DS JULY 2022 Batch
Module 21 : Deep learning part 3

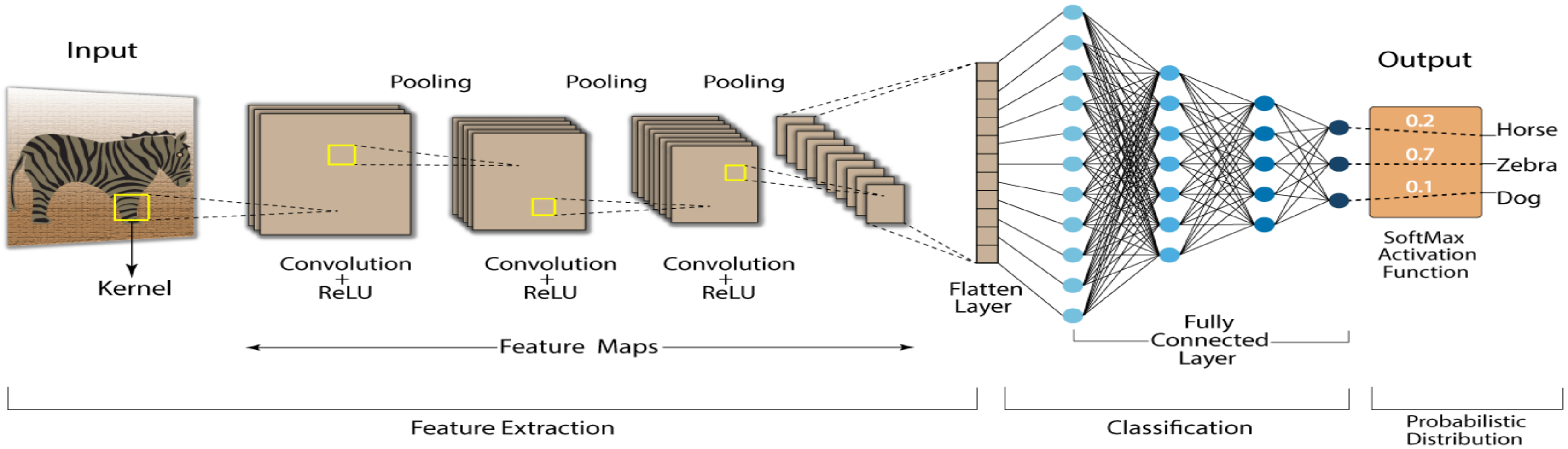
Topics

- Convolutional Neural Networks (CNN)
- ImageNet Dataset
- Project: Image Classification
- Different types of CNN architectures
- Using pre-trained model: Transfer Learning

Convolution Neural Network

CNN

- **Convolutional Neural Networks** also known as CNNs or ConvNets, are a type of feed-forward artificial neural network whose connectivity structure is inspired by the organization of the animal visual cortex.
- Small clusters of cells in the visual cortex are sensitive to certain areas of the visual field. Individual neuronal cells in the brain respond or fire only when certain orientations of edges are present. Some neurons activate when shown vertical edges, while others fire when shown horizontal or diagonal edges.
- These networks can handle a wide range of tasks involving images, sounds, texts, videos, and other media. **Professor Yann LeCunn** of Bell Labs created the first successful convolution networks in the late 1990s.



CNN

- **A Classic CNN:**

- Contents of a classic Convolutional Neural Network: -

  - 1.Convolutional Layer.

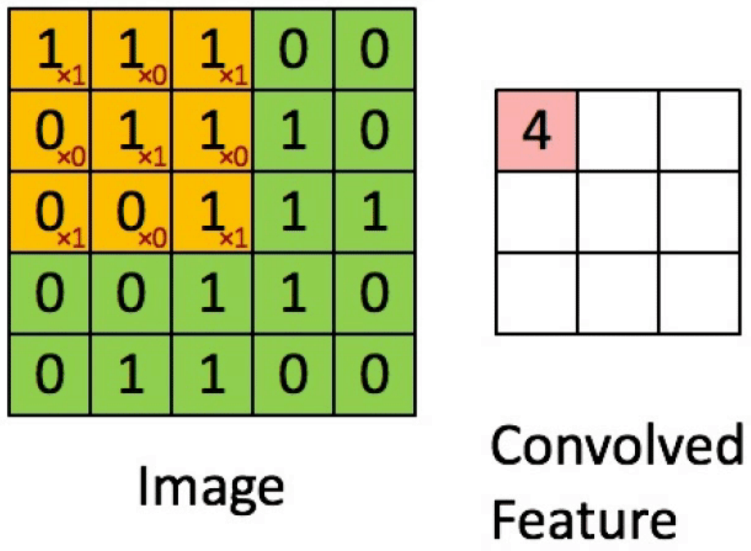
  - 2.Activation operation following each convolutional layer.

  - 3.Pooling layer especially Max Pooling layer and also others based on the requirement.

  - 4.Finally Fully Connected Layer.

CNN : Convolution layer

- **Convolutional Layer**
- In convolutional neural networks, the major building elements are convolutional layers. This layer often contains input vectors, such as an image, filters, such as a feature detector, and output vectors, such as a feature map. The image is abstracted to a feature map, also known as an activation map, after passing through a convolutional layer.
- ***Feature Map = Input Image x Feature Detector***
- A convolution is a grouping function in mathematics. Convolution occurs in CNNs when two matrices (rectangular arrays of numbers arranged in columns and rows) are combined to generate a third matrix.
- In the convolutional layers of a CNN, these convolutions are used to filter input data and find information.

