GoogLeNet Regression Model

Created by Unknown User (barrelchester), last modified on Apr 28, 2023

SimpleCube and GoogLeNet

The SimpleCube dataset is a smaller version of CUBE++ consisting of 1772 training images and 462 test images with a single illuminant. Images are 16-bit raw PNG of size 648 x 432 preprocessed with only the simplest debayering. The "SpyderCube" color target present in the lower right half of each image is cropped out.

This pytorch model uses the pretrained GoogLeNet model available through torchvision, with the classification layer replaced by an RGB regression output layer. For each of the following training runs a learning rate of 1e-4 was used with the AdamW optimizer. Runs of 50 epochs and 100 epochs were performed.

The model was tested with a cross validation (CV) set every epoch, and the model was stored every time the CV loss improved. After training, the last stored model was loaded and run on the test set provided with the SimpleCube dataset. The metrics returned by the test method are statistics about the angular error (in degrees) between the ground truth RGB vector and model output vector, and include mean, median, trimean, best 25, and worst 25.

The experiments are divided by pixel value normalization method.

Normalization Pixel Ranges

Below are the min, mean, standard deviation, and max values of a subset of the image data using various normalization methods.

norm	min	mean	std	max
none	0	2875.04	2061.43	15319.0
linear	0	0.08	0.12	0.92
log	0	6.79	2.94	9.63
linear, log	-9.57	-2.4	1.44	0
linear x 2^{16}	0	5244.25	7868.23	60670.73
linear x 2^{16} , log	0	6.89	3.13	11.01

Training Runs

01 Normalization

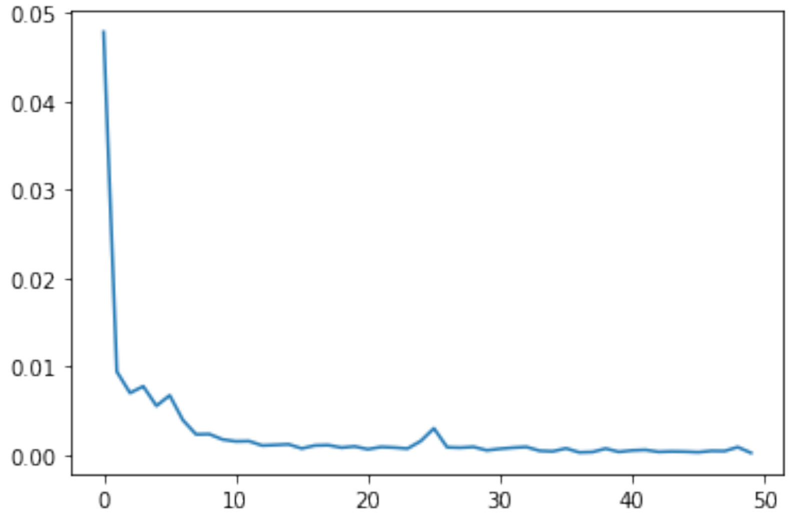
With this run, the 16-bit pixel values were scaled between 0 and 1 by dividing each by 2^{16}-1.

Resulting pixel values

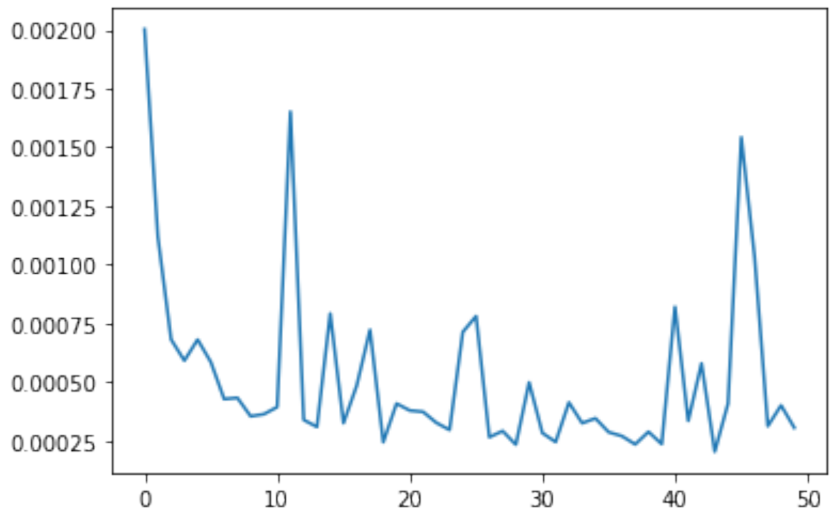
min	mean	max
0	0.03751504	0.1343557

Loss Plots - 50 Epochs

Training loss per epoch



CV loss



Test Results

Mean	Median	Trimean	Best 25	Worst 25
1.976820	1.254838	1.766730	0.428085	4.732106

