Topic	Digit Recognition-2	
Class Description	**The Student builds a prediction model which takes data input from the camera and makes predictions in real time. Students learn the usage of PIL library to perform operations on images.**	
Class	**C123**	
Class time	**45 mins**	
Goal	• Build the prediction algorithm which recognizes the digits and make prediction in real time.	
Resources Required	• Teacher Resources ○ VS Code ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen • Student Resources ○ VS Code ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen	
Class structure	**Warm Up Teacher-led Activity Student-led Activity Wrap up**	**5 mins 15 min 15 min 5 min**

CONTEXT

- **Use the camera to get the input from the video.**

Class Steps	Teacher Action	Student Action

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

Step 1: **Warm Up** **(5 mins)**	Hi<student name>. How are you doing today? Can you quickly tell me what we did in the last class?	**ESR:** - In the last class we wrote a prediction algorithm to recognize the digits from the images.
	Yes.. And until now we have provided the ready made data to train and then test the models that we have. Can you tell me what kind/type of data have we used before? We have used text data, images data until now. Today we are going to use the camera of our device to provide the data to our prediction model. Want to know how?	**ESR:** varied **ESR:** Yes!
	Let's get started then.	-
Teacher Initiates Screen Share		
CHALLENGE ● **Pair program to create the digit recognition model**		
Step 2: **Teacher-led Activity** **(15 min)**	So today we are going to pair program as **Pair programming is** an important technique for developing higher quality code, faster while also reducing risk of errors. I'll be helping you to write the code and while coding we'll explore how we are going	**ESR:** Yes!!

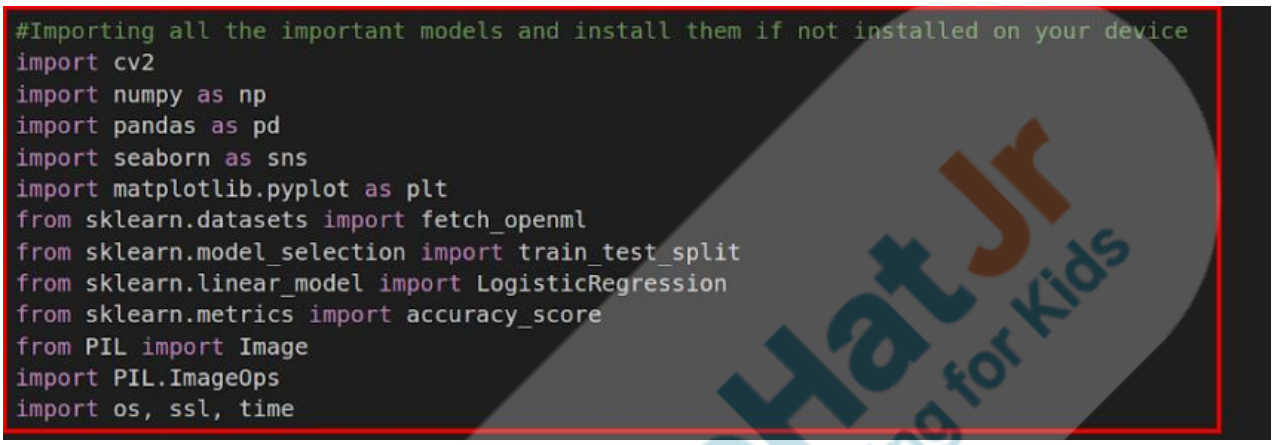
© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	to use the camera. Excited?	
	Let's start coding then without any wait.	-
Teacher Stops Screen Share		
	Now it's your turn. Please share your screen with me.	
• **Ask Student to press ESC key to come back to panel** • **Guide Student to start Screen Share** • **Teacher gets into Fullscreen**		
ACTIVITY • **Create and test the prediction model.**		
Step 3: Student-Led Activity (15 min)	Let's start by creating a virtual environment in a new directory - **python3.8 -m venv venv** Let's source the virtual environment - MACOS/UBUNTU - **source venv/bin/activate** WINDOWS - **venv\Scripts\activate.bat**	

