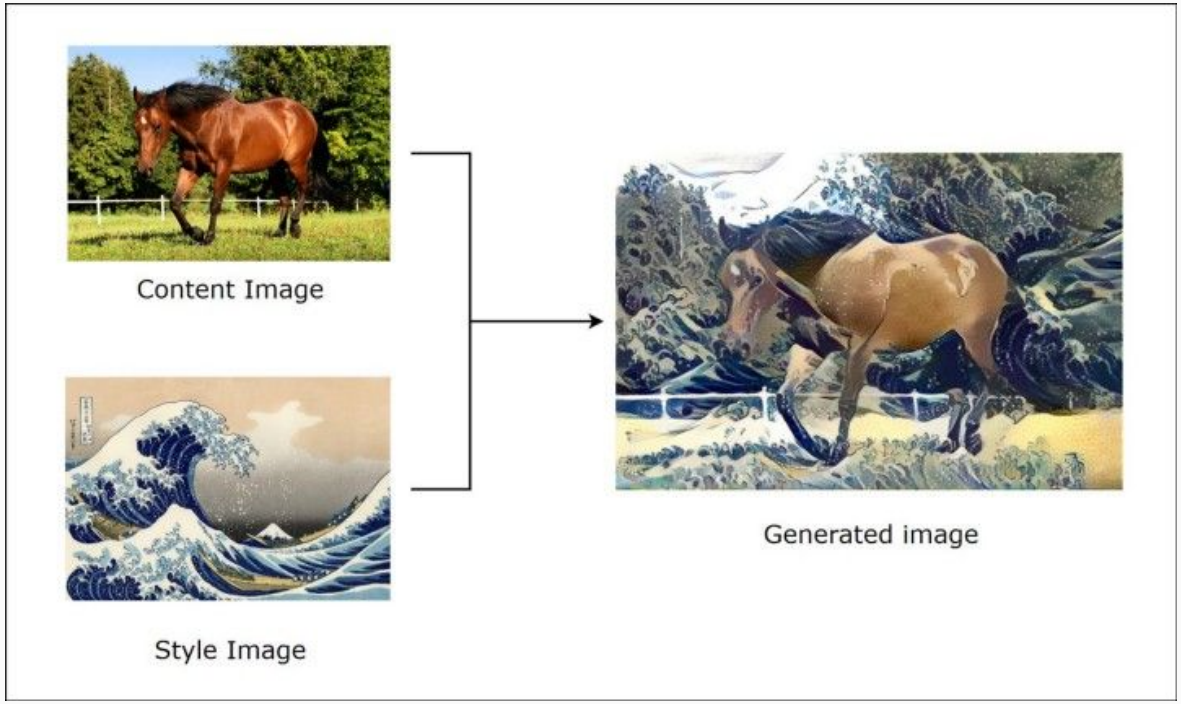
# NEURAL STYLE TRANSFER

# OVERVIEW



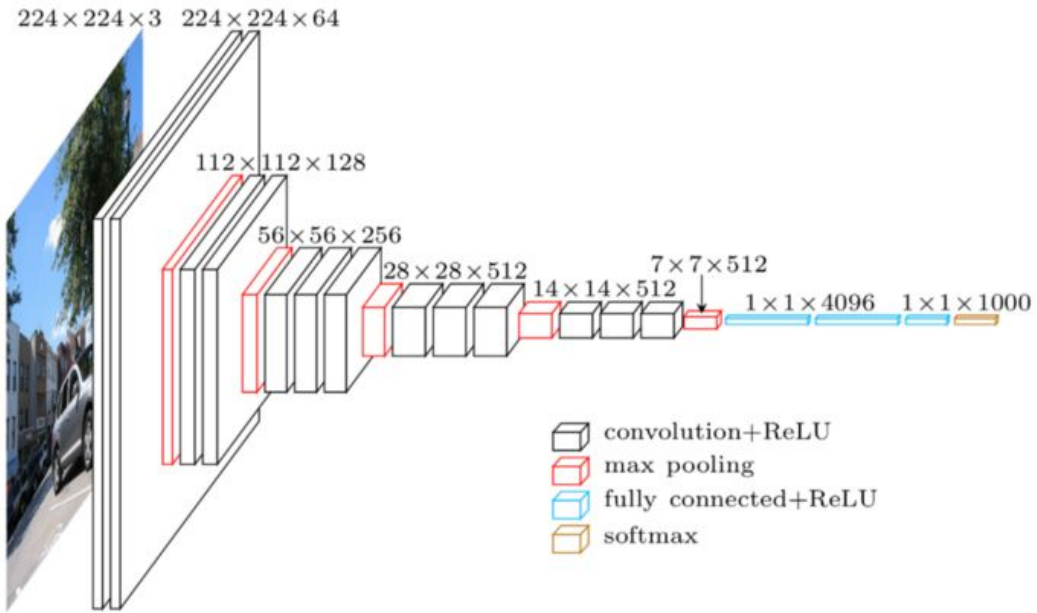

Convolutional Neural Networks were originally created for classification of images.

CNNs have lately been used in a variety of other tasks like Neural Style Transfer.

CNNs are one of the most interpretable models in Deep Learning because of our ability to visual their representations and understand what they might be learning.

SureStart

# UNPACKING A CNN




Feature maps they may have learnt to detect simple patterns, such that some neural units activate when they see a straight line or even for some other type of pattern which might not make any sense to a human eye but has huge value to this model.

This "**detection**" of straight lines or some pattern is called as learning a feature representation.

