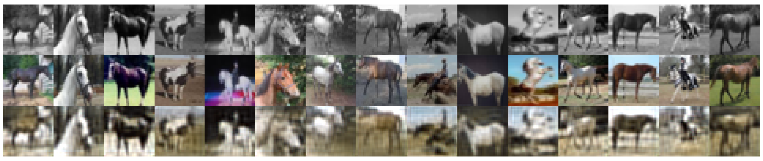
A. Colorization as Regression

1. In the model `RegressionCNN`, there are 6 convolution layers. Let k be the filter size, and n be the number of filter specified respectively via commandline, then the filter size and number of filters at each layers are as follows,

conv layer #	filter size	number of filters
1	$k \times k$	n
2	$k \times k$	$2n$
3	$k \times k$	$2n$
4	$k \times k$	n
5	$k \times k$	3
6	$k \times k$	3

2. Run `colour_regression.py`, the generated results looked similar to the original colored images. Specifically, homogeous regions of colors in the predicted images matches homogeous regions of color in the original colored images. However we see an apparent decrease in saturation in the set of predicted images. Also the predicted images are slightly blurred.



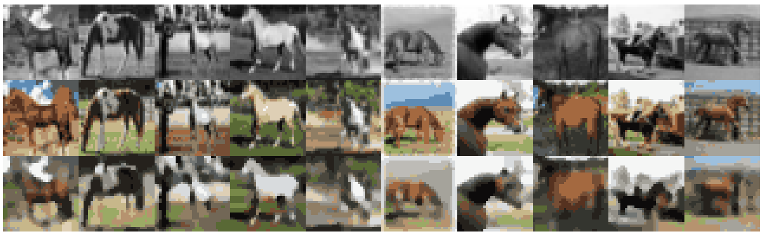
3. Using rgb color space can be problematic since it does not capture other attributes that are central to how human perceives color, such as lightness and saturation.

4. L2 regularization severely penalizes outliers, so will try to overfit the model to the dataset during training. If we frame colorization as a classification problem, then the loss function of each sample will not experience similar problem as when we frame colorization as a regression problem.

B. Colorization as Classification

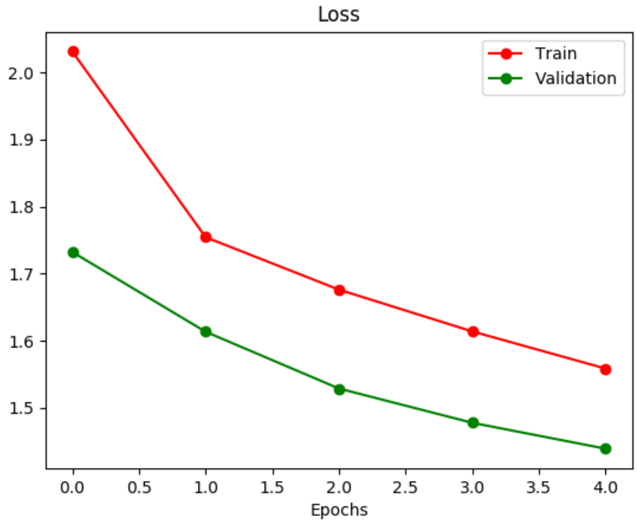
1.

2. The predicted images for model `CNN` have more saturated colors and in most cases matches with the original images when compared to `RegressionCNN`.



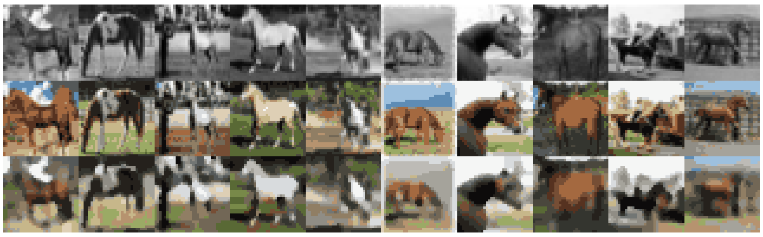
C. Skip Connections

1. Training curve as follows

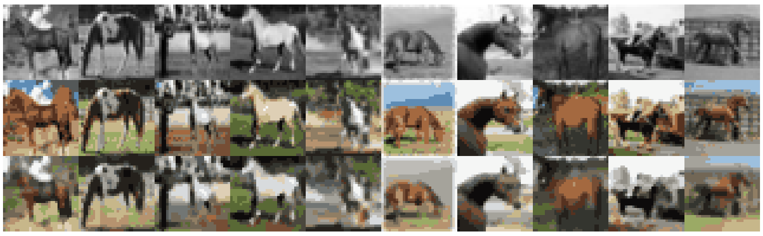


2. The answer is based on the pretrained UNet model provided (25 epochs). As the table below have shown, the skip connection improve validation loss and accuracy. Qualitatively, the UNet generated slightly more accurate colorization than CNN as shown below.

