Convolutional Neural Networks (CNN)

Module 1: Introduction to Neural Networks

- **1.1. Understanding Neural Networks**
  - Basics of neural networks.
  - The concept of weights, biases, and layers.
  - Role of Multilayer Perceptron (MLP) in deep learning.
- **1.2. Limitations of Multilayer Perceptrons**
  - Challenge of high-dimensional data (example: image with 2400 x 802 x 3 = 5,774,400 pixels).
  - Computational inefficiency in image processing.
  - Lack of spatial hierarchy in MLPs.

Module 2: Basics of Convolutional Neural Networks (CNN)

- **2.1. Why CNN?**
  - Resolution of MLP's limitations.
  - How CNN introduces spatial understanding and reduces dimensions efficiently.
- **2.2. Building Blocks of CNN**
  - **Convolution Operation**
    - Explanation of convolution.
    - How feature maps are created using filters (e.g., edge detection).
  - **ReLU Layer**
    - Introduction to non-linearity using ReLU.
    - Benefits of ReLU in activation functions.
  - **Pooling Layers**
    - Types: Max, Min, Mean Pooling.
    - Concept of stride and dimension reduction.
  - **Flattening**
    - Transforming feature maps into arrays for fully connected layers.
  - **Fully Connected Layer**
    - Explanation of input, hidden, and output layers in classification tasks.

Module 3: Steps in CNN Processing

1. **Loading the Dataset**: Handling datasets and preprocessing.
2. **Convolution Operation**: Extracting features from the input image.
3. **Pooling with Strides**: Reducing the size of feature maps (e.g., max pooling).
4. **Flattening the Data**: Preparing data for the fully connected layers.
5. **Fully Connected Layer**: Final classification of outputs (e.g., dog, cat, etc.).

Module 4: Image Classification Using CNN

- **4.1. Workflow Overview**
  - End-to-end image classification pipeline.
  - Conversion of images into pixel arrays.
  - Building, fitting, and compiling the CNN model.
- **4.2. Hands-On Example**
  - Step-by-step explanation of a CNN for an image classification task (e.g., identifying animals).
  - Visual representation of feature extraction, pooling, and classification.

Module 5: Comparison Between CNN and MLP

- **5.1. Advantages of CNN Over MLP**
  - Reduction in computational complexity.
  - Preservation of spatial structure.
  - Superior performance on image data.
- **5.2. When to Use CNN vs. MLP**
  - Scenarios and datasets where each model excels.

Module 6: Advanced Concepts

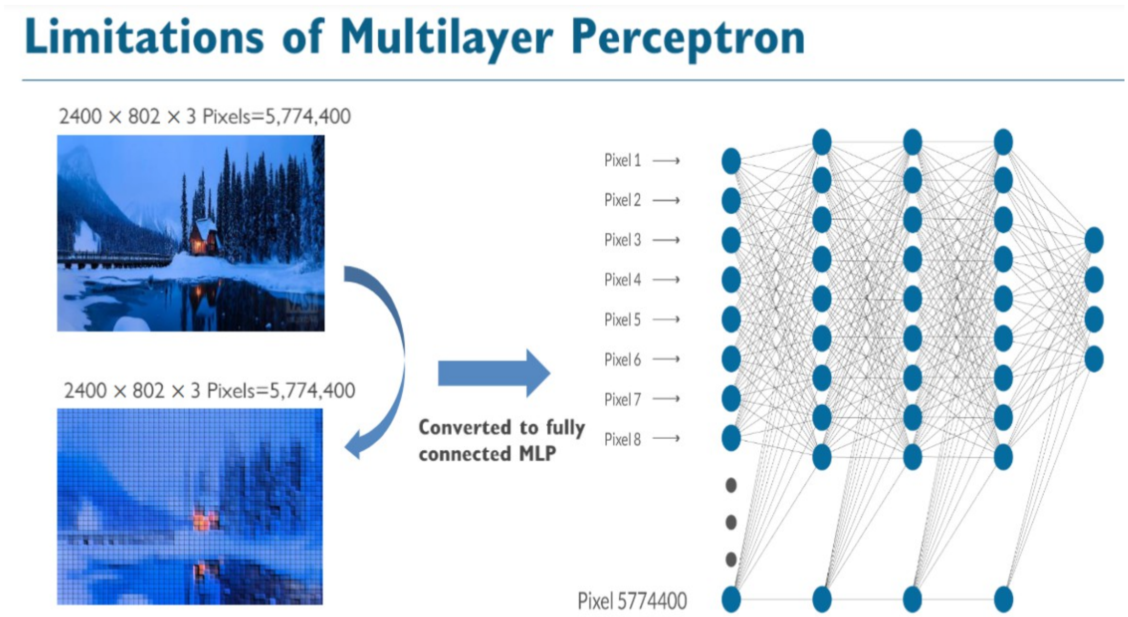
- **6.1. Different Types of Pooling**
  - Detailed comparison of max, min, and mean pooling.
  - Effect of stride and filter size on pooling performance.
- **6.2. Higher-Resolution Image Analysis**
  - Addressing challenges with high-resolution images using CNN.
  - Example applications in object detection and recognition.

