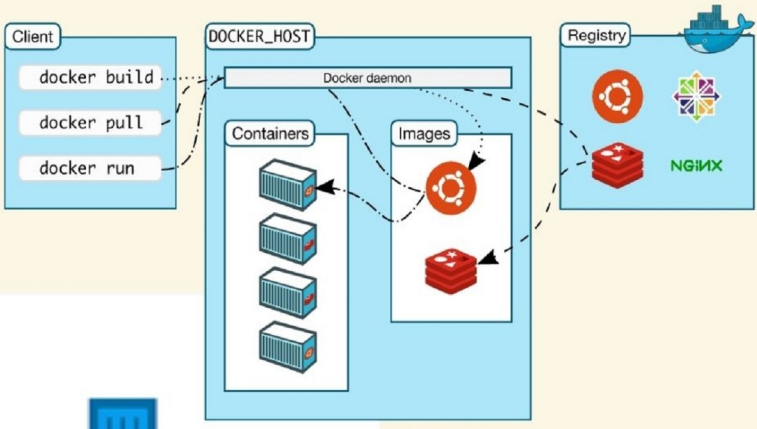
HOW TO DOCKERIZATION OPENCV PYTHON PROGRAM



By: Dulanji Hansika Wijekoon

MS21914300

How to Dockerization OpenCV Python Program

How to generate docker image for OpenCV video face detecting python program

Dulanji Wijekoon

This book is for sale at
http://leanpub.com/howtodockerizationopencvpythonprogram

This version was published on 2021-11-25

Contents

#

Introduction

This Book can be used as a guideline book to dockerize the OpenCV python program. Referring to this book can give a brief idea about OpenCV program and its usage and how to use and create OpenCV program. And also, the OpenCV program is running on a PyCharm and anaconda spider environment. Basic need of this book is how to do dockerization for python programs. As the main part this book describes the docker and docker environment. My main target of publishing this book is to give an idea to the reader about the docker environment. Also, I want to understand to my readers the advantages of the docker environment and I want to give experience about how to become familiar with this environment. Also, this book supplies all the guidelines about how to make OpenCV, how to install docker on Windows and How to pull a python program to docker hub as image and how to push it. Further here docker runs on a windows environment then this book readers can have a good idea about how to work with docker in windows environment. Because no more examples of docker dealing with window flatform. Also, I have included all instructions step by step with relevant images about python program creating things, docker installing and dealing with docker environment things. If having any relevant link with regarding codes and other stuff these all links are put at nearest of the steps. Finally, these are the main aims of publishing this book.

Chapter 1

What is OpenCV?

OpenCV means Open Computer Vision, so what is this Open Vision, if simply explain that, Computer Open Vision (OpenCV) is how computers see other external things? Like humans, so we get some examples of how we recognize apples and cherries?

Figure: 1

Subtly look at above figure 1, it has apple and cherry both together, how do you identify them separately?

Right…, You can identify apples and cherries both separately using their color, size and shape. Someone says I can identify by only taste without using this parameter. Yes, they are correct. But keep remember here we are talking about how to identify something or

someone using your vision sense.

That's over, we come back again to our computer vision identifying discussion. As I mentioned, apples and cherries can be identified using the above parameters. Suppose, how you remember those shapes, actually you and I, we both remember using our memory prototype. Our memory has a collection of prototypes about peoples, food, goods and other things with labels. According to that label memory identifier, this was an apple and this was a cherry.

Suppose, you never see the bike, Therefore, your memory has no idea about bike, if you see the bike, you cannot identify the bike, but if someone say about the bike shape, and other this you can have some idea but not clear, then you have lot of probability to make a mistake your prediction, because you have not exact clear idea about the bike.

Right… then look at figure 2. It has a bike image.

Figure: 2

Then, what idea about the bike, after seeing the image. Now you can have a clear idea about the bike. Because your eye processes that

image. From now on, wherever you see a bike, your memory has a clear prototype of the bike with a label. Using that memory, you can clearly identify the bike. Remember well, to detect this image you use your eye sensor to vision.

However, to train your memory you have to familiarize this image in your eye by looking at a variety of angels with millions of copies.

Actually, this part does not regard this computer vision. However, this part to your knowledge only, humans have five senses, these are sight or vision to eye sensor, hearing to ear sensor, smell or olfaction to nose sensor, taste or gustation to tongue sensor, touch or tactician to hand sensor.

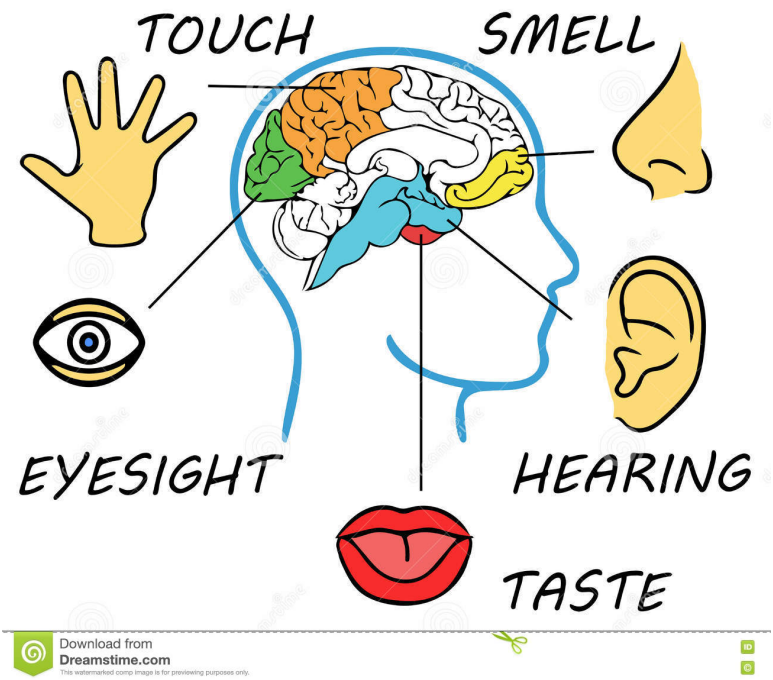


Figure: 3

By seeing figure 3 you can get a clear idea about the human's five

sensors. After absorbing environmental things through these five sensors. Then its absorbed things are sent to the brain by neurons. By the way, though, I mentioned the brain, for these duties actually use one part of the brain that is memory. According to neuron sent message, then, brain check memory and if can identify detect thing then memory sent to message again brain and according your consciousness you detect that thing. But if it is a new thing, memory has no idea about that. Then the brain saves those things as a new thing by labeling them.

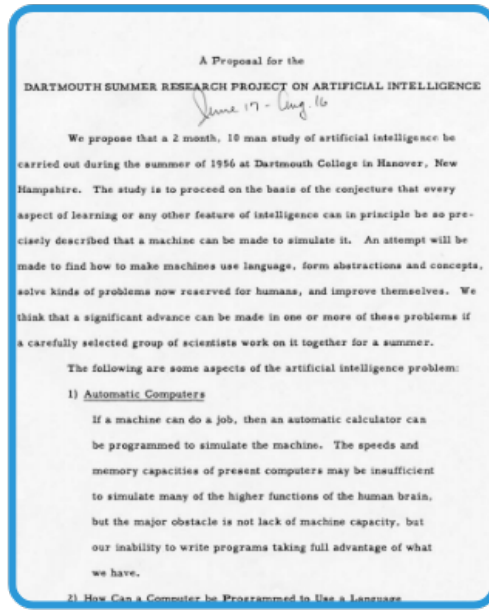
That is all, now we look at computer vision with some machine learning things. In considering Computer vision it also, can capture environment things using their vision by using camera sensor. Same as the human vision. Computer vision captured things are stored on a hard disk like human stores capture things in memory.

But Computer Vision has some differences when compared with humans.

Now, let's get an idea about recognizing images in computer vision. For doing that task computers have to train by introducing labeled pictures according to relevant tasks.

For recognizing processes in computer vision. First of all, we need to do classification for each object by labeling.

Before going to the classification process. Let's see the history of computer vision.



A Proposal for the

DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE

June 17 - Aug. 16

We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

The following are some aspects of the artificial intelligence problem:

1) Automatic Computers

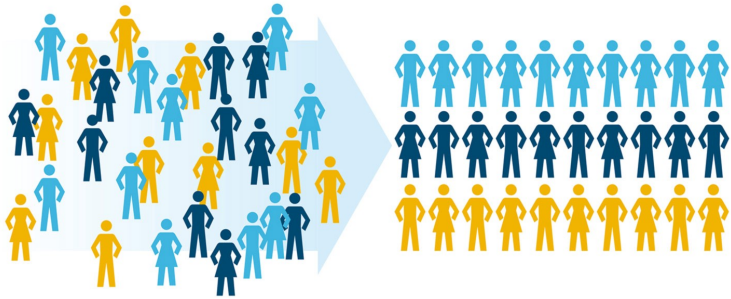
If a machine can do a job, then an automatic calculator can be programmed to simulate the machine. The speeds and memory capacities of present computers may be insufficient to simulate many of the higher functions of the human brain, but the major obstacle is not lack of machine capacity, but our inability to write programs taking full advantage of what we have.

2) How Can a Computer be Programmed to Use a Language

Figure: 4

Computer vision concept first came into the industry during 1960 in a university project as a **stepping stone to artificial intelligence**. Figure 4 was shared and the first computer vision concept was included in a research paper. Today computer vision is used everywhere for automation and robotics innovations.

It is for you to understand other processes in computer vision.



Then we come back again to our classification topic because, by learning the classification process earlier

Figure: 5

Look above figure 5 image it was classified people according to color. To classified, identified data we use pattern recognition, and image classification algorithms, software we not talk about deeply algorithms and patterns

In the classification process data was predicted according to target, categories or labels. If task was mapped as function (f)and input (x) then output is (y)that is discretized

To classify a data set we want to train the data set. Do this process we want a huge data set. For example, think about how you want to classify apples from other fruits. For these, we have to extract images of a variety of apple images. For an example consider below figure 6 apple images.

That image includes apples that are in variant types and colors.

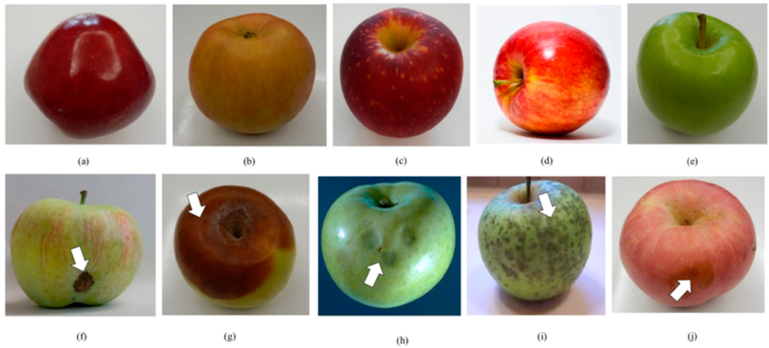


Figure: 6

These have several types of apples; you can detect different types of apples by using trains. For training using OpenCV image detection function you can use it to read the image RGB format from this program. Because computers do not have complex neural system arrangements to detect images and send messages to achieve image recognition like humans.

To do this process, we have to use virus type algorithms, software, techniques and programs, etc. Above mentioned that one of those functions is to identify color images. After detecting images according to color arrangement, they detect which type it is?

Because, the program does not know this spoiled apple and this good apple that it can recommend to eat. Function can understand only one thing is its color arrangement. According to that color, the developer otherwise inventor gives commands. Those types are category good, other types are category bad.

When the program detects some apple image firstly, they should identify that this was an apple in order to its shape and color. After that they detect whether the image is a good or bad apple to use. It is identified by Apple's external outlook. That outlook recognition by using color codes of per-trained images by developers.

Here I use the word "they" in the above paragraph to mean program.

Figure: 7

However next we see how to understand program color codes. Because the program cannot read words like humans, they have to understand only the binary "1" and "0" commands only (figure 7). Using that command, how can a computer understand an image?

I like to introduce that part using the gray scale image buffering concept. This is easy for you to understand the whole process in recognition of the image process.

Ok … First, we will start with the introduction with what is the gray scale?

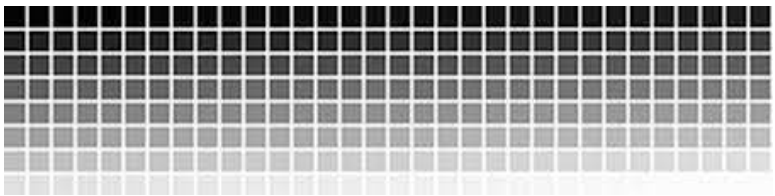


Figure: 8

If we simply explain about the gray scale. It was a color range

between back and whites as in figure 8. Actually, it was a grayscale color shade. That color shade is named as 1-255 range, otherwise it represents pixels. In this range 0 is represented by black and 255 is represented by white and other numbers are represented by grayscale color shades. Below figure 9 gives a clear idea about grayscale. This image was Abraham Lincoln's grayscale picture. Actually, this image was from the internet and it was easy to explain an image for the gray scale.

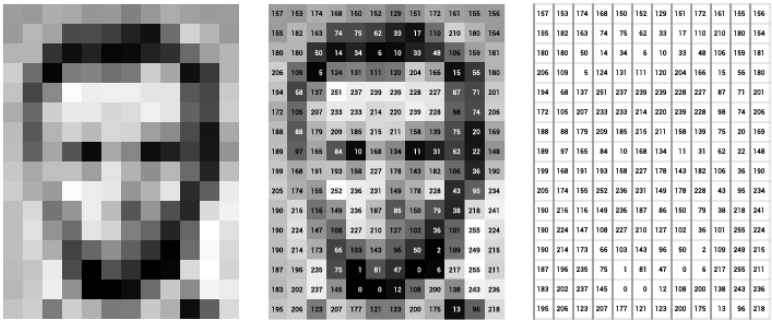


Figure: 9

Look at this image. This image each block is represented by the pixels. If we illustrated the image into the smallest part according to color range. It is represented by an 8-bit number. As an example, a computer wants to understand 28 grayscale numbers and how to detect that number?

Right...the 28-grayscale number, I can convert into binary as an 8-bit number by dividing 2 to reach the result remaining as 1(one) or 0(zero). Then the final binary for 28 is 11100.

The computer understands everything as a binary number. Also, the color of 28 represented is understood by it as a 111000 8-bit number. The basic theory behind gray scale and computer understanding is like this.

Then we look at how to understand RGB range because, in the day to day work we deal with not only grayscale images, we deal with

lots of color images.

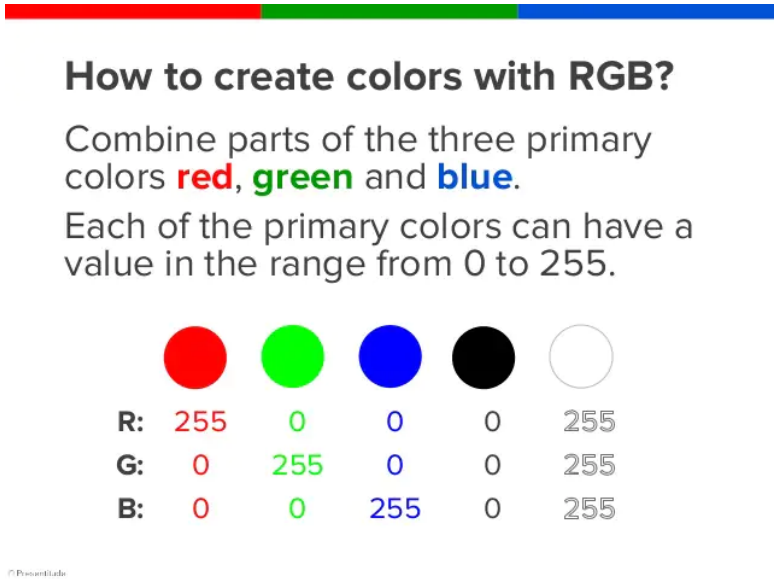


Figure: 10

As a figure 10 shown. Each of the primary colors have a value 0-255 range. Primary colors are Red, Green and Blue. By combining RGB values it makes another value. In the RGB range white color is represented as 255,255,255. Like these other colors are represented as a collection of three numbers of 0-255 value range. That is how the image processes on the computer.

Up to now we discuss how to detect normal images on a computer. But from this we discuss how to detect facial recognition by using computer vision. Because we are going to create a face recognition OpenCV application using python. For this process we brush up our knowledge about the face recognition of computer vision.

Before going to that part, we briefly discuss evaluation of computer vision.

