**Abstract**

In the Disk Diffusion Antibiotic Sensitivity test a thin film of bacteria applied on a plate is subjected to various antibiotics. The Zone of inhibition is a circular area around the spot of the antibiotic in which the bacteria colonies do not grow. The zone of inhibition can be used to measure the susceptibility of the bacteria towards the antibiotic. The process of measuring the diameter of this Zone of Inhibition can be automated using Image processing. In this work we used 100 of images to train a CNN model using YOLO multi-object detection methodology and Computer Vision, which will detect the zones of inhibition of the bacteria. This work demonstrates an effective approach of measuring the Zone of Inhibition by calculating the radius of the zone by drawing contours and setting the right value of threshold.

**INTRODUCTION**

The growth of bacterial colonies is usually affected when under the impact of an antibiotic. Some of the bacteria, though, may have developed tolerance to some of the antibiotics. Then a laboratory examination called as the Kirby-Bauer test is conducted to determine growing antibiotics should be used to successfully treat a patient [1]. The Kirby-Bauer method or the antibiotic susceptibility method for disk diffusion is used to determine that an antibiotic is successful for the treatment of specific bacteria. The test sample is normally gathered from any patient known to have been infected. On a petri-dish a particular component of the obtained sample is spread evenly. A portion of filter paper infused with a few various antibiotics would then be placed on the plate. This configuration is provision of quality services in agar. The antibiotic absorbs into agar and after that the density reduces radially from the point where the antibiotic was applied outwards. If the antibiotic is said to be effective against the particular concentration of the bacteria, the bacteria will not grow if the concentration of the agar at that point is greater than the effective concentration. This area of no bacterial growth is recognized as the Inhibition Zone. The area that has been the Zone of Inhibition without bacteria living should be in a different color relative to many other bacterial growth regions. The differentiation is labeled and clearly visible to the naked eye. Measuring the diameter of this Inhibition Zone will be consistent if the antibiotic is effective for treating the patient or not. The greater the diameter the antibiotic efficacy will be. The exact range of the diameters in the Clinical & Laboratory Standards Institute is well standardized,documented, and maintained.

The diameter value has to be checked against the prescribed values and correlated. Whether such a bacteria are susceptible or not, that conforms. Since the color of the Inhibition Zone is different, Image processing can be used to detect the different "circular" zones available. The identification is not easy, since many test cases need to be protected. A different algorithm has to be used to find overlapping regions. Another aspect is that there are two types of bacteria, one aerobic and the other anaerobic. It is said that the aerobic bacteria are gram-positive, and that the anaerobic bacteria are gram-negative. Therefore stains must be added. The main contribution of this paper is the presentation of an effective algorithm to measure the diameter of the bacteria's inhibition zone to a specific antibiotic that covers all possibilities including extreme cases such as overlapping and jagged edges

**REVIEW OF LITERATURE**

Man's symbiotic relationship with plants over time has yielded many invaluable benefits to the world. Besides the raw resources that form our variety of foods, medicines , cosmetics and flavor products, as well as other pharmaceuticals, are the most important plant products. (1996 Sofowora).

For thousands of years, nature has been a source of medicinal agents and an impressive number of modern drugs have been derived from natural sources, many of which were based on the uses of agents in traditional medicine (Cragg and Newman 2001). Herbal Medicine is defined as a branch of science where plant-based formulations are used in disease alleviation. It is also classified as phytomedicine or botanicals. Phytotherapy recently was adopted as a more valid synonym for herbal or botanical treatment (Singh 2007).

There is a revived interest in natural medicine today, and an growing appetite for more medicines from plant sources is turning researchers' eyes to herbal goods, aiming to produce stronger medicines against pathogenic microbial strains. This revival of interest in plant-derived drugs is mainly due to the current widespread belief that "green medicine" is safe and more reliable than the expensive synthetic drugs, many of which have adverse side effects such as allergies, anaemic problems, hair loss, liver damage, kidney failure and very expensive, which people can not afford (Parekh and Chanda 2006).

Today Infectious diseases in highly industrialized countries are a secondary but significant cause of death and the primary cause of death in many less developed countries (Iwu 1999). Our environment contains a wide range of infectious diseases but not all of them cause diseases. Five major forms of infectious agents exist: bacteria, protozoa, fungi , viruses, and helminths. Majority of human illnesses was the product of bacterial infections. The disease-causing bacterial agents include Escherichia coli spp., Pneumococcus spp., Staphylococcus spp., Mycobacterium spp., Vibrio spp. and Salmonella spp. Neisseria spp., Bacillus spp. and Treponema spp.

D'Amato et al.[3]'s paper called 'Current application of digital picture processing to biochemical and antimicrobial susceptibility research' discusses the roles of doing visual analysis and digital imaging antibiotic susceptibility studies. In this paper they find that image processing techniques would achieve comparable precision when opposed to visual inspection. It states that the Electronic Laboratory Imaging Machine, ALADIN, which has an associated image sensor, is able to interpret the picture as precisely as the human eye can.

Stager and Davis' paper, entitled "Automated Microorganism Identification Systems,"[4] has done a good job of identifying few other automated systems currently produced for the detection and measurement of antibiotic resistance. One of the automated systems listed in their paper, named Vitek GNI, made use of identification cards and turbidimetry monitoring to conduct susceptibility tests at a very high precision rate of 97.4 percent. Another automated method , known as the WalkAway device described in the same article, uses a photometer or flurometer to identify the bacteria's growth and development. Although this study has continued to produce successful results.

Jean B. Patel , Phd, D(ABMM) et al. in “Performance Standard for Antimicrobial Disk Susceptibility tests”explain about series of procedures to standardize the way disk diffusion tests are performed. Also defined are the efficiency, applications and limitations of the methods suggested by then current CLSI-. This norm provides the latest up-to - date knowledge for product discovery, definition, and QC use systematic procedures.

A computational approach to edge detection. The approach's success depends on defining a comprehensive set of goals for edge point computing. These targets must be sufficiently precise to delimit the detector's desired behavior while making minimal assumptions about the solution 's form. For a class of edges, we define detection and localization criteria, and present mathematical forms as functional on the operator's impulse response for those criteria. Thereafter a third requirement is applied to guarantee that the detector only has one answer to a single point.

