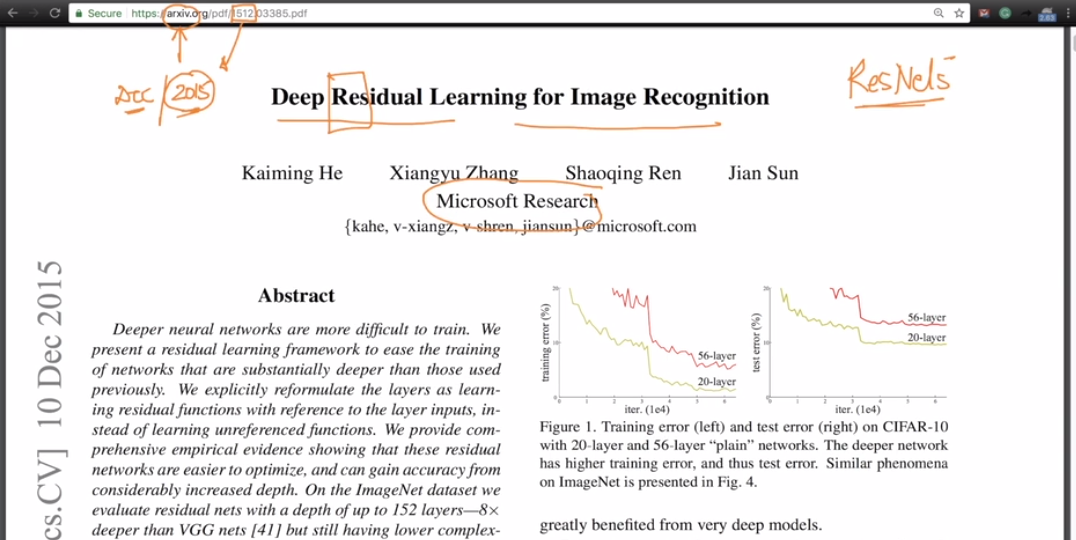
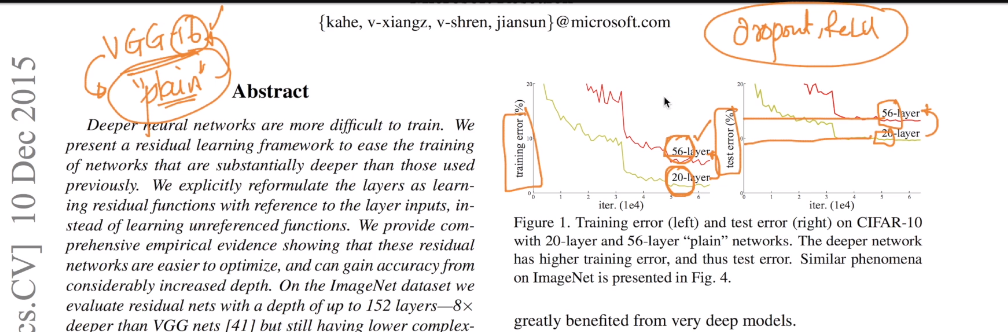
**Residual Network :**



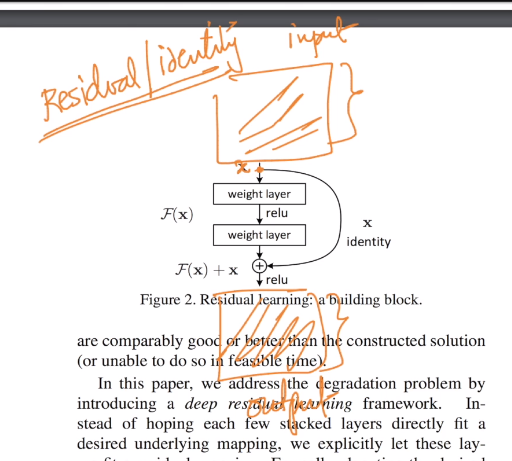
Why Resnet Comes in picture : before this when we use VGG16 plain network then as the number of layers increases its training error become worst as shown in below fig.

