HARCOURT BUTLER TECHNICAL UNIVERSITY
Kanpur

Department of Computer Science and Engineering

Session: 2023-2024

Project Report on

Plant Disease Detection

Submitted to:

Prof. Vandana Kaushik

Submitted by:

Divyanshi Gupta (200104025)

Himanshu Singh (200104035)

Himanshi Rathore (200104033)

Vanshika Garg (200104074)

(Final Year BTech CSE)

CONTENT:

ABSTRACT:

Deep Learning (DL), a subset of machine learning, has emerged as a transformative technology with profound implications across various domains. At its core, DL involves the training of artificial neural networks with multiple layers (deep neural networks) to automatically learn and extract complex patterns from vast datasets.

The abstract begins by elucidating the historical context and evolution of deep learning, tracing its roots within the broader field of artificial intelligence. The motivation for the widespread adoption of deep learning lies in its unparalleled ability to handle large-scale datasets and solve intricate problems that were once deemed insurmountable.

Fundamental components of deep learning, such as neural networks and their training algorithms, are explored, emphasizing the significance of deep architectures in capturing intricate relationships within data. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Generative Adversarial Networks (GANs) are highlighted as prominent architectures that have fueled breakthroughs in computer vision, natural language processing, and generative modeling.

The abstract then delves into the myriad applications of deep learning across industries. In computer vision, deep learning facilitates tasks like image recognition and object detection, while in natural language processing, it powers advancements in sentiment analysis and language translation. Healthcare has witnessed revolutionary transformations through deep learning, particularly in medical imaging analysis and personalized medicine. The financial sector leverages deep learning for fraud detection, risk assessment, and algorithmic trading.

Objective of Project :

- To detect unhealthy regions of plant leaves.
- Classification of plant leaf diseases using texture features.
- To analyze the leaf infection.
- To give remedy information to the user.
- To make this services available on Mobile App which can run on low level configuration devices.

INTRODUCTION:

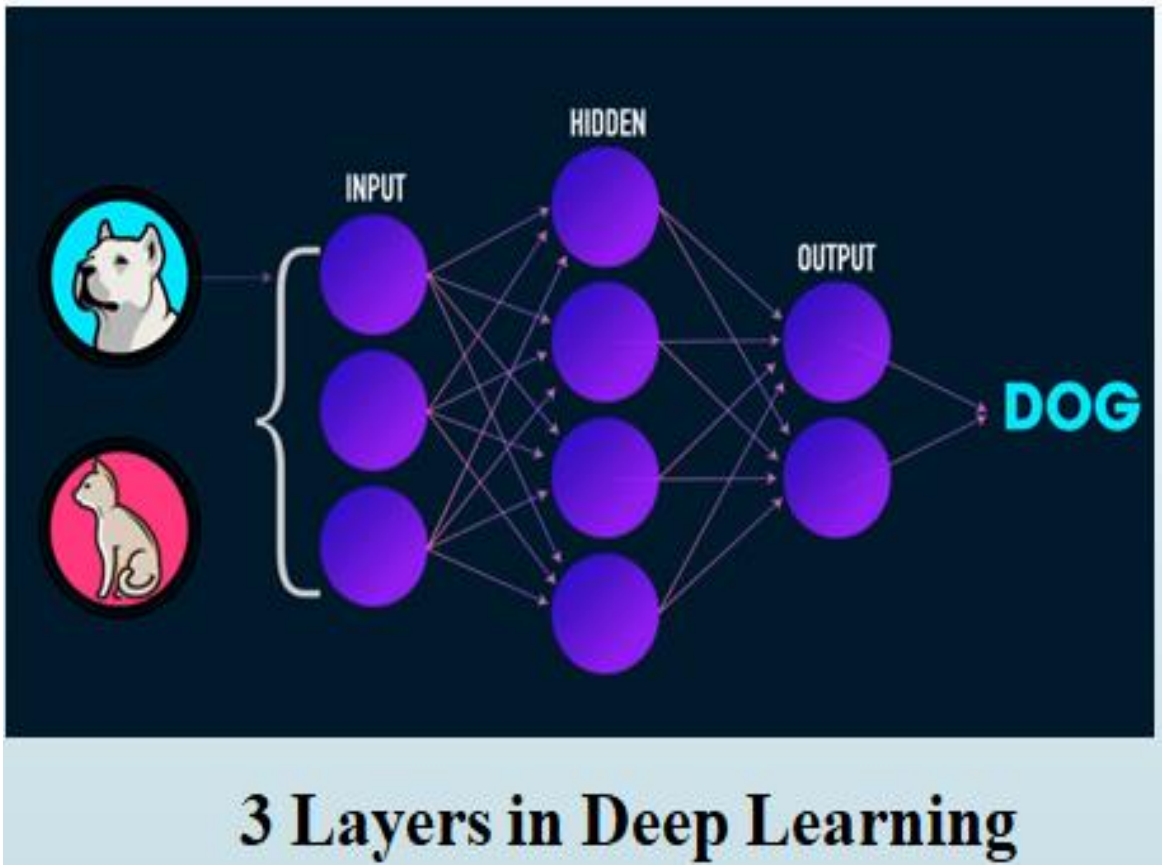
Deep Learning (DL) is a subset of machine learning that involves training artificial neural networks to perform tasks without explicit programming. It is inspired by the structure and function of the human brain, where interconnected neurons work together to process information. The term "deep" in deep learning refers to the use of deep neural networks, which have multiple layers (deep architectures) that enable the model to learn intricate features and representations from data.

