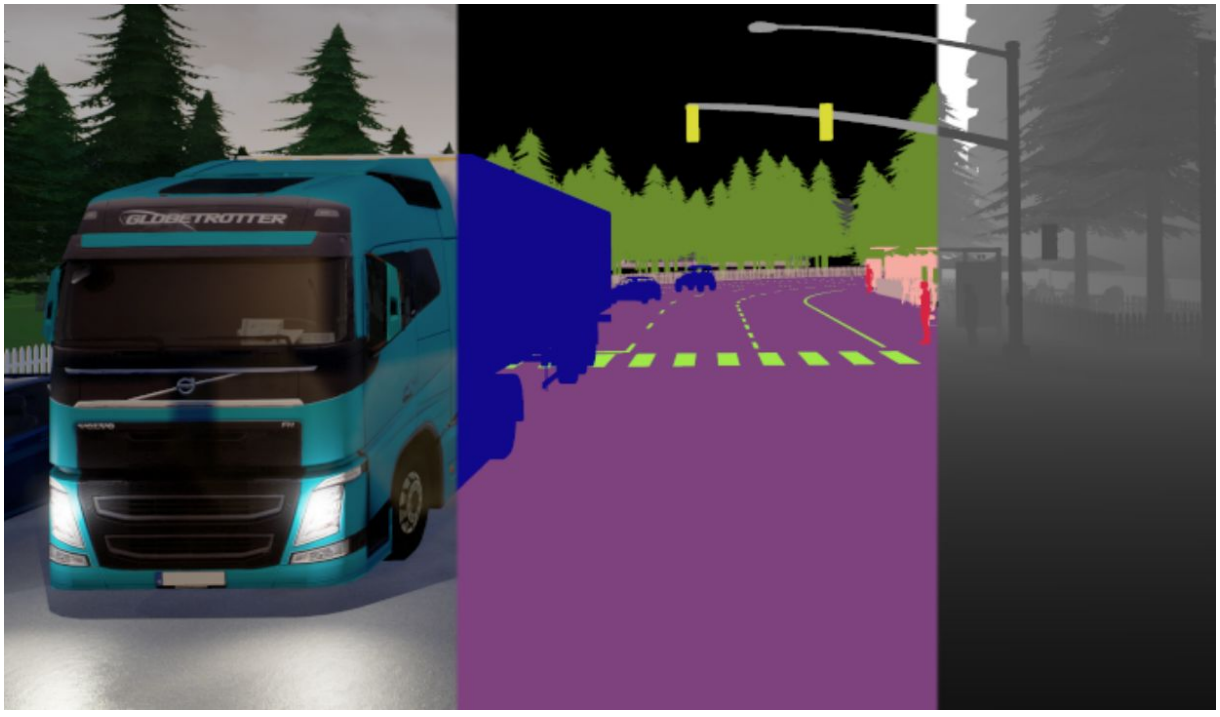
Realistisk simulering av fordon för säkrare, robustare och billigare utveckling av automatiserade fordon

Publik rapport



Författare: Anton Kloek, Machine Intelligence Sweden AB
Datum: 2020-01-31
Projekt inom Machine Learning

FFI Fordonsstrategisk Forskning och Innovation
VINNOVA Energimyndigheten TRAFIKVERKET
SCANIA VOLVO

Innehållsförteckning

Kort om FFI
FFI är ett samarbete mellan staten och fordonsindustrin om att gemensamt finansiera forsknings- och innovationsaktviteter med fokus på områdena Klimat & Miljö samt Trafiksäkerhet. Satsningen innebär verksamhet för ca 1 miljard kr per år varav de offentliga medlen utgör drygt 400 Mkr.

För närvarande finns fem delprogram; Energi & Miljö, Trafiksäkerhet och automatiserade fordon, Elektronik, mjukvara och kommunikation, Hållbar produktion och Effektiva och uppkopplade transportsystem. Läs mer på www.vinnova.se/ffi.

1 Sammanfattning

Ett av de absolut största problemen med att utveckla system för autonoma fordon är tillgången till annoterad data för supervised machine learning samt en realistisk simuleringsmiljö för reinforcement learning. I detta projekt utvecklas två simuleringsverktyg, ett för generering av syntetisk data och ett för förfining av syntetisk data. Verktygen, som vi kallar SweSim och RealSim, är specialanpassad för just domänen autonoma fordon, men kan generaliseras till flera domäner. RealSim använder sig av unsupervised maskininlärning för att lära sig att översätta syntetisk data från SweSim till data som återspeglar verkligheten bättre. Verktygen, inom ramen för detta projekt, ska återspegla svenska städer, landsväg och motorväg och både kunna användas för att lära intelligent beteende hos agenter, genom reinforcement learning, samt generera enorma mängder träningsdata i form av syntetisk sensordata tillsammans med tillhörande ground truth.