Antibiotic resistance research was conducted for all isolates using cefoxitin, gentamicin and rifampicin disk diffusion research and HiComb MIC scanning for vancomycin, erythromycin, ciprofloxacin, sulphamethizole, and teracycline. An increase in the staphylococcal isolate resistance, particularly the negative coagulase, has been noted. No statistically important differential was observed between BF forming and non-BF Staphylococci, however, 48.8% of BF forming strains were immune to methicillin (MR) and all were infected with multiple antibiotic resistance as 16% were immune to 3 antibiotics, 32% to 4 antibiotics, 39% to 5 antibiotics and 13% to all antibiotics studied. MR staphylococcal isolates were detected with high resistance to erythromycin (92.9%), gentamicin (60.7%), and ciprofloxacin (89.3%), showing a statistically significant difference compared to Staphylococci, which is sensitive to methicillin (63.6%, 9%, 66.7%, respectively). Finally , we conclude that the susceptibility of staphylococcal isolates to specific antibiotics is enhanced, and the development of BF is consistent with antibiotic multiresistance.

Research purposes are to identify emerging drug resistance in specific pathogens and to ensure medication tolerance for particular infections of choice. The most commonly employed forms of research include techniques for broth microdilution or fast automatic instrumentation, utilizing professionally available products and equipment. Manual methods to provide versatility and possible cost savings include the methods of disk diffusion and gradient diffusion. Each method has strengths and weaknesses including organisms which the method can accurately test.

**SYSTEM ANALYSIS**

**EXISTING SYSTEM**

Currently there is no computerized or digital system to achieve the results automatically. Each of the plates had to be processed manually by human workers to get the diameters of inhibition zones calculated. It took long time as well as more human resources to get the work done.

**DISADVANTAGES OF EXISTING SYSTEM:**

1. Need more time

2. Human worker dependent

3. Need to perform manually

4. Cost is more

**PROPOSED SYSTEM**

We proposed a CNN based method using YOLO multi-object detection module that can measure and detect the zone of inhibition as well as it can detect the antibiotic plates.

**ADVANTAGES OF PROPOSED SYSTEM**

1. Automatic detection and measurement

2. Independent of Human worker

3. Cost is less

4. Much faster to perform

5. More accurate

**HARDWARE REQUIREMENTS**

System : Pentium Dual Core.

Hard Disk : 120 GB.

Monitor : 15’’ LED

Input Devices : Keyboard, Mouse

Ram : 1GB.

**SOFTWARE REQUIREMENTS**

Operating system : windows 10

Coding Language : PYTHON 3.6

**FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

ECONOMICAL FEASIBILITY

TECHNICAL FEASIBILITY

SOCIAL FEASIBILITY

**ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as Ill within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

**TECHNICALFEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is Income, as he is the final user of the system.

**METHODOLOGY**

A neural network consists of several different layers such as the layer of input, at least one hidden layer, and a layer of output. In object detection, they are best used to recognize patterns such as edges (vertical / horizontal), shapes , colours, and textures. In this sort of neural network, the secret layers are convolutionary layers that serve as a filter which first receives input, transforms it using a particular pattern / feature, and sends it to the next layer. With more convolutionary layers they are changed in different ways each time a new input is sent to the next convolution layer. For example, the filter will define shape / color in a area in the first convolutionary layer, and the next one will infer the object it really is, and the last convolutionary layer will recognize the object as a dog. Essentially, as the data passes through more and more layers, the more complex trends that can be found by potential ones.

There are several variants in the CNN but it typically has three forms of layers: Convolution, pooling, and layers that are completely linked. The convolution layer consists of several convertion kernels that acquire input feature representations and create feature maps. A feature map is the result of converting the input to a learned kernel first, and then applying an element-wise non-linear activation function on the converted results. In mathematical terms, the value of the feature, zli, j, k, at a certain location I j) in the lth layer kth feature map, is

Zli, j, k = wlkTxli (1)

The wlk and blk are the weight vector, and the bias term of the lth layer kth filter, respectively. The kernel wlk is shared by the zl:,:,k, feature maps, for example, which is different from ANNs. This has the effect of reducing the sophistication of the concept and made it easy to learn. If a(•) is called the nonlinear activation function, the action value ali, j , k of the convolutionary feature zl:,:,k, is:

ali,j,k=a(zli,j,k) (2)

The advantage of the activation function is that it introduces to CNN non-linearities which are good for detecting non-linear characteristics. The most frequently used activation function is sigmoid, ReLu, and tanh. The pooling layer, which is usually between two convolution layers, has the role to achieve shift-invariance by minimizing the resolution of the feature maps. For every feature diagram, if the pooling method is pool(•), then:

yli,j,k=pool(alm,n,k),∀(m,n)∈Rij (3)

With Rij being a neighborhood located around I j). It can be average pooling and max pooling operations. The kernels in the first layers typically sense low-level features such as curves or corners, and the higher-level kernels are trained to represent more complex features. The optional full-connected layer at the end of the network has the function of taking all the neurons in the previous layer and connecting them to each of the current layer's single neurons, thereby producing global semant knowledge. The output layer is the last layer of a Convolutional Neural Network, and is normally used as a Softmax operator or SVM.

The loss function, minimised during training, is :

L=1N∑n=1N[l(θ;y(n),o(n))] (4)

where N is the number of the desired input-output relations (x(n),y(n)),n∈[1,...,N] and x(n) is the nth input data, y(n) is its target label, and o(n) is the output of the CNN

Convolutionary neural networks include neurons, like other neural networks, and learn weights and prejudices. CNN design assumes that the inputs are pictures, thereby encoding those properties into them. This increases forward feature performance and reduces the sum of parameters within the network. The distinction from other neural networks is that the CNN have three-dimensional neurons arranged: distance, height, and depth. Unlike in ANN, the neurons in a layer are connected only to a small portion of the neurons in the previous layer.

#### NETWORK ARCHITECTURE

Core of any CNN is the convolution layer. It is used to extract information from its input by using a number of filters, which are taught automatically to detect certain features in an image. The user determines the size of the filters, and their numbers. -- filter can search from the upper left side corner to the bottom right side corner via the data, thus generating a characteristic diagram. The neurons at the output are distributed in a volume equivalent to the number of filters, a height equivalent to hi−hf+1 (where hi is the height of the input and hf is the height of the filter), and a duration equal to li−lf+1 (where li is the duration of the input and lf is the length of the filter). As more convolutionary layers are connected in series, one such layer 's output becomes another's input, and its features are extracted again increasing the complexity level and hence the accuracy, but also increasing the training time and the risk of overfitting.The pooling layer performs the down-sampling in the width and height, reducing the dimensions of its input and, hence, reducing the number of parameters to be computed. This reduces the complexity of the network and the possibility of overfitting.

The pooling procedure works independently on every input depth slice, down-sampling them all in the same way. In specifying the pooling layer, each of the slices will be divided into a number of patches, equivalent in region to the filter size defined by the consumer. The most widely used filter size is (2,2), thereby splitting each slice into a number of neighboring yet disjunct patches of 2 neurons high and 2 neurons wide. The performance of the pooling layer would be a lower volume but the input will be equal in size.