Key Concepts of Deep Learning:

1. **Neural Networks:**
    - The foundational unit of deep learning is the artificial neural network, which is composed of nodes (neurons) arranged in layers. These layers include an input layer, one or more hidden layers, and an output layer.
    - Neurons in each layer are connected by weights, and these weights are adjusted during the training process to optimize the model's performance.
2. **Deep Neural Networks (DNNs):**
    - Deep learning emphasizes the use of deep neural networks with multiple hidden layers. These deep architectures enable the model to learn hierarchical representations of data, capturing complex patterns and features.
3. **Training Process:**
    - During training, the neural network is fed with labeled data, and the weights of the connections between neurons are iteratively adjusted to minimize the difference between the predicted outputs and the actual labels.

- Back propagation is a key algorithm used in this process, propagating the error backward through the network to update the weights.

4. **Activation Functions:**
    - Activation functions introduce non-linearities to the model, allowing it to learn and represent complex relationships in the data. Common activation functions include sigmoid, tanh, and rectified linear unit (ReLU).
5. **Architectures:**
    - Various architectures have been developed for different types of data and tasks. Convolutional Neural Networks (CNNs) are effective for image-related tasks, Recurrent Neural Networks (RNNs) for sequential data, and Generative Adversarial Networks (GANs) for generative tasks.



Three Layers in Deep Learning:

1. **Input Layer:**
    - The input layer is the first layer of the neural network and is responsible for receiving the initial raw input data. Each node in the input layer represents a feature or attribute of the input data. For example, in an image recognition task, each node in the input layer might represent the intensity of a pixel.
2. **Hidden Layers:**
    - Hidden layers are intermediate layers between the input and output layers. These layers process the input data through weighted connections and activation functions, allowing the network to learn

complex representations and patterns. Deep learning models often have multiple hidden layers, distinguishing them from shallow networks with fewer hidden layers.

- The number of nodes in each hidden layer and the architecture of these layers (e.g., fully connected, convolutional, recurrent) depend on the specific neural network architecture chosen for the task.

3. **Output Layer:**

    - The output layer is the final layer of the neural network, producing the model's predictions or classifications. The number of nodes in the output layer corresponds to the number of classes in a classification task. For example, in binary classification, there might be one output node representing the probability of belonging to one class, while the other node represents the probability of belonging to the second class.

    - The activation function used in the output layer depends on the nature of the task. For binary classification, a sigmoid activation function is commonly used, while softmax is used for multi-class classification.

