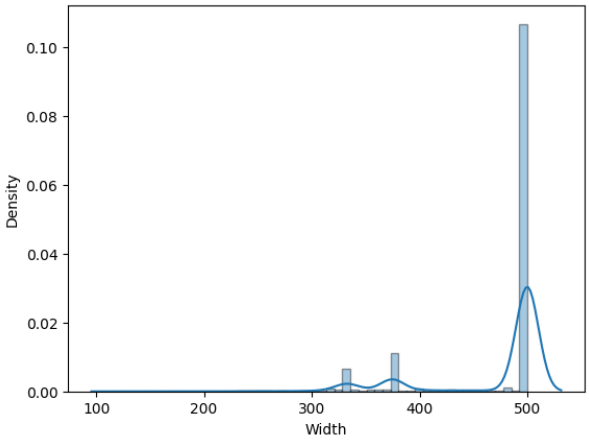
Assignment 3 - Report

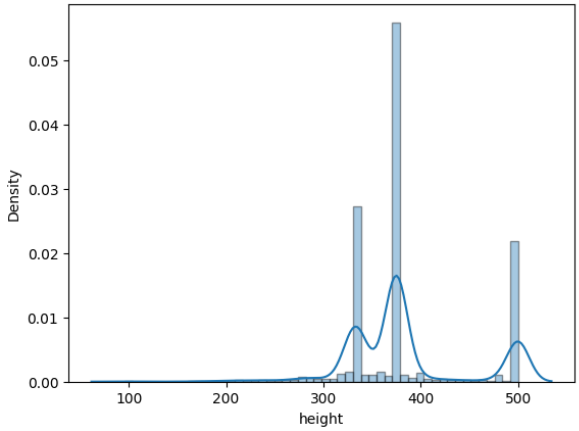
By: Ziv Yogev

Step 1

As requested, I created a class for data which receives a list of paths to read, and at each step forward reads one image and returns 3 images according to the appropriate size (72,144,288). The number of images in the data set is 5011 and I divided the data into a training and validation set by 20% for the validation set.

The average of the fiscals is 472X383 with a standard deviation of ~60, therefore most of the images are larger than 288^2. You can see from DF that most of the images are above the size of y_large, therefore we will inevitably have to lower the image quality for most of the images.



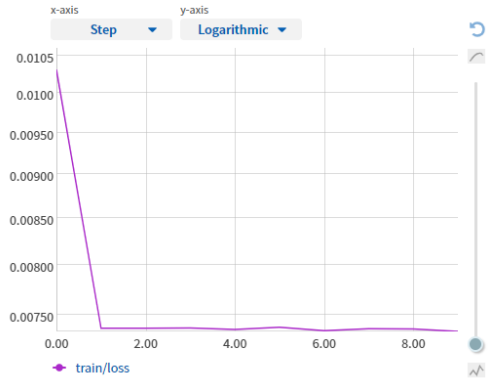


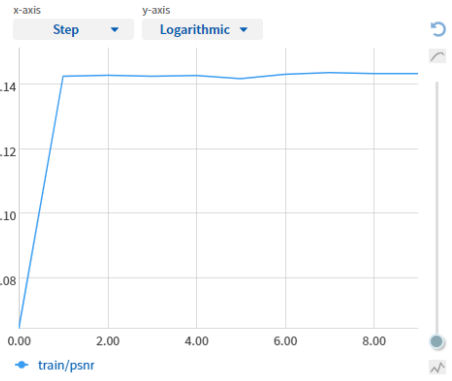
Tracking and monitoring of the data will be done with naptune.ai platform. I added the PSNR metric that describes the maximum possible ratio between the signal power and the noise power (physics) that affects the reliability of the displayed signal. We used a logarithmic scale so that we could measure the ratio in decibels. We would like this threshold to be as high as possible and offer a high reliability of the object's reconstruction.