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	Teacher helps the student to create a new folder and create virtual environment inside it and open the VS Code editor and create a file called "digit_recognition.py".	*Student opens the code editor and creates the file called digit recognition.*
	So what is the first thing that we always do while writing our code? Perfect. So let's import all the libraries that we are going to use. *Teacher helps the student to import the libraries to the code file.* ***Note:-** Check if the libraries are installed or not using pip show <library name> and install them using pip install <library name>.* *use **pip install pillow** for pil* Code:- #Importing all the important models and install them if not installed on your device **import cv2** **import numpy as np** **import pandas as pd** **import seaborn as sns** **import matplotlib.pyplot as plt** **from sklearn.datasets import fetch_openml** **from sklearn.model_selection import train_test_split** **from sklearn.linear_model import LogisticRegression**	**ESR:** We import the libraries. *The Student imports all the libraries.*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	from sklearn.metrics import accuracy_score **from PIL import Image** **import PIL.ImageOps** **import os, ssl, time**	
 ``` #Importing all the important models and install them if not installed on your device import cv2 import numpy as np import pandas as pd import seaborn as sns import matplotlib.pyplot as plt from sklearn.datasets import fetch_openml from sklearn.model_selection import train_test_split from sklearn.linear_model import LogisticRegression from sklearn.metrics import accuracy_score from PIL import Image import PIL.ImageOps import os, ssl, time ```		
	First we need some data to train our model and we are going to get it from an already available machine learning dataset. Let's do that! **Code:** **#Fetching the data** **X,y = fetch_openml('mnist_784', version=1, return_X_y=True)** **print(pd.Series(y).value_counts())** **classes = ['0', '1', '2' , '3', '4', '5', '6' , '7' ,'8' ,'9']** **nclasses = len(classes)**	*Student codes to use fetch to get the data from the machine learning dataset.*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

```
#Fetching the data
X, y = fetch_openml('mnist_784', version=1, return_X_y=True)
print(pd.Series(y).value_counts())
classes = ['0', '1', '2','3', '4','5', '6', '7', '8', '9']
nclasses = len(classes)
```

	Let's try to run this code and see if we are getting the right output?	

Note:- You might get the error

In any circumstances you can explain the concept of SSL as it will help students to recognize if the site is secured or not.

	Oh no! We got an error. It looks like it has something to do with SSL. Do you know what it is? This same code worked when we did it on Google Colab in the last class? What could have gone wrong this time?	**ESR:** Varied

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	Google Colab is available on the browser. When we go to any link, many of the links start with an **https://** instead of an **http://** The **s** in the **https://** means that we are trying to establish a secure connection. Browser provides us with SSL when we try to open a link, but here since we are running a python script locally on our machine, the **openML** thinks that our python script cannot be trusted since there is no SSL, which a browser automatically provides. To encounter this, if we look at the code above, we imported a library known as **ssl.** We can use it with the following code - Code: **#Setting an HTTPS Context to fetch data from OpenML** **if (not os.environ.get('PYTHONHTTPSVERIFY', '') and** **getattr(ssl, '_create_unverified_context', None)):** **ssl._create_default_https_context = ssl._create_unverified_context** **#Fetching the data** **X, y = fetch_openml('mnist_784', version=1, return_X_y=True)** **print(pd.Series(y).value_counts())**	

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	classes = ['0', '1', '2','3', '4','5', '6', '7', '8', '9'] **nclasses = len(classes)**	

```
#Setting an HTTPS Context to fetch data from OpenML
if (not os.environ.get('PYTHONHTTPSVERIFY', '') and
    getattr(ssl, '_create_unverified_context', None)):
    ssl._create_default_https_context = ssl._create_unverified_context

