The dataset if classified into 4 types. Images taken by drones.

Using CNN, it was 82-85% accuracy.

We are using transfer learning.

Keras:-

Transfer learning,

Resnet50 – 25mill paramaters - 🡪 accuracy not so good

Renet152V2 – 60 million parameters. 🡪 acuuracy amazing.

\*\*While testing the app—we should take the images from validation set.\*\*

Got highest accuracy with resnet152V2.

Less params – less time for execution.

In this project we are going to use keras. In keras we choose sequential model. As our dataset is consists of images, it contains layers, so the keras will assume each of the layer as one input type.

Generally, all layers in Keras need to know the shape of their inputs in order to be able to create their weights.

It creates its weights the first time it is called on an input, since the shape of the weights depends on the shape of the inputs.

Naturally, this also applies to Sequential models. When you instantiate a Sequential model without an input shape, it isn't "built": it has no weights (and calling model.weights results in an error stating just this). The weights are created when the model first sees some input data.

Findings in the trials of Android studio implementation

New implmetation

🡪Using react-flask+python ------------- react for front end, python for backend.

🡪created own environment to run flask react app and python backend part.

🡪to activate new env ----- .

First go to env where it is in folder.

python -m venv D:\Mobile\_app\_react\Imp\react-flask-app\fl\_api\ranjith\_env

ranjith\_env\Scripts\activate.bat

ranjith\_env\Scripts\activate