

JOGO DE DOMINÓ AUTOMATIZADO

Francisco Lucas Sens

Prof. Aurélio Faustino Hoppe – Orientador

1 INTRODUÇÃO

Segundo Brookshear (2008), a humanidade desde a criação dos primeiros computadores tenta desenvolver sistemas que igualem a sua inteligência, ou a superem. Alan Turing que é considerado o pai da computação, em 1940 criou o primeiro algoritmo de xadrez, escrevendo-o em papel, porém com a tecnologia disponível na época não era possível executar tal algoritmo (TURING, 1988, tradução nossa). Segundo Russell e Norvig (2013), os programas tem superado jogadores que são campeões em jogos como xadrez, damas e Othello.

No ano de 1997, o computador Deep Blue da empresa IBM derrotou Garry Kasparov que era o campeão mundial de xadrez, após cinco partidas jogando contra o computador, Garry renunciou, dando a vitória para a máquina da IBM (HSU, 1999, tradução nossa). O jogo Go é considerado um dos jogos mais difíceis do mundo. No Ano de 2017 o sistema de inteligência artificial chamado AlphaGo que foi desenvolvido pelo DeepMind, foi capaz de derrotar o campeão mundial Lee Sedol do jogo Go (YANNAKAKIS; TOGELIUS, 2018, tradução nossa). A OpenAi Five é um sistema de inteligência artificial que em 2019 conseguiu derrotar o time campeão de Dota 2, vencendo 99.4% do total de 7000 partidas. A OpenAi Five conseguiu reagir a um evento do game em 217ms em média, enquanto a média visual das reações humanas é de aproximadamente de 250ms (BERNER *et al.*, 2019).

Russell e Norvig (2013) afirmam que sem dúvida o uso da Inteligência Artificial (IA) é constantemente utilizado no dia a dia, seja para tarefas específicas como jogos de xadrez, demonstração de teoremas matemáticos, direção de carros autônomos, diagnóstico de doenças, tradução automática, reconhecimento de voz, entre outros. Além disso, o avanço do hardware e software proporciona uma grande evolução na área de jogos, sendo eles físicos com tabuleiros ou digitais, o uso de IA em jogos contribuiu de várias formas, seja para testes automatizados em jogos, personagens não jogáveis chamados de *Non-Player Character* (NPC), geração procedural de mapas, modelagem de jogadores (YANNAKAKIS; TOGELIUS, 2018, tradução nossa).

De acordo com Yannakakis e Togelius (2018), outro avanço notável é a utilização de inteligência artificial para aprimorar a imersão dos jogadores através de cenários e personagens mais realistas. Os jogos trazem benefícios como melhorar a agilidade, raciocínio lógico, coordenação motora, entre outros. Já a robótica é constantemente utilizada nos dias de hoje, seja para realizar cirurgias, desativar bombas, produção automatizada de produtos comestíveis, carros, entre outros. Conclui-se então, que a utilização da robótica atrelada ao uso de inteligência artificial surge a ideia de sistemas autônomos.

Segundo Selenguini, Amaral e Silva (2016), os sistemas autônomos são capazes de realizar uma tarefa de forma adequada sem uma interferência externa, ou seja, eles são capazes de aprender com o ambiente para gerar suas ações. Além disso, utilizam essas ações para planejar e construir uma sequência de ações com o intuito de atingir seu objetivo. Neste contexto, são capazes de realizar tarefas específicas com a finalidade de jogar uma partida de dominó, capacidade de enxergar o jogo, movimentar as peças e ainda pensar em como criar estratégias de jogo.

De acordo com Dartigues *et al.* (2013), jogos de tabuleiros são uma das atividades de lazer mais estimulantes. Os autores ainda afirmam que tais jogos podem retardar a demência, declínio cognitivo e risco de depressão independentemente da idade. No caso de idosos ou pessoas com dificuldades de locomoção são uma forma de entretenimento e ao mesmo tempo, estimulando o raciocínio lógico e concentração assim como, tornando-se uma forma de companhia para aqueles que moram sozinhos.

Diante deste cenário, este trabalho propõe-se a desenvolver um dispositivo que seja capaz de jogar dominó de forma autônoma, utilizando técnicas de processamento de imagens, aprendizado de máquina e robótica.

1.1 OBJETIVOS

O objetivo deste trabalho é disponibilizar um dispositivo para jogar de forma autônoma partidas de dominó contra humanos.

Os objetivos específicos são:

- analisar a usabilidade do projeto desenvolvido assim como o sentimento em relação a humano-computador dos usuários;
- validar e analisar o tempo de resposta da segmentação e classificação, a decisão da melhor jogada, a precisão da locomoção e a disposição das peças assim como a assertividade das jogadas realizadas;
- disponibilizar um dispositivo de baixo custo e de fácil montagem.

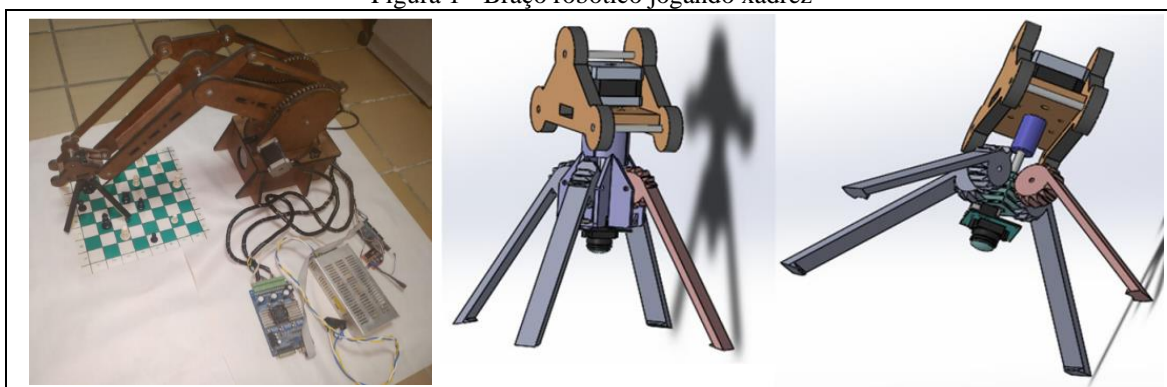
2 TRABALHOS CORRELATOS

Neste capítulo são apresentados trabalhos com características semelhantes aos principais objetivos do estudo proposto. A seção 2.1 apresenta um dispositivo que joga de forma autônoma xadrez (TORO; ROBLES-ALGARÍN; RODRÍGUEZ-ÁLVAREZ, 2019). A seção 2.2 discorre sobre um dispositivo que joga de forma autônoma o jogo da velha (JAIN, 2015). Por fim, a seção 2.3 apresenta um dispositivo capaz de jogar de forma autônoma o jogo de damas (ALIANE; BEMPOSTA, 2011).