Basic idea of computer vision is to mimic human vision. Consider-

ing all of computer vision programs that are made correspondingly to the human vision system. Basically, Computer vision is used to read data, identify extracts and images. Before coming to this computer vision a lot of this has to be done manually, after finding computer vision combined the scientist and innovation do unbelievable things. Lot of programs, a lot of devices and a lot of robots made using this computer vision. Also, the supercomputer is the one of output that are used computer vision

Below figure 11 is shown the evaluation of computer vision with the respectable persons

Figure 11

These come into several categories. That is video and photograph image recognition. In this book I instruct you to create face recognition on the video. But you can use both video and images.

Technically, this training model supervises the models by labeling and finally, gives an unlabeled data set and according to previous knowledge trained program can predict the identified the non-label data set.

In the Machine learning classification supervised learning is classified under lazy learning and Eager learning.

Considering the Pipe line of Machine learning. Because this OpenCV is the one of the concepts that is categorized under machine learning. Figure 12 is the simple explanation in the data set training pipeline of machine learning.

Figure: 12

Eager Leaning:

Construct the data set based on the given training data set. Before receiving data of classification. This Leaning type has much time to train the data set and less time to get the predictions.

Example for eager learning is Decision Tree, Naive Bayes, Artificial Neural Networks

Lazy learning:

Comparing Eager Leaning this gets much time to train and get predictions. Because, this is not classified data. Before receiving classification.

The example of Lazy learning is k-nearest neighbor, Case-based reasoning

Some supervised Learning algorithms. Before going to algorithms supervised learning categorized under classification and regression. Here I give example of classification algorithm of the supervised learning

Naive Bayer's-

Statistical classification technique based on bayes theorem. Beginning of machine learning must use this algorithm but nowadays it is not widely used.

Decision tree-

Continuously run according to training data till meet terminal condition.

Support Vector machine-

Used to classify data constructing the hyperplane. Based on hyperplane (decision boundary) data is classified as near data to the hyperplane is called support vector. As follows figure 13.

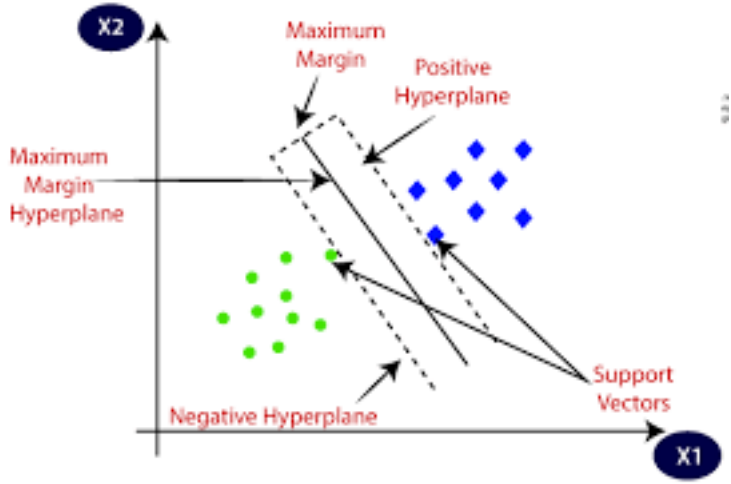


Figure: 13

K-Nearest Neighbor-

Adding new data points that are categorized under the nearest neighbor categories. Like figure 14.

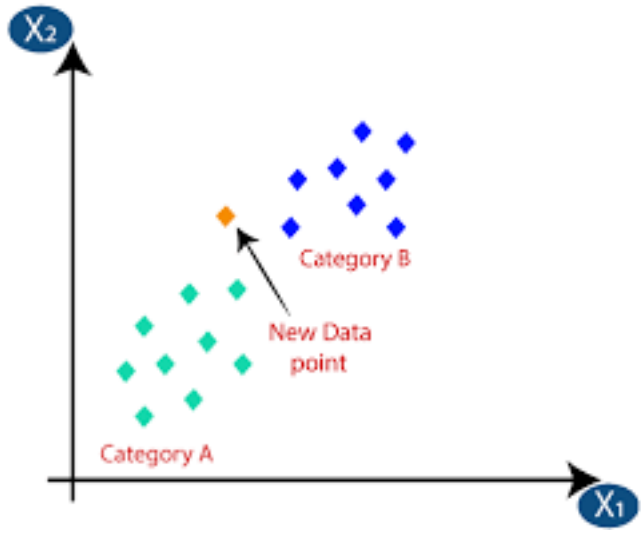


Figure: 14

Random Forest-

Assembled a decision tree using a trained data set like figure 15 fruit example.

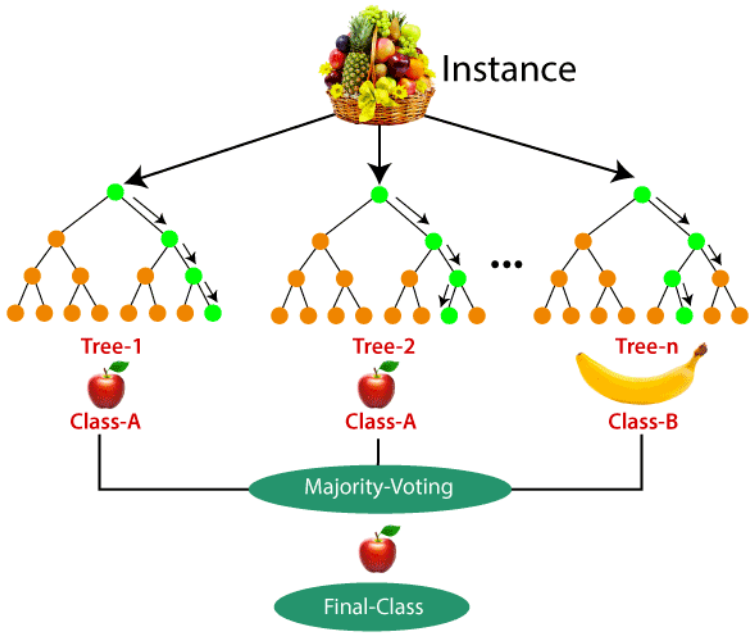


Figure: 15

Next, we look at how to create a face recognition application?

As the first step we have to make a database by capturing individual images to track specific mention formats. For example, to detect a human face we track two eyes, one mouth and one nose. And also recognized the face of that component where it should have been placed. These types of tracking information can include one database. As figure 16

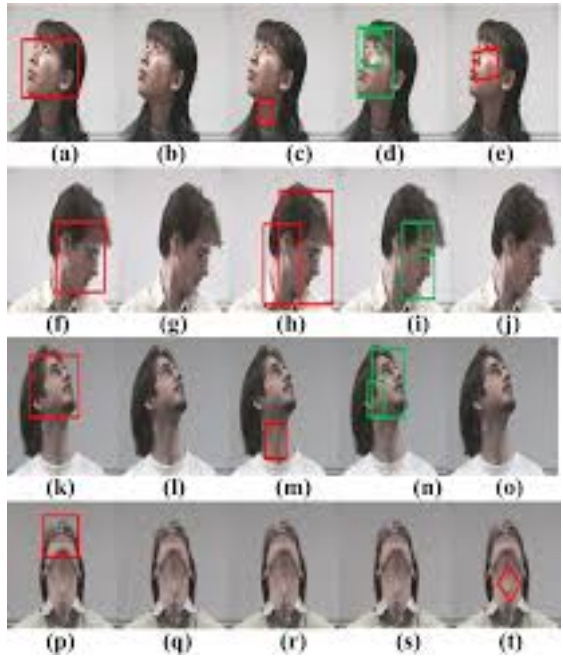


Figure: 16

Then as a second step you can classify the image according to the place of eyes, distance between lips and width of nose bridge, and unique characteristics of each person's face should be placed. Like this you should have to enter several key points to annotate the image

As a final step you can capture a new image. Using those images, you can do the measuring process again by comparing the above sample data. Then you can clearly add the measurement process according to the new data with some angle data.

Then finally you have to compare with the last and previous images measurement you can replace the new image with corresponding data.

The training data model process takes much time and is a huge process by doing as manually. However, small computer vision

applications use statistical learning algorithms such as linear regression, logistic regression, decision trees or support vector machines (SVM) to detect patterns and classify images and detect objects.

But deep learning does this process better than machine learning. Actually, deep learning is a subset of machine learning. Deep learning has been included in the concept of the neural network. It helps to read and detect images without face recognition instruction. For this developer to choose a preconstructed algorithm and train it with examples of the faces of the people it must detect. Finally, give the output according to the desire of the developer.

However, a lot of people use pre-made classifiers for face detection. In this OpenCV face dedication program I have used that type of classifier.

Next, we see some application of the computer vision (CV)

Self-Driving Cars

This is one of most commonly used CV applications. That was captured by using a car camera. And also sense external things using the sensor.

Facial Recognition

Here I am not talking hugely because, throughout this first chapter I discussed the face recognition process step by step.

Augmented Reality & Mixed Reality

Hugely use CV programs in augmented reality to demonstrate smart things from reality. Example for this smart phone smart glasses. To identify real words and determine real word thing is used Computer vision. And also, as instruction prediction computer vision things used in the augmented reality

Healthcare

To identify cancer, some x-rays and MRI scanners mostly use Computer Vision programs. According to pre -defined, pre-trained

data set. The classifier classifies the disease and other obstacles using this technology.

Defect and other Detection, **Assembly verification**, **Screen reader**, **Code and character reader (OCR)**, **robotics for bin picking** like this many more applications can be defined under the computer vision category. Nowadays computer vision is an indispensable thing in the new innovation world.

From this I like to share with you about the computer vision task

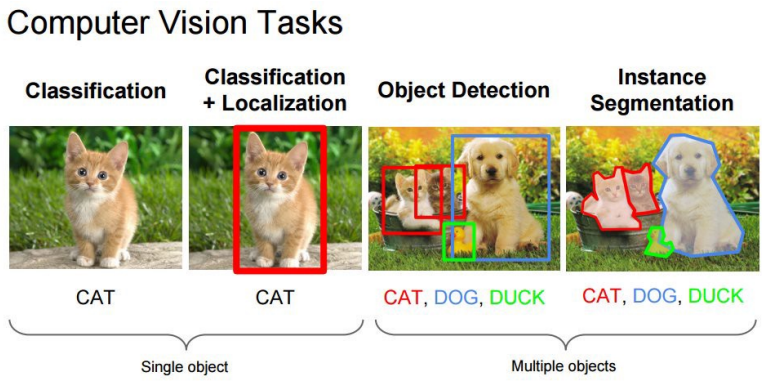


Figure: 17

According figure 17 we can categorize computer vision task as follow

Object Classification:

Basically, in this classification talk about the broad categories that are named on the photograph

Object Identification:

In this phase talk about the object type. For example, we have a human face image. According to that image this was a front face type image, this is nose type images, like this object can be identified. This section talks about that.

Object Verification:

In this section is the verification part of the object. This is the human face, this is the fruit image, the human face can be identified using trained measurement and the shapes.

Object Detection:

And this part is identifying the object location if get some examples, some human faced picture has with forest background, then according trained data set in this part identified the humans face from whole image. Also, this part can be used to identify specific people in the image sometimes without hair peoples, old peoples like that, according to a given specific category. This prediction is done in this area.

Object Landmark Detection:

In this phase describe the key point of the photograph. For example, eye gaps. Nose position, gap between upper lip and Norse, like this lot of key points are detected in this phase according to the output.

Object Segmentation:

In this phase talk about the pixels that belong to the object. Because computers detect some object using the pixel. According to binary code they create their own pixel and then create the whole image according to the pixel algorithm. Above section in detail talks about pixels and computer vision and how to understand computer vision images. If you do not remember you can go to page number 9 and brush up your memory.

Object Recognition:

In the section doing the object recognizing. If above image 17 the last two images have fog and cats both are in image. But our program was trained on how to identify the cat image from the crowd. According to the training data set in this section separately identified the cat image from the other objects.

Here if I very simply mentioned about step, this is not a simple process. To complete this process correctly we have to use lot of

methods algorithms some logics and lot of things to successfully done this process

Next let's see how to computer vision identify video objects and what are the task that included

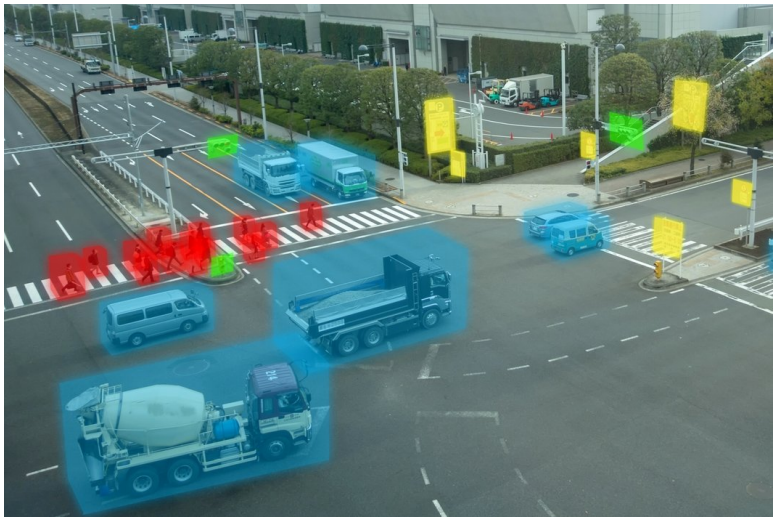


Figure: 18

Above image 18 you can see every object is detected separately by computer vision, to do this task computer vision analyses that video and categorize that object separately from each other object. For this to get output like this computer vision does some tasks behind. Now time knows the behind task about the video analysis of computer vision. Below I simply explain that details briefly point wise.

Video motion analysis

In this step measure the velocity of the object in the video or recoded capture by the video motion analyzer

In image segmentation

In this part images of video footage are divided into various sets of

the video view. For this is used to developed specific algorithms

Scene reconstruction

As named, in this step is created a 3D model of each scene according to the inputted series of images or video footage.

In image restoration, noise such as blurring is removed from photos using Machine Learning based filters.

In this step do the restore process. Let's see what is the restore process? In computer vision we get normal video, not record station made and edited video. All normal videos have common phenomena. That is noise. In this phase, remove that noise and remake the video or image series by cutting down the noise. That process is done by machine learning based filters.

According to my knowledge I explained the whole behind technical procedure of the computer vision as short and simply. If you are a beginner of this subject, you can understand these technical things easily. From this I will not talk about the behind the procedure of computer vision. Next, I would like to tell you what the advantages and disadvantages of the computer vision

Advantages of Computer Vision

As an advantage of computer vision, we can get a lot of things. In this era if any image detecting application, computer vision is the required item in this application.

Computer Vision applications are mainly responsible for detecting images. Through detecting images, we can get predictions by analyzing images. Main benefit of the computer vision procedure is, if we include some video or image series to the computer vision process. we can get only the wanted part by separating from other video content. By getting these parts we can do analyzing easier. For example, crime sectors can add defined details about crimes if we coded all cities CCTV by these trained crimes

data set. Then, if a crime is detected from the CCTV, then its details

can be sent to relevant authorities with location.

Also, Computer vision can be used to entertain things. By detecting your face your face can change into a dog face, cat face, by using some lenses. If you are using a snap app or Instagram app you can find that feature on this. Actually, I do not know any other lens application, however a lot of lenses use applications that run behind computer vision.

Also, we can recognize people that are not wearing face masks , If we add some dress code. Someone does not follow that dress code we can detect those people by using simple video capture.

Furthermore, In agriculture field. They can be used to detect healthy plants, weak plats and weed plants. If we trained the data set by adding images with disease plants according to that data by simple image clicking, farmers can recognize the plant diseases and if we add the solution for that disease farmers can add that solution easily and simply by one click on the screen. Computer vision is an indispensable thing in the world automation process. Using this computer vision you can catch up everything by far from the fingertip.

Disadvantages of Computer Vision

If it has a lot of advantages, computer vision can also have some disadvantages. Sometimes , this computer vision can be treated for someone. Using that computer vision some or something can be hacked by anonymous parties. Using this technical knowledge anyone can gather or track other people's details of location. Like this lot of advantages have in computer, however in near future, Court have to express new rules and regulation for targeting this new technology crimes.

Chapter 2

How to Create an OpenCV Face detection Program?

This chapter we mainly focus on how to create an openCV program in a python environment. Basically a lot of people use the pycharm and anaconda IDEs to run OpenCV programs on the python environment.

