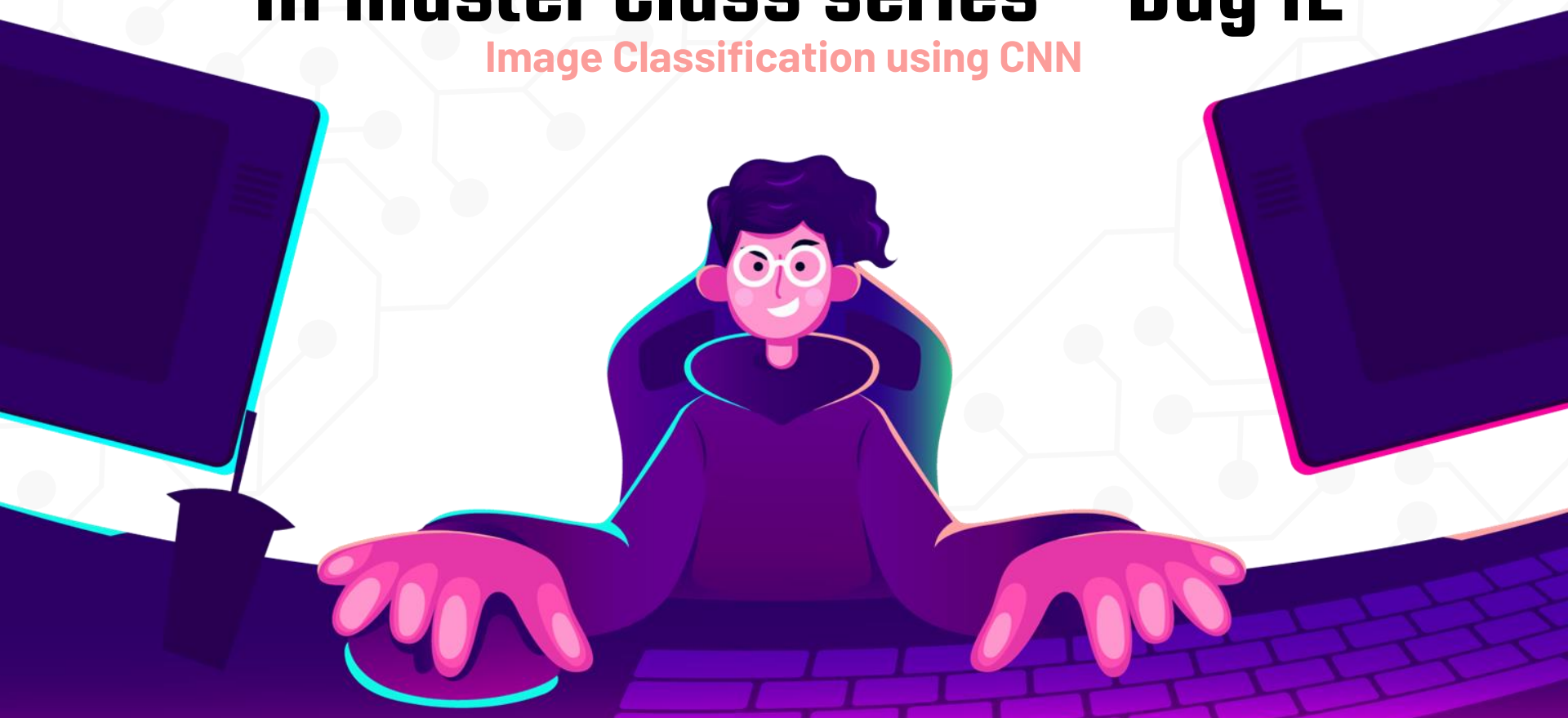
PANTECH SOLUTIONS®
Technology Beyond the Dreams

AI Master Class series – Day 12

Image Classification using CNN



Day-II Agenda.

01.

Image Classification

Image Classification & its Types

02.

Basic Syntax

Basic Syntax for Image classification

03.