* Average pooling: measures the number average within every patch and transfers it to the correct output location.
* Max Pooling: The limit for each patch is sent to the outputThis method has been seen to have improved efficiency and has been used in all the CNN pooling layers in this work.
* The flat layer is used to adjust the structure of the signal, rendering it an collection of 1 neuronal depth and height equivalent to the commodity duration between the range, depth and height of the input sheet. This layer is used in each CNN because a one-dimensional vector must be the output layer.
* The dropout layer is used to reduce overfitting by arbitrarily trimming a portion of the network nodes. This spontaneous decay of neurons in the network can be used to model a wide range of specific structures resulting in a stronger generalization of the CNN.
* The tightly linked layer is a totally attached, standard plate. Growing of the output neurons is related to all of the input neurons. This is normally applied in combination with a Softmax function at the output to give the predictions. The nodes at the layer origin, then, must include the input probabilities of the CNN belonging to both groups. As each of these nodes is linked to the layer with all the input neurons, each receives all the knowledge from the network's first portion, including the convolutionary and pooling layers. This means that the final prediction is made based on the entire input image, not just the output of a certain convolution or pooling

**YOLO: Real-Time Object Detection**

You only look once (YOLO) is a state-of-the-art, real-time object detection system. On a Pascal Titan X it processes images at 30 FPS and has a mAP of 57.9% on COCO test-dev. YOLOv3 is extremely fast and accurate. In mAP measured at .5 IOU YOLOv3 is on par with Focal Loss but about 4x faster. Moreover, you can easily tradeoff between speed and accuracy simply by changing the size of the model, no retraining required!

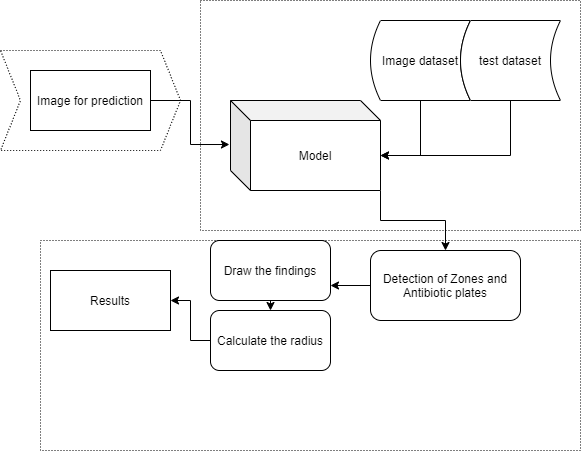
How It Works:

Prior detection systems repurpose classifiers or localizers to perform detection. They apply the model to an image at multiple locations and scales. High scoring regions of the image are considered detections.

We use a totally different approach. We apply a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.

Our model has several advantages over classifier-based systems. It looks at the whole image at test time so its predictions are informed by global context in the image. It also makes predictions with a single network evaluation unlike systems like R-CNN which require thousands for a single image. This makes it extremely fast, more than 1000x faster than R-CNN and 100x faster than Fast R-CNN. See our paper for more details on the full system.

**SYSTEM ARCHITECTURE**

****

We present YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, we frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance.

Our unified architecture is extremely fast. Our base YOLO model processes images in real-time at 45 frames per second. A smaller version of the network, Fast YOLO, processes an astounding 155 frames per second while still achieving double the mAP of other real-time detectors. Compared to state-of-the-art detection systems, YOLO makes more localization errors but is far less likely to predict false detections where nothing exists. Finally, YOLO learns very general representations of objects. It outperforms all other detection methods, including DPM and R-CNN, by a wide margin when generalizing from natural images to artwork on both the Picasso Dataset and the People-Art Dataset. We used about 97 laboratory antibiotic test images to train the model for measurement and detection of zone of inhibition as well as the detection of antibiotic plates.

**UML DIAGRAMS**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta- model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as Ill as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**GOALS:**

The Primary goals in the design of the UML are as follows:

* + 1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
    2. Provide extendibility and specialization mechanisms to extend the core concepts.
    3. Be independent of particular programming languages and development process.
    4. Provide a formal basis for understanding the modeling language.
    5. Encourage the growth of OO tools market.
    6. Support higher level development concepts such as collaborations, frameworks, patterns and components and Integrate best practices.
    7. **USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

**Identification of actors:**

**Actor:** Actor represents the role a user plays with respect to the system. An actor interacts with, but has no control over the use cases.

An actor is someone or something that:

Interacts with or uses the system.

Provides input to and receives information from the system.

Is external to the system and has no control over the use cases. Actors are discovered by examining:

Who directly uses the system?

Who is responsible for maintaining the system?

External hardware used by the system.

Other systems that need to interact with the system.

**Questions to identify actors:**

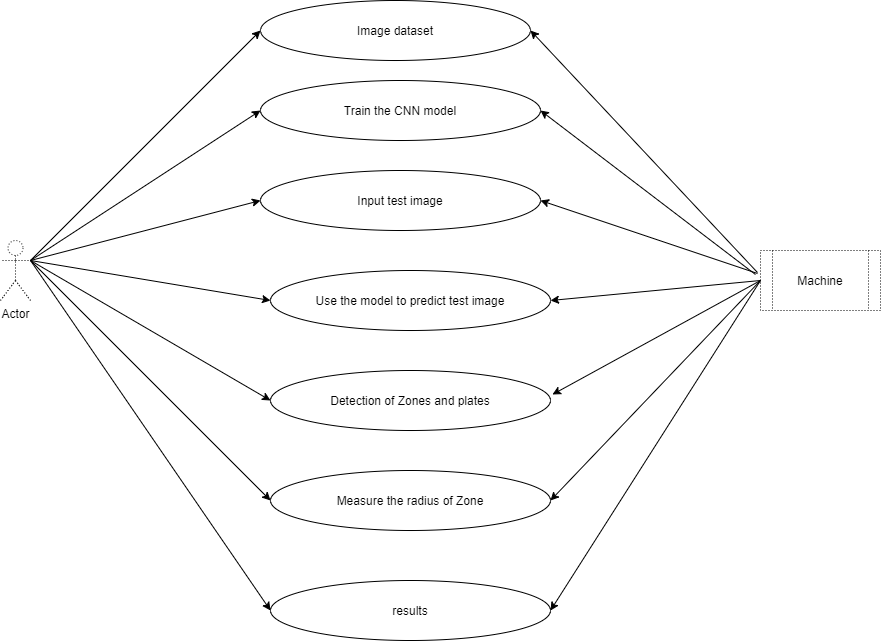
Who is using the system? Or, who is affected by the system? Or, which groups need help from the system to perform a task?

Who affects the system? Or, which user groups are needed by the system to perform its functions? These functions can be both main functions and secondary functions such as administration.

Which external hardware or systems (if any) use the system to perform tasks?