These three layers collectively form the basic structure of a neural network. The deep learning models with more hidden layers are capable of learning hierarchical representations of the input data, capturing intricate features and patterns. The depth of the network (i.e., the number of hidden layers) is a key factor in the success of deep learning models in handling complex tasks and large datasets.

Why we need Deep Learning:

In the era of unprecedented data volumes and complexity, the demand for sophisticated solutions to tackle intricate problems has led to the emergence of deep learning. Deep learning, a subset of machine learning, has become imperative for several reasons, transforming the landscape of artificial intelligence and data analytics.

- **Handling Complexity and Big Data:**
  Traditional machine learning approaches face limitations in handling vast datasets and capturing intricate patterns within them. Deep learning, with its deep neural networks, excels at automatically learning hierarchical representations from complex data. This depth allows deep learning models to discern intricate features, making them highly effective in tasks such as image recognition, natural language processing, and complex decision-making.

- **Feature Extraction and Abstraction:**
  Deep learning architectures, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), are designed to automatically extract relevant features and temporal dependencies from data. This capability is crucial in applications such as computer vision, where CNNs excel in image feature extraction, and in natural language processing, where RNNs capture sequential dependencies in text.

- **Unleashing the Power of Neural Networks:**
  The use of deep neural networks enables the modeling of complex relationships and non-linearities within data. As opposed to shallow networks, deep architectures allow for more expressive representations, empowering models to grasp intricate nuances in diverse datasets. This depth is particularly beneficial in capturing abstract representations, making deep learning ideal for tasks that demand a high level of understanding and abstraction.

- **Applications Across Industries:**
  Deep learning has showcased remarkable success across diverse industries. In healthcare, it aids in medical image analysis, disease diagnosis, and drug discovery. In finance, it powers fraud detection, risk assessment, and algorithmic trading. In autonomous systems, such as self-driving cars, deep learning enables perception and decision-making based on complex environmental cues.

- **Continual Advancements and Adaptability:**
  The field of deep learning continues to evolve, with continual advancements in model architectures, optimization techniques, and hardware capabilities. This adaptability ensures that deep learning remains at the forefront of innovation, addressing emerging challenges and pioneering solutions in areas such as natural language understanding, generative modeling, and reinforcement learning.

In a data-driven world where complexity abounds, the need for deep learning is evident. Its capacity to handle intricate patterns, extract meaningful features, and provide adaptable solutions positions it as a cornerstone in the advancement of artificial intelligence. From revolutionizing healthcare to enhancing autonomous systems, deep learning stands as a transformative force, unlocking new possibilities and reshaping the way we approach complex problems.

Requirements of Deep Learning:

Deep learning, a subfield of machine learning, has gained prominence for its ability to automatically learn and extract intricate patterns from data. To unleash the full potential of deep learning, several key requirements must be met, encompassing both hardware and software aspects.

Hardware Requirements:

1. **Processing Power:**

    - Deep learning models, especially large neural networks, demand substantial processing power. Graphics Processing Units (GPUs) or specialized hardware like Tensor Processing Units (TPUs) accelerate the training of deep neural networks, enabling faster computations and model optimization.

2. **Memory Capacity:**

    - The expansive datasets processed by deep learning models necessitate high memory capacity. Sufficient Random Access Memory (RAM) is crucial to store and manipulate large matrices during training and inference.

3. **Parallel Processing:**

    - Deep learning tasks involve numerous parallelizable computations. Hardware architectures that support parallel processing, such as GPUs, enhance the efficiency of training deep neural networks by performing multiple operations simultaneously.

Software and Framework Requirements:

1. **Deep Learning Frameworks:**

    - Utilizing dedicated deep learning frameworks, such as TensorFlow, PyTorch, or Keras, streamlines the implementation of complex neural network architectures. These frameworks provide high-level abstractions, making it easier to design, train, and evaluate deep learning models.

2. **Optimization Algorithms:**

    - Efficient optimization algorithms, like stochastic gradient descent (SGD) and its variants (e.g., Adam, RMSprop), are essential for minimizing the loss function during training. These algorithms enable the adjustment of model parameters to enhance predictive accuracy.