Linear Normalization - Black Point and Saturation Correction Normalization

This normalization is given by https://github.com/Visillect/CubePlusPlus/blob/master/challenge/make_preview.py

Black point is set to 2048 and saturation level to 2^{14}-1.

```
x = np.clip((x - self.black_lvl)/(self.saturation_lvl - self.black_lvl), 0, 1)
```

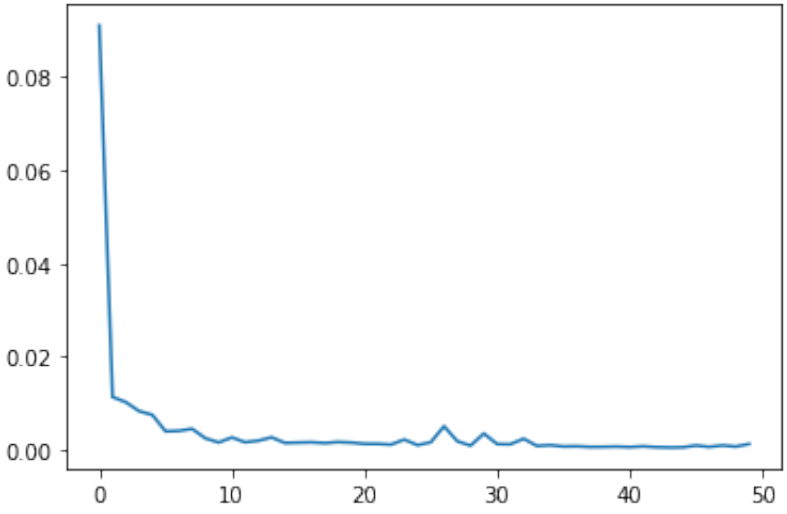
Resulting pixel values

min	mean	max
0	0.050967675	0.47136378

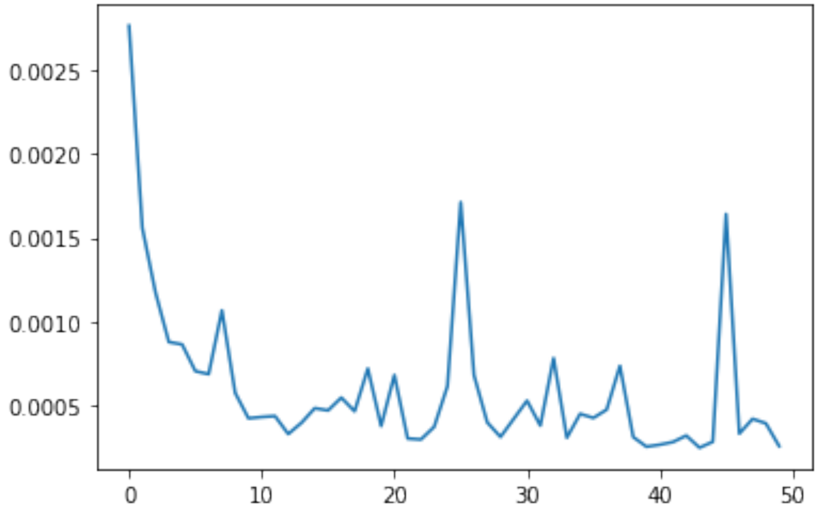
Loss Plots

50 Epochs

Training loss per epoch

