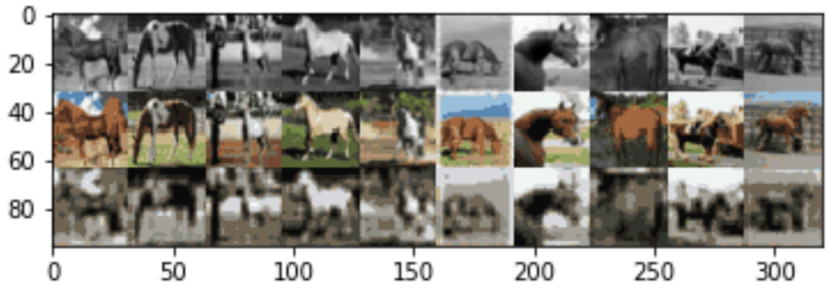
Part A
Question 2



The result does not look good for me, since the colour of the output images (first row) is much closer to the input images instead of the images with right colour(second row).

Question 3

Layers	Layer	Weight	Output	Connections
Layer 1	**MyConv2D-NF**	$k^2 \times channel \times NF + NF$	$W \times H \times NF$	$W \times H \times k^2 \times NF \times channel$
	Max Pooling	0	$\frac{1}{2}W \times \frac{1}{2}H \times NF$	0
	BatchNorm-NF	$2 \times NF$	0	0
Layer 2	**MyConv2D2NF**	$k^2 \times NF \times 2NF + 2NF$	$\frac{1}{2}W \times \frac{1}{2}H \times 2NF$	$\frac{1}{2}W \times \frac{1}{2}H \times k^2 \times NF \times 2NF$
	Max Pooling	0	$\frac{1}{4}W \times \frac{1}{4}H \times 2NF$	0
	BatchNorm-2NF	$2 \times 2 \times NF$	0	0
Layer 3	**MyConv2D-2NF**	$k^2 \times 2NF \times 2NF + 2NF$	$\frac{1}{4}W \times \frac{1}{4}H \times 2NF$	$\frac{1}{4}W \times \frac{1}{4}H \times k^2 \times 2NF \times 2NF$
	BatchNorm-2NF	$2 \times 2 \times NF$	0	0
Layer 4	**MyConv2D-NF**	$k^2 \times 2NF \times NF + NF$	$\frac{1}{4}W \times \frac{1}{4}H \times NF$	$\frac{1}{4}W \times \frac{1}{4}H \times k^2 \times 2NF \times NF$
	Upsample	0	$\frac{1}{2}W \times \frac{1}{2}H \times NF$	0
	BatchNorm-NF	$2 \times NF$	0	0
Layer 5	**MyConv2D-NC**	$k^2 \times 2NF \times NC + NC$	$\frac{1}{2}W \times \frac{1}{2}H \times NC$	$\frac{1}{2}W \times \frac{1}{2}H \times k^2 \times NF \times NC$
	Upsample	0	$W \times H \times NC$	0
	BatchNorm-NC	$2 \times NC$	0	0

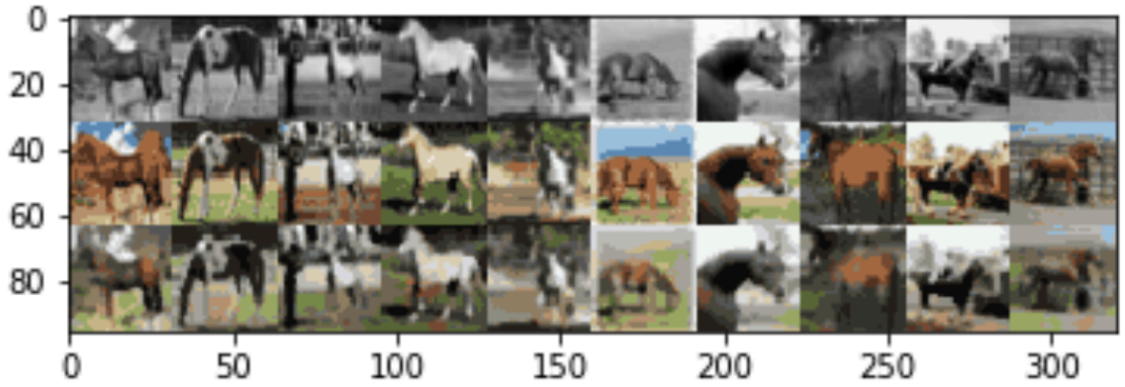
Layers	Layer	Weight	Output	Connections
Layer 6	**MyConv2D-NC**	$k^2 \times NC \times NC + NC$	$W \times H \times NC$	$W \times H \times k^2 \times NC \times NC$