3. **Data Preprocessing Tools:**

    - High-quality input data is fundamental to the success of deep learning models. Data preprocessing tools and techniques, including normalization, augmentation, and feature scaling, ensure that the input data is well-suited for training robust models.

Skills and Expertise:

1. **Domain Knowledge:**

    - A deep understanding of the problem domain is crucial for selecting appropriate architectures, preprocessing techniques, and interpreting model outputs. Domain expertise enhances the ability to tailor deep learning solutions to specific industry challenges.

2. **Programming Proficiency:**

    - Proficiency in programming languages commonly used in deep learning, such as Python, and familiarity with associated libraries and frameworks are prerequisites for effective model development, training, and deployment.

3. **Continuous Learning:**

    - The field of deep learning evolves rapidly. Staying updated on the latest advancements, attending conferences, participating in online communities, and engaging in continuous learning are essential to harness the full potential of deep learning technologies.

How Deep Learning works:

Deep Learning (DL) is a powerful subset of machine learning that emulates the human brain's neural network structure. It involves the training of deep neural networks with multiple layers to automatically learn and extract intricate patterns from data. The following paragraphs provide an overview of the working of deep learning, accompanied by a diagram illustrating its key components.

- **Neural Network Architecture:**
  At the heart of deep learning is the artificial neural network, mimicking the interconnected neurons in the human brain. A typical neural network consists of layers: an input layer, one or more hidden layers, and an output layer. Each layer contains nodes (neurons) connected by weights, and these connections undergo adjustments during the training process.

- **Training Process:**
  The training process begins with the presentation of labeled data to the neural network. The input layer receives the raw data, which is then propagated through the hidden layers, with each layer applying weighted connections and activation functions. The output layer produces predictions or classifications. During training, the model's performance is evaluated by comparing its predictions to the actual labels, and adjustments are made to the weights using optimization algorithms like backpropagation.

- **Activation Functions:**
  Activation functions, such as sigmoid or rectified linear unit (ReLU), introduce non-linearities to the model, allowing it to capture complex relationships in the data. These functions determine whether a neuron should be activated based on the weighted sum of its inputs.

- **Backpropagation:**
  The backpropagation algorithm is a key component of deep learning training. It involves the iterative adjustment of weights in reverse order, starting from the output layer and propagating back through the network. This process minimizes the difference between predicted and actual outcomes, optimizing the model for better performance.

- **Deep Architectures:**

The term "deep" in deep learning refers to the use of multiple hidden layers in neural networks. Deep architectures enable the model to learn hierarchical representations of data, capturing intricate features and patterns. This depth enhances the model's ability to understand and generalize complex relationships.

- **Applications:**
  Deep learning finds applications across diverse domains, including computer vision, natural language processing, healthcare, finance, and autonomous systems. The ability to automatically learn and extract features makes deep learning well-suited for tasks such as image recognition, sentiment analysis, medical image analysis, fraud detection, and autonomous vehicle navigation.

Unique Characteristics of Deep Learning:

Deep Learning (DL) stands out in the realm of artificial intelligence and machine learning due to its unique characteristics that distinguish it from traditional approaches.

- **Automatic Feature Extraction:**
  Automatic feature extraction is a key characteristic of deep learning, where the model autonomously identifies and learns relevant features from raw input data during the training process. Unlike traditional machine learning approaches that often require manual feature engineering, deep learning models automatically discern and utilize essential patterns, eliminating the need for explicit feature selection.

- **Makes use of High-dimensional data for training:**
  Deep learning excels in handling high-dimensional data, such as images, text, and audio. The inherent depth of neural networks allows them to process and learn from intricate patterns present in large and complex datasets. This capability makes deep learning particularly