model	validation loss	validation accuracy
CNN	1.5881	41.1%
UNet	1.3659	48.0%



(a) CNN



(b) UNet

Here we give two reasons why skip connections might improve perfomance of `CNN` models

(a) Skip connection helps with supplying features to the latter layers in the convolutional neural network where certain features are lost due to upsampling

(b) Skip connection might make optimization more effective as it allows for back-propagation to adjust parameters early in the network.

D. Dilated Convolution

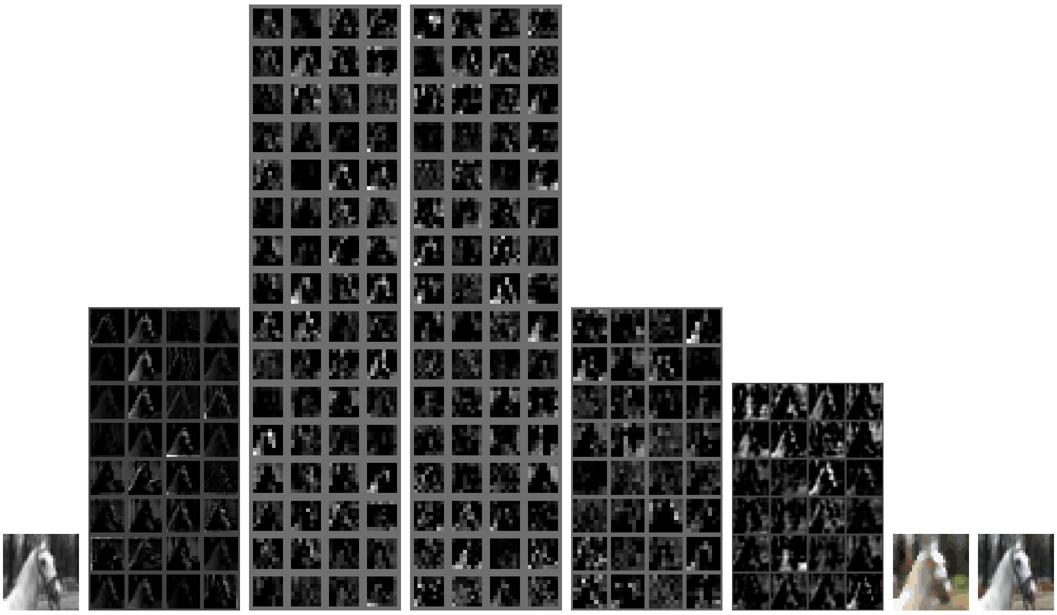
1. Let C_{in} be input channel size, C_{out} be output channel size, and k be kernel size. The weights and receptive field for different types of convolutions are shown below,

filter	weights	receptive field (filter) size
3 × 3 convolution	$9C_{in}C_{out}$	3 × 3
5 × 5 convolution	$25C_{in}C_{out}$	5 × 5
3 × 3 convolution with dilation 1	$9C_{in}C_{out}$	5 × 5

