

# Inteligencia Artificial Para Reconhecimento de Jogos Digitais Através de Imagens

Neuremberg V. P. Junior

<sup>1</sup>Instituto CEV (ICEV)

Rua Dr. José Auto de Abreu, 2929 - São Cristovão - 64055-260 - Teresina-PI

{neuremberg.junior}@somosicev.com

**Abstract.** *This meta-paper describes the style to be used in articles and short papers for SBC conferences. For papers in English, you should add just an abstract while for the papers in Portuguese, we also ask for an abstract in Portuguese (“resumo”). In both cases, abstracts should not have more than 10 lines and must be in the first page of the paper.*

**Resumo.** *Este meta-artigo descreve o estilo a ser usado na confecção de artigos e resumos de artigos para publicação nos anais das conferências organizadas pela SBC. É solicitada a escrita de resumo e abstract apenas para os artigos escritos em português. Artigos em inglês deverão apresentar apenas abstract. Nos dois casos, o autor deve tomar cuidado para que o resumo (e o abstract) não ultrapassem 10 linhas cada, sendo que ambos devem estar na primeira página do artigo.*

## 1. Introdução

Como vivemos em uma era de internet é normal acreditar que é possível conhecer tudo, porém, não é bem assim, atualmente as coisas são sempre bem rápidas e cada minuto é precioso e onde que isso está inserido no contexto de jogos ? A quantidade de mídia digital é gigantesca, e as vezes coisas que despertam a curiosidade são visualizadas, dentre elas os jogos. O publico mais jovem está acostumado a ver imagens, prints e video clips de jogos digitais o tempo todo, mas em grande parte das vezes vem o desejo de saber que jogo é aquele já que na maioria dos casos não possui a fonte do video ou do print e ai que percebe-se que não existe uma plataforma focada nesse objetivo, descobrir qual o nome ou aonde é aquele jogo de uma forma simples pratica e eficaz. Talvez seja mais simples só pesquisar no Google ? É uma solução aparentemente obvia, mas quando existir um lugar que já oferece todas as informações desejadas sem a necessidade de ir em sites adicionais já adiciona muito mais conforto a toda a experiencia.

## 2. Ideia de Solução

Não é de hoje que a inteligencia artificial e os jogos estão conectados, na verdade esse ambiente acabou sendo muito bom para o desenvolvimentos de novas tecnologias, como a logica Fuzzy. Pensando nesse problema, existe a ideia de uma plataforma personalizada na qual iria reconhecer qual jogo seria aquele que o usuário deseja saber, utilizando usando a técnica de reconhecimento de imagens a partir de uma Rede Neural Convolucional (CNN) que consiste em um sistema de neurônios artificiais interconectados que trocam mensagens entre si. As conexões tem pesos numéricos que são sintonizados durante o processo

de treinamento, para qual uma rede devidamente treinada responde corretamente quando apresentado à uma imagem ou padrão para analisar. A rede consiste em varias camadas de neurônios de detecção de recursos, cada camada tem muitos neurônios que respondem a diferentes combinações de entradas das camadas anteriores[Hijazi et al. 2015], por exemplo, é possível identificar o jogo por elementos na imagem como cores e padrões, analisando pixel por pixel, através disso é possível treinar uma inteligencia artificial que através de um print, uma captura de tela do meio da gameplay do jogo, e através dela vai ser possível fazer a distinção de certos elementos na hora do treinamento como a interface de itens, barra de vida ou menu de opções, oferecendo assim imediatamente a qual jogos aqueles elementos pertencem. Isso é mais complicado de ser feito quando é realizado o print de uma Cutscene, ou seja, uma cinemática que ocorre durante o jogo, pois nesses casos não é apresentados elementos capazes de serem aprendidos e modificados já que tudo muda o tempo o todo e nada é constante, ocorre casos de uma Cutscene se passar em um cenário que só aparece durante 10 segundos durante todo o jogo, isso acabaria causando um ruído, informações que são inúteis para o treinamento e também atrapalham os resultados, a I.A. é limitada nesses casos envolvendo Cutscenes, por esse motivo, o foco é realmente em elementos presentes na gameplay em si. Também serão feitas tentativas de se utilizar os modelos YOLO que é um método de detecção de objetos de passada única (single pass) que utiliza uma rede neural convolucional como extrator de características (features). Diferente de algoritmos anteriores de detecção de objetos, como R-CNN ou Faster R-CNN, ele apenas precisa olhar pela imagem uma única vez para enviar para a rede neural. Por isso ele recebe esse nome (You Only Look Once – “Você só olha uma vez”).E devido a essa característica, o YOLO foi capaz de conseguir uma velocidade na detecção muito maior do que as técnicas concorrentes, sem perder em acurácia vai ser usado para reconhecimentos de elementos na imagem, como pessoas ou animais, no caso do game, procurar identificar o personagem na tela caso o jogo seja em terceira pessoa como em Fortnite. E um modelo U-Net que é uma arquitetura de rede convolucional para segmentação rápida e precisa de imagens. Essas técnicas de Deep Learning vão ser utilizadas pois são as que mais se encaixam em reconhecimentos de imagens no momento e com o tema.