Under projektets gång ser vi att syntetisk data är användbart för träning av perceptions system men att ytterligare förbättringar behövs för att uppnå likvärdiga resultat med loggad data. Utöver detta ser vi att förfining av syntetisk data ger märkbart förbättrade resultat mot ren syntetisk och börjar närma sig dem resultat vi ser hos loggad utvärderingsdata. Slutligen, kombinering av förfinad syntetisk data och begränsad mängd loggad data kan i enskilda klasser slå träning med 100% loggad data. Detta tyder på en lovande framtid för förfinad syntetisk data, men mer arbete behövs.

2 Executive summary in English

One of the absolute biggest problems in developing systems for autonomous vehicles is the availability of annotated data for supervised learning and a realistic simulation environment for reinforcement learning. In this project, two simulation tools are being developed, one for generating synthetic data and one for refining synthetic data. The tools, which we call SweSim and RealSim, are specially adapted for the domain autonomous vehicles, but could be generalized to several domains. RealSim uses unsupervised machine learning to learn how to translate synthetic data from SweSim into data that better reflects reality. The tools, within the framework of this project, will reflect Swedish cities, country roads and highways and can both be used to teach intelligent behavior of agents, through reinforcement learning, and generate huge amounts of training data in the form of synthetic sensor data along with associated ground truth.

During the course of the project we see that synthetic data is useful for training perception systems but that further improvements are needed to achieve equivalent results with logged data. In addition, we see that refinement of synthetic data yields noticeably improved results against pure synthetic and is beginning to approach those results we see in logged evaluation data. Finally, combining refined synthetic data with a limited amount of logged data can, in individual classes, beat training with 100% logged data. This suggests a promising future for refined synthetic data, but more work is needed.

The final results indicate that we are reaching our goal of being able to replace 90% of logged data, but only in limited classes. Further work will be needed to improve the technology across all classes.

3 Bakgrund

För att utveckla robusta maskininlärningssystem för självkörande bilar behövs otroligt mycket data för göra säkra, robusta och kostnadseffektiva system. Den 7:e maj 2016 skedde det första dödsfallet för självkörande bilar när en Tesla i autopilot inte kunde särskilja en vit lastbil mot den vita himlen [1]. Dagens maskininlärningssystem blir bra på att lösa ett specifikt problem på de fallen systemet tränar på, s.k. träningsdata. Utöver dessa fall i träningsdatan finns det väldigt få garantier för hur väl systemet fungerar även för väldigt små avvikelser från dessa. I praktiken undanhåller man en del av sin data för att sedan mäta det tränade systemets prestanda på data som inte varit en del av träningsprocessen. På så sätt kan man få en indikation på hur väl systemet generaliserar utanför träningsdatan. Då naturliga bilder från kameror och sensordata har stor variation i sin natur krävs en väldigt stor mängd träningsdata för att göra systemen tillräckligt robusta. Datan ska inkludera variation i fordon, vägar, ljusförhållande, natur, årstid, väder, geografisk position, etc. Att införskaffa data med denna variation kan systematiseras genom att man kontinuerligt loggar data från sensorer i sin fordonsflotta. Dock är denna data i sig själv av begränsad nytta.

State-of-the-art algoritmer som analyserar omgivningen och försöker uppfatta var man kan köra (free space detection), hitta och klassificera objekt (object detection), etc., använder sig av "supervised deep learning" och kräver därför att man annoterat data, dvs att man skapar ett facit som algoritmen kan lära sig av [2, 3]. Detta innebär en extremt kostsam och tidskrävande process för att skapa sig ett facit som kan användas som träningsdata.

En annan ansats till självkörande bilar är att använda sig av "reinforcement learning" för att indirekt lära sig att förstå sin omgivning och navigera i den på ett säkert sätt [4]. Detta kräver dock tillgång till en omgivning där systemet lär sig på ett smart sätt genom trial-and-error. Detta är dock högst olämpligt att göra i en verklig miljö. Istället används en simuleringsmiljö som byggs för att vara så lik verkligheten som möjligt. Dessa virtuella miljöer återspeglar dock inte den komplexa verkligheten exakt, men har ändå visat sig vara användbara [5].