As below figure 1 those are the icons of pycharm and anaconda, spyder


PC

Figure: 1

Then we see how to install these IDEs on your computer. Again I should tell you that all IDEs are run on the windows environment. Therefore, all my examples are relevant to the Windows environment .

If your PC or laptop does not have a python environment first of all you should download python on your computer. This is the link for downloading the python in windows environment.

Link for download python [https://www.python.org/downloads/windows/](https://

After going to the above link you can see the interface below figure 2 . Then you can choose and download the python according to your requirement.

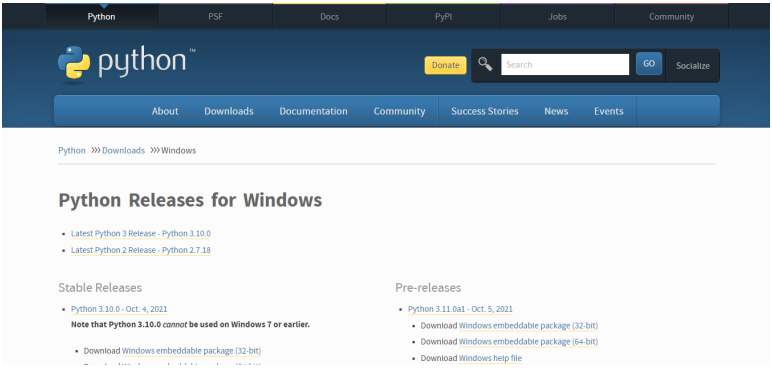


Figure: 2

After that install this on your PC then you can simply open python shell application as figure 3 by searching on windows

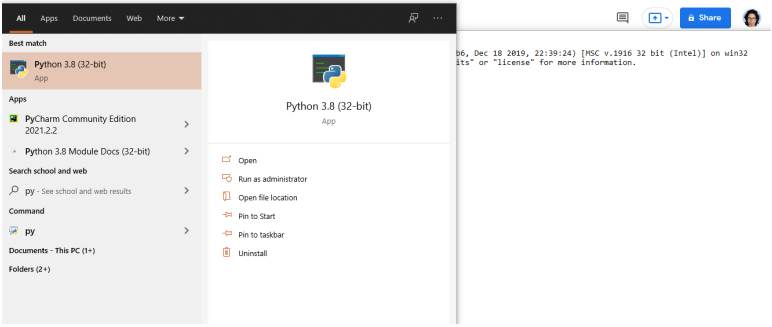


Figure: 3

Next we look at how to install pycharm

By clicking this link [https://www.jetbrains.com/pycharm/download/\#section=win

You can go to the pycharm download page as shown figure 4

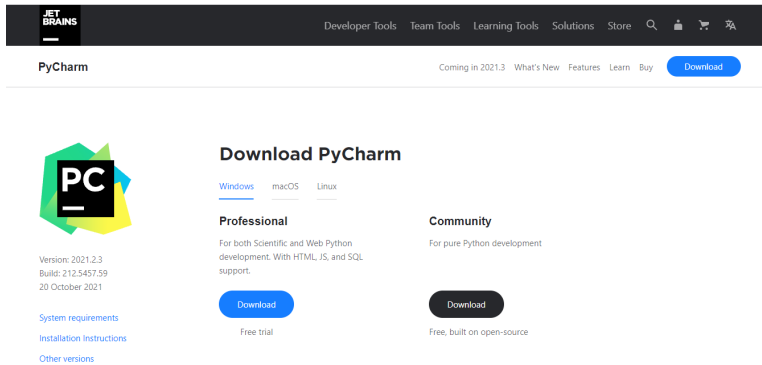


Figure: 4

Then you can download the pycharm version according to your needs. Here i download pycharm IDEs which are made for windows environments. Also, In this demonstration I have shown using the pycharm community version. When downloading the version they ask for a simple verification. You can choose that verification according to your want.

After that installation you can open Pycharm IDE like figure 5

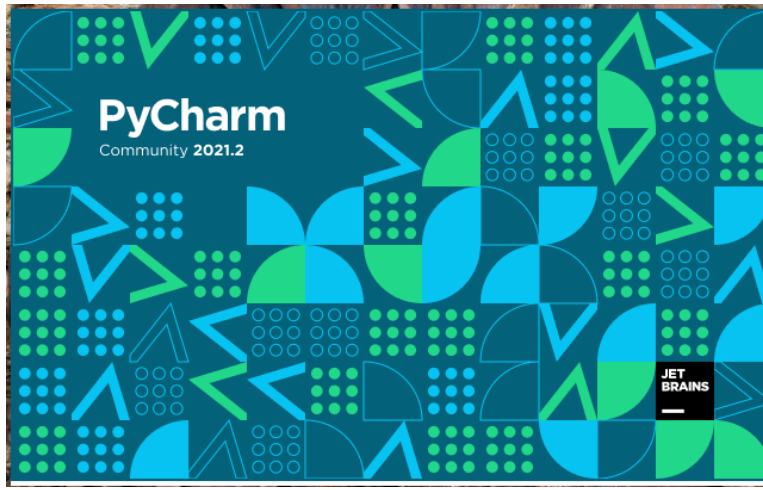


Figure: 5

After loading the pycharm you can see an interface like follow figure 6. If you have not already created the project you can click the new project button and then you can add the project name. Then the project will be created.

If you already have a python project you can choose that project and open it by clicking the open button.

In this example I have chosen the "Python Docker" project to open.

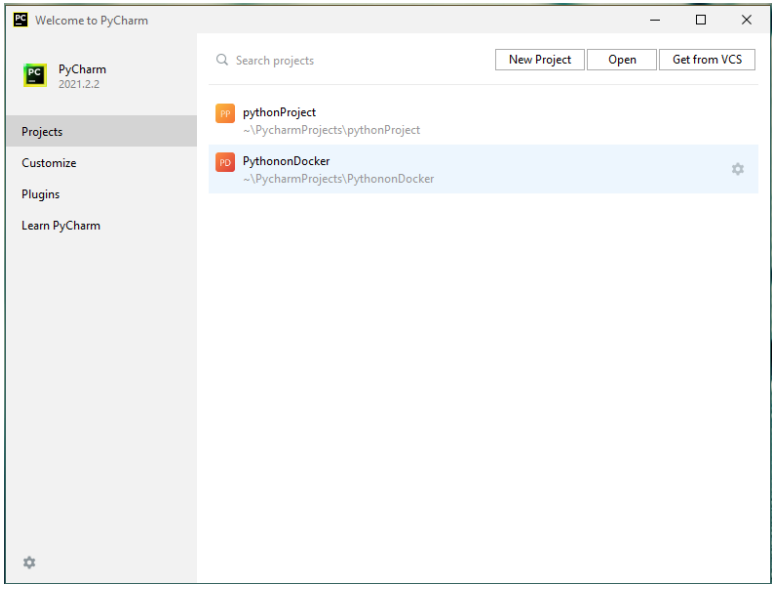


Figure: 6

After opening the project you can see the figure 7 interface. If you want to add a new python page or whatever you can do it by simply right clicking. In this figure main has been focused on the openCV python project.

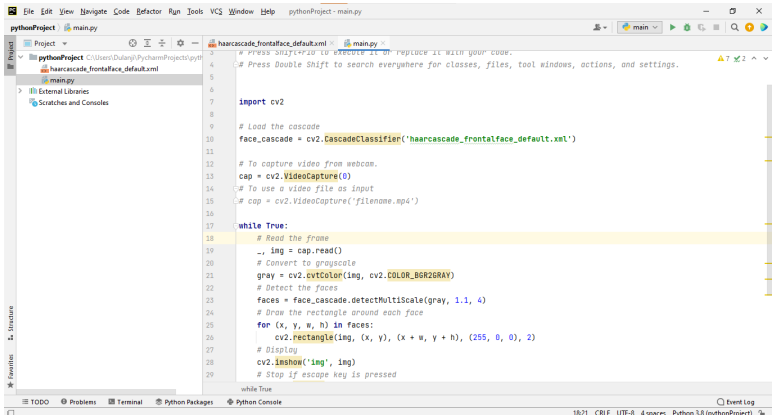


Figure: 7

Before placing coding you should import openCV libraries. In the pyCharm environment you can import this library under two ways

1st Method:

By clicking the bottom terminal, you can open the terminal tab. You can type this command

“pip install opencv-python” and enter then automatically download and install the openCv by this pipe line manager as Figure 8.

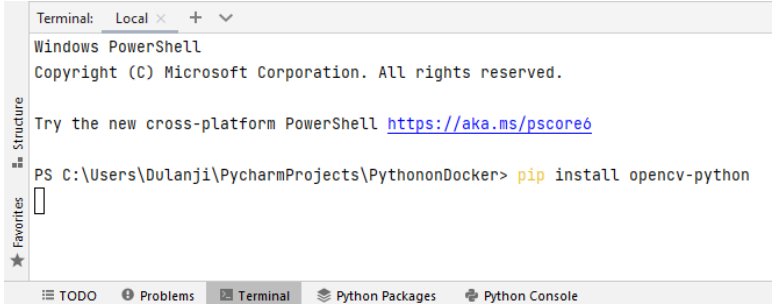


Figure: 8

2nd Method:

Second method is you can simply go to the setting and python interpreter as shown figure 9

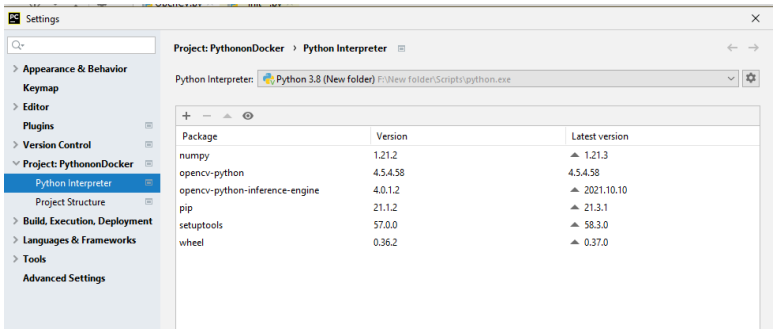


Figure: 9

And next by clicking your reverent project on the python interpreter you can see the download and install packages and relevant things regarding your application.

By clicking the "+" mark at the top. you can install new libraries.The figure 10 show the list view after clicking the "+" mark.

By choosing the relevant package you can click the package installation button. Here, I have picked the "opencv-python-interface-engine" package to install.

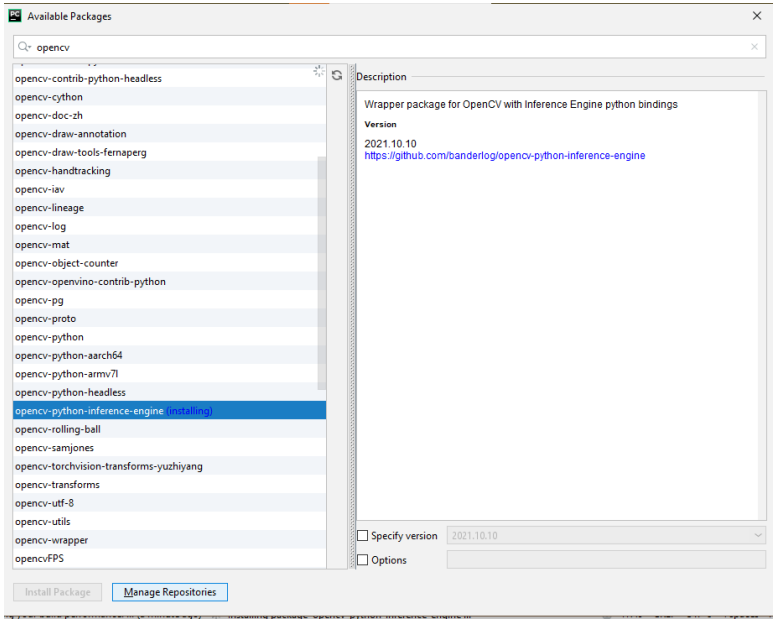


Figure: 10

As a next step you can go to the main python page and type "import cv2" command. Then the file does not show the error if your installation is done successfully.

Before coding, an important step is the xml file. The classification file is here i mean as a xml file. Before run code classification xml file and python coding file both should be placed at the same place.

Here is a classification file I have downloaded from the internet. Because if we create the classifier it is a huge process and it takes a lot of time to process. Therefore , here I download the already done classifier to do this process.

This is the link for the classifier :

[https://github.com/DulanjiH/OpenCV-Face-Dectection-Python-Program/blob/master/haarcascade_frontalface_-default.xml](https://github.com/DulanjiH/OpenCV-Face-

Dectection-Python-Program/blob/master/haarcascade_frontal-face_default.xml)

Ok … the figure 11 shows the python code for the OpenCV application. It has very simple and easy steps. Now we get some idea about the code example according to the comment lines order.

```
import cv2 # Line 1

face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')# Line 2

cap = cv2.VideoCapture(0)# Line 3

while True:

    img = cap.read()# Line 4

    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)# Line 5

    faces = face_cascade.detectMultiScale(gray, 1.1, 4)# Line 6

    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)# Line 7

    cv2.imshow('img', img)# Line 8

    k = cv2.waitKey(30) & 0xff# Line 9
    if k == 27:
        break

cap.release()# Line 10
```

Figure: 11

Line 1:

First line is responsible for importing the import “cv2” library to the application.

Line 2:

In this line load the classifier to the program. As I already mentioned, the classification xml file and that import main python file should be placed in the same place. Otherwise, it will give an error due to the compiler not finding the classifier path.

Line 3:

capture video from a web camera. If you have several web cameras, the first camera is used for capturing the video.Not only for the camera caption by changing line 3 as " *cap = cv2.VideoCapture('filename.mp4')*" this. You can capture an image of the video clip. "*'Filename.mp4"'* is the name of the video clip.

Line 4:

In the while loop continuously read the frame above capturing (video that access from cap variable) from stream.

Line 5:

Capturing things convert to grayscale as can understand the computer vision

Line 6:

Detecting the face according to above send grayscale order

Line 7:

According to the given x,y scales, the program is a drawn rectangle around the face that is detected from application.

Line 8:

Display a picture image on a video interface named "Img".

Line 9:

By clicking the "Esc" key you can stop the video capturing process

Line 10:

By this line release the video capture object

And last step you can run this program by simply clicking right top arrow key as figure 12

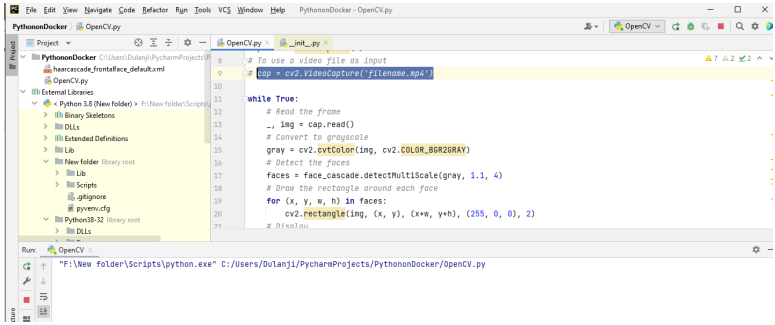


Figure: 12

This is a extra point : if you use pycharm IDE you can upload project from in the IDE by clicking

VCS -> state project on gitHub path without exiting the program. Figure 13 demonstrates that step.

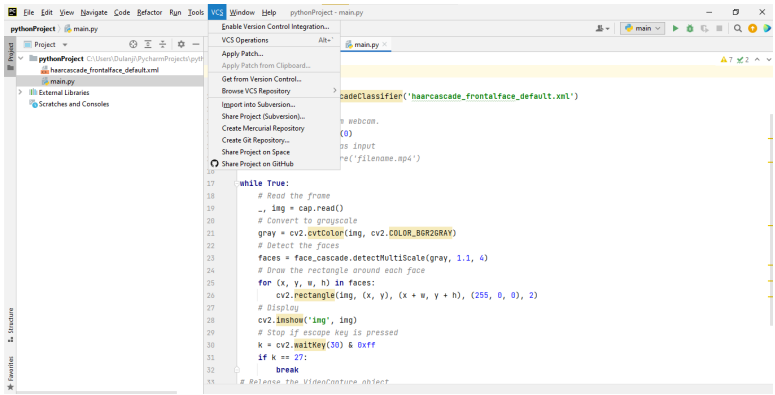


Figure: 13

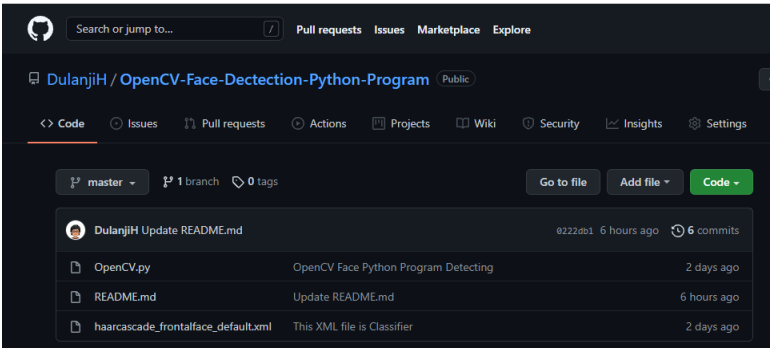
From 6 to 13 figures series are shown about creating openCV on python environments.