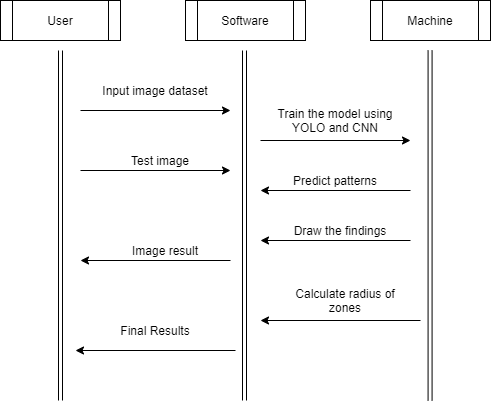
What problems does this application solve (that is, for whom)?

And, finally, how do users use the system (use case)? What are they doing with the system?

****

**SEQUENCE DIAGRAM**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

****

**ACTIVITY DIAGRAM**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the usiness and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

Repeat

Found

Not found

**Start**

Model

Output

Label the patterns

Find pattern

Patterns prediction

dataset

Input image

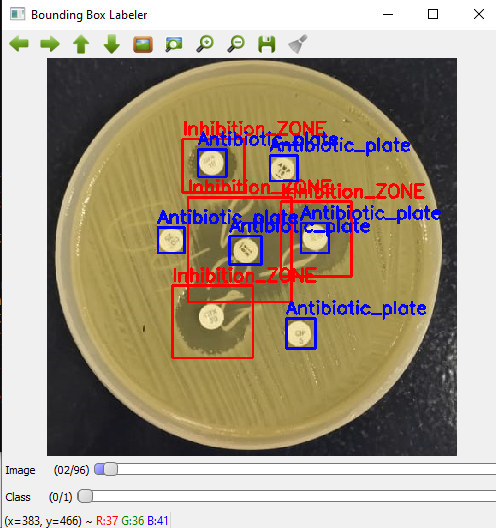
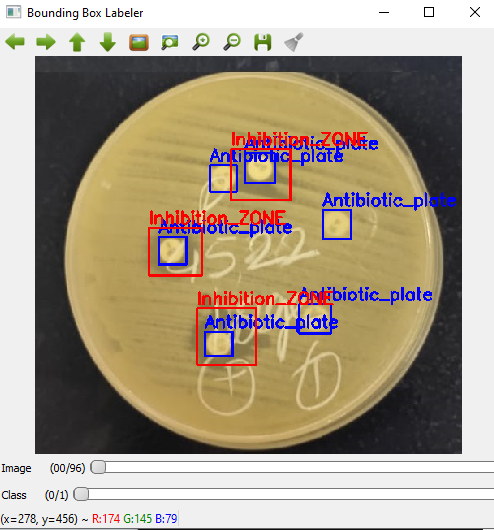
**SYSTEM IMPLEMENTATION**

To recognize the zone of inhibitions and antibiotic plates, we cropped only the top one fourth part of the image as that was the region where the patterns are located most of the time. From this, we locate the portion of the zone and plate by subtracting the binary image of the foreground of same region. Then we build a CNN model in order to separate the zone and plate images. This model is trained for the binary classification of zone and plate. These feature maps illustrate that the CNN learns the common hidden structures among the patterns in the training set and thus able to distinguish between a Zone and plates.

**Training the model:**

To trains our custom object detection models we need about 100 of jpg photographs that includes the required object (on our case it is Antibiotic test plates) in the photo. There are several methods of training our models. We used the annotation methodology to train ours.

Annotations:

****In this method, we use to draw a rectangle over the object where we found in the photo and label it with the appropriate class (Inhibition\_Zone, Antibiotic\_plate). This process will generate a text file that contains the co-ordinates of the marked area object and the label of the object. Below figure shows the annotation (red rectangle for zones and blue for antibiotic plates).

Then we used Dark-net to train our model with the help of Google colabretory to get the use of GPU and faster training. At least 2000+ iteration will make the model accurate.

Programs:

Setup.py (setup YOLO on the system)

from setuptools import setup, find\_packages

from setuptools.extension import Extension

from Cython.Build import cythonize

import numpy

import os

import imp

VERSION = imp.load\_source('version', os.path.join('.', 'darkflow', 'version.py'))

VERSION = VERSION.\_\_version\_\_

if os.name =='nt' :

ext\_modules=[

Extension("darkflow.cython\_utils.nms",

sources=["darkflow/cython\_utils/nms.pyx"],

#libraries=["m"] # Unix-like specific

include\_dirs=[numpy.get\_include()]

),

Extension("darkflow.cython\_utils.cy\_yolo2\_findboxes",

sources=["darkflow/cython\_utils/cy\_yolo2\_findboxes.pyx"],

#libraries=["m"] # Unix-like specific

include\_dirs=[numpy.get\_include()]

),

Extension("darkflow.cython\_utils.cy\_yolo\_findboxes",

sources=["darkflow/cython\_utils/cy\_yolo\_findboxes.pyx"],

#libraries=["m"] # Unix-like specific

include\_dirs=[numpy.get\_include()]

)

]

elif os.name =='posix' :

ext\_modules=[

Extension("darkflow.cython\_utils.nms",

sources=["darkflow/cython\_utils/nms.pyx"],

libraries=["m"], # Unix-like specific

include\_dirs=[numpy.get\_include()]

),

Extension("darkflow.cython\_utils.cy\_yolo2\_findboxes",

sources=["darkflow/cython\_utils/cy\_yolo2\_findboxes.pyx"],

libraries=["m"], # Unix-like specific

include\_dirs=[numpy.get\_include()]

),

Extension("darkflow.cython\_utils.cy\_yolo\_findboxes",

sources=["darkflow/cython\_utils/cy\_yolo\_findboxes.pyx"],

libraries=["m"], # Unix-like specific

include\_dirs=[numpy.get\_include()]

)

]

else :

ext\_modules=[

Extension("darkflow.cython\_utils.nms",

sources=["darkflow/cython\_utils/nms.pyx"],

libraries=["m"] # Unix-like specific

),

Extension("darkflow.cython\_utils.cy\_yolo2\_findboxes",

sources=["darkflow/cython\_utils/cy\_yolo2\_findboxes.pyx"],

libraries=["m"] # Unix-like specific

),

Extension("darkflow.cython\_utils.cy\_yolo\_findboxes",

sources=["darkflow/cython\_utils/cy\_yolo\_findboxes.pyx"],

libraries=["m"] # Unix-like specific

)

]

setup(

version=VERSION,

name='darkflow',

description='Darkflow',

license='GPLv3',

url='https://github.com/thtrieu/darkflow',

packages = find\_packages(),

scripts = ['flow'],

ext\_modules = cythonize(ext\_modules)

)

