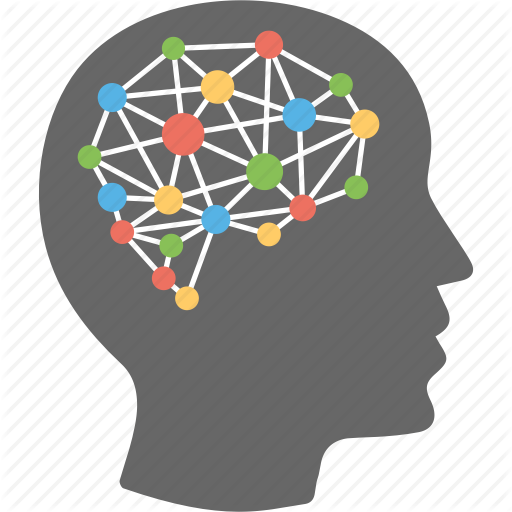
MACHINE LEARNING

Darknet , Caffe & Tensorflow



**ADARSH K SUDARSAN**

### DARKNET

**Darknet** is an open source neural network framework written in C and CUDA. It is fast, easy to install, and supports CPU and GPU computation.

# Data Preparation

**STEP 1**: Collect enough images in each class

Download **label\_tool** tool from this [link](https://github.com/adarshksudarsan/MACHINE_LEARNING) and prepare the environment.

**STEP 2**: Run main.py script of Yolo annotation tool & label the images

Create .txt-file for each .jpg-image-file

**STEP 3**:Run convert.py

Converts coordinates to yolo format.

**STEP 4**:Place all images and files under same folder and run process.py.

This will generate **test.txt** and **train .txt.**

# PRE\_TRAINING

Before start training we need to make some config files.

**STEP 1**: Create a folder under DARKNET root. (Say “hello”) also create corresponding files under “hello”.

* hello/obj.data
* hello/obj.names
* hello/yolo-obj.cfg

**Obj.data should contain:**

classes= 2 //number of classes

train = <path of train.txt>train.txt

valid = <path of test.txt>test.txt

names = obj.names

backup = hello/

**obj.names should contains:**

<names of classes in order>

Say:

Cat

Dog

**STEP 2**:copy a config file from DARKNET root /cfg to hello and do the required edits

**Line 3**: set batch=24, this means we will be using 64 images for every training step

**Line 4:** set subdivisions=8, the batch will be divided by 8 to decrease GPU VRAM requirements.

**Line 244**: set classes 2, the number of categories we want to detect

**Line 237**: set filters=(classes + 5)\*5 in this case filters=35

# TRAINING

**STEP 1:**Tostart training, YOLOv2 requires a set of convolutional weights.DOwnload a suitable convolutional weight.

**STEP 2: $**./darknet detector train hello/obj.data hello/yolo-obj.cfg darknet19\_448.conv.23

**//**darknet19\_448.conv.23 <convolutional weight used>

//weight file will be generated inside folder “hello”

When completed 100 iterations it will automatically store weights file and kill the process once the average loss is less than 0.06.

# TEST

**To test the model:**

$./darknet detector test hello/obj.data hello/yolo-obj.cfg <weight file> *<*sample.jpg*>*

### 

### 

### CAFFE

**Caffe** is a deep learning framework made with expression, speed, and modularity in mind. It is developed by Berkeley AI Research ([BAIR](http://bair.berkeley.edu)) and by community contributors.

# DATA PREPARATION

**STEP 1**: Collect enough images in each class.

We have used label\_tool previously ,for caffe we need .xml format(after annotation NB:.txt files are in the case of yolo).

So download another tool from [here](https://github.com/tzutalin/labelImg).

Or use yolo label\_tool itself and use this [script](https://github.com/adarshksudarsan/MACHINE_LEARNING/tree/master/yolo_to_caffe) to convert to xml format.

Or

If there is no need of annotation

STEP 1:Download your data set.(here we are taking an example of classification of dogs and cat images using AlexNet {dataset from [kaggle](https://www.kaggle.com/c/dogs-vs-cats/data)}).