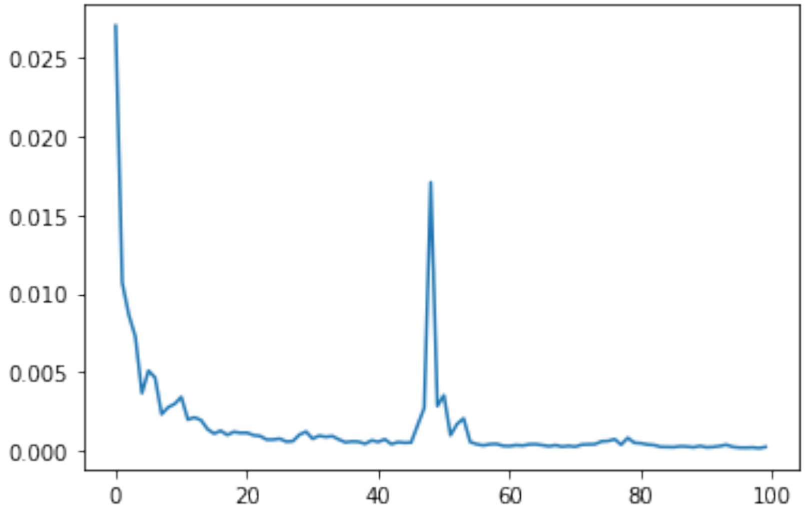


CV loss

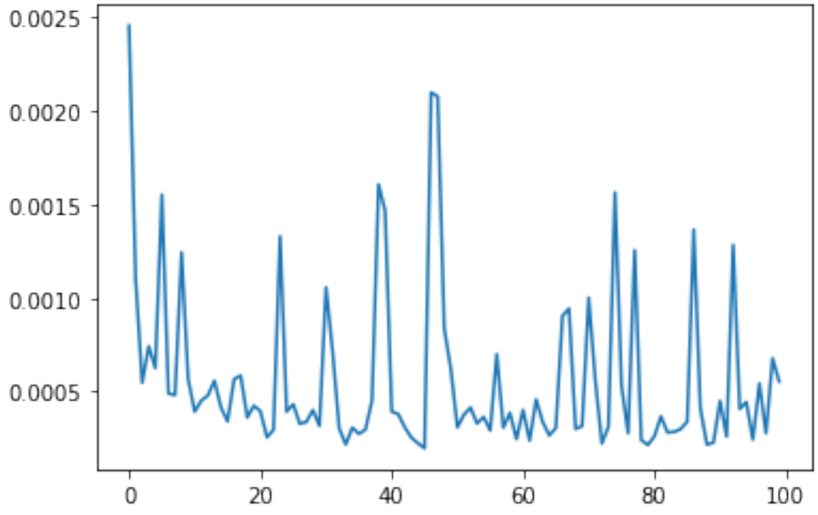


100 Epochs

Train

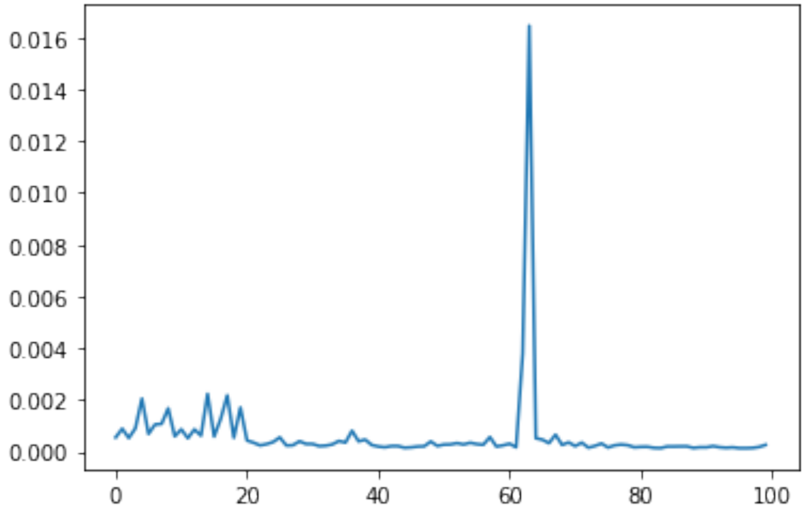


CV

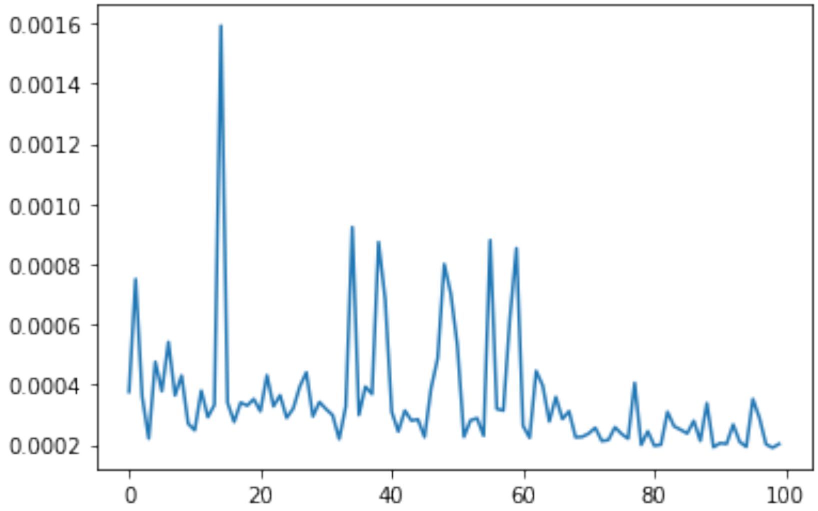


100-200 Epochs

Train



CV



Test Results

50 Epochs

Mean	Median	Trimean	Best 25	Worst 25
1.928371	1.305662	1.755973	0.414628	4.463303

100 Epochs

Mean	Median	Trimean	Best 25	Worst 25
1.810919	1.272023	1.648525	0.381628	4.250604

200 Epochs

Mean	Median	Trimean	Best 25	Worst 25
1.622245	1.034705	1.432307	0.280100	4.033163

