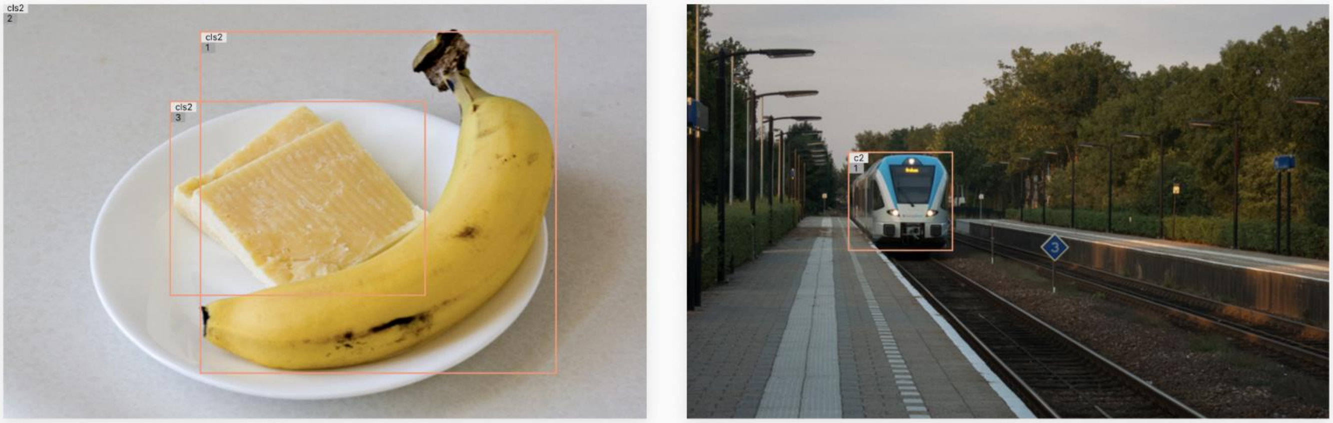
National University of Computer & Emerging Sciences, Karachi
Artificial Intelligence-School of Computing
Fall 2024, Lab Manual - 06

Course Code (AI4002)	**Course: Computer Vision Lab**
Instructor(s):	**Sohail Ahmed**

Objectives:

- Introduction to Image Classification
- Introduction to CNN
- AlexNet,
- GoogleNet

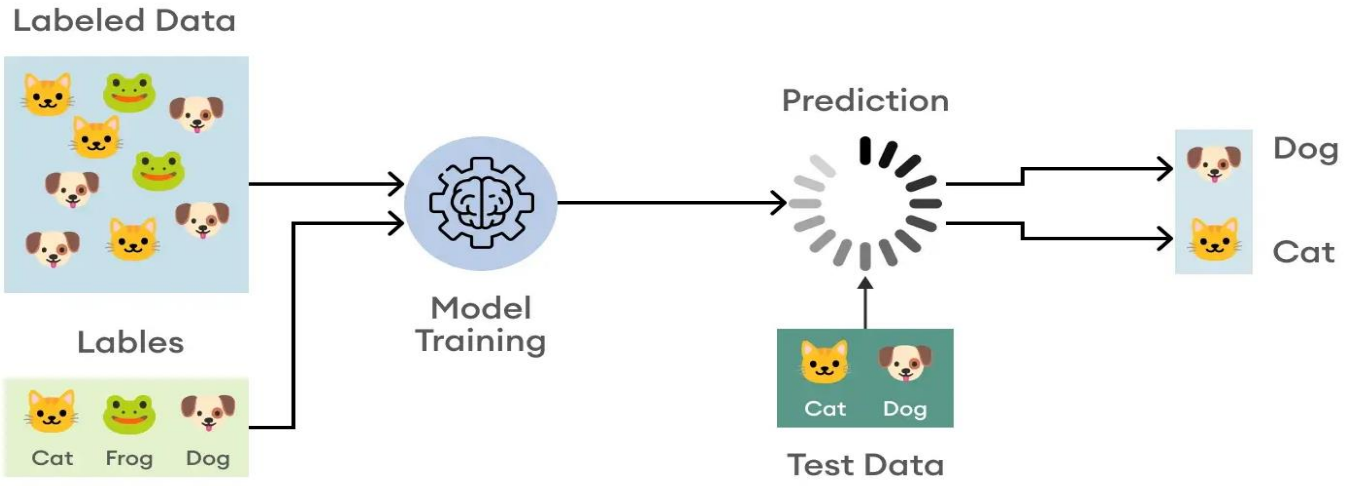
Introduction to Image Classification:



Object Localization

Object Detection

Image classification is a branch of computer vision that deals with categorizing images using a set of predetermined tags on which an algorithm has been trained.



