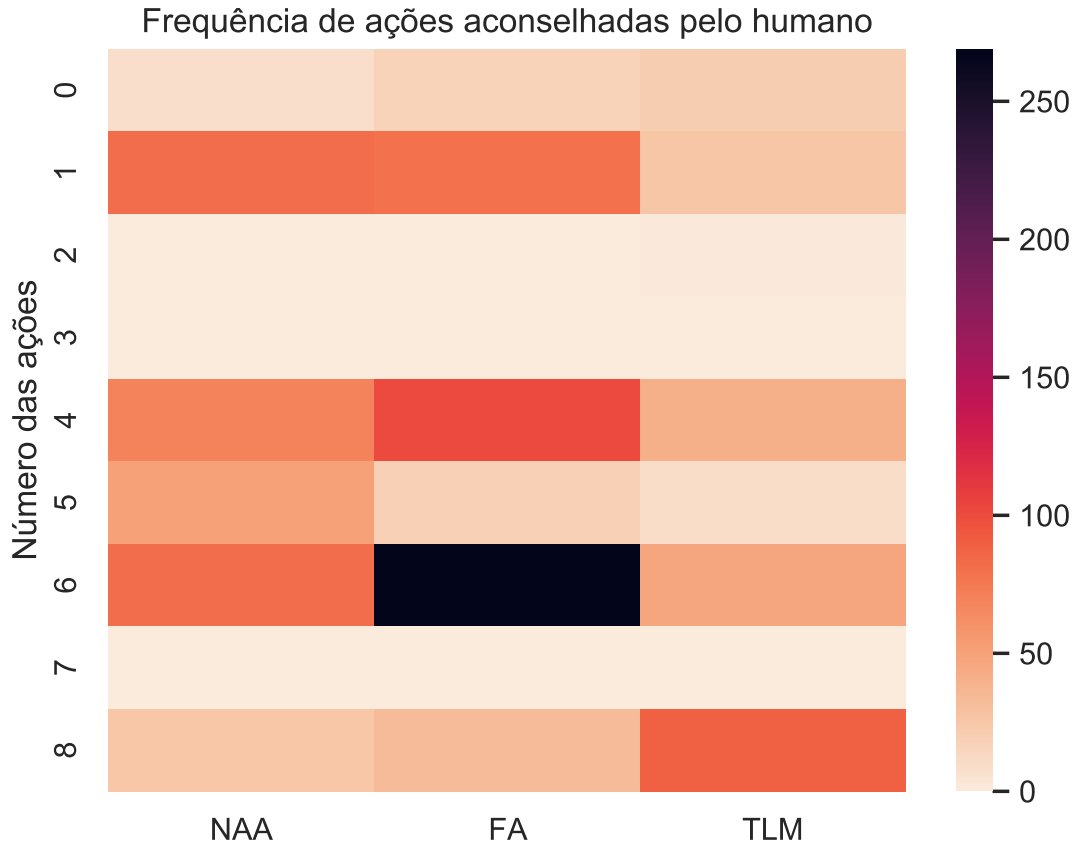

Figura 5.8: Análise da frequência de conselhos emitidas pelo humano durante o treinamento de um agente RL. Uma barra de cores é apresentada ao lado direito da figura, indicando que cores fracas representam baixas interações humanas. Em contrapartida, cores fortes indicam maiores interações humanas. Como não existe uma única ação correta a ser aplicada pelo agente em seu treinamento, é preciso analisar o quadro geral de interações. Nossa abordagem apresentou uma coluna com cores mais claras (baixas interações) ao comparar com os outros métodos.

5.4.2 Caso de uso 2: Imagens de aeronaves

Para o segundo conjunto de imagens, os agentes autônomos foram treinados com uma base dados limitada, sendo 40 imagens no total (exceto para o resultado do limite superior, onde foram utilizadas 330 imagens totalmente anotadas). A nossa abordagem segue uma metodologia original que inclui a seleção de imagens específicas para serem anotadas pelo ser humano e posteriormente utilizadas para treinamento do agente RL. Portanto, inicialmente o agente RL é treinado com as 40 imagens anotadas e ao fim do ciclo da metodologia, outras 40 imagens são selecionadas e anotadas pelo humano (sendo 20 imagens as mais fáceis e outras 20 as mais difíceis) Após treinamento, os algoritmos foram analisados em um conjunto de teste contendo 320 imagens não anotadas, de modo a gerar anotações automáticas de aeronaves. Todas as imagens utilizadas para treinamento quanto para teste, possuem aeronaves.

Como mostra a Figura 5.9, os agentes virtuais treinados pelas diferentes abordagens conseguiram criar aceitáveis quantidades de novas anotações. Isto porque as aeronaves localizadas nas imagens possuem dimensões maiores, implicando diretamente em poucas interações da parte do agente para encontrar o objeto. No entanto, nossa abordagem conseguiu se destacar das demais, apresentando o melhor resultado, gerando 290 novas anotações.

Os resultados listados na Tabela 5.4 mostram a comparação entre os modelos na detecção correta de uma aeronvae (TP), detecções erradas (FP), a precisão, recall e a média F1-score, sobre o conjunto de testes com 320 imagens não anotadas. A nossa abordagem apresenta o melhor resultado comparado entre todos os métodos (incluindo o limite superior). As linhas com fundo cinza da tabela, destacam as abordagens de aconselhamento. A nossa abordagem se destaca entre as demais abordagens.