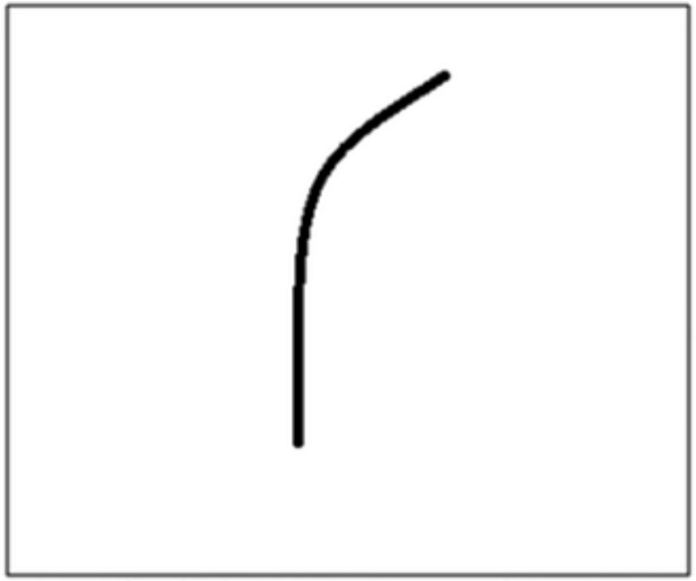


CNN : Convolution layer

- The CONVOLUTIONAL LAYER is related to feature extraction.

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter



Visualization of a curve detector filter

CNN : Convolution layer

- The operation involves multiplying the values of a cell corresponding to a particular row and column, of the image matrix, with the value of the corresponding cell in the filter matrix. We do this for the values of all the cells within the span of the filter matrix and add them together to form an output.

Visualization of the filter on the image

0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

Pixel representation of receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

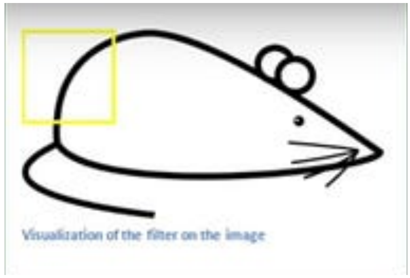
Pixel representation of filter

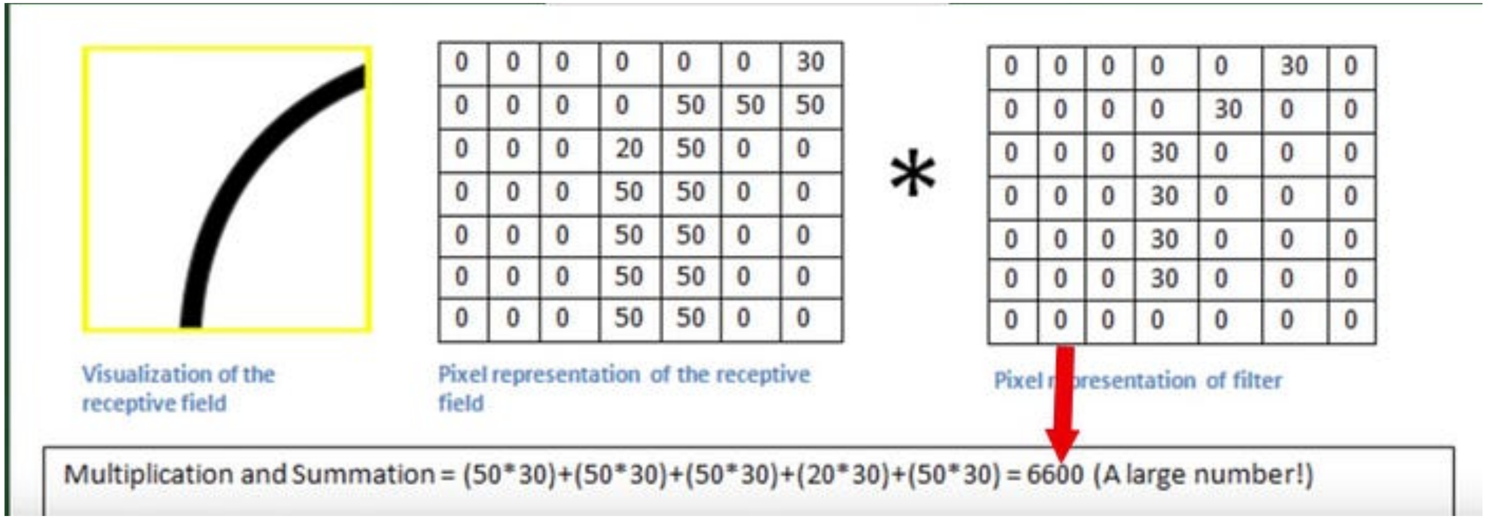
Multiplication and Summation = 0

- An example with result 0, thus confirming absence of the feature.

