**Mini Project Report on**



**FACE DETECTION USING DEEP LEARNING**



**Submitted in partial fulfilment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Ishmit Singh** **2018847**

***Under the Mentorship of***

**Mr. Ankit Gupta**

**Associate Professor**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**July 2023**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Face Detection using Deep Learning”** in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Mr. Ankit Gupta, Associate Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

**Diagram

Description automatically generated with low confidence**

Ishmit Singh 2018847

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | **1** |
| Chapter 2 | Literature Survey | **4** |
| Chapter 3 | Methodology | **6** |
| Chapter 4 | Result and Discussion | **9** |
| Chapter 5 | Conclusion and Future Work | **10** |
|  | References | **12** |

**Chapter 1**

**Introduction**

Real time Human detection is based on a popular field of study which is known as Computer Vision[1]. In this chapter the various aspects of computer vision, meaning of keywords used throughout the report like detection, enumeration and a summary of the concept used for this project is laid down.

* 1. **Computer Vision**

Computer Vision[1] is an interdisciplinary scientific field that deals with how computers can gain high-level understanding from the use of images and videos. In it, computers are trained in such a way that they can learn how to derive meaningful information from digital images, videos and other sorts of visual inputs given by the user.[1]



Different types of computer vision include image segmentation, object detection, facial recognition, edge detection, pattern detection, image classification, and feature matching.

Computer Vision in itself is a big domain and can be divided into many subdomains like scene reconstruction, object detection, event detection, video tracking, object recognition, 3D pose estimation, learning, indexing, motion estimation, visual serving, 3D scene modelling, and image restoration.[2]

* 1. **Applications of Computer Vision**

It has various applications[1] in all sorts of fields.

* Object Detection
* Screen Reader
* Intruder Detection
* Code and Character Reader
* Robotics
* Motion Analysis
* Image Restoration

In this project, from the vast topics that come under the topic of computer vision, I have used Object Detection[2]

* 1. **Detection and Enumeration in Computer Vision**

Detection is a technology related to computer vision and image processing that deals with detection of semantic objects present in a digital input (like humans, cars or buildings)

In Computer Vision there are a lot of methods of detection with varying amounts of accuracy in them according to their advancement levels. The ones that were invented at a very early stage produce many false detection cases as compared to the advanced methods that are now available.

Enumeration is the process of making a complete ordered list of all the objects that have been detected by the use of a method.

In this project I have used the human as an entity which is being detected by my project and along with it, it is also enumerating all the humans that are being detected through the digital media like image, video and camera footage.

* 1. **Face Detection**

Finding every occurrence of a human face in an image is the objective of face detection, which is most commonly completed by scanning the entire image at all feasible scales and comparing a small region at each location with templates or patterns of individuals.

This allows us to utilize a variety of established ways to identify faces in any image or video and to gather information on things like accuracy, the number of faces detected, and other things.

Some common methods are :

- **Using Haar Cascade Classifier**:[2]

o Here we make use of .xml file for face detection and using that we detect the faces in real time videos and images.

- **Using HOG(Histogram of Oriented Gradients)** :[2]

o Here we make use of predefined functions and with that we detect, and this case gives somewhat better accuracy as compared to Haar Cascade Classifier.

- **Using TensorFlow**:[3]

o TensorFlow is an open-source API from Google, which is widely used for solving machine learning tasks that involve Deep Neural Networks. And again, this method gives even better accuracy than the above two methods.

Here we have implemented the application using the third method and got almost the better accuracy.

**Chapter 2**

**Literature Survey**

J.Redmon et al.(Redmon,2016),introduced YOLO, a unified model for object detection. This model is simple to construct and can be trained directly on full images.This model may be built quickly and trained using only full photos.YOLO is trained on a loss function that directly relates to detection performance, in contrast to classifier-based techniques, and the entire model is trained concurrently.Fast YOLO is the literature's fastest general-purpose object detector and pushes the boundaries of real-time object detection. Additionally, YOLO generalises effectively to new domains, which makes it the best choice for applications that require quick, reliable object detection.[4]

