Arihant, AUV IIT Madras

CVISub-Team
Me19b111
Harshitha nugala

1) Apply Image processing on the image to make it easy to identify the number of sheets in the image.

- Firstly, the given image is renamed as ‘sheets.jpg’.
- Now the image is read in the jupyter notebook with the opencv command.
- Then in order to display the image with a proper size with gray scale version a function named display is created which plots the image in (12,10) inches space.
- In order to reduce the noise in the image we need to blur the image.
- Here, I have done the blurring in two different ways: median-blur technique and gaussian-blur technique.
- In median-blur, the inputs are image and kernel of size (7,7).
- In gaussian-blur, the inputs are image, kernel of size(7,7) and sigmaX which represents the gaussian deviation in x-direction.
- Now the processed images are passed into a gradient function.
- As we need the sheets count which are vertically placed, we should use x-gradient for a clear picture of sheets.
- Now the median_blurred image and gaussian_blurred image are sent into sobel command which takes inputs: image,syntax-cv2.CV_64F (64 floating point precision),1(x-gradient),0(y-gradient),kernel of size(5,5).
- The next step is to threshold the image. The command includes input:image,threshold_value,max_value of image,thresholding_type

2) Develop a Deep Learning based image classifier to group/classify images into 6 distinct categories

1. **Drone**
2. **Fighter-jet**
3. **Helicopter**
4. **Missile**
5. **Passenger-plane**
6. **Rocket**

- Given data is in a compressed folder, so we need to extract all and then upload the files to drive with the given code. The folder carries three files: train,test,validation.
- Each file has images of all the six categories.
- With the data we need to do data augmentation by importing ImageDataGenerator from keras.preprocessing.image.
- Data augmentation increases the diversity of data by cropping,padding etc
- Then we import VGG16 from applications in keras. If we look at the summary of the conv_base it is a 3D tensor. So we must convert it into 2D tensor by adding layers.flatten. But before that we have to insert the conv_base into our model.
- After converting it into 1D tensor dense layers are added with activation function and number of neurons.
- Compile the model with loss = categorical cross entropy and optimizer = ‘adam’, metrics=[acc].
- Then we use fit_generator to train the neural network. If there was no data augmentation then we can use fit.
- The plotted graph is between accuracy vs epochs and validation_accuracy vs epochs.