Blur Kernel Estimation using Deep Learning

Arun Patro | 13EE35026

Under the Supervision of
Dr. Rajiv Ranjan Sahay

Introduction
Blur

1. Blur is a distortion / degradation of images which results in unclear images that have lost features.
2. They can be classified into
   a. Defocus (optical)
   b. Motion

Motion Blur due to moving object

Introduction
Blur

1. Blur is a distortion / degradation of images which results in unclear images that have lost features.
2. They can be classified into
   a. Defocus (optical)
   b. Motion

Motion Blur due to camera jitter

Introduction
Blur

1. Blur is a distortion / degradation of images which results in unclear images that have lost features.
2. They can be classified into
   a. Defocus (optical)
   b. Motion

Defocus Blur due to incorrect focus in 3-D

Introduction

Uniform vs. Non-uniform Blur

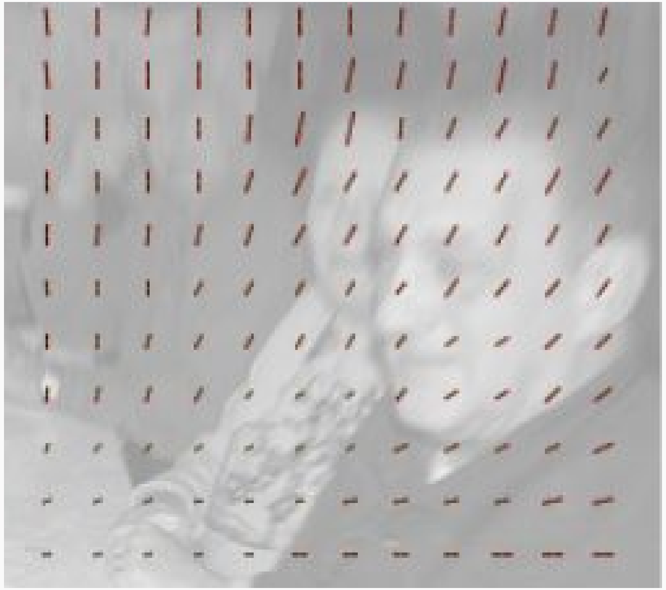
Uniform

Non-uniform

Introduction

Uniform vs. Non-uniform Blur

Uniform
(length = 10, angle = 45)



Non-uniform

Introduction

Image Deblurring

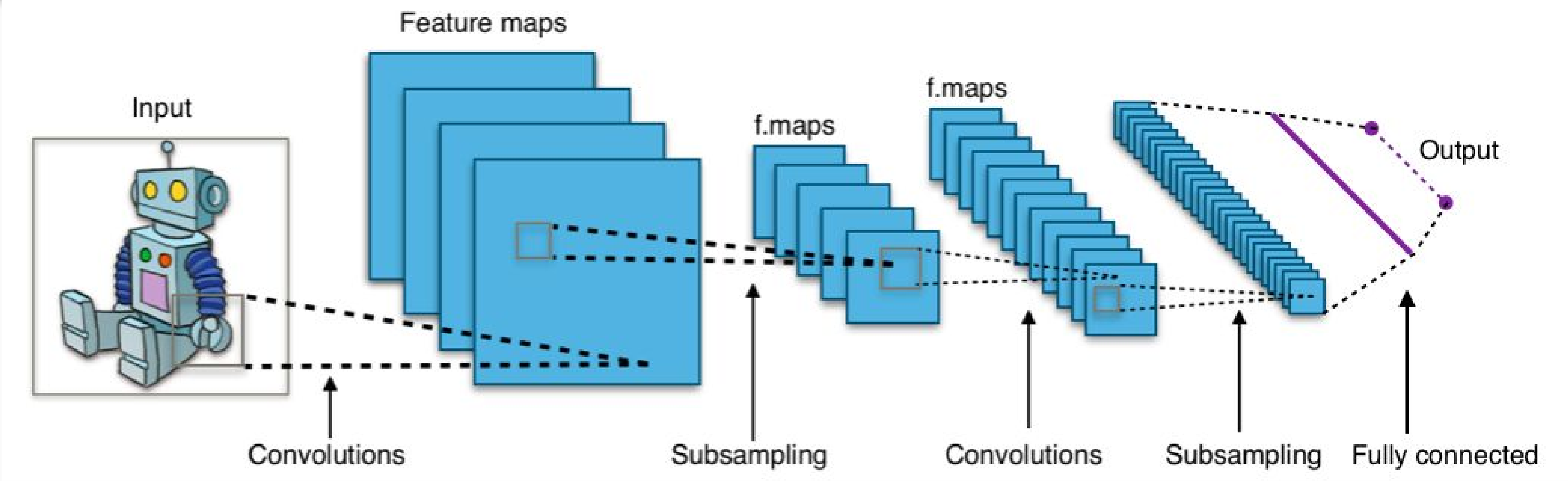
1. It is very challenging for a computer to do unsupervised Deblurring.

2. Blind Deconvolution techniques are severely under constrained as we need need estimate **h** and **F** given **G**.

3. Blind Deconvolution works by first estimating the **h** also known as Point Spread function and then inverting using convolution or conjugate gradient methods.

$$\mathbf{G}(x,y) = h(x,y)*\mathbf{F}(x,y) + n(x,y)$$

Blur Kernel Estimation using Deep Learning



Aim and Approach

AIM:

Estimating the non uniform blur kernel at each pixel using Convolutional Neural Networks (CNNs). Generating the sigma-map for a variantly blurred image.

APPROACH:

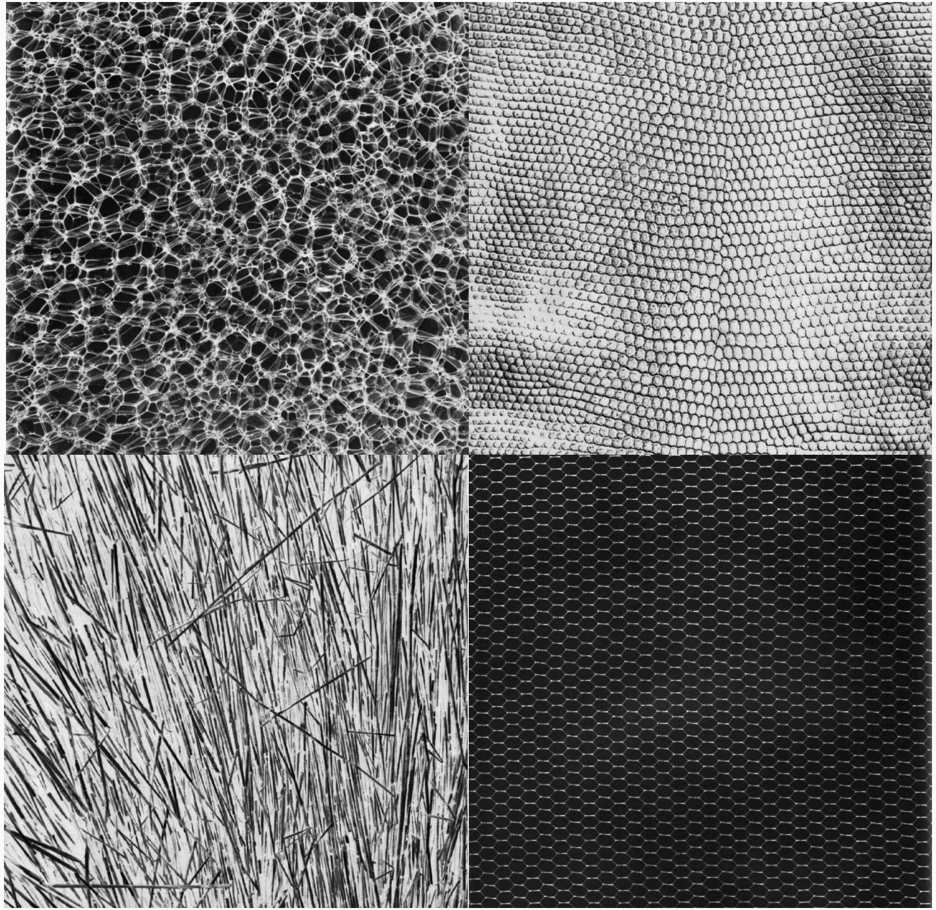
We train the CNNs to learn the sigma value of different gaussian blur kernels that model defocus blur. We start with training the CNN for 32x32 size patches and gradually reduce the patch size down to 1 pixel.

Dataset

1. **We used images from the Brodatz Textured Dataset**

2. Each Image was blurred with a gaussian blur kernel with sigma from [0.3,0.6, … ,3.0] taking any of the 10 discrete values.

3. 32x32 size non-overlapping patches were sampled from these images to form the dataset.