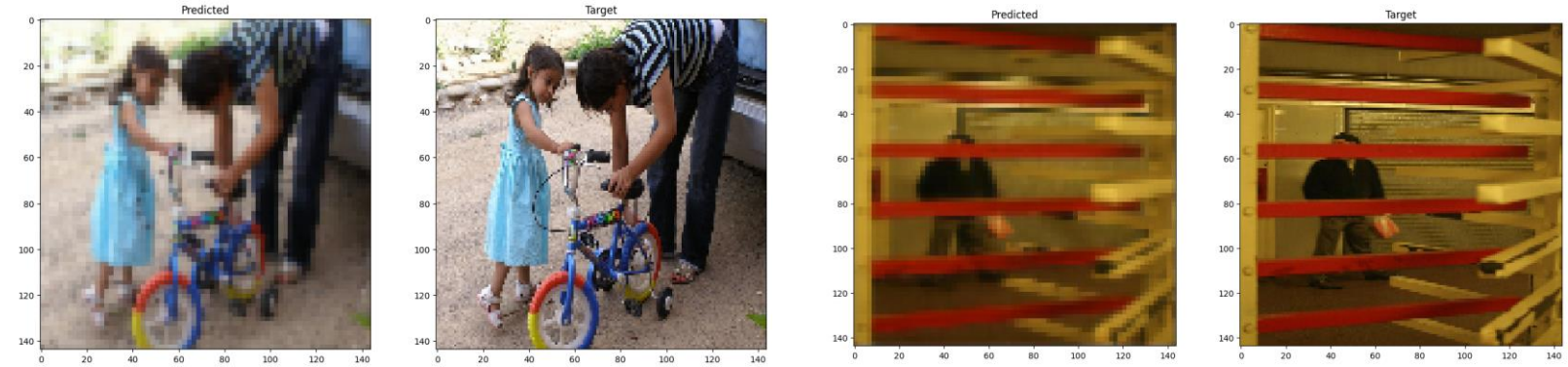
Here is an **example** of reading an image from the data loader:

Step 2

I implemented an initial network containing 2 convolution layers, an upsampling layer, and another convolution layer. I trained the model by using x and y_mid only. In X-axis Step mean Epoch.








```
Finish training epoch 7 with train loss: 0.007344
valid_loss = 0.007450 , psnr = 2.158
```

Parameters details:

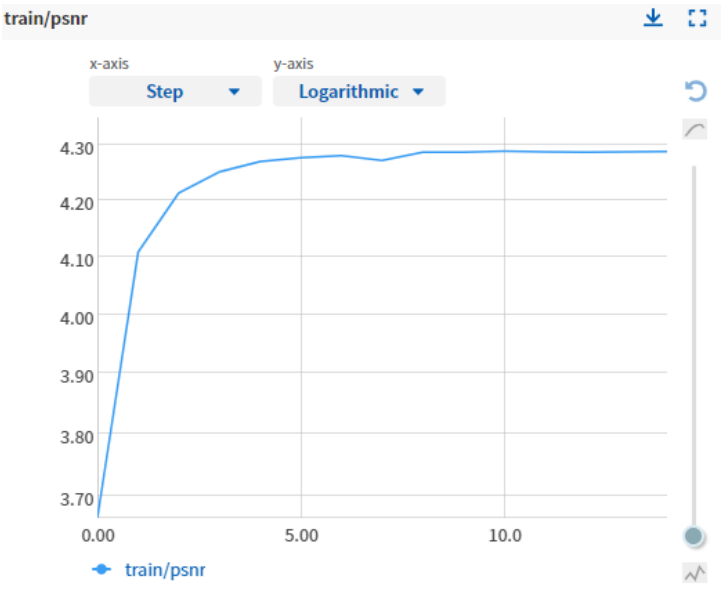
Parameter	Value	Explanation
Criterion	MSE	MSE is the most well-known and used loss function for prediction tasks when you want to differentiate at a certain distance between objects. I decided to start using it due to its effectiveness.
learning_rate	0.001	The learning rate is an important characteristic when we change the number of epochs in the parameter as well, so it is important to have a match between these two parameters. Our number of images is not large, so we would like a low lr.
Optimizar	Adam	
Batch_size	10	
Limit	10	10 steps with Un-improvement until stop training

