SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107
Accredited by NAAC-UGC with 'A' Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of AI &DS

Course Name – 19AD602 DEEP LEARNING

III Year / VI Semester

Unit 3-DIMENSIONALITY REDUCTION

Topic: Introduction to Convnet

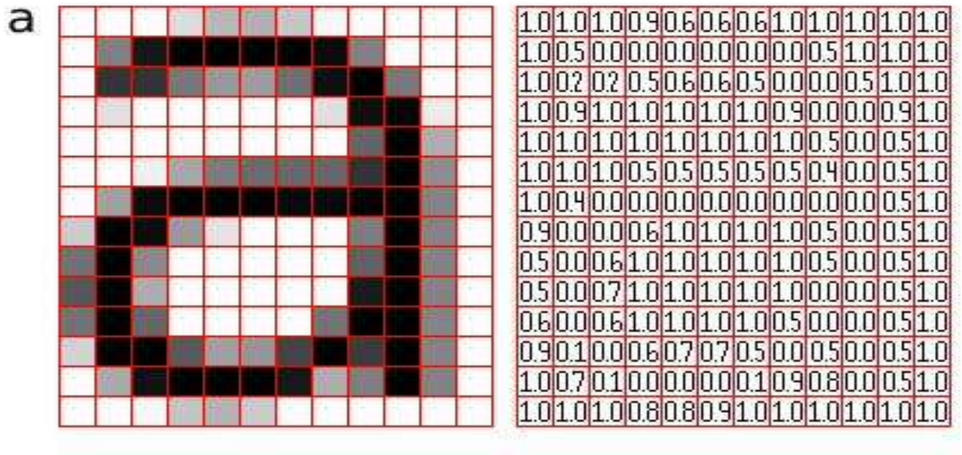
Case Study: Real-Time Object Detection with CNNs

A company developed an AI-powered surveillance system using a Convolutional Neural Network (CNN). The system could detect and classify objects (e.g., vehicles, people, and animals) in real-time with 95% accuracy, enhancing security and operational efficiency.

CNN:

- A Convolutional Neural Network, also known as CNN or ConvNet , is a class of neural networks that specializes in processing data that has a grid-like topology, such as an image.
- A digital image is a binary representation of visual data.
- It contains a series of pixels arranged in a grid-like fashion that contains pixel values to denote how bright and what color each pixel should be.

Representation of image as a grid of pixels :



ALEX NET:

Overview/Purpose

- Animal data set
- Using AlexNet to help identify the distinguishing characteristics in animals to determine whether they are classified as “land” (terrestrial) or “sea” (aquatic) creatures.

Real World Application

- If AlexNet is successful, this could help pave the way for further use, integration, and development of AlexNet and AI in detection/feature analysis

What is AlexNet?

- AlexNet is a deep convolutional neural network designed for image classification tasks. AlexNet processes an input image through a series of convolutional layers, activation functions, pooling operations, and fully connected layers to predict the probability of the image belonging to a set of predefined classes.

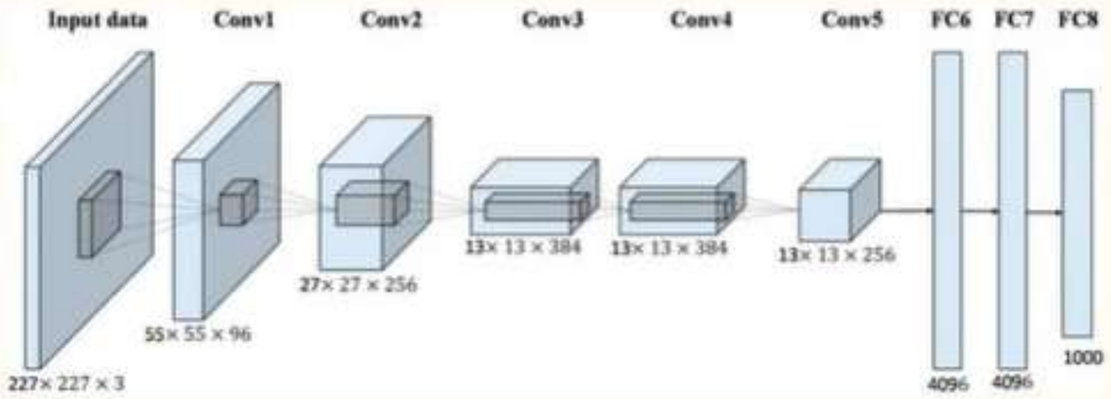
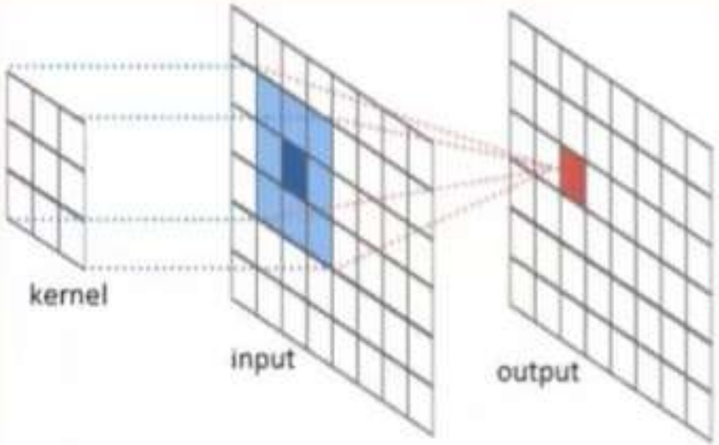


