



UNIVERSIDADE
ESTADUAL DE LONDRINA

Relatório do Projeto de Inteligência Artificial aplicada ao “T-Rex Game” (Google Chrome) - Feira de Profissões UEL 2025

Alef Vinícius Caires Carvalho – 202501220013 - alef.v.caires.carvalho@uel.br

Andrew Willian Freitas – 202501220024 - andrew.willian@uel.br

Emilly Vitória Zamperlini - 202501220319 - emilly.vitoria0@uel.br

Isabela Ribeiro de Oliveira – 202501220660 - bela.forastieri@uel.br

Kauã Felipe Martins – 202501220159 - kaua.felipe.martins@uel.br

Leonardo Madeira Alves Pereira – 202501220488 - leonardo.madeira@uel.br

Matheus Felipe Aguiar Barbosa – 202501220193 - matheus.barbosa@uel.br

Professor orientador: Prof. Dr. Bruno Squizzato Façal

Departamento de Computação – Curso de Ciência de Dados e Inteligência Artificial –
Universidade Estadual de Londrina (UEL)
Caixa Postal 10.011 - 86.057-970 - Londrina – PR – Brasil

Abstract. *The project presents the development of an artificial intelligence solution for the “T-Rex Chrome Dino” game, created by students from the Data Science and Artificial Intelligence program at UEL for the UEL Career Fair 2025. Initially, the goal was to create an autonomous agent using reinforcement learning, but the focus shifted to an interactive approach: a computer vision system that uses a webcam to capture human movements (jump or not jump) and translate them into game commands, promoting greater audience engagement. The image database was expanded using data augmentation techniques, and the final solution employs a neural network implemented in TensorFlow, ensuring accuracy and adaptability in dynamic environments. The project demonstrates the practical integration of AI and computer vision in digital games, with potential for educational and interactive applications at scientific outreach events.*

Keywords: *artificial intelligence; computer vision; digital games; neural networks; human-computer interaction.*

Resumo. *O projeto apresenta o desenvolvimento de uma solução de inteligência artificial para o jogo “T-Rex Chrome Dino”, criada pelos alunos do curso de Ciência de Dados e Inteligência Artificial da UEL para a Feira de Profissões UEL 2025. Inicialmente, buscou-se criar um agente autônomo por aprendizado por reforço, mas o foco evoluiu para uma abordagem interativa: um sistema de visão computacional que utiliza webcam para captar movimentos humanos (pulo ou não pulo) e traduzi-los em comandos no jogo, promovendo maior engajamento do público. O banco de dados de imagens foi expandido com técnicas de data augmentation, e a solução final utiliza uma rede neural em TensorFlow, garantindo precisão e adaptabilidade em ambientes dinâmicos. O projeto demonstra a integração prática de IA e visão computacional em jogos digitais, com potencial para aplicações educacionais e interativas em eventos de divulgação científica.*

Palavras-chave: *inteligência artificial; visão computacional; jogos digitais; redes neurais; interação humano-máquina.*

1. Introdução

O jogo "T-Rex Chrome Dino" é um jogo simples embutido no navegador Google Chrome, criado em 2014 pelos desenvolvedores da Google: Sebastien Gabriel, Alan Bettis e Edward Jung. O *game* aparece automaticamente quando o usuário tenta acessar uma página da internet estando *offline*, ocasião em que é exibido um dinossauro pixelado na tela. Ao pressionar a barra de espaço ou a tecla de seta para cima (ou tocar no dinossauro em dispositivos móveis), o jogo inicia.

No jogo, o usuário controla um tiranossauro que corre automaticamente da esquerda para a direita em um cenário desértico em preto e branco. O objetivo é evitar obstáculos, como cactos (obstáculos no solo) e pterossauros (obstáculos no ar), pulando ou agachando-se para sobreviver o máximo possível. Conforme o jogo avança, a velocidade aumenta gradativamente, tornando o desafio mais difícil até que o dinossauro colida com um obstáculo, encerrando a partida. O jogo também alterna entre modo dia e noite, com mudanças visuais que acompanham o progresso do jogador. A simplicidade e o caráter de "infinite runner" fazem do T-Rex Chrome Dino um clássico entretenimento e distração para momentos sem conexão à internet.