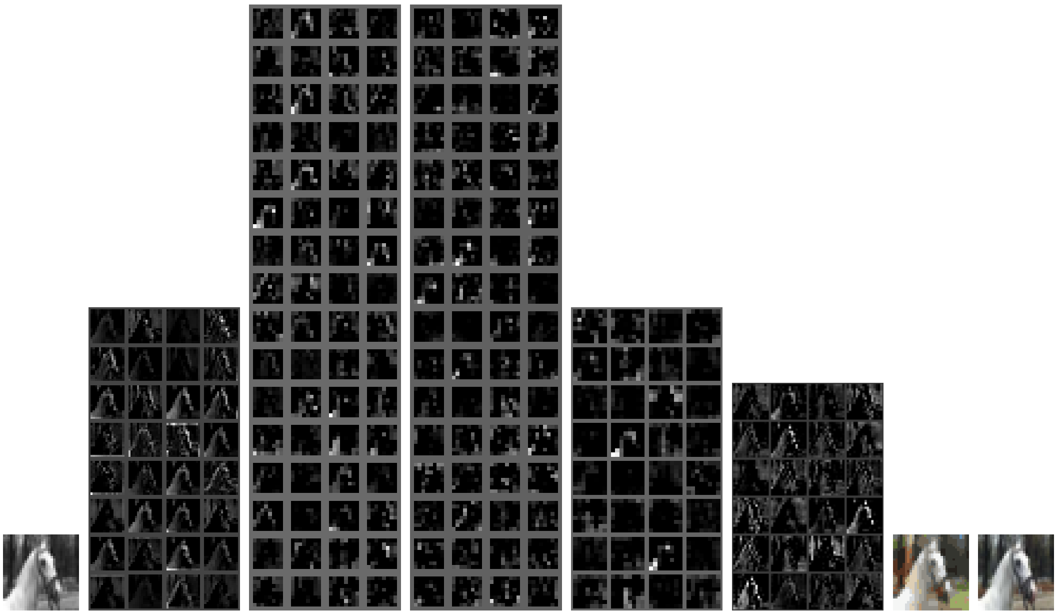
2. `DilatedUNet` replaces the middle convolution with a dilated convolution with dilation 1 here instead of another convolution because it is the layer that represents a turning point from max pooling to upsampling. Dilated convolution with dilation 1, by enlarging receptive field, will preserve the feature map size for purpose described. Additionally, dilated convolution integrate knowledge of a wider context in the input feature map by enlarging the receptive field.

E. Visualizing Intermediate Activations

1. The activations in first few layers preserve the contour, or shape of the input image approximately and the weights (color in greyscale) are typically similar in similar regions of the feature map. The activation in latter layers have weights (color in greyscale) that differ in the same region of the feature map. This implies that the output color channels in some region of the generated image are affected by one or a few input feature map. The later layers



2. We notice that the activations in layers (especially activations for the first and last convolution) for `UNet` resemble of a blend the corresponding layers in `CNN`. The earlier layers capture some color pattern and the latter layers exhibit inclusion of contour/shape pattern.



F. Conceptual Problems

1. Only b would be helpful in augmenting the data for our `CNN` model. The model would be exposed to a larger variety of cases where the subject of the image is a horse. Larger training dataset helps with generalization of the model and thus improve performance. Options like c and d would make the model overfit as max pooling makes model translation invariant. Option a would not be useful in this case, since an image flipped upside down does not describe realistic scenario. Option e would not be useful since the added image consists of subject other than horse, which we aim to color.

2. A list of some hyperparameters

   (a) number of filters (depth)

   (b) stride

   (c) padding

   (d) convolution filter size (receptive field)

   (e) pooling filter size (spatial extent)

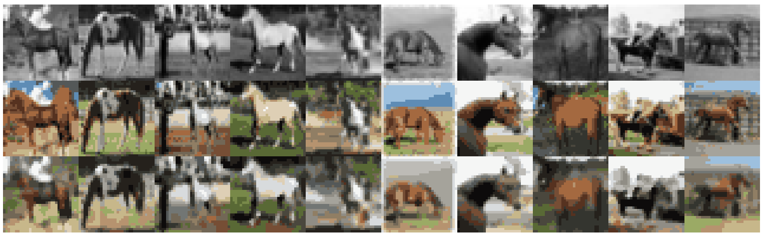
   (f) dilation

G. Dilated Convolution Implementation

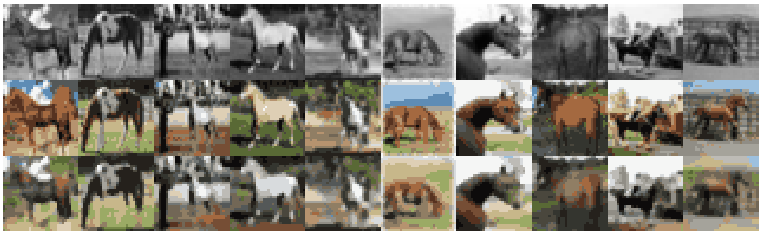
1.

2. Quantitatively, the `DUNet` model achieved worse validation loss and accuracy than `UNet` but better validation loss and accuracy than `CNN`. Qualitativey, `DUNet` performed similarly to `UNet`, although we observed there re miss-colorizations in some regions of the predicted image. Dilated filters maybe more efficient in cases where there are consecutive convolution layers stacked together; The receptive field for dilated convolution layer grows faster without the added cost of trainable parameters.

model	validation loss	validation accuracy
`CNN`	1.5881	41.1%
`UNet`	1.3659	48.0%
`DUNet`	1.4353	45.8%



(a) `UNet`



(b) `DUNet`