

[← Go Back to Introduction to Computer Vision](#)

☰ Course Content

FAQ - Introduction to Computer Vision

1. What is Computer Vision?

Computer Vision is a field of Artificial Intelligence that enables computers or systems to derive or extract meaningful information from a digital image or video data and take actions or recommendations based on this information.

2. What is the Normalization of pixels?

For most image data, the pixel values are integers with values between 0 and 255. Normalization scales these pixel values between 0 and 1 before modeling.

3. Why should Normalization be performed on pixels?

Neural networks process inputs using small weight values and inputs with large integer values can disrupt or slow down the learning process, so it is a good practice to normalize the pixels so that the values would range between 0 and 1.

4. How is smoothing performed on images?

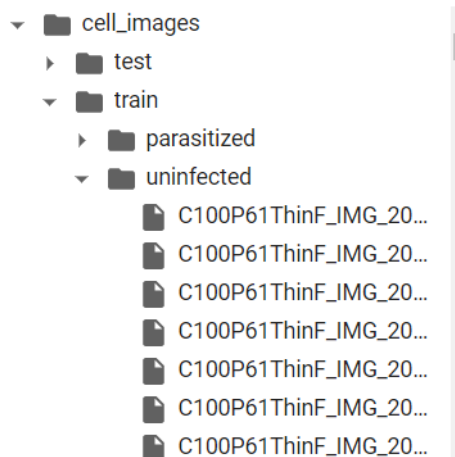
Smoothing an image is required to reduce noise and blur the false edges. Smoothing is usually performed using Gaussian Blur. Gaussian filtering is highly effective at removing noise from the image. We can use the below code to perform Gaussian Blur.

```
blur_image = cv2.GaussianBlur(original_image, (5,5), 0)
```

5. Will the images always be provided in zip file format?

The image datasets are usually provided in the following formats:

- **Folders/Directories:** The whole dataset can be given in a single directory or a zip file which can be extracted to get a directory. This directory can contain sub-folders like training and testing. The training and testing folders may contain individual sub-folders for each category of image. The whole structure can be observed in the below image.



- The dataset can be loaded from the **TensorFlow datasets** as shown in the below image -

```
# Loading the data
from tensorflow.keras.datasets import mnist
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11493376/11490434 [=====] - 0s 0us/step
11501568/11490434 [=====] - 0s 0us/step
```

- The images can also be given as **numpy arrays** and the labels to the corresponding images can be given in a CSV file. The images that are converted into numpy arrays can be given in images.npy file and the labels corresponding to each image can be given in the labels.csv file. The np.load() function can be used to read the numpy arrays and the pd.read_csv() function from pandas can be used to read the csv file as shown in the below image -

```
# Load the image file of dataset
images = np.load('/content/drive/MyDrive/Computer_vision_online_template/Plants-seedling-modified/images.npy')

# Load the labels file of dataset
labels = pd.read_csv('/content/drive/MyDrive/Computer_vision_online_template/Plants-seedling-modified/labels.csv')
```

6. Do we need to split the image data into train and test sets before pre-processing?

The splitting of image data should be performed before any operations or pre-processing steps on images.

One way is to divide the data set into two subsets:

• Training set: a subset to train a model.

• Test set: a subset to test the model.

Separating the data enables you to evaluate your model generalization capabilities and have an idea of how it would perform on unseen data. Good performance on the test set is a useful indicator of good performance on new data in general, assuming that:

• The samples were drawn independently and at random from the distribution to create the test set.

• The test set is large enough.

The second way is to divide the data set into three subsets:

• Training set: a subset to train a model.

• Validation set: a subset to validate and tune our model.

• Test set: a subset to test the model.

7. What is the difference between MNIST and Fashion MNIST datasets that are used in the course?

The MNIST data consists of images of 10 handwritten digits from 0 to 9, and this dataset can be directly loaded using the load_data() function of Tensorflow. Whereas Fashion MNIST can also be loaded using the same load_data() function, it consists of fashion images like shoes, shirts, etc belonging to 10 different categories.

8. What is the difference between resize() and reshape() function(in context of images)?

The reshape() function changes the shape only and not the number of pixels. For example, the image of size 6 x 4 can be reshaped into 12 x 2. Here we did not change the number of pixels. But an image of 6 x 6 can be resized into 10 x 10 using the resize function. Here the number of pixels are increased.

9. Which function can be used to get the names of labels that are encoded using an encoder?

The inverse_transform() function can be used to decode the labels from the encoded vectors.
For example: if a label 'car' is encoded using a label binarizer into an array of [0,0,1] then inverse_transform can be used to get the label name from this array.

10. How to unzip a zipfile in Google Colab?

unzip command can be used to unzip the file in google colab

```
!unzip "path to the file"
```