CNN : Convolution layer

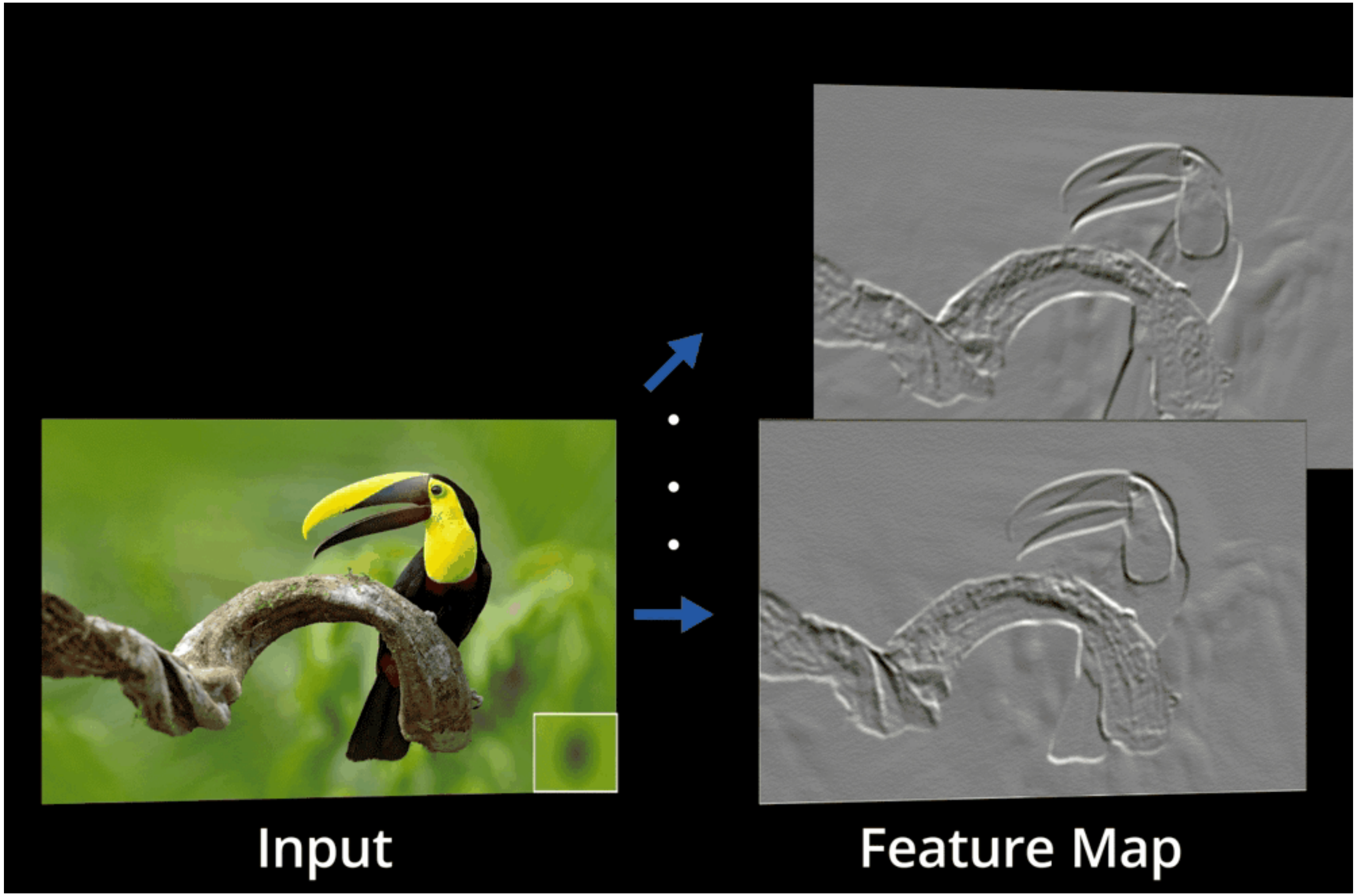
- the basic intuition to be taken from here, is that, if the feature is present in a particular part of the image, it returns a very large value on convolution, while for the other places, it would return a small value, depicting that the feature is not present.





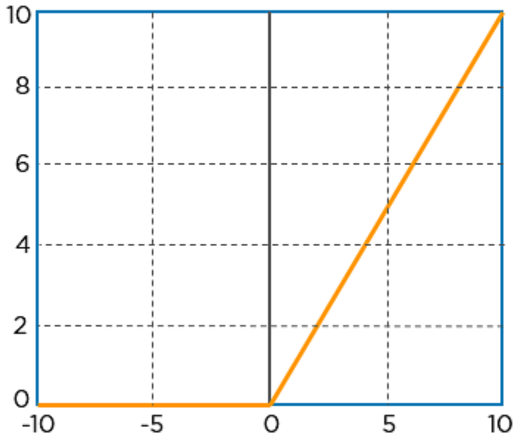
- An example with a large value as result, thus confirming the presence of the feature

