PLANT DISEASE DETECTION & RECOMMENDATION SYSTEM

(7. Deployment – Part 2)

1) *DEFINE SCOPE*
2) *COLLECT DATA*
3) *PREPROCESS DATA*
4) *CHOOSE MODEL*
5) *MODEL TRAINING*
6) *EVALUATE THE MODEL*
7) *DEPLOYMENT*
8) *FARMER USUABILITY*
9) *GATHER FEEDBACK*

1. What is tensorflow lite?

- TensorFlow Lite (TFLite) is a collection of tools to convert and optimize TensorFlow models to run on mobile and edge devices.
- TensorFlow Lite is an open-source deep learning framework designed for on-device inference (Edge Computing)

2. What is tensorflow?

- TensorFlow is an open-source software library for AI and machine learning with deep neural networks.
- TensorFlow for computer vision was developed by Google Brain for internal use at Google and open-sourced in 2015.
- Today, it is used for both research and production at Google.

3. Tensorflow vs Tensorflow lite

- TensorFlow: The full-fledged, general-purpose machine learning framework. It's used for training models, building complex architectures, and research. It's designed to run on powerful machines (servers, GPUs).
- TensorFlow Lite: A lightweight version of TensorFlow designed specifically for deploying and running models on resource-constrained devices like smartphones, embedded systems, and IoT devices. It's focused on inference (making predictions with a pre-trained model), not training.
- So, TensorFlow Lite doesn't contain the entire TensorFlow library. It contains a smaller, optimized set of components that are necessary for running pre-trained TensorFlow models on mobile and embedded devices.

4. What is Edge Computing?

- Edge computing is a new paradigm that performs computing at the edge of the network.
- As opposed to cloud computing, edge computing moves closer to the user and closer to the source of data.
- At the edge of the network, it is lightweight for local, small-scale data storage and processing.

5. What is paradigm?

- A paradigm is a model or pattern for something that may be copied.
- a theory or a group of ideas about how something should be done, made, or thought about.

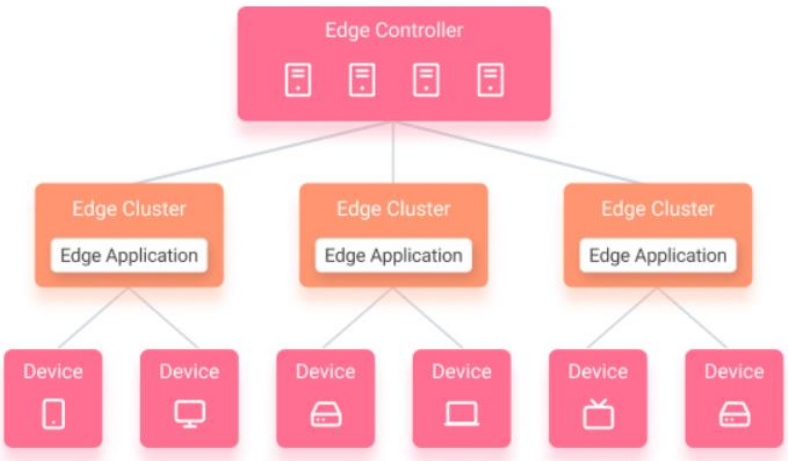
6. Why to use Edge Computing?

- The number of smart devices connected to the Internet is growing fast, generating large-scale data (Big Data) at the network edge.
- The massive amount of data causes problems such as bandwidth load, slow response speed, poor security, and poor privacy in traditional cloud computing models.
- Hence, traditional cloud computing is no longer sufficient to support today's requirements for smart services and intelligent data processing, so edge computing technologies have emerged.

7. What are the characteristics of Edge computing?

- Edge computing is characterized in terms of **high bandwidth**, **ultra-low latency**, and **real-time access** to network information that can be used by several applications.