#Fetching the data
X, y = fetch_openml('mnist_784', version=1, return_X_y=True)
print(pd.Series(y).value_counts())
classes = ['0', '1', '2','3', '4','5', '6', '7', '8', '9']
nclasses = len(classes)
```

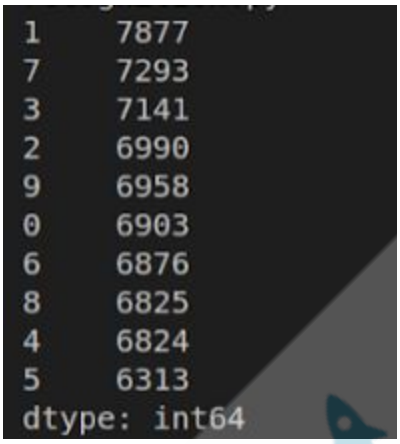
Now, let's try to understand this.

Here, we first have an if condition where we are checking if our python environment's **"PYTHONHTTPSVERIFY"** is an empty string. This means that our Python script is not HTTPS Verified.

Next, we are seeing if our **ssl**'s unverified context is created or not. Remember, a browser provides the ssl context by default.

Now, if our script does not have https verification and if it also does not have an ssl's unverified context, then we are creating a default https unverified context for our python script using the SSL module.

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	We will learn more about SSL and different protocols when we deep dive into **networking**.	
	Now when we try to run our code, let's see if the error is resolved?	
		
	Perfect! Now we have the data. What should be our next step? Yes so let's split the data to train and test the model and also scale it to make sure that the data has equal number of points. *<Teacher helps the student with the code>* Code: **#Splitting the data and scaling it** **X_train, X_test, y_train, y_test = train_test_split(X, y,**	**ESR:** Split the data to train and test the model. *Student codes to split the data and scale it.*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

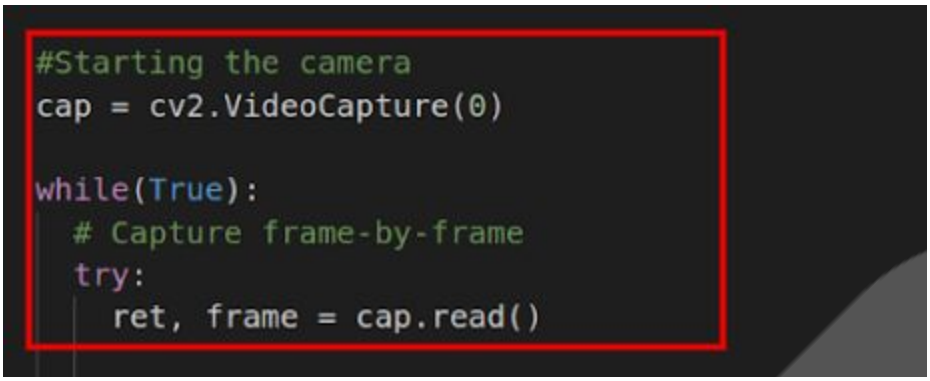
	random_state=9, train_size=7500, test_size=2500) **#scaling the features** **X_train_scaled = X_train/255.0** **X_test_scaled = X_test/255.0**	
 `#Splitting the data and scaling it` `X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=9, train_size=7500, test_size=2500)` `#scaling the features` `X_train_scaled = X_train/255.0` `X_test_scaled = X_test/255.0`		
	Now we need to fit the data into our logistics regression model so that it can predict the accuracy with maximum efficiency. And make a prediction. *Teacher helps the student with the code.* Code: **#Fitting the training data into the model** **clf = LogisticRegression(solver='saga', multi_class='multinomial').fit(X_train_scaled, y_train)** **#Calculating the accuracy of the model** **y_pred = clf.predict(X_test_scaled)** **accuracy = accuracy_score(y_test, y_pred)** **print(accuracy)**	*Student codes to fit the data into the model and make a prediction.*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

```
#Fitting the training data into the model
clf = LogisticRegression(solver='saga', multi_class='multinomial').fit(X_train_scaled, y_train)