Types of image classification

Depending on the problem at hand, there are different types of image classification methodologies to be employed. These are binary, multiclass, multilabel, and hierarchical.

1. **Binary:** Binary classification takes an either-or logic to label images, and classifies unknown data points into two categories. When your task is to categorize benign or malignant tumors, analyze product quality to find out whether it has defects or not, and many other problems that require yes/no answers are solved with binary classification.

2. **Multiclass**: While binary classification is used to distinguish between two classes of objects, multiclass, as the name suggests, categorizes items into three or more classes. It's very useful in many domains like NLP (sentiment analysis where more than two emotions are present), medical diagnosis (classifying diseases into different categories), etc.

3. **Multilabel:** Unlike multiclass classification, where each image is assigned to exactly one class, multilabel classification allows the item to be assigned to multiple labels. For example, you may need to classify image colors and there are several colors. A picture of a fruit salad will have red, orange, yellow, purple, and other colors depending on your creativity with fruit salads. As a result, one image will have multiple colors as labels.

4. **Hierarchical:** Hierarchical classification is the task of organizing classes into a hierarchical structure based on their similarities, where a higher-level class represents broader categories and a lower-level class is more concrete and specific. Let's get back to our fruits and understand the concept based on a juicy example.

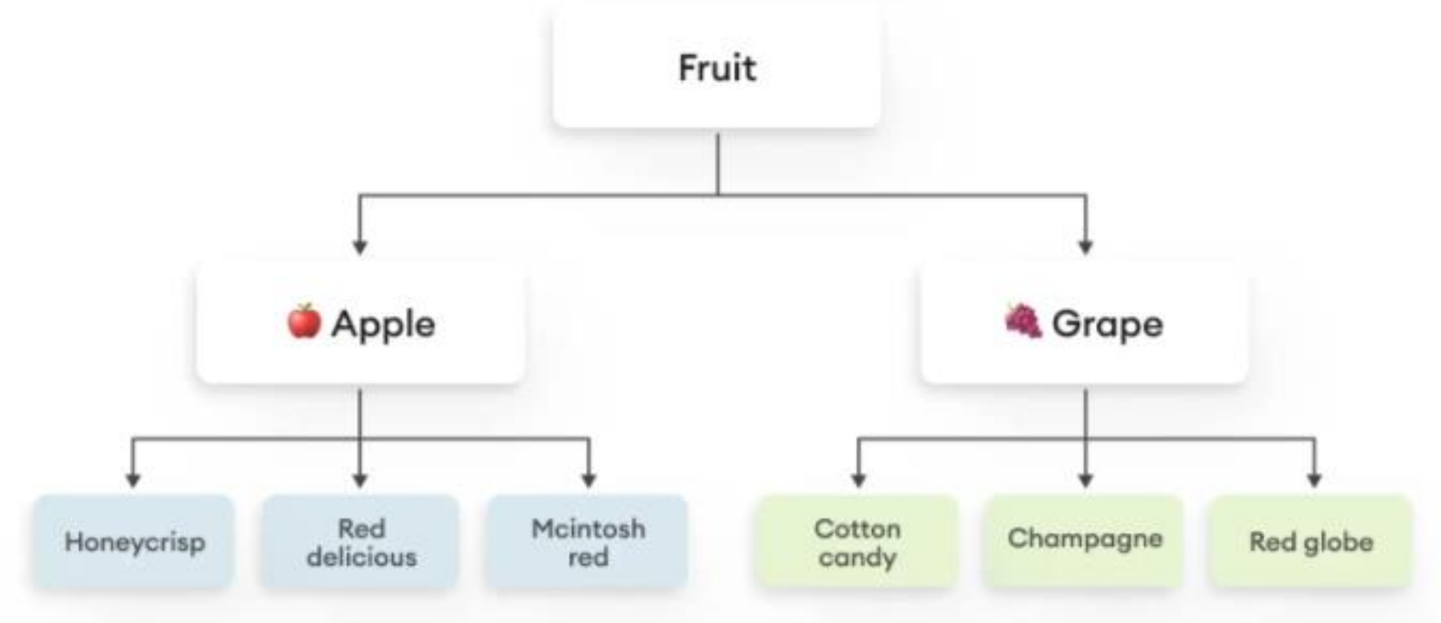
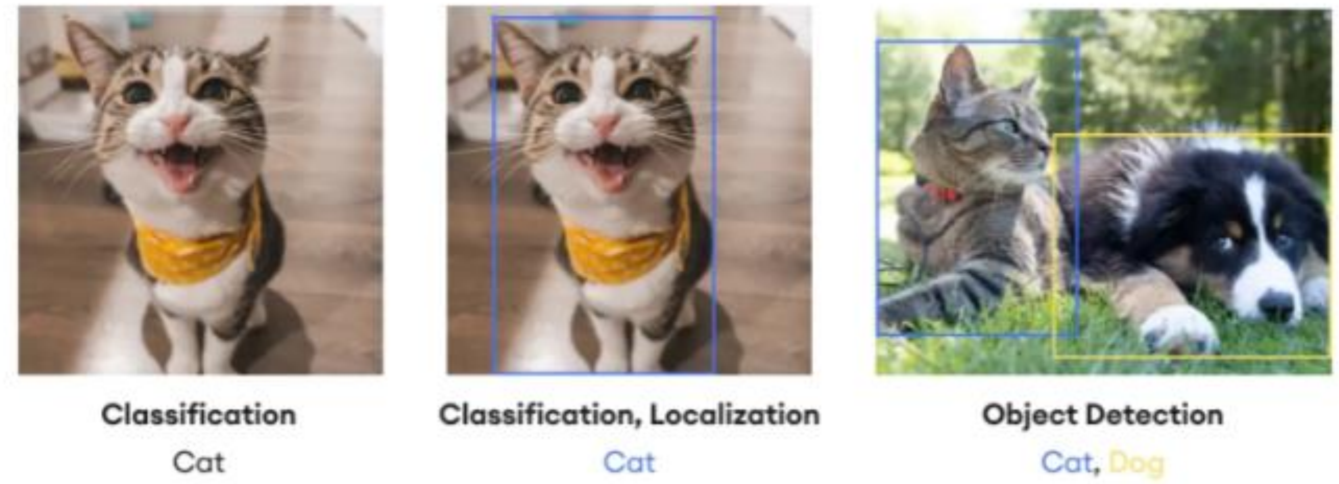


