Question 1:

1. Question asked to apply Image processing to given Image to make it easy to identify number of sheets.
2. I started by loading Image and converting it to grey-scale. It is done by using OpenCV function cv2.cvtColor(image, cv2.COLOR_RGB2GRAY). In this function, at pixel values are replaced by taking average of all channels and placing it as the new pixel value I.e. (R+ G + B)/3
3. After gray-scaling image, we do edge detection, for this firstly we apply Gaussian Blur which smoothers the Image and removes noise in it, in which a (3,3) kernel with stride 1 is slides over image, in each slide Gaussian function is applied to the window

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

4. After it, we use popular Canny Edge detection algorithm to detect the edges in image. cv2.Canny(edge_image, 30, 120)

$$Edge\_Gradient\ (G) = \sqrt{G_x^2 + G_y^2}$$

$$Angle\ (\theta) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

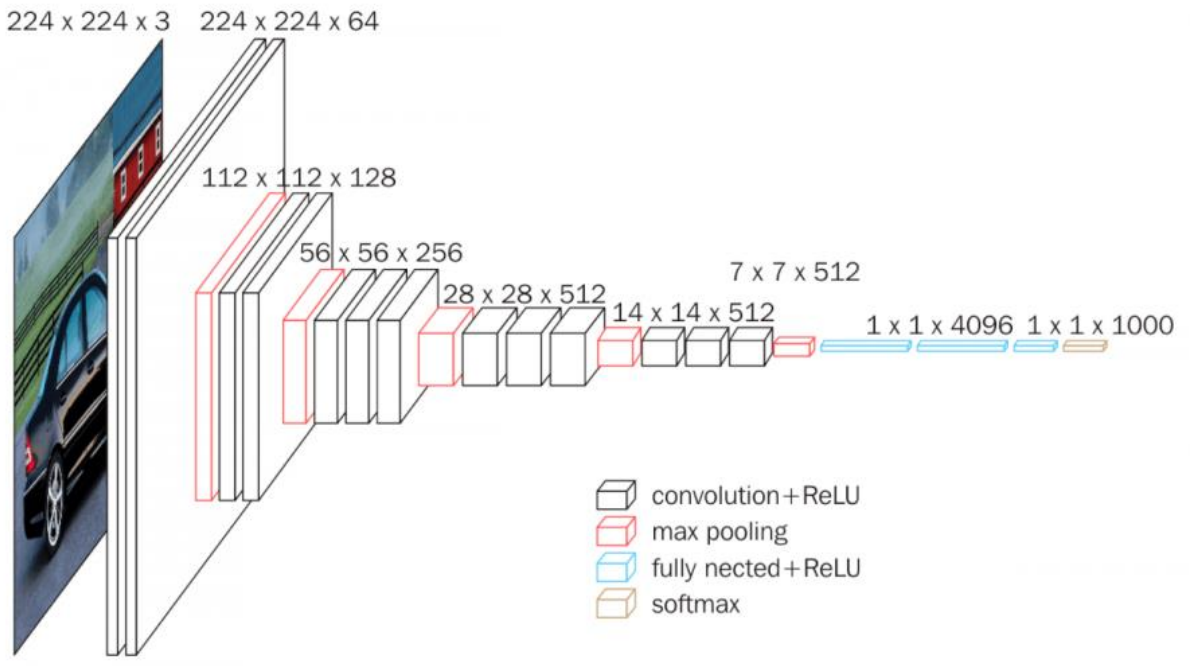
Where G_x and G_y are gradient in X and Y respectively.

5. After it Non-max suppression is applied to the Image to remove pixels which don't contribute to edges and after that edge not withing thresholds are neglected to finally reach edges in Image.
6. After it we apply erosion and dilation to image, in erosion black noise in Image is removed and due to it Images shrinks and after that we apply dilation to compensate for erosion, it involves convolution.

Question 2:

1. In question, it is asked to train a model to detect images.
2. We use pre-trained VGG16 model and apply transfer learning for this question.
3. Firstly, we load all make a helper function which loads image and 3d Array of Image which is resized to (224 x 224 x 3) which is required Input shape for VG16 Algorithm.
4. We load Images and split it into train = 0.9 and test = 0.1 split.
5. Convert Y to one-hot vector version for training.
6. We load VGG16 from keras and replace its last layer with our SoftMax layer with 6 labels.
7. We train model categorical cross entropy loss function and Adam optimizer with accuracy as metrics.
8. Then, I reported Epoch vs accuracy on training data and confusion metrics, as well as classification report on it.

VGG16:



1. As name suggests, VGG16 architecture is a Deep – learning architecture with 16 layers. It consists of 13 convolution layers with ReLU activation function, 2 fully-connected layers at end, and a SoftMax activation layer at end for prediction.
2. It has been observed that as layers gets deep our model starts to learn more complex structure. Ex, in first few layers, it tries to find some pattern, edges, lines, etc. After few layers it tries to find some shapes and further it tries to find complex structures like a cat, or a traffic signal, or wheels of car, etc.
3. For our model, there were 134,285,126 parameters, out of which 119,570,438 in fully-connected layers and SoftMax layer were trained for this task.
4. The accuracy on the test-data was about 60%. It can be further improved by using more training data, training full layers instead of transfer learning or using more powerful architectures like residuals networks, Inception network, etc.