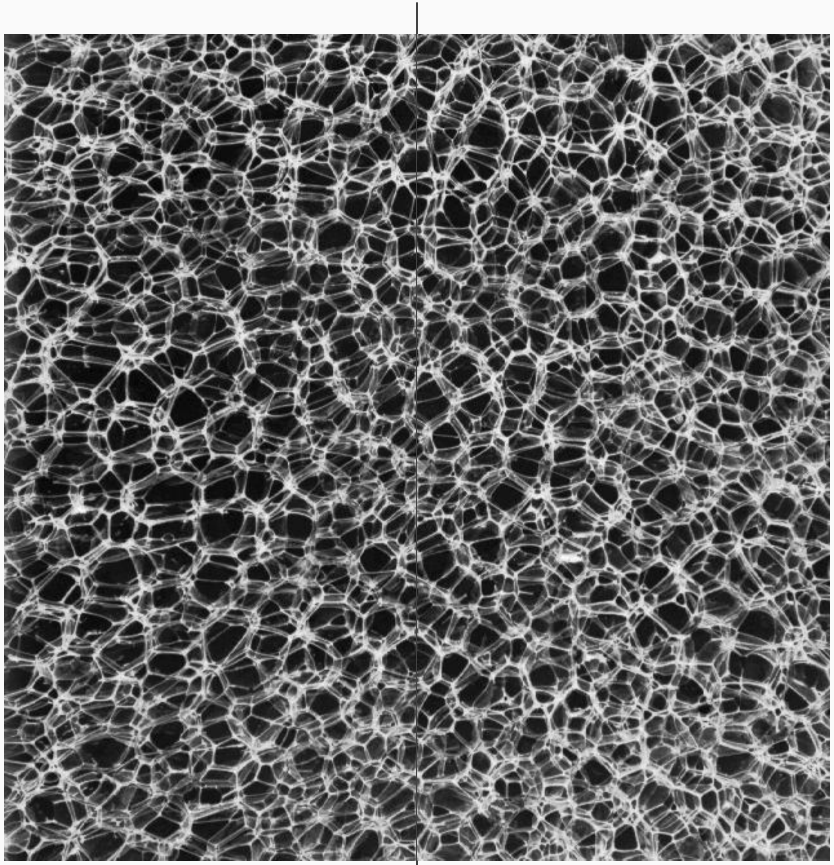
Brodatz Textured Dataset

Dataset

1. We used images from the Brodatz Textured Dataset

2. **Each Image was blurred with a gaussian blur kernel with sigma from [0.3,0.6, … ,3.0] taking any of the 10 discrete values.**

3. 32x32 size non-overlapping patches were sampled from these images to form the dataset.



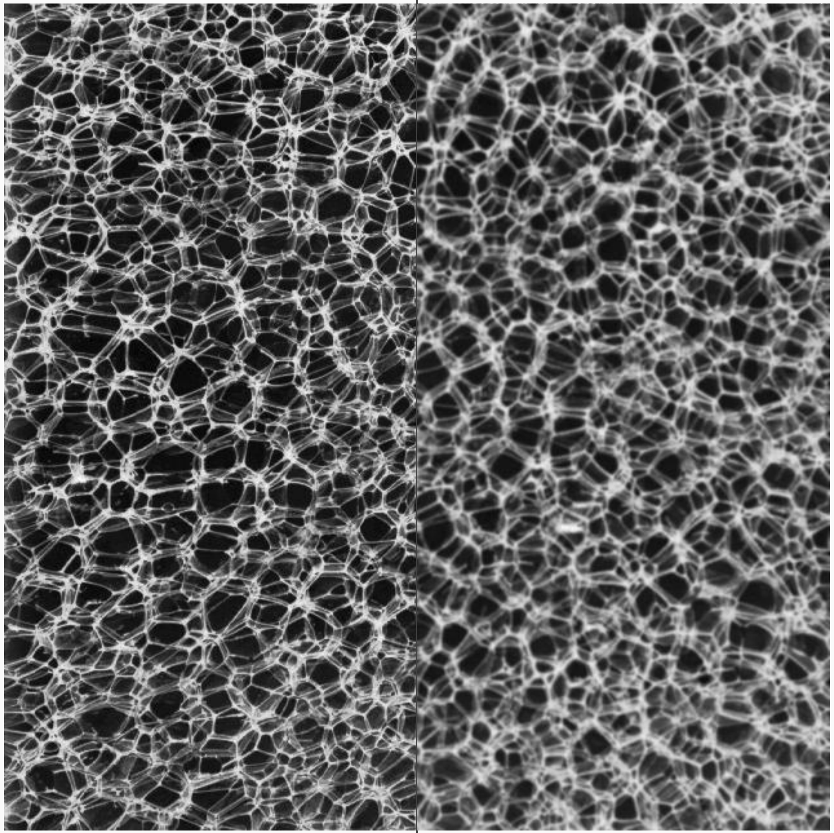
Unblurred	Sigma = 0.3

Dataset

1. We used images from the Brodatz Textured Dataset

2. **Each Image was blurred with a gaussian blur kernel with sigma from [0.3,0.6, … ,3.0] taking any of the 10 discrete values.**

3. 32x32 size non-overlapping patches were sampled from these images to form the dataset.



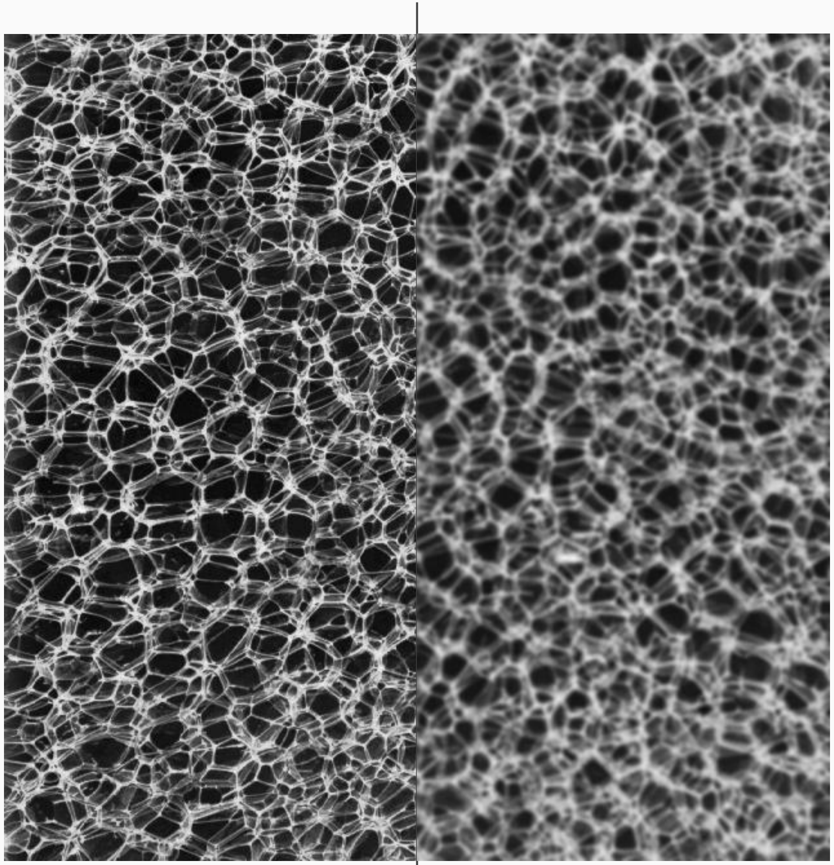
Unblurred	Sigma = 0.9

Dataset

1. We used images from the Brodatz Textured Dataset

2. **Each Image was blurred with a gaussian blur kernel with sigma from [0.3,0.6, … ,3.0] taking any of the 10 discrete values.**

3. 32x32 size non-overlapping patches were sampled from these images to form the dataset.