Por se tratar de um jogo de mecânica simples, de fácil acesso/implementação e amplamente disseminado entre o público, o T-Rex Chrome Dino foi escolhido pelo grupo de alunos do primeiro ano e professores orientadores do Curso de Ciência de Dados e Inteligência Artificial da Universidade Estadual de Londrina como o projeto discente a ser apresentado à comunidade na 13ª Feira de Profissões da UEL no ano de 2025.

2. Referências iniciais e primeiras considerações.

A ideia inicial do trabalho consistia em implementar um modelo de *machine learning* capaz de aprender a jogá-lo com base em instruções simples. A implementação inicial teve inspiração no vídeo postado pelo desenvolvedor

Saral Tayal na plataforma YouTube¹ e seu respectivo repositório no site GitHub² em que o computador aprende jogar por meio de técnicas de aprendizado por reforço, especificamente o algoritmo Q-Learning, sem qualquer modificação no código original do jogo. No caso citado, o agente de inteligência artificial interage diretamente com a versão real do Chrome Dino, capturando imagens da tela para analisar o ambiente e tomar decisões. Essas imagens são processadas, reduzidas em resolução e convertidas para preto e branco, facilitando a identificação dos obstáculos e do cenário.

O algoritmo de aprendizado por reforço que permite que o agente aprenda a tomar decisões maximizando as recompensas ao longo do tempo, sem precisar de um modelo prévio do ambiente. O Q-Learning constrói uma tabela de valores (Q-table) que representa a qualidade de realizar cada ação em cada estado. À medida que o agente joga, ele atualiza esses valores de acordo com as recompensas recebidas, aprendendo por tentativa e erro quais ações são mais vantajosas em cada situação.

No projeto do Chrome Dino analisado, o agente recebe como entrada uma sequência de quatro frames consecutivos, o que permite compreender o movimento e a velocidade dos obstáculos. Com base nessas informações, o agente decide entre pular, agachar ou não fazer nada.

O processo de aprendizado é guiado por recompensas: o agente recebe pontos positivos ao sobreviver e negativos ao colidir com obstáculos. As experiências vividas durante as partidas são armazenadas em uma memória chamada “experience replay”, permitindo que o agente revise situações passadas para aprimorar sua tomada de decisão futura.

¹ TAYAL, Saral. **AI masters the Chrome Dino Game | Reinforcement Learning Tutorial**. YouTube, 4 jul. 2020. Disponível em: <https://www.youtube.com/watch?v=WMYG6IEgMfw>. Acesso em: 30 jun. 2025.

² _____. **ChromeDinoAI: This repo complements the YouTube video on how to beat the Chrome Dino game using RL AI techniques**. 2020. Disponível em: <https://github.com/SaralTayal123/ChromeDinoAI>. Acesso em: 30 jun. 2025

Com o tempo, a IA aprende as melhores estratégias para avançar no jogo, tornando-se capaz de superar desafios cada vez mais complexos e alcançar maiores scores.

Ocorre que, por se tratar de um projeto a ser apresentado em uma Feira, seria interessante que os alunos interessados a ingressar no Curso de Ciência de Dados e Inteligência Artificial (e demais transeuntes) tivessem uma verdadeira conexão com o trabalho apresentado. Dessa forma, buscou-se por outras soluções que possibilitassem as pessoas a interagirem com o *game*.

Em novas pesquisas, sugeriu-se como inspiração o projeto criado pelo desenvolvedor João Tinti e demonstrado em seu canal do YouTube³ e em seu repositório⁴. Nele, criou-se uma solução em que o algoritmo de aprendizado supervisionado captura imagens do jogador humano por meio de uma webcam e, em tempo real, decide se ele está pulando (ou não) para transmitir o comando ao dinossauro do game. O código foi escrito em linguagem Python e utilizou uma a biblioteca Scikit-Learn para visão computacional e machine learning para treinar o modelo.