A Figura 5.10 mostra o resultado de uma análise da frequência de conselhos emitidas pelo humano durante o treinamento de um agente RL. A partir de um mapa de interação é possível analisar as frequências de intervenções humanas por ações em cada método ao decorrer do treinamento. Dentre os métodos avaliados, embora o nosso não possua a menor frequência de interações, a participação humana é apenas para aconselhamento e não para gerar novos rótulos de imagens, portanto, não a torna mais cara e sim essencial para o ensino e aprendizado do agente RL como apresenta na Tabela 5.4.

A partir dos dados anotados pelo agente RL, treinamos o algoritmo YOLO. Realizamos dois experimentos, o primeiro foi treinar o modelo com os dados escassos e o segundo a partir dos novos dados gerados.

Conforme é apresentado na Tabela 5.5, o agente RL conseguiu gerar

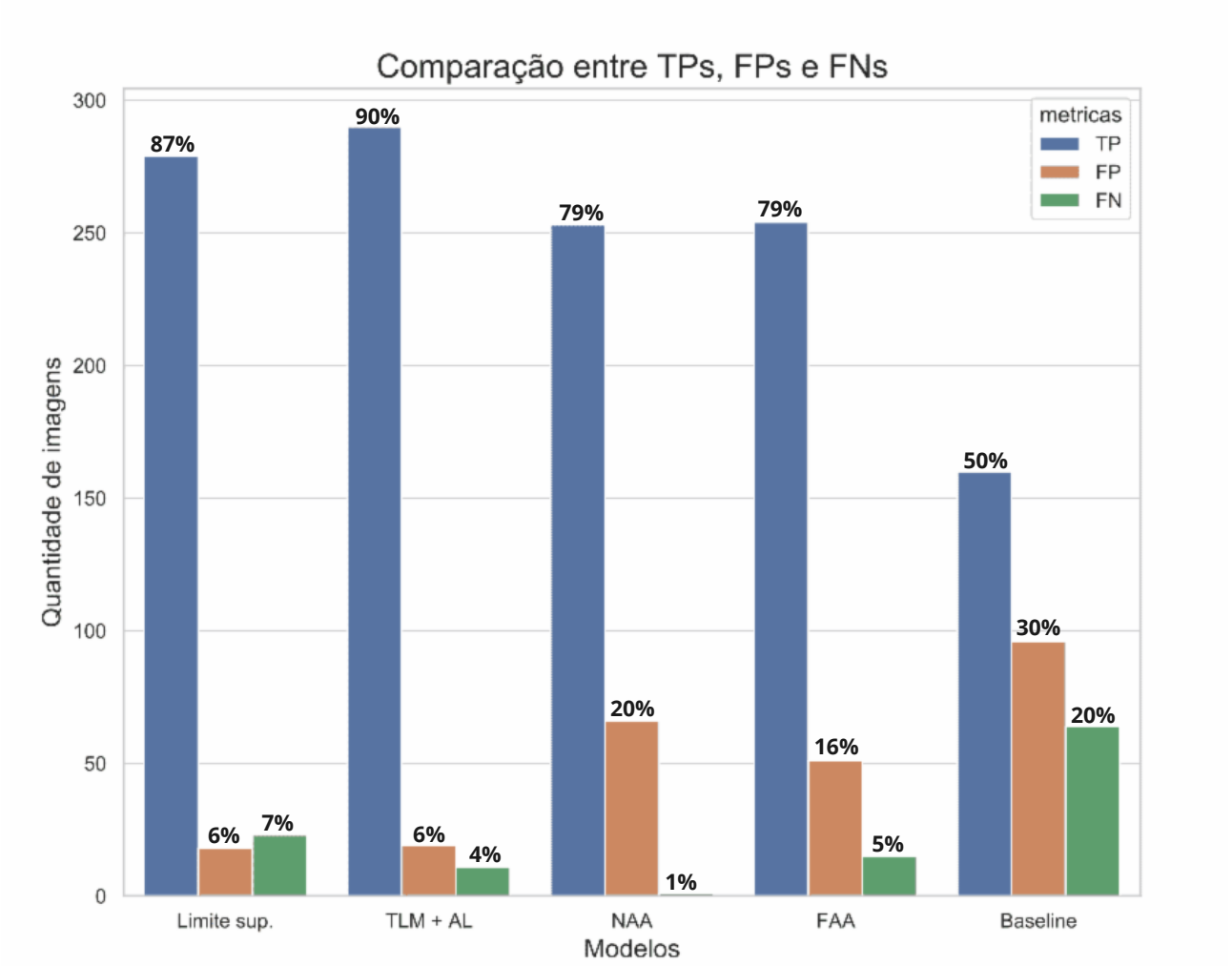

Figura 5.9: Resultado dos modelos sendo aplicados em um conjunto de teste para criação de novas anotações na forma de caixa delimitadora em imagens com aeronaves. Nosso método apresentou o melhor resultado em comparação com limite superior, baseline e os outros métodos de aconselhamento. Além de apresentar o melhor resultado na quantidade de gerar novas imagens, também apresenta um índice baixo de predições erradas (FPs e FNs).