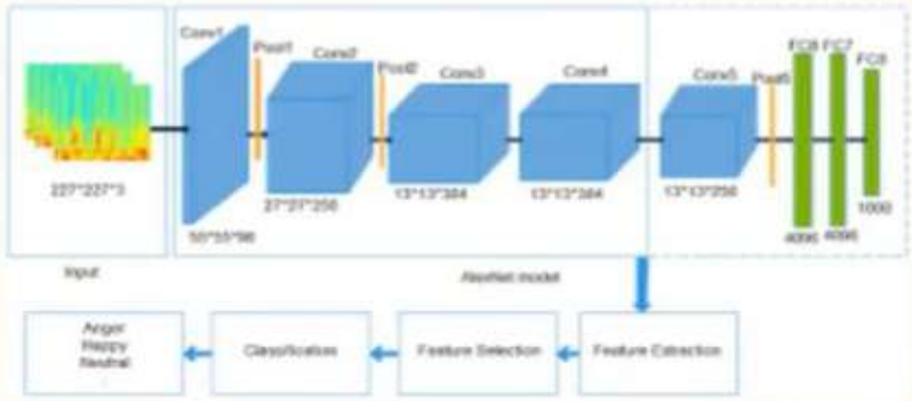
Image Convolution



- AlexNet uses convolutional layers for image feature extraction. A convolutional layer applies a set of filters to an input image and produces a set of output feature maps

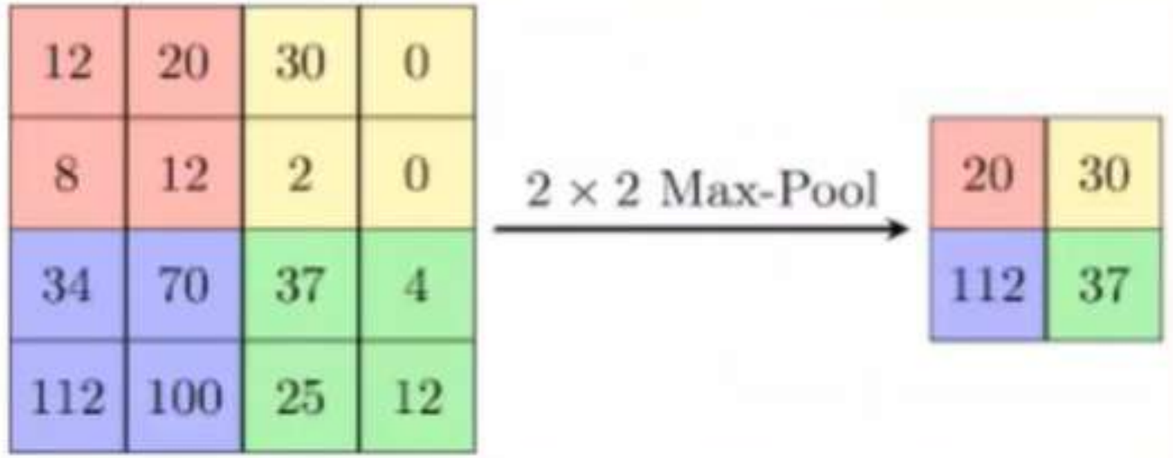
Feature Extraction

- AlexNet uses a set of convolutional layers for feature extraction from the input image. The filters in these layers are designed to learn local patterns and features from the image, including edges, textures, and shapes



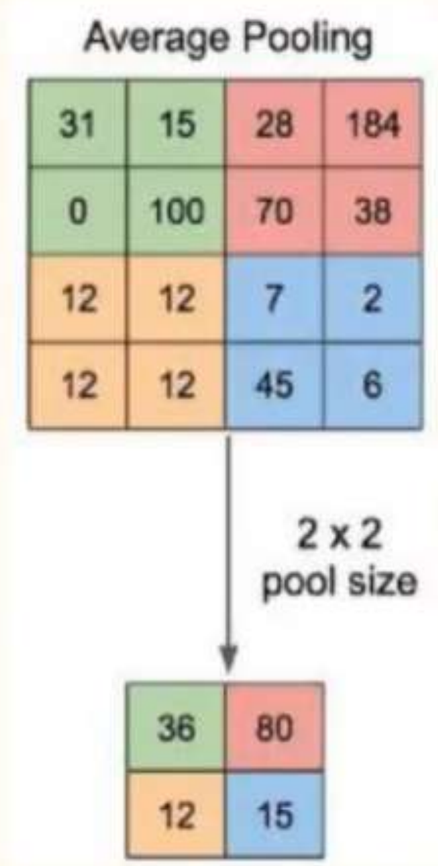
Max Pooling

AlexNet uses max pooling layers for reducing the spatial dimensions of the features maps produced by the convolutional layers. Max pooling partitions each feature map into non-overlapping rectangular regions and outputs the maximum value within each region