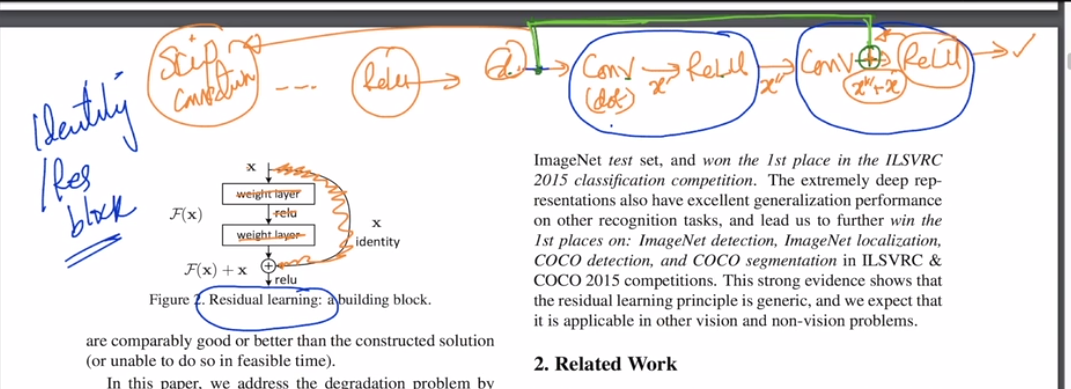
is of plain network here training error of 56 layer is more than 20 layer which shouldn’t be.



So the solution for this is residual network in this we use skip connection as shown below.

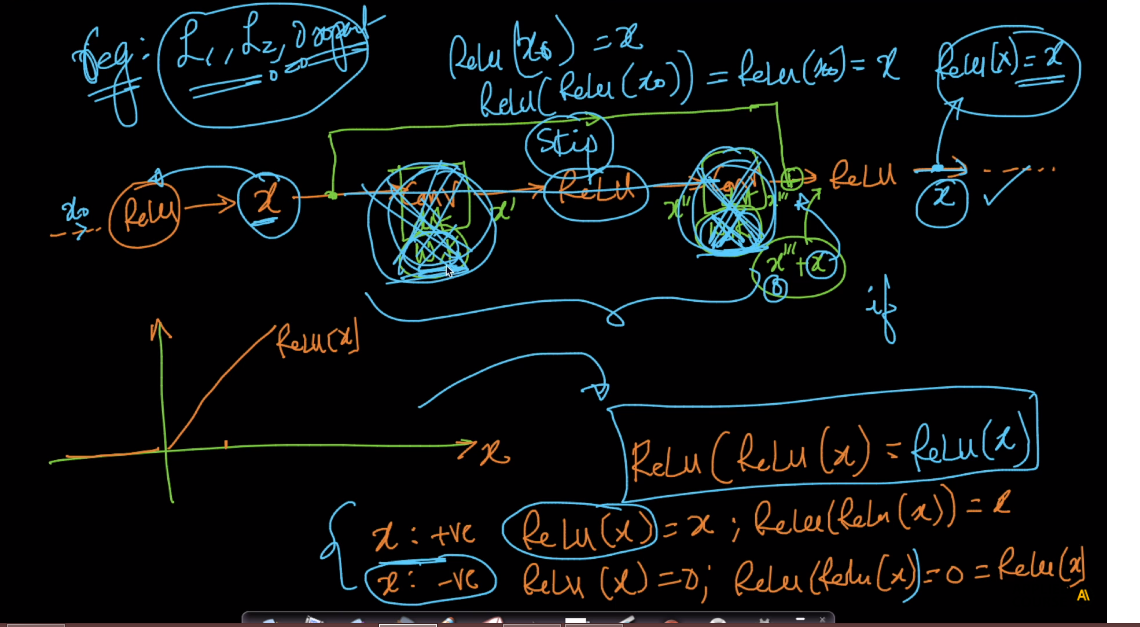
Before this residual/identity unit there is a network and also after this as well there is a network.