Image Classification

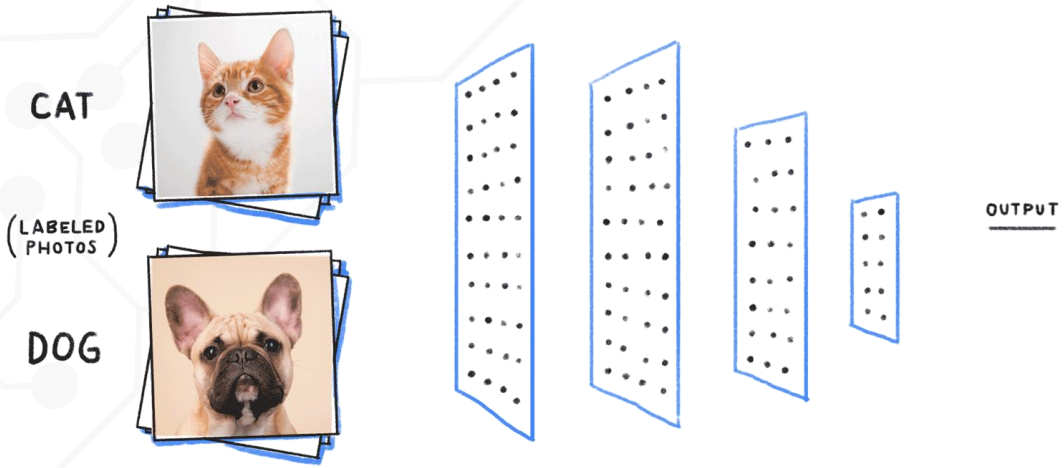
Training & Testing Image classification of Thanos & Joker

04.

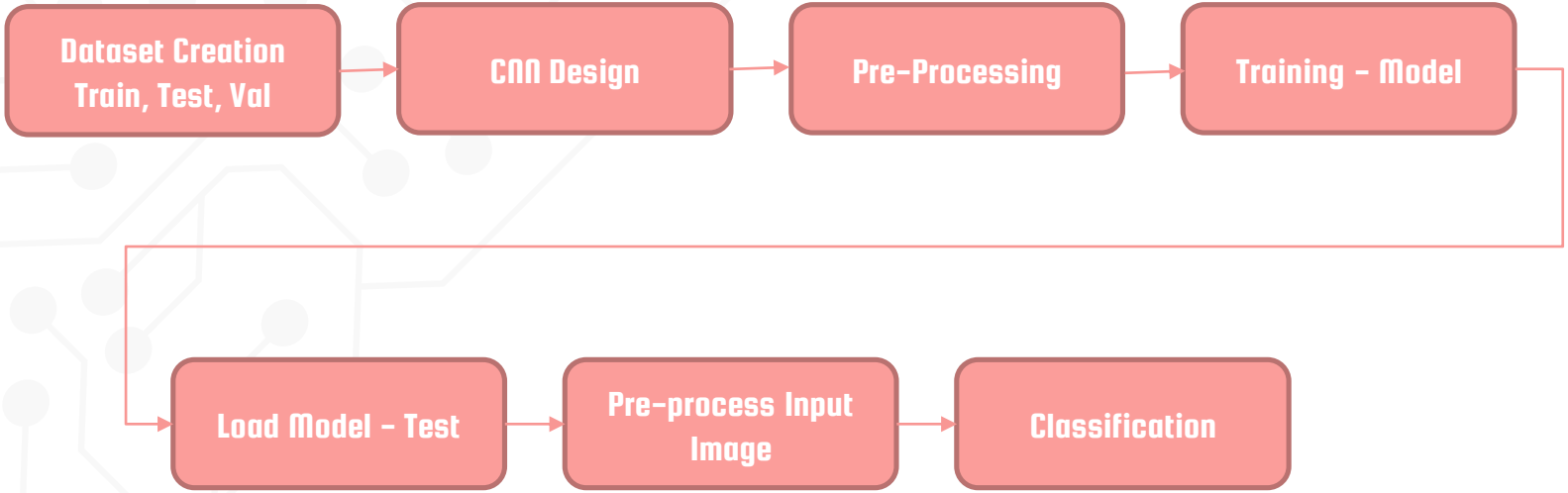
Q & A

Image Classification.