CNN : Convolution



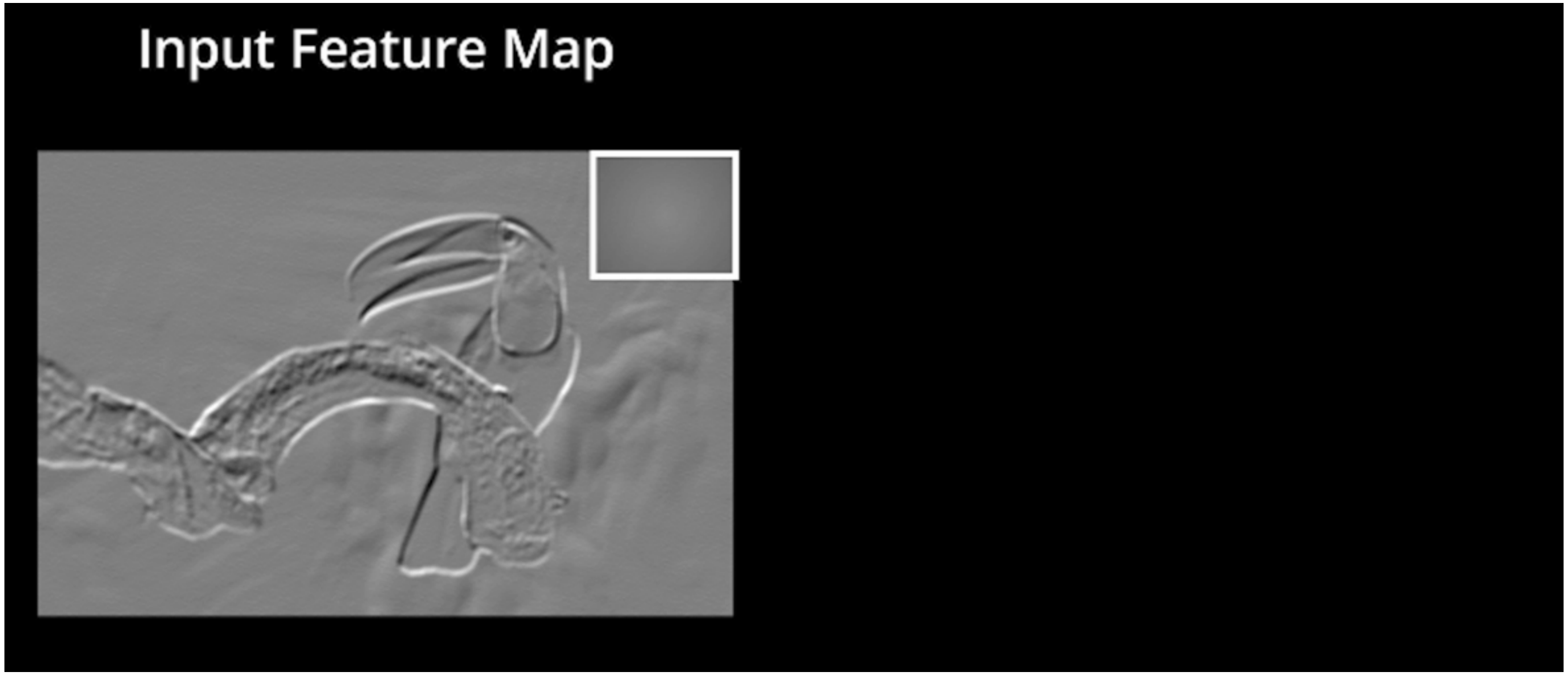
CNN : Activation

- **ReLU layer**
- ReLU stands for the rectified linear unit. Once the feature maps are extracted, the next step is to move them to a ReLU layer. ReLU performs an element-wise operation and sets all the negative pixels to 0. It introduces non-linearity to the network, and the generated output is a rectified feature map. Below is the graph of a ReLU function:



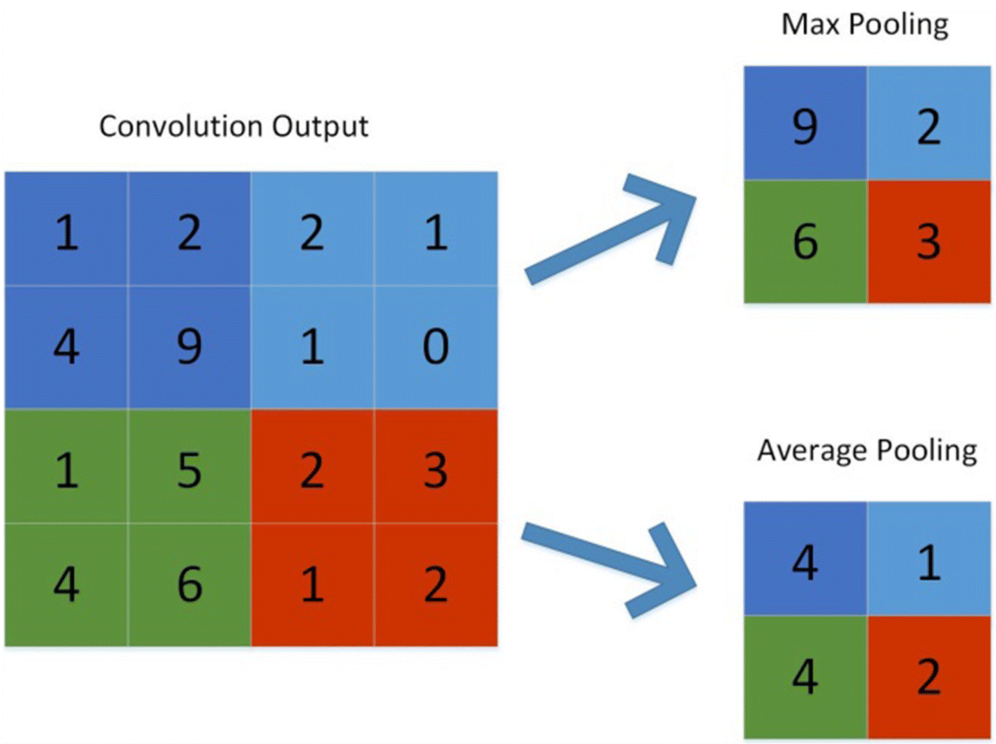
- The original image is scanned with multiple convolutions and ReLU layers for locating the features.