Train\_folder.py(setup the training folder)

import os

path='data/obj/'

imgList=os.listdir('Images')

print(imgList)

textFile=open('train.txt','w')

for img in imgList:

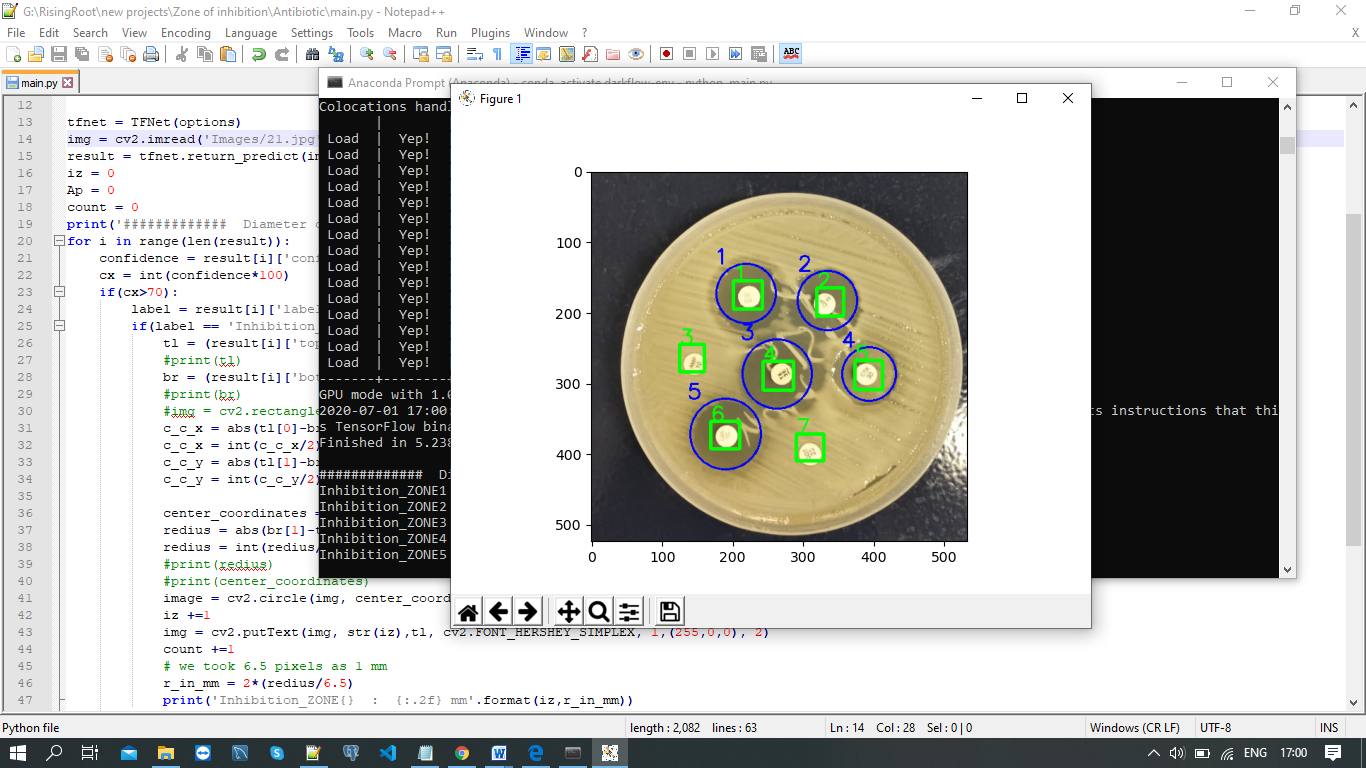
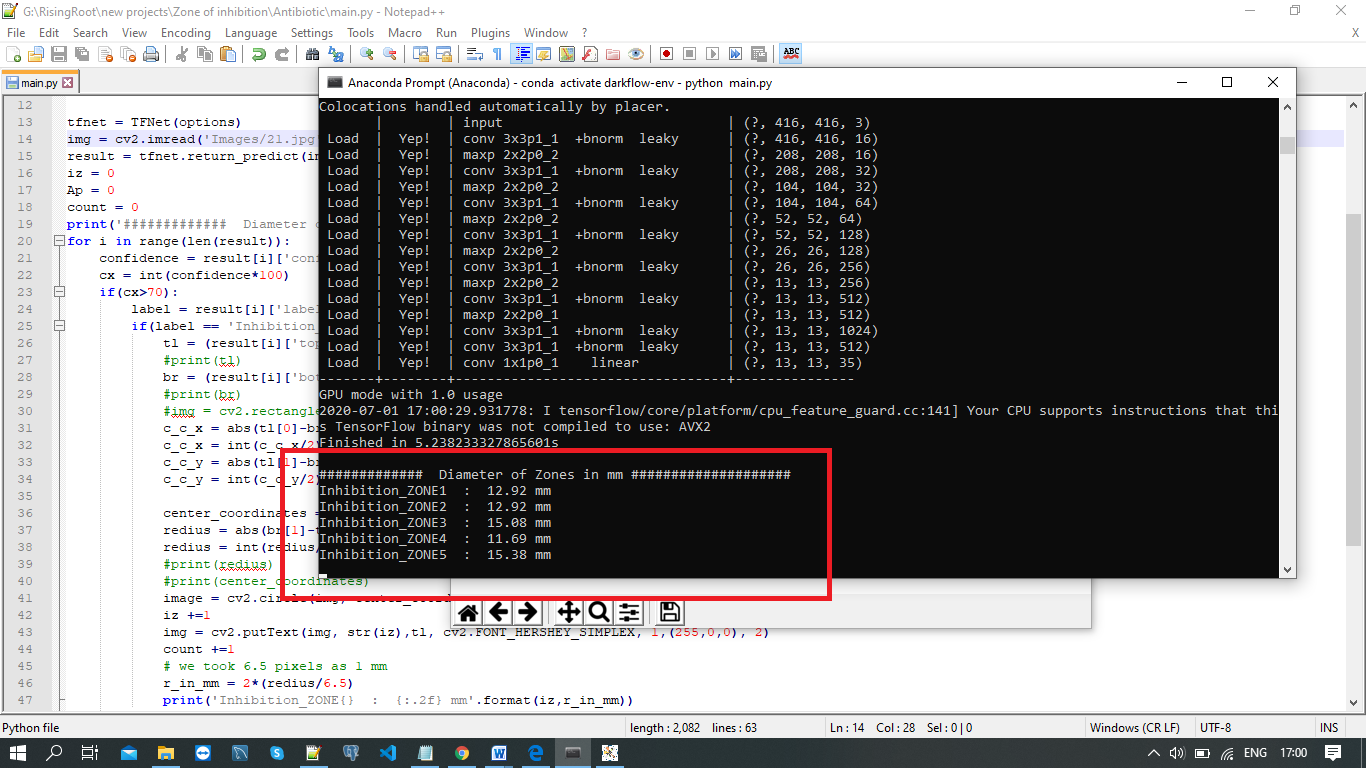
imgPath=path+ img +'\n'