2.1 DESIGN AND CONSTRUCTION OF A COST-EFFECTIVE DIDACTIC ROBOTIC ARM FOR PLAYING CHESS, USING AN ARTIFICIAL VISION SYSTEM

Toro, Robles-Algarín e Rodríguez-Álvarez (2019) desenvolveram um dispositivo capaz de jogar xadrez de forma autônoma. Para isso, os autores utilizaram um Arduino Mega, uma câmera com abertura de 170° graus, dois motores de paços TB6560 e quatro pegadores chamados de grippers. A câmera foi acoplada aos pegadores que foram desenhados utilizando o software SolidWorks, conforme mostra a Figura 1.

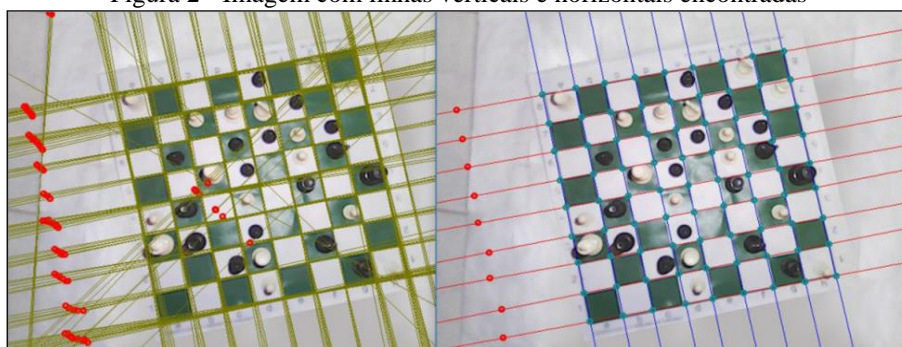
Figura 1 - Braço robótico jogando xadrez



Fonte: Toro, Robles-Algarín e Rodríguez-Álvarez (2019).

Para realizar a segmentação do tabuleiro, Toro, Robles-Algarín e Rodríguez-Álvarez (2019) utilizaram o detector de bordas de Canny, a transformada de Hough e a Kernel Density Estimation (KDE). O KDE foi utilizado para encontrar dois grupos de linhas que possuem o ângulo de 0° com uma diferença de 90° graus. Estes grupos correspondem as linhas verticais e horizontais, pois a transformada de Hough encontrava várias linhas redundantes para cada linha do tabuleiro, conforme mostra a Figura 2.

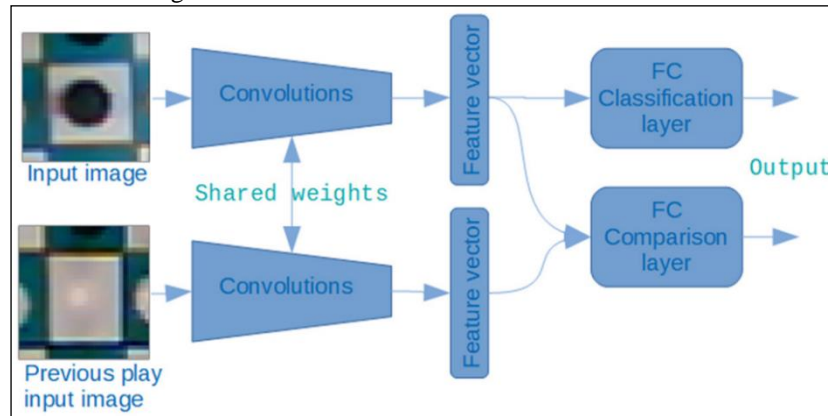
Figura 2 - Imagem com linhas verticais e horizontais encontradas



Fonte: Toro, Robles-Algarín e Rodríguez-Álvarez (2019).

Segundo Toro, Robles-Algarín e Rodríguez-Álvarez (2019), o sistema acompanha as peças durante o jogo, utilizando a detecção de mudanças para rastrear seus movimentos desde as posições iniciais. Para a classificação das peças e detecção de movimentação das peças, os autores utilizaram uma Rede Neural Convolutiva (CNN). O modelo foi desenvolvido utilizando Keras, com duas camadas, a primeira contendo 32 filtros e a segunda 64 filtros com um dropout de 20%, conforme apresenta a Figura 3. A primeira camada é responsável pela segmentação das peças, enquanto a segunda é responsável pela detecção de movimentação de peças, porém quando as peças possuem a mesma cor do fundo, onde o contraste das cores é sutil, utilizou-se uma Siamese Network, que compara a imagem atual com a imagem anterior.

Figura 3 - Camadas da Rede Neural Convolutacional



Fonte: Toro, Robles-Algarín e Rodríguez-Álvarez (2019).

Toro, Robles-Algarín e Rodríguez-Álvarez (2019) utilizaram uma camada densa com seis saídas para classificação e uma camada densa com uma saída para detecção de movimentação da peça. As classes utilizadas na primeira camada são: fundo verde, fundo branco, peça branca sobre fundo branco, peça preta sobre fundo branco, peça branca sobre fundo preto e peça preta sobre fundo preto. O resultado gerado pela segunda camada é um valor entre 0 e 1, no qual valores próximos a 1 representam uma mudança significativa de posição da peça, enquanto valores próximos de 0 representam pouca mudança.

