Synthetic corruption of images for anomaly detection using autoencoders

[ISPG seminar]

Anne-Sophie Collin

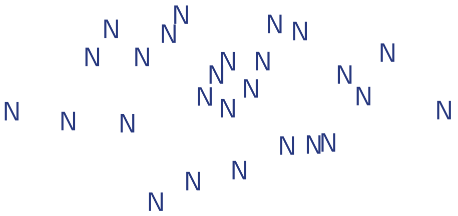
Supervisor
Christophe De Vleeschouwer

3rd of JUNE 2020

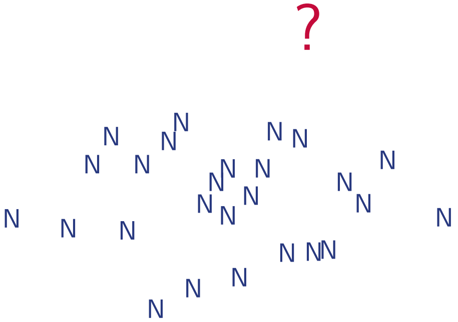
Anomaly Detection

N

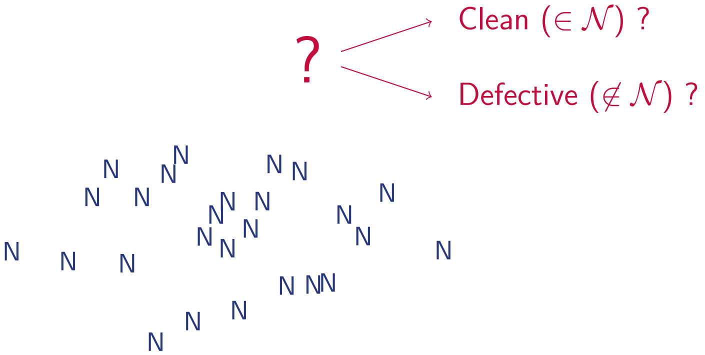
Anomaly Detection



Anomaly Detection



Anomaly Detection

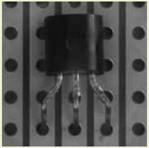



Clean
Clean
Defective
Clean
Defective
Clean
Clean
Defective
Clean
Clean
Defective
Clean
Clean
Clean
Defective

Defective

Clean

Clean

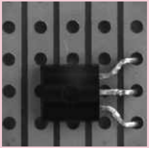


Clean

Clean

Clean

Defective



Clean

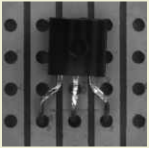
Defective

Defective

Defective

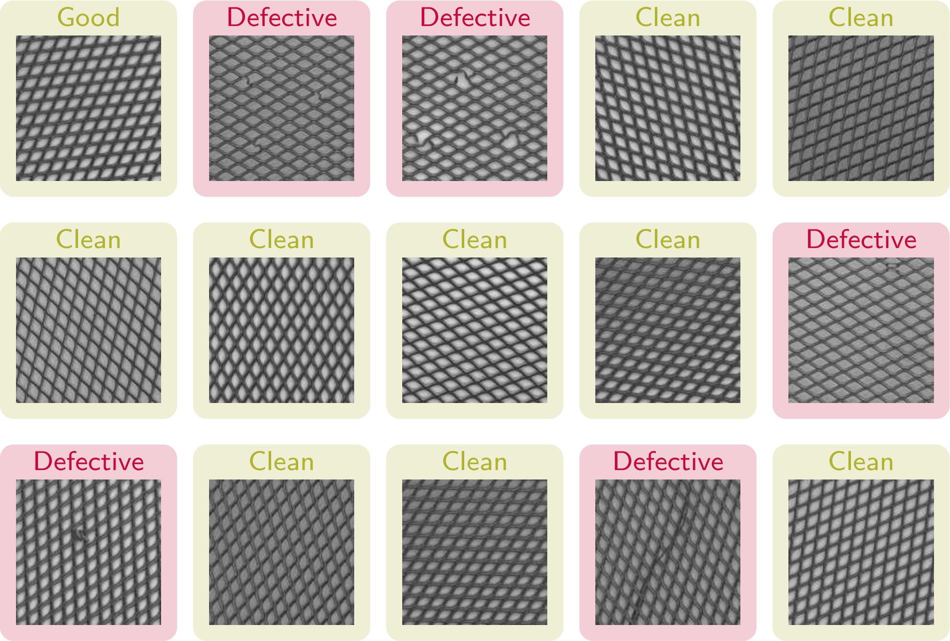
Clean

Clean



Clean

Defective


Good
Defective
Defective
Clean
Clean
Clean
Clean
Clean
Clean
Defective
Defective
Clean
Clean
Defective
Clean

