Dimension Detector

Bachelor of Science in Software Engineering
San Jose State University
December 2017

Members:
Paul Diaz
Kito Mam
Sheethal Mathew
Justin Passanisi

Advisor:
Dr. David Anastasiu

Table of Contents

1. Chapter 1 Project Overview

Dimension Detector is a standalone application that detects different kinds of objects from a photo, such as doors, buildings and calculates its dimension. Dimension Detector is implemented using technologies and languages such as Python3, TensorFlow, CUDA, YOLO framework, and Darknet. These are used to aid the system in training for object detection. Implementation is divided into four different parts: building GUI, building data sets, which includes photo annotation, system training, and implementation of algorithms for building dimension calculation. Dimension Detector's interface is implemented using Python3, and is tested and executed on Windows, MacOS, and Linux. The team is using VGG annotator, an open source web-based annotator, to mark up different objects from photos. The team also implemented a code to extract information from JSON files into text files (JSON files are provided by VGG Annotator as an output of annotation). These text files are important pieces of data that are needed for training the system. The third implementation is training the system with Darknet YOLO v2 framework. The final implementation is to execute the three different estimations of building dimension calculation identified by the system. A Titan XP GPU will be aiding our implementation of a YOLO v2 framework for faster training of data and object recognition.

1. Chapter 2 Requirements

a. Hardware

	NVIDIA CUDA NVIDIA CUDA is a parallel computing platform and programming model created by NVIDIA. This technology will be one of the key technology to help in computational performance of training by utilizing the GPU (TitanXP).
	Linux Linux is a computer operating system family, and one of the most prominent examples of free software and open source development. Its underlying source code can be modified, used, and redistributed by anyone, freely.

b. Software

	Python 3.* + Python is an object-oriented, interpreted, and interactive programming language created in the late 1980's. Python 3 is the new version of the Python language released on December 3, 2008.
	Python's Pillow Imaging Library (PIL) Pillow Imaging Library (PIL) is a library that can used to process images to the Python interpreter. This library supports different file formats, and provides powerful image processing and graphics capabilities.
	TKinter (Tk) TKinter is Python's de-facto standard Graphical User Interface toolkit package. It is a thin object-oriented layer on top of Tcl/Tk.
	YOLO v2 YOLO v2 is framework technology that uses convolutional neural networks to train images. YOLO v2 works using a neural network to learn the connection between the input and the output of the object of interest.
	Darknet Darknet is an open source neural network framework written in CUDA. Darknet enables object detection from images using a pre-trained models, the weights used in YOLO v2.
	TensorFlow TensorFlow is an open source software library for numerical computation using data flow graphs. It was developed by Google in 2015 to make it easier for developers to design, build, and train deep learning models.
	VGG Image Annotator VGG Image Annotator is web-based online tool used for annotating all the objects of interest in a photo. These objects of interest will be identified and isolated by inboxing and labeling each one. Photo annotation creates a textual description of the objects identified, which includes pixelheight, pixelwidth, pixelheight and pixelwidth of the object contained in the box.

c. Other Requirements

There must be no text file other than the ReadMe.txt file in the root folder inside zip file.

2. Chapter 3 Installation

a. Python 3 Installation

Linux and UNIX operating systems generally include with the latest Python Version. For Windows and Mac computers, Python installation tutorials can be found at https://www.python.org/downloads/. For this project, we are using Python version 3.

b. Python's Pillow Imaging Library (PIL) Installation

Installation of Python generally include the library Pillow. If need PIL to be installed, download tutorials can be found at https://pypi.python.org/pypi/Pillow/2.7.0.

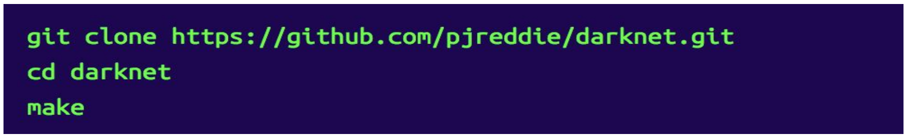
c. TKinter Installation

If Python version 3 is installed in Linux, UNIX, and Windows operating system, TKinter is included in this version. Developers can immediately use the library by explicitly include `from tkinter import *` in the program. For Mac users, whether Python 3 is installed, Tkinter needs to be separately installed. Download tutorials can be found at https://www.python.org/download/mac/tcltk/.

d. Darknet Framework Installation

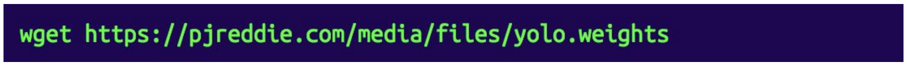
To install Darknet Framework:

1. Clone github Darknet git repository, which is https://github.com/pjreddie/darknet. This can be done by using Terminal or Command Prompt by:



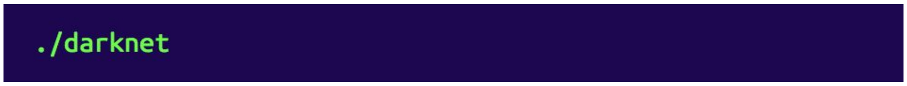
```
git clone https://github.com/pjreddie/darknet.git
cd darknet
make
```