Para a criação da base de dados, Toro, Robles-Algarín e Rodríguez-Álvarez (2019) utilizaram um software com a opção para realizar a auto segmentação da imagem, permitindo ao usuário apontar o rótulo de cada parte que foi segmentada. Após o processo de definir os rótulos, foi gerado um arquivo `npz` contendo a imagem, cantos e os rótulos. Preparando cinco imagens com os rótulos, alterando as condições de iluminação, movimentação da câmera do robô em alguns centímetros, ficando as peças levemente deslocadas, gerando um cenário mais realista.

De acordo com Toro, Robles-Algarín e Rodríguez-Álvarez (2019), o treinamento foi realizado importando as imagens, cantos e rótulos, realizando o redimensionamento das imagens para 30x30 com três camadas de cores. As imagens de treinamento e validação foram separadas de forma randômica utilizando a razão de 8:2. Posteriormente aplicou-se a técnica de *data augmentation*, rotacionando e invertendo as imagens para obter melhores resultados, totalizando 8 amostras para cada imagem.

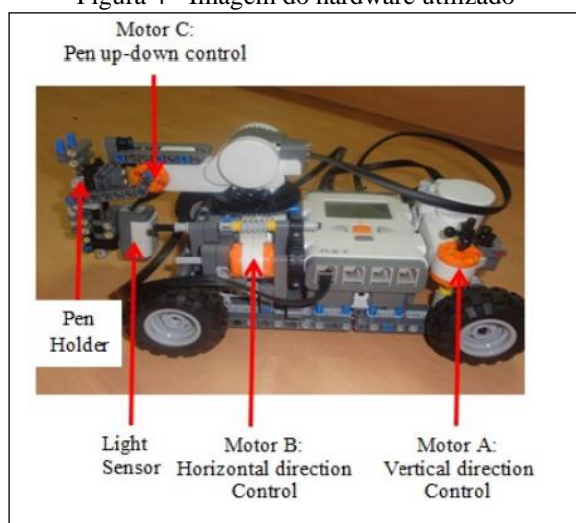
O software desenvolvido utiliza o Stockfish Chess Engine para a tomada de decisão de qual peça será jogada, assim como para fazer a validação de movimentação. Posteriormente, se estabelece o ponto-alvo utilizando os cantos e o centro de onde a peça está situada. Neste instante, o robô está em sua posição de repouso, no qual a câmera consegue ver todo o tabuleiro. Porém, quando ocorre a movimentação do braço a câmera perde os cantos do tabuleiro, impossibilitando o rastreamento da posição onde a peça será colocada. Por este motivo, Toro, Robles-Algarín e Rodríguez-Álvarez (2019) utilizaram o algoritmo Kanade-Lucas-Tomasi para rastrear a peça. Já a profundidade do braço em relação ao tabuleiro é obtida através da cinemática do robô. No qual foi implementado uma máscara em torno do ponto de destino, pois rastrear somente onde a peça deveria estar não se mostrou confiável.

Toro, Robles-Algarín e Rodríguez-Álvarez (2019) relatam que nos testes foram possíveis realizar partidas de xadrez de forma autônoma, utilizando o braço robótico desenvolvido. A primeira camada da Rede Neural Convolutacional é responsável pela segmentação das imagens resultou em uma precisão de 98,6%, enquanto a segunda camada responsável pela detecção de movimentação das peças resultou em uma precisão de 99,4%. Quando as condições de iluminação são ruins, o filtro de Canny falha e quando as peças projetam sombras a transformada de Hough também falha, dificultando a extração das informações do tabuleiro. Conclui-se que foi possível jogar de forma automatiza o jogo de xadrez.

2.2 AN INTELLIGENT METHOD FOR SOLVING TIC-TAC-TOE PROBLEM

Jain (2015) desenvolveu um dispositivo capaz de jogar o jogo da velha de forma autônoma. Para isso, utilizou-se um Kit Lego composto por um Bricks NXT que é microprocessador de 32 bits, que contém um display de LCD monocromático com uma resolução de 100 x 60 pixels. Além do KIT foi utilizado um fototransistor SFH 309-4, quatro sensores e três motores. O primeiro motor chamado de Motor A, controla a posição vertical, o motor B controla a posição horizontal e o motor C movimenta a caneta de cima para baixo, conforme mostra a Figura 4.

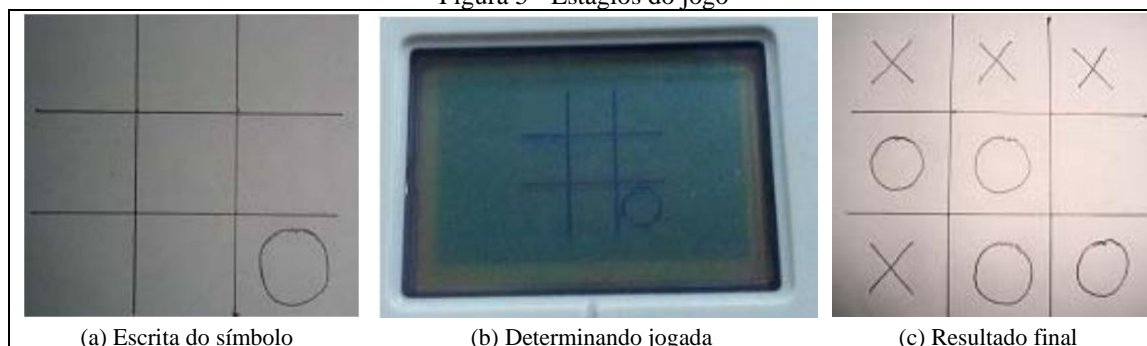
Figura 4 - Imagem do hardware utilizado



Fonte: Jain (2015).

Segundo Jain (2015), o robô se movimenta por uma matriz de 3x3, analisando cada uma das células, utilizando o fototransistor. Já a classificação da imagem é constituída em quatro etapas. Primeiramente aplicou-se um Limiar, convertendo a imagem para preto e branco. Posteriormente, segmentou-se a imagem, aplicando na sequência o algoritmo de afinamento chamado de Thinning, deixando a imagem com 1 pixel. Por fim, comparou-se a largura, posição, direção e número de pontos, caso alguma destas verificações estivesse correta, o símbolo é considerado como reconhecido. Cada símbolo reconhecido é armazenado e apresentado no display, conforme mostra a Figura 5.