Clean

Clean

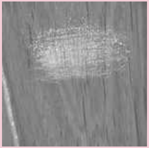
Defective

Defective

Clean

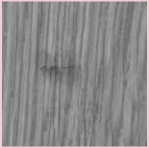
Clean

Defective



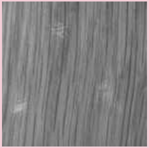
Clean

Defective

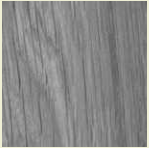


Defective

Defective



Clean



Defective

Clean

Clean

Approaches

Classification Based

Multi-class

Reconstruction Based

Out-of-Distribution Based

One-class

Approaches

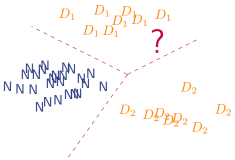
Classification Based

Multi-class

Reconstruction Based

Out-of-Distribution Based

One-class



$$p({\color{red}?} \in \mathcal{N}) > p({\color{red}?} \in \mathcal{D}_i)$$

Approaches

Classification Based

Multi-class

Reconstruction Based

Out-of-Distribution Based

One-class

Not suited, why?

- The scarcity of abnormal events makes normal vs abnormal classes unbalanced
- The result is unreliable for defective samples that are not sampled from one of the defective classes considered during training

Approaches

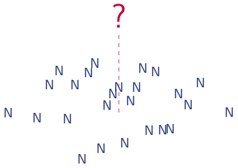
Classification Based

Multi-class

Reconstruction Based

Out-of-Distribution Based

One-class



$$||? - \text{proj}_{\mathcal{N}}?|| < \tau$$

Approaches

Classification Based

Multi-class

Reconstruction Based

Out-of-Distribution Based

One-class

Project the sample onto the normal space and compare the input it with its projection.

Approaches

Classification Based

Multi-class

Reconstruction Based

Out-of-Distribution Based

One-class

Project the sample onto the normal space and compare the input it with its ~~projection~~.

reconstruction.

Difficult, why?

- Restrict the reconstruction to lie onto normal space exclusively
- Fix the rejection threshold

Approaches

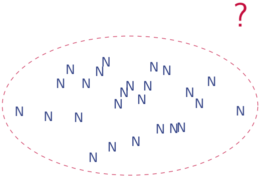
Classification Based

Multi-class

Reconstruction Based

Out-of-Distribution Based

One-class



$$p({\color{red}?} \in \mathcal{N}) < \tau$$

Approaches

Classification Based | Reconstruction Based | Out-of-Distribution Based

Multi-class | One-class

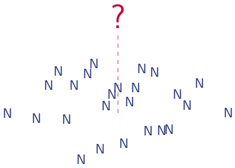
Difficult, why?

- Characterize the distribution of the normal class
- Fix the rejection threshold

Our Method

Classification Based | Reconstruction Based | Out-of-Distribution Based

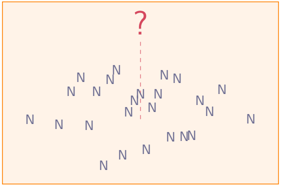
Multi-class | One-class



$$||? - \text{proj}_{\mathcal{N}}?|| < \tau$$

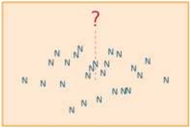
Reconstruct a clean image

Classification Based	Reconstruction Based	Out-of-Distribution Based
Multi-class	One-class	



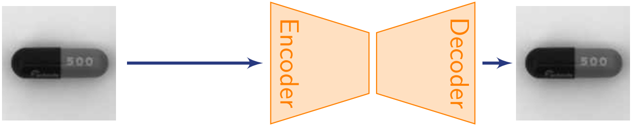
$$||? - \text{proj}_{\mathcal{N}} ?|| < \tau$$

Reconstruct a clean image (1)