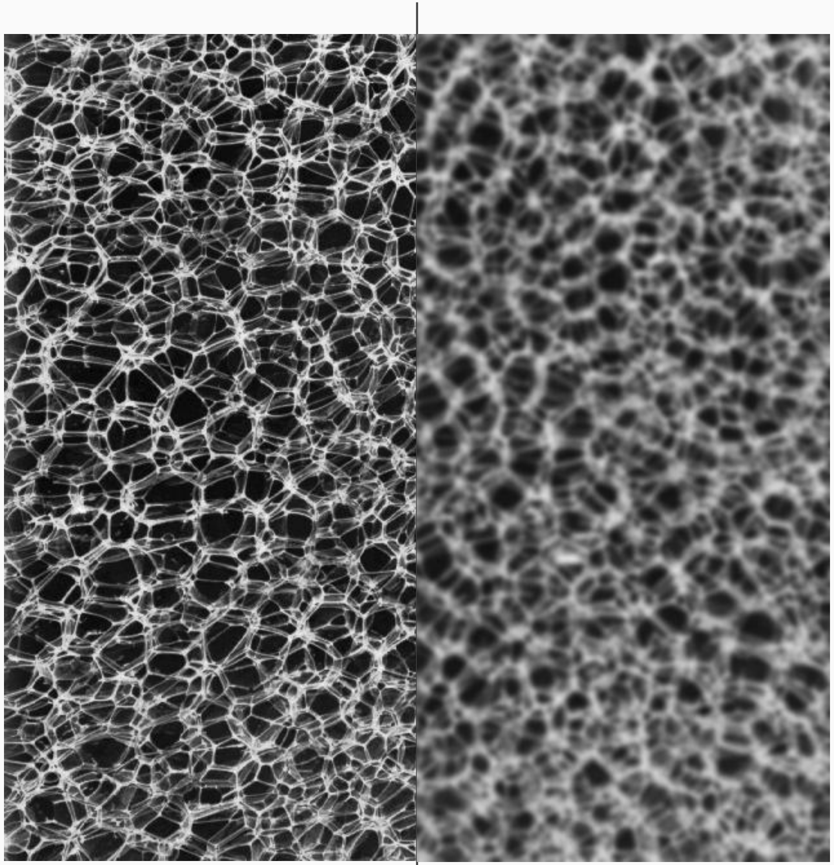
Unblurred	Sigma = 1.5

Dataset

1. We used images from the Brodatz Textured Dataset

2. **Each Image was blurred with a gaussian blur kernel with sigma from [0.3,0.6, … ,3.0] taking any of the 10 discrete values.**

3. 32x32 size non-overlapping patches were sampled from these images to form the dataset.



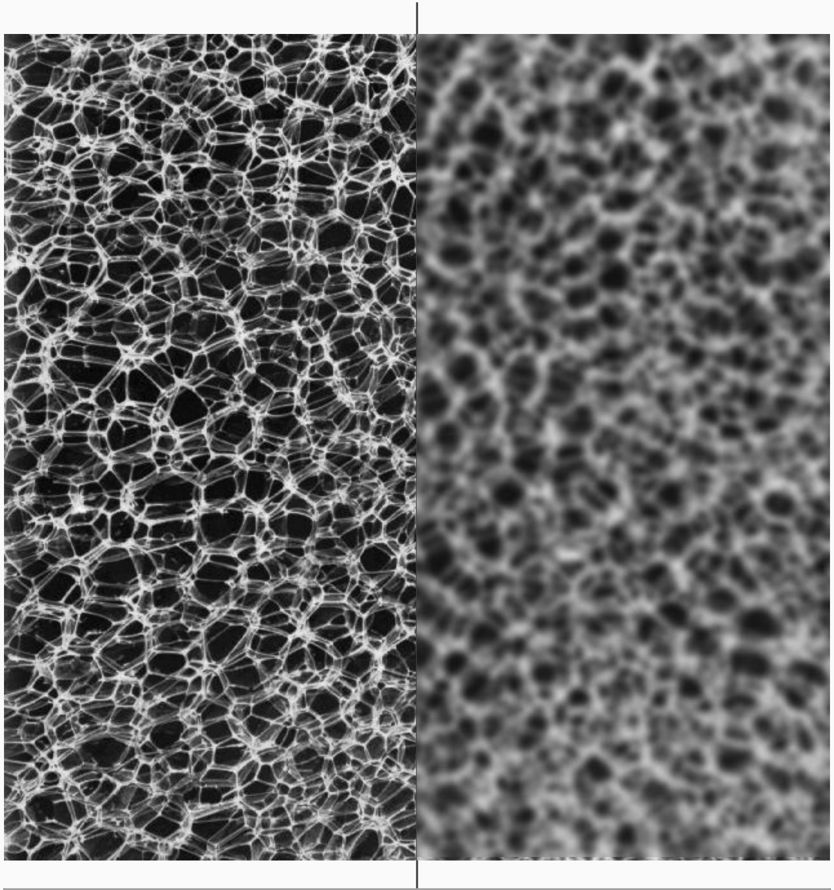
Unblurred	Sigma = 2.1

Dataset

1. We used images from the Brodatz Textured Dataset

2. **Each Image was blurred with a gaussian blur kernel with sigma from [0.3,0.6, … ,3.0] taking any of the 10 discrete values.**

3. 32x32 size non-overlapping patches were sampled from these images to form the dataset.



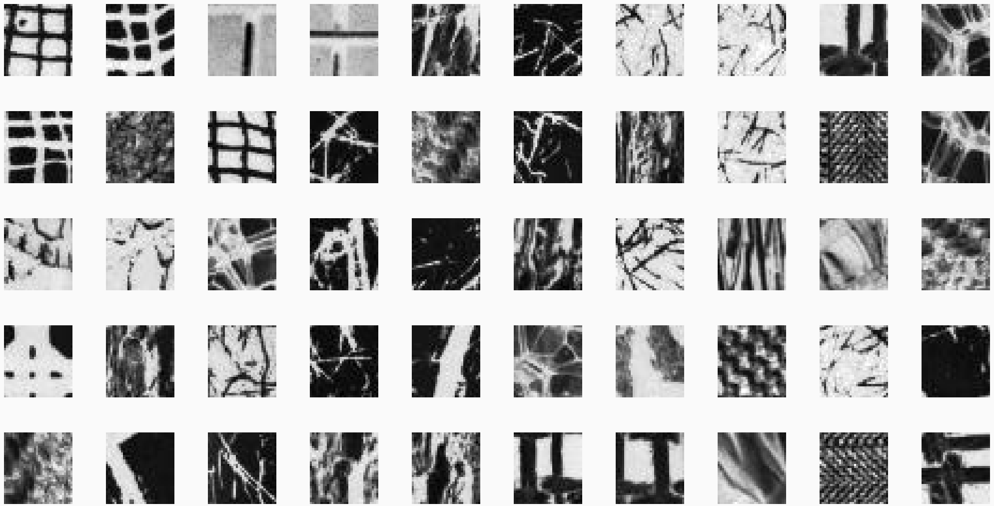
Unblurred	Sigma = 3.0

Dataset

1. We used images from the Brodatz Textured Dataset

2. Each Image was blurred with a gaussian blur kernel with sigma from [0.3,0.6, … ,3.0] taking any of the 10 discrete values.

3. **32x32 size non-overlapping patches were sampled from these images to form the dataset.**



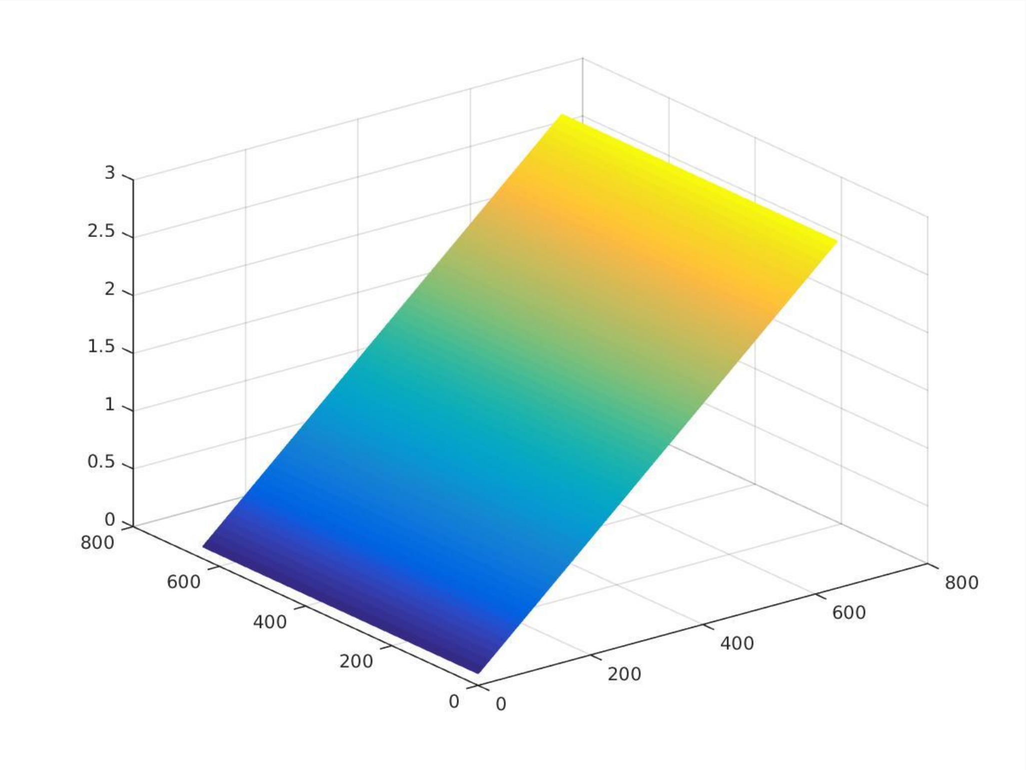
32x32 non overlapping patches taken from blurred images