Image classification vs. object detection

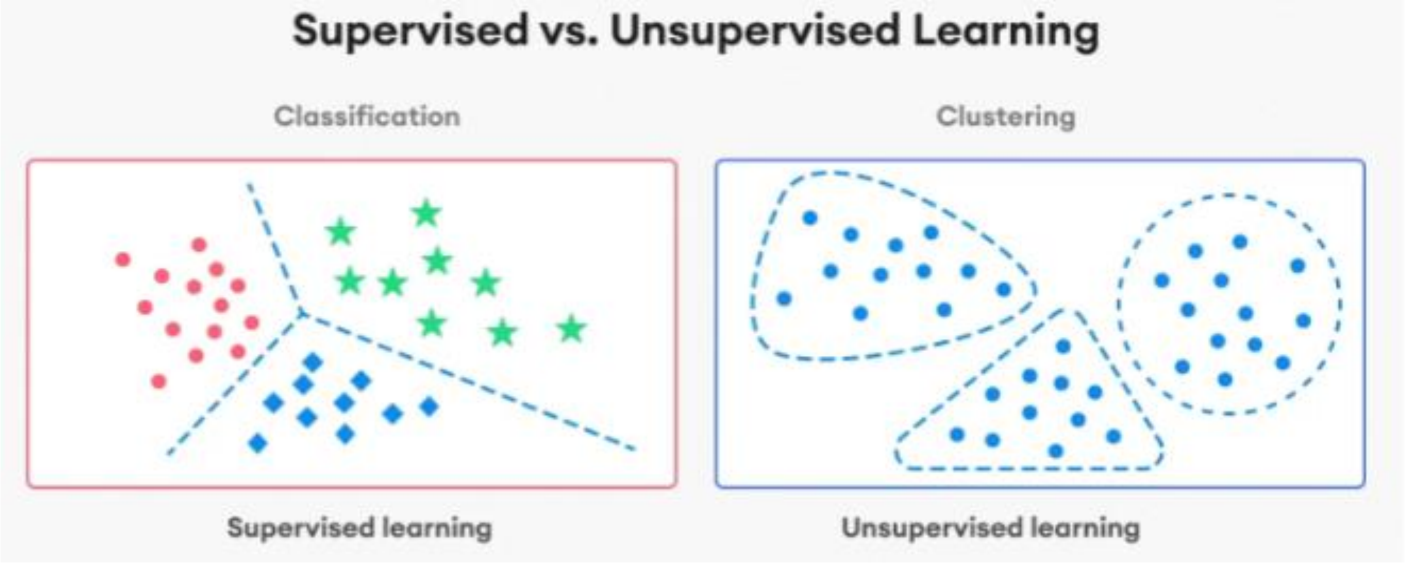
Image classification refers to assigning a specific label to the entire image. On the other hand, object localization goes beyond classification and focuses on precisely identifying and localizing the main object or regions of interest in an image. By drawing bounding boxes around these objects, object localization provides detailed spatial information, allowing for more specific analysis.