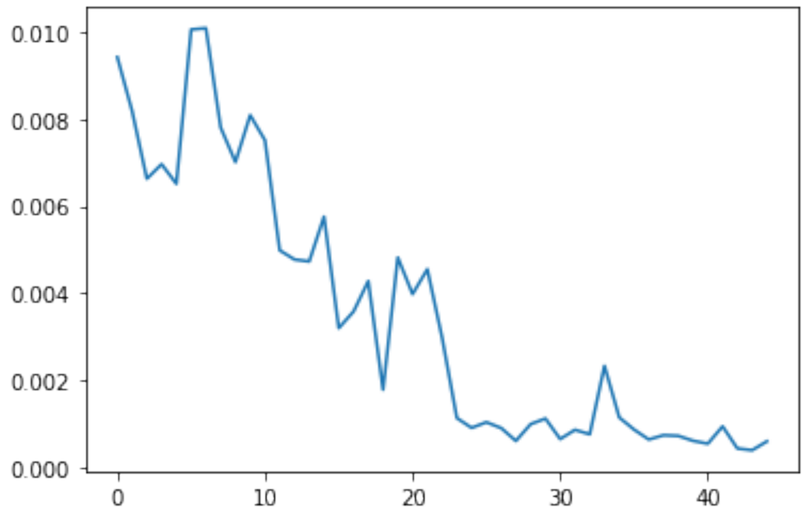
Log Normalization

Logarithm is applied directly to non-0 raw pixel values.