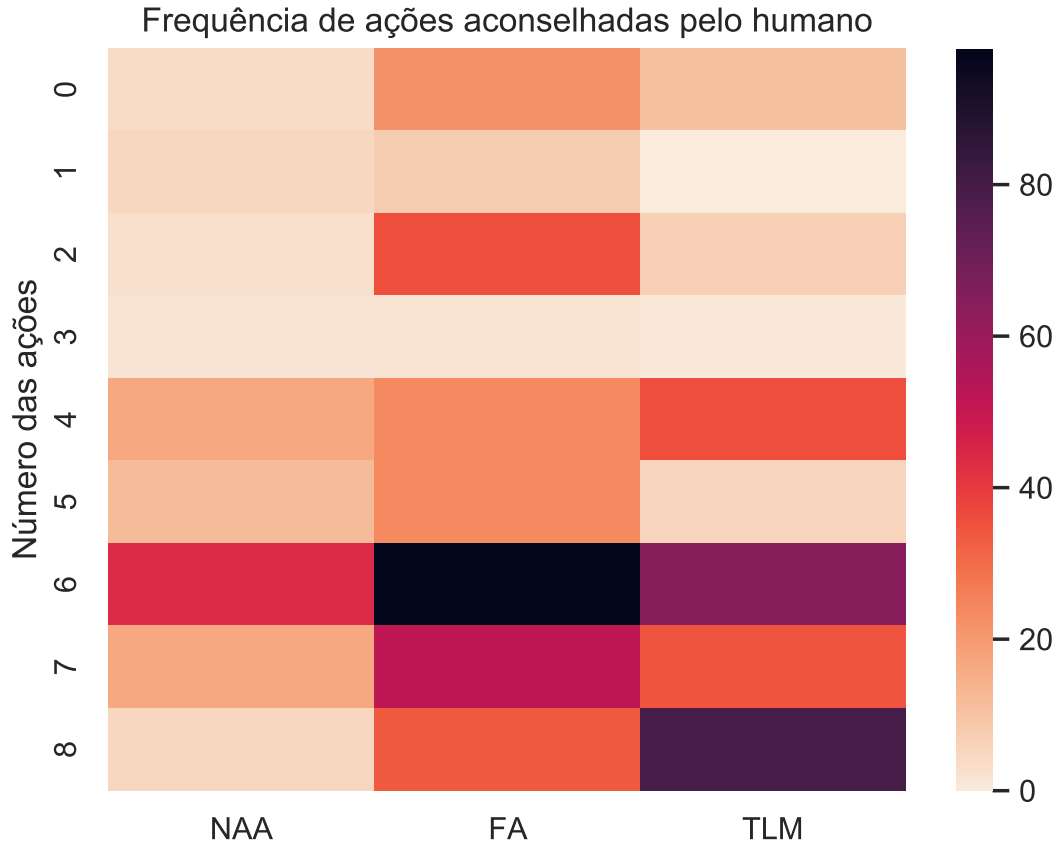

Figura 5.10: Análise da frequência de conselhos emitidas pelo humano durante o treinamento de um agente RL. Uma barra de cores é apresentada ao lado direito da figura, indicando que cores fracas representam baixas interações humanas. Em contrapartida, cores fortes indicam maiores interações humanas. Como não existe uma única ação correta a ser aplicada pelo agente em seu treinamento, é preciso analisar o quadro geral de interações. Embora nosso método apresente uma coluna com cores mais fortes (média e alta interação), a participação humana é apenas para aconselhamento e não para gerar novos rótulos de imagens, portanto, não a torna mais cara e sim essencial para o ensino e aprendizado do agente RL.

Tabela 5.4: Comparação dos resultados na criação de caixas delimitadoras sobre imagens de aeronaves no conjunto de testes.

Método	**TP**	**FP**	**FN**	**Prec.**	**Rec.**	**F1-Score**	**IoU**
Limite sup	279	18	23	0.93	0.92	0.93	0.87
TLM + AL	**290**	**19**	**11**	**0.93**	0.96	**0.95**	**0.90**
NAA	253	66	1	0.80	**0.97**	0.88	0.79
FAA	254	51	15	0.83	0.94	0.88	0.79
Baseline	160	96	64	0.62	0.71	0.66	0.50

Tabela 5.5: Avaliação das novas anotações de aviões em um modelo supervisionado.

Método	**Num. Imagens**	**mAP**
YOLO	40	0.77
YOLO + TLM	**290**	**0.94**

dados que podem contribuir no desempenho do modelo.

5.5 Conclusão

Este capítulo apresentou uma proposta que possibilita o treinamento de um agente RL para gerar novas anotações automaticamente do tipo caixa delimitadora, reduzindo assim os esforços humanos na aquisição de novas anotações. Duas características chaves foram abordadas e implementadas. Uma foi a abordagem de aconselhamento entre humano professor e um agente aprendiz, denominada *Try a Little More (TLM)*. A segunda foi uma metodologia de ensino onde a evolução do aprendizado do agente é gradual, aprendendo com imagens mais fáceis e, conforme for adquirindo conhecimento, aprende com as imagens mais difíceis. A combinação entre essas duas características levaram ao êxito do processo de ensino do agente RL.

Avaliamos nossa abordagem em conjuntos de dados de imagens médicas e imagens de aviões. Comparamos o método proposto com outros trabalhos encontrados na literatura. Nossos experimentos mostraram que, ao empregar nossa metodologia em um conjunto de dados não anotados, nosso agente

consegue gerar novas anotações de forma consistente o suficiente para ser utilizada por modelos supervisionados do estado da arte

6
Conclusões gerais e Trabalhos futuros

Este capítulo apresenta as conclusões gerais desta tese, os principais resultados obtidos nos experimentos e as contribuições. Também são apresentadas as publicações de artigos ao longo da pesquisa, bem como as limitações e os trabalhos futuros.