CNN : Activation



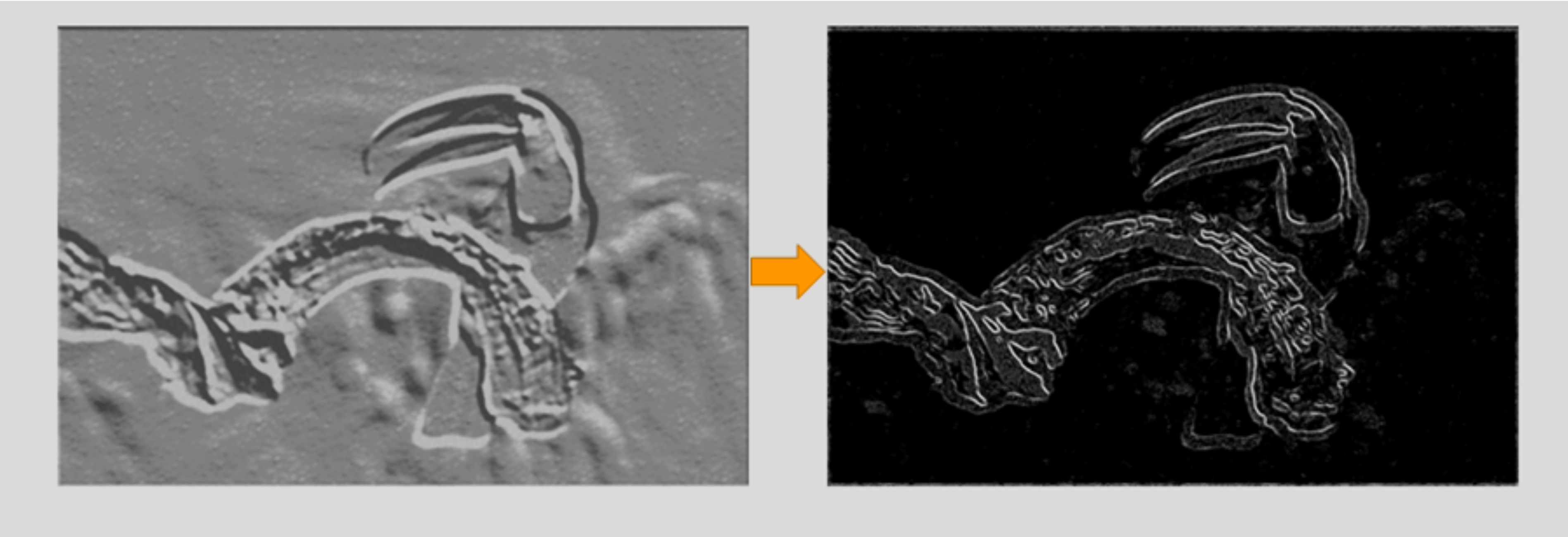
CNN : Pooling layer

- Pooling is a down-sampling operation that reduces the dimensionality of the feature map. The rectified feature map now goes through a pooling layer to generate a pooled feature map.
- The following are some methods for pooling:
  - **Max-pooling**: It chooses the most significant element from the feature map. The feature map's significant features are stored in the resulting max-pooled layer. It is the most popular method since it produces the best outcomes.
  - **Average pooling**: It entails calculating the average for each region of the feature map



