Exjobbs Manual

Margareta Vi

June 2018

1 Introduction

Detta är en manual för hur man kan träna upp ett neuron nätverk på den dator som jag har använt under exjobbet. Jag kommer att beskriva hur man skapar träningsdata, hur man tränar och hur man sedan använder sig av det färdigtränade nätverket. All kod finns på datorn under /Document/exjobbV2 samt på repot github.com/marvi154/exjobbV2.

I detta repo finns det 2 mappar: Blender files, alltså CAD modeller inlästa som blender filer. Utils: Smått och gott funktioner som används ibland Sedan så ligger det en massa .py filer i huvudmappen, dessa filer har används ofta.

Allting körs ifrån terminalen.

2 Skapa träningsdata

Träningsdata omfattar bilder samt .xml fil per bild som talar om vart i bilden som objekten finns.

Det som används i detta steg är rendering.sh

I rendering.sh filen så har jag specificerat vart alla blender filer ligger, vart bakgrundstexturer, texturer på objekten finns samt vart bilderna ska sparas.

OBS!! BILDERNA SKA HA FORMATET JPEG/JPG, DÅ DETTA ÄR ENDA FORMATET SOM NÄTVERKET TAR IN.

Det viktiga är att den mappen som blender filerna finns i, ska det INTE FINNAS några submappar, då jag ej har tagit hänsyn till detta!

Blenderfilerna i sig själva är lite pilligt, det är viktigt att objekten inte är för stora så att objekten fyller hela bilden som renderas. Detta (storleken på objekten) får användaren själv justera i blender programmet. Enklast är att man justerar i blender tar ett kort och tittar på den innan man kör en full rendering.

Objektet startposition ska vara koordinatsystemets origio.

Kolla igenom bilderna så att de ser vettiga ut. Exempel är att objekten syns i bilden och inte är för stor.

I dags läget så renderas bilderna enbart med 1 objekt, för att renderingen med flera objekt ska ske måste flera objekt finnas med i samma .blend fil samt så måste funktionen som skapar .xml filerna justeras. så att varje objekts position finns med.

3 Innan du tränar nätverken

För att träna upp ett nätverk så bör terminalen vara i den virtuella miljön tensorflow1-7. Detta görs genom att köra: source /Documents/tensorflow1-7/bin/activate
Sedan så måste köra denna rad:
export PYTHONPATH=$PYTHONPATH:‘pwd‘:‘pwd‘/slim köras ifrån mappen /Documents/tensorflow1/models/research för att tensorflow ska ha allting den behöver.

Det finns 4 saker som behövs göras innan träning kan ske.

1. Dela upp den data du har renderat i 3 delar, en för träning, evaluering testning. Skapa en mapp med namnen ”train”,”eval”, ”test”. Dessa namn kommer att spela roll i nästa steg. Alla bilder ska ha fått en motsvarande .xml fil där positionerna är nedskrivna.

2. Kör skriptet xml_to_csv.py $Path\to\parentfolde$ **path to parent** folder är mappen som har ”train”, ”test”, ”eval” som submappar. Det kommer att skapas 3 filer: train_labels.csv osv för eval och test.

3. Sedan ska skriptet:
generate_tfrecord.py –csv-input=PATH_TO_XXX_lables.csv
–image_dir=PATH_TO_XXX_IMAGES
–output=OUTPUTH_PATH_FOR_FILE köras för varje _labels.csv fil som du har skapat. Observera att detta är ett ena långt kommando.

4. Skapa en fil som heter labelmap.pbtxt varje klass som finns med i ditt nya dataset ska ha en egen rad. Kolla på hur labelmap.pbtxt ser ut så förstår du. OBS! numreringen börjar på 1. O är reserverad för ”ingen ground truth class”

4 Träning

Nu för träning Exemple: python train.py –logtostderr –train_dir=training/
–pipeline_config_path=training/faster_rcnn_inception_v2_pets.config

- –logtostderr används för att få en vettig utskift
- –train_dir är vart modellerna ska sparas
- –pipeline_config_path är vart XXX.config filen sparas

.config filen är en fil som specificerar hur nätverket ser ut. Det är i denna fil som du specificerar hur länge du vill träna t.ex. Samt sökvägen till .record filerna som skapas i steg 3 ovan.

Om du skulle vilja evaluera nätverket så kör du python eval.py –logtostderr –eval_dir=evaluation/
–pipeline_config_path=training/faster_rcnn_inception_v2_pets.config –checkpoint_dir=training/

där checkpoint_dir pekar på train_dir mappen.

5 Efter träning

Efter träningen så ska du köra ett likande kommando:
python export_inference_graph.py –input_type=image_tensor
–pipeline_config_path=/home/xmreality/Desktop/ikea_dataset/configs/faster_rcnn_inception_v2_coco.config
–trained_checkpoint_prefix=/home/xmreality/Documents/exjobb_results/ikea/SSD_inception/train/model.ckp
100000
–output_directory=/home/xmreality/Documents/exjobb_results/ikea/50percent/FasterRCNN/inference_graph
Det som behövs ändras är pekningarna av

- – pipeline_config_path: samma som vid träning
- – trained_checkpoint_prefix: är den modell som du skulle vilja komma på, dem ska finnas sparade i ”train_dir”
- – output_directory: vart du vill ha grafen.

För att köra med webkamera så koppla in en extern webkamera och kör följande skript:

python object_detection/Object_detection_webcam.py –model_name=/home/xmreality/Desktop/Results_netwo
–path_to_labels=/home/xmreality/Desktop/ikea_dataset/labelmap.pbtxt –number_of_classes=6

- –number_of_classes hur många objekt du har tränat på, ska stämma överens med labelmaps.pbtxt

Ett typiskt fel är att programmet inte hittar kameran, gå då in i Object_detection_webcam.py och ändra så att rätt kamera väljs,
# Initialize webcam feed
video = cv2.VideoCapture(0) (från 0-1 brukar hjälpa)
kolla i filen Object_detection_scratch.txt som ligger på datorns skrivbord för kommando exempel.