6.1
Conclusões

Esta tese apresenta uma nova abordagem no campo de data-centric IA. A partir do problema real da falta de dados anotados, investigamos como desenvolver uma abordagem que contribua na criação de novas anotações autônomas e redução dos esforços humanos em gerá-las para o treinamento de modelos supervisionados em detecção de objetos (nossa questão de pesquisa). Um processo para obter novas anotações pode ser uma operação complicada, onde a atenção do anotador e conhecimento impactam diretamente na qualidade das novas anotações.

Partindo da utilização de um algoritmo de aprendizado por reforço profundo, propomos uma metodologia de ensino que envolve interação humano-computador na evolução do aprendizado de um agente autônomo para gerar novas anotações na forma de caixas delimitadoras em conjuntos de dados de imagens. Isso confirma a hipótese apresentada na Seção 1.5

Avaliamos nossa abordagem em conjuntos de dados de imagens médicas e imagens de aviões. Comparamos o método proposto com outros trabalhos encontrados na literatura. Nossos experimentos mostraram que, ao empregar nossa metodologia em um conjunto de dados não anotados, nosso agente consegue gerar novas anotações de forma consistente o suficiente para ser utilizada por modelos supervisionados do estado da arte.

6.2
Contribuições

A principal contribuição deste estudo é a criação de uma metodologia de treinamento para um agente autônomo. Inspirados no método de ensino *interacionista ou construtivista*, desenvolvemos um método de aconselhamento

que coloca o ser humano como um mediador do aprendizado de um agente RL. Denominamos nossa abordagem de *"Try a Little More" (TLM)*. Criamos uma metodologia baseada nessa forma de interação humana em ensinar um aluno. Desse modo, ao fim do processo um agente possui o conhecimento necessário para resolver os problemas de forma autônoma. Nossa metodologia é baseada em 5 passos que se complementam, implicando diretamente na evolução da aprendizagem do agente RL.

6.3 Publicações

A partir dos resultados obtidos ao longo da pesquisa, dois artigos foram publicados:

- Cardia da Cruz, Leonardo, et al. **"A Self-adaptive Serious Game for Eye-Hand Coordination Training."** International Conference on Human-Computer Interaction. Springer, Cham, 2020.
- da Cruz, Leonardo C., et al. **"Enabling Autonomous Medical Image Data Annotation: A human-in-the-loop Reinforcement Learning Approach."** 2021 16th Conference on Computer Science and Intelligence Systems (FedCSIS). IEEE, 2021.

6.4 Limitações e Trabalhos Futuros

Dentre as limitações observadas, menciona-se que o método desenvolvido para aconselhamentos, o TLM, não oferece uma interface visual para o anotador. Quando o agente RL solicita por aconselhamentos, a imagem é salva localmente com a caixa delimitadora que representa o estado observado naquele momento. O humano-professor precisa abrir a imagem para visualizar e analisar qual ação aconselhará o agente. Esses aconselhamentos são enviados a partir do teclado. Estender esse trabalho para um estudo em interação humano-computador e experiência do usuário, pode aprimorar a participação humana com o agente RL.

As imagens a serem anotadas pelo humano e selecionadas através da nossa metodologia necessitam de uma interface externa que geram anotações de caixas verdadeiras no formato XML. É importante ter uma ferramenta acoplada a interface de interação para facilitar a interação.

Apesar de testarmos nossa metodologia em três bases de dados diferentes, é necessário testar e avaliar o agente autônomo na criação de anotações em outros conjuntos de dados para validar sua generalização.

Além disso, o trabalho foi feito com imagens que tivessem apenas um único objeto da classe de interesse. Essa limitação foi para simplificar a implementação dos elementos do framework de aprendizado por reforço e também a arquitetura da rede. Portanto, a quantidade de caixas delimitadoras criadas de forma autônoma pelo nosso agente é restrita a uma única caixa por imagem.

Como trabalhos futuros propõe-se a criação de uma interface visual que permita em tempo real a visualização do deslocamento do agente RL pelo ambiente, facilitando assim na identificação de qual conselho deverá ser enviado naquele momento pelo anotador-professor. Além disso, possibilitar também através dessa interface a inserção dos aconselhamentos.

Como esse trabalho está limitado a criação de uma única caixa delimitadora, estender a implementação para aumentar a quantidade de objetos que ele pode localizar em uma imagem, permitindo a criação de caixas delimitadoras para múltiplas classes, pode tornar nossa abordagem ainda mais prática.

Na versão atual do trabalho, a verificação das anotações geradas pelo agente é através de uma comparação entre as caixas delimitadoras da verdade absoluta. No entanto, ao aplicar essa abordagem para um conjunto de dados sem nenhuma anotação, ou seja, sem anotações de verdade absoluta para comparar, é preciso validar as anotações feitas pelo agente sem ser com o humano olhando cada caso. Implementar uma solução para esse problema, aumentará a aplicabilidade do trabalho em casos reais.

Além disso, por apresentar uma contribuição voltada para criação de uma metodologia de ensino para um agente autônomo, os experimentos aplicados nesta tese foram realizados e comparados com outros métodos da literatura voltados para a técnica de aconselhamentos e algoritmos de aprendizagem por reforço. No entanto, é importante realizar experimentos e comparativos com outros métodos voltados para geração de anotações, tais como: Aumento de Dados (*Data Augmentation*) [94], Aprendizado Supervisionado Híbrido [69], Aprendizado Ativo (AL) [64], entre outros. Assim, a investigação desta abordagem sobre outras abordagens de redução nos esforços de realizar novas anotações pode apresentar outros resultados relevantes.

