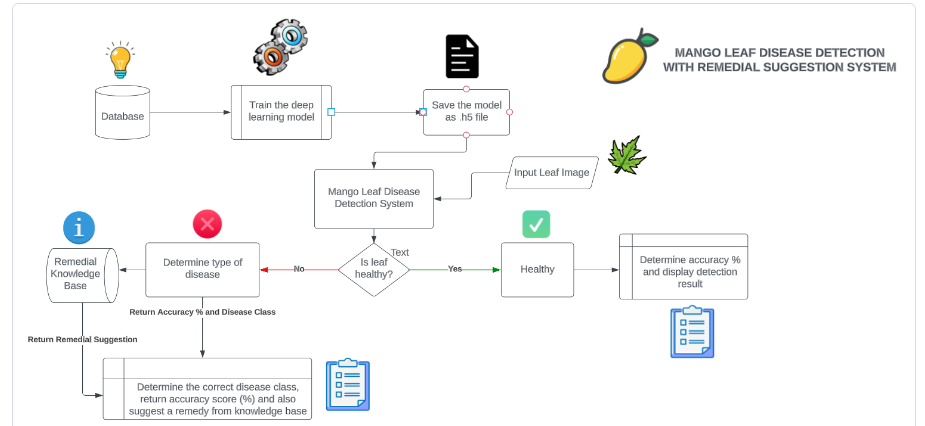
Deep-learning apps for image processing made easy: A step-by-step guide

 The process of creating a custom deep-learning app:

1. Train your image-processing model
2. Develop, test, and save your model
3. Build the app's UI and embedding functions
4. Test the app using image samples
5. Try out the app on your local machine
6. Share your app on GitHub
7. Deploy your app to the Streamlit Community Cloud

**App overview**

The app integrates image capture, preprocessing, ML disease detection models, a disease database, and user-friendly interfaces. It captures input images, processes them using ML models for disease classification, and provides real-time results along with personalized treatment suggestions. And the best part? You can ensure that your model stays accurate and your database is up to date with regular updates.



**Step 1. Train your image-processing model**

Let’s start by training an image processing model. Generate the EfficientNet Deep Learning model as described [here](https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html?ref=blog.streamlit.io) and follow these steps (I used a [Kaggle dataset](https://www.kaggle.com/datasets/aryashah2k/mango-leaf-disease-dataset?ref=blog.streamlit.io)):

* Open a new tab in your favorite browser
* Type "[Colab](https://colab.research.google.com/?ref=blog.streamlit.io)" or "[Google Colab](https://colab.research.google.com/?ref=blog.streamlit.io)" in the search bar to get to Google Colab notebooks
* Create a new Jupyter notebook
* Follow along with [this code](https://github.com/MainakRepositor/MLDP/blob/master/mango-leaf.ipynb?ref=blog.streamlit.io) or upload this notebook to [Colab](https://colab.research.google.com/?ref=blog.streamlit.io)
* Download the [Mango Leaf Disease Dataset](https://www.kaggle.com/datasets/aryashah2k/mango-leaf-disease-dataset?ref=blog.streamlit.io) from Kaggle (to train the ML model)
* Upload this [dataset](https://www.kaggle.com/datasets/aryashah2k/mango-leaf-disease-dataset?ref=blog.streamlit.io) to Google Drive and enable sharing (paste the link in the file access snippet)

**Step 2. Develop, test, and save your model**

With your trained model ready, let’s dive into testing and evaluation. Save your model as a .h5 file, a powerful tool that makes creating the app super easy.

First, set a variable in which you can import the model class. The keyword 'history' is used to record the training metrics while the models perform an epoch. An epoch is an iterative method of training to increase the accuracy and fit best to the extracted set of parameters on the data

**Step 3. Build the app's UI and embedding functions**

This is where the magic happens! Let’s use the power of Streamlit to create an interactive and intuitive app.

**Step 4. Test the app using image samples**

Before sharing your app with the world, make sure everything works. Test it using sample images to see how it performs in real-time (in this case, mango plant leaf images).

**Step 5. Try the app on your local machine**

Before deploying your app, go ahead and test it on your local machine. Just watch this video and follow along.

**Step 6. Share your app on GitHub**

Now, share your app with the community. Push your app's repository to GitHub by following [these steps](https://github.com/MainakRepositor/Git-steps?ref=blog.streamlit.io). When done, your repo should look something like this:

**Step 7. Deploy your app to the Streamlit Community Cloud**

Showcase your app to the world! Simply deploy your repo through the Streamlit Community Cloud (learn how in this [video](https://youtu.be/kXvmqg8hc70?ref=blog.streamlit.io)).

Reference:

<https://blog.streamlit.io/deep-learning-apps-for-image-processing-made-easy-a-step-by-step-guide/>

LeafNet: A proficient convolutional neural network for detecting seven prominent mango leaf diseases

Research document has been attached separately and the URL link for reference is down below:

<https://www.sciencedirect.com/science/article/pii/S2666154323002946>