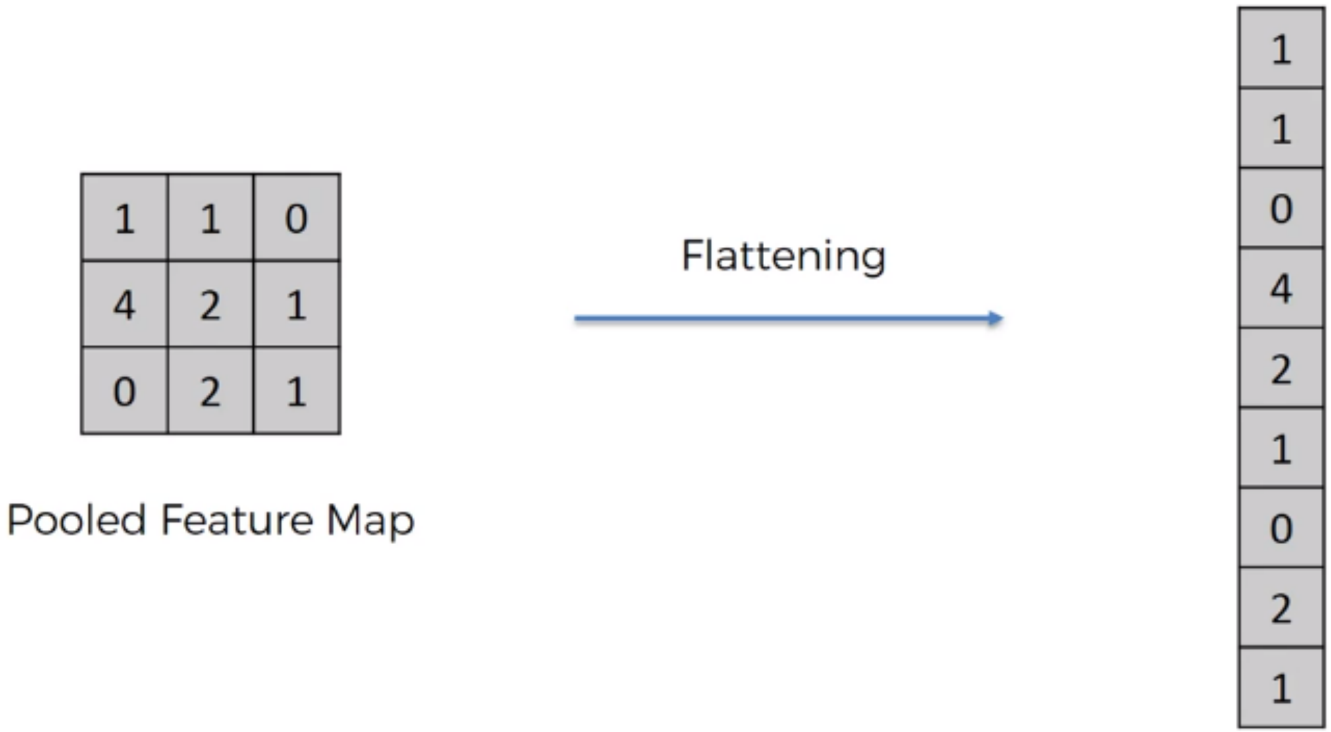
CNN : Pooling layer

- The pooling layer uses various filters to identify different parts of the image like edges, corners, body, feathers, eyes, and beak.



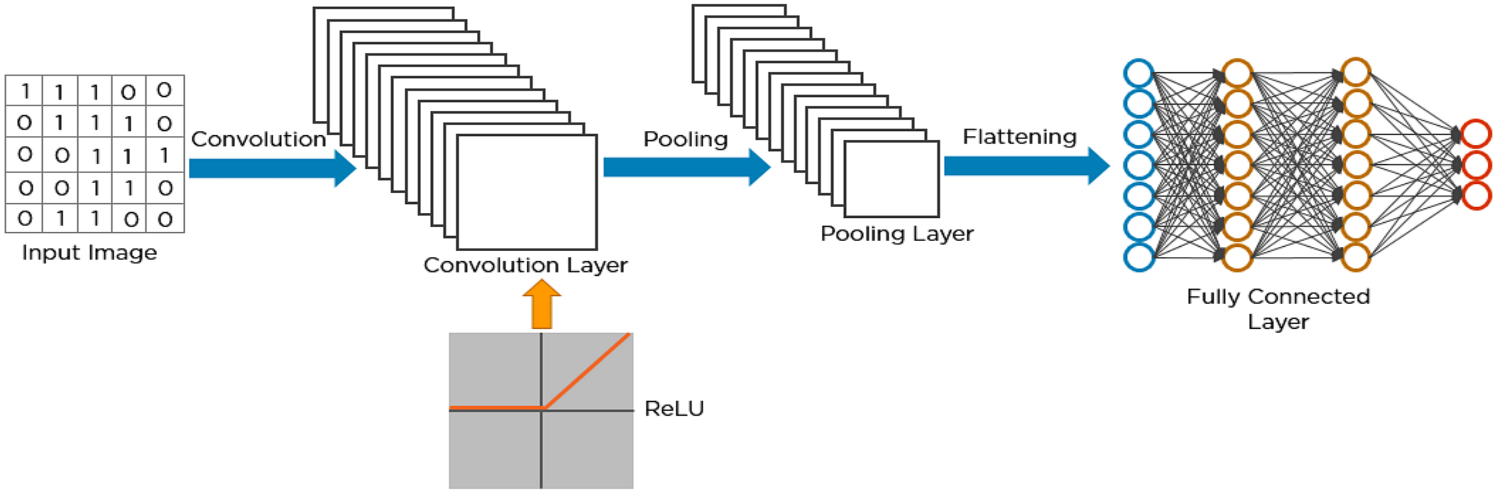
CNN : Flattening

- Flattening is used to convert all the resultant 2-Dimensional arrays from pooled feature maps into a single long continuous linear vector.

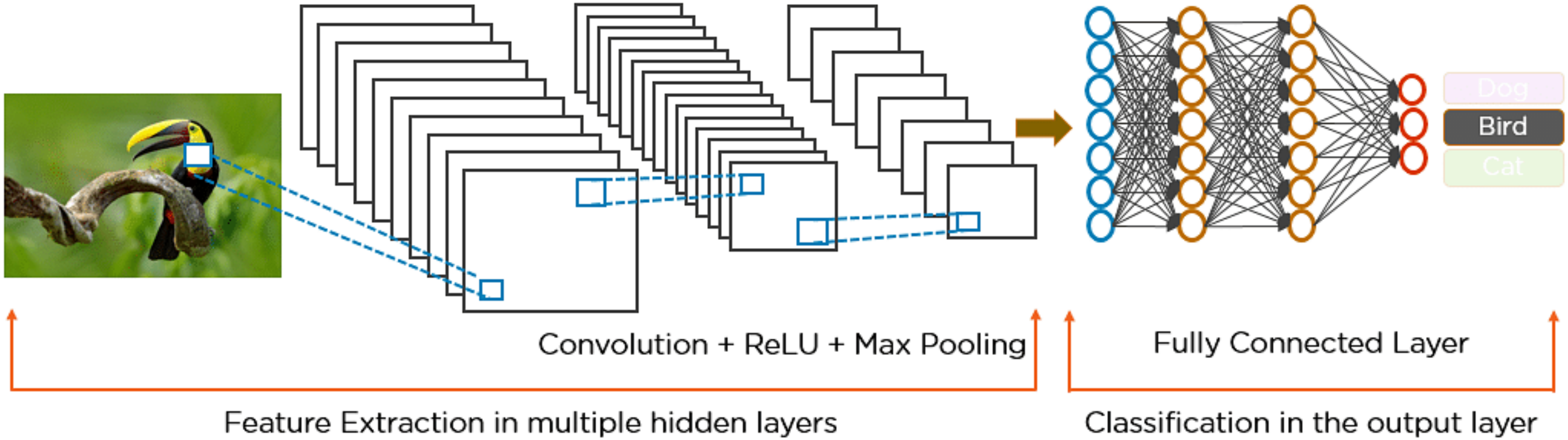


CNN : Fully Connected layer

- **FULLY CONNECTED LAYER:**
- This layer forms the last block of the CNN architecture, related to the task of classification. This is essentially a Fully connected Simple Neural Network, consisting of two or three hidden layers and an output layer generally implemented using ‘Softmax Regression’, that performs the work of classification among a large no of categories.
- In this, the input image from the previous layers are flattened and fed to the FC layer. The flattened vector then undergoes few more FC layers where the mathematical functions operations usually take place. In this stage, the classification process begins to take place.