Algoritmos de DRL possuem diversos hiperparâmetros que auxiliam no treinamento do modelo. Encontrar uma melhor combinação desses valores pode aumentar o desempenho da metodologia proposta.

Por fim, arquitetar uma função de recompensa que estimule o agente a encontrar uma solução ótima ou receber punições que o leve a uma mudança de comportamento, é um desafio. Portanto, realizar a customização das recompensas para casos específicos, como o de mamografia, pode gerar uma abertura

para novos experimentos e resultados promissores.

Referências bibliográficas

[1] Recent advances and clinical applications of deep learning in medical image analysis. *Medical Image Analysis 79* (2022), 102444.

[2] Amin, H., and Siddiqui, W. J. Cardiomegaly. *StatPearls [internet]* (2020).

[3] Bai, H., Cao, M., Huang, P., and Shan, J. Self-supervised semi-supervised learning for data labeling and quality evaluation. *arXiv preprint arXiv:2111.10932* (2021).

[4] Bellman, R. Dynamic programming. *Science 153*, 3731 (1966), 34–37.

[5] Bhardwaj, A., Di, W., and Wei, J. *Deep Learning Essentials: Your hands-on guide to the fundamentals of deep learning and neural network modeling*. Packt Publishing Ltd, 2018.

[6] Borenstein, E., and Ullman, S. Combined top-down/bottom-up segmentation. *IEEE Transactions on pattern analysis and machine intelligence 30*, 12 (2008), 2109–2125.

[7] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. *arXiv preprint arXiv:1606.01540* (2016).

[8] Brust, C.-A., Käding, C., and Denzler, J. Active learning for deep object detection. *arXiv preprint arXiv:1809.09875* (2018).

[9] Budach, L., Feuerpfeil, M., Ihde, N., Nathansen, A., Noack, N., Patzlaff, H., Harmouch, H., and Naumann, F. The effects of data quality on machine learning performance. *arXiv preprint arXiv:2207.14529* (2022).

[10] Budd, S., Robinson, E. C., and Kainz, B. A survey on active learning and human-in-the-loop deep learning for medical image analysis. *Medical Image Analysis* (2021), 102062.

[11] Caicedo, J. C., and Lazebnik, S. Active object localization with deep reinforcement learning. In *Proceedings of the IEEE international conference on computer vision* (2015), pp. 2488–2496.

[12] Chollet, F. *Deep learning with Python*. Simon and Schuster, 2021.

[13] Costa, M. S. P. Maria montessori e seu método. *Linhas Críticas, Brasília 7*, 13 (2001), 305–320.

[14] Cruz, F., Wüppen, P., Magg, S., Fazrie, A., and Wermter, S. Agent-advising approaches in an interactive reinforcement learning scenario. In *2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)* (2017), IEEE, pp. 209–214.

[15] Da Silva, F. L., Hernandez-Leal, P., Kartal, B., and Taylor, M. E. Uncertainty-aware action advising for deep reinforcement learning agents. In *Proceedings of the AAAI conference on artificial intelligence* (2020), vol. 34, pp. 5792–5799.

[16] dos Santos Junior, E. S., de Oliveira Machado, A., Macedo, M. C., and dos Santos Souza, A. C. Reinforcement learning para treino do pac-man em speedrun. *Brazilian Journal of Development 5*, 11 (2019), 25927–25957.

[17] Du, B., Wang, Z., Zhang, L., Zhang, L., Liu, W., Shen, J., and Tao, D. Exploring representativeness and informativeness for active learning. *IEEE transactions on cybernetics 47*, 1 (2015), 14–26.

[18] Elezi, I., Yu, Z., Anandkumar, A., Leal-Taixe, L., and Alvarez, J. M. Towards reducing labeling cost in deep object detection. *arXiv preprint arXiv:2106.11921* (2021).

[19] ES, Q. Õ. *EDUCAÇÃO FÍSICA NA ESCOLA*. PhD thesis, Universidade Estadual Paulista, 2001.

[20] Esteva, A., Chou, K., Yeung, S., Naik, N., Madani, A., Mottaghi, A., Liu, Y., Topol, E., Dean, J., and Socher, R. Deep learning-enabled medical computer vision. *NPJ digital medicine 4*, 1 (2021), 1–9.

[21] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *nature 542*, 7639 (2017), 115–118.

[22] Fausett, L. V. *Fundamentals of neural networks: architectures, algorithms and applications*. Pearson Education India, 2006.

[23] Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., and Hovy, E. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075* (2021).

[24] Frazier, S., and Riedl, M. Improving deep reinforcement learning in minecraft with action advice. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* (2019), vol. 15, pp. 146–152.

[25] Ghesu, F.-C., Georgescu, B., Zheng, Y., Grbic, S., Maier, A., Hornegger, J., and Comaniciu, D. Multi-scale deep reinforcement learning for real-time 3d-landmark detection in ct scans. *IEEE transactions on pattern analysis and machine intelligence 41*, 1 (2017), 176–189.

[26] Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press, 2016.

[27] Grabski, F. *Semi-Markov processes: Applications in system reliability and maintenance*, vol. 599. Elsevier Amsterdam, 2015.

[28] Grigoletto, R. Scoring functions evaluation for active learning in humanoid robotics, 2020.