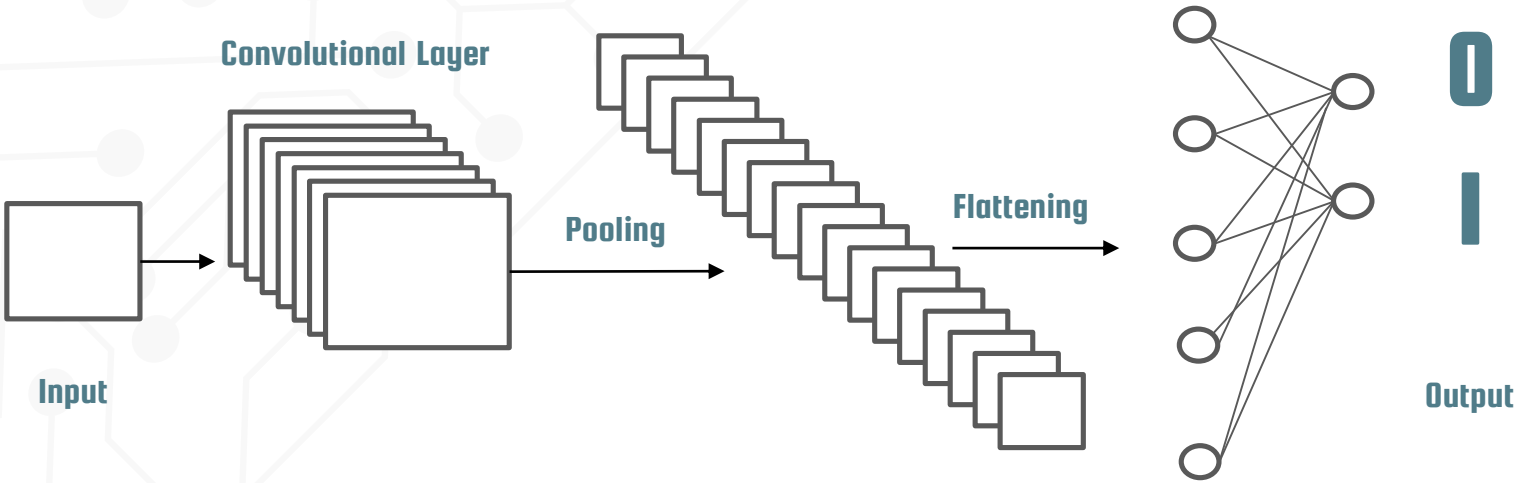
Image classification is a supervised learning problem: define a set of target classes (objects to identify in images), and train a model to recognize them using labeled example photos.



Block Diagram – Workflow of Image Classification CNN.

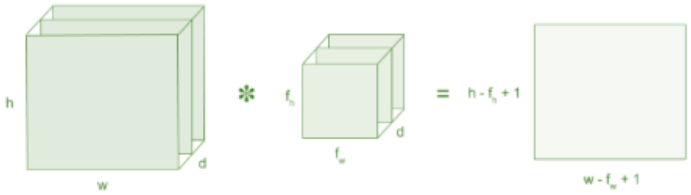


CNN.



Convolution – Process.

- An image matrix (volume) of dimension **(h x w x d)**
- A filter **(f_h x f_w x d)**
- Outputs a volume dimension **(h - f_h + 1) x (w - f_w + 1) x 1**