Network Architecture

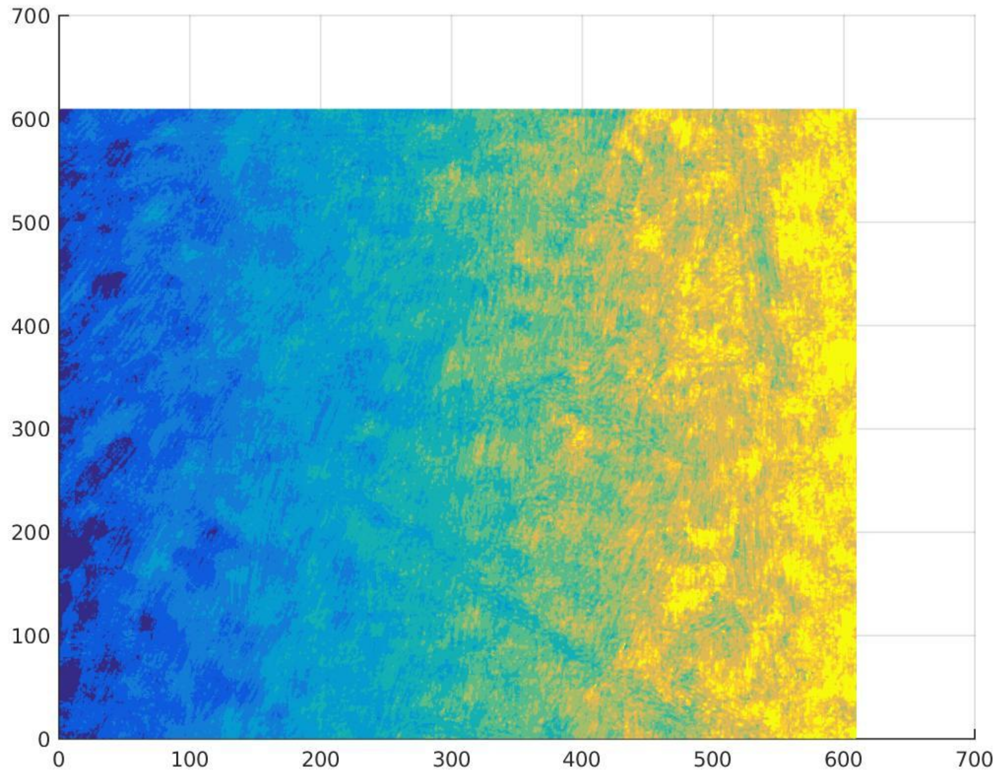
```
th> model
nn.Sequential {
  [input -> (1) -> (2) -> (3) -> (4) -> (5) -> (6) -> (7) -> (8) -> (9) -> (10) -> (11) -> (12) -> (13) -> (14) -> (15) -> (16) -> (17) -> (18)
 -> (19) -> (20) -> (21) -> (22) -> (23) -> (24) -> (25) -> (26) -> output]
  (1): cudnn.SpatialConvolution(1 -> 4, 3x3)
  (2): cudnn.ReLU
  (3): cudnn.SpatialConvolution(4 -> 8, 3x3)
  (4): cudnn.ReLU
  (5): cudnn.SpatialConvolution(8 -> 16, 3x3)
  (6): cudnn.ReLU
  (7): cudnn.SpatialConvolution(16 -> 32, 3x3)
  (8): cudnn.ReLU
  (9): cudnn.SpatialConvolution(32 -> 64, 3x3)
  (10): cudnn.ReLU
  (11): cudnn.SpatialConvolution(64 -> 64, 3x3)
  (12): cudnn.ReLU
  (13): cudnn.SpatialConvolution(64 -> 64, 5x5)
  (14): cudnn.ReLU
  (15): cudnn.SpatialMaxPooling(2x2, 2,2)
  (16): cudnn.SpatialConvolution(64 -> 64, 5x5)
  (17): cudnn.ReLU
  (18): nn.View(1024)
  (19): nn.Linear(1024 -> 1600)
  (20): nn.ReLU
  (21): nn.Linear(1600 -> 800)
  (22): nn.ReLU
  (23): nn.Linear(800 -> 100)
  (24): nn.ReLU
  (25): nn.Linear(100 -> 30)
  (26): nn.LogSoftMax
}
```