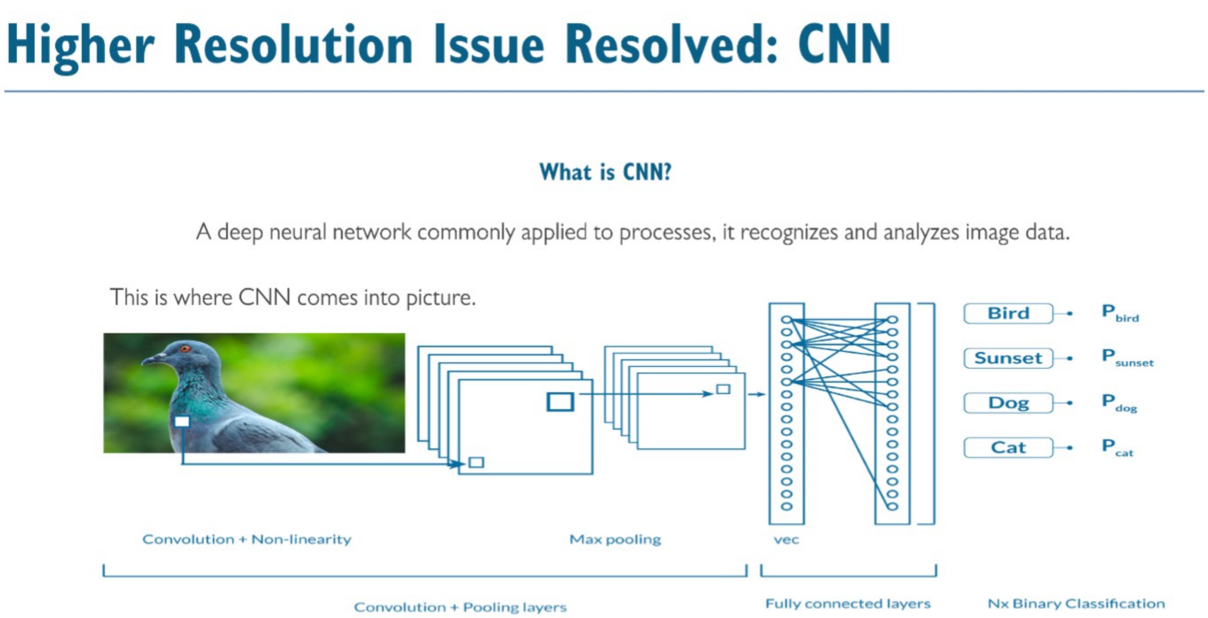
Module 7: Practical Applications

- **7.1. Real-World Examples of CNN**
  - Medical imaging.
  - Autonomous vehicles.
  - Facial recognition systems.
- **7.2. Building and Deploying a CNN**
  - Tools and frameworks (e.g., TensorFlow, PyTorch).
  - Deployment strategies.