En annan viktig aspekt av en kompetent simuleringsmiljö är att man kan variera scenarios och variationer i väg, omgivning, väder, vegetation, etc., i den utsträckning simuleringsverktyget tillåter. Därmed kan man täcka in många svåra och ovanliga scenerier som inte fångats vid verklig insamling av data och se till att systemet är säkert och robust även vid sådana scenarier. Man kan också spela upp scenarier för att testa sitt tränade system för att se hur väl de fungerar. Men både för att träna och/eller utvärdera ett system in en virtuell miljö ställer extremt höga krav på att denna miljö är tillräckligt bra för att ge pålitliga och meningsfulla resultat.

Vidare finns en annan otroligt stor fördel med simuleringsmiljöer, som kommer att exploateras i detta projekt, nämligen att de kan användas för att generera träningsdata till supervised machine learning automatiskt. Detta då den exakta omgivningen är känd och tillgänglig i simuleringsmiljön. Således kan väldigt stor mängd träningsdata genereras väldigt billigt och mycket snabbare och bättre än om det skulle annoteras för hand av människor.

Användandet av simuleringsmiljöer, eller till och med datorspel, är inget nytt för att generera träningsdata för maskininlärningssystem för computer vision [5]. Metoden har visat sig effektiv i att förbättra prestandan för maskininlärningssystem för en mängd olika problem, t.ex. pedestrian detection [6], human pose detection [7, 8], object detection [2, 3], image segmentation och image

depth estimation [9]. Flera av dessa studier använde en enkel form av domain adaptation genom att systemet först tränas på syntetisk data för att sedan finjusteras på den verkliga datan. På så sätt nåddes ett resultat som var bättre än att bara använda den riktiga datan [9].

En ortogonal ansats till vår metod är att fokusera på att träna systemet som har egenskaper som är invarianta för domänen. På så sätt kan de tränas på simulerad data och tillämpas i verkligheten, dvs en annan domän. Detta kallas för “domain adaptation”. Domain adaptation kan dock kombineras med ansatsen att kombinera riktig och syntetisk data för att nå förbättrad prestanda [10].

Dagens teknik för rendering av väg och omgivning är imponerande, men det är fortfarande relativt enkelt för en människa att säga om en bild kommer från verkligheten eller simulering, se figur 1.

Figur 1. Till vänster: simulerad bild. Till höger: verklig bild

4 Syfte, forskningsfrågor och metod

Syftet med detta projekt är att generera syntetisk träningsdata såpass väl att den blir tillräckligt bra för att användas vid uppträning av perceptions system som kan appliceras i verkligheten. Då målet med med syntetisk data till stor del är motiverat av kostnadsbesparingar så inkluderar syftet även att lösa syntetisk data på ett sätt som är ekonomiskt gångbart och skalbart.

Syntetisk data, Generalisering av syntetisk data till loggad data. Utvärdera syntetisk data mot loggad, utveckla verktyg, tekniker och processer för att göra syntetisk data mer verklig alternativt mer generaliserbar.

Två frågor som uppkommer när man använder en simuleringsmiljö för att dra slutsatsen om den riktiga världen
1. Hur lik verkligheten är simuleringen?
2. Fungerar lösningen i verkligheten när den är framtagen i en simulerad miljö?

Båda dessa frågor blir dock beroende av flertalet variabler så som att definitionen av verkligheten i detta fall blir den data som loggats. Detta medför att ett mått på likhet mellan verklighet (loggad data) och syntetisk data är helt beroende på vilken verklighet och huruvida den syntetiska miljön anpassats därefter.

Då detta problem blir utanför projektets omfång kommer både insamling av data och anpassning av simuleringsmiljö begränsas till Sverige. Det blir då lättare att formulera mätbara mål.

De mätbara mål vi har för projektet är:
1. Kan vi ta bort 90% av den annoterade datan för att behålla samma prestanda, om vi Kombinerar den loggade datan med syntetisk data? Vad betyder detta i kostnadsminskning?
2. Kan vi mäta en förbättring i prestandan för användarfallen i projektet när metoden för att förfina inputen (RealSim) används?

Metoden i projektet kommer följa en iterativ utvecklingsprocess där varje steg utvärdera utvärderas för att få en djupare förståelse för vad som krävs för att uppnå resultatet som krävs för dem mätbara målen.