It's safe to assume that CNN does not learn to encode what image is but it actually learns to encode what image represents or what contents are visible in the image.

SureStart

# STYLE TRANSFER FROM IMAGE REPRESENTATIONS

- As training the model over a ten thousands of images per class the model is able to generate similar feature representation for many different images given they belong to same class or have similar content or style.
- Hence it makes sense to use the difference in value of feature representation of generated image w.r.t content and style image to guide the iterations through which we produce the generated image itself.

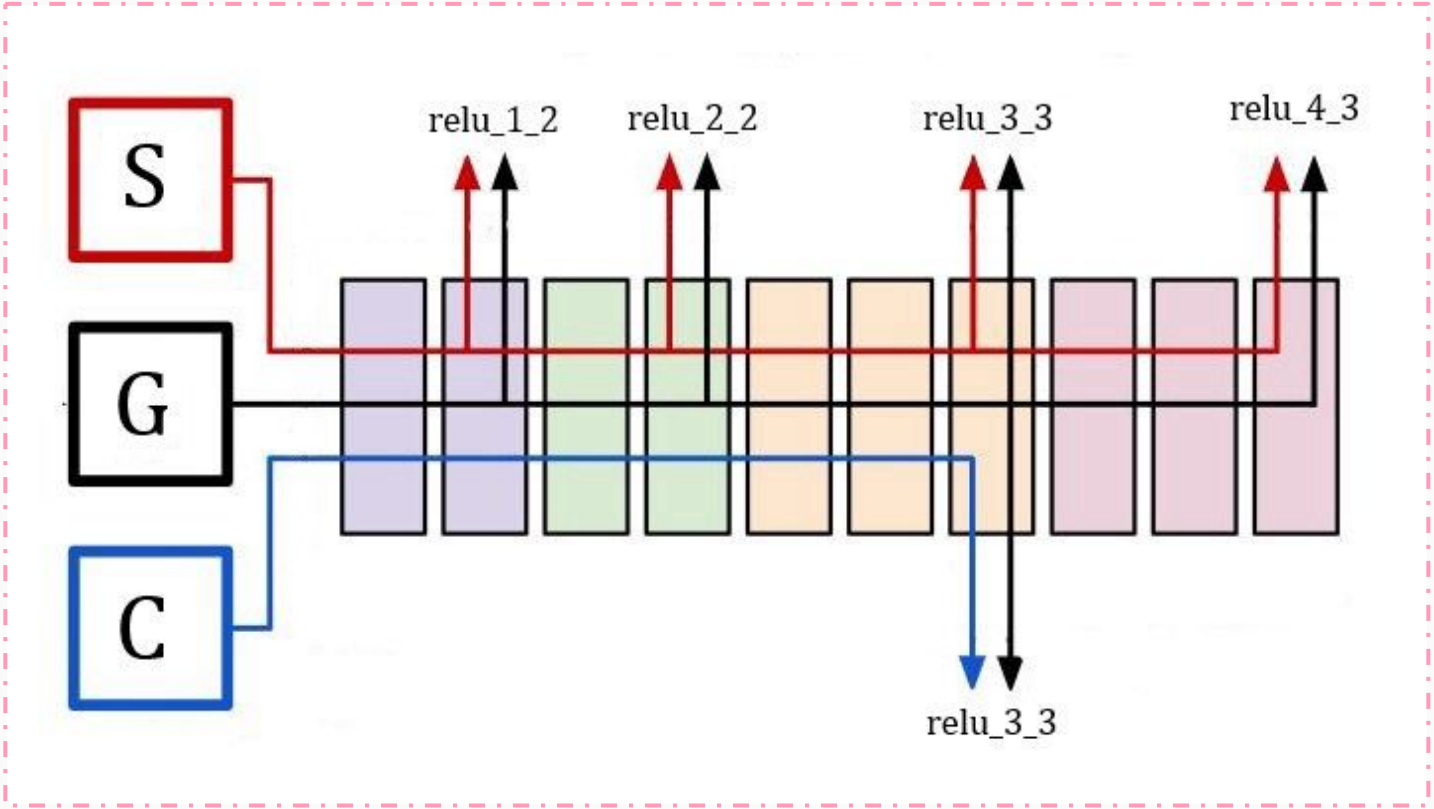
But how do we make sure that content image (C) and generated image (G) are similar with respect to their content and not style?