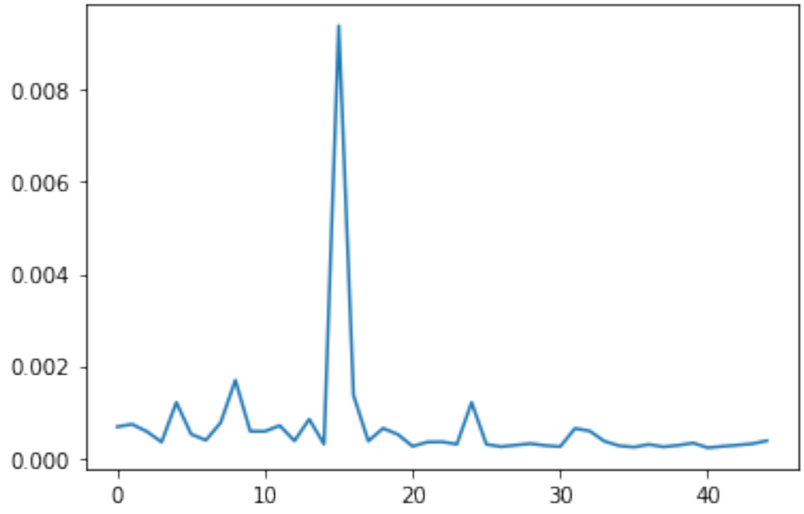
Loss Plots

50 Epochs

Training loss per epoch

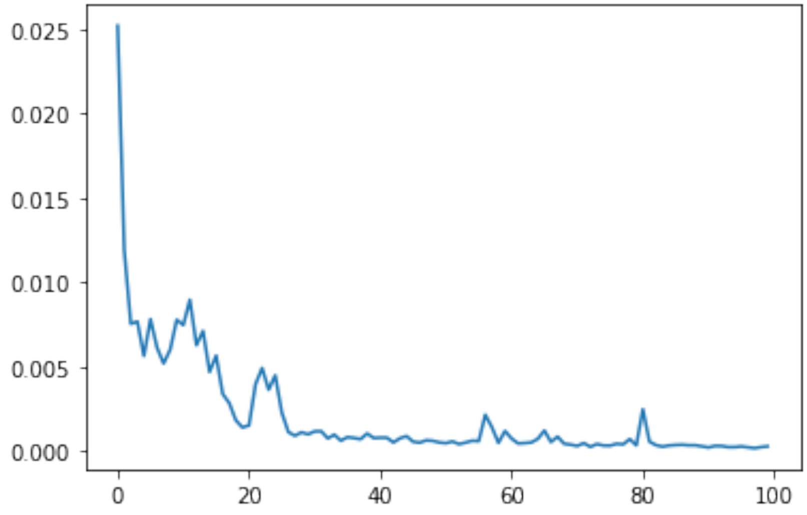


CV loss

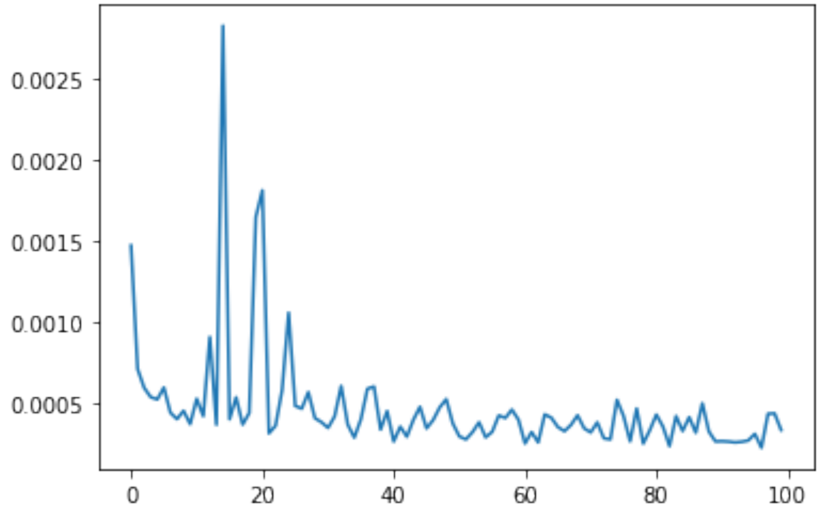


100 Epochs

Train

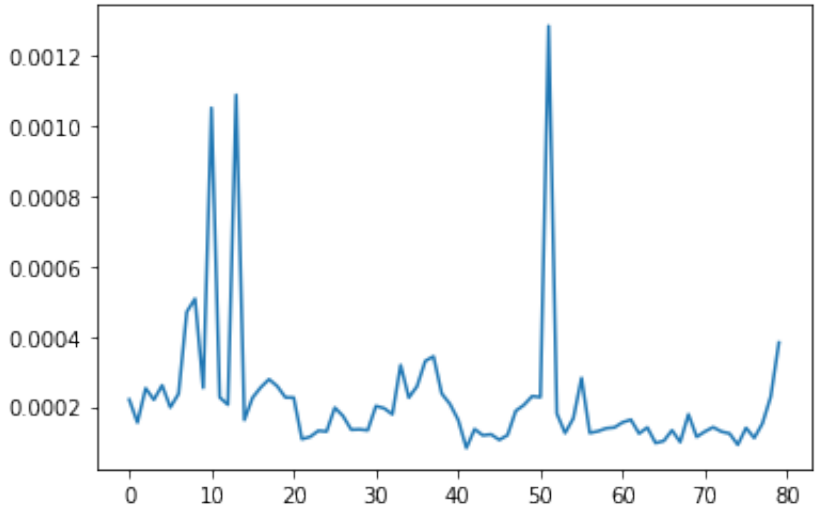


CV

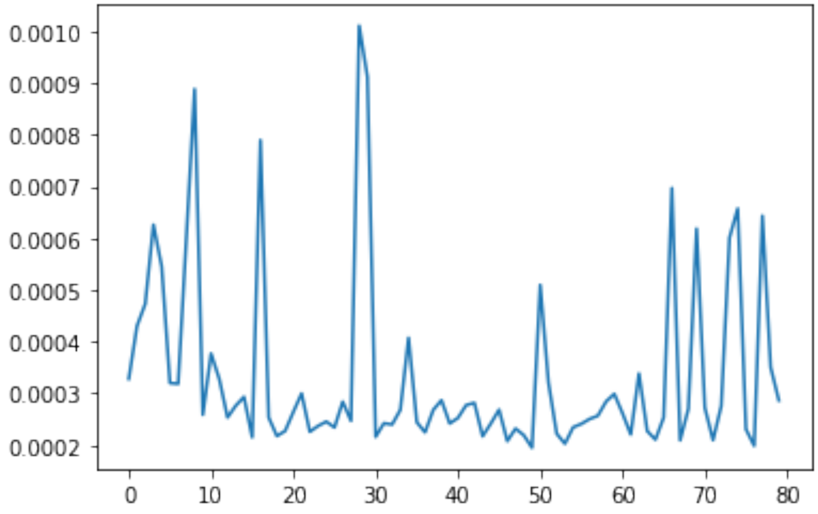


100-200 Epochs

Train



CV



Test Results

These results were slightly worse than the linear results.

50 Epochs

Mean	Median	Trimean	Best 25	Worst 25
1.946382	1.303034	1.732936	0.479687	4.582974