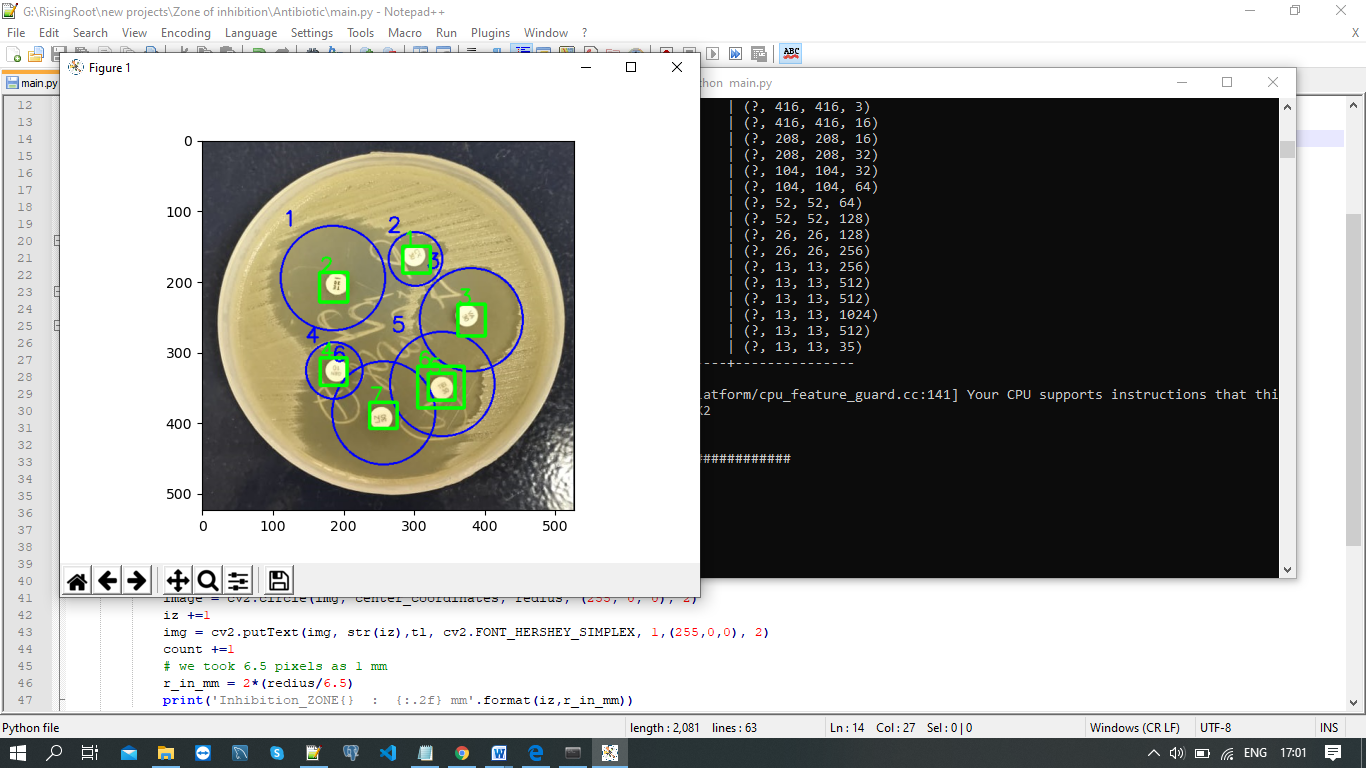
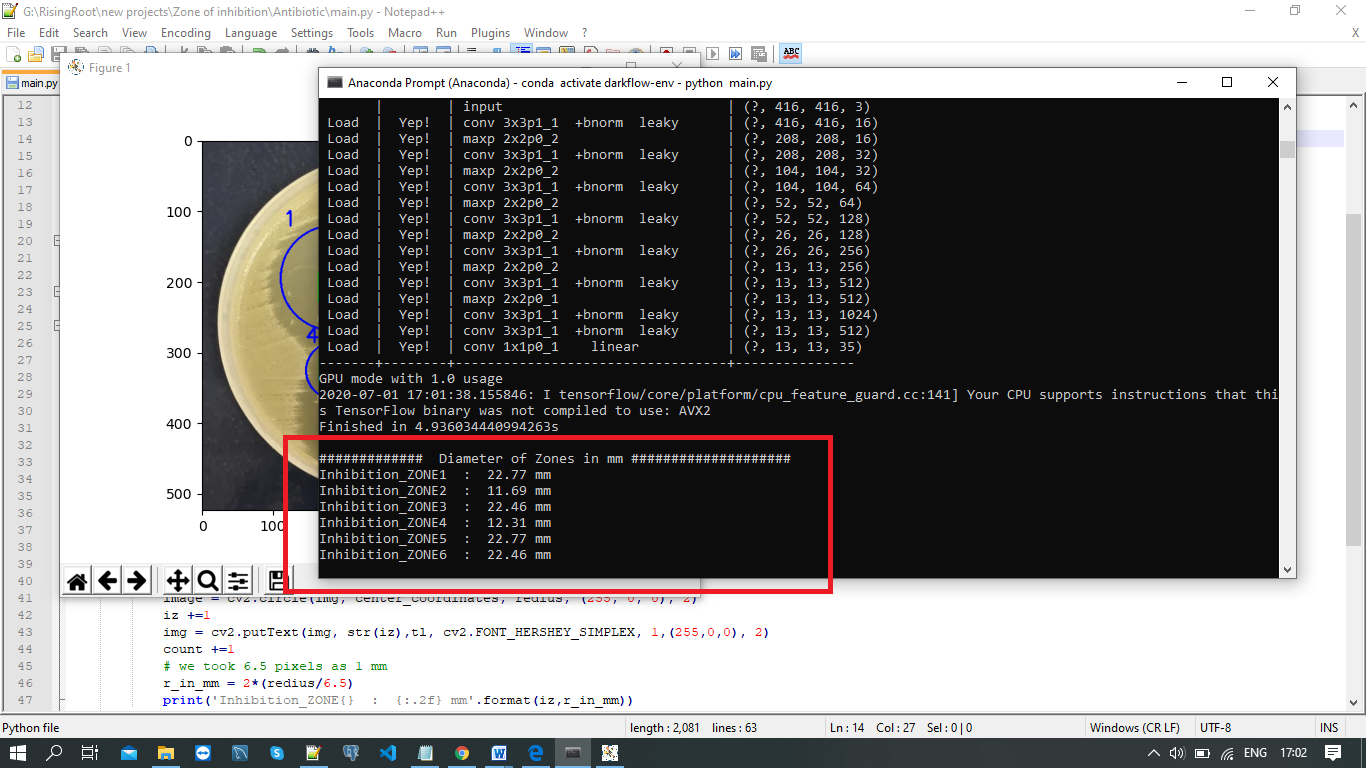
textFile.write(imgPath)

