Week 1 Assessment – Forest Fire Detection

What is Deep Learning ?

Deep learning is a class of machine learning techniques that use multi-layered neural networks to model complex relationships in data by automatically learning hierarchical features. It's inspired by the structure and function of the human brain.

Key Features of Deep Learning:

- **Neural Networks**: Systems of interconnected nodes (neurons) organized in layers.
- **Multiple Layers**: Deep learning models have many hidden layers between input and output, allowing them to learn high-level features.
- **Automatic Feature Extraction**: Unlike traditional machine learning, deep learning models can automatically discover the best features from raw data.
- **High Performance with Big Data**: Deep learning shines when working with large datasets and high computational power (like GPUs).

Common Applications:

- **Image recognition** (e.g., facial recognition)
- **Natural language processing** (e.g., chatbots, translation)
- **Speech recognition**
- **Autonomous vehicles**
- **Medical diagnosis**

Neural Network

A **neural network** is a computational model inspired by the human brain. It consists of layers of interconnected nodes (also called neurons) that process data by assigning weights and applying activation functions to learn patterns and make decisions.

Key Components of a Neural Network:

- **Input Layer**: Takes in the raw data.
- **Hidden Layers**: Perform computations through weighted connections and activation functions.

- **Output Layer**: Produces the final prediction or classification.
- **Weights & Biases**: Learnable parameters that adjust during training to minimize error.
- **Activation Functions**: Add non-linearity, helping the network learn complex patterns (e.g., ReLU, Sigmoid, Tanh).

Types of Neural Networks:

1. **Feedforward Neural Network (FNN)**
    - Data flows in one direction (input → hidden → output).
    - Commonly used for basic classification and regression tasks.
2. **Convolutional Neural Network (CNN)**
    - Specialized for processing grid-like data such as images.
    - Uses convolutional layers to detect patterns like edges and textures.
3. **Recurrent Neural Network (RNN)**
    - Designed for sequential data like time series or text.
    - Has loops allowing information to persist (good for memory).
    - Variants include LSTM and GRU for better long-term memory.
4. **Generative Adversarial Network (GAN)**
    - Composed of two networks: a generator and a discriminator.
    - Used to generate new data similar to training data (e.g., deepfakes).
5. **Radial Basis Function Network (RBFN)**
    - Uses radial basis functions as activation functions.
    - Typically used for function approximation and classification.
6. **Modular Neural Network (MNN)**
    - Consists of independent networks (modules) that perform sub-tasks.
    - Increases efficiency and scalability.
7. **Transformer Networks**
    - Use self-attention mechanisms instead of recurrence.

- Power models like GPT, BERT for natural language tasks.

CNN in simple words

A **CNN** is a type of deep learning model designed to process **image data**. It works by using layers of filters (called **convolutional layers**) to automatically detect features in an image—like **edges, textures, or shapes**. These features are then combined in deeper layers to recognize **more complex patterns**, such as objects or faces.

Key points:

- **Convolutional layers**: Extract features from the image.
- **Pooling layers**: Reduce the size of the data to make it faster and prevent overfitting.
- **Fully connected layers**: Make final decisions (like classifying an image as a "cat" or "dog").

CNNs are widely used in **image recognition, medical imaging, self-driving cars, and face detection**.

Here's a more focused **project pipeline** for a **Forest Fire Detection** system using a **CNN model**, covering the specific stages you mentioned:

Forest Fire Detection – Project Pipeline

1. Data Collection

- **Sources**:
  - Open datasets (e.g., Kaggle, MODIS, FIRMS).

  Categories:
  - Fire images
  - Smoke images
  - Normal (no fire/smoke) images
- **Format**: JPG/PNG images or video frames extracted to images.

2. Image Processing

- **Resize** all images to a fixed size (e.g., 224×224) for input consistency.
- **Normalize** pixel values to range [0, 1].
- **Convert** to grayscale or RGB depending on model needs.
- **Remove noise** if necessary (optional filtering or blurring).

3. Image Augmentation

- Helps the model generalize better and reduces overfitting.
- Techniques used:
    - Rotation
    - Flipping (horizontal/vertical)
    - Zoom
    - Brightness adjustment
    - Cropping

Tools: Use libraries like **TensorFlow**.

4. CNN Model Used

- **Base model**: Custom CNN or Pre-trained CNN (e.g., **MobileNetV2**, **ResNet50**, or **VGG16**) via **Transfer Learning**.
- **Architecture (example)**:
    - Input layer (224x224x3)
    - Conv2D + ReLU + MaxPooling
    - Multiple convolution + pooling layers
    - Flatten
    - Dense layer

- Output layer (Sigmoid for binary classification: fire vs. no fire)

5. Testing & Evaluation

- **Dataset Split**: Train (70%) / Validation (20%) / Test (10%)
- **Evaluation Metrics**:
  - Accuracy
  - Precision, Recall
  - F1-Score (important for fire detection)
  - Confusion Matrix (to check false positives/negatives)
- **Visual Tools**:
  - Loss and accuracy plots
  - ROC curve (if needed)