100 Epochs

Mean	Median	Trimean	Best 25	Worst 25

1.824393	1.164573	1.656479	0.324168	4.399156

200 Epochs

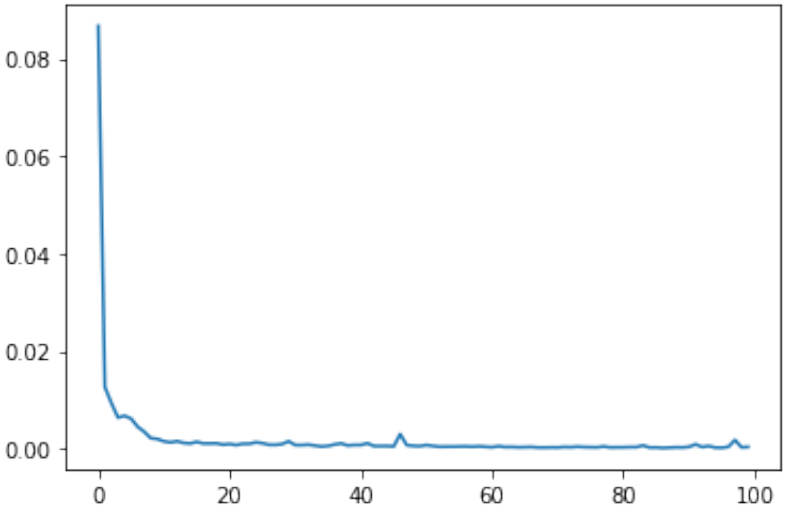
Mean	Median	Trimean	Best 25	Worst 25
1.738321	1.066977	1.579767	0.307531	4.316566

Linear then Log

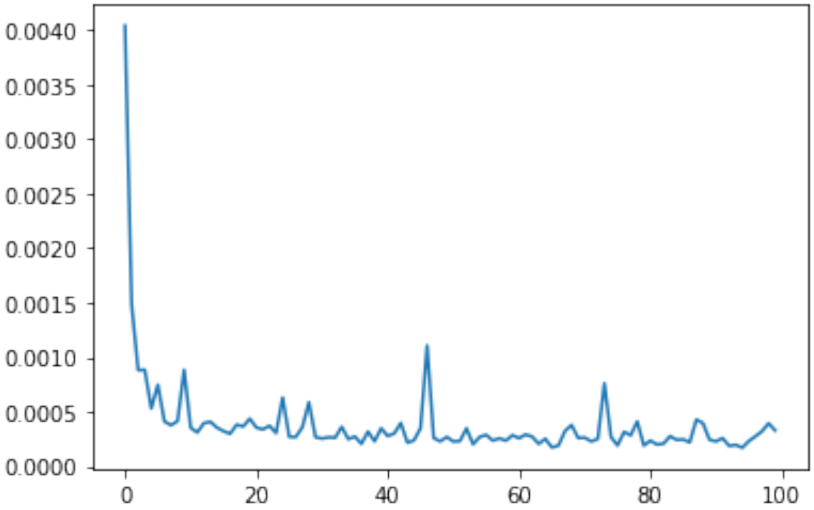
Loss Plots

100 Epochs

Train

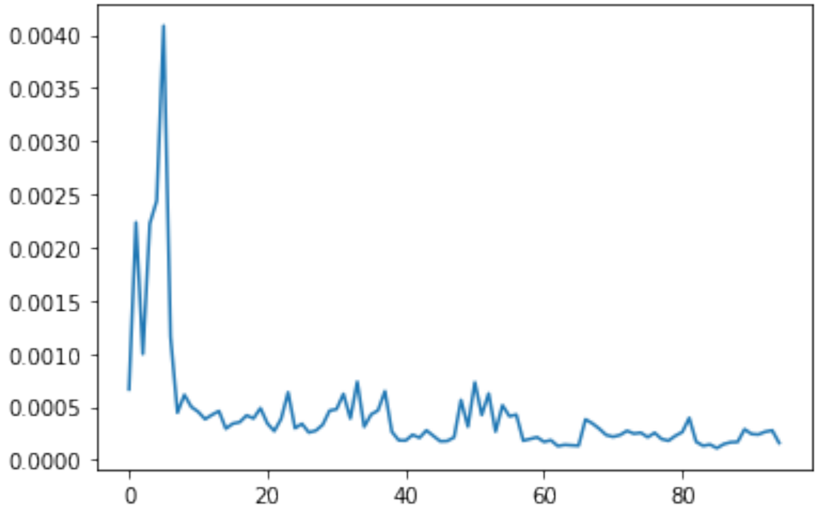


CV

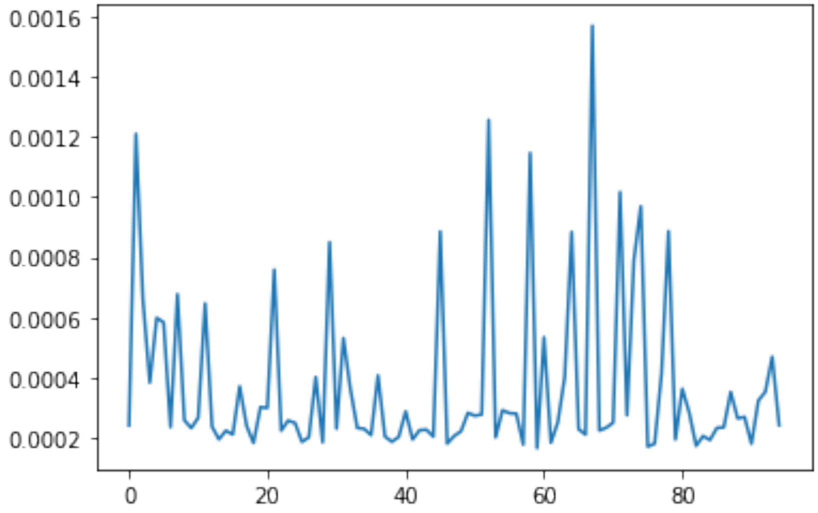


100-200 Epochs

Train



CV



Test Results

So far these are the best test results.