5 Mål

Projektet har delats upp i totalt 6 Arbetspaket (AP). AP1 var projektledning så de enda målen relaterade till det är rapportering med mera. I övriga AP är målen beskrivna per AP nedan.

AP2
Utrusta lastbil med sensorer för loggning av data. Detta inkluderar Lidar, Kamera, IMU, GPS och möjlighet att läsa data från CAN bussen.

AP3
- Loggning av data. Loggningen kommer ske iterativt efter behov, och täcka en variation av miljöer såsom stad, motorväg och landsväg.
- Annotering av loggad data. Annoterad data behövs i projektet för att kunna utvärdera resultat.

AP4
Utveckla möjligheten att generera syntetisk träningsdata för perception i självkörande bilar. Detta inkluderar:
- Utveckling av simuleringsmiljö och sensorer.
- Modellering av svenska miljöer och svenska assets.
- Mjukvara för generering av data efter valda förhållanden såsom väder, scenario med mera.

AP5
- Utveckla tekniker och verktyg för att kunna förbättra syntetisk data så att den bättre generaliserar vid utvärdering mot loggad data.
- Utveckla en komplett pipeline för utvärdering och förbättring av ovanstående komponenter.
- Simple Perceptions prototyp för att kunna träna upp och utvärdera på loggad data.

AP6
Utvärdering av syntetisk data mot loggad data.

6 Resultat och måluppfyllelse

AP2

Utrustning av lastbil tog initialt lite längre tid än planerat. Förseningen berodde till största del på leverans av hårdvara. Detta ledde till förseningar i insamling av data.

AP3

Insamling/Loggning av data kom igång efter ordinarie schema pga av i AP2 nämnda förseningar. Denna försening hade däremot ingen långsiktig negativ effekt på projektet pga:

1. Insamlingen var tidigt planerad i projektet för att ha god buffer för eventuella förseningar.
2. Öppna dataset kunde användas så länge vilket även medför bättre jämförbarhet.

Annotering av data nådde ej upp till vision både när det gäller mängd och kvalitet. Detta beror på att budgeten för detta arbete var naivt planerad och därmed alldeles för liten. Detta medförde utvärdering behövde bero mer på publika dataset än våran egen loggade data.

AP4

Att bygga hel spelmotor från grunden var självklart aldrig ett alternativ, utan projektet var tänkt att byggas på Unreal Engine (UE4) [13]. UE4 är en av dem kraftfullast grafik motorerna som finns och är tillgänglig open source. Detta ger en gedigen och skalbar grund som regelbundet får tillgång till senaste forskningen inom grafik förbättringar.

Utöver detta så användes även Carla [14]. Carla är ett open source projekt med många liknande funktioner som hade planerats för SweSim. Carla utvecklades ursprungligen för AD forskning men har varit begränsad till Amerikanska miljöer och assets.

Utnyttjandet av Carla gjorde att större del av budgeten kunde användas till dem delar av projektet som uppfyller våra mål, dvs att skapa just svenska miljöer och assets. Detta inkluderar allt från städer, samhällen, vägar, vägmarkeringar, vägskyltar, vegetation, byggnader, människor och helhetskänsla.

Från SweSim genererades 3 generationer av dataset. Där varje generation syftade till att förbättra dem mest akuta brister vi såg i varje generation. Generation 1 skapades som ett ordinarie benchmark.

Inför generation 2 drogs slutsattsen att alla klasser underpresterade vid utvärdering. För att lösa detta så skapades 3 nya världar med större variation. Utöver detta så skapades även en ny form av gångtrafikanter (Pedestrian) som kunde slumpa flera faktorer i utseende vid skapande och på så sätt få en rikare data specifikt riktat mot denna klass.

Resultatet av detta blev att evaluerings resultatet i just klassen “Pedestrian” höjdes med 0.28 (IoU). Detta kan jämföras med resten av klasserna som i genomsnitt höjdes 0.07 (IoU).