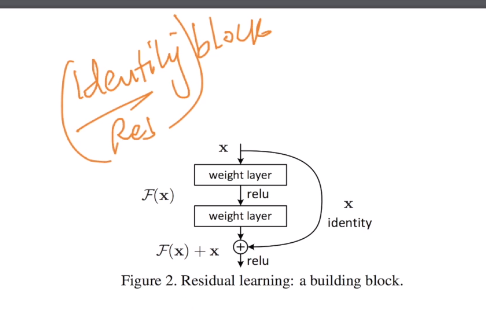


So what it do is take the output of relu(x) then apply it after second convolution layer (before relu) so that if in between two convolution is not useful and if we use regularization l1,l2, dropout then it makes this convolution 0 and by this we are not able to provide further information to network and also it effects in backpropagation as derivative becomes 0. So we use skip connection i.e if it becomes 0 then we add that x to o/p of second conv. So that we got relu(x) which is nothing but in this case relu(relu(x)) = x

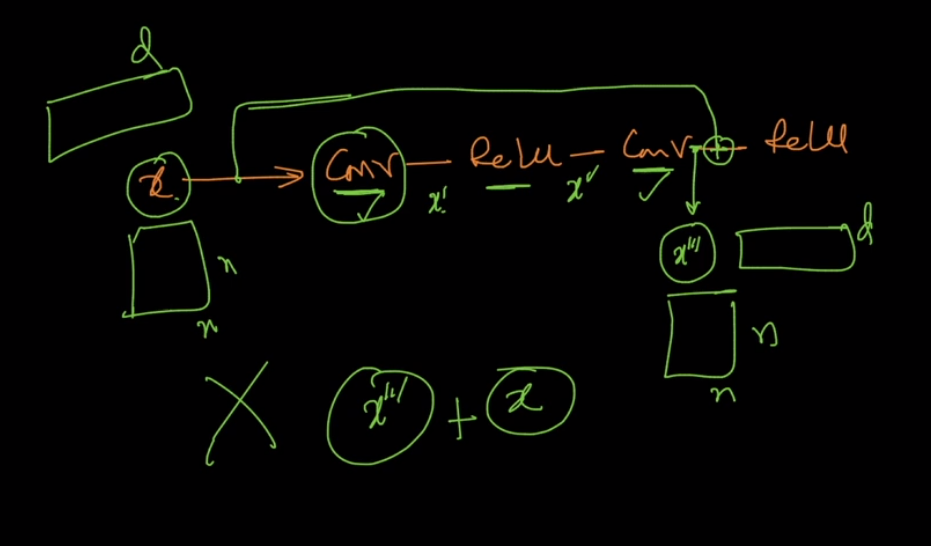
If we don’t do this then relu(0)= 0 and it creates the problem of vanishing gradient as well.



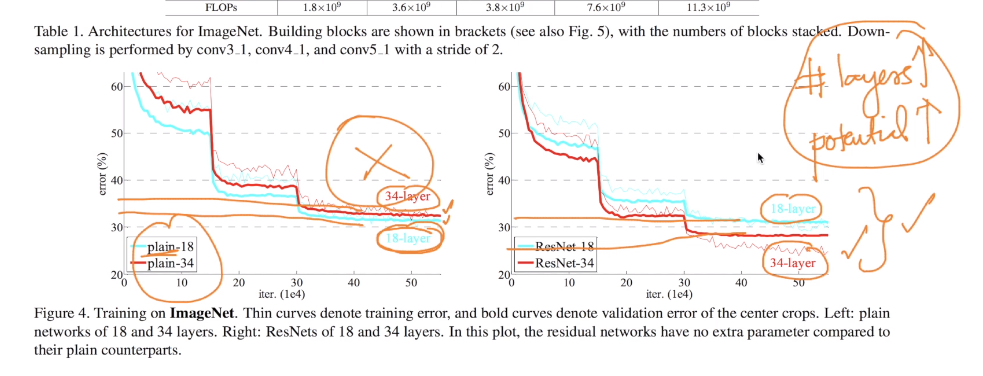
This block is also known as identity block because if that 2 layer are not useful then we just provide input x to output.



But 1 thing we should keep in mind while applying data in this residual block, data should be of same size if x image is of n\*n then x’’’ images should be of n\*n.



Now after applying this resnet we got expected result i.e as no. of layers increases, training error decreases.



Link : <https://github.com/keras-team/keras-applications/blob/master/keras_applications/resnet50.py>

Comments :