And how do we make sure that generated image only inherits similar style representation from style image (S) and not the entire style image itself?

Answer: Divide the loss into two parts – (a) Content loss, and (b) Style loss

1. During each iteration all the three images i.e. content image, style image and generated image are passed through the CNN model.
2. The value of the hidden unit's activation which encode feature representation of the given image at certain layers are taken as input to these loss functions.
3. In other terms, take the output of the layers in the CNN, there isn't any specific rule on selection of layers.

## REPRESENTATIONS



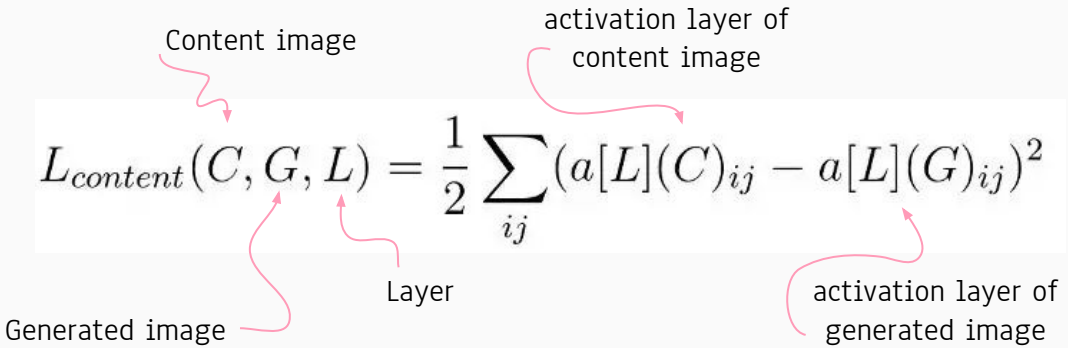
# CONTENT LOSS

Let's take the feature representation of only one of the layers, let's consider 7th convolution layer of a CNN

To calculate the content loss we pass both content image and generated image through a CNN and get the activation values (i.e outputs) of 7th conv layer for both of these images.

It can have ReLU for its activation, we will denote this layer's output generally as relu_3_3 because it is the output of third conv layer of third set/block of convolutions.

Our aim is to preserve the original content in the generated image by making sure to minimize the difference in feature representation which logically focuses on the difference between content of both the images.




SureStart

# STYLE LOSS

We will consider feature representation of many convolution layers from shallow to deeper layers of the model.

Unlike content loss we can't just find the difference in activation units.

We need to find the correlation between these activations across different channels of the same layer and to do this we need something called as the **Gram Matrix**.

Once we have both content and style loss, we add them up and use any optimizer to perform gradient descent to change generated image such that it decreases its loss after each iteration.


SureStart