Also, this whole area I upload on github and below has its link.

Link for the github project: [https://github.com/DulanjiH/OpenCV-Face-Dectection-Python-Program](https://github.com/DulanjiH/OpenCV-

Face-Dectection-Python-Program)

If you are interested in trying this program you can get the whole project, that includes python code, xml file.

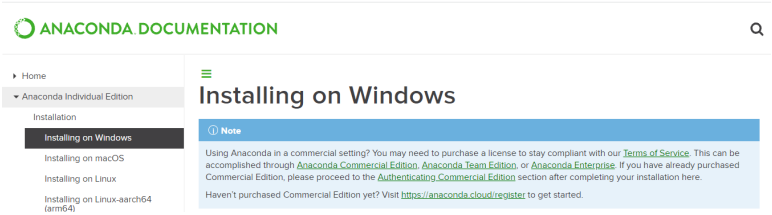


From this I am going to discuss how to create an openCv program on an anaconda environment using spyder IDE.

First of all you should download the anaconda to your PC. by going to the anaconda documentation page you can download the anaconda version as you need.

This is the link for anaconda documentational site: [https://docs.anaconda.com/anaconda/install/windows/](https://docs.anaconda.con

By clicking above link you can go to this site



This site is included each and every information about the how to install anaconda on windows

After successfully installing you can see anaconda navigator and other stuff by simply searching as anaconda on taskbar as figure 14

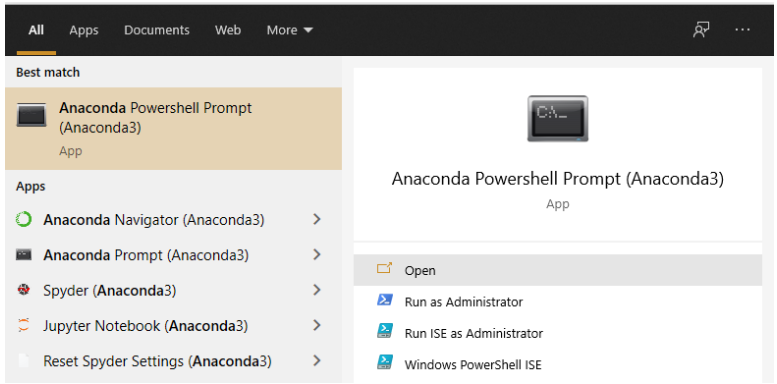
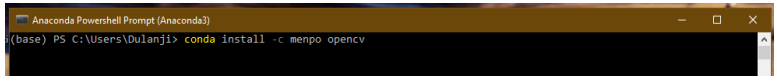


Figure: 14

you can open anaconda power shell as figure 15

Figure: 15

As a second step you can type “conda install -c menpo opencv” command to download whole package of OpenCV as figures 16



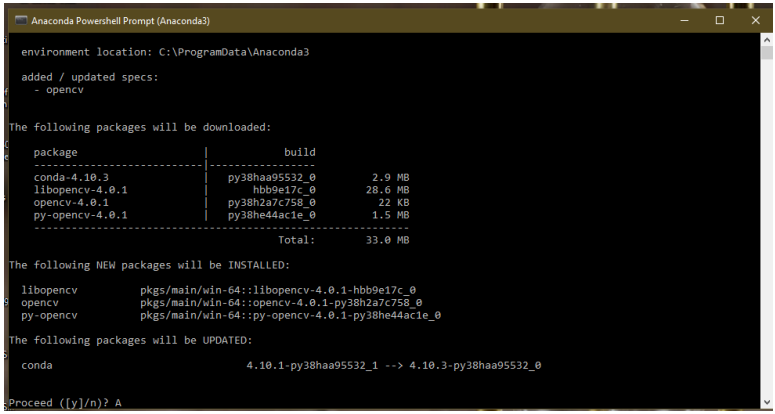


Figure: 16

After successfully installing that package, they ask to proceed this process continuously . by typing “y” yes command you can download successfully package as follow 17 figures

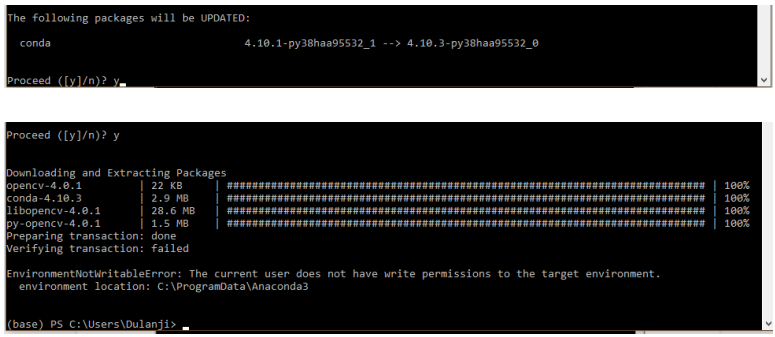


Figure: 17

By typing spyder on the anaconda shell you can open spyder IDE as below figure 18

Figure : 18

And you can open a new python file by clicking at the left top file icon and you can create your code on this. If you successfully installed the OpenCV pipe line. you can check it by typing import command on this python file.

- This anaconda IDE is not like PyCharm, this does not highlight errors on its file. It pops up an error when I run that command.

- Therefore, to check whether your installation is successful, you have to run that code. Before running this code, you have to save that code someplace on the pc.

- Then you can simply copy and paste code above PyCharm python code as follow figure 19. Finally, you can save that code and run by clicking the top green color try angel.

- In this Be sure to save python program code and xml file at the same place.

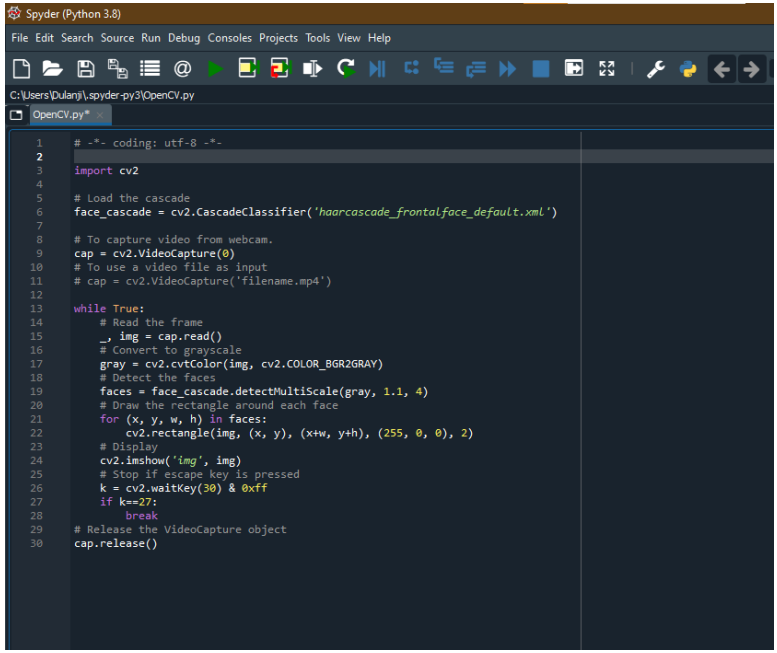


Figure: 19

After running the OpenCV program successfully. you can see the output as figure 20. Look at this program. In this program, the human face is detected by drawing a rectangle around the human face.

The below picture has a stand fan, door and some furniture, but this program is to detect a human face only from any other things.

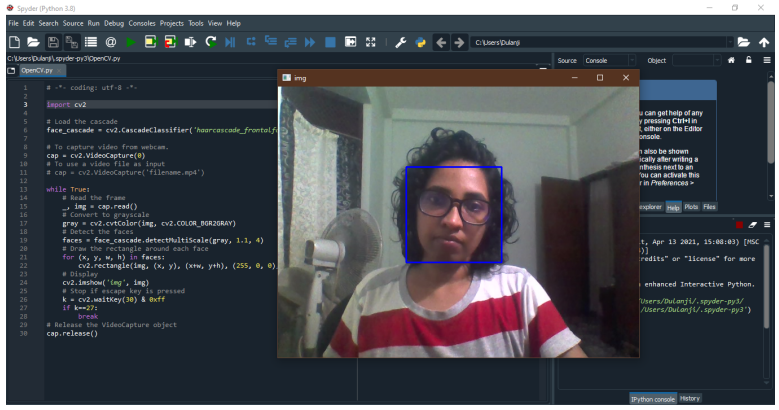
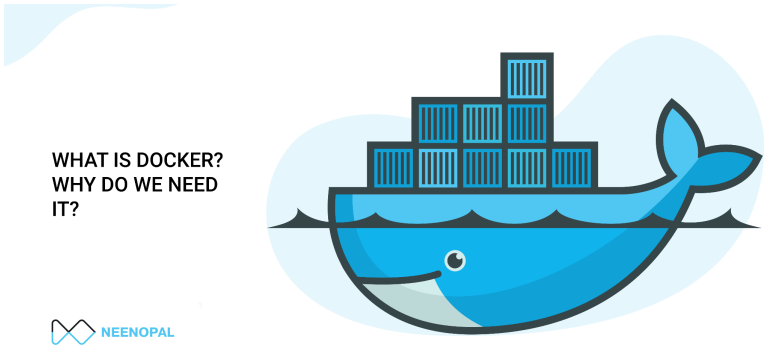

Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Dulanj
OpenCV.py
img
IPython console
History

Figure: 20

Chapter 3

What is Docker?

Docker is an open-source containerization platform for users that need to take docker advantages. It empowers designers to bundle applications into holders—normalized executable parts consolidating application source code with the working framework (operating system) libraries and conditions needed to run that code in any climate. Holders improve on the conveyance of dispersed applications and have become progressively famous as associations shift to cloud-local turn of events and cross breed multi-cloud conditions.



Engineers can make holders without Docker, yet the platform makes it more straightforward, easier, and more secure to fabricate, convey and oversee compartments. Docker is basically a tool compartment that empowers engineers to assemble, convey, run, update, and stop holders utilizing basic orders and work-saving mechanization through a solitary Programming interface.

What would I be able to utilize Docker for?

Quick, reliable conveyance of your applications

Docker smoothest out the advancement lifecycle by permitting engineers to work in normalized conditions utilizing neighborhood compartments which give your applications and administrations. Compartments are extraordinary for persistent joining and consistent conveyance (CI/Cd) work processes.

Think about the accompanying model situation:

Your engineers compose code locally and share their work with their associates utilizing Docker holders.

They use Docker to drive their applications into a test climate and execute robotized and manual tests.

At the point when engineers track down bugs, they can fix them in the advancement climate and redeploy them to the test climate for testing and approval.

When testing is finished, getting the fix to the client is pretty much as basic as pushing the refreshed picture to the creation climate.

Responsive arrangement and scaling

Docker's holder-based stage takes into consideration exceptionally compact jobs. Docker compartments can run on a designer's neighborhood PC, on physical or virtual machines in a server farm, on cloud suppliers, or in a combination of conditions.

Docker's transportability and lightweight nature additionally make it simple to progressively oversee responsibilities, increasing or destroying applications and administrations as business needs direct, in close to continuous.

Running more jobs on a similar equipment

Docker is lightweight and quick. It gives a feasible, financially savvy option to hypervisor-based virtual machines, so you can utilize a greater amount of your register ability to accomplish your business objectives. Docker is ideally suited for high thickness

conditions and for little and medium organizations where you want to accomplish more with less assets.

From this let's discuss the architecture of the docker. Why do we need docker architecture and actually, docker is the culture?

Why uses Docker

Docker is so well known today that "Docker" and "holders" are utilized conversely. Be that as it may, the main compartment related innovations were accessible for quite a long time — even many years (connect dwells outside IBM) — before Docker was delivered to the general population in 2013.

Most remarkably, in 2008, Linux Containers (LXC) was carried out in the Linux bit, completely empowering virtualization for a solitary occasion of Linux. While LXC is as yet utilized today, fresher innovations utilizing the Linux bit are accessible. Ubuntu, a cutting edge, open-source Linux working framework, additionally gives this ability.

Docker improved the local Linux containerization capacities with innovations that empower:

Improved and seamless portability: While LXC holders frequently reference machine-explicit arrangements, Docker compartments run without change across any work area, server farm and cloud climate.

Much lighter weight and more granular updates: With LXC, different cycles can be consolidated inside a solitary compartment. With Docker compartments, just one interaction can run in every holder. This makes it conceivable to construct an application that can keep running while one of its parts is brought down for an update or fix.

Robotized compartment creation: Docker can naturally fabricate a holder dependent on application source code.

Compartment forming: Docker can follow variants of a holder picture, roll back to past adaptations, and follow who constructed

a rendition and how. It can even transfer just the deltas between a current rendition and another one.

Holder reuse: Existing compartments can be utilized as base pictures—basically like formats for building new holders.

Shared compartment libraries: Designers can get to an open-source vault containing a large number of clients contributed holders.

Today Docker containerization likewise works with Microsoft Windows servers. Furthermore, most cloud suppliers offer explicit administrations to help designers fabricate, boat and run applications containerized with Docker.

Docker Architecture

Docker utilizes customer server engineering. The Docker customer converses with the Docker daemon, which does the truly difficult work of building, running, and dispersing your Docker holders. The Docker customer and daemon can run on a similar framework, or you can interface a Docker customer to a distant Docker daemon. The Docker customer and daemon convey utilizing a REST Programming interface, over UNIX attachments or an organization interface. Another Docker customer is Docker Create, that allows you to work with applications consisting of a bunch of compartments. Life following figure 1

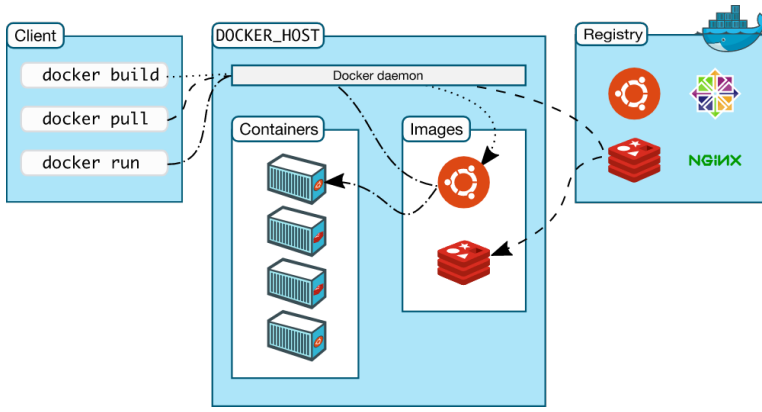


Figure: 1

The Docker daemon

The Docker daemon (dockerd) gives their attention to the API and other so on the work properly. To use that API for managing docker objects, image, network, container and volume. Also, demons have the ability to communicate with other demons to manage and give continued good services for docker users and organizations.

The Docker client

The Docker client (docker) connects with the client using the primary way of communication that interacts with the target or user that is in the docker environment. If you use commands such as docker run, push and pull and so on and also the client sends these commands to dockerd, which carries them out. Also, the docker command is used to be based on the Docker API. Under that API environment the Docker client can communicate with more than one daemon.

Docker Desktop

Docker Desktop is an exe that installs easily. Actually, it is easy to install in MAC or windows because windows do not have the Linux environment to make that environment. We enable WSL2 to make that environment under the windows environment. After that

install docker desktop will help you to make a build, share, roll, pull and push containerized application. In other words, a docker image environment. Docker desktop has been included Docker Desktop includes the Docker daemon (dockerd), the Docker client (docker), Docker Compose, Docker Content Trust, and micro services like cabernet.

Docker registries

A Docker registry is used to store Docker images. Docker Hub is the example for the docker registry and that the public and anyone can access that they have an account in docker hub. Docker hub is a public and open-source registry. In the docker hub store docker image and anyone making docker image that all are pulled on the docker hub under public or private category. You can make your own registry for person wise or organization wise

When you use the docker pull or docker run commands, the required images are pulled from your configured registry. When you use the docker push command, your image is pushed to your configured registry.