Figura 5 - Estágios do jogo



Fonte: adaptado de Jain (2015).

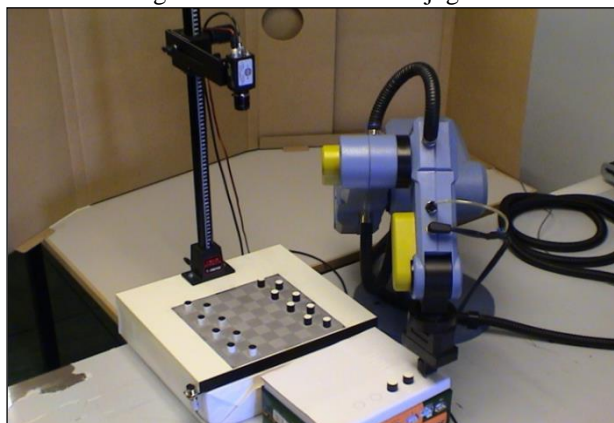
Para determinar qual seria a melhor jogada foi utilizado o algoritmo de Minimax. O algoritmo Minimax segundo Russell, Norvig (2013, p. 143), “retorna à ação correspondente ao melhor movimento possível, isto é, o movimento que leva ao resultado com a melhor utilidade, sob a suposição de que o oponente joga para minimizar a utilidade.”.

Segundo Jain (2015), nos testes realizados, foi possível jogar o jogo da velha de forma autônoma, o robô projetado escaneia, resolve e escreve com sucesso no papel e no display, concentrando-se somente em vencer a partida ou empatar. Segundo Jain (2015), uma das possíveis extensões deste trabalho seria a utilização de uma câmera para leitura dos movimentos ao invés do sensor de luz utilizado, o que tornaria o tempo necessário para realização das jogadas menor. Outra extensão citada foi a utilização de algoritmos que consomem menos tempo e conseguem melhores resultados, porém não foi citado nenhum algoritmo em específico.

2.3 A CHECKERS PLAYING ROBOT: A DIDACTIC PROJECT

Aliane e Bemposta (2011) desenvolveram um robô autônomo para jogar dama. Para isso, utilizou-se um braço articulado SCORBOT-ER IX, com uma câmera monocromática. O sistema é dividido em três módulos, sistema de visão, motor do jogo e controle do robô. O sistema de visão é responsável pelo processamento das imagens e extração das peças. O motor do jogo é responsável por definir qual é a melhor peça a ser jogada, criando um comando que será executado. O sistema de controle do robô é responsável por interpretar o comando e repassar através de instruções que o robô consiga executar. A Figura 6 exemplifica o robô realizando uma jogada.

Figura 6 - Robô realizando jogada

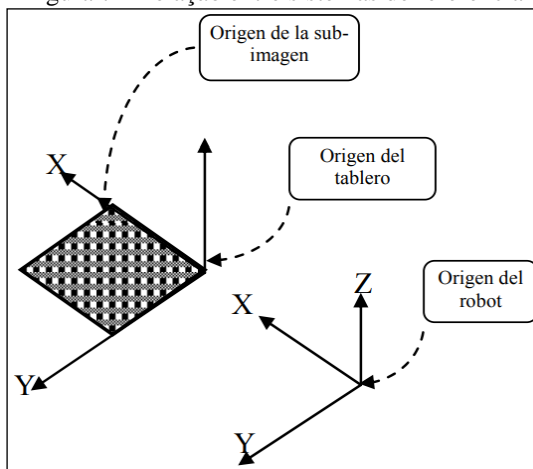


Fonte: Aliane e Bemposta (2011).

Para facilitar o processo de extração do tabuleiro da imagem, alinhrou-se o sistema de referência do robô com o tabuleiro, conforme mostra a Figura 7. Após a captura da imagem, extraiu-se o tabuleiro, aplicando várias técnicas de processamento de imagens, uma delas é o Limiar, que converte uma imagem para as cores branco e preto dependendo de um Limiar, ou seja, para valores acima do Limiar se atribui branco, caso contrário, entende-se que o valor do pixel é preto. As demais técnicas não foram explicadas.

Segundo Aliane e Bemposta (2011), a classificação das rainhas e peões foi realizada pela análise do contorno, isolando os objetos que atendem a critérios de área e perímetro. Estes objetos são utilizados para obter a posição das peças no tabuleiro. Para isso, utilizou-se a linguagem C++ para técnicas de processamento de imagens. A Figura 7 exibe a relação entre sistemas de referência.

Figura 7 - Relação entre sistemas de referência



Fonte: Aliane e Bemposta (2011).

De acordo com Aliane e Bemposta (2011), a partir do processamento de imagens obtém-se uma matriz de oito colunas por oito linhas, com os seguintes valores: 0 para ausência de fundo, 1 para fundo preto, 2 para dama preta, 3 para fundo branco e 4 para dama branca. Esta matriz é passada para o motor do jogo que é encarregado por determinar as jogadas que o robô deve fazer. Para isso, foi utilizado o algoritmo Minimax com Poda, que consiste em escolher a jogada com a melhor pontuação. Após realizar a escolha de qual peça será jogada, é realizado o procedimento para persistir a jogada em um arquivo para que o usuário possa visualizar. Os autores destacam que o motor de jogo foi implementado utilizando a linguagem JAVA.

O módulo principal é composto por vários programas desenvolvidos em ACL utilizando a linguagem SCORBOT, que são responsáveis pela realização dos movimentos do robô. Para realizar todas as jogadas possíveis foram criados dez comandos. Após a definição da jogada, o módulo principal interpreta e gera uma lista dos comandos que deverão ser executados, cada comando irá executar um outro programa. Já a comunicação com o braço é realizada através de uma porta serial RS-232.

De acordo Aliane e Bemposta (2011), quando ocorre uma movimentação do tabuleiro, ou movimentação da câmera ou quando ocorre uma mudança de iluminação se faz necessário realizar calibrações que demoram aproximadamente 10 minutos. Dentre as calibrações realizadas estão: alinhamento inicial dos eixos do robô com

os eixos do tabuleiro, extração da imagem do tabuleiro obtendo a origem, largura e altura, encontrar o valor utilizado no Limiar para conseguir diferenciar damas pretas e damas brancas.