Object detection on the other hand is the method of locating items within and image assigning labels to them, as opposed to image classification, which assigns a label to the entire picture.

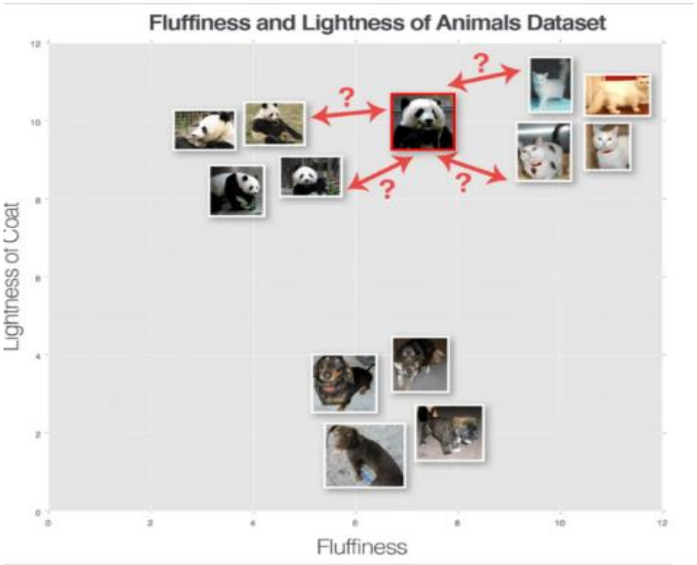
How image classification works

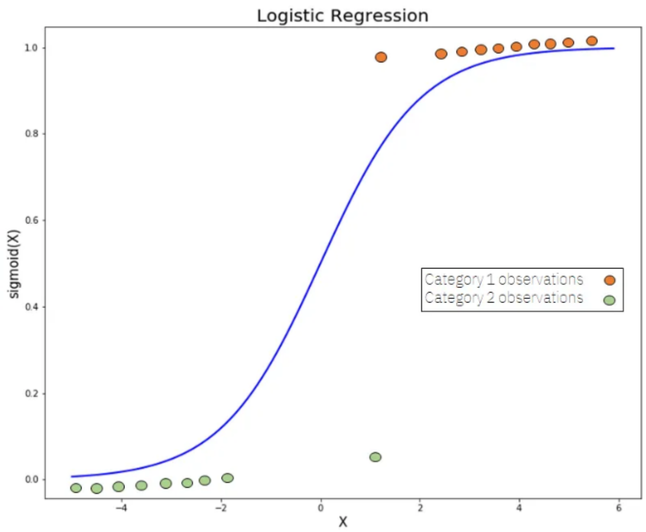
Image pre-processing -> feature extraction -> object classification