#Calculating the accuracy of the model
y_pred = clf.predict(X_test_scaled)
accuracy = accuracy_score(y_test, y_pred)
print("The accuracy is :- ",accuracy)
```

	Alright now the fun part begins. Now we'll use the camera to give input to our prediction model to make the predictions. So we'll start the camera and then capture every frame . Here we'll use the try except block as try block lets us test the code for errors and except block lets us handle the errors. The try except block has 2 blocks. 1st **try** where you write the code which you want to execute. 2nd **except** block which catches the errors from the try block . *Teacher helps the student with the code.* Code: **#Starting the camera** **cap = cv2.VideoCapture(0)** **while(True):** **# Capture frame-by-frame** **try:** **ret, frame = cap.read()**	*Student codes to start the camera and capture every frame.*

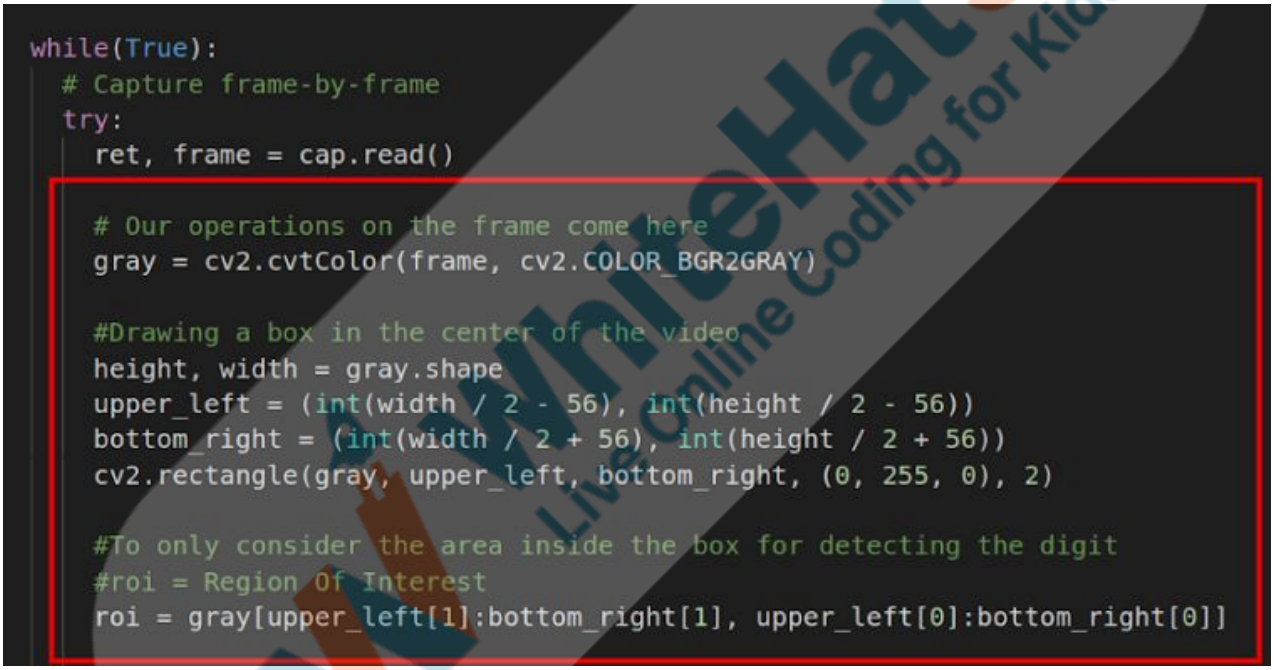
© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.



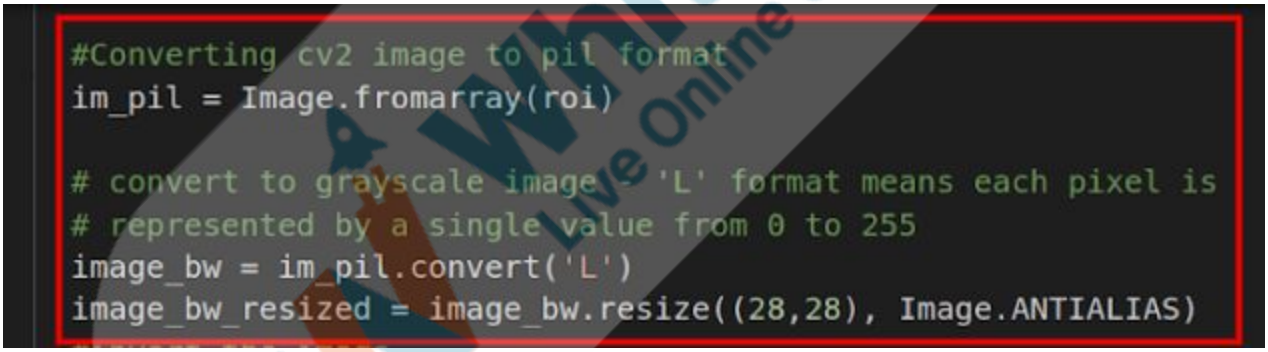
```
#Starting the camera
cap = cv2.VideoCapture(0)