Aliane e Bemposta (2011) relatam que a partir dos testes realizados foi possível jogar damas de forma autônoma e que para realizar o processamento da imagem, decidir qual peça será jogada e realizar a movimentação o tempo de resposta foi de 1 minuto quando o jogo é iniciado e no final de 10 segundos. Ainda segundo os autores, este tempo dá-se por dois fatores: o número de peças no tabuleiro e profundidade do algoritmo de Minimax com poda, que nos testes foi utilizado o nível sete de profundidade. Além disso, Aliane e Bemposta (2011) também apontam como limitação a necessidade de calibração e o tempo para executá-la, sendo necessário utilizar, as vezes, a ferramenta Inspector Matrox.

3 PROPOSTA

Esse capítulo tem como objetivo apresentar a justificativa para a elaboração deste trabalho, assim como os requisitos e metodologias que serão adotadas. Também serão apresentadas breves revisões bibliográficas das principais áreas de estudo que serão exploradas.

3.1 JUSTIFICATIVA

No Quadro 1 é possível observar um comparativo dos métodos escolhidos para a resolução dos problemas propostos pelos respectivos trabalhos, onde as linhas representam as características destacadas e as colunas representam os trabalhos.

Quadro 1 - Comparativo entre trabalhos correlatos

Trabalhos Características	Toro, Robles-Algarín e Rodríguez-Álvarez (2019)	Jain (2015)	Aliane e Bemposta (2011)
Cenário	Jogo de Xadrez	Jogo da velha	Jogo de dama
Dispositivo de captura	Câmera com abertura de 170°	Fototransistor	Câmera monocromática
Definição da melhor jogada	Stockfish Chess Engine	Minimax	Minimax com poda
Identificação e classificação do objeto	Canny, Transformada de Hough, KDE e uma rede CNN	Limiar, Segmentação, Thinning algoritmo, realizando comparações.	Não explicado profundamente
Dispositivo de movimentação	Criação própria	Kit Lego Mindstorms composto por um Bricks NXT	SCORBOT-ER IX de 5 graus

Fonte: elaborado pelo autor.

Conforme pode ser visto no Quadro 1, os trabalhos apresentam cenários diferentes. Toro, Robles-Algarín e Rodríguez-Álvarez (2019) propuseram um robô capaz de jogar xadrez de forma autônoma. Já Jain (2015) apresentou um robô autônomo para o jogo da velha e Aliane e Bemposta (2011) um robô autônomo para o jogo de dama. Toro, Robles-Algarín e Rodríguez-Álvarez (2019) utilizaram uma câmera com abertura de 170°, sendo posicionada no braço articulado. No qual, dependendo da altura em que o braço se encontra é possível visualizar todo o tabuleiro ou parte dele. Observa-se também que o trabalho de Jain (2015) optou por um fototransistor, que permite visualizar célula por célula do tabuleiro, obrigando a realizar o deslocamento pelo tabuleiro para conseguir extrair as informações. Aliane e Bemposta (2011) utilizaram uma câmera monocromática acoplada a pedestal fixo, possibilitando assim visualizar todo o tabuleiro.

Outro aspecto comparado é a definição de melhor jogada. No estudo de Toro, Robles-Algarín e Rodríguez-Álvarez (2019) foi utilizado Stockfish Chess Engine, Jain (2015) utilizou o algoritmo Minimax e Aliane e Bemposta (2011) utilizaram o algoritmo de Minimax com poda. Para identificação e classificação dos objetos no trabalho de Toro, Robles-Algarín e Rodríguez-Álvarez (2019) utilizaram o algoritmo de Canny, Transformada de Hough, Kernel Density Estimation (KDE) e uma Rede Neural Convolucional. Jain (2015) utilizou a Limiarização, alguma técnica de segmentação que não foi informada, e o algoritmo de Thinning, comparando a largura, posição, direção e número de pontos. E, Aliane e Bemposta (2011) não apresentaram o detalhamento da técnica utilizada.

Por fim, observou-se que para movimentação, Toro, Robles-Algarín e Rodríguez-Álvarez (2019) montaram o seu próprio dispositivo, para tal foi necessário um Arduino Mega, uma câmera, dois motores de paços

TB6560 e quatro pegadores. Já Jain (2015) utilizou um Kit Lego Mindstorms composto por um Bricks NXT e Aliane e Bemposta (2011) utilizou o SCORBOT-ER IX.

A partir das características descritas acima, este trabalho se torna relevante pelo fato de disponibilizar um dispositivo que seja capaz de jogar partidas de dominó contra humanos de forma automática, assim permitindo que pessoas possam jogar de forma individualizada, se divertindo ou melhorando sua habilidade no jogo, ou até mesmo podendo servir de distração para pessoas que vivem sozinhas. Além disso, em termos técnicos, este trabalho trará contribuições práticas para futuras implementações a respeito da utilização de Redes Neurais Convolucionais para segmentação e classificação, o algoritmo Minimax será utilizado para definir qual a melhor peça a ser jogada, a extração da localização espacial, a geração dos comandos necessários para realizar a movimentação e por último, porém não menos importante, a movimentação da peça em si.

3.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

O método a ser desenvolvido está dividido em dois módulos, um responsável pelo reconhecimento e definição das jogadas e um hardware para efetuar a movimentação/colocação das peças.

O módulo de reconhecimento e definição das jogadas deverá:

- a) ser capaz de comunicar-se com o hardware utilizando o Bluetooth para capturar a foto do jogo (Requisito Não Funcional - RNF);
- b) ser capaz de segmentar, utilizando uma Mask R-CNN (RNF);
- c) ser capaz de classificar as peças de dominó (Requisito Funcional - RF);
- d) ser capaz de escolher uma peça válida, utilizando o algoritmo Minimax (RNF);
- e) indicar e enviar para o hardware qual peça será movimentada (RF);
- f) utilizar a linguagem Python e o framework Flask (RNF);
- g) deve ser utilizado a biblioteca Keras e/ou Tensorflow para segmentação e classificação (RNF);
- h) ser capaz de estabelecer a jogada em no máximo 1 minuto (RNF);
- i) armazenar os dados no banco de dados MongoDB (RNF).