Docker objects

When you work with Docker, you have to deal with building, running, pushing and pulling images, containers, networks, volumes, plugins, and other objects. These keywords are called docker objects in the docker environment.

Images

An image is just like a blueprint with read only features. It has included instructions for creating a Docker container. Often, an image is based on the main image, here the main image is, suppose you do a python program but python programs have different versions python 3.8, python 3. 9 Likewise, so many versions have been included in python. If your image builds under version 3.8 your main is 3.8 python image and under that image your python project is built. When you pull an image and run the image your

container makes a python 3.8 environment to run your image successfully.

If you like to create and publish your own image docker hub first of all you have to create dockerfile. That holds all instructions, dependencies and environment information to build your own image, when you create a dockerfile, then dockerfile has included all steps which are used to run docker image and that is defined using specific syntax which can understand only dockerfile. According to that syntax of dockerfile docker image is built. Each instruction in a Dockerfile creates a layer in the image. When you change the Dockerfile and rebuild the image, only those layers which have changed are rebuilt. This is part of what makes images so lightweight, small, and fast, when compared to other virtualization technologies.

Containers

A Container is a runnable occasion of a picture. When you work with containers you handle create, start, stop, move, or erase a container utilizing the Docker Programming interface or CLI. You can associate a compartment to at least one organization, connect capacity to it, or even make another picture dependent on its present status.

Generally, a container is relatively well isolated from other containers and its host machine. You can control how isolated a container's network, storage, or other underlying subsystems are from other containers or from the host machine.

A container is characterized by its image just as any design choices you give to it when you make or start it. At the point when a holder is eliminated, any progressions to its express that are not put away in tireless capacity vanish.

As below the figure 2 is shown that the structure of the application is worked on under the docker containerized environment.

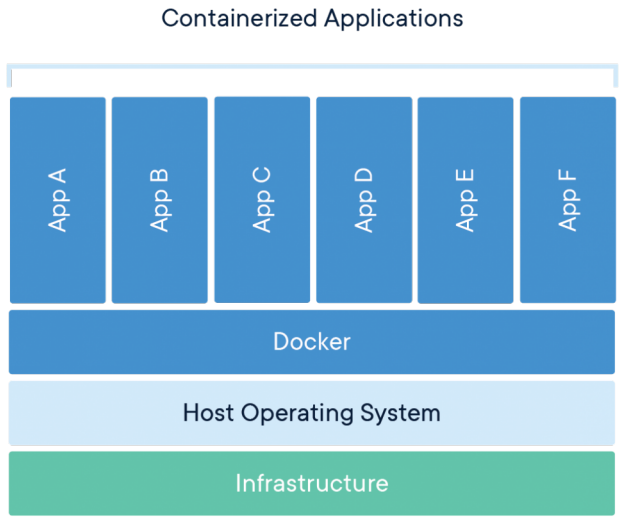


Figure: 2

Then we pay attention to the difference between a container environment and a virtual environment. What are the behaviors of the VM and container?

Containers are made conceivable by process seclusion and virtualization abilities incorporated into the Linux portion. These abilities -, for example, control gatherings (groups) for assigning assets among processes, and namespaces for confining a cycles access or perceivability into different assets or spaces of the framework - empower numerous application parts to share the assets of a solitary occurrence of the host working framework similarly that a hypervisor empowers various virtual machines (VMs) to share the central processor, memory and different assets of a solitary equipment server.

Accordingly, holder innovation offers all the usefulness and advan-

tages of VMs - including application confinement, savvy versatility, and superfluity - in addition to significant extra benefits:

Lighter Weight: In contrast to VMs, compartments don't convey the payload of a whole operating system case and hypervisor; they incorporate just the operating system cycles and conditions important to execute the code. Holder sizes are estimated in megabytes (versus gigabytes for some VMs), utilize equipment limits, and have quicker startup times.

More prominent asset proficiency: With compartments, you can run a few folds the number of duplicates of an application on similar equipment as you can utilizing VMs. This can lessen your cloud spending.

Further developed engineer usefulness: Contrasted with VMs, holders are quicker and more straightforward to convey, arrange and restart. This makes them ideal for use in consistent coordination and nonstop conveyance (CI/CD) pipelines and a superior fit for improvement groups embracing Deft and DevOps rehearsals.

Virtual Machine Vs Container

In conventional virtualization, a hypervisor virtualizes actual equipment. The outcome is that each virtual machine contains a visitor operating system, a virtual duplicate of the equipment that the operating system needs to run and an application and its related libraries and conditions. VMs with various working frameworks can be run on a similar actual server. For instance, a VMware VM can run close to a Linux VM, which runs close to a Microsoft VM, and so forth

Rather than virtualizing the fundamental equipment, holders virtualize the working framework (regularly Linux or Windows) so every individual compartment contains just the application and its libraries and conditions. Holders are little, quick, and compact in light of the fact that, in contrast to a virtual machine, compartments don't have to remember a visitor operating system for each

occasion and can, all things considered, essentially influence the highlights and assets of the host operating system.

Very much like virtual machines, holders permit engineers to further develop central processor and memory usage of actual machines. Holders go significantly further, nonetheless, in light of the fact that they additionally empower microservice designs, where application parts can be conveyed and scaled all the more granularly. This is an alluring option in contrast to increasing a whole solid application in light of the fact that a solitary part is battling with the heap. Below figure 3 demonstrates the structure furthermore.

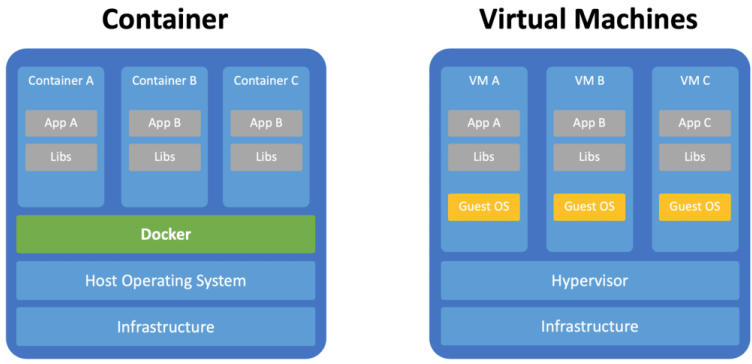


Figure: 3

After following this chapter, now you have brief knowledge about the docker, its environment and its keyword and usage. Then the next chapter talks about advantages and disadvantages about docker.

#

#

Chapter 4

Docker environment and its Advantages and Disadvantages of Docker.

Advantages of Docker

Following are a few benefits of Docker, how about we examine them exhaustively

Profit from Venture and Cost Investment funds

Docker's first benefit is return for money invested. Particularly for enormous, set up organizations, which need to create consistent income over the long haul, the arrangement is just better assuming that it can drive down costs while raising benefits.

Fast Arrangement

It can diminish sending to seconds. It is a direct result of the way that it can make a compartment for each interaction and even doesn't boot an operating system. Along these lines, even without stressing over the expense to bring it up once more, it would be higher than whatever is reasonable, Information can be made just as annihilated.

Security

Docker ensures that applications that are running on holders are totally isolated and separated from one another, according to a security perspective, by allowing us unlimited authority over traffic streams and the board.

Effortlessness and Quicker Arrangements

The manner in which Docker improves on the issues is one of its vital advantages. It gives adaptability to clients to take their own arrangement, put that into the code, and further convey it with next to no issues.

Be that as it may, the prerequisites of the foundation are not generally connected with the climate of the application, as Docker can be utilized in a wide assortment of conditions.

CI/CD Pipeline Productivity

With the assistance of Docker, we can fabricate a holder picture and can additionally utilize that equivalent picture over each progression of the arrangement cycle.

The upside of it is the capacity to isolate non-subordinate advances and furthermore run them in equal amounts. Furthermore, the length of time it takes from work to creation might accelerate strikingly.

Constant Mix

With regards to Constant Mix, Docker functions admirably as a feature of its pipelines alongside devices like Travis, Jenkins, and Wercker.

These devices can save the new form as a Docker picture, each time our source code is refreshed, simply label it with an adaptation number and push to Docker Center, then, at that point, send it to creation.

Cons of Docker

A few weaknesses of Docker are examined, here:

Missing elements

A huge load of element demands is under progress, similar to compartment self-enlistment, and self-assesses, duplicating records from the host to the holder, and some more.

Information in the compartment

There are times when a compartment goes down, so from that point onward, it needs a reinforcement and recuperation procedure, in spite of the fact that we have a few answers for that they are not mechanized or not truly adaptable yet.

Run applications as quick as an uncovered metal serve

In examination with the virtual machines, Docker compartments have less overhead however not zero overhead. In the event that we run an application straightforwardly on an exposed metal server we get genuine uncovered metal speed even without utilizing holders or virtual machines. In any case, Compartments don't run at exposed metal paces.

Give cross-stage similarity

The one significant issue is assuming that an application is intended to run in a Docker compartment on Windows, then, at that point, it can't run on Linux or the other way around. In any case, Virtual machines are not exposed to this impediment.

Along these lines, this constraint makes Docker less appealing in some profoundly heterogeneous conditions which are made out of the two Windows and Linux servers.

Run applications with graphical interfaces

As a rule, Docker is intended for facilitating applications that suddenly spike in demand for the order line. However, we have a couple of ways (like X11 sending) by which we can make it conceivable to run a graphical interface inside a Docker holder, nonetheless, this is cumbersome.

Henceforth we can say, for applications that require rich interfaces, Docker is anything but a decent arrangement.

Tackle all your security issues

In basic words, we really want to assess the Docker-explicit security dangers and ensure we can deal with them prior to moving responsibilities to Docker.

The purpose for it is that Docker makes new security challenges like the trouble of observing various moving pieces inside an enormous scope, dynamic Docker environment.

Chapter 5

Install Docker on Windows Environment.

First of all, you should download docker desktop to your PC. Using this link, you can download docker for window environment

Actually, you can pull and push your image using CMD on your computer without a docker desktop environment. But docker desktop environment is easy for you handle all stuff of docker

Link for docker desktop download to widows [https://docs.docker.com/desktop/wind

Once you click the link this link automatically redirects below page as figure 1.

Figure: 1

And then you can install docker desktop exe on your PC. Before running docker desktop exe we have to check some performance and WSL2 backend on your computer. Now we are going to check that parts

Before enable WSL2 you should checking windows version, you can do it hit start and type “winer” then you can see about window as figure 2

Figure: 2

Figure 3 is shown basic requirement that is you should complement to run docker desktop on your computer.

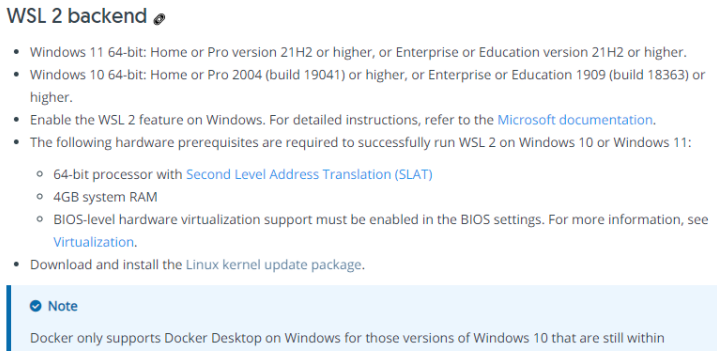


Figure: 3

First or second requirements are eligible on your PC then you can enable WSL 2 environment. Next steps are dedicated to enabling the WSL2 environment on your PC.

Actually, Docker Desktop Document gives you all details to install docker desktop on your PC. But here I give a demonstration guideline step by step on how to do it correctly without wasting your time.

https://docs.microsoft.com/en-us/windows/wsl/install-manual Following this link you can easily access that command steps to install WLS2

Step 1: Actually, WSL is making a Linux environment on top of windows to run docker desktop.

Therefore, first of all we should enable windows system to Linux

We can run this command on CMD or PowerShell on windows, however before running this command we should open administrative mode. You can right click on CMD and give it permission to run as administrative mode like figure 4. Same process you have to do, if you use a power shell.

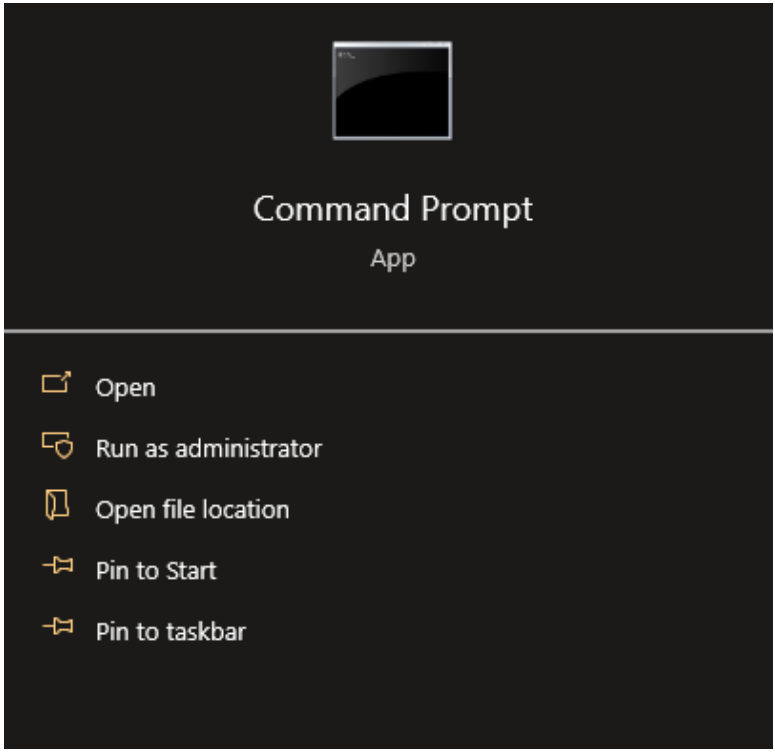


Figure: 4

And then you can write " **dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart**" in front of CMD as figure 5.

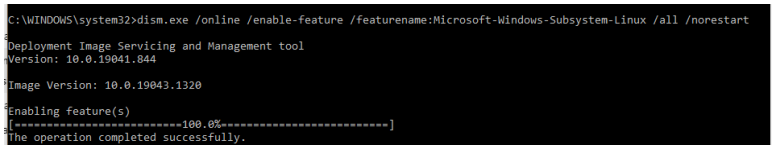


Figure:5

Step 2: checking requirements to run WSL2. Actually, this step is already done. Figure is show how to do it.

If I remember again that command, if you run WLS then you must complete following

- For x64 systems: Version 1903 or higher, with Build 18362 or higher.

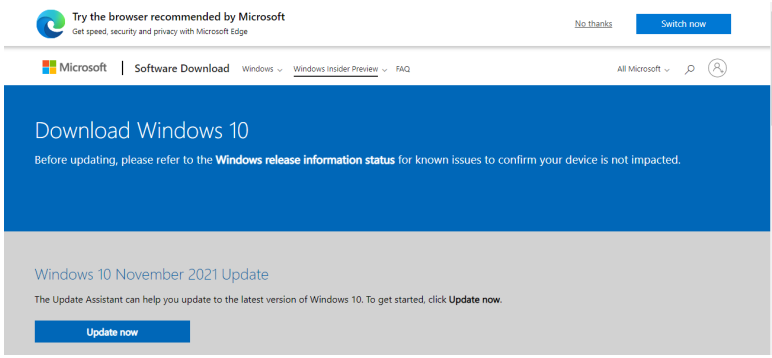
either,

- For ARM64 systems: Version 2004 or higher, with Build 19041 or higher

Requirement, if not complete that environment you have to go and update your windows by the latest version. Otherwise, you cannot run make WSL2 environment on your computer.

If you cannot update your system using normal component then you can use this below site and complete your updating

Link for windows updating https://www.microsoft.com/en-us/software-download/windows10



Also, System type find by typing this command i **“systeminfo | find “System Type” “on** PowerShell or CMD without run as administrator mood as figure 6

Figure: 6

Step 3: In this step you have to enable virtual machine features to your computer. Because, without virtual machine features you cannot run WSL 2 on your PC under windows environment.

To enable this feature, you have to run "**dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart**" on your computer in CMD or PowerShell under an administrative environment.

Figure 7 is shown how to run that command.