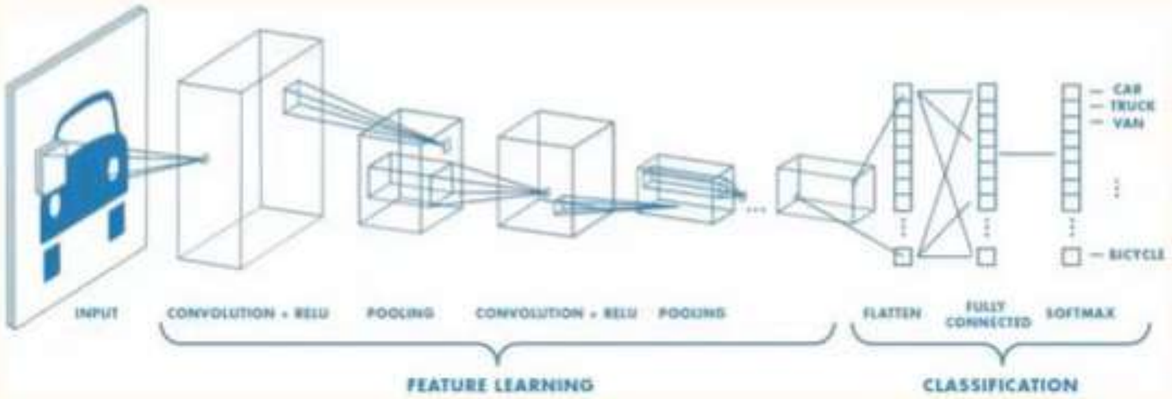
Average pooling

- Average pooling is used in AlexNet's last fully connected layer to reduce the number of parameters in the network and prevent overfitting, while preserving some spatial information that may be useful for classification.
- By taking the average of the feature maps, the network is able to obtain a single feature vector that summarizes the most important information from each feature map



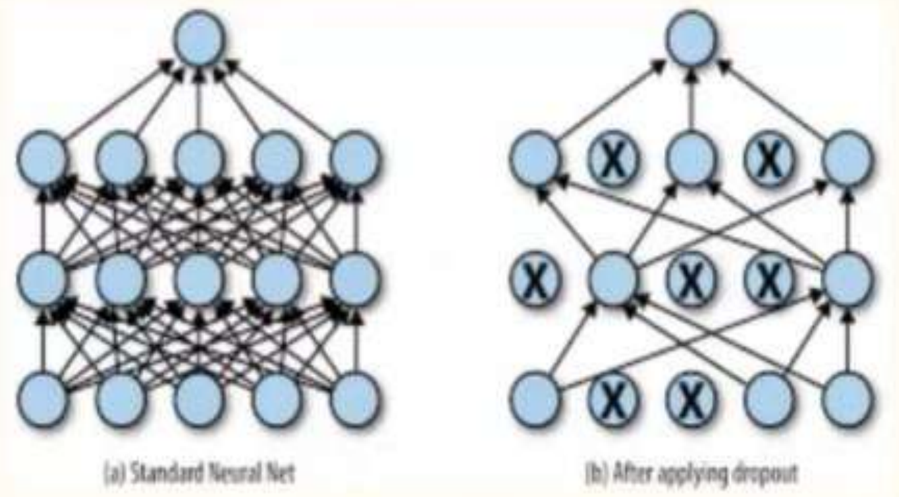
ReLU

AlexNet uses the ReLU (Rectified Linear Unit) activation function in the convolutional and fully connected layers. This introduces non-linearity to the network, which allows it to learn more complex and non-linear functions of the input data.



Dropout

Dropout regularization is used in the fully connected layers to prevent overfitting. Dropout randomly drops out a fraction of the units in a layer during training, forcing the remaining units to learn more robust and diverse features.



Results

```
----------
Train Acc: 1.0000
Valid Acc: 0.9412
13  of  19
----------
Train Acc: 0.9815
Valid Acc: 0.9412
14  of  19
----------
Train Acc: 1.0000
Valid Acc: 0.9412
15  of  19
----------
Train Acc: 0.9815
Valid Acc: 0.9412
16  of  19
----------
Train Acc: 1.0000
Valid Acc: 0.9412
17  of  19
----------
Train Acc: 0.9630
Valid Acc: 0.9412
18  of  19
----------
Train Acc: 0.9815
Valid Acc: 0.8824
19  of  19
----------
Train Acc: 0.9815
Valid Acc: 0.8824
Training complete
```

Activity: Image Classification using CNN

1. **Objective**: Train a CNN to classify images into categories (e.g., cats, dogs, and birds).
2. **Steps**:
   - Gather a dataset of labeled images.
   - Build a CNN using TensorFlow or PyTorch.
   - Train the model, evaluate its accuracy, and test it on unseen data.
3. **Outcome**: Understand how CNNs process visual data and improve classification accuracy through parameter tuning.

THANK YOU