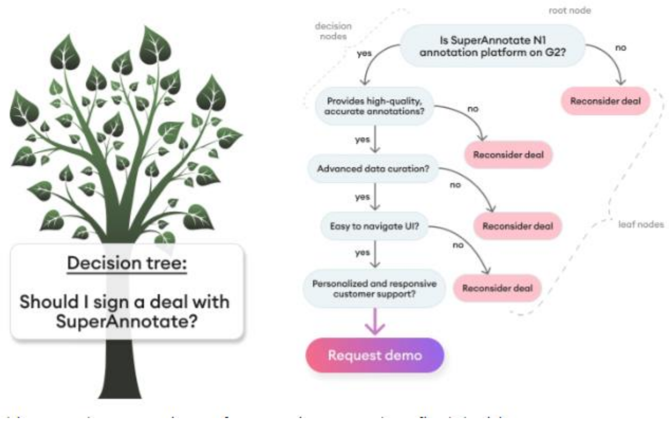


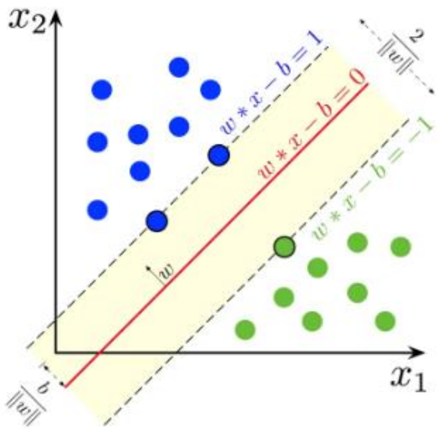
Some of the algorithms

1. Logistic regression
2. K nearest neighbors
3. Support vector machines
4. Decision trees







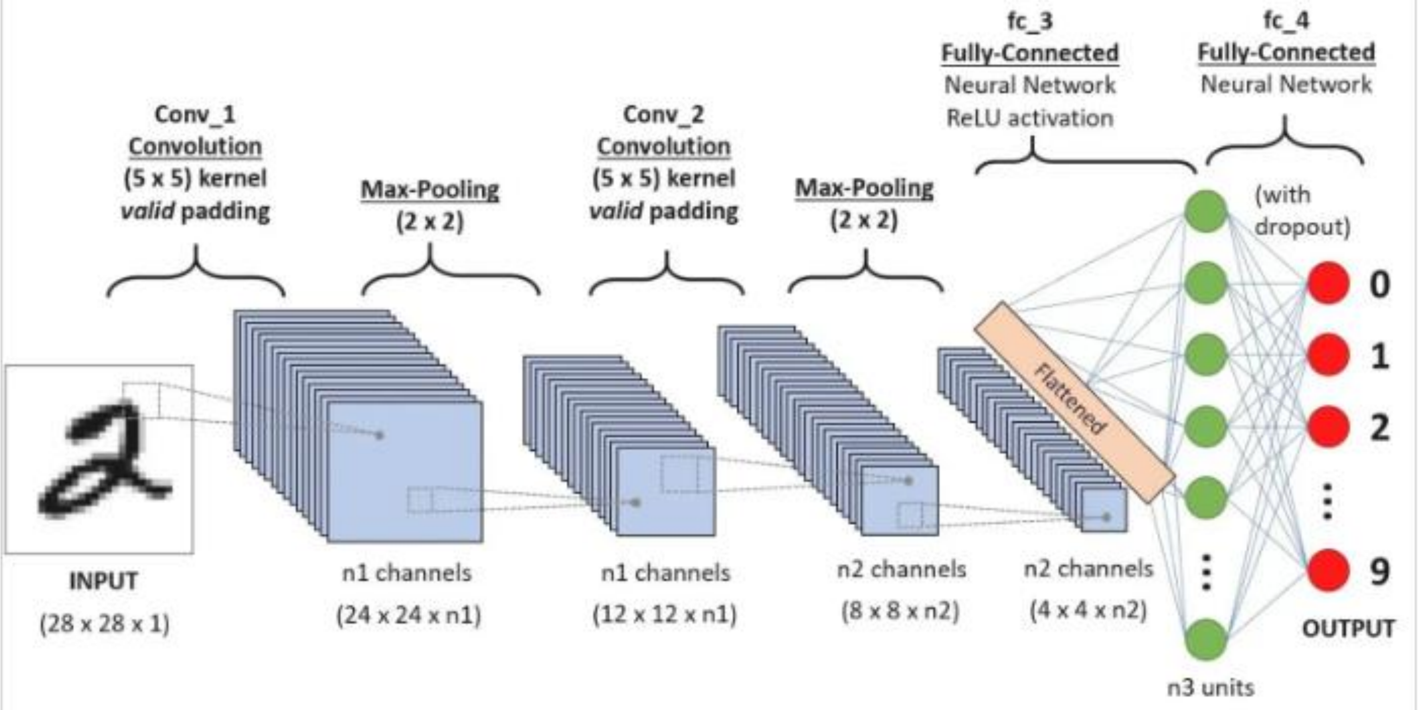


<u>Deep neural networks for image classification</u>

Deep learning methods have proven to take computer vision tasks to an even higher level of accuracy and efficiency, all thanks to convolutional neural networks (CNNs). Note that CNNs are indeed a part of supervised learning algorithms, and due to their significance and current prominence in image classification, we spared a separate section to discuss it.
The variety of layers, starting with the input layer, to the hidden inner layers, and output layer are what make the network "deep." In brief, these are the core components of convolutional neural networks:

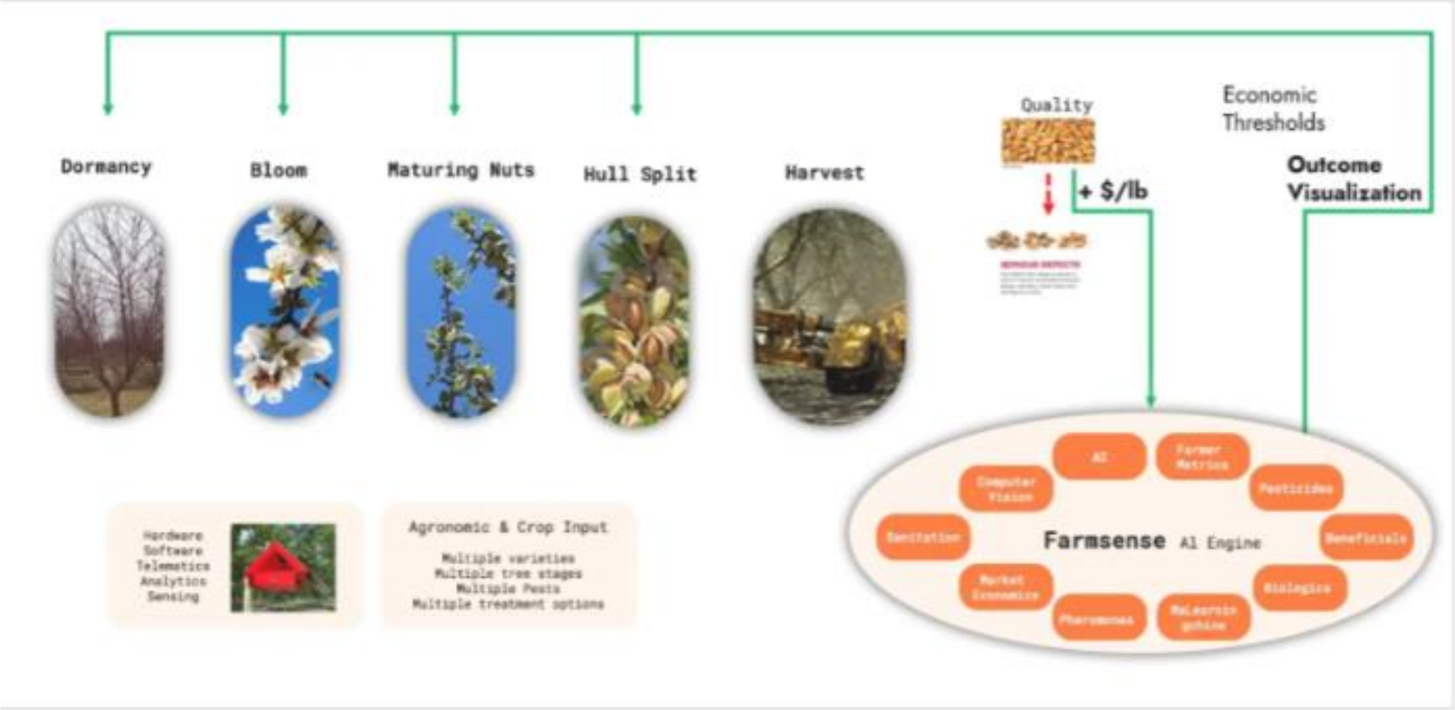
1. **<u>Input layer:</u>** The first layer of each CNN is the input layer, where images or videos are taken and pre-processed and then passed to the next layers.
2. **<u>Convolution layer:</u>** The next layer applies learnable filters to extract features from images. The output of this layer is a feature map that indicates the presence or absence of particular features in the input image.
3. **<u>Pooling layer:</u>** The extracted features are then passed to the pooling layer, where the large images are shrunk down while making sure the most important information is preserved. The most common pooling operation, max pooling, selects the maximum value of each sub-region of the feature map.
4. **<u>ReLU layer:</u>** The ReLU (Rectified Linear Unit) changes every negative number of the pooling layer to 0 to maintain mathematical stability and keep learned values from being stuck around 0 or jumping into infinity. Surprisingly, this simple function can allow your model to account for non-linearities and interactions very well.

5. **Fully connected layer:** The fully connected layer takes the output of the previous layers and produces a final classification. Each neuron in this layer is connected to all neurons in the previous layer.