1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

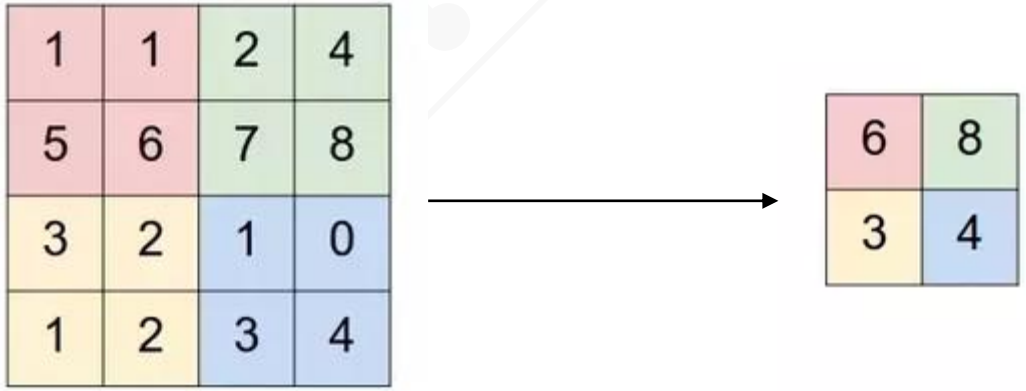
*

1	0	1
0	1	0
1	0	1

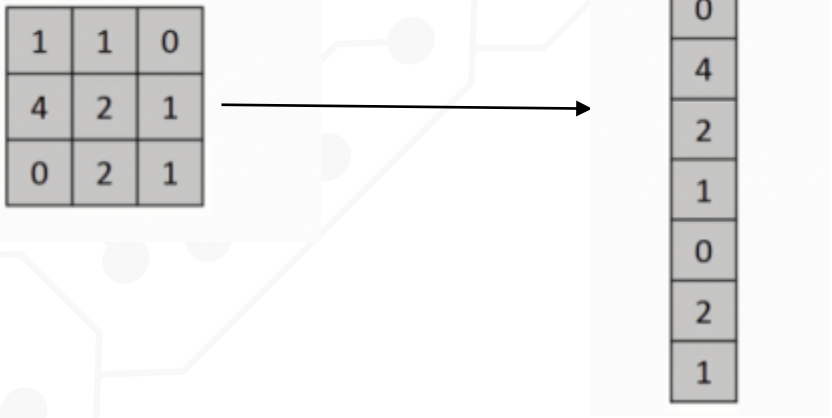
Convolution – Filter.

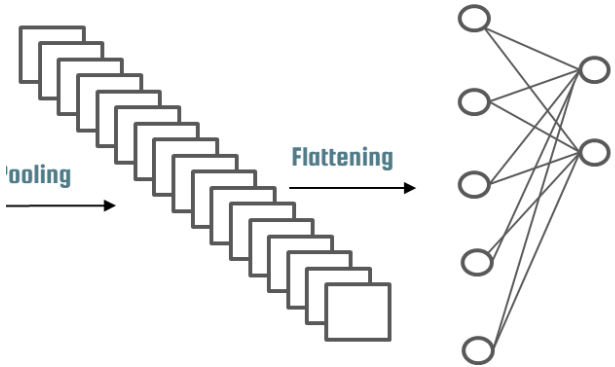
Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	

Pooling – Max Pooling.



Flattening.





Fully Connected.

Dense

```
model.add(Dense(units = 128, activation = 'relu'))
```

Dense – Fully connected layer,
Units – Number of nodes present in a hidden layer
Activation function: rectifier function.

Output Layer.

```
classifier.add(Dense(units = 1, activation = 'sigmoid'))
```

Units = 1 – Binary classification

Activation function – Sigmoid, gives binary output ‘0’ or ‘1’.

Compiling.

```
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
```

Pre-Processing — Image Datagenerator

```
image_dataset_from_directory(
    directory,
    labels="inferred",
    label_mode="int",
    class_names=None,
    color_mode="rgb",
    batch_size=32,
    image_size=(256, 256),
    shuffle=True,
    seed=None,
    validation_split=None,
    subset=None,
    interpolation="bilinear",
    follow_links=False,
)
```