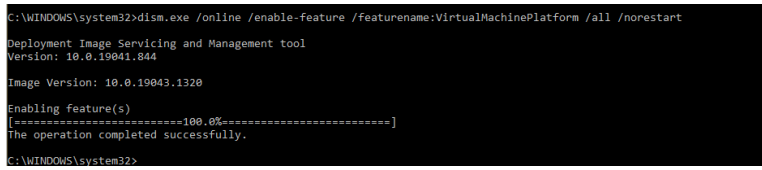
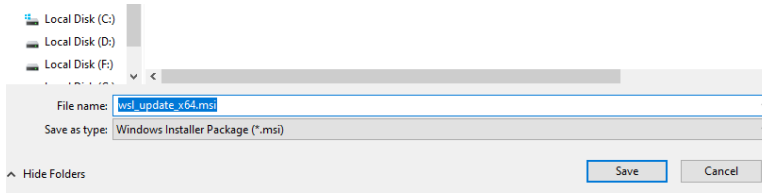


Figure: 7

Step 4: In this step you have to download Linux kernel package you can download that package using this below link

Linux kernel package download [WSL2 Linux kernel update package for x64 machines](https://wslstorestorage.blob.core.windows.net/wslblob/wsl_update_x64.msi)

After clicking this command, you can download this **wsl_update_x64.msi** package on your PC on that you selected place. Like below image



After doing that process you have to restart the PC to install WSL and update WSL 2 as figure 8

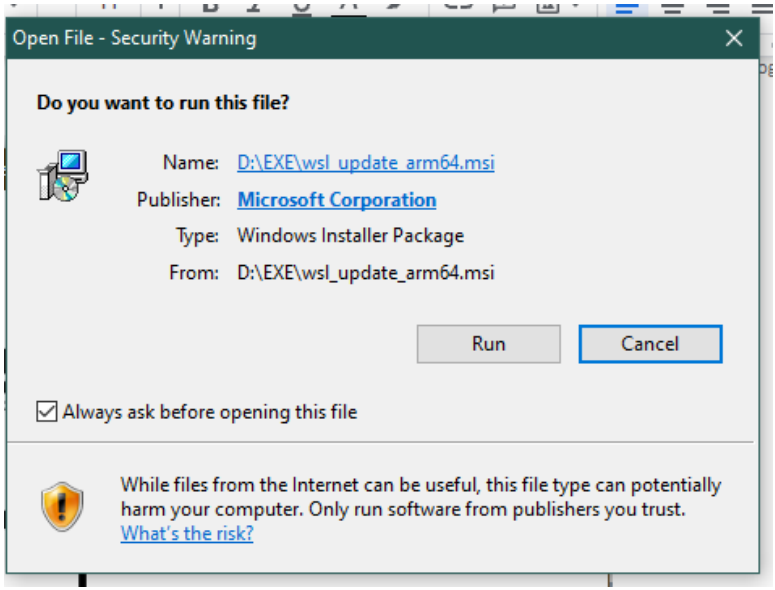


Figure:8

You can click the run command and give permission to run this process on your computer. This have taken two three minutes to complete that task and after that complete setup of Windows Subsystem of Linux you can finish your task as figure 9

What is this WLS, actually that gives permission to Linux containers to run without an emulator. Users can leverage Linux workplace without maintaining both Linux and windows build scripting environment. Because docker desktop runs on top of the WLS 2 environment. Also, WSL2 provides file system sharing, boot time and many more other features to docker desktop users.

Furthermore, WSL2 provides for docker desktop users dynamic memory allocation features and it is easy to improve resource assumption. If we furthermore explain about resource assumption

that means, docker desktop consumes the required amount of CPU and memory resource that is needed for doing relevant work.

By enabling CPU and memory intensive tasks docker desktop can easily and faster build containers for making environments to pull images. Also, WSL 2 helps to start docker daemon after the start work in a cool environment within a small-time duration.

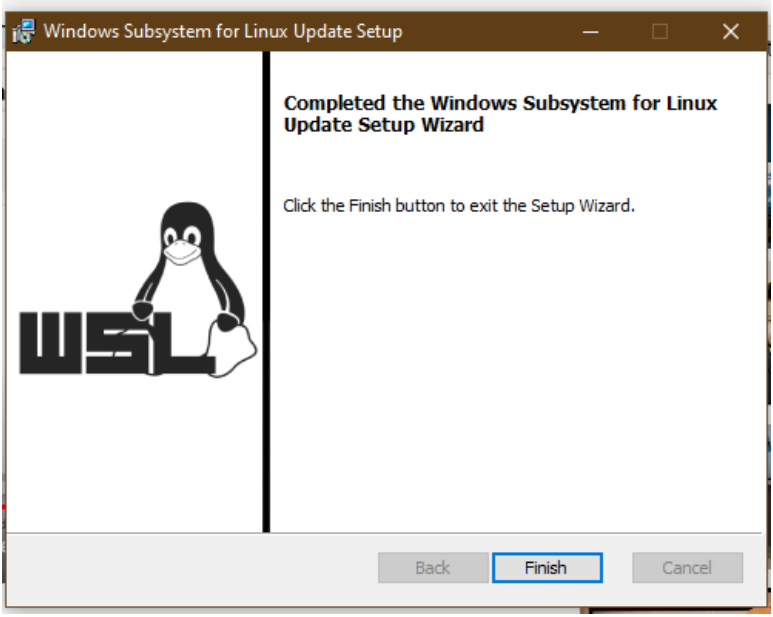


Figure: 9

Step 5: About you going to know about WSL 2 and what is responsibility and what are the benefit that fix on the windows environment in this step we are going to set WSL 2 as a default version to run docker desktop

For this we have to run "**wsl -set-default-version 2" this** command at top of the CMD or PowerShell under the administrative mood. Figure 10 shows how to run it on the PowerShell. You know step 3 was to run on CMD. But this step I run on PowerShell. Because I

want show how that run on both environment at same time what are the environment of the PowerShell in administrative mood

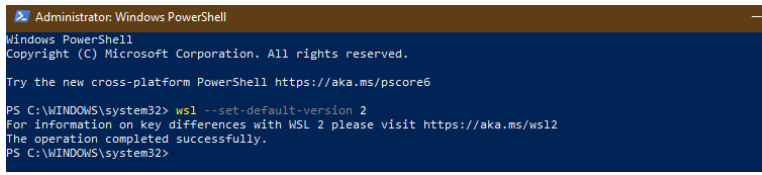
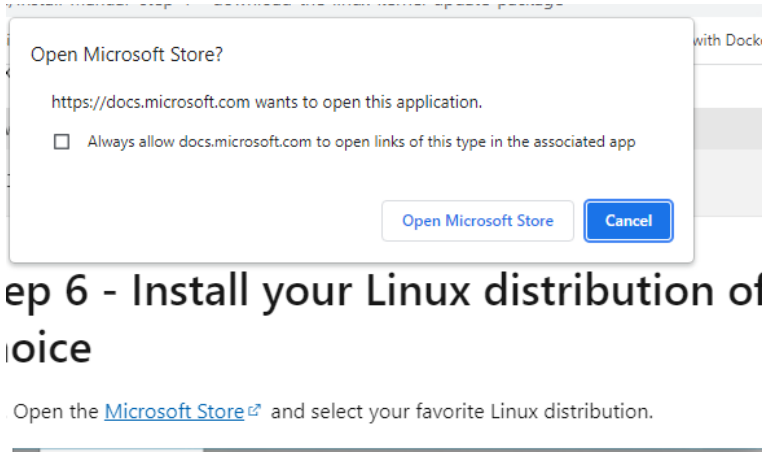


Figure: 10

Step 6: Here we install Linux distribution according to your choice. You can open Microsoft store using this link

Link for Microsoft store [Microsoft Store](https://aka.ms/wslstore)



And choose your favorite Linux distribution such as Debian, kali, ubuntu etc. However, here I choose Ubuntu as my distribution environment and figure 11 is shown about that.

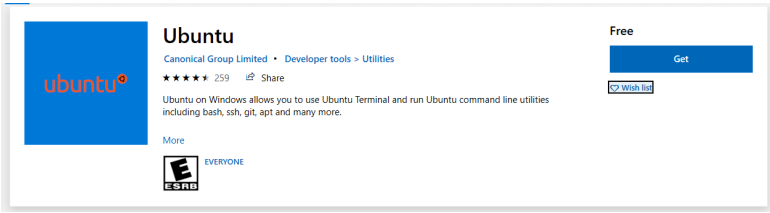


Figure: 11

Figure 12 shows the ubuntu installation to my PC

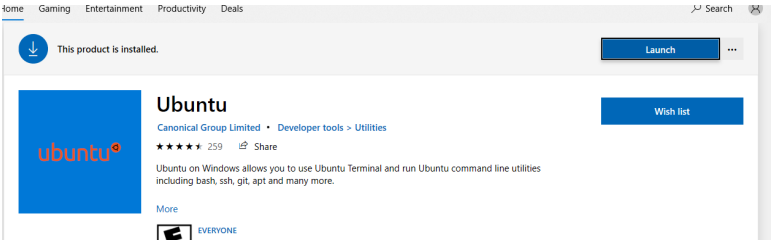


Figure: 12

After completing download you can launch it by clicking the launch button as above figure 12 shown.

Step 7: After completing, launch Ubuntu on your PC. You can see the ubuntu CMD as figure 13. You can open ubuntu CMD by searching on application

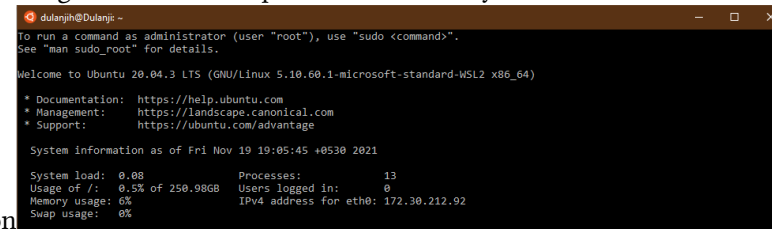


Figure: 13

After open the Ubuntu CMD, first of all you have to create your username and password on ubuntu environment like figure 14 as below

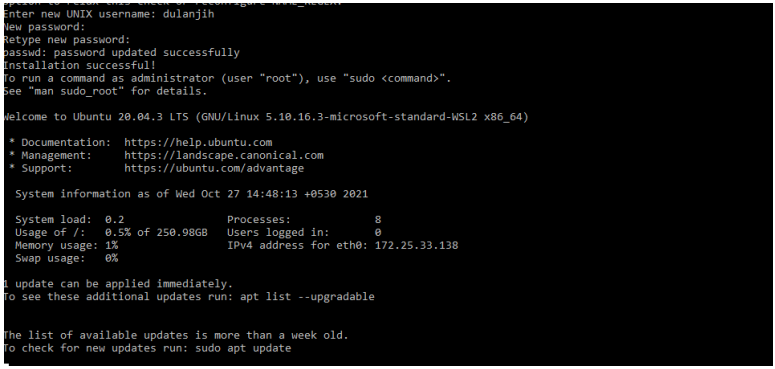


Figure: 14

Step 8: This is the time to install docker on your PC. You can download docker desktop exe from docker hub. In the beginning of this chapter, I already mention that site and link to it.

But here I again put a link for download for docker desktop exe for your easy usage. Link for download docker desktop exe [https://docs.docker.com/desktop/windows/install/](https://docs.docker.com/deskto
. After double clicking in front of the exe you can run docker desktop for windows as figure 15.

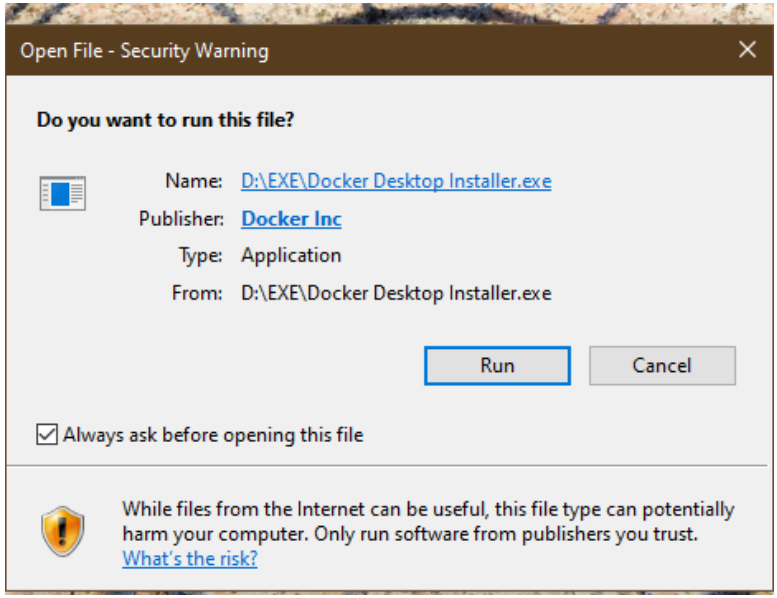


Figure: 15

After running exe, you can see the configuration part as figure 16 and you can select both of the configurations and click the ok button.

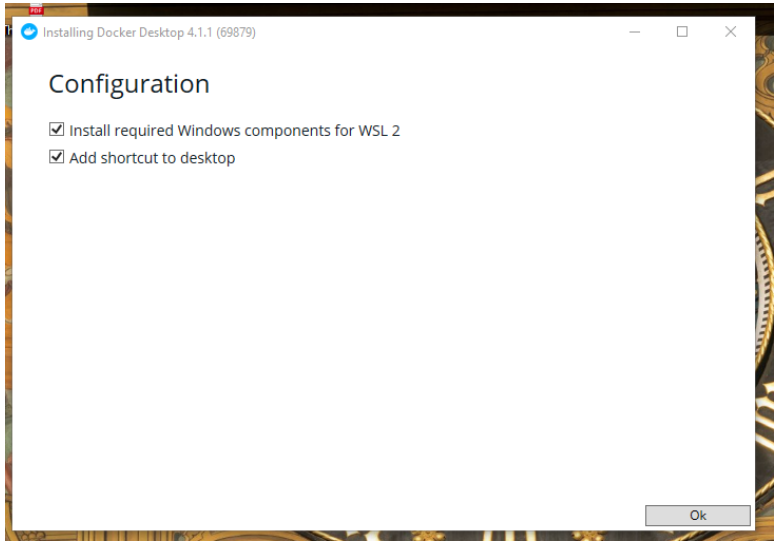


Figure: 16

And then you have to wait a couple of minutes until unpacking the file as figure 17 is shown. This time capacity is dependent on your computer performance.

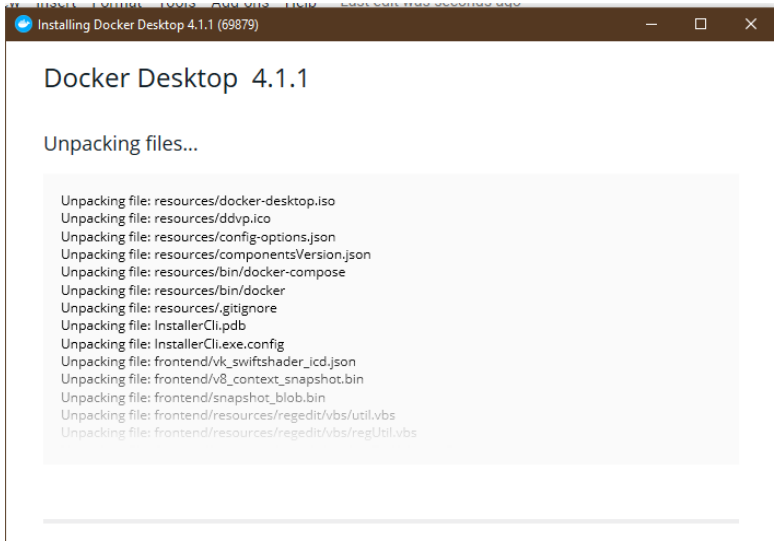


Figure: 17

When completing all the configuration, you can see a view like figure 18 as below. After that you have to restart your PC to complete installation. you can restart the process by clicking this blue color button on the view.

