



# Detecção de objetos em um jogo em tempo real.

## Robson Zagre Jr. e Nancy Baygorrea

## Introdução

Com a evolução das tecnologias, novos gráficos foram sendo implementados na indústria de vídeo games, muitos dos quais buscaram representar a realidade de forma mais fiel. Nesse sentido, uma questão interessante que aparece é se há a possibilidade de, por meio de técnicas clássicas e/ou deep learn, detectar objetos em tempo real os quais foram treinados baseados realidade. Com o intuito de responder isso, e na tentativa de generalizar a realidade mediante um jogo, foi implementado uma abordagem clássica para detecção de carro a qual pode ser comparada com uma YOLO.

## Metodologia

### Clássica:

Para a abordagem clássica, nos baseamos no trabalho desenvolvido pelo **changsub**. Utilizando um dataset com labels de veículos e não veículos em imagens de 64x64, foi desenvolvido features referentes a histogramas dos canais de cores e informações de HOG. Essas features foram então utilizadas para se treinar um SGD Classifier o qual foi utilizado no projeto por permitir treinamento online.

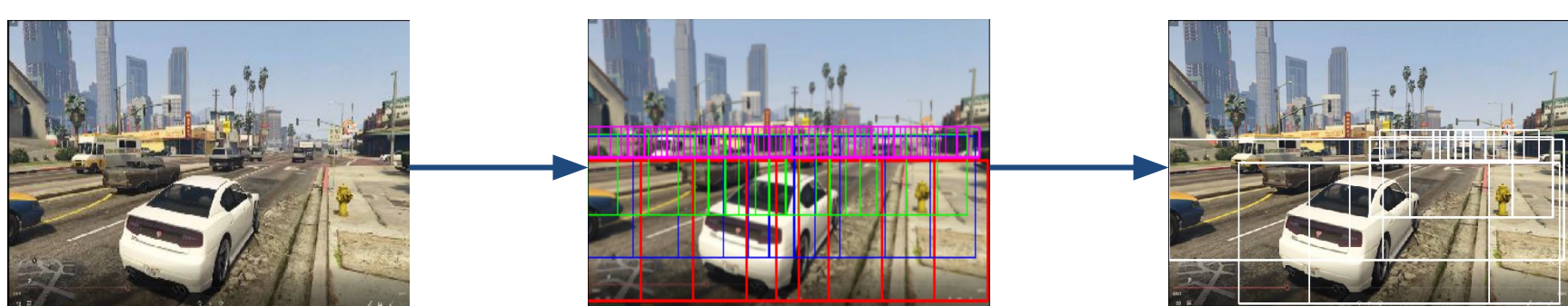
A fim de encontrar os melhores parâmetros para o pipeline da abordagem clássica, foi desenvolvido um CustomSGDClassifier a fim de introduzir tais variáveis para o modelo, dessa forma, conseguimos utilizar o Grid Search, técnica que procura por melhores hiperparâmetros, para encontrar os valores que melhor apresentaram a melhor acurácia.

Nosso algoritmo consegue performar em aproximadamente 1 frame a cada 1.3 segundos, em contraste com os 4.28 segundos para 1 frame no projeto de referência. Em contrapartida, mesmo com uma boa acurácia no dataset de teste (97%), o teste do jogo não apresentou bons resultados, poderíamos tentar implementar Heatmap ponderado a fim de reduzir os ruído, podendo trazer melhorias.

O pipeline final para o dataset base de acordo com a busca de hiperparâmetros foi a transformação da imagem em HSL e aplicação do Histograma de gradientes Orientados (HOG).



Já para as imagens oriundas do frame do jogo, foram redimensionadas para 480x640, aplicado um filtro Gaussiano de Blur, desenvolvidas janelas de diferentes tamanhos para a busca aplicando em cada uma o pipeline acima. A última imagem demonstra o resultado do classificador.

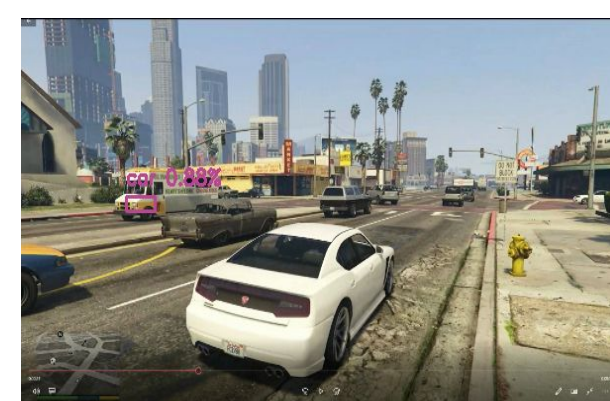


### Deep Learn:

Para a abordagem por deep learn, as questões de performance também foram levadas em consideração, por esse motivo utilizamos a YOLO v3.

Em contraste com a abordagem clássica, claramente esse método conseguiu classificar corretamente mais objetos e com uma melhor acurácia, além de pode classificar um total de 80 objetos.

Duas implementações da Darknet da YOLO foram utilizadas, a YOLOv3-320, a qual apresentou um bom resultado, porém como o jogo estava sendo executado paralelamente, ocorreu uma queda no desempenho mas satisfatório o suficiente para execução em tempo real. Para melhorar a performance, foi testado a YOLOv3-tiny e como esperado houve melhoria do FPR em detrimento da confiabilidade.



### Observações:

O jogo utilizado foi o GTAV por conta de sua qualidade gráfica próxima a realidade mesmo sendo um jogo relativamente antigo. Os gráficos do jogo foram colocados em modo Normal e seu aspecto para 800x600. A máquina a qual foi realizada o teste possui um core i5-7300HQ, 8GB de RAM e com uma placa gráfica GeForce GTX 1050.

## Conclusões

O estudo demonstrou que abordagens clássicas para a generalização de dataset reais encontram dificuldades, como o esperado, mesmo assim, os resultados podem ser melhorados caso haja um dataset classificado com mais variedade de dados de não veículos. Alguns aspectos interessantes foram a execução a 1 frame por ~1.3 secs e a utilização de Grid Search para a definição dos valores dos parâmetros de pré-processamento, o qual pode ser utilizado em outros problemas para possíveis melhorias.

A abordagem por deep learning demonstrou uma boa generalização com uma boa taxa de FPS, mesmo sendo testada em um jogo e não em vídeos da vida real.

Em conclusão, com os avanços dos gráficos e das técnicas de Visão Computacional, o teste de experimentos em um ambiente controlado como um videogame pode ser interessante e utilizado durante o desenvolvimento de projetos.

## Referência

- <https://github.com/windowsub0406/Vehicle-Detection-YOLO-ver>
- [https://www.youtube.com/watch?v=GGeF\\_3QOHGE&list=PLMoSubG1Q\\_r8nz4C5Yvd17KaXy8p0ufPH](https://www.youtube.com/watch?v=GGeF_3QOHGE&list=PLMoSubG1Q_r8nz4C5Yvd17KaXy8p0ufPH)