Aishwarya Sarkale (2018) et al. proposed that humans are quite good at differentiating items through eyesight. However, object detection is a problem for machines. The area of computer science has now been brought to neural networks. Artificial neural networks are another name for neural networks. Artificial neural networks are computer representations of the brain that aid in the identification and detection of objects. This study describes and illustrates the accuracy of many neural network types, including ANN, KNN, FASTER R-CNN, 3D-CNN, and RNN. The accuracy of various Neural Networks is examined and compared through the analysis of numerous research articles, and it can be said that in the test cases provided, the ANN provides the highest accuracy for object detection.[5]

Richard Socher (2018) et al. proposed recent advances in 3D sensing technologies make it possible to easily record colour and depth images which together can improve object recognition. For this new 3D modality, the majority of current approaches rely on extremely well-designed features. We provide a model for learning features and categorising RGB-D images that combines convolutional and recursive neural networks (CNN and RNN). Low-level, translationally invariant characteristics are learned by the

CNN layer and then used as inputs by several fixed-tree RNNs to create higher order.

features. Convolution and pooling are combined into one effective, hierarchical operation by RNN scanning. The primary finding of our study is that even RNNs with random weights can create effective features. Our model outperforms comparable architectures.

like two-layer CNNs in terms of performance on a common RGB-D object dataset while also being more precise and quicker during training and testing.[6]

Karanbir Chahal (2018) et al. proposed Object detection is the identification of an object in the image along with its localization and classification. It has a wide range of uses and is essential to vision-based software systems. This study aims to do a thorough analysis of current deep learning-based object detection techniques. Various algorithms, quality measures, speed/size trade-offs, and training approaches are among the subjects covered in the survey. The SSD class of single step detectors and the Faster R-CNN class of two step detectors are the two object detection algorithms that are the subject of this paper. The exploration of novel light weight convolutional base architectures also addresses methods for building portable and quick detectors on low powered devices. In the end, a thorough analysis of each detector's advantages and disadvantages brings us to the current state of the art.[7]

R. Sujeetha (2019) et al. proposed object detection and tracking may be a significant, energetic, but unreliable and developing topic in computer vision. Due to the tracking modules' widespread use in official surveillance, security applications, and many other uses, researchers have developed a variety of optimised and specialised ways. However, there are issues with implementing object detection and tracking in real-time, such as tracking in real-time and providing suitable optimised results, using overly dynamic computation to find the efficient performance with respect to time factor, or having multiple objects to track making this task more challenging. Although many strategies have been developed, there is still much room for improvement. Nevertheless, during this literature review, we have seen some impressive and varied approaches to object identification and tracking. In this technique, we'll use the TensorFlow and OpenCV

libraries, the CNN algorithm, and label the discovered layers while simultaneously checking the labelling accuracy. Live input video will be obtained for the same thing for validation purposes, where objects will be detected. It is possible to recreate the same thing for real-time use by adding external hardware. The correct, efficient, and optimised algorithm for object tracking and detection can be seen in the conclusion.[8]

**Chapter 3**

**Methodology**

**3.1 TensorFlow with Deep Learning[3]**

TensorFlow is an open-source toolkit for numerical computation and large-scale machine learning that was developed by the Google Brain team and was made available to the public in 2015. By combining a variety of deep learning and machine learning models and techniques (also known as neural networks), TensorFlow makes them useful by the method of common programming metaphors.[3]

Dataflow graphs—structures that depict how data flows across a graph, or a collection of processing nodes—are made possible by TensorFlow and can be created by programmers. Each link between nodes, or edge, in the graph, or tensor, is a multidimensional data array. Each node in the graph represents a mathematical process.

**3.2 Working of TensorFlow API**

In this project, TensorFlow API v2[3] is used to detect Humans in digital media through the process of MobileNet SSD with Deep learning. Since there are multiple approaches for using TensorFlow Deep learning one was used since Deep Learning uses convolutional neural networks to identify objects and their location. Since these networks are unsupervised, the system identifies humans on their own as well as their areas making the process much faster and easier to use.

This Process works in this way[3]

MobileNet SSD runs on a base layer known as MobileNet with several convolutional layers. Since it's a single convolution base layer with multiple other layers, the architecture detects objects at a single pass by identifying the locations of the bounding boxes and not the box shape itself. Each listed bounding box in MobileNet SSD contains.

