**CHAPTER 4**

**SYSTEM DESIGN**

**System design** is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. The purpose of the System Design process is to provide sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture.

**4.1 Architecture**

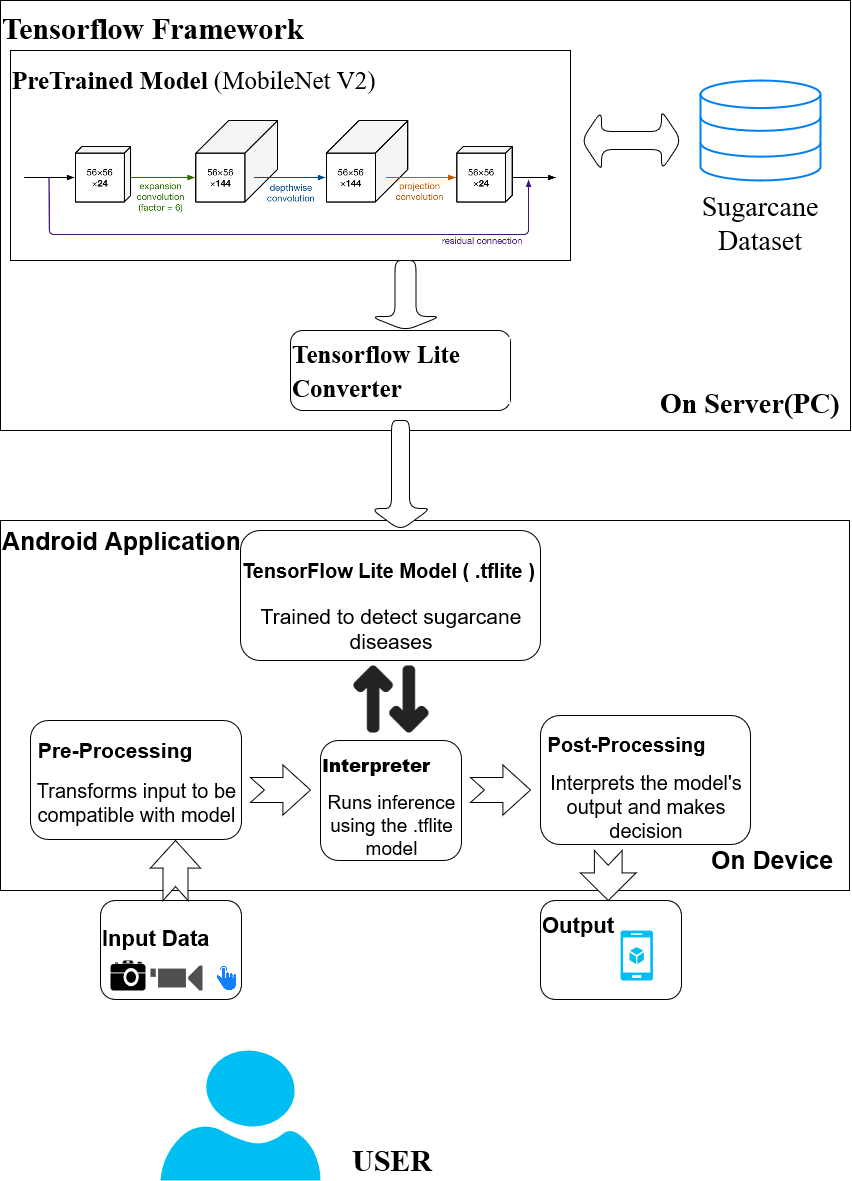


FIGURE 4.1: SYSTEM ARCHITECTURE OF PROPOSED SYSTEM.