Além disso, o projeto utilizou as bibliotecas PyGame para recriar o jogo localmente e a OpenCV capturar as fotografias pela webcam e processá-las (reduzindo formatos, dimensões, cores etc.).

Para interagir com o jogo, o código criado simula pressionamentos de teclas no sistema operacional. Quando o algoritmo captura uma imagem do jogador humano pulando e compara a imagem carregada no seu modelo “jumping”, ele envia automaticamente o comando de pulo (barra de espaço), fazendo com que o personagem salte no momento certo. Todo esse processo é

³ TINTI, João. **DINO do CHROME e Machine Learning! Jogando sem as mãos com Python!** YouTube, 12 out. 2020. Disponível em: <https://www.youtube.com/watch?v=sIV0hbIJGgk>. Acesso em: 30 jun. 2025.

⁴ TINTI, João Vitor. **DINO GAME**. Disponível em: https://github.com/joaotinti75/Pygame/blob/master/Chrome_Dinosaur_Game/DINO_GAME.ipynb. Acesso em: 30 jun. 2025.

feito em tempo real, demonstrando a aplicação prática de visão computacional e de machine learning para resolver desafios simples de jogos digitais.

Em resumo, a IA não aprende jogando (*lazy algorithm*), pois a sua função é somente automatizar a jogabilidade ao reconhecer visualmente o ato do jogador “pulando”, utilizando bibliotecas de visão computacional.

Ocorre que, muito embora o projeto já tenha sido criado, a versão disponibilizada pelo autor possuía diversas limitações, visto que não contava com um script otimizado de captura de imagens para treinamento do modelo. Afinal, o projeto somente tirava fotos em intervalos regulares de tempo, pouco importando se o jogador estava pulando (subida, descida) ou se movimentando aleatoriamente. Essa limitação acabava por dificultar o treinamento do modelo e a melhoria da acurácia nos testes, já que a separação e tratamento das fotografias deveria ocorrer manualmente, tal como realizado pelo próprio autor.

Além do mais, o modelo não aprendeu a jogar o jogo original, mas tão somente uma versão local criada pelo próprio autor por meio biblioteca PyGame, o que tornou a experiência menos convidativa e desafiadora, já que não jogava o jogo real incorporado ao navegador Google Chrome.

Por fim, o script criado não permitia a sua aplicação a diversos jogadores ou a treinamento com ruídos ou situações não homogêneas (exemplo: detalhes em paredes ao fundo), tal como ocorreria no caso do ambiente real da Feira de Profissões, o que tornou necessário reinventá-lo, conforme será exposto no tópico a seguir.

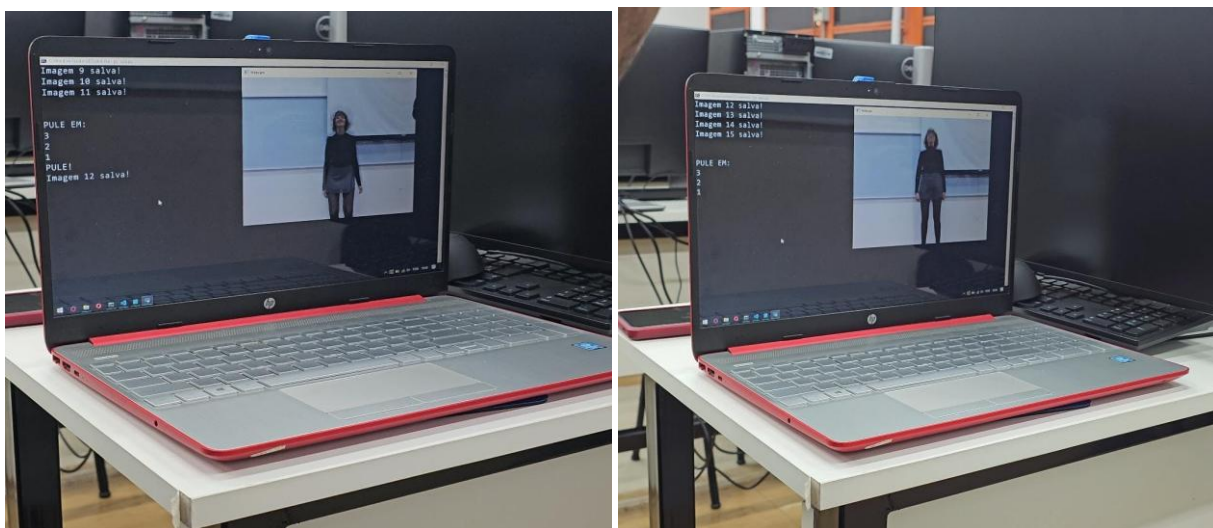
3. Da criação à implementação do algoritmo.