Fonte:<https://iaexpert.academy/2020/10/13/deteccao-de-objetos-com-yolo-uma-abordagem-moderna/>

### **3. Seus experimentos, incluindo a metodologia utilizada para avaliação**

Os experimentos iniciados foram com uma CNN usando 2 datasets com 1000 imagens cada, foi separado mais ou menos 500 imagens para cada, um continha imagens para o jogo Minecraft e o outro para o jogo Among Us, o primeiro passo foi fazer o sistema reconhecer os datasets e validar o treinamento o que não deu certo, pois estava apresentando um fenômeno no qual após apenas 2 dos 10 treinamentos já apresentava 100 por cento de acurácia e 0 de loss, comportamento que sugere que o modelo estava sofrendo overfitting, pois não foram instalados camadas suficientes para um treinamento mais preciso, então foi utilizado uma abordagem qualitativa, aplicando uma função que iria plotar a performance do modelo no dataset de treinamento e validação durante o treinamento e depois representá-la em formato de gráficos, após isso foi adicionado uma camada de dropout, que age desligando alguns parâmetros durante determinada fase do treinamento para equilibrar a relevância de todos os nós, foi aplicado também a data augmentation, que irá realizar

pequenas alterações nas features como rotacionar ou dar um zoom por exemplo, aumentando a quantidade de imagens sem precisar de mais imagens, a aplicação do modelo MobileNETV2 que é utilizado em classificação de imagens, é baseada em uma estrutura residual invertida onde as conexões residuais são entre o "gargalo" da camada, ou seja, no ponto onde elas mais se reduzem. A camada de expansão intermediária usa convoluções de profundidade leve para filtrar recursos como uma fonte de não-linearidade. Depois da implantação desses modelos o treinamento foi feito novamente com 20 epochs, o resultado final foi bem mais satisfatório com um accuracy de: 0.9847 e um loss de : 0.1337, de 32 predições o modelo acertou 29, porém só foi treinada por poucas rodadas então mais testes eram necessários e a possibilidade do overfitting ainda está ocorrendo não podia ser ignorada.

Foi feito então o teste com 100 epochs de para observar que os resultados, os mesmos modelos foram usados e mesmas imagens, os resultados se mostraram satisfatórios, a accuracy foi de : 0.9959 e um loss de 0.0627 já que de 100 previsões esse modelo errou apenas 3, tornando ele assim com mais ou menos 97 por cento de acurácia, mas novamente a possibilidade do overfitting ainda pode ser real mesmo depois de usar vários modelos para preveni-lo.

Logo veio a ideia de experimentar como o modelo iria reagir quando fosse apresentado a todas as imagens do dataset e como já era de se esperar os resultados foram catastróficos, com um numero de 100 epochs, além de demorar mais de 5 minutos para processar cada rodada a acurácia não aumentava, pelo contrario ficava estática em 0.1005 , enquanto pelo contrario o loss só aumentava chegando a incríveis -525.0552, então ficou claro que esse modelo não iria suportar uma quantidade tão grande de dados com poucos filtros e pouco numero de epochs, então foi interrompido o treinamento.

Então o mesmo modelo foi usado para testar com apenas 4 jogos, God of War, Roblox, Terraria e Fortnite, uma teoria surgiu e através desse teste e do próximo ela poderia ser comprovada. O resultado desse teste feito com 4 jogos também foi horrível, com a acurácia não subindo a um valor aceitável, então a teoria veio a tona de que: Talvez jogos com imagens muito diferentes que não possuam um padrão estivessem causando um efeito de ruído tão grande que o treinamento todo era prejudicado, os 2 primeiros jogos testados foram Among Us e Minecraft, que possuem um padrão grande de gameplay só variando a cor dos lugares e coisas desse tipo, já jogos como God of War, podem possuir 70 imagens diferentes, apresentados cutscenes por exemplo, não contribuindo nada para o treinamento e além disso ainda estava atrapalhando.

Para comprovar isso um teste foi feito com 100 epochs com o dataset dos jogos Terraria e Fortnite que apresentam um padrão em sua gameplay, não variando muito com relação a coisas acontecendo na tela, e os resultados se mostraram bons, porém com uma acurácia muito elevada levando a crer que o modelo sofreu overfitting. Já com os dados de jogos como God of War e Genshin Impact que possuem muitas cenas diferentes e sem relação entre si o treinamento apresenta resultados abaixo da média com loss alto.

Os modelos de Yolo não deu para ser implementado pois requer que as imagens já possuam algumas identificações, como identificador de onde estão os objetos relevantes para o treinamento, o marcando com uma "box", coisa que o dataset utilizado não possui, e um treinamento específico nesse modelo requer mais tempo.

Depois foi feito a tentativa de usar um modelo Sequencial com 3 camadas de dropout, para analisar e treinar o Dataset contendo os jogos Minecraft e Terraria, os teste resultaram em o modelo provavelmente sofrendo de overfitting com os valores de acurácia resultando em : accuracy: 1.0000 e loss: 7.2923e-04

#### **4. Conclusão**

O modelo CNN feito é bom para identificação de imagens no contexto de jogos quando o mesmo não apresenta muitas diferenças em seu total, jogos sem cutscenes ou menus tendem a apresentar resultados melhores, já que a análise da matrix se torna mais uma análise de cores de padrões, já em jogos como God Of War, que apresenta muita cutscenes e muitas cenas não funciona muito bem, e apresenta um loss bem alto, mas nada 100 por cento confirmado ainda. É sim possível criar uma Inteligencia Artificial para reconhecer o jogos do Print, porem tem varias condições como a presença de cutscenes e em casos de jogos que não possuam um grande dataset para o treinamento bem efetivo, acaba tornando uma tarefa bem mais complicada, existem ainda jogos tão antigos que não possuem muitos registros logo, não possuem muitas imagens e se possuem é em uma resolução baixa. Então deve ser realizado muita pesquisa antes de tentar algo nessa escala.

#### **Referências**

Hijazi, S., Kumar, R., Rowen, C., et al. (2015). Using convolutional neural networks for image recognition. *Cadence Design Systems Inc.: San Jose, CA, USA*, 9.