8. Concept of Edge Computing and Edge Devices



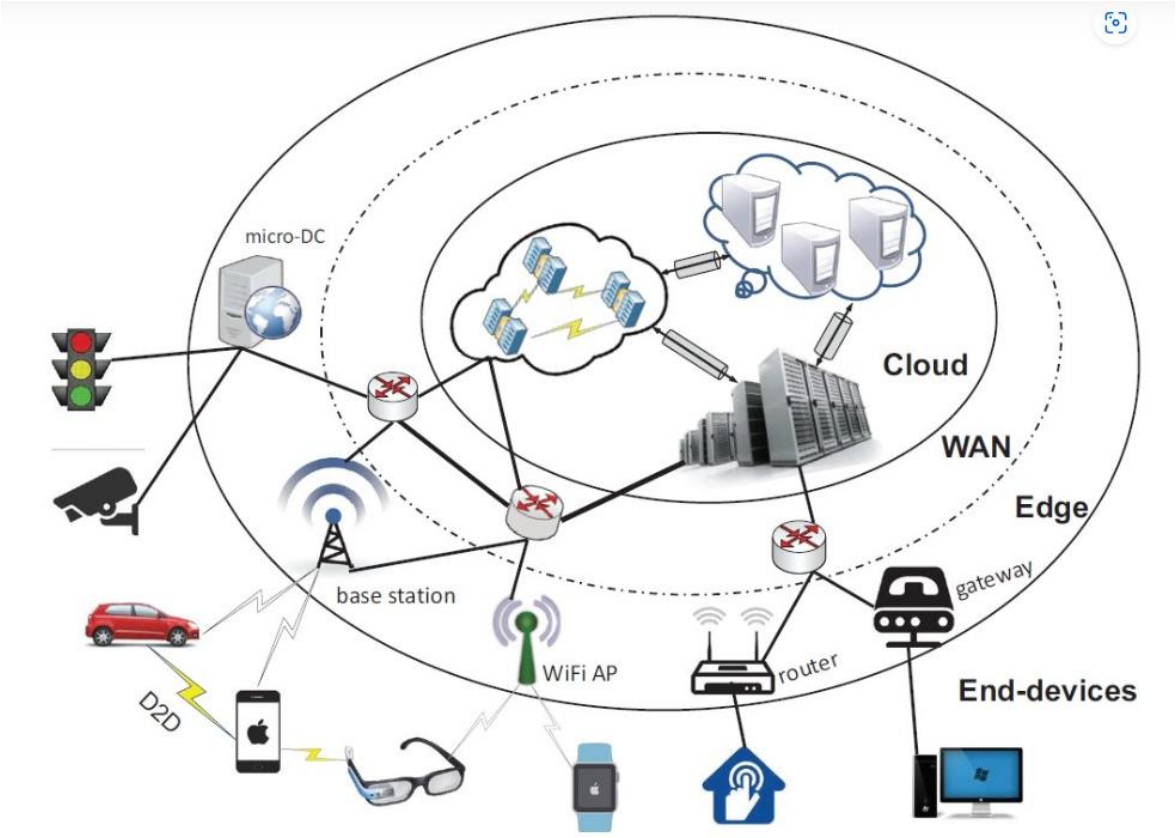
9. What are the Edge devices?

- Edge devices have spread to all aspects of society, such as smart homes, surveillance cameras, autonomous vehicles, intelligent production robots, and more.

10. What is the need for edge computing?

Based on the continuous and massive growth of the volume of data and various data processing requirements, cloud-based big data processing has shown many shortcomings:

- Real-time: The transmission of large amounts of data to the cloud results in a large load of network transmission bandwidth, resulting in data transmission delay. Cloud computing will not be able to meet real-time business requirements.
- Energy consumption: The computational power of central data centres has increased significantly requiring more energy and resources. Cloud computing is not able to meet the increasing demand for optimized energy consumption.
- Security and privacy: Uploading the data to the cloud and storing them in a centralized environment comes with privacy leakage risks or attacks. The probability of data en-route attacks is higher in cloud computing than in edge computing caused by the longer path to the server.

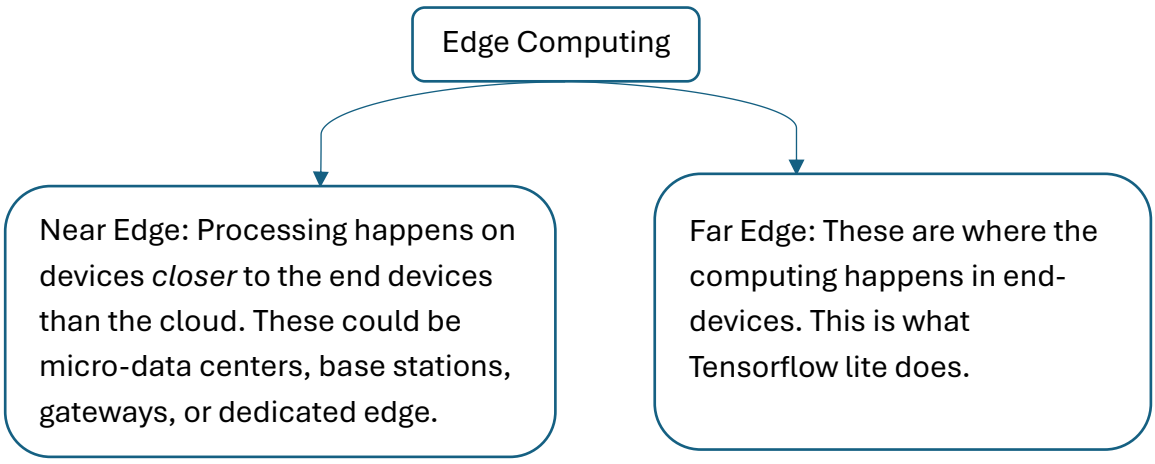


Connectivity and the need for edge computing

References:

- TensorFlow Lite - Computer Vision on Edge Devices [2024 Guide] - viso.ai
- What Is Edge Computing? A Practical Overview - viso.ai
- Exploring the Main Types of Edge Computing | Databank
- Far Edge vs. Near Edge in Edge Computing - Tech | 5G, SDN/NFV & Edge Compute
- A Step-by-Step Guide to Convert Keras Model to TensorFlow Lite (tflite) Model | by Khorshed Alam | Medium