Applications of Image Classifications

Agriculture etc



Key Image Classification Metrics

If we’re looking to train our models to function similarly to the human brain, then monitoring how well each model performs is of utmost importance. For your model to pass the test and be used in a real world setting, a few things need to be considered, including accuracy, precision, recall, and F1 score.

Accuracy

The most basic function measurement in classification metrics is accuracy. Hare, we measure the number of correct predictions made by your model as a proportion of the total number of predictions.

$$\text{Accuracy} = \frac{\#\ of\ correct\ predictions}{\#\ of\ total\ predictions}$$

Precision

precision is the proportion of positive identifications by your model that are classified correctly. Precision is calculated in the following way.

$$\text{Precision} = \frac{TP}{TP+FP}$$

Recall

The proportion of positive identifications that were classified correctly

$$\text{Recall} = \frac{TP}{TP+FN}$$

F1 Score: the measurement of compromise

Precision and recall each influence the other, the F1 score serves the purpose of calculating a single number to represent both values. To solve for F1, input your data into the following formula,

$$\text{F1 Score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

The F1 score provides an invaluable look at your metrics in that it assigns equal balance to both precision and recall. Your F1 score also provides you with information about your precision and recall calculations and further helps you adjust your model in the following ways:

1. If your F1 score is high, then both recall and precision are high
2. A low F1 score is indicative of low precision and recall
3. A medium F1 score tells you that one of precision and recall is low while the other is high

AlexNet:

AlexNet, developed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, is a landmark model that won the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) in 2012. It introduced several innovative ideas that shaped the future of CNNs.

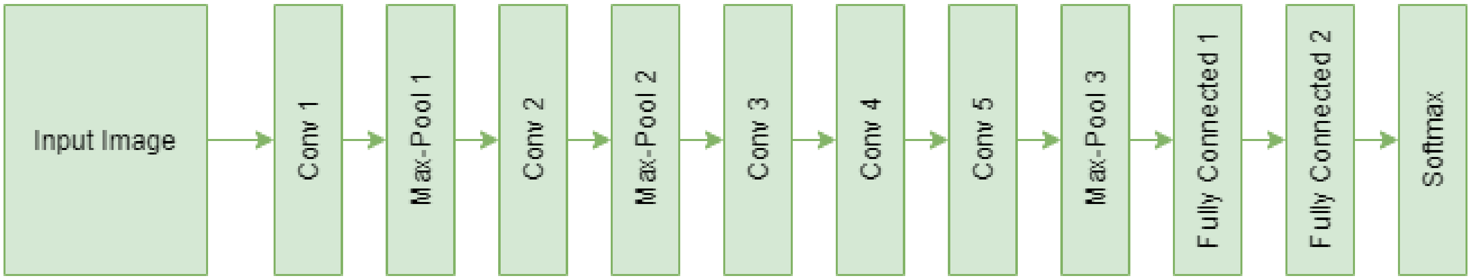
AlexNet Architecture:

AlexNet consists of 8 layers, including 5 convolutional layers and 3 fully connected layers. It uses traditional stacked convolutional layers with max-pooling in between. Its deep network structure allows for the extraction of complex features from images.

- The architecture employs overlapping pooling layers to reduce spatial dimensions while retaining the spatial relationships among neighboring features.
- Activation function: AlexNet uses the ReLU activation function and dropout regularization, which enhance the model's ability to capture non-linear relationships within the data.

The key features of AlexNet are as follows: -

- AlexNet was created to be more computationally efficient than earlier CNN topologies. It introduced parallel computing by utilizing two GPUs during training.
- AlexNet is a relatively shallow network compared to GoogleNet. It has eight layers, which makes it simpler to train and less prone to overfitting on smaller datasets.
- In 2012, AlexNet produced ground-breaking results in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). It outperformed prior CNN architectures greatly and set the path for the rebirth of deep learning in computer vision.
- Several architectural improvements were introduced by AlexNet, including the use of rectified linear units (ReLU) as activation functions, overlapping pooling, and dropout regularization. These strategies aided in the improvement of performance and generalization