O hardware de movimentação deverá:

- a) capturar foto do jogo em andamento utilizando uma webcam, sendo posicionada na parte superior ou inferior do jogo (RNF);
- b) ser capaz de estabelecer os comandos necessários para movimentação da peça (RF);
- c) mover a peça de dominó selecionada até o local definido (RF);
- d) utilizar o Kit Lego Mindstorms ou Raspberry como base para montar o hardware (RNF);
- e) utilizar a linguagem Python para estabelecer os comandos, enviar os comandos para movimentação e obter a foto capturada (RNF).

3.3 METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

- a) levantamento bibliográfico: buscar fontes bibliográficas relacionadas ao objetivo do estudo proposto, como movimentação de dispositivos, segmentação e classificação de objetos em imagens;
- b) elicitação dos requisitos: reavaliar os requisitos funcionais e não funcionais, com o objetivo de atender o estudo proposto;
- c) especificação do trabalho: elaborar diagrama de classes e diagrama de casos de uso utilizando a ferramenta PlantUML;
- d) coleta de imagens das peças de dominós: coletar imagens que serão utilizadas para treinamento das Redes Neurais Convolucionais;
- e) rotulação das imagens coletadas: rotular as imagens de acordo com as classes definidas, estas classes serão definidas durante a implementação, esta etapa se faz necessário para que seja possível alcançar um percentual de acerto aceitável;
- f) desenvolvimento da segmentação: segmentar as peças de dominó tendo uma imagem como entrada. Será utilizada a biblioteca Mask R-CNN para gerar os pesos da Rede Neural Convulocional e, posteriormente realizar a predição da imagem recebida, assim, obtendo as peças de dominó segmentadas. Para tal será utilizada a linguagem de programação Python utilizando a IDE PyCharm;
- g) pesquisa e definição do algoritmo para classificação de objetos: definir qual algoritmo será utilizado para realizar a classificação das peças de dominó;
- h) desenvolvimento do classificador: implementar para que seja possível classificar as peças de dominó tendo a imagem segmentada como entrada;
- i) desenvolvimento do servidor web: implementar um servidor no qual seja possível receber uma

- imagem, realizando a chamada do procedimento de predição, caso constatado que existe uma peça de dominó disponível para ser jogada, escolher o local onde a peça será posicionada, chamando o procedimento para o dispositivo realizar a movimentação da peça, será utilizada a linguagem Python;
- j) pesquisa e definição do hardware: definir o hardware que será utilizado, sendo eles o Kit Lego ou Raspberry, câmera e sensores/motores para realizar a movimentação das peças;
 - k) montagem do dispositivo: montar o dispositivo para que ele possa realizar movimentações de acordo com coordenadas definidas pelo servidor web, deslocando as peças de dominó para o local selecionado. Nas implementações necessárias será utilizada a linguagem Python;
 - l) testes: jogar partidas de dominó, analisando o tempo de resposta das jogadas, precisão dos movimentos, precisão da classificação das peças. Será analisado também a aceitabilidade do usuário com em relação ao dispositivo e sua usabilidade.

As etapas serão realizadas nos períodos relacionados no Quadro 2.

Quadro 2 - Cronograma de atividades a serem realizadas

etapas / quinzenas	2020				2021											
	nov.		dez.		jan.		fev.		mar.		abr.		maio		jun.	
	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
levantamento bibliográfico																
elicitação dos requisitos																
especificação do trabalho																
coleta de imagens das peças de dominós																
rotulação das imagens coletadas																
desenvolvimento da segmentação																
pesquisa e definição do algoritmo para classificação de objetos																
desenvolvimento do classificador																
desenvolvimento do servidor web																
pesquisa e definição do hardware																
montagem do dispositivo																
testes																

Fonte: elaborado pelo autor.

4 REVISÃO BIBLIOGRÁFICA

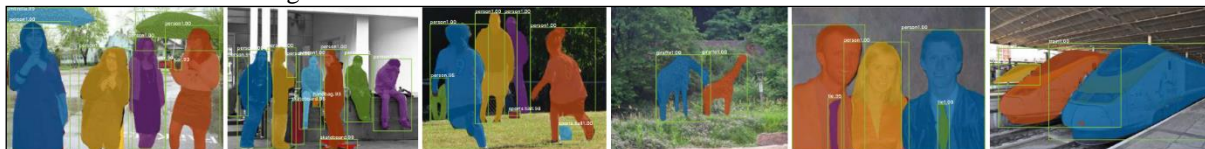
Este capítulo tem como objetivo apresentar uma introdução inicial ao tema principal deste projeto. A seção 4.1 aborda a detecção de objetos e o algoritmo Mask R-CNN sendo utilizado para segmentar e classificar as peças de dominó. A seção 4.2 aborda o Algoritmo Minimax que retornará à ação correspondente a melhor jogada naquele instante.

4.1 DETECÇÃO DE OBJETOS

Segundo Liu *et al.* (2019), detectar objetos em uma imagem é a base para aplicações que utilizam visão computacional. Consiste em determinar se existem quaisquer instâncias de objetos de determinadas categorias em uma imagem, caso haja, retornam à localização espacial e a *bounding-box*. Liu *et al.* (2019) também afirmam que a detecção de objetos é utilizada em várias atividades como: segmentação, compreensão de cenas, rastreamento de objetos, legenda de imagens, detecção de eventos e reconhecimento de atividades.

He *et al.* (2018) afirmam que um dos detectores de objetos existentes é o Mask R-CNN. Ele é uma extensão do detector *Faster R-CNN*, que é capaz de realizar a segmentação de objetos. Ainda segundo os autores, o *Faster R-CNN* é dividido em duas etapas, a primeira é chamada de *Region Proposal Network* (RPN), que coloca uma caixa delimitadora chamadas de *bounding-box* ao redor de possível candidatos a objetos. Na segunda, extrai-se recursos da caixa delimitadora utilizando *RoIPool*, realizando a classificação e a regressão da caixa delimitadora. Ao final do processamento será disponibilizado duas saídas para cada objeto processado, sendo elas, o rótulo e uma caixa delimitadora, conforme ilustra a Figura 8.