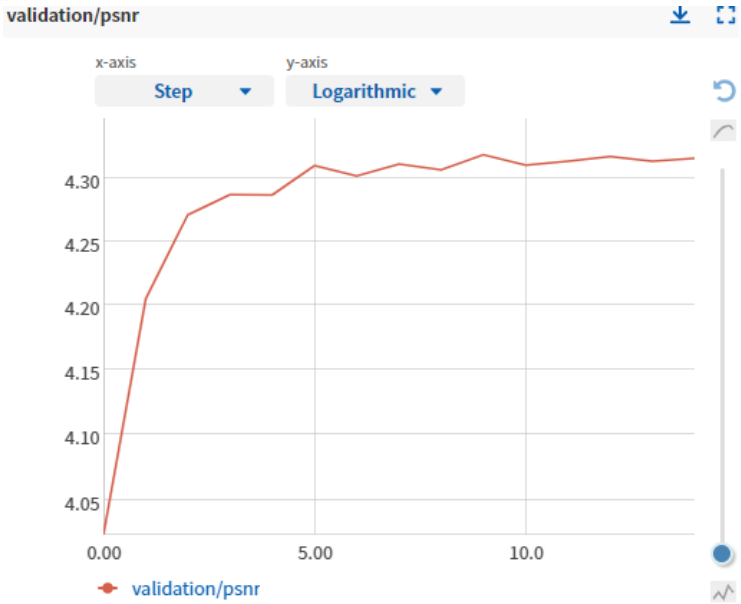
I tried to change the function loss (MAE) and learning rate (5e-3) with 15 epochs to see if there is improvement in psnr metric, or in loss. As you can see, there is improve in the results of PSNR but not on loss. In terms of the visibility of the image, the blur is still present and has even increased, but the color classification has become clearer.

```
valid_loss = 0.048435 , psnr = 4.269
```

Step 3

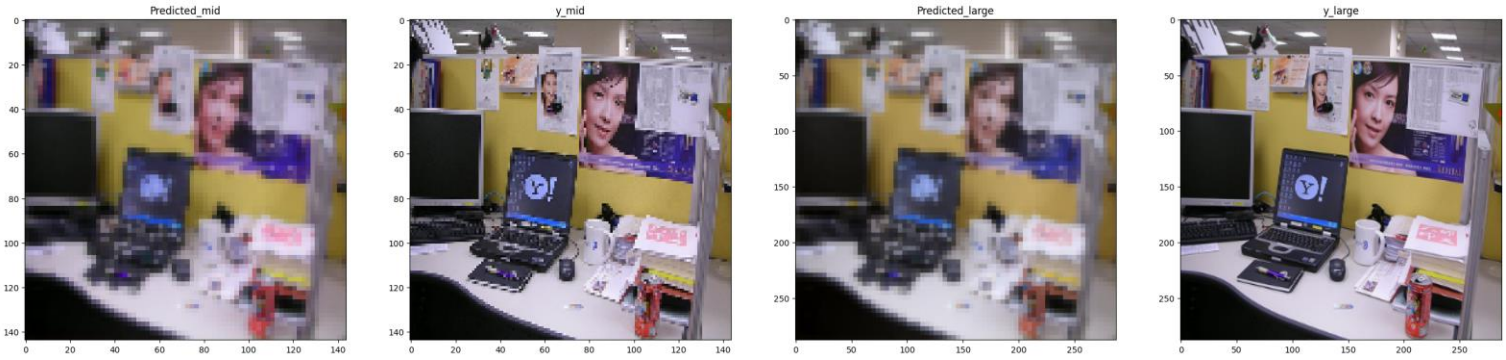
Add another block so we can predict y_mid and y_large.






```
Finish training epoch 15 with train loss: 0.014645
valid_loss = 0.014841 , psnr = 4.317
```

The results here was better than previous step.