* the offset information of the box in four corners (cx, cy, w, h) and
* the probabilities of C class (c1, c2, …cp).

In this project I have set the Threshold accuracy for the detection process as 80% ,so the object detected with accuracy more than the threshold accuracy, we declared it as the well detected human, and display detection indicator around that human during the process. I have set this threshold to prevent false detection to det displayed while the detection process.

**3.3 Working of the Project**

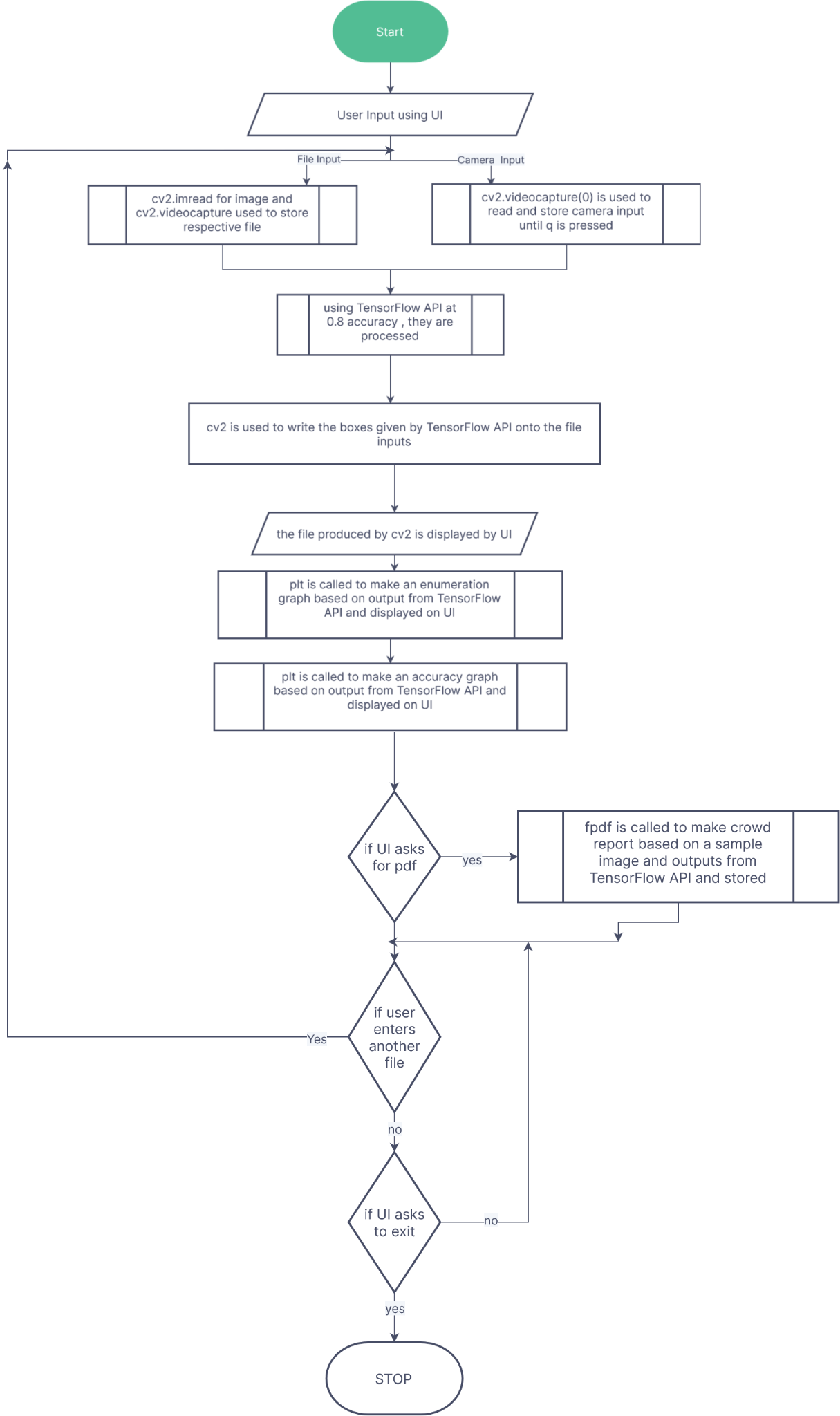
Many Libraries are used in this project. These include -

