Social Distancing Analyzer: Queries

Date: 1sept 2021

1. what is the name of the depth perception model u are going to use? in general terms how does it work?

I’m planning to use the Encoder-Decoder type model architecture because it’s an image translation task.

The encoder will encode the feature map of the input image and the decoder will upscale this feature map into a depth map.



We are using Residual network type depth architecture as an encoder having some skip connections like the Resnet50 model.

The encoder will encode images into a fine feature map according to our requirement from the RGB image.

In Decoder, we are using the bunch of Deconv layers for upsampling feature maps into depth maps as per need.

In the loss function, we are using

1. mean square loss such that the predicted depth map value is close to the actual depth map.
2. Using custom edge loss: for improving sharpness between two different objects. If there are different objects this loss makes their boundary depth more clear. Using Sobel detector or implementing canny edge detection part for improving depth map edges.
3. what metrics do you obtain after the training model that determine the performance of the model? (names of the metrics and what they mean)?

I’m planning to calculate Absolute\_relative\_error, Root mean square error, Root mean square log error, delta error, delta2 error, delta3 error, square relative error.

I’m choosing these metrics because most of the depth models evaluated on the basis of these metrics criteria( reference: papers with code)

1. the dataset that you are going to start searching for in the next days what data does it have? videos? 2 camera points of view per video? I don’t know

I’ll try to find multiple datasets:

1. Dataset for Depth model:

Dataset having one input RGB image(taken by camera) and their respective depth map.

1. Dataset for person detection
2. I’ll collect some videos using 2 different cameras having the same focal length with a known distance between them. ( I’ll collect this dataset later after completing the rest of the work for testing purposes mainly.)

Date 2 Sept 2021

Milestone 2( Deadline 12 Sept)

1. Dataset obtention, what data is it? What’s the dataset name? where did u obtain it from?
2. Trained depth model, what are we training? what is the architecture (in general terms) of the model, what type of model is this, labelled modelled and using CNNs?

I’m trying 2 different type of architecture of models parallally.

1. First is Adabins models for depth estimation.
2. Trying encoder-decoder architecture in which architecture is similar to resnet50 but at some places i changed no of filters in models. And we build a custom decoder using the Conv2d\_transpose layer.
3. Metrics, why are these metrics good for measuring the performance of the model? What are the matric results?

References:

## **Dataset:**

1. **Human detection dataset:**

Description: CrowdHuman is a benchmark dataset to better evaluate detectors in crowd scenarios. The CrowdHuman dataset is large, rich-annotated and contains high diversity. CrowdHuman contains 15000, 4370 and 5000 images for training, validation, and testing, respectively. There are a total of 470K human instances from train and validation subsets and 23 persons per image, with various kinds of occlusions in the dataset. Each human instance is annotated with a head bounding box, human visible-region bounding box and human full-body bounding box. We hope our dataset will serve as a solid baseline and help promote future research in human detection tasks.

Link:<https://www.crowdhuman.org/>

PS: Downloaded dataset in my local WIndow but not uploaded on Google drive because of the very large size of the dataset.

1. **NYU\_DEPTHV2 dataset:**

Description: The NYU-Depth V2 data set is composed of video sequences from a variety of indoor scenes as recorded by both the RGB and Depth cameras from the Microsoft Kinect. It features:

* 1449 densely labelled pairs of aligned RGB and depth images
* 464 new scenes taken from 3 cities
* 407,024 new unlabeled frames
* Each object is labelled with a class and an instance number (cup1, cup2, cup3, etc)

PS: Dataset Downloaded in my laptop as well as uploaded on google drive.

Descripción: El conjunto de datos NYU-Depth V2 está compuesto por secuencias de vídeo de una variedad de escenas interiores grabadas por las cámaras RGB y Depth de Microsoft Kinect. Cuenta con:

1449 pares de imágenes de profundidad y RGB alineadas densamente etiquetadas

464 nuevas escenas tomadas de 3 ciudades

407,024 marcos nuevos sin etiqueta

Cada objeto está etiquetado con una clase y un número de instancia (cup1, cup2, cup3, etc.)

1. **KITTY dataset:**

Description: It contains over 93 thousand depth maps with corresponding raw LiDaR scans and RGB images, aligned with the ["raw data"](http://www.cvlibs.net/datasets/kitti/raw_data.php) of the KITTI dataset.

Given a large amount of training data, this dataset shall allow the training of complex deep learning models for the tasks of depth completion and single image depth prediction. Also, we provide manually selected images with unpublished depth maps to serve as a benchmark for those two challenging tasks.

PS: Dataset Downloaded in my laptop as well as uploaded RGB images on google drive.

Descripción: Contiene más de 93 mil mapas de profundidad con los correspondientes escaneos LiDaR sin procesar e imágenes RGB, alineados con los "datos sin procesar" del conjunto de datos KITTI.

Dada una gran cantidad de datos de entrenamiento, este conjunto de datos permitirá el entrenamiento de modelos complejos de aprendizaje profundo para las tareas de finalización en profundidad y predicción de profundidad de una sola imagen. Además, proporcionamos imágenes seleccionadas manualmente con mapas de profundidad no publicados que sirven como punto de referencia para esas dos tareas desafiantes.

# Model Architecture

Before jumping to model architecture lets cover some layers and some model architecture we used for building model,

1. CNN model architecture

(<https://bit.ly/3CMeDOq>), (<https://bit.ly/3ESgzGC>) By going through this link you will understand about CNN model and Convolutional layer, pooling layer, Fully connected layer(Dense layer), Softmax layer( It’s kind of Activation function), Output layer.