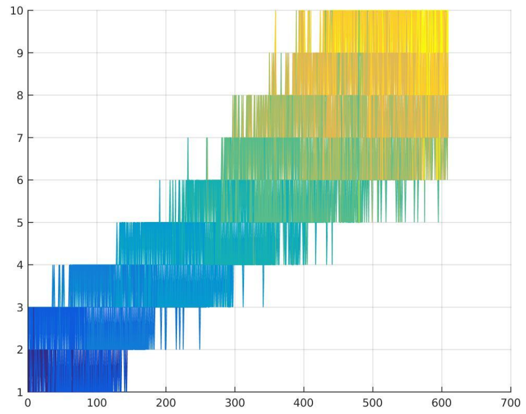
Textured Images

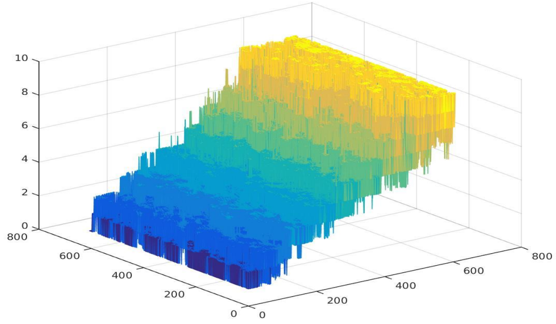
Ramp Blur



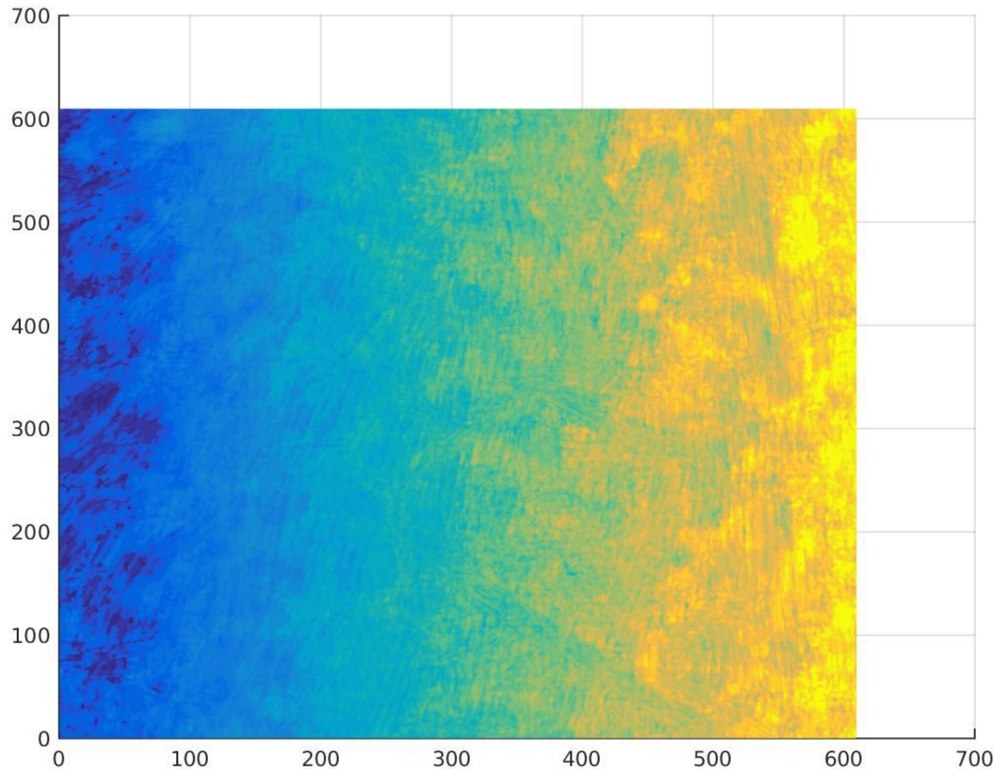
Textured Images Ramp Blur RMSE: 0.2350

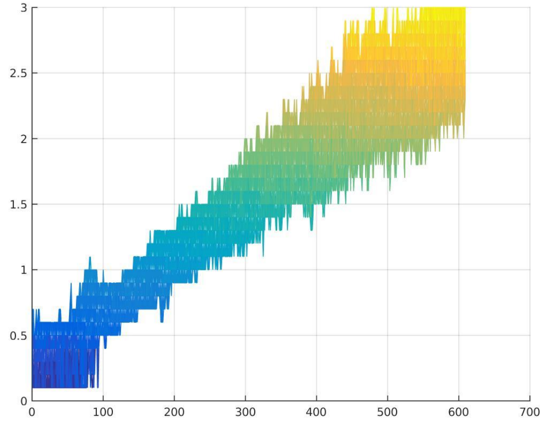


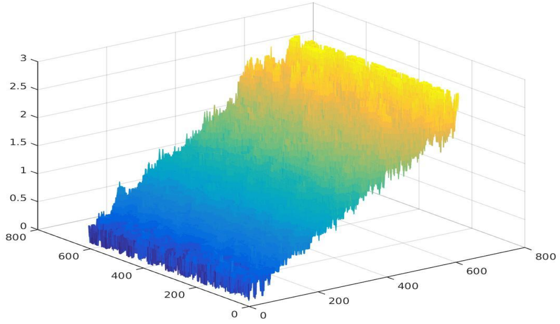




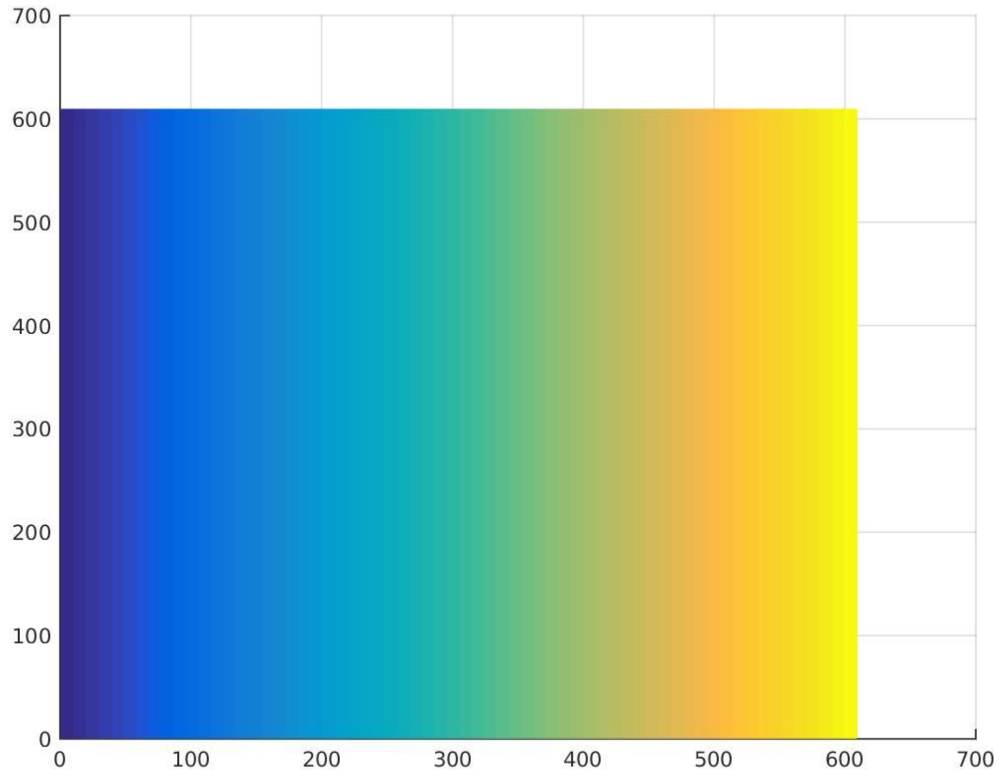
Textured Images Ramp Blur RMSE: 0.1645

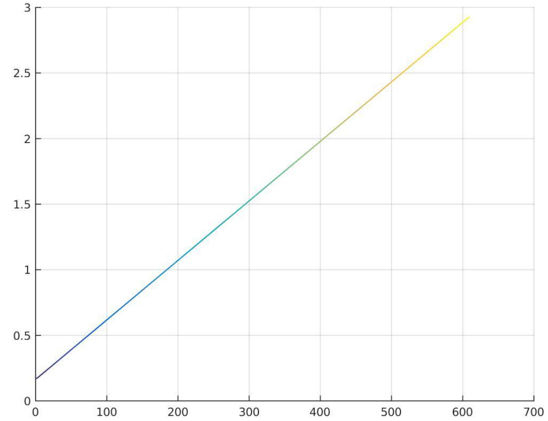


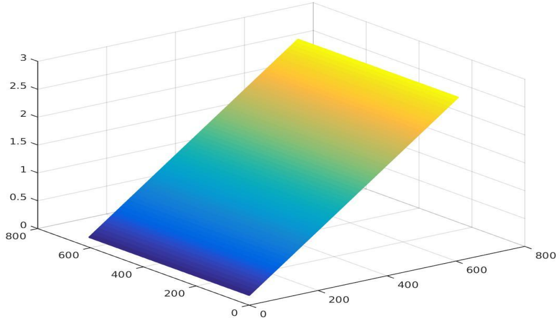




Textured Images Ramp Blur RMSE: 0.000116





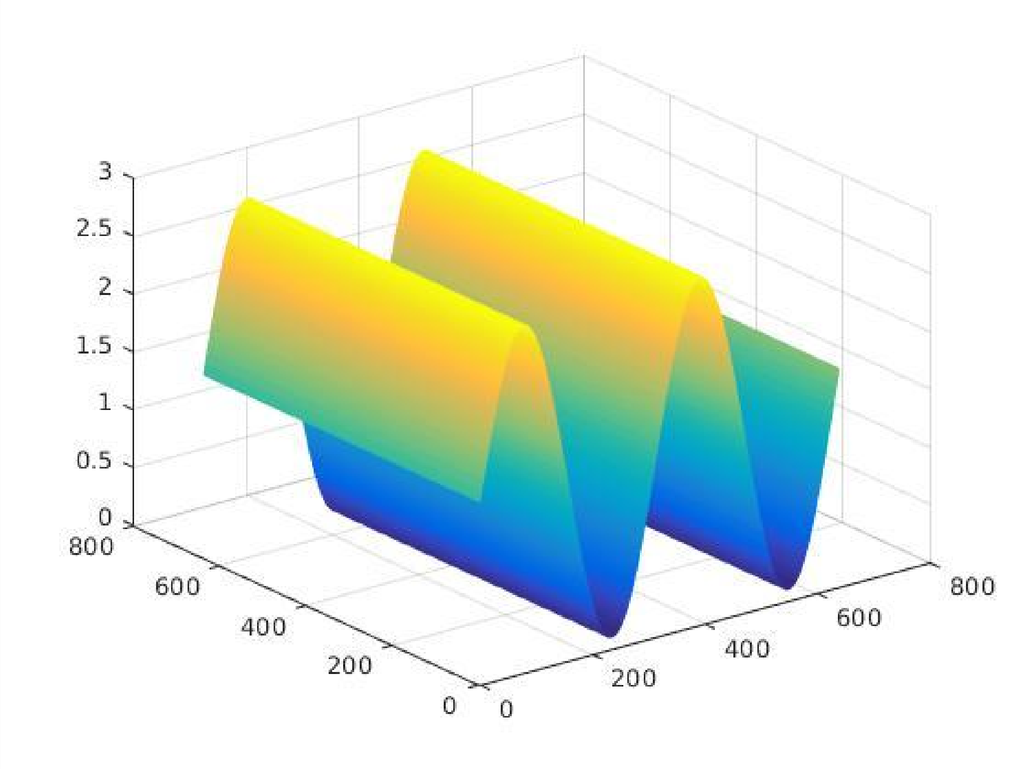