Inför skapandet av dataset generation 3 lades lite fokus på flera klasser, bla vägskyltar, trottoarer och städer i helhet. Vägskyltar såg en höjning på 0.06 (IoU). Trottoarer såg ingen höjning, detta kan möjligtvis förklaras med att arbetet med förbättring framförallt fokuserade på trottoarkanter då detta är ett av perceptions systemets möjligheter att skilja mellan trottoar och vanlig väg men att

detta var en felaktig tes. Förbättringarna i städerna i helhet ledde till en höjning av klassen “Wall” med 0.01 (IoU)

AP5
Under projektets gång har hundratals RealSim experiments gjorts. Ett återkommande problem med RealSim resultaten har varit hur detta skall utvärderas. Då RealSim delvis bygger på en adversarial metod så uppstår ett instabilitetsproblem där optimering sker i två dimensioner. Detta innebär att kvantitativ utvärdering är svår och en mer kvalitativ utvärdering är både tidskrävande och inte alltid fullt pålitlig. För att få en fullt pålitlig kvantitativ utvärdering behövs hela pipelinen från RealSim och framåt köras. Dvs:

1. Förfina dataset med Realsim
2. Träna upp perceptions system med förfinad data.
3. Utvärdera perceptions system på loggad data.

Att köra hela denna pipeline är tids och kostnadskrävande och att då göra det för varje RealSim experiment var inte realistiskt. För att komma runt detta har experimenten först bedöms kvalitativt genom manuell inspektion av förfinad data.

Dem experiment som genererat intressanta resultat har sedan vidare utvecklats och hyper parameter optimerats för att slutligen bli en del av slut utvärderingarna i AP6 där den kompletta pipelinen har stått för ett konkret kvalitativt slutresultat.

Totalt har 4 olika refiners nått full utvärdering. Dessa kallas framföver Refiner G1, Refiner G2, Refiner Mix1 och Refiner Mix2

AP6
För full utvärdering och möjlighet att besvara frågan “Kan vi ta bort 90% av den annoterade datan för att behålla samma prestanda, om vi kombinerar den loggade datan med syntetisk data? Vad betyder detta i kostnadsminskning?” behövdes en full pipeline som även inkluderar finträning (finetuning) med 10% loggad data. Dvs:

1. Generera dataset med SweSim
2. Förfina data med RealSim
3. Träna upp perceptions system med förfinad data
4. Finträna med 10% loggad data.
5. Utvärder på 100% loggad data.

Dessa steg utfördes på alla generationer dataset och i fallen dataset Generation 2 och 3 utfördes utvärderingar med alla refiners. I Generation 1 utfördes bara utvärdering med fullt synthetisk data och förfinad data med Refiner G1.

I dataset Generation 1 missades alla mål med stor marginal

I dataset Generation 2 missades målet i alla klasser utom en. För att få en bättre bild av hur långt från målet resultatet var repeterades steg 4 och 5 av pipelinen ovan, fast med ökad procent loggad data i steg 4. Detta resulterade i följande:
Med 20% loggad data slogs 3 klasser i utvärdering.
Med 30% loggad data slogs 3 klasser i utvärdering.

Med 40% loggad data slogs 4 klasser i utvärdering.

Resultaten ovan är baserade på Refiner Mix2 då detta var den bäst presterande refinern i denna evaluerings runda.

I dataset Generation 3 var resultaten liknande. Vid finträning med 10% loggad data slogs bara en klass. På grund av tid och pengabrist kunde inte samma samma ökning av loggad data i steg 4 utföras.

Trots att det totala antalet slagna klasser inte ökade mellan utvärdering av Gen2 och Gen3 så kan man ändå se en positiv trend då de totala mIoU resultatet för ren förfinad data gick upp med 0.02 från gen2 till gen3. Slut testerna på Gen 3 datasetet är även dem baserade på Mix2 Refinern.

För att svara på andra delen av ovanstående forskningsfråga, Vad betyder detta i kostnadsminskning?, behövs först metoderna jämföras mot varandra i arbetsprocess.

För manuellt loggad och annoterad data behövs:
1. Utrustning av fordon med sensorer.
2. Insamling av data
3. Manuell annotering.

Ovan ser vi att det första steget är en engångshändelse och därmed engångskostnad, det andra steget sker i behov av mängd data och variation. Det tredje steget sker för att göra datan användbar.

Motsvarande steg för syntetisk data blir:
1. Utveckling av grund simulator.
2. Utveckling av miljöer
3. Generering av data.

På samma sätt som för loggad data ser vi att steg ett är en engångshändelse och därmed engångskostnad, det andra steget sker efter behov av variation (och delvis mängd). Det tredje steget ger dig tillgång till data, dvs gör den användbar.

Kostnaderna för steg 1 och 2 kan ses som likvärdiga mellan de två teknikerna. Den stora skillnaden kommer i steg 3.