Create the following folders

1:caffe/examples/DogsCatsKaggle

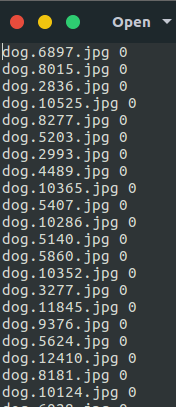
2:caffe/data/DogsCatsKaggle/train

Place the dataset under caffe/dataDogsCatsKaggle/train:

STEP 2:create test and train files:

create [create\_kaggle\_txt.py](https://github.com/adarshksudarsan/MACHINE_LEARNING/blob/master/caffe_files/create_kaggle_txt.py) inside caffe/examples/DogCatsKaggle and Run from there.

This will create train.txt and test.txt.under caffe/data/DogsCatsKaggle



STEP 3: Converting data to LMDB format.(this format makes the network easy to handle data)

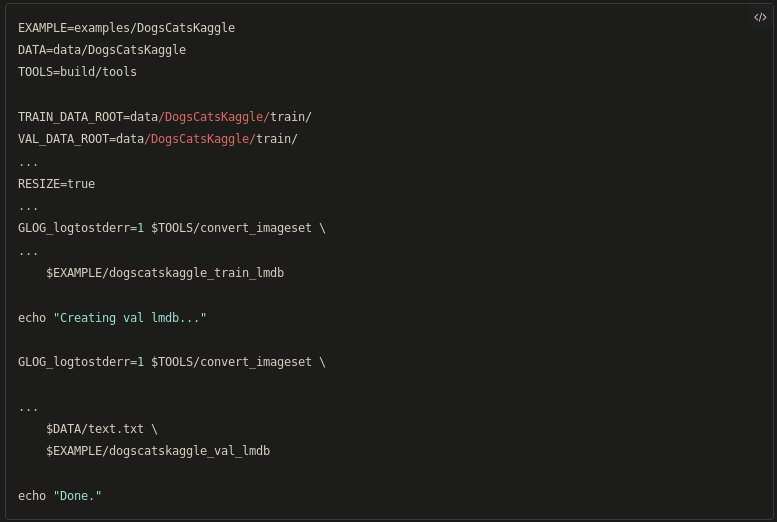
The necessary codes to convert to LMDB are already there in caffe folder.

We just need some modifications according to data set we are using.

Copy the required shell script using the command.

|  |
| --- |
| $ sudo cp examples/imagenet/create\_imagenet.sh examples/DogsCatsKaggle/ |

Make the following changes as shown in the figure below:



Run the script inside caffe folder.

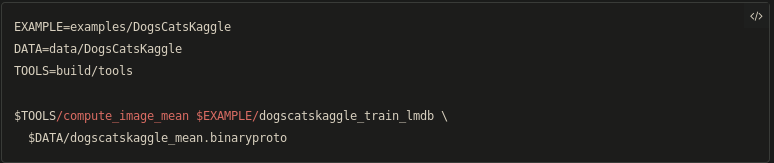
|  |
| --- |
| $./examples/DogsCatsKaggle/create\_imagenet.sh |

STEP 4:Make the mean image

Copy the required script :

|  |
| --- |
| $ sudo cp examples/imagenet/make\_imagenet\_mean.sh examples/DogsCatsKaggle/ |

Make the following modification:



Run the script from caffe folder:

|  |
| --- |
| $ ./examples/DogsCatsKaggle/make\_imagenet\_mean.sh |

# PRE TRAINING:

Make a folder caffe/model/dogscatskaggle\_alexnet

Copy some required files:

|  |
| --- |
| $ sudo cp bvlc\_alexnet/solver.prototxt dogscatskaggle\_alexnet/  $ sudo cp bvlc\_alexnet/train\_val.prototxt dogscatskaggle\_alexnet/ |

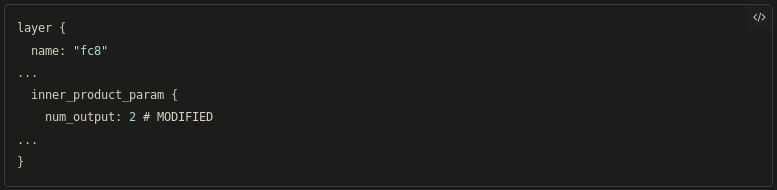
Make modifications to solver.prototxt as shown in figure:



Then make some modifications to train\_val.prototxt as shown in figure below:



For the last layer fc8:replace 1000 with 2(since we are only using 2 kinds of images)



# TRAINING

Run the code below:

|  |
| --- |
| $ ./build/tools/caffe train --solver=models/dogscatskaggle\_alexnet/solver.prototxt |