CNN


Dog
Bird
Cat
Convolution + ReLU + Max Pooling
Fully Connected Layer
Feature Extraction in multiple hidden layers
Classification in the output layer

ImageNet Data

ImageNet

- **ImageNet Dataset**
- ImageNet is a large dataset of annotated photographs intended for computer vision research.
- The goal of developing the dataset was to provide a resource to promote the research and development of improved methods for computer vision.
- Based on statistics about the dataset recorded on the ImageNet homepage, there are a little more than 14 million images in the dataset.
- A little more than 21 thousand groups or classes (synsets), and a little more than 1 million images that have bounding box annotations (e.g. boxes around identified objects in the images).
- The photographs were annotated by humans using crowdsourcing platforms such as Amazon’s Mechanical Turk.
- The project to develop and maintain the dataset was organized and executed by a collocation between academics at Princeton, Stanford, and other American universities.

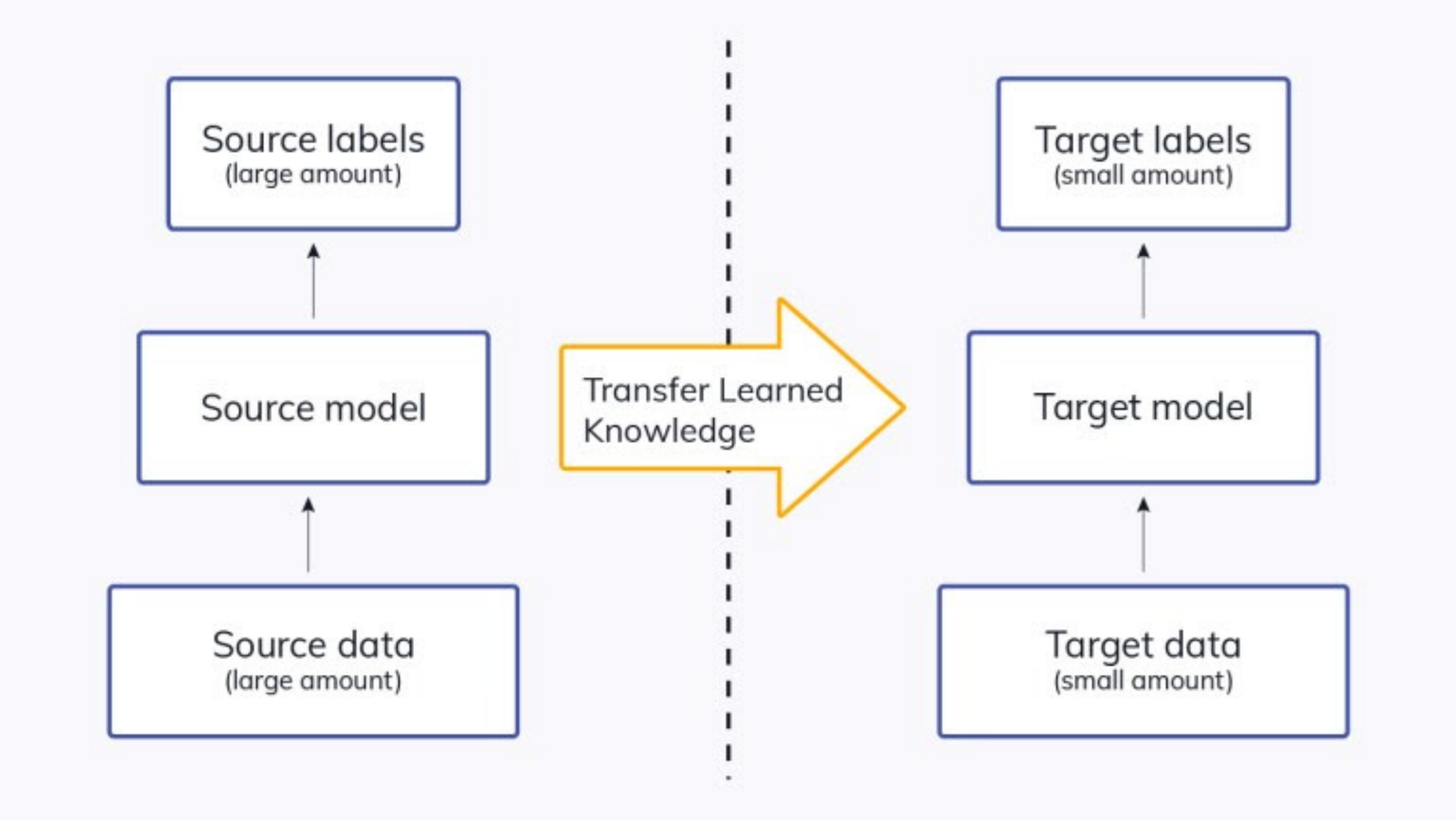
ImageNet

Transfer Learning

Transfer learning

- **Transfer learning**
- A common and highly effective approach to deep learning on small image datasets is to leverage a pre-trained network.
- A pre-trained network is simply a saved network previously trained on a large dataset, typically on a large-scale image classification task.
- If this original dataset is large enough and general enough, then the spatial feature hierarchy learned by the pre-trained network can effectively act as a generic model of our visual world, and hence its features can prove useful for many different computer vision problems, even though these new problems might involve completely different classes from those of the original task.
- This is referred to as transfer learning where we are transferring knowledge learned from solving one problem to another problem.
- For instance, one might train a network on ImageNet (where classes are mostly animals and everyday objects) and then re-purpose this trained network for something as remote as identifying furniture items in images. Such portability of learned features across different problems is a key advantage of deep learning.