[29] Gupta, V., Taylor, C., Bonnet, S., Prevedello, L. M., Hawley, J., White, R. D., Flores, M. G., and Erdal, B. S. Deep learning-based automatic detection of poorly positioned mammograms to minimize patient return visits for repeat imaging: A real-world application. *arXiv preprint arXiv:2009.13580* (2020).

[30] Harmon, M. E., and Harmon, S. S. Reinforcement learning: A tutorial.

[31] Haussmann, E., Fenzi, M., Chitta, K., Ivanecky, J., Xu, H., Roy, D., Mittel, A., Koumchatzky, N., Farabet, C., and Alvarez, J. M. Scalable active learning for object detection. In *2020 IEEE intelligent vehicles symposium (iv)* (2020), IEEE, pp. 1430–1435.

[32] Ibarz, J., Tan, J., Finn, C., Kalakrishnan, M., Pastor, P., and Levine, S. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research 40*, 4-5 (2021), 698–721.

[33] Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A. A., Yogamani, S., and Pérez, P. Deep reinforcement learning for

autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems* (2021).

[34] König, J., Malberg, S., Martens, M., Niehaus, S., Krohn-Grimberghe, A., and Ramaswamy, A. Multi-stage reinforcement learning for object detection. In *Science and Information Conference* (2019), Springer, pp. 178–191.

[35] Krening, S. *Humans teaching intelligent agents with verbal instruction*. PhD thesis, Georgia Institute of Technology, 2019.

[36] Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images.

[37] Kwan, M. L., Kushi, L. H., Weltzien, E., Maring, B., Kutner, S. E., Fulton, R. S., Lee, M. M., Ambrosone, C. B., and Caan, B. J. Epidemiology of breast cancer subtypes in two prospective cohort studies of breast cancer survivors. *Breast Cancer Research 11*, 3 (2009), R31.

[38] Lapan, M. *Deep reinforcement learning hands-on*. Packt publishing, 2020.

[39] Li, G., He, B., Gomez, R., and Nakamura, K. Interactive reinforcement learning from demonstration and human evaluative feedback. In *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)* (2018), IEEE, pp. 1156–1162.

[40] Li, J., Chen, Y., Zhao, X., and Huang, J. An improved dqn path planning algorithm. *The Journal of Supercomputing 78*, 1 (2022), 616–639.

[41] Li, W., Zhao, R., and Wang, X. Human reidentification with transferred metric learning. In *Asian conference on computer vision* (2012), Springer, pp. 31–44.

[42] Li, X., Hu, X., Qi, X., Yu, L., Zhao, W., Heng, P.-A., and Xing, L. Rotation-oriented collaborative self-supervised learning for retinal disease diagnosis. *IEEE Transactions on Medical Imaging 40*, 9 (2021), 2284–2294.

[43] Liang, H., Yang, L., Cheng, H., Tu, W., and Xu, M. Human-in-the-loop reinforcement learning. In *2017 Chinese Automation Congress (CAC)* (2017), pp. 4511–4518.

[44] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco:

Common objects in context. In *European conference on computer vision* (2014), Springer, pp. 740–755.

[45] Lin, Z., Harrison, B., Keech, A., and Riedl, M. O. Explore, exploit or listen: Combining human feedback and policy model to speed up deep reinforcement learning in 3d worlds. *arXiv preprint arXiv:1709.03969* (2017).

[46] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., Van Der Laak, J. A., Van Ginneken, B., and Sánchez, C. I. A survey on deep learning in medical image analysis. *Medical image analysis 42* (2017), 60–88.

[47] Liu, Q., Cui, C., and Fan, Q. Self-adaptive constrained multi-objective differential evolution algorithm based on the state–action–reward–state–action method. *Mathematics 10*, 5 (2022), 813.

[48] Liu, T., Siegel, E., and Shen, D. Deep learning and medical image analysis for covid-19 diagnosis and prediction. *Annual Review of Biomedical Engineering 24* (2022).

[49] Liu, Z., Wang, J., Gong, S., Lu, H., and Tao, D. Deep reinforcement active learning for human-in-the-loop person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision* (2019), pp. 6122–6131.

[50] Mandel, T., Liu, Y.-E., Brunskill, E., and Popović, Z. Where to add actions in human-in-the-loop reinforcement learning. In *Thirty-First AAAI Conference on Artificial Intelligence* (2017).

[51] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[52] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *nature 518*, 7540 (2015), 529–533.

[53] Moghbel, M., Ooi, C. Y., Ismail, N., Hau, Y. W., and Memari, N. A review of breast boundary and pectoral muscle segmentation methods in computer-aided detection/diagnosis of breast mammography. *Artificial Intelligence Review* (2019), 1–46.

[54] Monarch, R. M. *Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI*. Simon and Schuster, 2021.

[55] Monowar, K. National institutes of health chest x-ray dataset. `https://www.kaggle.com/khanfashee/nih-chest-x-ray-14-224x224-resized`, May 2020.

[56] Morales, M. *Grokking deep reinforcement learning*. Manning Publications, 2020.

[57] Moreira, I., Rivas, J., Cruz, F., Dazeley, R., Ayala, A., and Fernandes, B. Deep reinforcement learning with interactive feedback in a human–robot environment. *Applied Sciences 10*, 16 (2020), 5574.

[58] Najar, A., and Chetouani, M. Reinforcement learning with human advice. a survey. *arXiv preprint arXiv:2005.11016* (2020).

[59] Nanni, L., Maguolo, G., and Paci, M. Data augmentation approaches for improving animal audio classification. *Ecological Informatics 57* (2020), 101084.