Accuracy

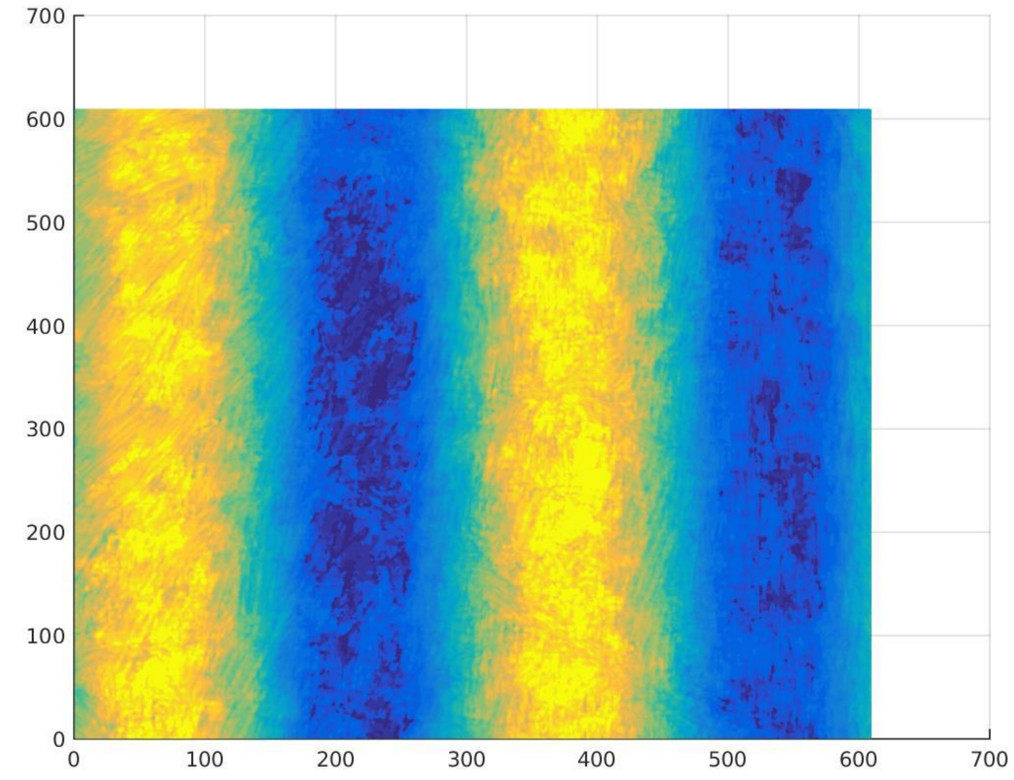
Classes	Sigma Step Size	MSE	RMSE
10	0.3	0.0552	0.2350
30	0.1	0.0270	0.1645
		1.3506e-08	1.1621e-04

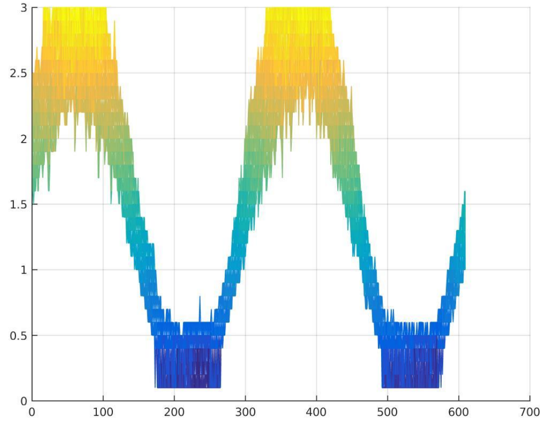
Textured Images

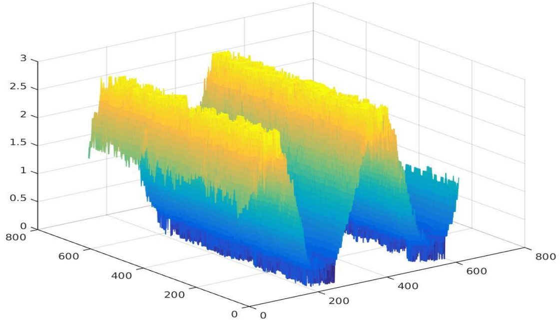
Sin Blur



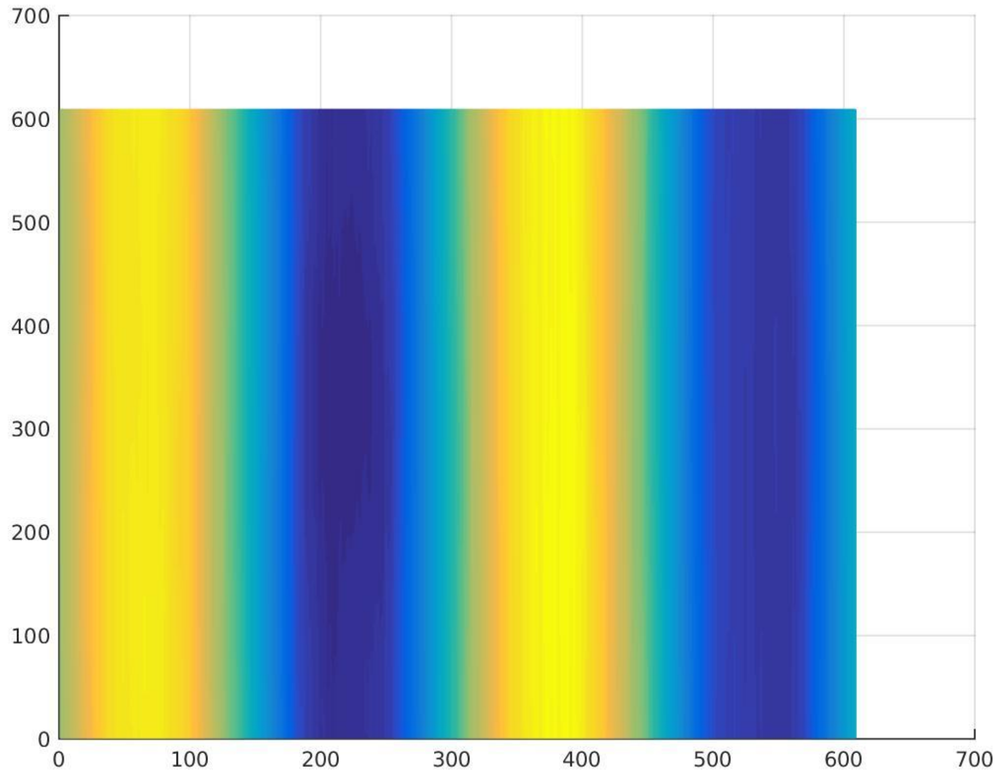
Textured Images Sin Blur RMSE: 0.1991

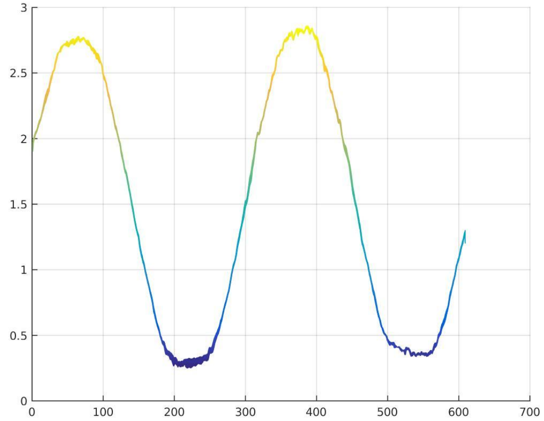


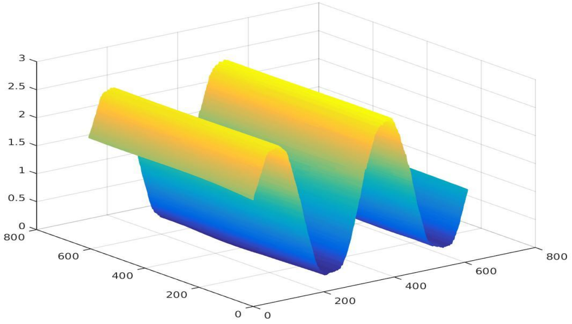




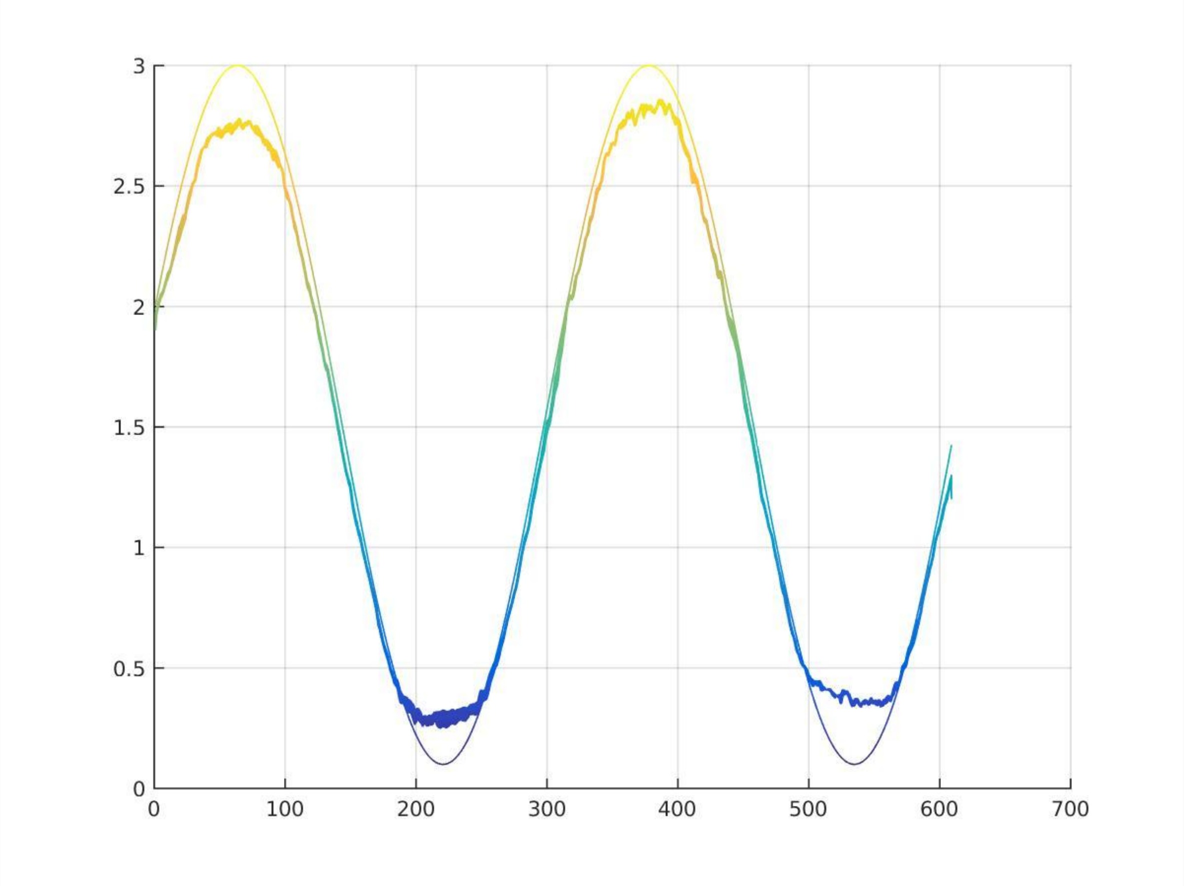
Textured Images   Sin Blur   RMSE: 0.1331