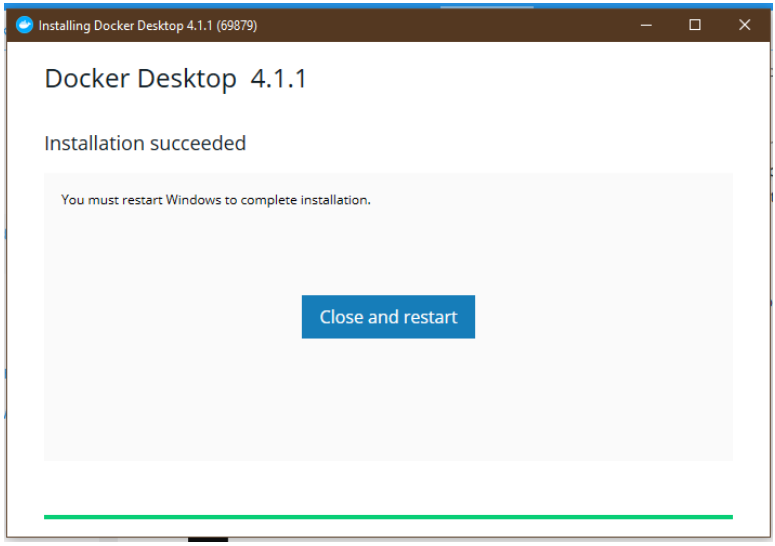


Figure: 18

Step 9: After restarting the PC, before opening docker desktop we can correctly check the docker install on the PC by typing command on CMD or PowerShell "**docker**" without running an administrative mood. Which proof docker is already installed on your machine as figure 19 below.

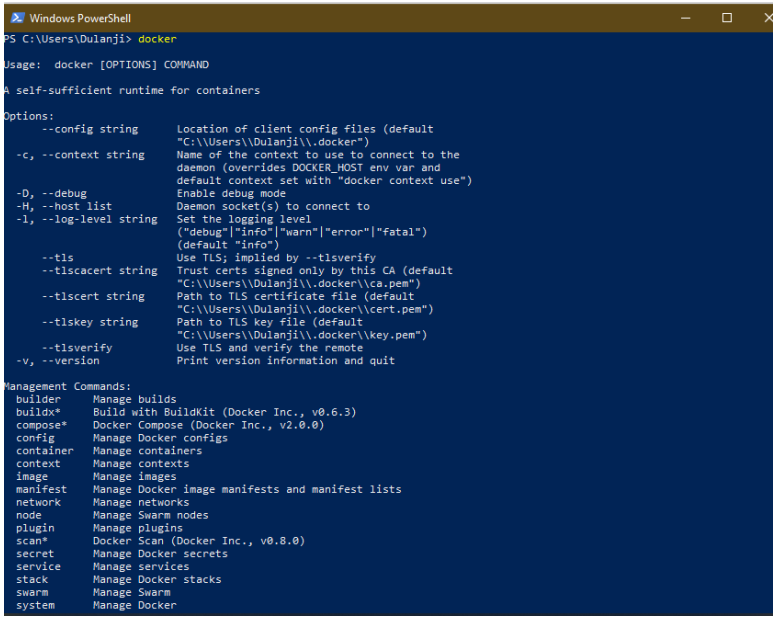


Figure: 19

And furthermore, if you properly check docker you can type ***"docker image ls"*** command on windows power shell. And then you can see the error ***"docker daemon is not running"***. Because we have no image in our docker environment as figure 20

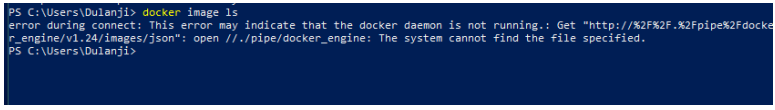


Figure: 20

Also, you can check docker version using "**docker -v**" command on top of CMD or PowerShell without run as administrative mood

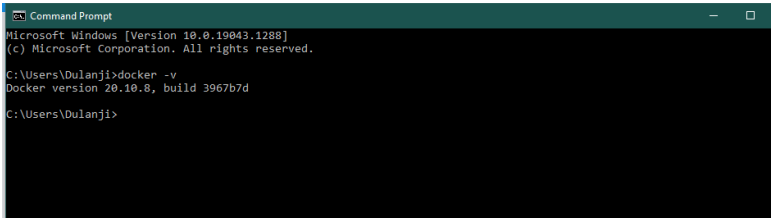


Figure: 21

Step 10: If you already do not have docker account you can Sign Up using below link

Sign up for decker engine you can use [https://hub.docker.com/](https://hub.docker.c

.

When you click the link, you can see below figure 22 interface

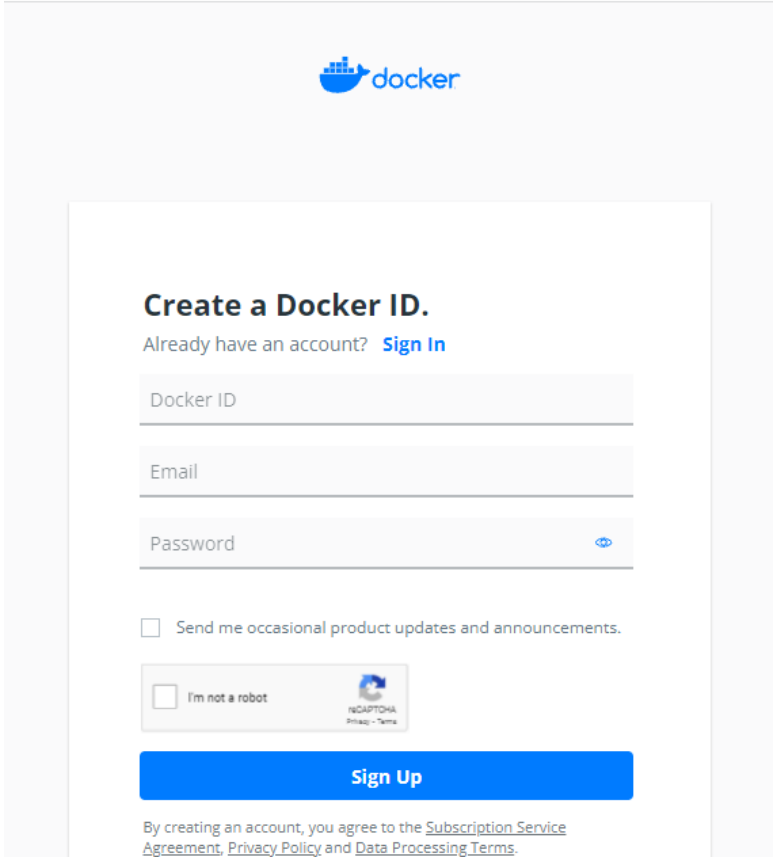


Figure: 22

However, if you already have docker ID otherwise docker hub account then you can sign in before open decker desktop.

After restarting, when you open the docker desktop again you have to accept terms and condition as below figure 23

Figure: 23

After accepting terms and conditions you can see Sign in with docker ID pop as following figure 24. You can give already created docker ID credentials to the sign in process.

Figure: 24

After that login, if you successfully install and configure all things you can see docker desktop interface like as following figure 25.

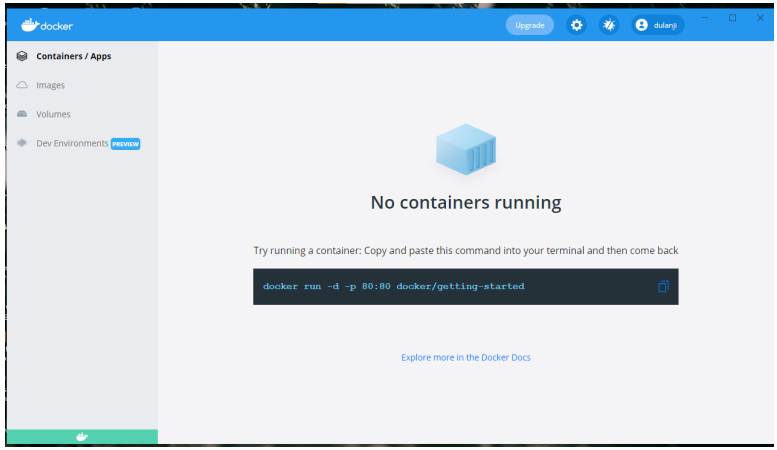

docker
Upgrade
dulanji
Containers / Apps
Images
Volumes
Dev Environments
PREVIEW
No containers running
Try running a container: Copy and paste this command into your terminal and then come back
docker run -d -p 80:80 docker/getting-started
Explore more in the Docker Docs

Figure: 25

#

#

#

#

Chapter 6

How to create docker Image?

Before pushing the image to the docker hub we should create a dockerfile.

For creating this file, first of all I make a folder by using the "**mkdir Dockerfiles**" command on top of CMD or PowerShell. as figure 1 is shown.

```
C:\Users\Dulanji>cd Desktop

C:\Users\Dulanji\Desktop>mkdir DockerFiles
```

Figure: 1

This is an additional step if you want to go to some specific location from the CMD on the PC you can use the "**cd" command with specific location**. Then you can go to that file location and make your folder or text file.

As figure 2 you can see the DockerFile folder as we make on using CMD command

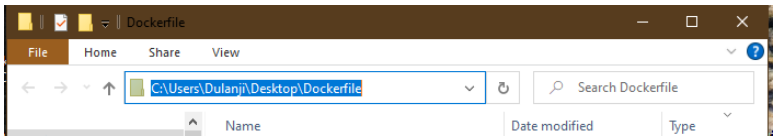


Figure: 2

Next step now let's see how to create dockerfile by using CMD in front of the dockerfile folder we already created.

By giving "**cd DockerFiles**" you can go to the inside of the docker file and then as next command you can type "**null>Dockerfile**" command on top of the CMD to create a dockerfile in DockerFiles folder. As figure 3.

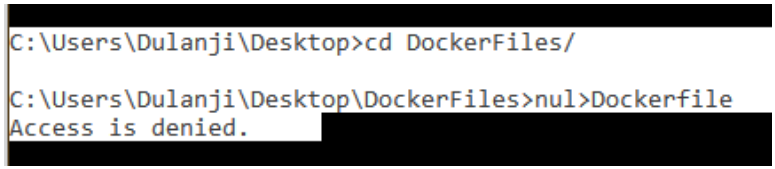


Figure: 3

As following figure 4 you can see we created Docker file by using CMD

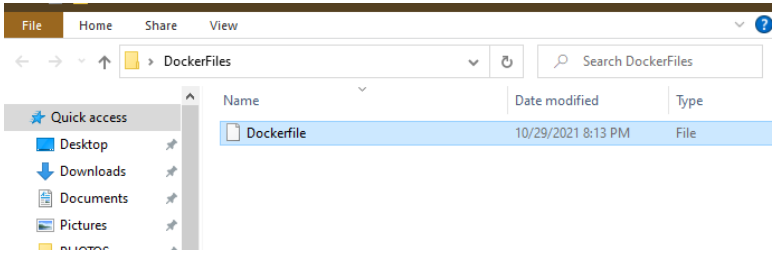


Figure: 4

Then as next step we have to open this Dockerfile on visual studio code and type docker commands. As follows figure 5.

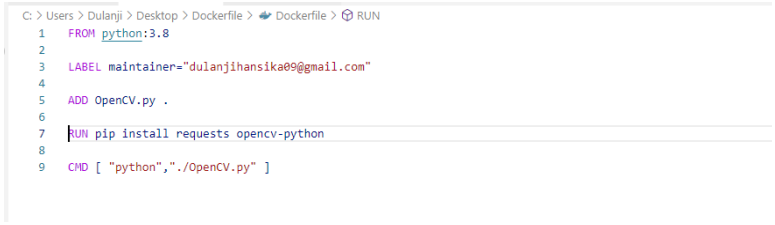


Figure: 5

Now let's see what this dockerfile command is and how it is worth

on this program. Already mentioned as chapter 3 dockerfile has included all workflows to build a docker image. Therefore, docker image is built according to the dockerfile instruction.

Now let's see what these instructions mean.

Ohhhh…! I forgot to tell you that I already created the openCV.py file and xml file I put into the dockerfile. Because, it is for me to access from the command line, because all the needed files are placed in the same location. If you do not like to do this. You can place your project file and other things in a separate location. But when you access that thing, you have to mention those things with a location path.

After you create a dockerfile you have to write that workflow in the dockerfile. For these we use some key work now, let's see what are the meanings from these keywords.

If a lot of keywords do a lot of things in the dockerfile, however, to build my docker image I used several keywords only, so here, I explained about that keywork. What is the responsibility of that keyword here?

Actually, I used below five keywords only.

FROM python:3.8

LABEL maintainer="dulanjihansika09@gmail.com"

ADD OpenCV.py .

RUN pip3 install requests opencv-python

CMD ["python3","./OpenCV.py"]

FROM: Specify the parent image that you intend to build and initialize the stage of the new build image. Here I build a python image therefore, i should mention what version of the build image. You can use these three formats for "FROM" instruction to specified build stage

FROM [-platform=<platform>] <image> [AS <name>]

Or

FROM [-platform=<platform>] <image>[:<tag>] [AS <name>]

Or

FROM [-platform=<platform>] <image>[@<digest>] [AS <name>]

**

LABEL:** As a key, value pair we can add metadata using the label instruction. Be sure to keep space between label and value. Also, for one image you can put more than one label and one label you can put several key value pairs. Further, you can add multiple labels in multiple lines or single label values.

LABEL multi.label1="value1" multi.label2="value2" other="value3"

**LABEL multi.label1="value1" **

**multi.label2="value2" **

other="value3"

Here I use a label to give information about who maintains this program. According to this information when you make an image here is put the name or organization set as author of the image.

MAINTAINER <name>

ADD: In this instruction copy file, folder and online URL forms and also those are added to the filesystem of image at path <dest> in this dockerfile I added python.py file as an added document.

**

RUN:** In this command execute all commands in front of the base image to create a new image. In this instruction we should mention import libraries and other things that need to be done to create a new docker image.

**

CMD:** The main responsibility of CMD instruction is to provide

default instruction to create containers. Here, to run this program you must want a python environment. If anyone pulls this image, they container must make a python environment to run the downloaded image. By this line make that environment to run docker image.

MD [”executable”,”param1”,”param2”] (exec form, this is the preferred form)

CMD [”param1”,”param2”] (as default parameters to ENTRYPOINT)

CMD command param1 param2

Ok … next let’s see how to create a docker image by using this dockerfile. Before making a docker image you should create a repository in docker hub.

I think you already downloaded and set docker desktop you must have a docker hub account. Otherwise, you cannot access your docker desktop application.

If your docker hub session expired you can sign in again and go to the repository tab then you can see the UI like below figure 6.

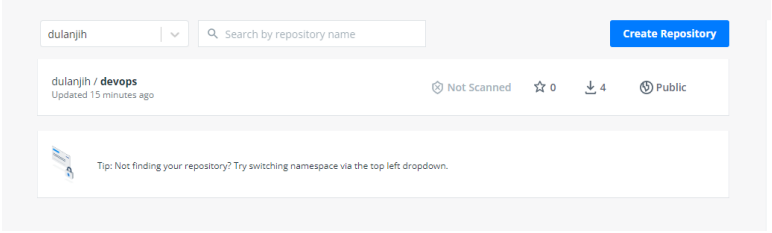


Figure: 6

You can create a new repository by clicking the right side top blue color “create Repository” button.

After clicking the create repository button you can see figure 7 UI as below. And next you can create a new repository by setting name and description in relevant areas.

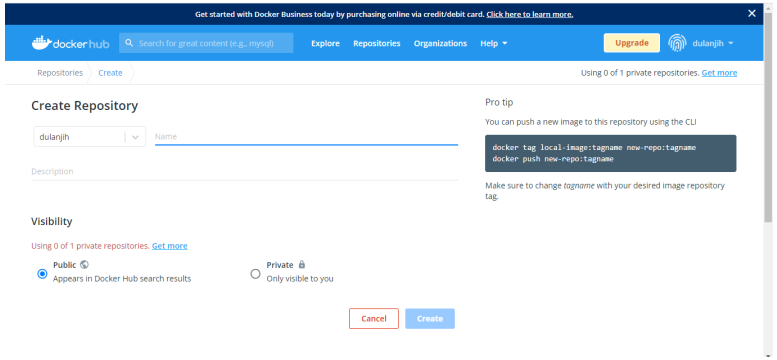


Figure: 7

And also, you can set privacy as public or private by picking the checkbox at the bottom.

After creating the repository, you can see the UI like below figure 8. After creating repository, they will automatically create the dedicated docker push tag for the repository.