GoogleNet

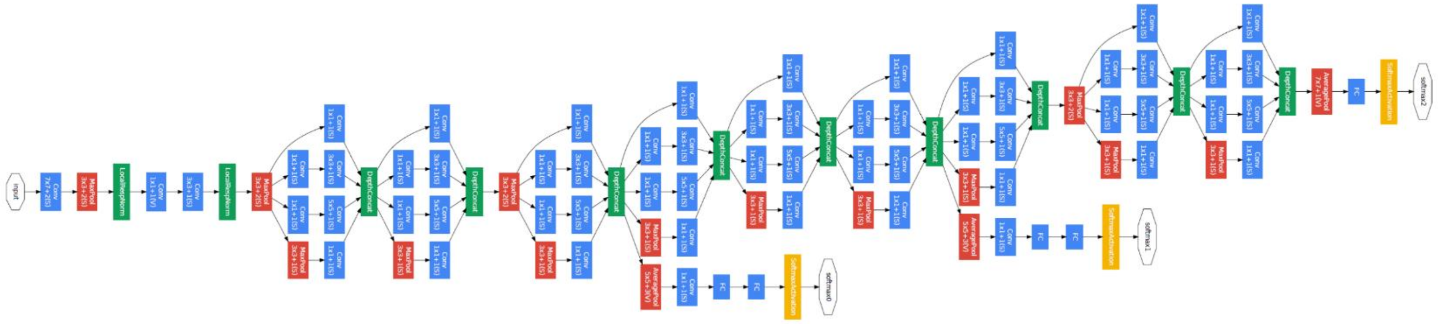
GoogleNet is also known as (Inception v1), it was developed by a team at Google led by Christian Szegedy. It won the ILSVRC in 2014 and introduced several innovative concepts that aimed to address the challenges faced by deep neural networks.

1. Inception Modules: GoogleNet utilizes inception modules which use a deep, multi-branch architecture. It is composed of multiple parallel convolutional layers with different filter sizes. This allows the model to capture features at various scales and resolutions simultaneously.

2. Dimensionality Reduction: To reduce computational complexity and improve efficiency, GoogleNet employs 1×1 convolutional layers for dimensionality reduction before applying larger convolutions. This helps to preserve important spatial information while reducing the number of parameters.
3. Auxiliary Classifiers: GoogleNet uses auxiliary classifiers at intermediate layers during training to combat the vanishing gradient problem and provide additional regularization.

The key features of GoogleNet are as follows:

1. GoogleNet tried to overcome deep CNNs' computational inefficiencies. It uses the Inception module which reduces the number of parameters in the network and boosts computing efficiency. It outperformed AlexNet in terms of accuracy while using fewer parameters compared to AlexNet.
2. GoogleNet is a considerably deeper network with 22 levels. Its depth enables it to collect more intricate characteristics and patterns from images, allowing it to perform better on larger and more complicated datasets.
3. In 2014, GoogleNet won the ILSVRC, beating AlexNet. It demonstrated the efficiency of its Inception module by achieving improved accuracy while utilizing fewer parameters.



Differences between AlexNet and GoogleNet

Features	AlexNet	GoogleNet
Architecture	Deep (8 layers)	Deep (22 layers)
Activation Function	ReLU	ReLU
Pooling	Overlapping	Non-overlapping
Convolution	Consecutive	Parallel (inception)
Dimensionality	No reduction	1×1 Convolution
Regularization	Dropout	Auxiliary Classifiers

Codes Are Provided Separately:

Lab Task

1. Gender Classification- Develop a CNN model that can classify the gender of individuals in images. Use a dataset of human faces labeled with gender information. Combining Haar Cascade for face detection and landmarks can help in accurately cropping and aligning faces before classification. (You can use the dataset https://www.kaggle.com/datasets/ashishjangra27/gender-recognition-200k-images-celeba).

2. Animal Facial Expression Recognition- Create a CNN-based model for recognizing facial expressions, such as happiness, anger, sadness, and others. Utilize a dataset of facial images labeled with these expressions (Use the dataset I provided you in the second lab). Additionally, use Haar Cascade for face detection and landmarks to improve the model's accuracy in localizing facial features.

3. Age Estimation- Build a system that estimates the age of a person in a Video. This is a regression task where you predict the age as a continuous value. You can use a CNN-based architecture like GoogleNet and fine-tune it for age estimation. (This should be based on live video Streaming, Dataset https://www.kaggle.com/datasets/arashnic/faces-age-detection-dataset).

4. Food Recognition- Create a model to classify different types of food items in images. This could be a useful application for dietary analysis or food recommendation systems. AlexNet or GoogleNet can be employed for this task. (Use Dataset: https://www.kaggle.com/datasets/sainikhileshreddy/food-recognition-2022/data)

5. Hand Gesture Recognition- Implement a system that can recognize and classify hand gestures in real-time using a camera feed. CNN models, combined with Haar Cascade for hand detection, can be used to achieve this. (https://www.kaggle.com/code/benenharrington/hand-gesture-recognition-database-with-cnn/input)