[60] Navidi, N. Human ai interaction loop training: New approach for interactive reinforcement learning. *arXiv preprint arXiv:2003.04203* (2020).

[61] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning.

[62] Ng, A. Y. *Shaping and policy search in reinforcement learning*. University of California, Berkeley, 2003.

[63] Nguyen, T., Hua, B.-S., Nguyen, D. T., and Phung, D. Single-click 3d object annotation on lidar point clouds.

[64] Olmin, A., Lindqvist, J., Svensson, L., and Lindsten, F. Active learning with weak labels for gaussian processes. *arXiv preprint arXiv:2204.08335* (2022).

[65] Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems 29* (2016).

[66] Otoofi, M. Object localization using deep reinforcement learning Mohammad Otoofi. Master's thesis, University of Glasgow, Scotland, 2018.

[67] Ottoni, A. L. C., Nepomuceno, E. G., de Oliveira, M. S., Cordeiro, L. T., and Lamperti, R. D. Análise da influência da taxa de aprendizado e do fator de desconto sobre o desempenho dos algoritmos q-learning e sarsa: aplicação do aprendizado por reforço na navegação autônoma. *Revista Brasileira de Computação Aplicada 8*, 2 (2016), 44–59.

[68] Padilla, R., Netto, S. L., and da Silva, E. A. A survey on performance metrics for object-detection algorithms. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)* (2020), IEEE, pp. 237–242.

[69] Pan, J., Bi, Q., Yang, Y., Zhu, P., and Bian, C. Label-efficient hybrid-supervised learning for medical image segmentation. *arXiv preprint arXiv:2203.05956* (2022).

[70] Pang, K., Dong, M., Wu, Y., and Hospedales, T. Meta-learning transferable active learning policies by deep reinforcement learning. *arXiv preprint arXiv:1806.04798* (2018).

[71] Pellegrini, J., and Wainer, J. Processos de decisão de markov: um tutorial. *Revista de Informática Teórica e Aplicada 14*, 2 (2007), 133–179.

[72] Pepik, B., Stark, M., Gehler, P., and Schiele, B. Teaching 3d geometry to deformable part models. In *2012 IEEE conference on computer vision and pattern recognition* (2012), IEEE, pp. 3362–3369.

[73] Polyzotis, N., and Zaharia, M. What can data-centric ai learn from data and ml engineering? *arXiv preprint arXiv:2112.06439* (2021).

[74] Poole, D. L., and Mackworth, A. K. *Artificial Intelligence: foundations of computational agents*. Cambridge University Press, 2010.

[75] Ramalho, A. S. B., Olaia, L., and Gabini, W. S. Diversidade no educar: um estudo sobre metodologias através da pedagogia waldorf e do método montessori. *Revista Eletrônica da Educação 4*, 1 (2021), 22–42.

[76] Rameh, L. Método paulo freire: Uma contribuição para a história da educação brasileira. *V Colóquio Internacional Paulo Freire-Recife 19* (2005).

[77] Ravichandiran, S. *Hands-on reinforcement learning with Python: master reinforcement and deep reinforcement learning using OpenAI gym and TensorFlow*. Packt Publishing Ltd, 2018.

[78] Rebuffi, S.-A., Ehrhardt, S., Han, K., Vedaldi, A., and Zisserman, A. Semi-supervised learning with scarce annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (2020), pp. 762–763.

[79] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 779–788.

[80] Reitmaier, T., Calma, A., and Sick, B. Transductive active learning–a new semi-supervised learning approach based on iteratively refined generative models to capture structure in data. *Information Sciences 293* (2015), 275–298.

[81] Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Gupta, B. B., Chen, X., and Wang, X. A survey of deep active learning. *ACM Computing Surveys (CSUR) 54*, 9 (2021), 1–40.

[82] Ristani, E., Solera, F., Zou, R., Cucchiara, R., and Tomasi, C. Performance measures and a data set for multi-target, multi-camera tracking. In *European conference on computer vision* (2016), Springer, pp. 17–35.

[83] Rosenblatt, F. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957.

[84] Sager, C., Janiesch, C., and Zschech, P. A survey of image labelling for computer vision applications. *Journal of Business Analytics 4*, 2 (2021), 91–110.

[85] Saripalli, V. R., Pati, D., Potter, M., Avinash, G., and Anderson, C. W. Ai-assisted annotator using reinforcement learning. *SN Computer Science 1*, 6 (2020), 1–8.

[86] Schmarje, L., Liao, Y.-H., and Koch, R. A data-centric image classification benchmark. In *NeurIPS 2021 Datacentric AI workshop* (2021).

[87] Semsarian, C., Ingles, J., Maron, M. S., and Maron, B. J. New perspectives on the prevalence of hypertrophic cardiomyopathy. *Journal of the American College of Cardiology 65*, 12 (2015), 1249–1254.

[88] Sener, O., and Savarese, S. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489* (2017).

[89] Settles, B. Active learning literature survey.

[90] Settles, B. Active learning literature survey. 2010. *Computer Sciences Technical Report 1648* (2014), 10.

[91] Sewak, M. *Deep Reinforcement Learning: Frontiers of Artificial Intelligence*. Springer Singapore, 2019.

[92] Shalev-Shwartz, S., and Ben-David, S. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[93] Shorten, C., and Khoshgoftaar, T. M. A survey on image data augmentation for deep learning. *Journal of big data 6*, 1 (2019), 1–48.