Under projektets budget annoterades en viss mängd data. Slut priset per bild landade på 82 kr/bild. Även om detta skulle kunna sänkas genom outsourcing till andra länder så finns det även dyrare exempel där varje bild annoterats av tre personer för att få ett perfekt resultat.

Motsvarande kostnad för syntetisk data blir ca 0,1 kr/bild. Detta inkluderar strömkostnad för dator och viss mängd manuellt arbete.

På forskningsfrågan “Kan vi mäta en förbättring i prestandan för användarfallen i projektet när metoden för att förfina inputen (RealSim) används?” är svaret tveklöst ja. I samtliga fall har data förfinad med RealSim technologin slagit utvärderingsresultat av motsvarande dataset utan förfining. I det fall med störst skillnad, dataset Gen 3 med Refiner Mix2, så var ökningen i utvärderingsresultat 0.04 (mIoU)

7 Spridning och publicering

7.1 Kunskaps- och resultatspridning

Hur har/planeras projektresultatet att användas och spridas?	Markera med X	Kommentar
Öka kunskapen inom området	X	
Föras vidare till andra avancerade tekniska utvecklingsprojekt	X	Ett nytt FFI projekt där tekniken används har redan påbörjats.
Föras vidare till produktutvecklingsprojekt	X	Tekniken vidareutvecklas internt för produktifiering.
Introduceras på marknaden	X	Kontakt med potentiella kunder har redan påbörjats.
Användas i utredningar/regelverk/ tillståndsärenden/ politiska beslut		

7.2 Publikationer

Inga publikationer har ännu släppts, men mycket kunskapsspridning har skett genom presentationer av teknik och resultat under projektets gång. Dessa presentationer har regelbundet skett med en publik som haft en blandning av forskare, entreprenörer och storföretag som kan bli potentiella kunder. Exempel på ställen där tekniken presenterats inkluderar:
CampX invigning
BADEN SWII
Telematics Valley
MobilityXLabs (flera event)
AI Innovation of Sweden

8 Slutsatser och fortsatt forskning

Experimenten visar att tekniken fungerar. Dock så krävs ett antal iterationer av förbättringar för att uppnå ett användbart resultat. Eftersom förbättringarna i varje iteration kan användas för all framtida generering av syntetisk data så är alla framsteg som gjorts här stora steg till kommersialisering. Ytterligare arbete behövs för att lyfta den syntetiska grund datan till att konkurrera med loggad data i samtliga klasser. Detta arbete kommer fortsätta delvis i ett nytt Vinnova FFI projekt, ett projekt har redan påbörjats.

9 Deltagande parter och kontaktpersoner

1. https://www.theguardian.com/technology/2016/jun/30/tesla-autopilot-death-self-driving-car-elon-musk
2. Baochen Sun and Kate Saenko. From virtual to reality: Fast adaptation of virtual object detectors to real domains. In BMVC, 2014.
3. Joerg Liebelt and Cordelia Schmid. Multi-view object class detection with a 3d geometric model. In CVPR, 2010.
4. Jeff Michels et al.. "High speed obstacle avoidance using monocular vision and reinforcement learning" ICML, 2005.
5. Geoffrey R. Taylor, Andrew J. Chosak, and Paul C. Brewer. OVVV: Using virtual worlds to design and evaluate surveillance systems. In CVPR, 2007.
6. Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In CVPR, 2005.
7. Alireza Shafaei and James J. Little. Real-time human motion capture with multiple depth cameras. In CRV, 2016.
8. Jamie Shotton, Ross Girshick, Andrew Fitzgibbon, Toby Sharp, Mat Cook, Mark Finocchio, Richard Moore, Pushmeet Kohli, Antonio Criminisi, and Alex Kipman. Efficient human pose estimation from single depth images. TPAMI, 2013.
9. Alireza Shafaei. Play and Learn: Using Video Games to Train Computer Vision Models. CoRR, abs/1608.01745, 2016.
10. Ganin Y, Lempitsky V. Unsupervised domain adaptation by backpropagation. arXiv preprint arXiv:1409.7495. 2014.
11. Shrivastava A, Pfister T, Tuzel O, Susskind J, Wang W, Webb R. Learning from Simulated and Unsupervised Images through Adversarial Training. arXiv preprint arXiv:1612.07828. 2016.
12. Phillip Isola et al.. "Image-to-Image Translation with Conditional Adversarial Networks" , 2016.
13. Unreal Engine: https://www.unrealengine.com/
14. Carla: http://carla.org/