Transfer learning

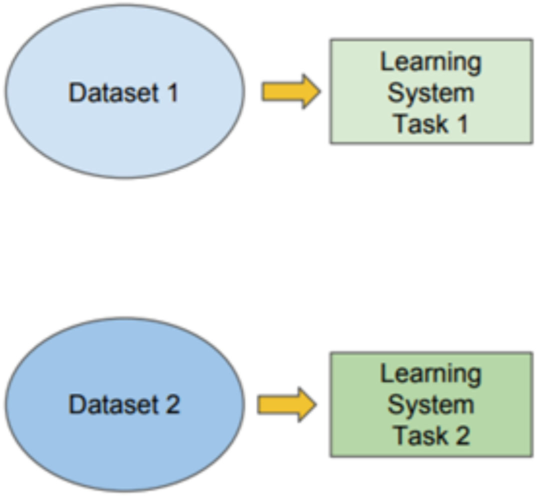


Transfer learning Vs Machine learning

Traditional ML vs Transfer Learning

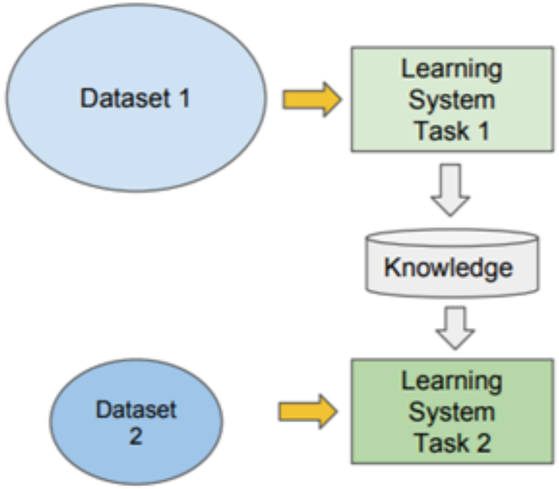
Traditional ML

- Isolated, single task learning:
  - Knowledge is not retained or accumulated. Learning is performed w.o. considering past learned knowledge in other tasks

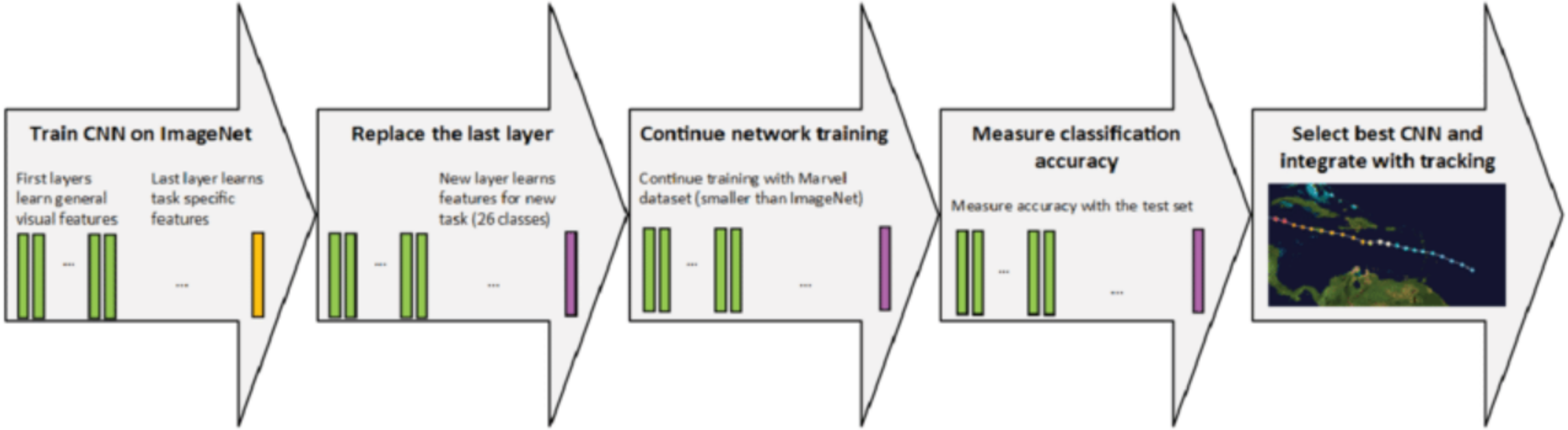


Transfer Learning

- Learning of a new tasks relies on the previous learned tasks:
  - Learning process can be faster, more accurate and/or need less training data



Transfer learning:Process



Transfer learning : Pre – Trained models

- **Pre – Trained models / Different types of CNN Architectures**
- For computer vision:
  - VGG-16
  - VGG-19
  - Inception V3
  - Xception
  - ResNet -50
  - Dense Net
  - And many more …………


Thank you!