Download pre-trained weights file by running the command below:



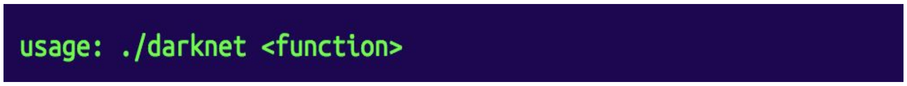
```
wget https://pjreddie.com/media/files/yolo.weights
```

2. To check if installation is successful, run the command below.



```
./darknet
```

3. Output should be:



```
usage: ./darknet <function>
```

For more installation details and information, please go to https://pjreddie.com/darknet/install/.

e. **YOLOv2 Installation**

Darknet Framework comes with YOLO v2, therefore YOLOv2 installation is not necessary.

f. **TensorFlow Installation**

TensorFlow is built and tested on 64-bit computers operating systems:

- Mac OS 10.11 (El Capitan) or later
- Ubuntu 14.04 or later
- Windows 7 or later

Installation tutorials for any of the operating system mentioned above can be found at https://www.tensorflow.org/install/.

3. Chapter 4 Run Programs

a. Dimension Detector GUI

User interface of Dimension Detector is the only part that user interacts with when using the program. On the left side of the window, user can upload an image using the top button for object detection and dimension calculation. On the top right side of the window shows the original image and by clicking the button below initiates the detection of specified objects in the image uploaded. Once the image with objects detected is loaded, user can click on button at the very bottom to find the measurement of a building of interest.

To run/open up GUI program for Dimension Detector, locate in the root file and double click on ***DimensionDetector.py,*** or using terminal or command prompt, run the command ***python DimensionDetector.py.***

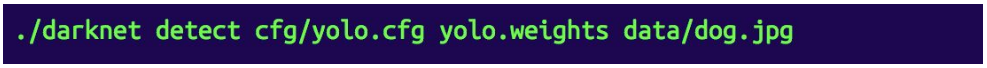
To upload a photo from local device, click on the right side top button. This should load the image selected on the UI's body frame and initiate object detection and building dimension calculation.

To view the original uploaded image, click on the top right button.
To view the photo with labeled images detected, click on the middle button right side of the window. To view the photo with building with measurements click on button on bottom right, this populates an image on center of the frame with all the

objects and the buildings recognized and size displayed at the top of bounding box around building.

b. Darknet YOLO Object Detection

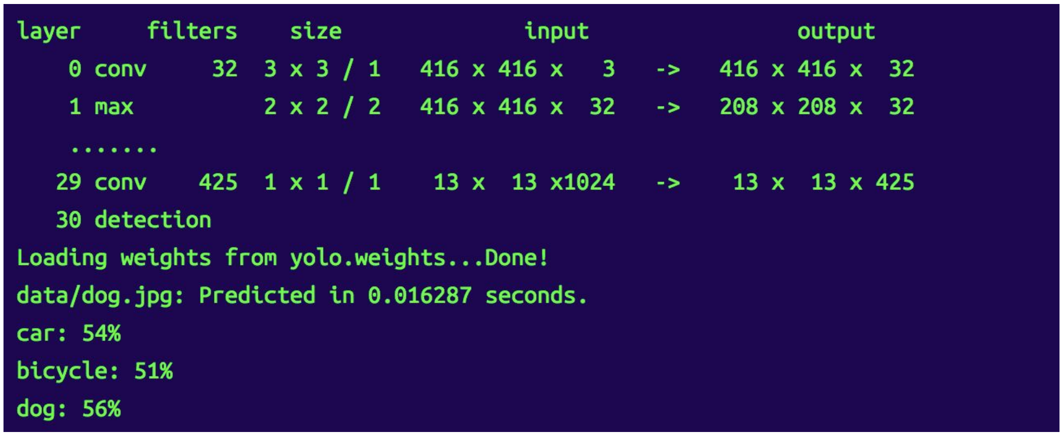
To run the detector, locate the directory where the all the files located. Then run the commands below:



```
./darknet detect cfg/yolo.cfg yolo.weights data/dog.jpg
```

Note: dog.jpg is the target photo to be analyzed for object detection. This photo must be located in the data file.

While running, you will see the following in the terminal or command prompt:



```
layer     filters    size              input                output
    0 conv     32  3 x 3 / 1   416 x 416 x   3   ->   416 x 416 x  32
    1 max          2 x 2 / 2   416 x 416 x  32   ->   208 x 208 x  32
    .......
   29 conv    425  1 x 1 / 1    13 x  13 x1024   ->    13 x  13 x 425
   30 detection
Loading weights from yolo.weights...Done!
data/dog.jpg: Predicted in 0.016287 seconds.
car: 54%
bicycle: 51%
dog: 56%
```

The program creates a copy of the original photo labeled with the objects detected. Result prediction will be called as “predictions.png.” This photo will be found in “darknet” Folder.

For more training details and information, please go to https://pjreddie.com/darknet/install/.