Com as referências para o projeto, passamos à fase de implementação. Em primeiro lugar, foi criado o repositório no GitHub⁵ para se possibilitar a construção do projeto e escolhida a linguagem Python para sua implementação.

⁵ FREITAS, Andrew Willian; MARTINS, Kauã Felipe; PEREIRA, Leonardo Madeira Alves. **Repositório do projeto da introdução do curso Ciência de Dados e Inteligência Artificial no evento na 13ª Feira de Profissões da Universidade Estadual de Londrina**. 2025. Disponível em: <https://github.com/wllandrew/projeto-feira-de-profissoes-uel>. Acesso em: 30 jun. 2025.

Logo após, foi necessário criar um robusto banco de dados com várias fotografias com vários alunos no grupo, a fim de se obter dados diversos e enriquecer o processo de aprendizado do algoritmo.

Para tanto, foi criado um script autoral de “Frames Capture”, que tirava fotos automaticamente e já as separava de acordo com cada rótulo em duas pastas automaticamente: jumping (“pulando”) ou no-jumping (“não pulando”). A ideia foi implementar uma contagem regressiva no script, a fim de conciliar o momento da foto ao pulo do usuário humano. Assim, foi possível otimizar o processo de captura de imagens. Veja-se, como exemplo, fotografias do dia do treinamento:



Fotos 1 e 2 – Fotos tiradas do momento do script “Frames Capture” capturando fotos do usuário modelo pulando e não pulando.

Ocorre que, como somente foram tiradas fotografias com ruídos ao fundo, o modelo ficou viciado e não foi possível aumentar a sua acurácia nos treinamentos posteriores, pelo que foi necessário obter novas fotografias mais “limpas” (seja na ausência de ruídos, seja na clareza do que seria um “pulo” e o que seria um “não-pulo”).

Para aumentar a variabilidade dos dados, foi utilizada a biblioteca Albumentations, que é uma ferramenta voltada para aumento de imagens (*data augmentation*) voltada para projetos de visão computacional e aprendizado profundo. Ela permite aplicar rapidamente uma grande variedade de transformações em imagens, como rotações, flips, redimensionamentos,

alterações de brilho e contraste, adição de ruído, entre outras técnicas criando variações realistas das imagens originais para tornar os modelos de IA mais robustos e capazes de generalizar melhor.

Entretanto, com o aumento do banco de fotografias, o tempo para treinamento do modelo aumenta exponencialmente, o que torna custosa e demorada a apuração do sucesso da implementação (acurácia). Além disso, os testes evidenciados ficaram pesados, o que dificultava a implementação do modelo em máquinas com potencial de processamento baixo ou reduzido.