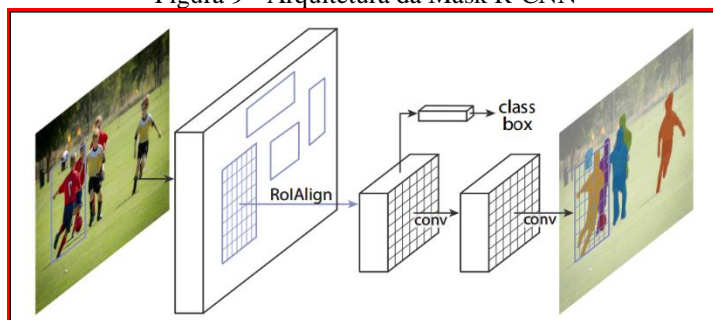
Figura 8 - Caixa delimitadora e rótulo do detector Mask R-CNN



Fonte: He *et al.* (2018).

O detector *Mask R-CNN* adota as duas etapas do detector *Faster R-CNN*, porém na segunda etapa além de realizar a predição da classe e disponibilizar a caixa delimitadora, em paralelo é disponibilizado uma máscara binária para cada região de interesse chamada de *RoI*. O detector *Mask R-CNN* foi criado com várias arquiteturas, sendo elas, a arquitetura de espinha dorsal chamado de *backbone* e a cabeça da rede. A espinha dorsal é uma rede neural convolucional, sendo utilizada para a extração de características da imagem por inteiro. Nesta arquitetura foi avaliado a ResNet e ResNeXt com profundidade de 50 ou 101 camadas, sendo aplicado uma *Feature Pyramid Network* (FPN), que utiliza uma arquitetura *top-down* com conexões laterais para construir uma pirâmide de recursos na rede. Permitindo um grande ganho na acurácia e na velocidade (HE *et al.*, 2018). A Figura 9 apresenta a arquitetura da Mask R-CNN.

Figura 9 - Arquitetura da Mask R-CNN



Fonte: He *et al.* (2018).

Segundo He *et al.* (2018), a cabeça da rede foi utilizada para o reconhecimento da caixa delimitadora e a previsão da máscara que é aplicada para cada região de interesse (classificação e regressão). Além disso, os autores também afirmam que o detector *Mask R-CNN* é considerado o estado da arte no quesito de segmentação de imagem e detecção de objeto com caixa delimitadora. O modelo unificado consegue simultaneamente prever a caixa delimitadora, realizar a segmentação e obter os pontos chave, executando um quadro em média a 200 milissegundos, desta forma, processando 5 quadros por segundo. A estrutura foi criada de forma a ser facilmente utilizada por outras pessoas (HE *et al.*, 2018).

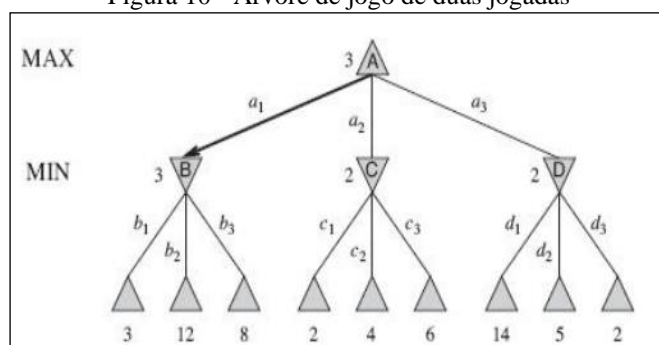
4.2 ALGORITMO MINIMAX

Segundo Russell e Norvig. (2013), o algoritmo Minimax retorna à ação correspondente ao melhor movimento possível, levando em consideração o que o oponente joga, tentando minimizar seus movimentos. Neste sentido, o algoritmo executa uma exploração completa em profundidade da árvore do jogo até o nó folha, propagando os valores minimax de volta pela árvore à medida que a recursão retorna. Os autores também afirmam que o número de estados de jogo que se examina é exponencial em relação ao número de movimentos.

Considerando um jogo que tenha dois jogadores, sendo chamados de MAX e MIN, MAX será o que fará o primeiro movimento, MIN será o que fará o segundo movimento, acontecendo um revezamento entre eles até que o jogo se encerre. Nota-se, que os jogos são expressos na forma de árvores, onde os nós são estados do jogo e as bordas são os movimentos. Por exemplo, a árvore do jogo da velha é relativamente pequena, sendo $9! = 362.880$ nós terminais, porém em jogos como xadrez, há mais de 10^{40} nós, dificultando sua representação computacional. Porém, independentemente do tamanho da árvore é responsabilidade do MAX encontrar uma boa jogada (RUSSELL; NORVIG, 2013).

De acordo com Russell e Norvig (2013), a solução ótima para um jogo, é uma sequência de jogadas que faz o jogador chegar a um estado terminal, representando uma vitória. Porém, MIN tem uma relação direta com este estado final. Desta forma, MAX deve encontrar uma estratégia que minimize os movimentos de MIN. Conforme mostra a Figura 10, os nós Δ são nós que MAX detém a vez para efetuar um movimento e nós ∇ são nós que MIN detém a vez para efetuar um movimento. Nós folhas (terminais) contém números que são os valores de utilidade do estado terminal. Os demais nós estão identificados com seus valores minimax.

Figura 10 - Árvore de jogo de duas jogadas



Fonte: Russell e Norvig (2013).

O valor minimax do nó B é definido através de uma análise de suas folhas. É analisado o valor de b_1 , b_2 e b_3 , que consecutivamente é 3, 12, 8. Destes valores é obtido o menor valor, que neste cenário é 3, ou seja, B recebe o valor minimax 3. O nó C por sua vez contém os valores 2, 4 e 6, sendo o menor valor 2. Desta forma, o valor minimax de é 2. Este processo é aplicado para todos os nós daquele nível. Analisando a Figura 10, a jogada que leva a um estado mais alto é o a_1 , porque o nó B contém um valor maior de minimax. Já a melhor jogada de MIN é b_1 , porque contém um valor menor de minimax (RUSSELL; NORVIG, 2013).

