

Synthetic computer vision: get our hands dirty

Proposal Report

MiaoDX (缪东旭)

2016218041

miaodx@tju.edu.cn

ABSTRACT

In this project, I will choose some simulation tools and explore their potential usage in obtaining synthetic dataset for computer vision research and applications. The results can help researchers get more specific synthetic data for their specialized purpose, and thus save lots of time compared with pruning existing dataset for usage. And since synthetic data have many advantages compared with real-world data in privacy, accuracy and they are more controllable and available for the public, so it will be worthy to put effort on this subject.

KEYWORDS

Synthetic dataset, computer vision, simulation

1 Introduction

Simulation technologies have been widely used in both academic and industrial, like the physics simulation system[1] to simulate robots before put them into production line and various commercial game engines[2], recently there appear a tendency of using synthetic image datasets and virtual environment provided by these simulation systems to do computer vision research[3][4][5].

2 Related Work

There are lots of prior works on this subject, and in fact, there is a list of related papers you can find on github page synthetic-computer-vision[8]. And some papers I want to highlight are:

UnrealCV: Connecting Computer Vision to Unreal Engine[3], which provide an tool/plugin for the Unreal Engine 4, and make it possible that all games made on it can be used for generating datasets for computer vision, they also used the synthetic datasets get from the game engine to diagnose a deep network algorithm, more specifically, a Faster-RCNN model trained on PASCAL[9] and found some interesting results.

Learning Physical Intuition of Block Towers by Example[4] is one of Facebook AI research's work, they also use Unreal Engine 4 and combine it with Torch and explored the ability of deep feed-forward models to learn intuitive physics in that paper.

Using Virtual Worlds, Specifically GTA5, to Learn Distance to Stop Signs[5] used GTA5, the closed-sourced famous game to produce rich detailed images and trained a convolutional neural network, which mimics human vision to detect stop signs and estimate the distance to them based on individual images.

And there is an AI company, OpenAI, who open sourced lots of stuff for developing, comparing, measuring and training AI algorithms in virtual environment[10][11][12].

3 Our Work

I will not introduce another novel way to synthesize data or promote existing methods to a new level, instead, as the project name implied, I will digest these tools and methods, and make some experiments and replicate the results of some papers. So that after this project, I can get the necessary knowledge to get synthetic dataset in various ways and then use them to do other research stuff afterwards.

3.1 Outline

There are (at least) three main fields of synthetic systems:

- Physics Simulations, like Bullet, Havok, MuJoCo, ODE and PhysX, etc
- Game Engines, like Unity3D, Unreal, cocos2d-x, godot and so on
- Closed source games with (unofficial/community) API for developer, like GTA V

The first one can be part of the second one, and for synthetic computer vision, the second and third one can be more appealing and suitable since they provide abundant images in various scenarios, and what's more, they looks really real.

3.2 Planed experiments

Papers/experiments I want to replicate or tools I want to try:

- Unrealcv, Connecting Computer Vision to Unreal Engine
- Using Virtual Worlds, Specifically GTA5, to Learn Distance to Stop Signs
- DeepGTAV: A plugin for GTAV that transforms it into a vision-based self-driving car research environment.
- OpenAI gym: A toolkit for developing and comparing reinforcement learning algorithms.
- OpenAI universe: A software platform for measuring and training an AI's general intelligence across the world's supply of games, websites and other applications.
- OpenAI roboschool: Open-source software for robot simulation

4 Time Line

Week 3: unrealcv

Week 5: DeepGTAV

Week 6: openai gym & openai roboschool

Week 7-8: paper work

REFERENCES

NOTE: again, I am using links instead of more official citation format, will change in the final report.

- [1] Real-Time Physics Simulation, <http://bulletphysics.org/>
- [2] Games engines, <https://github.com/showcases/game-engines>
- [3] UnrealCV: Connecting Computer Vision to Unreal Engine, <https://arxiv.org/abs/1609.01326>
- [4] Learning Physical Intuition of Block Towers by Example. <https://arxiv.org/abs/1603.01312>
- [5] Using Virtual Worlds, Specifically GTA5, to Learn Distance to Stop Signs, http://orfe.princeton.edu/~alaink/SmartDrivingCars/DeepLearning/GTAV_TB_Final.pdf
- [6] synthetic-computer-vision , <https://github.com/unrealcv/synthetic-computer-vision>
- [7] unrealcv , <https://github.com/unrealcv/unrealcv>
- [8] synthetic-computer-vision, <https://github.com/unrealcv/synthetic-computer-vision>
- [9] py-faster-rcnn, <https://github.com/rbgirshick/py-faster-rcnn>
- [10] OpenAI gym, <https://github.com/openai/gym>
- [11] OpenAI universe, <https://github.com/openai/universe>
- [12] OpenAI roboschool, <https://github.com/openai/roboschool>