1. Pooling layer[(](https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/#:~:text=Pooling%20layers%20are%20used%20to,generated%20by%20a%20convolution%20layer.)<https://bit.ly/3kDHldR>)
2. Activation Functions and its need in neural Network( <https://bit.ly/3CHyTAz>)
3. Batch Normalization layer( <https://bit.ly/3i0MkUg>)
4. Resnet model Architecture: (<https://bit.ly/2XHi8pY>) go through this link for understanding resnet model and its need. After reading this blog if you still have any questions regarding this feel free to ask.
5. Conv2d transpose( <https://bit.ly/3u7pQWu>)
6. Model architecture is based upon encoder-decoder architecture because our problem statement is related to encoder-decoder architecture.

**Encoder:**

Here Encoder work as a feature extractor

Let me first explain what i meant by feature extractor,

Features are part of patterns of an object in an object in an image that helps to identify it.

Generally, we are using Convolutional Neural Network(CNN), Since CNN have a strong ability to extract complex features that express the image in much more detail, learn the task-specific features and are much more efficient.

1. The input size of encoder architecture is (240,320).
2. The encoder is similar to the resnet50 model( hope you learn about Resnet50 model architecture in the above link).
3. Decrease depth of the model by removing some layers in the resnet50 model
4. Encoder output size after decreasing depth is (30,40,1024).

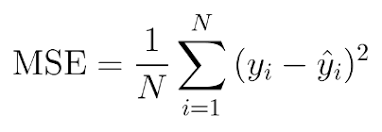
**Decoder:**

1. The decoder is doing just opposite work from decoder this will decode features encoded by the encoder to required output, in our case to depth map of on image.
2. For decoding feature maps encoded by the encoder, we are using Conv2dTranspose layers for upsampling. ( hope you learn about conv2d transpose layer functionality from the above-provided link).
3. There are skip connections in between decoder layers.( Hope you get an idea of skip connection in Resnet50 model) the same concept is here too. If you have any questions feel free to ask. Just comment on your questions.
4. The output of the decoder is the depth\_map of the object.

Loss Function:

In neural networks, we are looking to minimize error. As such the objective function is referred to as loss function and the value calculated by the loss function is referred to as simply “loss”.

In-Depth estimation task we need our depth to be nearer and nearer to Ground truth Depth(actual depth). So we are using Mean square error as a loss function. Equation mentioned below:



# Person Detection

I’m using the yolov5 model, for making my model capable of detecting only persons.

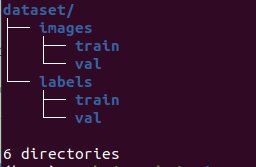
I’m listing all the steps for performing any kind of object detection task using yolov5

1. First clone official YoloV5 repository code

[git clone https://github.com/ultralytics/yolov5.git](https://github.com/ultralytics/yolov5.git)

1. Download Crowdhuman dataset

[crowdhuman.org/download.html](https://www.crowdhuman.org/download.html)

1. Preprocess dataset 

i) For training yoloV5 we need to transform the dataset fist

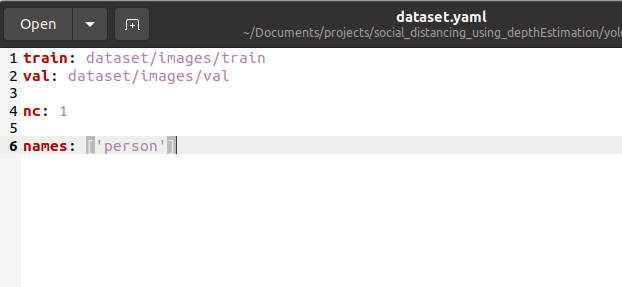
ii) We need to create a dataset in this format

iii) train folder inside the images folder contains training images and the train folder in labels folders contain corresponding training labels.

1. Install all the dependencies

pip install -r requirements.txt

1. Create one dataset.yaml file using the command mentioned below

touch dataset.yaml

1. Dataset.yaml file contains training images path, validation images path, no of classes and list of the names of classes as mentioned in the image.
2. There is four variant of yolov5 model

Small, medium, large, xlarge

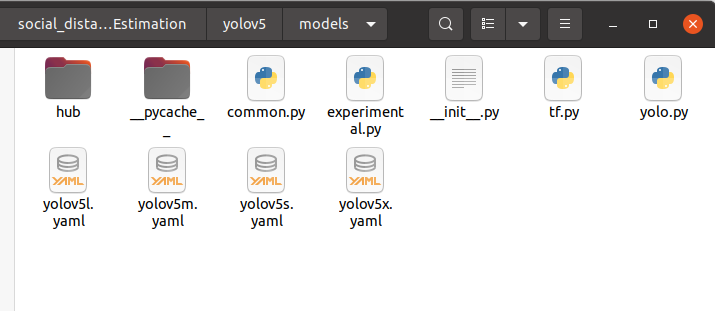
There is separate cfg(configuration file) provided in the models folder.

You need to select variant

Yolov5s.yaml is the configuration file for smallest model variant.

Similarly yolov5m.yaml configuration file for medium size model variant.

So on….



1. For the training model use below command

python train.py --data “provide path of the dataset.yaml file” --cfg “path of selected model configuration file”

Let’s take one example

Dataset.yaml file is present in the same directory and we selected small variant

path od dataset.yaml file dataset.yaml

path of small variant cfg file models/yolov5s.yaml

python train.py --data dataset.yaml --cfg models/yolov5s.yaml

1. After training the model, we can test model

python detect.py --source 0 # webcam

file.jpg # image

file.mp4 # video

path/ # directory

path/\*.jpg # glob

'https://youtu.be/NUsoVlDFqZg' # YouTube

'rtsp://example.com/media.mp4' # RTSP, RTMP, HTTP stream