Convolution Layer:

1. Convolution layers slide over image and do edge detection.

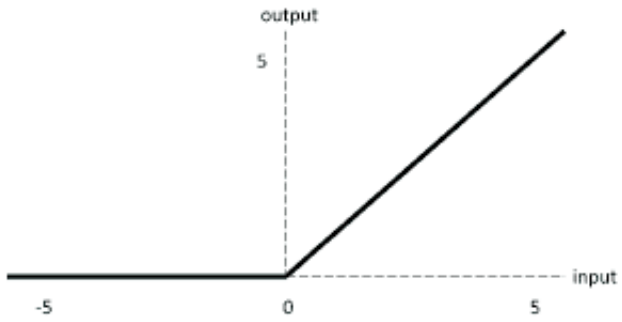
2. For Image of shape h x w x c, after passing through Convolution layer of size f x f x c with padding of p and stride s, resulting Image has shape h' = (h + 2p – f)/s + 1, w' = (w + 2p – f)/s + 1
3. Weights of Convolution layers are initialized randomly and later changed using back-propagation.
4. Convolution layers work better than normal layers on Images, as they have less trainable parameters than fully-connected layer, as they share weights by sliding over Image. They perform better in Image detection task
5. For VGG16, Convolution layer has size (3 x 3) with 'same' padding.

Max – Pooling Layer:

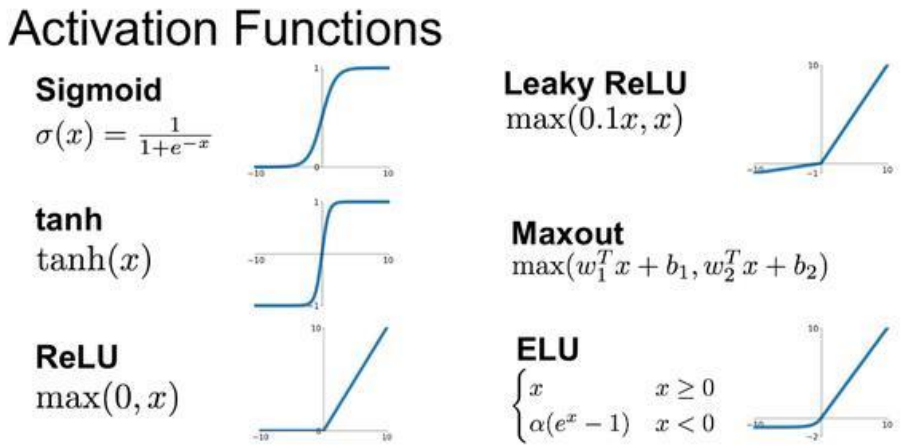
1. Pooling helps to reduce size of Image. In Max-pooling this is done by taking max – number in the given slide. Ex, for a 224 x 224 x 64 layer, after passing through max-pooling window of size of (2 x 2) and stride of 2, it size becomes 128 x 128 x 64.
2. Pooling is used generally after some convolution layers.

ReLU Activation:

1. Activation Functions determine output of layer.



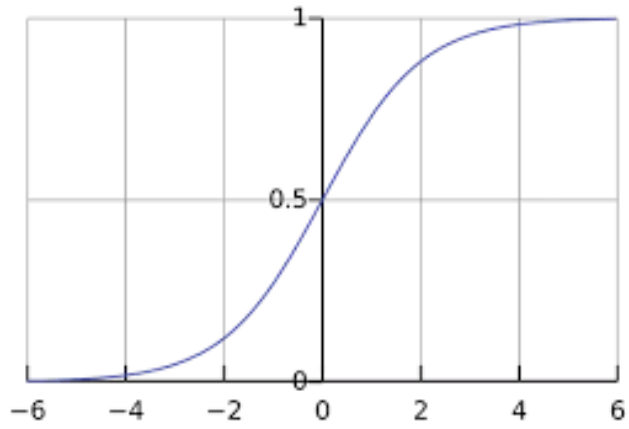
2. Other activation functions include,



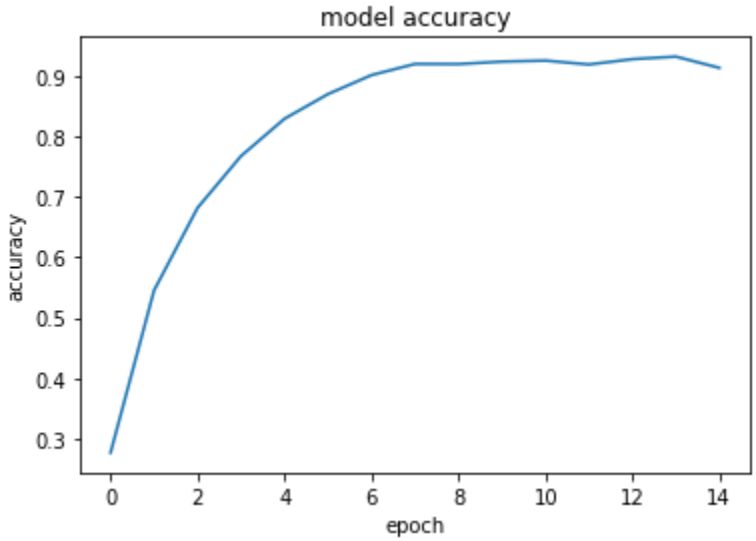
SoftMax Activation:

Softmax

$$f_i(\vec{a}) = \frac{e^{a_i}}{\sum_k e^{a_k}}$$



Results:



Confusion Matrix

[[74 38 2 144 64 7]

[2 333 11 41 32 10]

[0 50 301 17 20 13]

[102 116 13 262 49 22]

[11 56 7 45 278 13]

[2 67 18 39 34 240]]

Classification Report precision recall f1-score support

rocket	0.39	0.22	0.28	329
passener-plane	0.50	0.78	0.61	429
missile	0.86	0.75	0.80	401
helicopter	0.48	0.46	0.47	564
fighter-jet	0.58	0.68	0.63	410
drone	0.79	0.60	0.68	400
accuracy			0.59	2533
macro avg	0.60	0.58	0.58	2533
weighted avg	0.60	0.59	0.58	2533