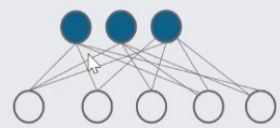
Module 8: Conclusion

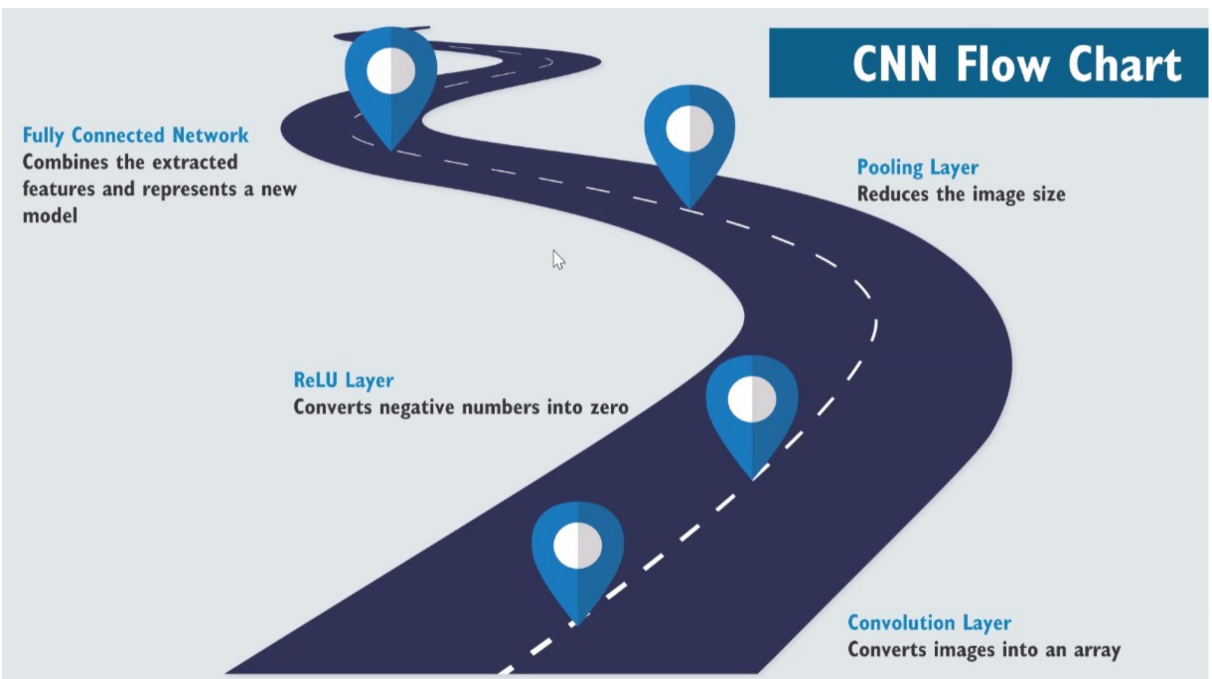
- Recap of key concepts.
- Future of CNN and its evolving landscape in AI and machine learning.



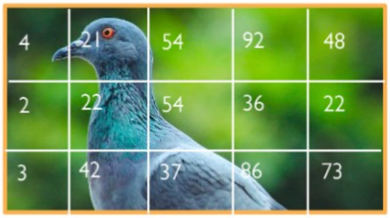


CNN vs. MLP

CNN	MLP
▪ Layers in CNN: Convolution → ReLU → Pooling → Fully connected	▪ Layers in MLP: Input→ Fully connected→ Output
▪ Reduces the number of parameters by sharing weights (Feature mapping)	▪ Does not reduce the number of parameters as it is fully connected
▪ Can easily work with high pixelated images	▪ Because of fully connected neurons it does not work smoothly with high pixelated images
	



Working of Convolutional Layer



4	21	54	92	48
2	22	54	36	22
3	42	37	86	73

×

0	-1
1	1

Filter/Feature map

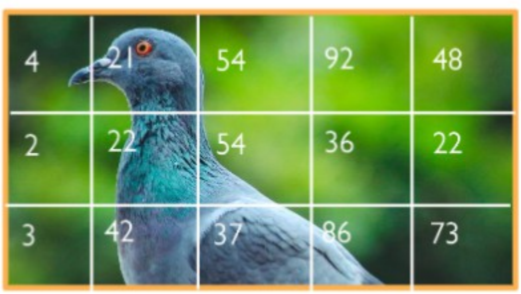
Terminology Alert!