Bound\_box.py (Annotaions process)

The code is there in the file Bound\_box.py itself. It is too big to type in the documentation

**SCREENSHOTS**





**CONCLUSION**

The Zone of Inhibition is a very important measure that is required in the pre-diagnostic tests at a laboratory. The final output, which is, whether a particular antibiotic is effective against a given bacteria present in the patient or not, is the precursor to any treatment that would be administered to that patient. Hence an easy way of calculating the diameter of the Zone of Inhibition was a necessity. This work presents methodologies which are capable of returning the accurate diameter of the Zone of Inhibition by using Image Processing techniques. It reduces the amount of human interaction required to calculate and tabulate the diameter of the zone. Apart from that it also provides a means to directly see which antibiotic the bacteria is susceptible to by including a lookup of the standard specifications. This simple system can be very helpful in rural and semi-urban areas where there is a lack of skilled work force. It is also helpful in areas where there is a need for quick identification and tabulation of details as in the case of an emergency.

**DEVELOPMENT TOOLS**

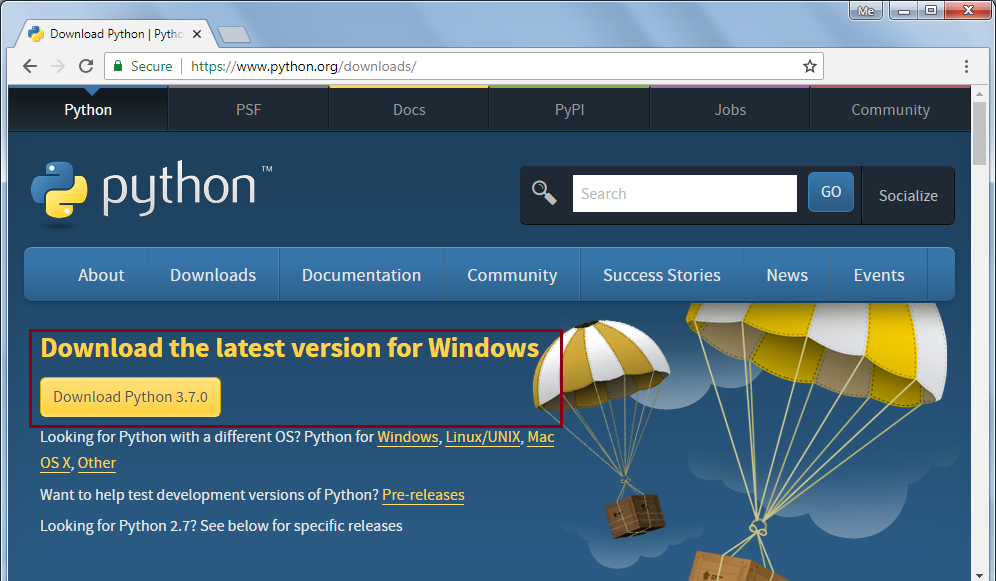
Python Environment Setup

## Python is available for Windows, Linux, Mac OS and some other platforms like IBM AS/400, iOS, Solaris, etc.

## To install Python on your local machine, obtain from https:/www.python.org/downloads a copy of the standard Python software distribution based on your operating system, hardware architecture and local machine version.

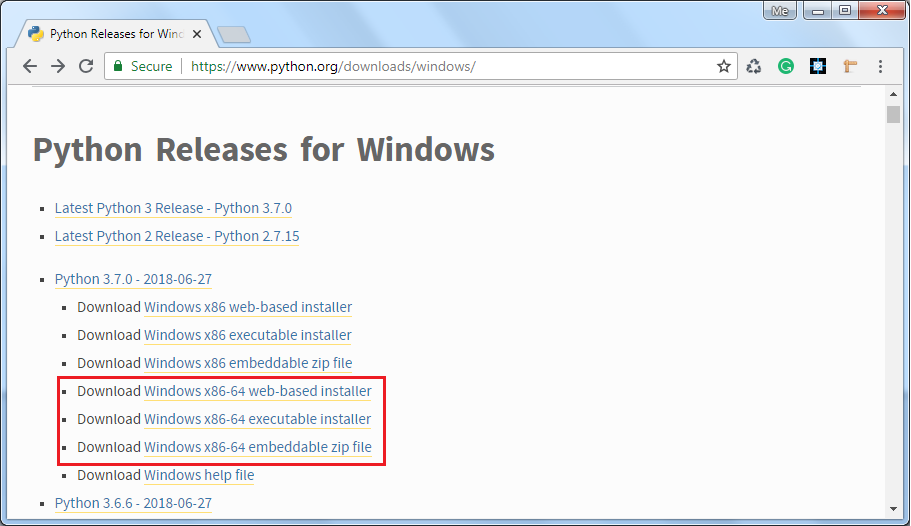
## Install Python on Windows:

You'll need to download the installer to install Python on a Windows platform. Installing Python on Windows is provided with a web-based installer, executable installer, and embeddable zip files. Visit https:/www.python.org/downloads/windows, and download the installer using the hardware design of the local computer.

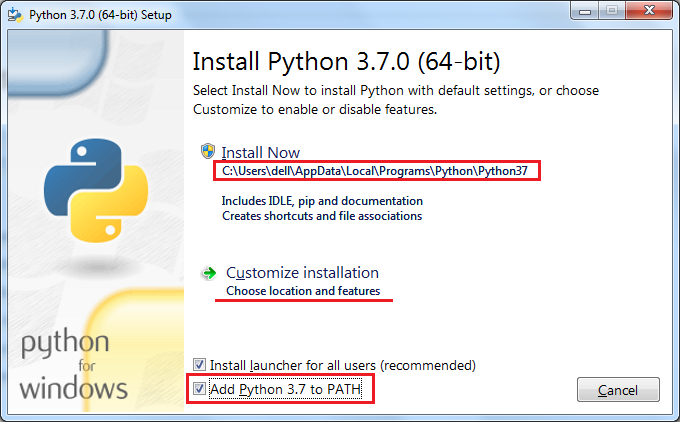
The Web-based installer requires an active link to the Internet. So, you can download the executable installer on your own too. Visit https:/www.python.org/downloads and press the Python 3.7.0 Download button below. (3.7.0 as of this writing is the new version)

Download Python Library

This is a 32 bit python-3.7.0.exe download. Go to https:/www.python.org/downloads/windows for the 64 bit installer, then pick the correct 64 bit installer as seen below.