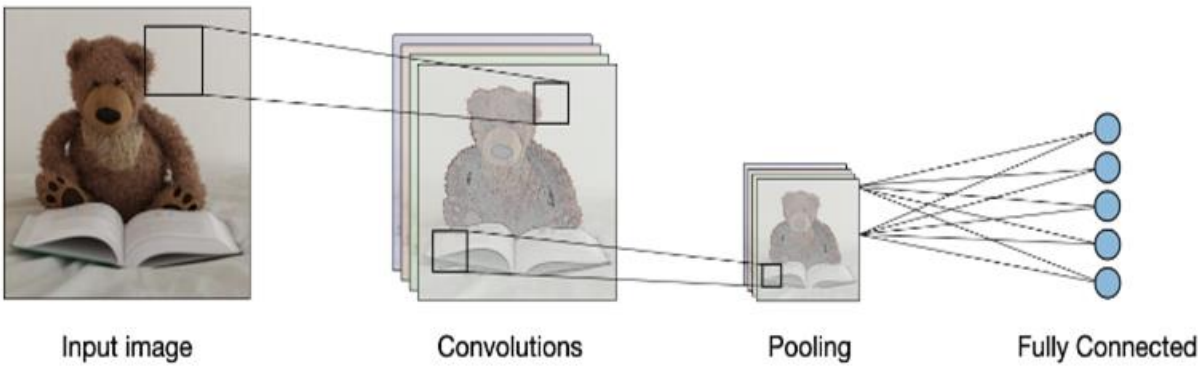
suitable for tasks where the raw input data has a multitude of features or dimensions.

- **Defines higher level features from lower level features obtained from the previous layer:**
  In a deep neural network, the hierarchical architecture allows the model to define higher-level features based on lower-level features obtained from the preceding layer. Each layer captures progressively abstract representations, enabling the network to understand complex relationships within the data. This process of defining hierarchical features contributes to the model's ability to learn intricate and nuanced patterns.

- **It is a representation learning method:**
  Deep learning is a form of representation learning, where the model learns to represent the input data in a hierarchical and abstract manner. Through the layers of the neural network, the model progressively refines its understanding of the data, capturing essential features and relationships. This representation learning is a distinctive aspect of deep learning, allowing it to adapt to diverse and complex datasets.

- **Can solve real-world or unconstrained problems such as NLP, image recognition, etc.:**
  The versatility of deep learning enables it to address real-world or unconstrained problems effectively. In natural language processing (NLP), deep learning models can understand and generate human-like language, while in image recognition, they can automatically identify and classify objects within images. The ability to handle high-dimensional data and automatically extract features makes deep learning applicable to a wide range of domains, from healthcare to finance and autonomous systems.

Data Preprocessing :



Technical Part :

Importing Libraries

In [2]:
```
import tensorflow as tf
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

Training Image Preprocessing ¶

In [5]:
```
training_set = tf.keras.utils.image_dataset_from_directory(
    'train',
    labels="inferred",
    label_mode="categorical",
    class_names=None,
    color_mode="rgb",
    batch_size=32,
    image_size=(128, 128),
    shuffle=True,
```

```
Found 140590 files belonging to 38 classes.
```

Validation Image Preprocessing

In [5]:
```
validation_set = tf.keras.utils.image_dataset_from_directory(
    'valid',
    labels="inferred",
    label_mode="categorical",
    class_names=None,
    color_mode="rgb",
    batch_size=32,
    image_size=(128, 128),
    shuffle=True,
```

Found 35144 files belonging to 38 classes.

In [8]:
```
for x,y in training_set:
    print(x,x.shape)
    print(y,y.shape)
    break
```

```
...
[176.5  168.5  166.5 ]
[168.   160.   158.  ]
[168.75 160.75 158.75]]