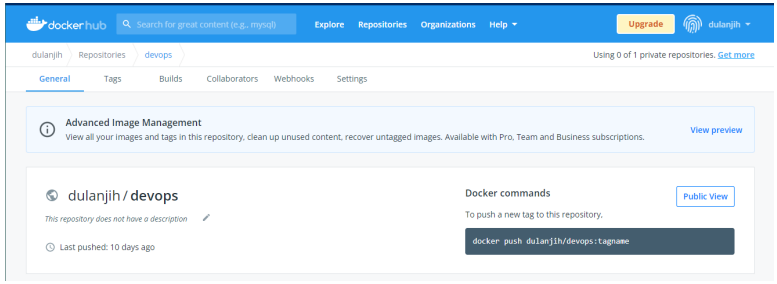


Figure: 8

Now we set up an external environment to build the image. At this point we talk about how to build a docker image in a windows environment.

For this first of all you should open CMD, PowerShell (not need to open as an administrative mood) or visual studio code.

If you use PowerShell or CMD first of all you should login to docker

hub by using the "**docker login**" command as below figure 9. Then as the next step they ask to login username and password. You should provide correct credentials for it. If your credentials are correct as a next step CMD or PowerShell show a "**login succeeded**" message.

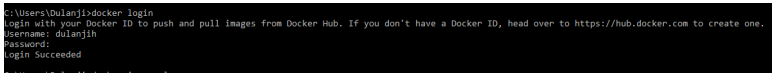


Figure: 9

And then you should go to the dockerfile located folder using cd command by typing top of the CMD or PowerShell. Command is "**cd Desktop**" and "**cd Dockerfile**" **like** figure 10

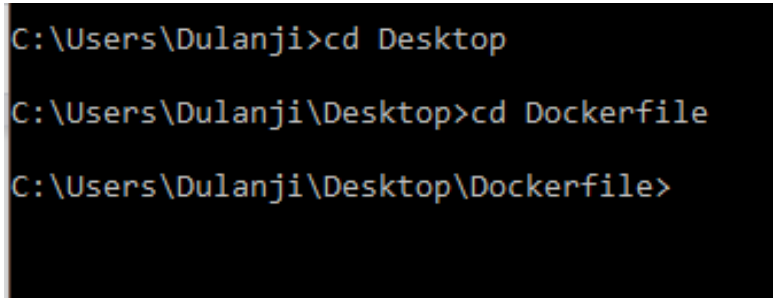


Figure: 10

Then your CMD is in your dockerfile and then by typing ***"docker build -t dulanjih/devops"*** command you can build a docker image. Docker image is built according to the dockerfile information. Actually, here dulanji/devops is docker image name. To create a dockerfile takes two to three minutes. But it depends on your PC performance. These steps have been shown below figure 11.

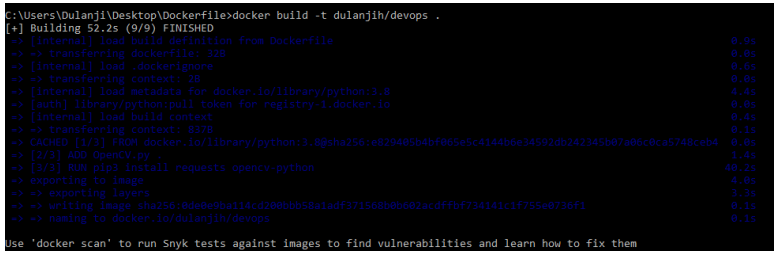


Figure: 11

Also, if you use the visual studio code terminal. First of all, open dockerfile and then open a new terminal. After that you can type cd command to change directory to find dockerfile location. And then you can type ***"docker build -t dulanjih/devops"*** command to build the image as figure 12.

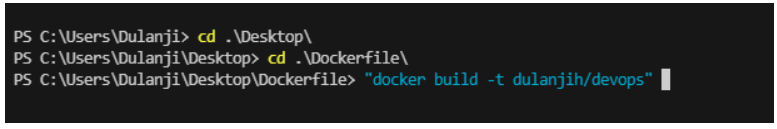


Figure: 12

If you are correctly doing this process. Now your docker image must build successfully. You can see it by opening the docker desktop image tab as figure 13 below.

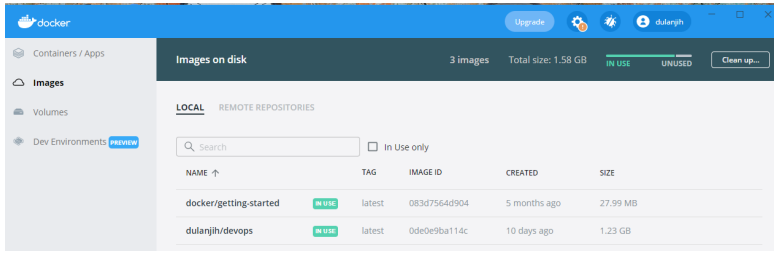


Figure: 13

Chapter 7

How to Push an OpenCV program to Docker Hub as Image?

If you correctly created the docker image you can see on it your docker desktop image tab as the last chapter last image.

In this step you can easily push your docker image into docker hub by clicking the push hub button. It is visible to you when you click the top on the three dots icon of the relevant image. As figure 1

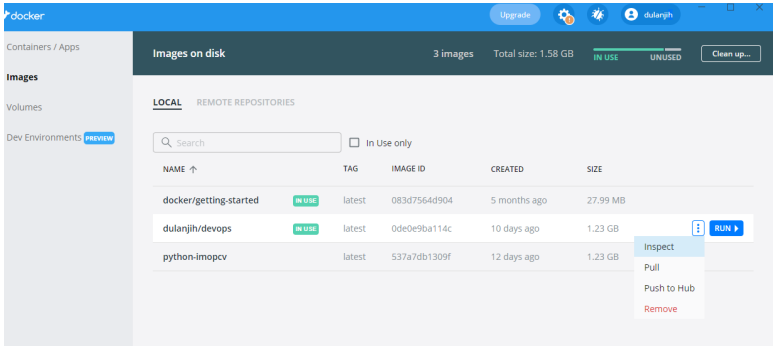


Figure:1

Also, you can easily pull, push, run and remove images using this docker hub. however, to do this all functions your docker desktop have an image.

Before you push the image, be sure to make a repository on top of the docker hub. Otherwise, you docker image pushing things not doing successfully. And also, check again you are signed in on docker hub.

After clicking the push to hub button then you can see the image is pushing to the docker hub like figure 2 below.

If you have instruction further instruction regarding your image you can put in README area which has below at the repository

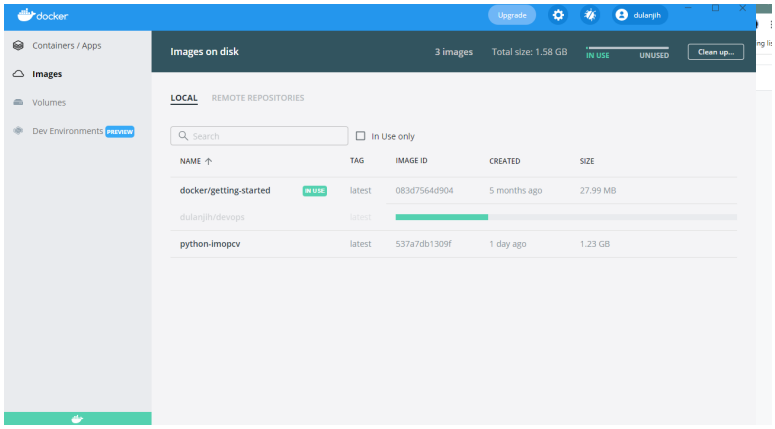


Figure: 2

To complete this process, docker desktop takes two to three minutes. However, it depends on your PC features.

If you are willing to do this process using command line, Power-Shell or visual studio code terminal. You can use the "**docker push <hub-user>/<repo-name>:<tag>**" command and push your image to the already created repository.

If you use command line (CMD) or PowerShell, be sure to login to docker hub before pushing the image. You can use the "docker login" command to sign in to the docker hub. Already I mentioned the process of creating an image chapter. Therefore, here I do not go to explain in detail about that part.

When you do all processes correctly now you can see the pushed image on your created repository as figure 3 below.

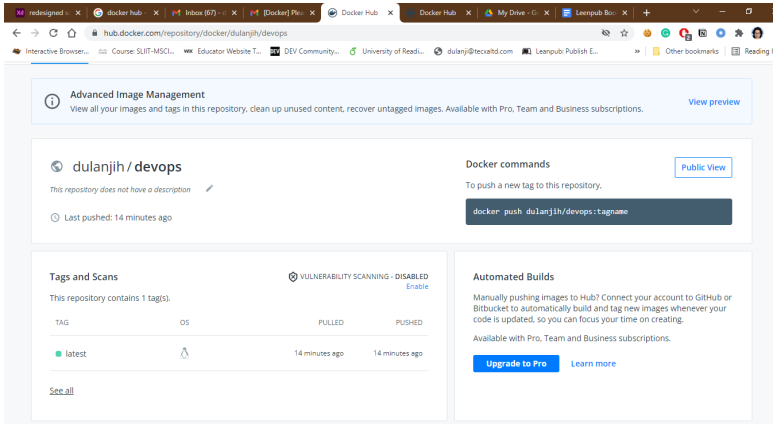

hub.docker.com/repository/docker/dulanjih/devops
Advanced Image Management
View all your images and tags in this repository, clean up unused content, recover untagged images. Available with Pro, Team and Business subscriptions.
View preview
dulanjih / devops
This repository does not have a description
Last pushed: 14 minutes ago
Docker commands
Public View
To push a new tag to this repository,
docker push dulanjih/devops:tagname
Tags and Scans
VULNERABILITY SCANNING - DISABLED
Enable
This repository contains 1 tag(s).
TAG
OS
PULLED
PUSHED
latest
14 minutes ago
14 minutes ago
See all
Automated Builds
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.
Available with Pro, Team and Business subscriptions.
Upgrade to Pro
Learn more

Figure: 3

#

Chapter 8

What is CI/ CD Pipeline?

Actually, what is this pipeline and how does it work?

CI/ CD abbreviation is Continuous Integration and Continuous Delivery. As its abbreviation CI/CD pipeline is collection of steps used to deliver new versions of software or series of software.

CI/CD pipelines give their attention to improving the software outcomes by using DevOps or site reliability engineering (SRE) approaches.

A basic idea of the introduction of the CI/CD pipelines is to improve the process of application development by monitoring and automating their approaches, by doing parallel integration and testing phases also, by doing delivery and development of software.

Although it is capable of executing manually at top of the software, each and every of the steps of a CI/CD pipeline, the true value can be determined through doing that each and every process by using the automation tool of CI/CD pipelines.

Sonar Cloud and Jenkin are the automated tools for doing CI/CD pipeline things. However, Jenkins is open source and it has a self-contained java-based program with packages for Windows, macOS, and other Unix-like operating systems. Also, it has hundreds of plugins and Jenkins is supported for building, deploying, and automating for software DeVos projects.

Figure 1 has shown several CI/CD pipeline automated tools and their behaviors.

	Jenkins	circleci	TeamCity	Bamboo	GitLab
Open source	Yes	No	No	No	No
Ease of use & setup	Medium	Medium	Medium	Medium	Medium
Built-in features	3/5	4/5	4/5	4/5	4/5
Integration	★ ★ ★ ★ ★	★ ★ ★	★ ★ ★ ★	★ ★ ★	★ ★ ★ ★
Hosting	On premise & Cloud	On premise & Cloud	On premise	On premise & Bitbucket as Cloud	On premise & Cloud
Free version	Yes	Yes	Yes	Yes	Yes
Build Agent License Pricing	Free	From $39 per month	From $299 one-off payment	From $10 one-off payment	From $4 per month per user
Supported OSs	Windows, Linux, macOS, Unix-like OS	Linux or MacOS	Windows, Linux, macOS, Solaris, FreeBSD and more	Windows, Linux, macOS, Solaris	Linux distributions: Ubuntu, Debian, CentOS, Oracle Linux

Figure: 1

Now let's see what the DeVos related CI/CD pipeline things and so far. However, if we are dealing with docker we cannot forget the docker container. Containers are also ideal for automation and DevOps pipelines, including continuous integration[1] and continuous deployment[2] (CI/CD) implementation. Therefore, this whole dockizing program is done under the CI/CD pipeline.

[1] https://www.ibm.com/cloud/learn/continuous-integration
[2] https://www.ibm.com/cloud/learn/continuous-deployment

#

Chapter 9

How to Pull Docker images on Docker Hub?

If you do not have a docker image you can pull it into your internal environment and then you can run the image.

By typing "**image ls**" command you can list down all images that already have images in your docker environment. Figure 1 is shown how to do an image list down. However, before list down docker images you should change directly to the relevant folder.

```
C:\Users\Dulanji\Desktop\Dockerfile>docker image ls
REPOSITORY                TAG       IMAGE ID       CREATED         SIZE
dulanjih/devops           latest    0de0e9ba114c   7 minutes ago   1.23GB
python-imopcv             latest    537a7db1309f   42 hours ago    1.23GB
docker/getting-started    latest    083d7564d904   5 months ago    28MB
```

Figure: 1

Using this command "docker pull [OPTIONS] NAME[:TAG|@DIGEST]" you can pull your image

Using docker image id. Here is an example.

"**docker pull ubuntu:20.04**"

Also, you can run your image using "**docker run -dp 3000:3000 dulanjih/devops**' ' **command** to run your pulled image. As figure 2 below.

```
C:\Users\Dulanji\Desktop\Dockerfile>docker run -dp 3000:3000 dulanjih/devops
7cc1cc08fcd8be7bac7e7208b0303e57ea0c4664421f5b7894bcf326a07869ce
```

Figure:2

References

1. *Menon, A. (2019). Face Detection in 2 Minutes using OpenCV & Python. [online] Medium. Available at: https://towardsdatascience.com/face-detection-in-2-minutes-using-opencv-python-90f89d7c0f81.*

2. *Ilija Mihajlovic (2019). Everything You Ever Wanted To Know About Computer Vision. Here's A Look Why It's So Awesome. [online] Medium. Available at: https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e.*

3. *Ilija Mihajlovic (2019). Everything You Ever Wanted To Know About Computer Vision. Here's A Look Why It's So Awesome. [online] Medium. Available at: https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e.*

4. *How Artificial Intelligence Revolutionized Computer Vision: A Brief History. (2019, May 16). Retrieved from Motion Metrics website: https://www.motionmetrics.com/how-artificial-intelligence-revolutionized-computer-vision-a-brief-history/*

5. *Sidath Asiri (2018). Machine Learning Classifiers. [online] Medium. Available at: https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623.*

6. *Ghoneim, S. (2019). 5 Steps to correctly prepare your data for your machine learning model. [online] Medium. Available at: https://towardsdatascience.com/5-steps-to-correctly-prep-your-data-for-your-machine-learning-model-c06c24762b73.*

7. *WhatIs.com. (n.d.). What is grayscale? - Definition from WhatIs.com. [online] Available at: https://whatis.techtarget.com/definition/grayscale [Accessed 23 Oct. 2021].*

8. *What is a CI/CD pipeline?-2021. [online] Available at: <https://www.redhat.com/en/topics/devops/what-cicd-pipeline> [Accessed 10 November 2021].*

9. *Anon, 2021. Best 14 CI/CD tools you must know: Updated for 2021. Katalon Solution. Available at: https://www.katalon.com/resources-center/blog/ci-cd-tools/ [Accessed November 15, 2021].*

10. *Anon, 2018. Advantages and disadvantages of Docker - Learn Docker. DataFlair. Available at: https://data-flair.training/blogs/advantages-and-disadvantages-of-docker/ [Accessed November 15, 2021].*
11. *Anon, 2021. Docker Overview. Docker Documentation. Available at: https://docs.docker.com/get-started/overview/ [Accessed November 15, 2021].*
12. *Anon, Containers vs. Virtual Machines (VMS): What's the difference? IBM. Available at: https://www.ibm.com/cloud/blog/containers-vs-vms [Accessed November 15, 2021].*
13. *Anon, Understanding devops. Red Hat - We make open source technologies for the enterprise. Available at: https://www.redhat.com/en/topics/devops [Accessed November 15, 2021].*
14. *By: IBM Cloud Education, What is Docker? IBM. Available at: https://www.ibm.com/cloud/learn/docker [Accessed November 15, 2021].*

The End

To...

Dear All users,

I would like to thank all the users that refer to this book. I want to highlight how to use a python program and how to dockerize that program under the CI/CD pipeline. According to my knowledge I do it successfully, also here I give basic guideline of python, docker and docker CI/CD pipeline. Actually, in this book all docker steps

are done under the guideline of docker hub document. Considering keywords that are very expanded very briefly. If you like to furthermore study you can use my reference. Because if I use any kind side all are added under my references.

Thank you.

W.M Dulanji Hansika Wijekoon.

MS21914300