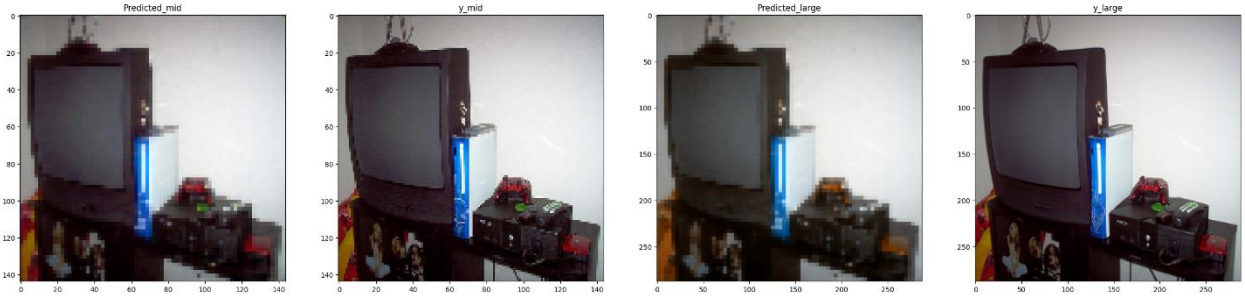
The quality of color matching and saturation for the higher quality image is less good than the lower quality image.

Step 4

Add residual blocks with ReLU activion in place of conv2D layers before Upsampling. At this Step, an attempt was made to improve the model by changing the following hyperparameters:

Rows with diff only / Show cell changes		AS-24	AS-25	AS-26	AS-27
parameters/batch_size		10	10	10	10
parameters/betas		-	-	(0.9, 0.95)	-
parameters/criterion		MSE	MAE	MAE	MAE
parameters/learning_rate		0.001	0.002	0.02	0.0002
parameters/Lmit		10	10	10	10
parameters/max_epochs		25	15	15	20
train/loss	min	0.0209415	0.106328	0.459761	0.0996419
train/psnr	max	4.00341	4.15753	2.24731	4.20899
validation/loss	min	0.021347	0.105297	0.459128	0.0996389
validation/psnr	max	4.02061	4.19291	2.27849	4.23574

When you look at the graph, you can see that the convergence of the models is relatively similar. In all parameters, the "AS-28" experiment was the most successful, so we will use its characteristics.



Example of "AS-28" experiment in after 15 epochs (Best result)

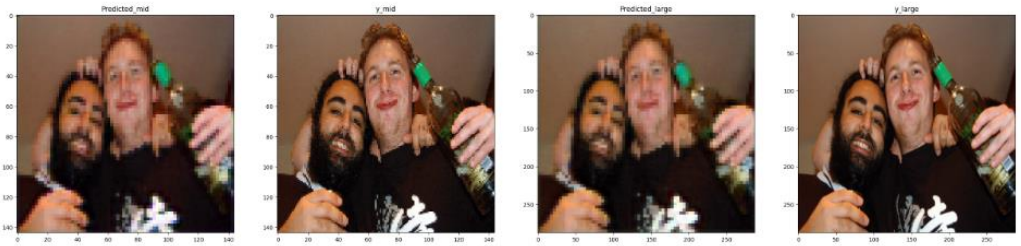
Step 5

Replace residual blocks and defined dilated convolutional block with Leaky ReLU activion.
At this step a comparison is made between training the model with ReLU vs Leaky ReLU. The model that uses Leaky ReLU presented the images in a clearer visual way.
As we can see, there is not a lot of different between the measurements of the models.