while(True):
  # Capture frame-by-frame
  try:
    ret, frame = cap.read()
```

	As we are going to be giving our model direct input from a video we don't want it to get confused due to all the colors so we'll set the color of the video to gray and also draw a rectangle in the center of the video. This rectangle will be the region of interest. This rectangle will serve the purpose as it will be the only place area where the model will detect the digit. *Teacher helps the student with the code.* Code: **#Drawing a box in the center of the video** **height, width = gray.shape** **shape- It returns a tuple of number of rows, columns and channels** **upper_left = (int(width / 2 - 56), int(height / 2 - 56))** **bottom_right = (int(width / 2 + 56), int(height / 2 + 56))** **cv2.rectangle(gray, upper_left,**	*Student codes to make the color of the video to gray and create a rectangle which would be the region of interest.*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	bottom_right, (0, 255, 0), 2) rectangle function creates a rectangle **#To only consider the area inside the box for detecting the digit** **#roi = Region Of Interest** **roi = gray[upper_left[1]:bottom_right[1], upper_left[0]:bottom_right[0]]**	
 ``` while(True): # Capture frame-by-frame try: ret, frame = cap.read() # Our operations on the frame come here gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) #Drawing a box in the center of the video height, width = gray.shape upper_left = (int(width / 2 - 56), int(height / 2 - 56)) bottom_right = (int(width / 2 + 56), int(height / 2 + 56)) cv2.rectangle(gray, upper_left, bottom_right, (0, 255, 0), 2) #To only consider the area inside the box for detecting the digit #roi = Region Of Interest roi = gray[upper_left[1]:bottom_right[1], upper_left[0]:bottom_right[0]] ```		
	The images that we are getting from the camera are cv2 images and we need to convert them to pil format to use them. We'll also represent the pil format image with a value between 0 and 255 and resize it to 28 by 28 size. Code:	*Student codes to convert the images to pil format and represent it with a value between 0 and 255 and resize it to 28 by 28 size.*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	#Converting cv2 image to pil format **im_pil = Image.fromarray(roi)** **Image.fromarray()** Creates an image memory from an object exporting the array interface **# convert to grayscale image - 'L' format means each pixel is** **# represented by a single value from 0 to 255** **image_bw = im_pil.convert('L')** convert() converts pixels to gray scale. **image_bw_resized = image_bw.resize((28,28), Image.ANTIALIAS)** resize() resizes the image.	
	 `#Converting cv2 image to pil format` `im_pil = Image.fromarray(roi)` `# convert to grayscale image - 'L' format means each pixel is` `# represented by a single value from 0 to 255` `image_bw = im_pil.convert('L')` `image_bw_resized = image_bw.resize((28,28), Image.ANTIALIAS)`	
	There is still a small issue and that is we can see the images inverted in our camera. And by inverted I mean it looks like a mirror image. What can we do to fix this issue? Perfect!. we'll invert the image. After inverting the image we also have to make it scalar to get the minimum pixel and limit its value	**ESR:** We can invert the image again.

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	between 0 and 255 and then getting the maximum pixel of the image. Code: **#invert the image** **image_bw_resized_inverted = PIL.ImageOps.invert(image_bw_resized)** **pixel_filter = 20** **#converting to scalar quantity** **min_pixel = np.percentile(image_bw_resized_inverted, pixel_filter)** percentile function converts the values in scalar quantity **#using clip to limit the values between 0,255** **image_bw_resized_inverted_scaled = np.clip(image_bw_resized_inverted-min_pixel, 0, 255)** clip is used to limit the values **max_pixel = np.max(image_bw_resized_inverted)** max function get the maximum of the given numbers	*Student codes to invert the image and then convert it into a scalar quantity to get the minimum pixel and the maximum pixel.*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

```
#invert the image
image_bw_resized_inverted = PIL.ImageOps.invert(image_bw_resized)
pixel_filter = 20
#converting to scalar quantity
min_pixel = np.percentile(image_bw_resized_inverted, pixel_filter)
#using clip to limit the values between 0,255
image_bw_resized_inverted_scaled = np.clip(image_bw_resized_inverted-min_pixel, 0, 255)
max_pixel = np.max(image_bw_resized_inverted)
```