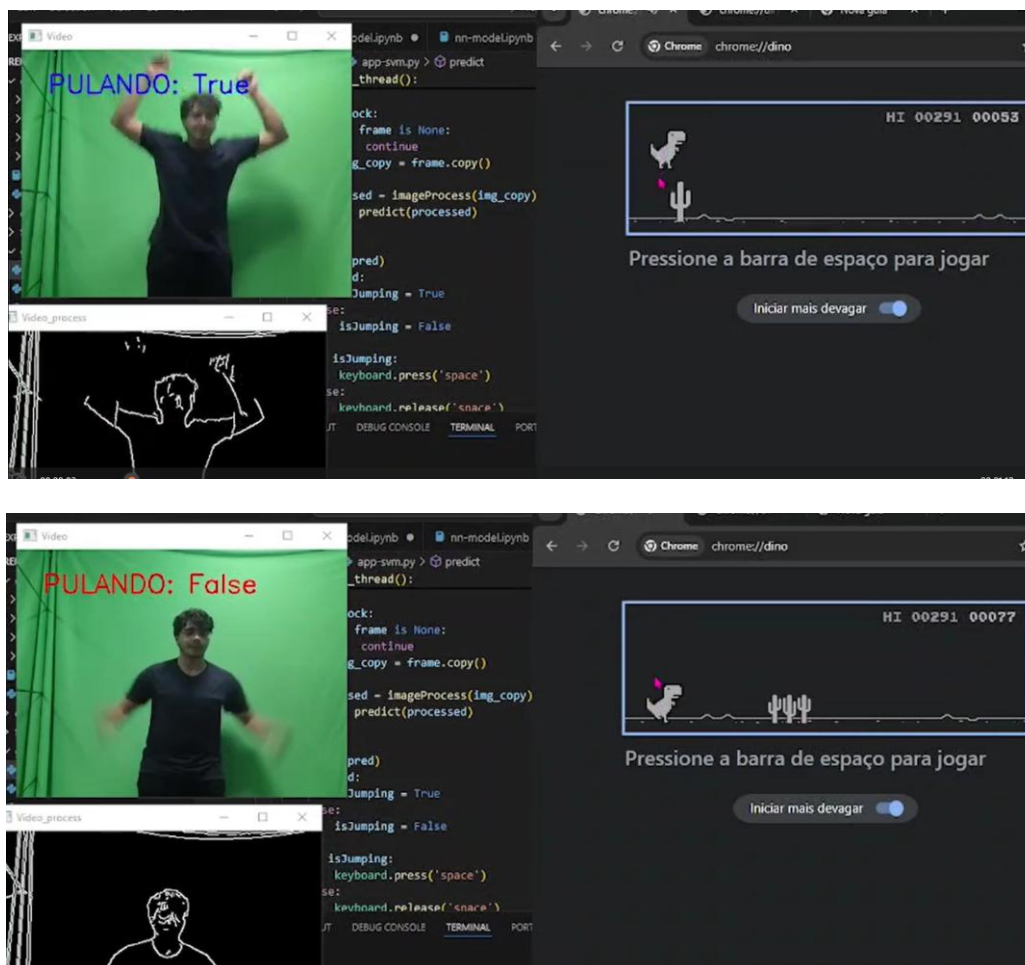
Necessário foi, então, otimizar o modelo tanto em tamanho absoluto quanto em eficiência, uma vez que o processo de captura das imagens já era satisfatório. Sendo assim, optou-se – em primeiro lugar - que a implementação do modelo deveria retornar ao modelo inicialmente proposto, isto é, no jogo do Navegador (e não localmente).

Sendo assim, foi alterado o modelo de SVM (Grid Search) para um modelo de rede neural (TensorFlow). A diferença fundamental entre o método SVM (Support Vector Machine) inicialmente testado e as redes neurais (como as implementadas no TensorFlow) está na abordagem e arquitetura.

O SVM é um algoritmo supervisionado clássico, utilizado principalmente para classificação e regressão. Ele funciona buscando um hiperplano que maximize a margem entre diferentes classes no espaço dos dados. Caso os dados não sejam linearmente separáveis, o SVM utiliza funções de kernel para projetar os dados em espaços de maior dimensão, onde a separação pode ser possível. O SVM é eficiente em conjuntos de dados menores, mais estruturados e quando se deseja maior interpretabilidade do modelo, pois sua decisão é baseada em poucos pontos críticos chamados vetores de suporte. Isso significa que o modelo não era adequado para lidar com uma grande variabilidade de dados, o que dificultaria o acesso de diferentes jogadores e a garantia da acurácia do projeto.