* **Custom Tkinter[9] -** for UI
* **Tkinter[10]-** for UI
* **PIL[10] -** using ImageTk and image to input image.
* **OpenCV[2] -** Interaction with UI and TensorFlow API to create the outputs of the images and videos by using PIL. also interacts with matplotlib for making plots. holds the whole program together.
* **TensorFlow[3] -** To detect the humans in the given input.
* **Matplotlib.pyplot[10] -**  To use tensorflow output and make accuracy and enumeration graphs.
* **argparse [10]-** to pass arguments from UI.
* **fpdf [10] -** to generate a crowd report pdf of answer.

**3.3.1 Flowchart on the project**

Assumptions in the flowchart

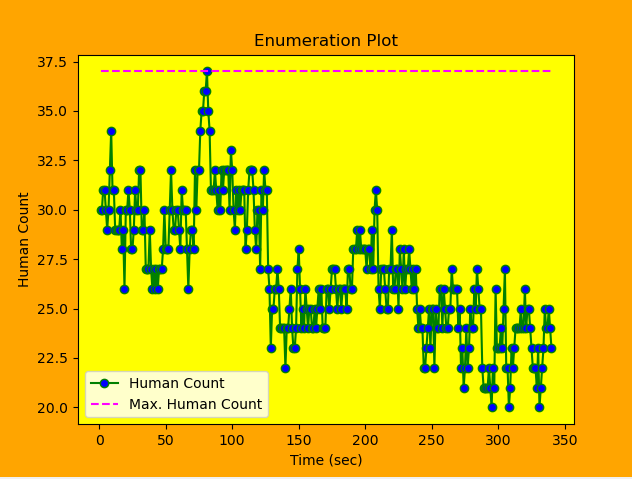
1. open cv is used as cv2
2. Matplotlib.pyplot is used as plt