[[ 91.5   82.5   89.5 ]
 [156.25 146.25 154.25]
 [160.5  150.5  158.5 ]
 ...
 [174.   166.   164.  ]
 [171.25 163.25 161.25]
 [172.75 164.75 162.75]]]], shape=(32, 128, 128, 3), dtype=float32) (32, 128, 128, 3)
tf.Tensor(
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 1. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]], shape=(32, 38), dtype=float32) (32, 38)
```

To avoid overshooting

1. Choose small learning rate default 0.001 , we are taking 0.0001
2. There may be chance of underfitting, so increase number of neuron
3. Add more Convolution layer to extract more important feature from images, there may be possibility that model unable to capture relevant feature or model is confusing due to lack of feature so feed with more feature

Building Model

In [38]:
```
from tensorflow.keras.layers import Dense,Conv2D,MaxPool2D,Flatten, Dropout
from tensorflow.keras.models import Sequential
```

In [39]:
```
model=Sequential()
```

Compiling Model

In [53]:
```
model.compile(optimizer=tf.keras.optimizers.Adam(
    learning_rate=0.0001),loss='categorical_crossentropy',metrics=['accuracy'])
```

In [54]:
```
model.summary()
```

Model: "sequential_3"

Layer (type)	Output Shape	Param #
conv2d_12 (Conv2D)	(None, 128, 128, 32)	896
conv2d_13 (Conv2D)	(None, 126, 126, 32)	9,248
max_pooling2d_5 (MaxPooling2D)	(None, 63, 63, 32)	0
conv2d_14 (Conv2D)	(None, 63, 63, 64)	18,496
conv2d_15 (Conv2D)	(None, 61, 61, 64)	36,928
max_pooling2d_6 (MaxPooling2D)	(None, 30, 30, 64)	0
conv2d_16 (Conv2D)	(None, 30, 30, 128)	73,856
conv2d_17 (Conv2D)	(None, 28, 28, 128)	147,584
max_pooling2d_7 (MaxPooling2D)	(None, 14, 14, 128)	0
conv2d_18 (Conv2D)	(None, 14, 14, 256)	295,168
conv2d_19 (Conv2D)	(None, 12, 12, 256)	590,080
max_pooling2d_8 (MaxPooling2D)	(None, 6, 6, 256)	0

Building Convolutional Layer

In [41]:
```
model.add(Conv2D(filters=32,kernel_size=3,padding='same',activation='relu',input_shape=[128,128,3]))
model.add(Conv2D(filters=32,kernel_size=3,activation='relu'))
model.add(MaxPool2D(pool_size=2,strides=2))
```

In [42]:
```
model.add(Conv2D(filters=64,kernel_size=3,padding='same',activation='relu'))
model.add(Conv2D(filters=64,kernel_size=3,activation='relu'))
model.add(MaxPool2D(pool_size=2,strides=2))
```

In [43]:
```
model.add(Conv2D(filters=128,kernel_size=3,padding='same',activation='relu'))
model.add(Conv2D(filters=128,kernel_size=3,activation='relu'))
model.add(MaxPool2D(pool_size=2,strides=2))
```

In [44]:
```
model.add(Conv2D(filters=256,kernel_size=3,padding='same',activation='relu'))
model.add(Conv2D(filters=256,kernel_size=3,activation='relu'))
model.add(MaxPool2D(pool_size=2,strides=2))
```

In [45]:
```
model.add(Conv2D(filters=512,kernel_size=3,padding='same',activation='relu'))
model.add(Conv2D(filters=512,kernel_size=3,activation='relu'))
model.add(MaxPool2D(pool_size=2,strides=2))
```

In [47]:
```
model.add(Dropout(0.25)) #To avoid overfitting
```

In [48]:
```
model.add(Flatten())
```

In [49]:
```
model.add(Dense(units=1500,activation='relu'))
```

In [52]:
```
## Output Layer
model.add(Dense(units=38,activation='softmax'))
```

Software and Hardware Requirements:

- TensorFlow
- Keras
- Windows10

- 4GB of RAM for running Android Studio
- 2GB of Disk space for development
- Mobile with minimum of 2GB RAM

Conclusion and Future-work :

- The use of automated monitoring and management systems are gaining increasing demand with technological advancement.
- In the agricultural field loss of yield mainly occurs due to widespread disease.
- Mostly the detection and identification of the disease is noticed when the disease advances to severe stage therefore, causing the loss in terms of yield, time and money.
- The proposed system is capable of detecting the disease at the earlier stage as soon as it occurs on the leaf, Hence saving the loss and reducing the dependency on the expert to a certain extent is possible.
- It can provide the help for a person having less knowledge about the disease, Depending on these goals, we have to extract the features corresponding to the disease.