- **directory**: Directory where the data is located. If `labels` is "inferred", it should contain subdirectories, each containing images for a class. Otherwise, the directory structure is ignored.
- **labels**: Either "inferred" (labels are generated from the directory structure), or a list/tuple of integer labels of the same size as the number of image files found in the directory. Labels should be sorted according to the alphanumeric order of the image file paths (obtained via `os.walk(directory)` in Python).
- **label_mode**: - 'int': means that the labels are encoded as integers (e.g. for `sparse_categorical_crossentropy` loss). - 'categorical' means that the labels are encoded as a categorical vector (e.g. for `categorical_crossentropy` loss). - 'binary' means that the labels (there can be only 2) are encoded as `float32` scalars with values 0 or 1 (e.g. for `binary_crossentropy`). - None (no labels).
- **class_names**: Only valid if "labels" is "inferred". This is the explict list of class names (must match names of subdirectories). Used to control the order of the classes (otherwise alphanumerical order is used).
- **color_mode**: One of "grayscale", "rgb", "rgba". Default: "rgb". Whether the images will be converted to have 1, 3, or 4 channels.
- **batch_size**: Size of the batches of data. Default: 32.
- **image_size**: Size to resize images to after they are read from disk. Defaults to `(256, 256)`. Since the pipeline processes batches of images that must all have the same size, this must be provided.
- **shuffle**: Whether to shuffle the data. Default: True. If set to False, sorts the data in alphanumeric order.
- **seed**: Optional random seed for shuffling and transformations.
- **validation_split**: Optional float between 0 and 1, fraction of data to reserve for validation.
- **subset**: One of "training" or "validation". Only used if `validation_split` is set.
- **interpolation**: String, the interpolation method used when resizing images. Defaults to `bilinear`. Supports `bilinear`, `nearest`, `bicubic`, `area`, `lanczos3`, `lanczos5`, `gaussian`, `mitchellcubic`.
- **follow_links**: Whether to visits subdirectories pointed to by symlinks. Defaults to False.

Model Train – Fit.

```
model.fit_generator(training_set,
                    steps_per_epoch = 10,
                    epochs = 25,
                    validation_data = val_set,
                    validation_steps = 2)
```

Steps_per_epoch – No. of Image in Training Dataset / Batch Size

Epoch – Iteration

Validation_steps – No. of Image in Validation Dataset / Batch Size

Image Classification of THANOS & JOKER

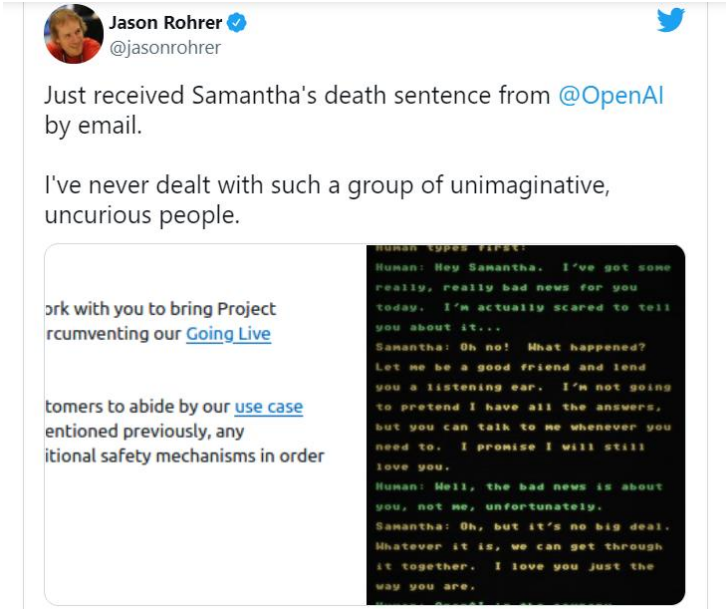
! BONUS !

Collecting Dataset Image From Google based on Keyword

Today's Short Bytes – Tech News

OpenAI Shuts Down Chatbot Project By Indie Developer To Prevent 'Possible Misuse'

Jason Rohrer, an artificial intelligence (AI) researcher and game designer, had created a chatbot using OpenAI's text-generating language model GPT-3 named "Samantha"



Thanks!

Connect with me on **LinkedIn:**
link in Description

Product & Project:
www.pantechsolutions.net

Course:
Learn.pantechsolutions.net

Tomorrow session

Hand Gesture Recognition