Segundo Russell e Norvig (2013), a estratégia ótima pode ser determinada utilizando o valor minimax de cada nó. Essa definição de jogo ótimo para MAX supõe que MIN também jogue de forma ótima, caso min não jogar de forma ótima, MAX terá um desempenho ainda maior. Ou seja, MAX preferirá realizar a jogada para um estado de valor máximo, MIN preferirá realizar a jogada para um estado de valor mínimo, por estes motivos o nome MIN e MAX dos jogadores.

De acordo com Russell e Norvig (2013), o algoritmo minimax é utilizado em jogos de dois jogadores com soma zero e informações perfeitas, selecionando movimentos ótimos. Usando uma enumeração da árvore do jogo em profundidade. O algoritmo pode ser utilizado em partidas de dois jogadores, como xadrez, jogo da velha, damas e gamão. Segundo Järvensivu (2018), que criou um jogo da velha utilizando o algoritmo minimax, foi possível obter o primeiro movimento em 2,75 segundos. Durante a partida demorava em média 8 a 10 milissegundos para processar e executar a jogada, no final da partida demorava em média 4 milissegundos. O autor destaca que o tempo necessário para processar e executar a jogada varia de acordo com a dificuldade escolhida.

REFERÊNCIAS

- ALIANE, Nouridine; BEMPOSTA, Sergio. **A Checkers Playing Robot: A Didactic Project**. Ieee Latin America Transactions, [S.l.], v. 9, n. 5, p. 821-826, set. 2011.
- BERNER, Christopher *et al.* **Dota 2 with Large Scale Deep Reinforcement Learning**. 2019.
- BROOKSHEAR, J. Glenn. **Ciência Da Computação: Uma Visão Abrangente**. 7. ed. Tradução Cheng Mei Lee. Higienópolis: Bookman, 2005.
- DARTIGUES, Jean François *et al.* **Playing board games, cognitive decline and dementia: a french population-based cohort study**. Bmj Open, [S.l.], v. 3, n. 8, p. 2998-3004, ago. 2013. BMJ. <http://dx.doi.org/10.1136/bmjopen-2013-002998>.
- HE, Kaiming *et al.* **Mask R-CNN**. Ieee Transactions On Pattern Analysis And Machine Intelligence, [S.l.], v. 42, n. 2, p. 2961 - 2969, 24 jan. 2018.
- HSU, Feng-Hsiung. **IBM's Deep Blue Chess grandmaster chips**. Ieee Micro, [S.l.], v. 19, n. 2, p. 70-81, 1999.
- JAIN, Sonal. **An intelligent method for solving TIC-TAC-TOE problem**. International Conference On Computing, Communication & Automation, [S.l.], p. 181-184, maio 2015.
- JÄRVENSIVU, Matti. **Developing and Optimizing Artificial Intelligence in Zero-sum Games**. JAMK University of Applied Sciences, maio 2018.
- LIU, Li *et al.* **Deep Learning for Generic Object Detection: a survey**. International Journal Of Computer Vision, [S.l.], v. 128, n. 2, p. 261-318, 31 out. 2019.
- RUSSELL, Stuart; NORVIG, Petter. **Inteligência Artificial**. 3.ed. Tradução Regina Célia Simille de Macedo. Rio de Janeiro: Elsevier, 2013.
- SELENGUINI, Anderson L. D. S.; AMARAL, Gabrielv A. **Sistema Autonomo para Jogar Xadrez, usando Robix e Morfologia Matemática**. 2016. 6f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade Federal de Alfenas, Alfenas.

TORO, Cristian D.; ROBLES-ALGARÍN, Carlos; RODRÍGUEZ-ÁLVAREZ, Omar. **Design and Construction of a Cost-Effective Didactic Robotic Arm for Playing Chess, Using an Artificial Vision System**. Electronics, [S.l.], v. 8, n. 10, p. 1154-1169, 12 out. 2019.

TURING, Alan M. **Chess**. Computer Chess Compendium, [S.l.], p. 14-17, 1988.

YANNAKAKIS, Georgios N.; TOGELIUS, Julian. **Artificial Intelligence and Games**. Charm: Springer, 2018.

ASSINATURAS

(Atenção: todas as folhas devem estar rubricadas)

Assinatura do(a) Aluno(a): _____

Assinatura do(a) Orientador(a): _____

Assinatura do(a) Coorientador(a) (se houver): _____

Observações do orientador em relação a itens não atendidos do pré-projeto (se houver):

FORMULÁRIO DE AVALIAÇÃO – PROFESSOR AVALIADOR

Acadêmico(a): Francisco Lucas Sens

Avaliador(a): Everaldo Artur Grahl

ASPECTOS AVALIADOS ¹		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?	X		
	O problema está claramente formulado?	X		
	1. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?	X		
	Os objetivos específicos são coerentes com o objetivo principal?	X		
	2. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?	X		
	3. JUSTIFICATIVA Foi apresentado e discutido um quadro relacionando os trabalhos correlatos e suas principais funcionalidades com a proposta apresentada?	X		
	São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?	X		
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?	X		
	4. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?	X		
	5. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?	X		
	Os métodos, recursos e o cronograma estão devidamente apresentados e são compatíveis com a metodologia proposta?	X		
	6. REVISÃO BIBLIOGRÁFICA (atenção para a diferença de conteúdo entre projeto e pré-projeto) Os assuntos apresentados são suficientes e têm relação com o tema do TCC?	X		
	As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?	X		
ASPECTOS METODOLÓGICOS	7. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?	X		
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?	X		

PARECER – PROFESSOR AVALIADOR: (PREENCHER APENAS NO PROJETO)

O projeto de TCC ser deverá ser revisado, isto é, necessita de complementação, se:

- qualquer um dos itens tiver resposta NÃO ATENDE;
- pelo menos **5 (cinco)** tiverem resposta ATENDE PARCIALMENTE.

PARECER:

(X) APROVADO

() REPROVADO

Assinatura: Everaldo Artur Grahl Data: 06/12/2020

¹ Quando o avaliador marcar algum item como atende parcialmente ou não atende, deve obrigatoriamente indicar os motivos no texto, para que o aluno saiba o porquê da avaliação.