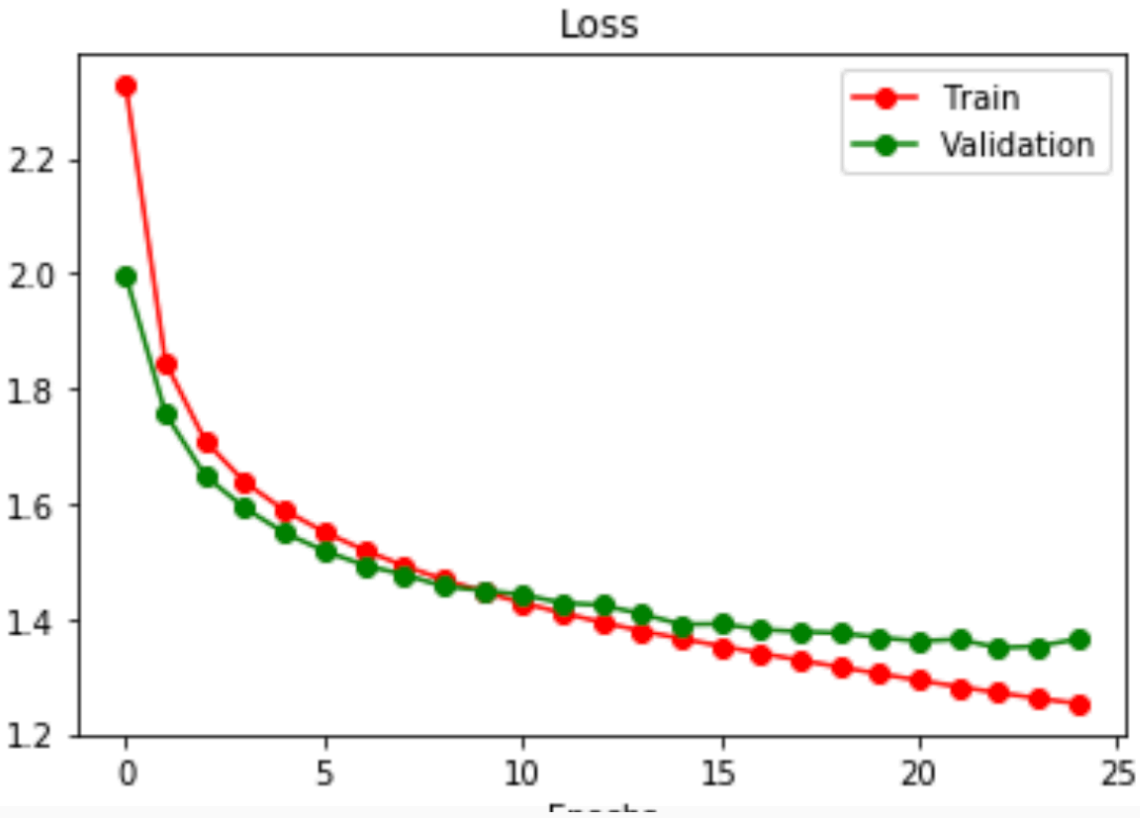
Total_weight = $27NF + 72NF^2 + 9NC^2 + 9NF \times NC + 4NC$
Total_output = $2368NF + 2304NC$
Total_connection = $9216NF + 8064NF^2 + 2304NF \times NC + 9216NC^2$

After doubled:
Total_weight = $27NF + 72NF^2 + 9NC^2 + 9NF \times NC + 4NC$
Total_output = $4 \times (2368NF + 2304NC) = 9472NF + 9216NC$
Total_connection = $4 \times (9216NF + 8064NF^2 + 2304NF \times NC + 9216NC^2)$
$= 36864NF + 32256NF^2 + 9216NF \times NC + 36864NC^2$

Question 4
The prepossessing step will not affect the $2 \times NF$output. Since this operation does not result in any overflows, so the prepossessing step only did a linear transformation to the input x ($y = ax + b$), and in the neural network, we calculate $wx + b$ is also a linear function. Thus, in the training step, it will generate an alternative w' and b' to reach the optimal.

Part B
Question 2





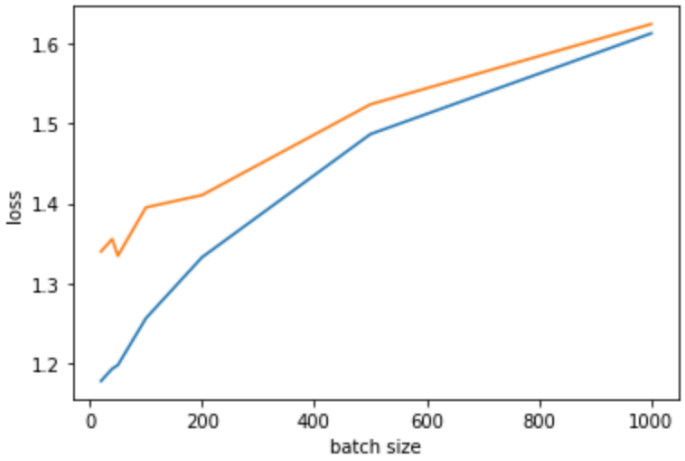
Question 3

The result is better than the previous, the the accuracy increased from 30% to around 50% with the validation loss dropped from 1.99 to 1.32. And for pictures, the colour output from Unet is closer to the coloured images.

Reason1: the skip connections have more parameters than the previous, and more features are included in layers.

Reason2: When we calling max pooling and up sample, parts of the features in images will be lost. With the skip connections, it will restore some features that ignored in the max pooling with the previous layers.

Question 4



If we change the batch size and remain everything else unchanged, the losses (training loss and validation loss) increase when batch size get larger, but training time decreases. The out put images with large batch size has lower quality.

Part C

Question 4

fine-tuning an entire pre-trained model will has the memory complexity$O(n)$ times larger than the memory complexity of fine-tuning only the last layer. However, both models have forward operation, and their computation cost will be same.

Question 5

If we increase the height and the width of the input image by a factor of 2, the memory for the feature map increased by 4, but the number of parameters will remain the same thus the complexity of fine-tuning will not be affected.