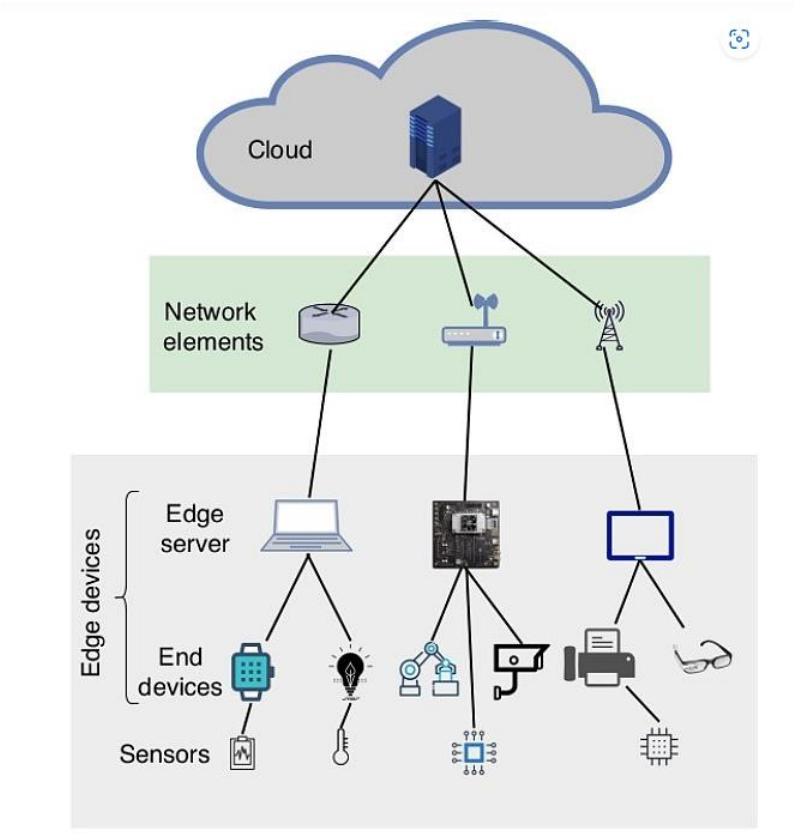
11. What are the types of Edge Computing defined by distance?



12. Which is suitable for our project, Far edge or Near edge?

Far Edge (On-Device - TensorFlow Lite)	**Near Edge (Edge Server)**
Advantages: • True offline functionality (no communication needed). • Lower latency (faster predictions). • Privacy (data stays on the device).	**Advantages:** • Less resource constraint on the device. • Easier model updates. • Can handle more complex processing.
Disadvantages: • Resource constraints (model size, processing power). • Model updates can be more complex.	**Disadvantages:** • Requires some level of connectivity (even if offline from the cloud). • Higher latency than on-device processing. • Privacy considerations (data is sent to the edge server).

13. Architecture of Edge Computing:



General overview of the edge computing architecture

14. Details of Architecture of Edge Computing.

15. What is edge controller and edge cluster?

16. Framework Vs Library Vs Module

- **Library:** A collection of pre-written code (functions, classes, etc.) that you can reuse in your programs. Libraries provide functionality for specific tasks (e.g., NumPy, Matplotlib). You import a library to access and use its functions.
- **Framework:** A more comprehensive and structured collection of code that provides a foundation or template for building applications. Frameworks often have a specific architecture or set of rules that you need to follow. They provide a high-level structure and take care of many underlying details. TensorFlow (the full version) is a machine learning framework.
- **Module:** A self-contained unit of code within a library or framework. Modules organize related functions and classes into separate files or namespaces. For example, TensorFlow has different modules (e.g., tf.keras for building neural networks, tf.lite for TensorFlow Lite conversion).

Analogy:

- **Library:** Like a toolbox containing individual tools (functions) that you can use.
- **Framework:** Like a workshop with a predefined layout and tools that are organized for specific tasks. It provides a structure and guidance for building things.
- **Module:** Like a drawer within the toolbox that contains a set of related tools.

17. Data Curation for Generating a TensorFlow Lite Model:

- Tensorflow Lite models are represented with the .tflite file extension, which is an extension specifically for special efficient portable formats called FlatBuffers.
- FlatBuffers is an efficient cross-platform serialization library for various programming languages and allows access to serialized data without parsing or unpacking.
- This methodology allows for a few key advantages over the TensorFlow protocol buffer model format.
- Advantages of using FlatBuffers
    - reduced size and faster inference,
    - use minimal compute and memory resources.
    - Can add metadata with human-readable model descriptions + m/c readable data, to enable the automatic generation of pre-processing & post-processing pipelines during on-device inference.

18. Generation of TensorFlow Lite Model:

- Use an existing TensorFlow Lite Model.
- Create a TensorFlow Lite Model using TensorFlow Lite Model Maker.
- Convert a TensorFlow Model into TensorFlow Lite Model.

19. What are the quantization techniques?