The components of the system architecture are explained below:

1. **USER:** The user here is the client who would be using the android application. The user needs to upload an image of the region to be inspected for diseases in a sugarcane crop. On which the inference will be returned to the user on device.
2. **Android Application:** It consists of the TensorFlow Lite model which will perform on device inference by invoking the TensorFlow Lite Interpreter. It also pre-processes the input data to be compatible with the .tflite model and post-processes the output result and displays the final Result to the user.
3. **TensorFlow Framework:** A pre-trained model( here, **MobileNet V2)** is a saved network that was previously trained on a large dataset, typically on a large-scale image-classification task. This pre-trained model is the base model on which the Sugarcane dataset is trained to obtain a custom model. This is a resource intensive process that occurs in the Server machine and is known as Transfer Learning. The Custom CNN model is saved and converted to .tflite format by TensorFlow Lite Converter.

## **4.2 Major Algorithm**

# MobileNetV2 Model

MobileNetV2 is a neural network model developed at Google, and pre-trained on the ImageNet dataset, a large dataset of 1.4M images and 1000 classes of web images. MobileNets are small, low-latency, low-power models parameterised to meet the resource constraints of a variety of use cases that is optimised for mobile devices.

MobileNetV2 improves the state-of-the-art performance of mobile models on multiple tasks and benchmarks as well as across a spectrum of different model sizes. It is a very effective feature extractor for object detection and segmentation. For instance, for detection, when paired with Single Shot Detector Lite, MobileNetV2 is about 35 percent faster with the same accuracy than MobileNetV1.

It builds upon the idea of using depth-wise separable convolutions as efficient building blocks. MobileNet has two new features:

* Linear bottlenecks between the layers: Experimental evidence suggests that using linear layers is crucial as it prevents nonlinearities from destroying too much information. Using non-linear layers in bottlenecks indeed hurts the performance by several percent, further validating our hypothesis
* Shortcut connections between the bottlenecks

### The Basic Structure of MobileNetV2

The bottlenecks of the MobileNetV2 encode the intermediate inputs and outputs while the inner layer encapsulates the model’s ability to transform from lower-level concepts such as pixels to higher level descriptors such as image categories. With traditional residual connections, shortcuts enable faster training and better accuracy.

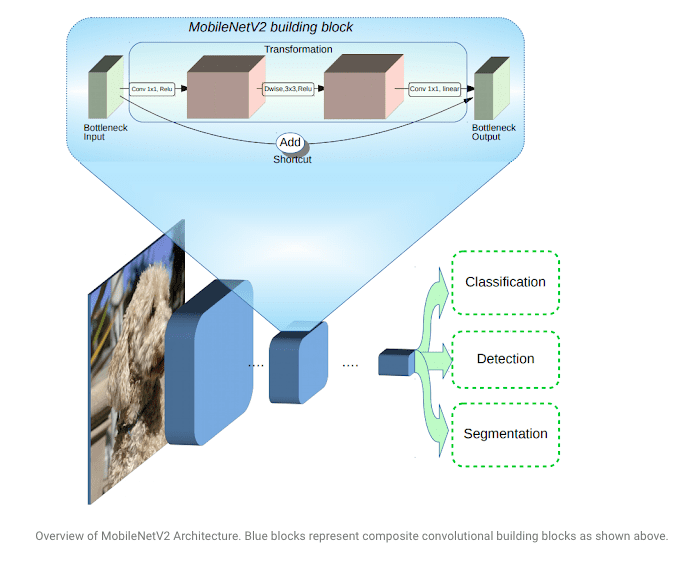


Fig: MobileNet V2 Architecture

The basic building block is a bottleneck depth-separable convolution with residuals. The architecture of MobileNetV2 contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers. The researchers have tailored the architecture to different performance points, by using the input image resolution and width multiplier as tunable hyperparameters, that can be adjusted depending on desired accuracy or performance trade-offs. The primary network  (width multiplier 1, 224 × 224), has a computational cost of 300 million multiply-adds and uses 3.4 million parameters. The network computational cost ranges from 7 multiply-adds to 585M MAdds, while the model size varies between 1.7M and 6.9M parameters.

MobileNet has several properties that make it suitable for mobile applications and allows very memory-efficient inference and utilises standard operations present in all neural frameworks. Thus MobileNetV2 provides a very efficient mobile-oriented model that can be used as a base for many visual recognition tasks like disease detection in plants.