Já as redes neurais implementadas no TensorFlow, são compostas por múltiplas camadas de neurônios artificiais que aprendem representações hierárquicas dos dados. Elas são extremamente flexíveis e capazes de lidar com

grandes volumes de dados e dados não estruturados, como imagens, áudio e texto. Redes neurais profundas conseguem aprender padrões complexos e realizar tarefas como reconhecimento de imagens e processamento de linguagem natural, mas exigem mais dados para treinamento e maior poder computacional e tendem a ser menos interpretáveis, funcionando como "caixas-pretas". O modelo final ficou, graficamente, da seguinte forma:



Fotos 3 e 4 – Mostram o modelo já treinado compreendendo quando o jogador está pulando (PULANDO: True) e quando não está pulando (PULANDO: False). No canto superior esquerdo é possível visualizar a imagem original captada pela webcam; no canto inferior esquerdo, a imagem já pré-processada que será interpretada pelo algoritmo; e, do lado direito, o game rodando no Browser.

Maiores informações acerca da documentação e funcionamento do modelo podem ser obtidas na documentação oficial do projeto no GitHub indicada na referência do presente trabalho.

4. Conclusão.

O projeto de inteligência artificial aplicada ao jogo “T-Rex Chrome Dino” alcançou resultados expressivos ao integrar técnicas modernas de visão computacional e redes neurais para criar uma experiência interativa e acessível. A transição do enfoque inicial em agentes autônomos para um sistema que utiliza a webcam para captar os movimentos humanos permitiu maior engajamento dos visitantes, especialmente em ambientes dinâmicos como a Feira de Profissões da UEL. O uso de *data augmentation* e a implementação em TensorFlow garantiram robustez ao modelo, tornando-o adaptável a diferentes usuários e condições de uso.

A importância deste projeto para a Feira de Profissões está em sua capacidade de aproximar estudantes do universo da ciência de dados e inteligência artificial de forma prática e lúdica. Ao proporcionar uma experiência interativa, o projeto não só desperta o interesse dos visitantes, mas também demonstra, na prática, como conceitos avançados podem ser aplicados em soluções reais e acessíveis.

Esse tipo de iniciativa é fundamental em eventos de divulgação científica, pois permite que futuros estudantes visualizem o potencial transformador da tecnologia e compreendam melhor o cotidiano e as possibilidades do curso, contribuindo para escolhas profissionais mais informadas e conscientes.

5. Referências Bibliográficas.

FREITAS, Andrew Willian; MARTINS, Kauã Felipe; PEREIRA, Leonardo Madeira Alves. **Repositório do projeto da introdução do curso Ciência de Dados e Inteligência Artificial no evento na 13ª Feira de Profissões da Universidade Estadual de Londrina**. 2025. Disponível em: <https://github.com/wllandrew/projeto-feira-de-profissoes-uel>. Acesso em: 30 jun. 2025.

TAYAL, Saral. **AI masters the Chrome Dino Game | Reinforcement Learning Tutorial**. YouTube, 4 jul. 2020. Disponível em: <https://www.youtube.com/watch?v=WMYG6IEgMfw>. Acesso em: 30 jun. 2025.

_____. **ChromeDinoAI: This repo complements the YouTube video on how to beat the Chrome Dino game using RL AI techniques**. 2020. Disponível em: <https://github.com/SaraTayal123/ChromeDinoAI>. Acesso em: 30 jun. 2025

TINTI, João Vitor. **DINO GAME**. Disponível em: https://github.com/joaotinti75/Pygame/blob/master/Chrome_Dinosaur_Game/DINO_GAME.ipynb. Acesso em: 30 jun. 2025.

_____. **DINO do CHROME e Machine Learning! Jogando sem as mãos com Python!** YouTube, 12 out. 2020. Disponível em: <https://www.youtube.com/watch?v=sIV0hbIJGgk>. Acesso em: 30 jun. 2025.