Difference

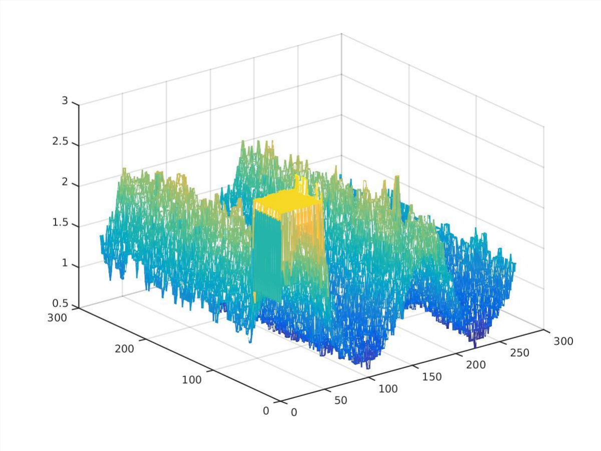


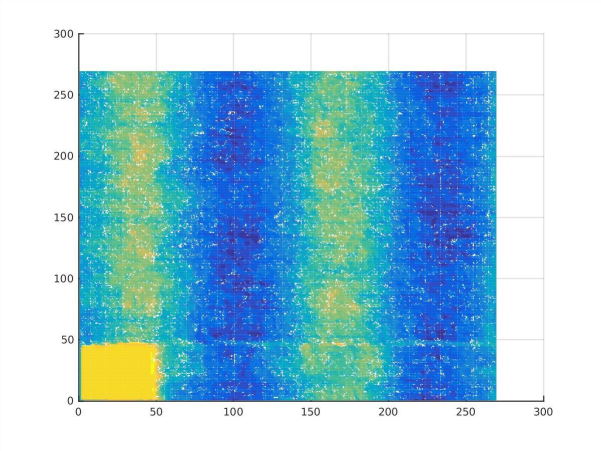
Accuracy

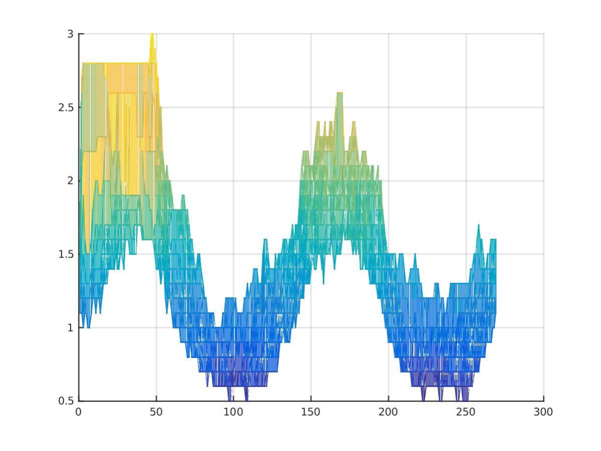
Smoothing iterations	MSE	RMSE
0	0.0397	0.1991
1	0.0357	0.1891
12000	0.0177	0.1331

Text Images

1-D Sin Blur

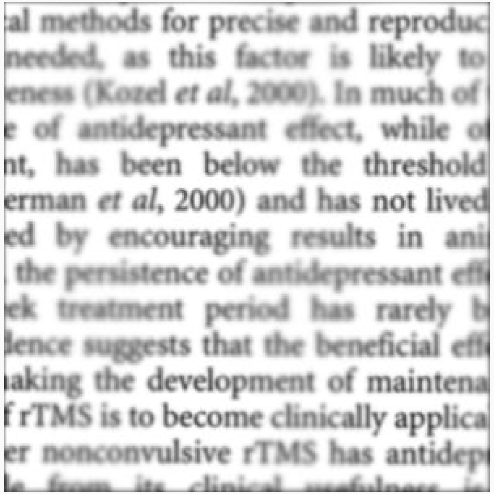


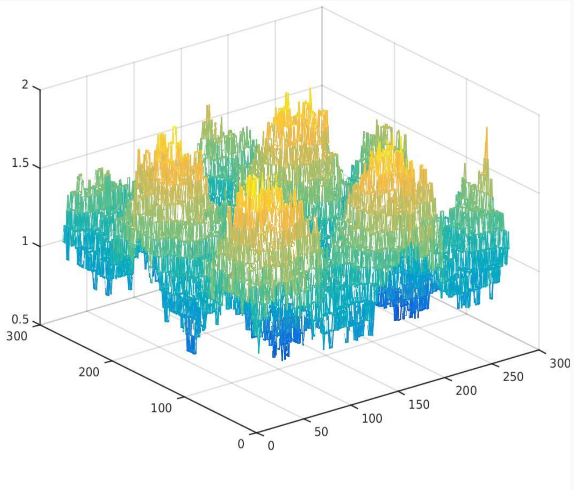


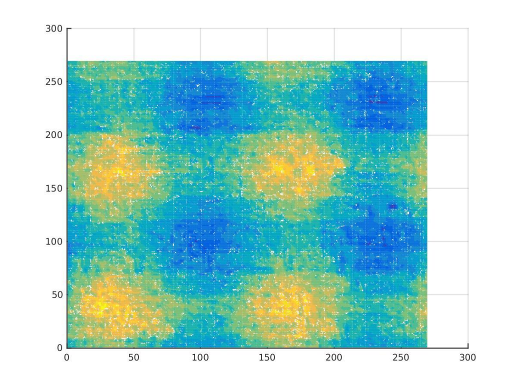


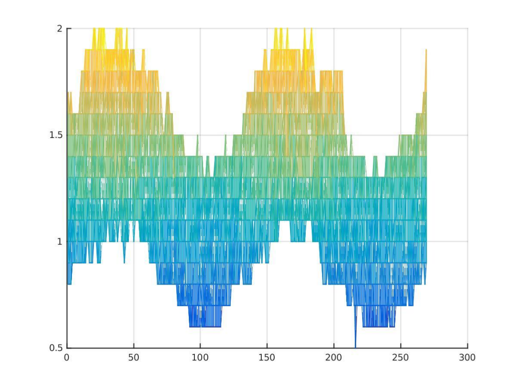
Text Images

2-D Sin Blur

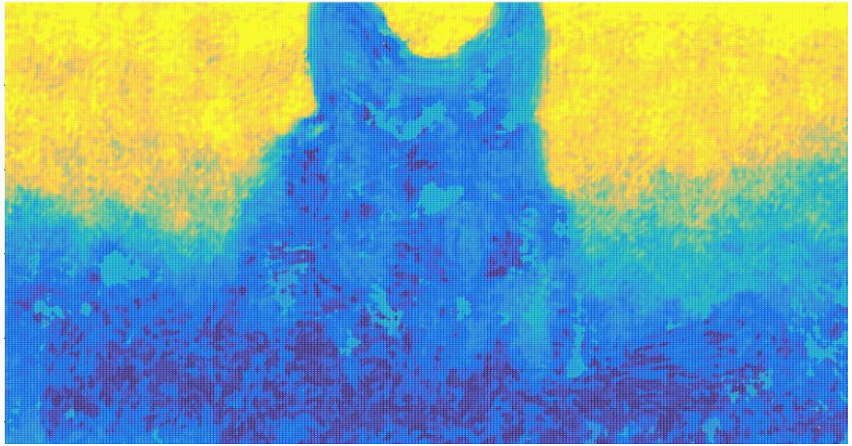

al methods for precise and reproduc
needed, as this factor is likely to
eness (Kozel *et al*, 2000). In much of
e of antidepressant effect, while o
nt, has been below the threshold
erman *et al*, 2000) and has not lived
ed by encouraging results in ani
, the persistence of antidepressant eff
ek treatment period has rarely b
lence suggests that the beneficial eff
aking the development of maintena
f rTMS is to become clinically applica
er nonconvulsive rTMS has antidep
le from its clinical usefulness is



