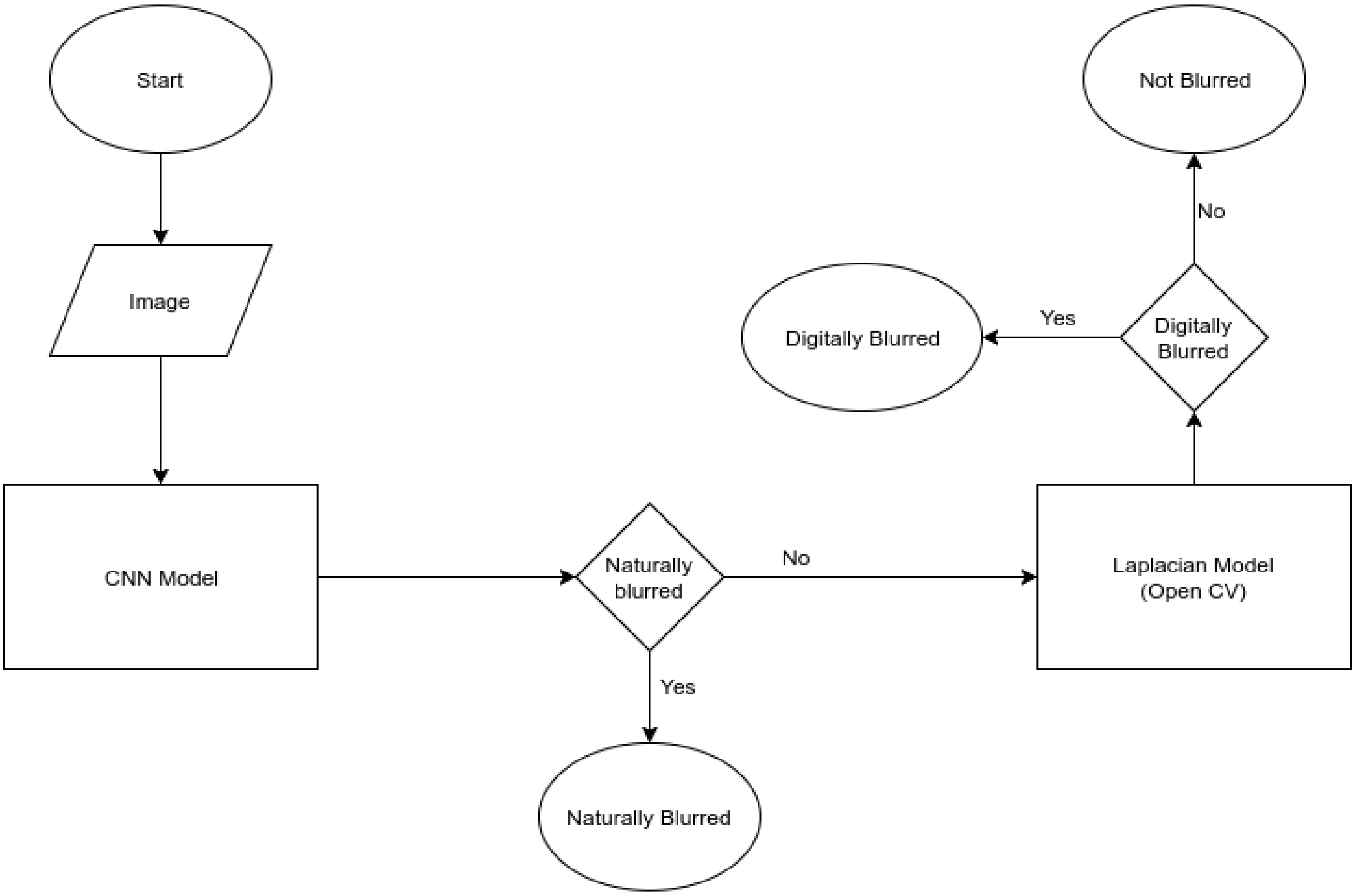
Blurred Image Classifier

Conclusion (Approach)

- Now to Check whether the image is Naturally Blurred or Digitally Blurred
- First feed the image into the CNN Model and check whether the image is Naturally Blurred or not. If it is Naturally Blurred then that image is Naturally Blurred, if it's not then check with OpenCV Model whether the image is Digitally Blurred or not. If it is Digitally Blurred then that image is Digitally Blurred image otherwise the image is not blurred .



To Reach This Conclusion :

Requirements

- numpy
- pandas
- opencv
- Keras

Using OpenCV

Using Laplacian

- the goal of this operators is to measure the amount of edges present in images, through the second derivative or Laplacian
- You simply take a single channel of an image (presumably grayscale) and convolve it with the following 3 x 3 kernel
- And then take the variance (i.e. standard deviation squared) of the response.
- If the variance falls below a pre-defined threshold, then the image is considered blurry; otherwise, the image is not blurry.
- To find Predefined Threshold I am checking what is the best threshold by trying various thresholds and analyzing them. You can find the details below

Refrence - https://www.pyimagesearch.com/2015/09/07/blur-detection-with-opencv/

- By Taking the accuracies of Naturally blurred , Digitally Blurred , Artificially Blurred i have taken a suitable threshold
- Threshold : 100 --> Accuracy : 55 Threshold : 300 --> Accuracy : 80 Threshold : 500 --> Accuracy : 90 Threshold : 700 --> Accuracy : 94 Threshold : 900 --> Accuracy : 95 Threshold : 1100 --> Accuracy : 97
- There is a sudden Boost in Accuracy from 100 to 700 then it becomes somewhat constant . So the suitable threshold will be 700 according to this situation.
- Accuracies in Evaluation Dataset
  - Digitally Blurred Dataset -> 93%
  - Naturally Blurred Dataset -> 41%
- We can say that this model is performing good for Digitally Blurred Dataset but not good for Naturally Blurred Dataset

Using CNN

- Model Used -> LENET (Conv -> MaxPool -> Conv -> MaxPool -> Flatten -> FC -> FC)
- Batch Size -> 32
- Epochs -> 5

Process :

- Making of a Training and Testing Dataset for
  - Naturally Blurred Dataset and undistorted
  - Digitally Blurred Dataset and undistorted
    - Combining All Blurred and undistorted
- Making of LENET Model
- Compiling The Model
- Making Of DataAugmented Data for Improving Accuracy

- Training The Model
- Evalutaiong The Model on Evalutaion Set
- To Improve Accuracy Further :
    - Try on Differet Models -> VGG16, Inception
    - Try some Regularization Techniques(DropOut)
    - Increase The number of Epochs
    - More Data Augmentation
- Accuracies in Evaluation Set
    - Naturally Blured Dataset vs Undistorted -> 61% (Naturally Blurred Evaluation Dataset)
    - Digitally Blured Dataset vs Undistorted -> 18% (Digitally Blurred Dataset)
    - Naturally Blured + Digitally Dataset vs Undistorted ->
        - Naturally Blurred dataset -> 59%
        - Digitally Blurred dataset -> 11%
- This Model is performing well for Naturally Blurred Dataset(Accuracy can be further improved by applying various different models (Inception,VGG16 etc.)) but not good for Digitally Blurred Dataset.

Note - Accuracy can be further enhanced (I can further improve the accuracy by Hyper-Parameters tuning and regularization)

- Laplacian Model is performing well for Digitally Blurred Images
- CNN LENET Model is performing well for Naturally Blurred Images
- More model should be tried with this dataset along with some data augmentation and regularization techniques to improve accuracy in case of CNN Models.