100 Epochs

Compare with Mean 1.81 for linear.

Mean	Median	Trimean	Best 25	Worst 25
1.658237	1.135563	1.495763	0.391509	3.877453

200 Epochs

Compare with mean 1.62 for linear.

Mean	Median	Trimean	Best 25	Worst 25
1.503393	0.997599	1.328497	0.288351	3.652337

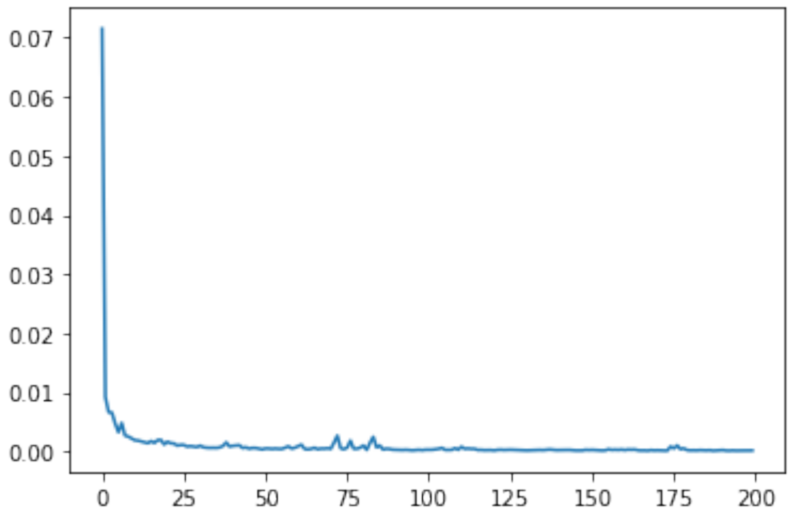
Linear * 2^{16} then Log

For this experiment I just trained 200 epochs in one run.

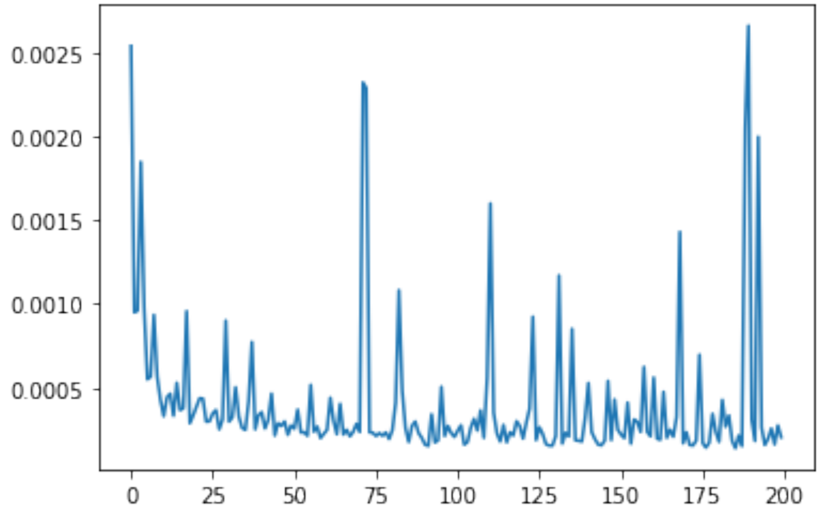
Loss Plots

200 Epochs

Train



CV



Test Results

An improvement over linear-log results.

200 Epochs

Compare with mean 1.50 for linear-log.

Mean	Median	Trimean	Best 25	Worst 25
1.420396	0.869077	1.242575	0.286096	3.530126

Aggregated Results

Baselines

Gray world

Mean: 6.36, Median: 6.28, TriMean: 6.28, Best25: 2.33, Worst25: 10.58

Contrastive Learning for Color Constancy (CLCC)

Mean: 1.84, Median: 1.31, TriMean: 1.42, Best25: 0.41, Worst25: 4.20

Googlenet triplet (from my assignment2, based off CLCC) - on Gray Ball and CUBE++