Filter to extract important features from the image

3	22	-2	10
23	25	87	137

- Extracted features from the image
- Dimension reduced

Working of ReLU Layer

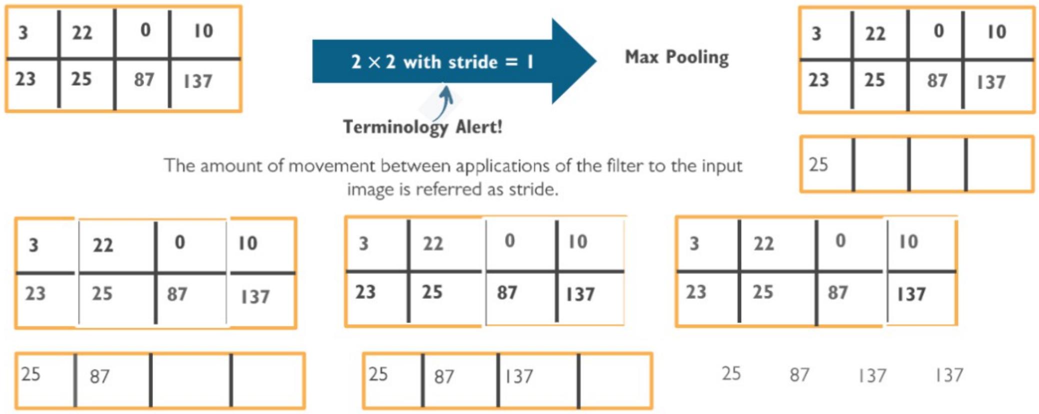


3	22	-2	10
23	25	87	137

Applying ReLU layer

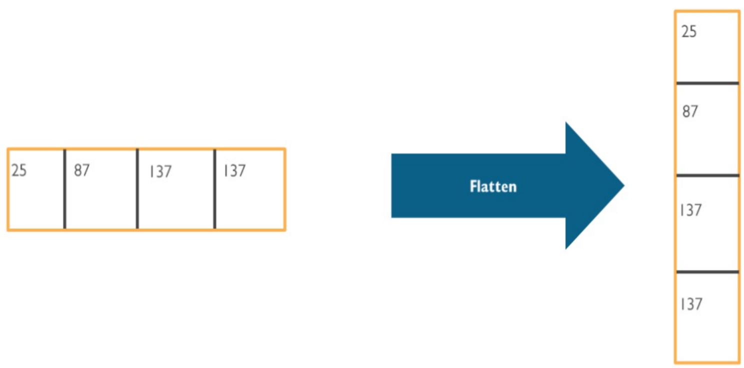
3	22	0	10
23	25	87	137

Working of Pooling Layer

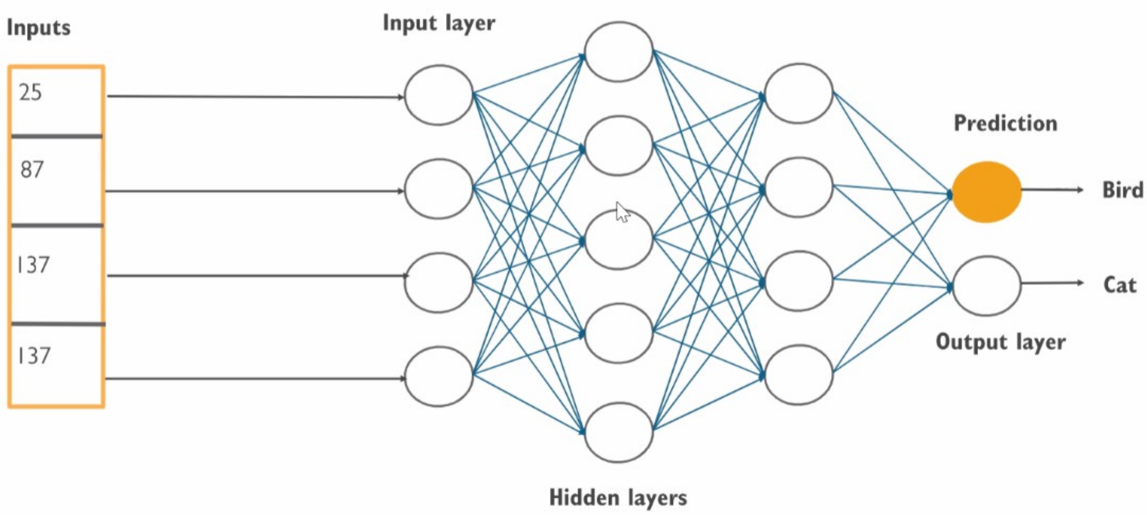


Flatten the Data

Converting the pooled feature map into an array is known as data flattening.



Working of Fully Connected Layer



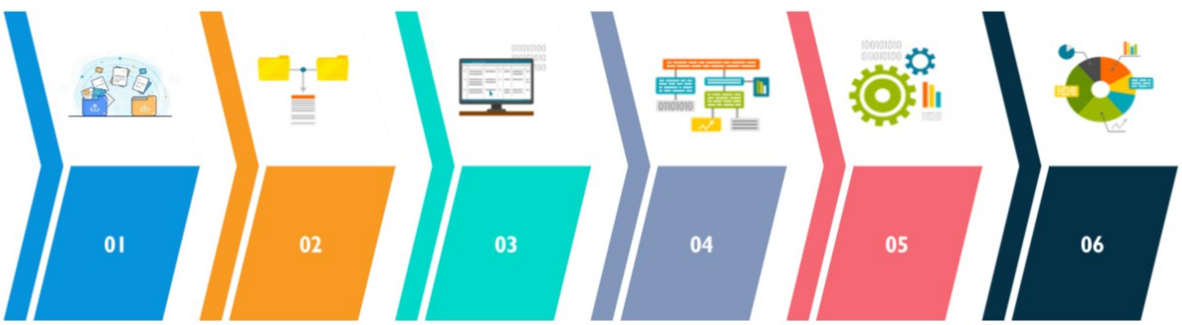
Problem Scenario

Summit Technologies is inviting candidates to interview them for the position of Data Scientist. Let's assume you are one of the candidates. During the interview, you are asked the following question:
Consider a data set consisting of two classes, Cats and Dogs from Dropbox. Build a model to classify the images using TensorFlow 2.x.
Now, you must build the model that predicts the names of the animals and which class it belongs to.

Sample of animal

Image Classification Using CNN

The following steps are performed for image classification using CNN:



01. Loading the data set and unzipping the files
02. Initializing the parameters
03. Checking for the channel first and rescaling the image
04. Building the CNN model
05. Fitting and compiling the model
06. Prediction by the model