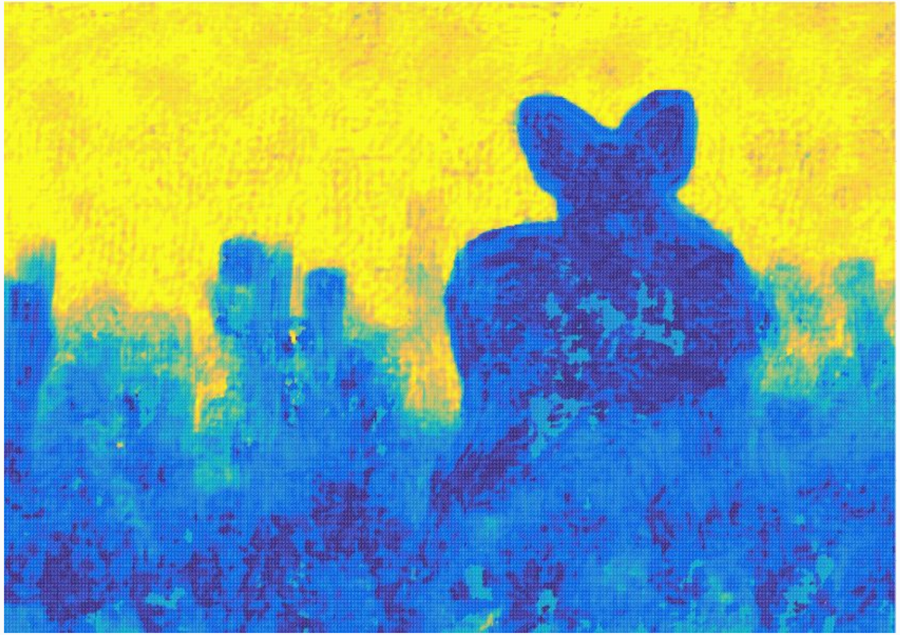




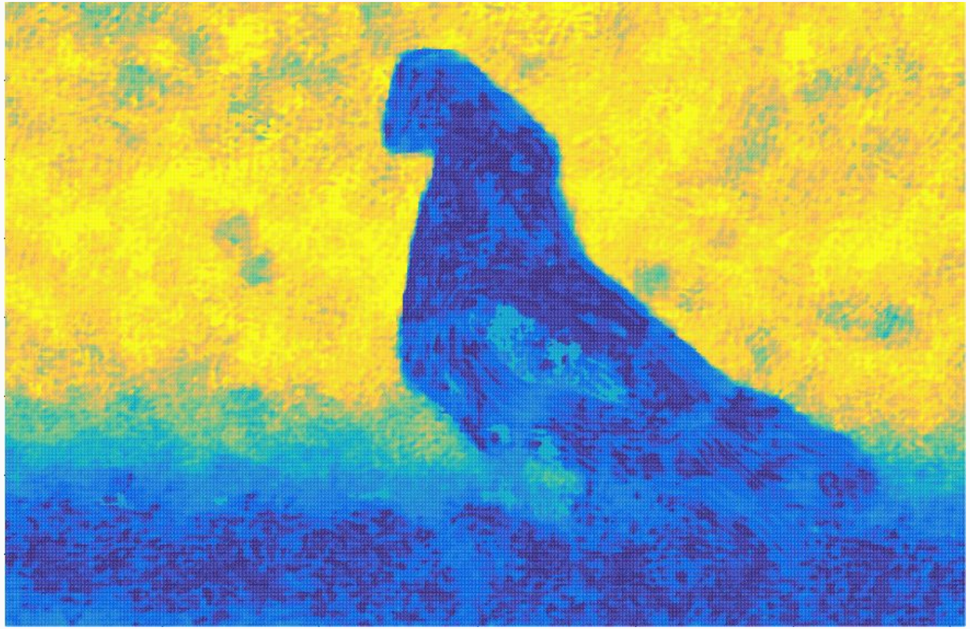
Natural images with defocused background



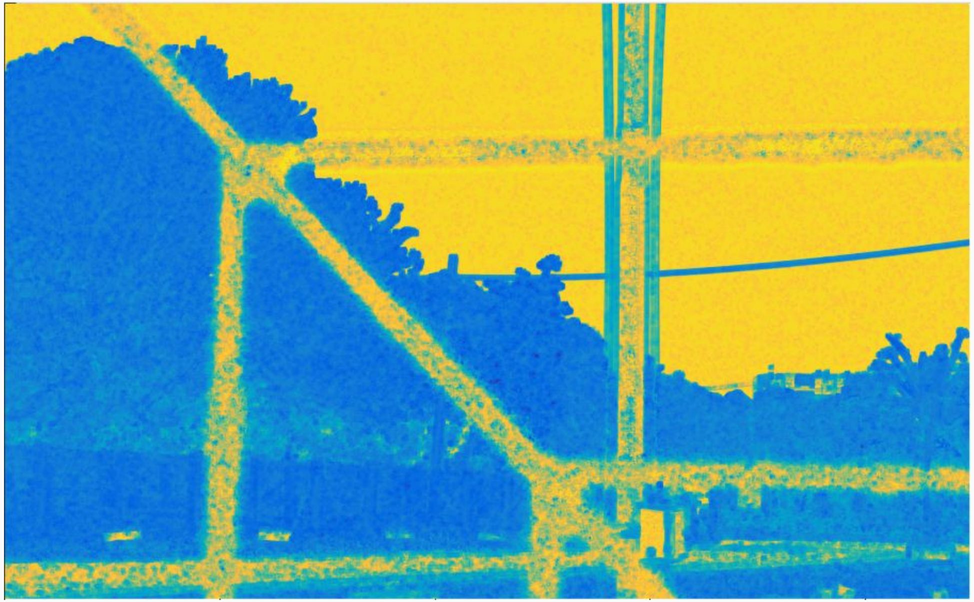
Natural images with defocused background



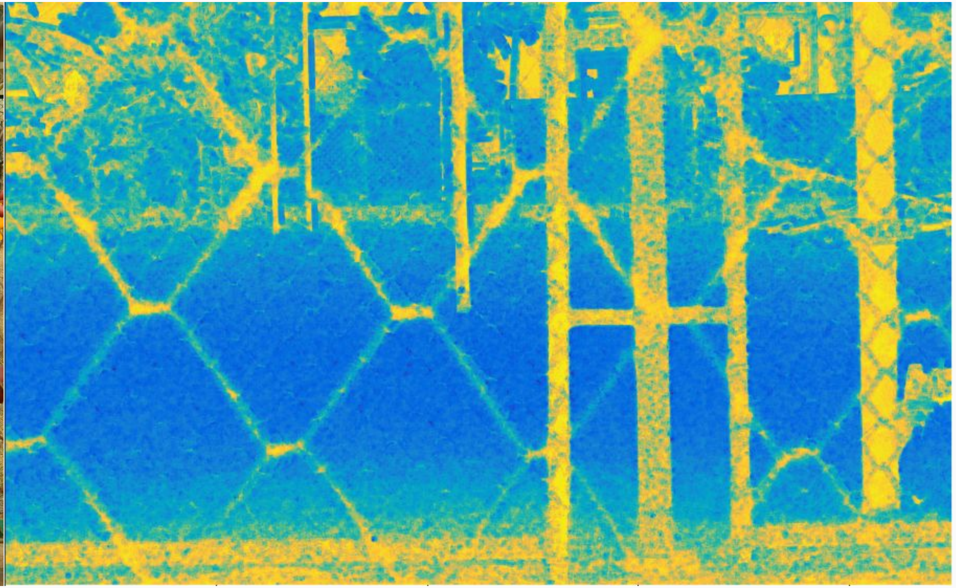
Natural images with defocused background



Natural images with defocused foreground



Natural images with defocused foreground



Conclusion

1. We see that the model is able to learn 30 classes and gives a coarse prediction of the sigma map.
2. We can getter better results if we reduce the quantization level to 60 classes.
3. Ideally, Blur Kernel Estimation is a regression problem. In the future we hope to use Fully Convolutional Networks.

Thank You