Mean: 1.51, Median: 1.049 TriMean: 1.418, Best25: 0.397, Worst25: 3.308

SqueezeNet-FC4 - on reprocessed Color Checker Dataset

Mean: 1.65, Median: 1.18, TriMean: 1.27, Best25: 0.38, Worst25: 3.78

Fast Fourier Color Constancy (FFCC) - on Gehler-Shi

Mean: 1.61, Median: 0.86, Trimean: 1.02, Best25: 0.23, Worst25: 4.23

FFCC - on Cheng

Mean: 1.99, Median: 1.31, Trimean: 1.43, Best25: 0.35, Worst25: 4.75

Illuminant Estimation Challenge 2 - General Track Leaderboard - CUBE++

Mean: 1.605, Median: 0.966, Trimean: 1.084, Best25:NA, Worst25: 4.084

Best published baseline scores:

FFCC - on Gehler-Shi

Mean: 1.61, Median: 0.86, Trimean: 1.02, Best25: 0.23, Worst25: 4.23

Scores in bold that exceed this baseline. Top scores for each metric are in red.

Norm	Epochs	Mean	Median	Trimean	Best 25	Worst 25
01	50	1.976820	1.254838	1.766730	0.428085	4.732106
linear	50	1.928371	1.305662	1.755973	0.414628	4.463303
log	50	1.946382	1.303034	1.732936	0.479687	4.582974
linear	100	1.810919	1.272023	1.648525	0.381628	4.250604
log	100	1.824393	1.164573	1.656479	0.324168	4.399156
linear-log	100	1.658237	1.135563	1.495763	0.391509	**3.877453**
linear	200	1.622245	1.034705	1.432307	0.280100	**4.033163**
log	200	1.738321	1.066977	1.579767	0.307531	4.316566
linear-log	200	**1.503393**	0.997599	1.328497	0.288351	**3.652337**
linear-expand-log	200	**1.420396**	**0.869077**	1.242575	0.286096	**3.530126**

Compared with the baselines, the linear-expand-log model trained for 200 epochs beats 3 of the 5 metrics of the top baseline.

Learning Curve Comparisons

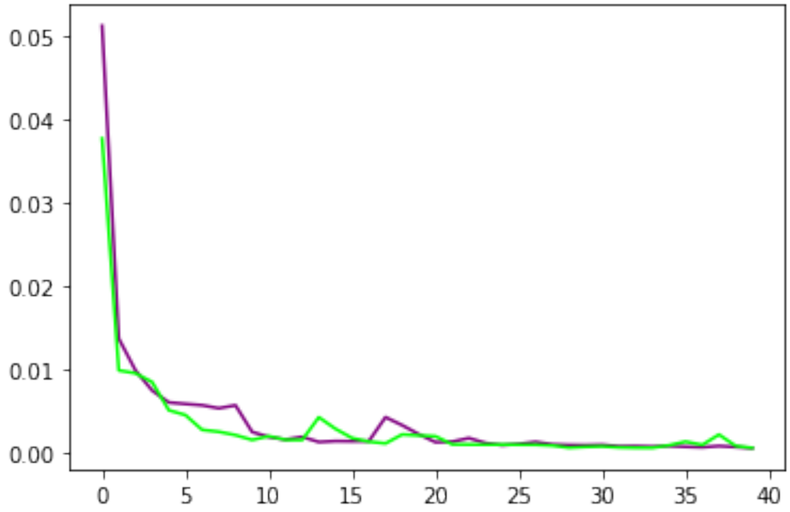
Linear and Linear-log

As can be seen, there is a general tendency for the log training curves to be a little more stable.

Train

linear = purple

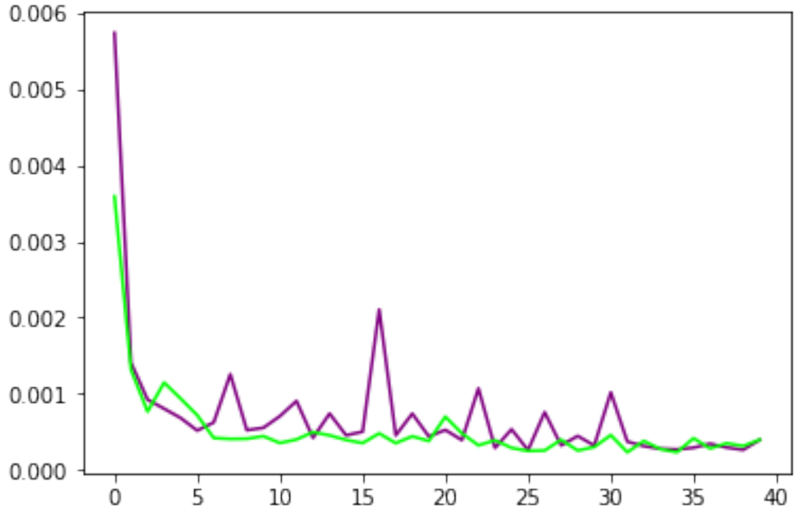
linear, log = green



CV

linear = purple

linear, log = green



No labels