[94] Shorten, C., and Khoshgoftaar, T. M. An exploration of consistency learning with data augmentation. In *The International FLAIRS Conference Proceedings* (2022), vol. 35.

[95] Shurrab, S., and Duwairi, R. Self-supervised learning methods and applications in medical imaging analysis: A survey. *arXiv preprint arXiv:2109.08685* (2021).

[96] Subramanya Vokuda, P. Interactive object detection. *Technical Report/Hochschule Bonn-Rhein-Sieg University of Applied Sciences, Department of Computer Science* (2019).

[97] Suganyadevi, S., Seethalakshmi, V., and Balasamy, K. A review on deep learning in medical image analysis. *International Journal of Multimedia Information Retrieval 11*, 1 (2022), 19–38.

[98] Sun, L., and Gong, Y. Active learning for image classification: A deep reinforcement learning approach. In *2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI)* (2019), IEEE, pp. 71–76.

[99] Sun, Y., Mallick, T., Balaprakash, P., and Macfarlane, J. A data-centric weak supervised learning for highway traffic incident detection. *arXiv preprint arXiv:2112.09792* (2021).

[100] Sutton, R. S. Reinforcement learning: Past, present and future. In *Asia-Pacific Conference on Simulated Evolution and Learning* (1998), Springer, pp. 195–197.

[101] Sutton, R. S., and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

[102] Tajbakhsh, N., Jeyaseelan, L., Li, Q., Chiang, J. N., Wu, Z., and Ding, X. Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. *Medical Image Analysis* (2020), 101693.

[103] Tajmajer, T. Modular multi-objective deep reinforcement learning with decision values. In *2018 Federated conference on computer science and information systems (FedCSIS)* (2018), IEEE, pp. 85–93.

[104] Tardy, M., and Mateus, D. Looking for abnormalities in mammograms with self-and weakly supervised reconstruction. *IEEE Transactions on Medical Imaging 40*, 10 (2021), 2711–2722.

[105] Torrey, L., and Taylor, M. Teaching on a budget: Agents advising agents in reinforcement learning. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems* (2013), pp. 1053–1060.

[106] Tsuneki, M. Deep learning models in medical image analysis. *Journal of Oral Biosciences* (2022).

[107] Tsutsui, S., Crandall, D., and Yu, C. Human-inspired data-centric computer vision.

[108] Vailoppilly, A. P., Sakthivel, R., and Kumar, R. S. All-in-one data cleansing tool.

[109] Van Engelen, J. E., and Hoos, H. H. A survey on semi-supervised learning. *Machine Learning 109*, 2 (2020), 373–440.

[110] Vokuda, P. S. *Interactive Object Detection*. Hochschule Bonn-Rhein-Sieg, 2019.

[111] Wang, J., Yan, Y., Zhang, Y., Cao, G., Yang, M., and Ng, M. K. Deep reinforcement active learning for medical image classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (2020), Springer, pp. 33–42.

[112] Wang, K., Zhang, D., Li, Y., Zhang, R., and Lin, L. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology 27*, 12 (2016), 2591–2600.

[113] Wang, M., and Hua, X.-S. Active learning in multimedia annotation and retrieval: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST) 2*, 2 (2011), 1–21.

[114] Watkins, C. J., and Dayan, P. Q-learning. *Machine learning 8*, 3 (1992), 279–292.

[115] Wei, T., Faulkner, T. A. K., and Thomaz, A. L. Extending policy shaping to continuous state spaces (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence* (2021), vol. 35, pp. 15919–15920.

[116] Yang, J., Fan, J., Wei, Z., Li, G., Liu, T., and Du, X. Cost-effective data annotation using game-based crowdsourcing. *Proceedings of the VLDB Endowment 12*, 1 (2018), 57–70.

[117] Yang, Y., Ma, Z., Nie, F., Chang, X., and Hauptmann, A. G. Multi-class active learning by uncertainty sampling with diversity maximization. *International Journal of Computer Vision 113*, 2 (2015), 113–127.

[118] Yin, W., Heinecke, S., Li, J., Keskar, N. S., Jones, M., Shi, S., Georgiev, S., Milich, K., Esposito, J., and Xiong, C. Combining data-driven supervision with human-in-the-loop feedback for entity resolution. *arXiv preprint arXiv:2111.10497* (2021).

[119] Ying, H., Wang, H., Shao, T., Yang, Y., and Zhou, K. Unsupervised image generation with infinite generative adversarial networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 14284–14293.

[120] Zhang, S., Yin, J., and Guo, W. Pool-based active learning with query construction. In *Foundations of Intelligent Systems*. Springer, 2011, pp. 13–22.

[121] Zhang, X., Chang, D., Qi, W., and Zhan, Z. A study on different functionalities and performances among different activation functions across different anns for image classification. In *Journal of Physics: Conference Series* (2021), vol. 1732, IOP Publishing, p. 012026.

[122] Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., and Tian, Q. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision* (2015), pp. 1116–1124.

[123] Zheng, L., Yang, Y., and Tian, Q. Sift meets cnn: A decade survey of instance retrieval. *IEEE transactions on pattern analysis and machine intelligence 40*, 5 (2017), 1224–1244.

[124] Zhou, Z.-H. A brief introduction to weakly supervised learning. *National science review 5*, 1 (2018), 44–53.

[125] Zhu, J., and Ma, M. Uncertainty-based active learning with instability estimation for text classification. *ACM Transactions on Speech and Language Processing (TSLP) 8*, 4 (2012), 1–21.