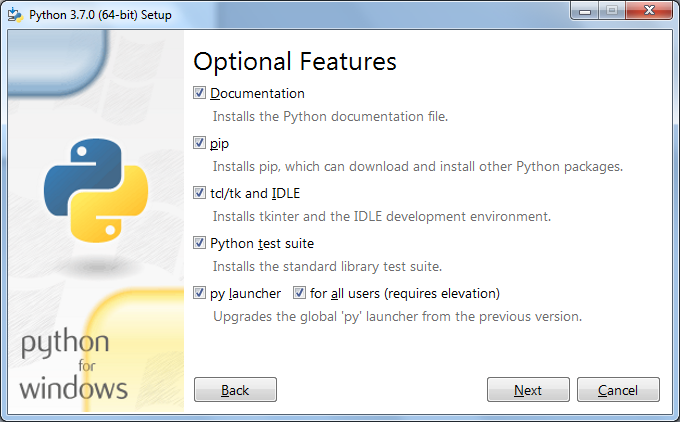


Download Python for Windows 64 bit

Download the executable installer for Windows x86-64 and double click on it to launch the installation wizard for python as seen below.

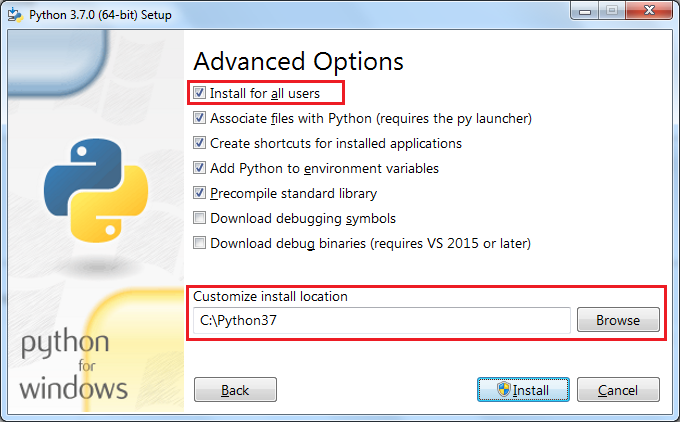
Python Installation Wizard

Installation is a simple wizard-based process. As you can see in the above figure, the default installation folder will be C:\Users\{UserName}\AppData\Local\Programs\Python\Python37 for Python 3.7.0 64 bit. Check the **Add Python 3.7 to PATH** checkbox, so that you can execute python scripts from any path. You may choose the installation folder or feature by clicking on **Customize installation**. This will go to the next step of optional features, as shown below.



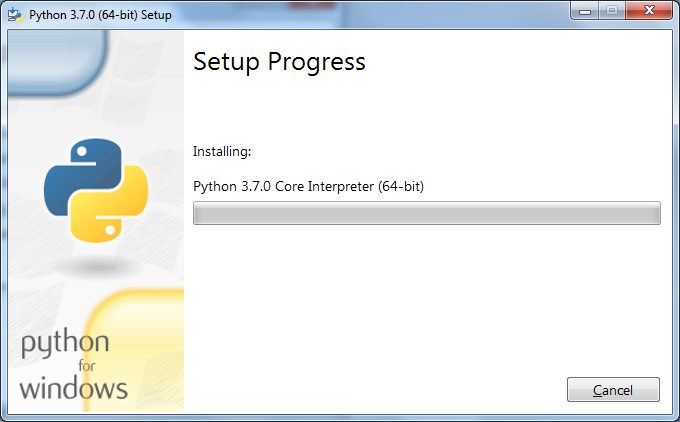
Python Installation Wizard

Click Next to continue.



Python Installation Wizard

In Advanced Options, select the **Install for all users** option so that any user of your local machine can execute Python scripts. Also, choose the installation folder to make a shorter path for Python executable (something like C:\python37), keeping the rest of the choices to default and finally click on the Install button.



Python Installation Wizard

After successful installation, you can start working with Python on your local machine.

**Libraries Used**

**NUMPY**

NumPy is a module in Python describing 'Numerical Programming.' This is the central science programming library that includes a versatile n-dimensional array structure, offers resources to combine C, C++ etc. This is also important in linear algebra, power for random variables, etc. As an efficient multi-dimensional container for generic data, NumPy array can also be used.

**NumPy Array:**Numpy array is a powerful N-dimensional array structure in rows and column shape. We can create, and control, numpy arrays from nested Python lists. The next query that should pop up in your head to run such numpy operations is:

**How do I install NumPy?**

Go to your command prompt to install Python NumPy, and type "pip update numpy." Upon completion of the download, go to your IDE (for example: PyCharm) and simply import it by type in: "Load as np number"

**Cv2** Is a computing interface library specifically targeted to real-time machine vision. It is a software used for Image Processing in plain language. It is primarily used to run all Photos based activities. Cv2: Cv2. Method imread) (loads an picture of the specified text. When the picture can not be interpreted (due to incomplete script, insufficient permissions, incompatible or invalid format) instead this process returns an empty matrix.

**Installation of Cv2**

Download and double-click the new OpenCV update from sourceforge platform to uninstall it.

1. Goto tab, opencv / build / python/2.7.

2. Photocopy cv2. C:/Python27 / lib / site-packaging pyd.

3. Open the Python IDLE and select Python Terminal codes following. > > > Cv2 import > > cv2 print.

**Glob** - Glob is a general term used for describing techniques for matching defined patterns according to Unix shell laws. Linux and Unix systems and shells also support global and include glob) (feature in device libraries as well. The glob module is used in Python to retrieve files / pathnames which fit a given sequence.

**Imagedraw** - The module ImageDraw provides basic 2D graphics for artifacts with an image. It can be used to produce new pictures, annotate or retouch current pictures, and build web-using graphics on the fly.

**Matplotlib** - Matplotlib is a Python programming language plotting module, and its NumPy numerical calculation extension. It offers an object-oriented API for the usage of general-purpose Interface toolkits like Tkinter, wxPython, Qt or GTK+ to integrate plots into applications.

**Installation** **of Matplotlib**

Download the installer, and run it. Next you will need matplotlib installer. Go to https:/pypi.python.org/pypi/matplotlib/ and scan for a wheel file that fits the Python edition you are using (a file ending in. whl).

**Keras** - Keras is an open-source library of neural networks written in Python. It can operate atop TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML. This is built to allow for easy experiments with deep neural networks and is based on being user-friendly, flexible and extensible.

**Installation of Keras**

Keras is installed in two ways. The first is to use the Python PIP installer, or a normal GitHub clone update. Using the PIP app, we must install Keras, because this is the preferred version. Again, we test the performance of the updated version.