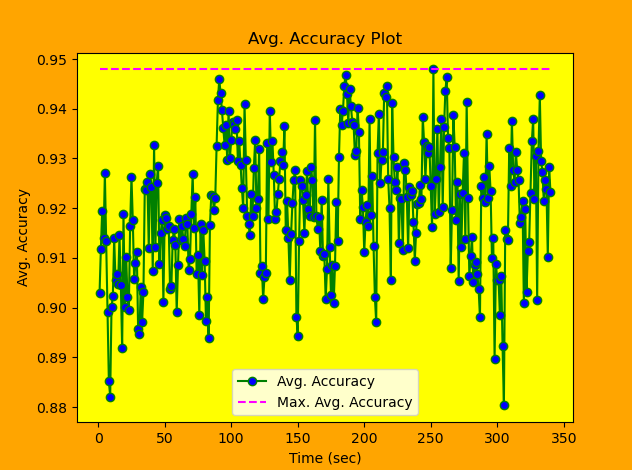


**Chapter 4**

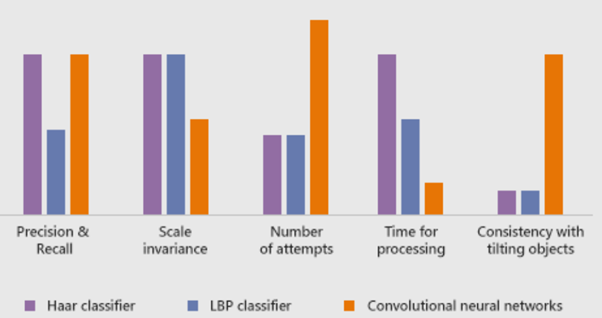
**Result and Discussion**

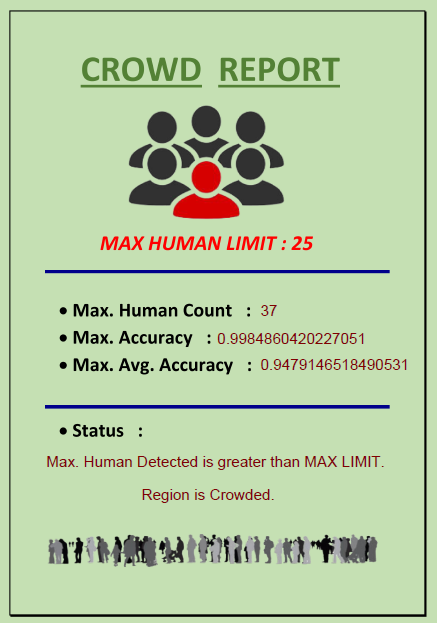
By using the methodology on an example case of an 11 second video consisting of the maximum number of faces as 37 at a given point of time in the video, the accuracy and enumeration graphs produced are [9][10]-





By analysing these graphs, it is determined that the project is indeed very accurate with an mean highest accuracy of 0.9479 thanks to the use of TensorFlow and MobileNet SSD with Deep learning[3]. In contrast to the other methods Haar Cascade[2] and Histograms of Oriented Gradients[2] this is much better as their accuracy had come out to be around 0.7 in the best-case scenario.



As data collected from different approaches , TensorFlow (Convolutional neural Networks) with OpenCV might be the slowest among all the methods but the payoff is much better than any other method.

A pdf generated by the project in the end encapsulates the gist of the project’s output.

**Chapter 5**

**Conclusion and Future Work**

In the last section of the project, we generate a Crowd Report, which will give some message based on the results we got from the detection process. For this we took some threshold human count, and we gave different messages for different results of human count we got from the detection process.

Now coming to the future scope of this project or application, since in this we are taking any image, video or with camera we are detecting humans and getting count of it, along with accuracy. So, some of the future scope can be :

- This can be used in various malls and other areas, to analyse the maximum people count, and then provide some restrictions on the number of people to have at a time at that place.

- This can replace various mental jobs, and this can be done more efficiently with machines.

- This will ultimately lead to some kind of crowd-ness control in some places or areas when implemented in that area.

In the future, I would like to work on this again to work on making this whole process faster since currently it can be a bit slow.

**References**

[1] Jan Erik Solem “Programming Computer Vision with Python”,1st ed, O'Reilly Media,

Inc. ,2012

[2] Gary Bradski, “Learning OpenCV: Computer Vision with the OpenCV Library”,1st

ed,O′Reilly,2008

[3]Giancarlo Zaccone, Md. Rezaul Karim and Ahmed Menshawy “Deep Learning with

TensorFlow” ,1st ed , Packt Publishing,2017

[4] J. Redmon., S. Divvala, R. Girshick, & A. Farhadi, “You Only Look Once:

Unified, Real-Time Object Detection”. In IEEE Conference on *Computer Vision*

*and Pattern Recognition (CVPR)*,2016

[5] Aishwarya Sarkale, Kaiwant Shah, Anandji Chaudhary, Tatwadarshi P. N., “A

Literature Survey: Neural Networks for Object Detection”, *VIVA-Tech International*

*Journal for Research and Innovation,* Vol. 1, No. 1, Article No. 9 , 2018

[6] Richard Socher, Brody Huval, Bharath Bhat, Christopher D. Manning, Andrew Y. Ng,

“Convolutional-Recursive Deep Learning for 3D Object Classification”, International

Conference on *Computational Intelligence and Communication Networks*, 2018.

[7] Karanbir Chahal1 and Kuntal Dey, “A Survey of Modern Object Detection Literature

using Deep Learning”, *International Research Journal of Engineering and Technology*

(IRJET), vol. 8, no. 9, 2018

[8] R. Sujeetha, Vaibhav Mishra , “Object Detection and Tracking using Tensor Flow”,

*International Journal of Recent Technology and Engineering* (IJRTE),vol.8, no.1,

May 2019.

[9] TomSchimansky /Custom Tkinter [Online]. Accessed on 15th May 2023:

<https://github.com/TomSchimansky/CustomTkinter>

[10] Frederik Lundh,”Python Standard Library: An Annotated Reference for Python 2.0”,1st ed, O'Reilly Media, Inc.,2001