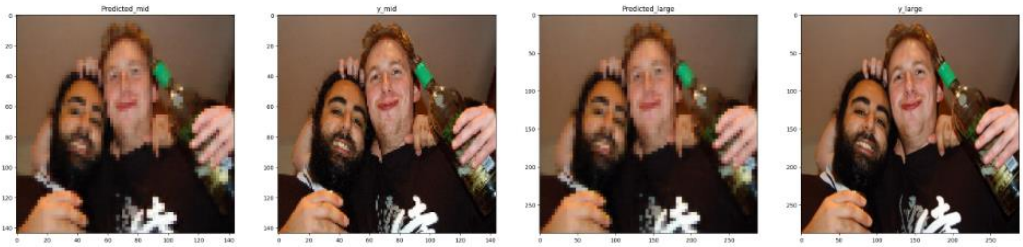
		AS-28	AS-31	AS-32
train/loss	min	0.0928736	0.0919456	0.0918786
train/psnr	max	4.29331	4.30266	4.29846
validation/loss	min	0.0908715	0.0901954	0.0900053
validation/psnr	max	4.34211	4.35219	4.35255

Step 6

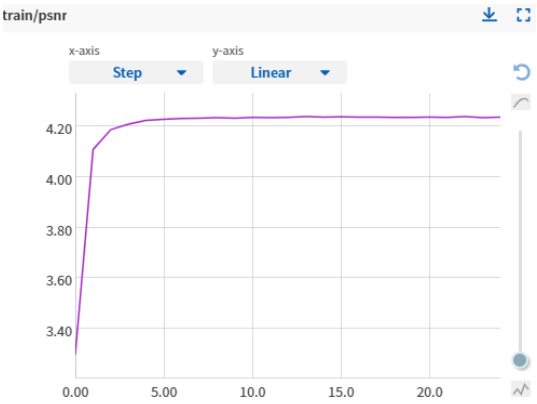
Add pretrained network (VGG16) and use feature extractor to the network.
At this step you can see that the images do come out in the appropriate shades, but they still lack the contrast to reduce the blur in the image. The model improves all over the epochs.

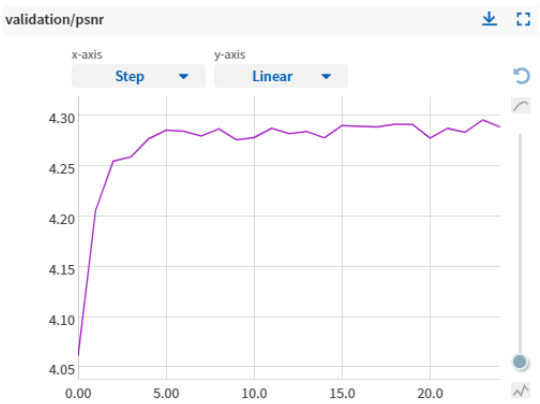


Epoch 3



Epoch 19

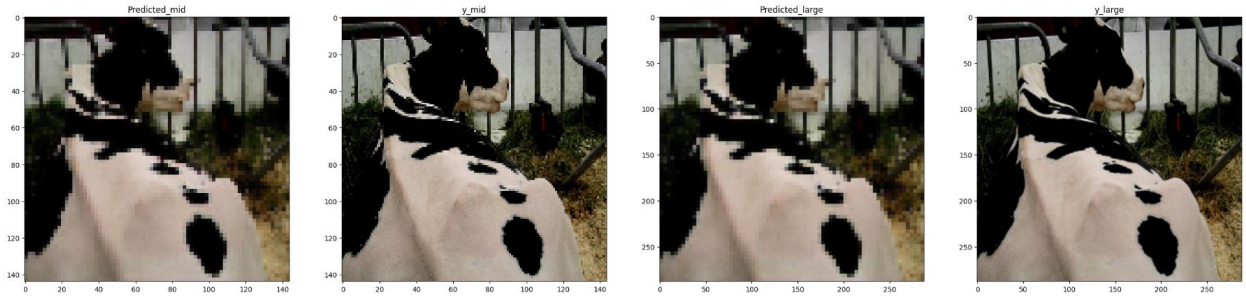




Step 7

Replace Upsampling with Pixel_Shuffle, an operation to implement efficient sub-pixel convolutions with a stride of $\frac{1}{r}$.

Last epoch:



There are not many differences between the predicted image for y_mid and y_large. However, there is an improvement in the PSNR measurement, and it is the highest performance among all the models.

validation/psnr	max	4.43357