	Now we just need to change this data in an array so that it can be used in our model to make a prediction. And then make a prediction. *Teacher helps the student with the code.* Code: **#converting into an array** **image_bw_resized_inverted_scaled = np.asarray(image_bw_resized_inverted_scaled)/max_pixel** np.asarray() converts given values into array **#creating a test sample and making a prediction** **test_sample = np.array(image_bw_resized_inverted_scaled).reshape(1,784)** **test_pred = clf.predict(test_sample)** **print("Predicted class is: ", test_pred)**	*Student codes to convert the data into an array and make a prediction.*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

```
#converting into an array
image_bw_resized_inverted_scaled = np.asarray(image_bw_resized_inverted_scaled)/max_pixel
#creating a test sample and making a prediction
test_sample = np.array(image_bw_resized_inverted_scaled).reshape(1,784)
test_pred = clf.predict(test_sample)
print("Predicted class is: ", test_pred)
```

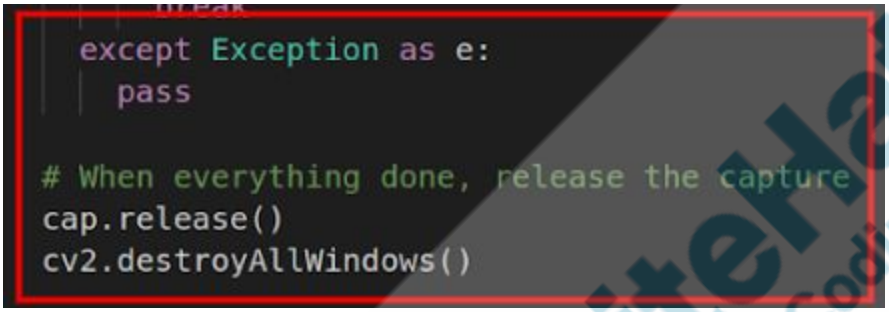
	Finally, we want to show the resulting frame. And we'll also add a key control to turn off the camera. We'll use either the "esc" or "q" key to close or stop the code from running. *Teacher helps the student with the code.* Code: **# Display the resulting frame** **cv2.imshow('frame',gray)** **if cv2.waitKey(1) & 0xFF == ord('q'):** **break**	*Student codes to display the resulting frame and adds key control to break the code from running.*

```
# Display the resulting frame
cv2.imshow('frame',gray)
if cv2.waitKey(1) & 0xFF == ord('q'):
  break
```

	So this is where the code in the try block ends and in the except block, we don't want to deal with any errors so we'll write a pass condition to ignore the errors. And when everything is done close the camera and close all the windows.	*Student codes to pass the errors in the except block. And close the camera and close all the windows when everything is done.*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	Teacher helps the student with the code. Code: **except Exception as e:** **pass** **# When everything done, release the capture** **cap.release()** **cv2.destroyAllWindows()**	



Teacher Guides Student to Stop Screen Share

FEEDBACK

- **Appreciate the student for their efforts**
- **Identify 2 strengths and 1 area of progress for the student**

Step 4: **Wrap-Up** **(5 min)**	So I hope you liked and had fun in today's class. Can you quickly tell me what we learned today.	**ESR:** varied *Student speaks about his/her learnings from today's class.*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	We'll keep on having fun and learning new stuff in our upcoming class. Did you enjoy it?	**ESR:** Varied
	Looking forward to our next class	-
Teacher Clicks 		
Additional Activities	*Encourage the student to write reflection notes in their reflection journal using markdown.* Use these as guiding questions: ● What happened today? - Describe what happened - Code I wrote ● How did I feel after the class? ● What have I learned about programming and developing games? ● What aspects of the class helped me? What did I find difficult?	*The student uses the markdown editor to write her/his reflection in a reflection journal.*

Activity	Activity Name	Links
Teacher Activity 1	Solution	https://github.com/whitehatjr/digital-recoginition-2/blob/master/digit_recognition.py

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.