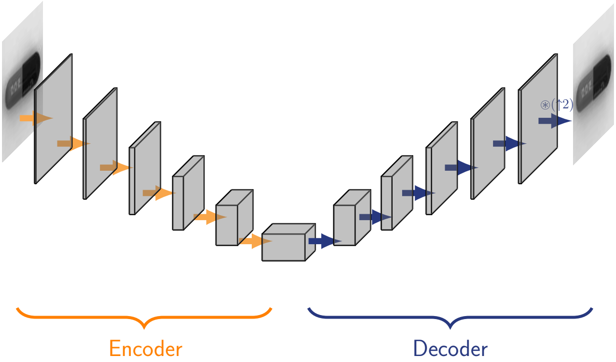


Model 1 (Baseline) :

Train an Autoencoder (AE) to perform an identity mapping
$\rightarrow$ Only clean images in the training set



Architecture of the autoencoder (AE)

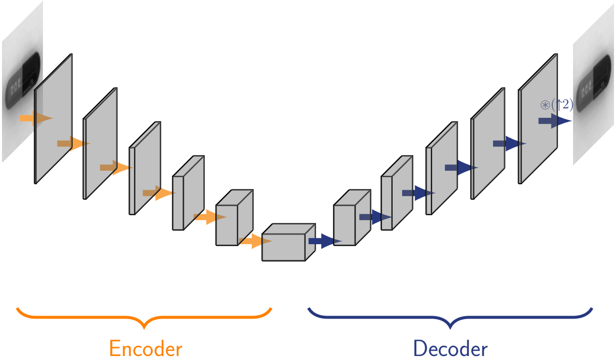


Strided convolution (5x5) + BN + LeakyReLU

Upsampling ($\uparrow 2$) + Convolution (5x5) + BN + LeakyReLU

$\circledast(\uparrow 2)$ Upsampling ($\uparrow 2$) + Convolution (5x5) + Sigmoid

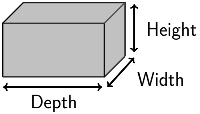
Architecture of the autoencoder (AE)



Strided convolution (5x5) + BN + LeakyReLU

Upsampling (↑2) + Convolution (5x5) + BN + LeakyReLU

⊛(↑2) Upsampling (↑2) + Convolution (5x5) + Sigmoid



Convolutional layer

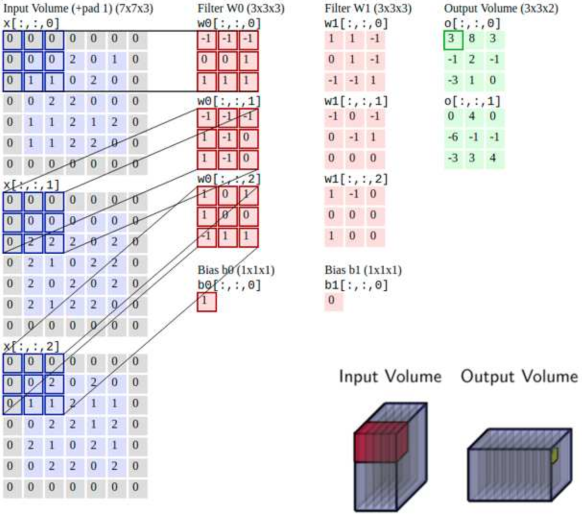


Image source CS231n: Convolutional Neural Networks for Visual Recognition 2020, Fei-Fei Li and Andrej Karpathy and Justin Johnson, http://cs231n.stanford.edu/

Convolutional layer

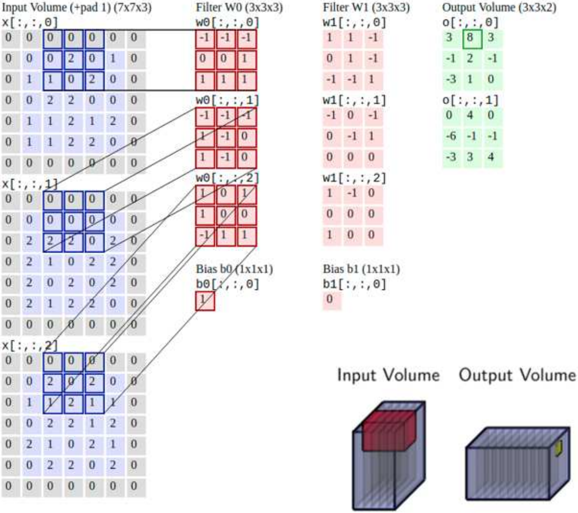


Image source CS231n: Convolutional Neural Networks for Visual Recognition 2020, Fei-Fei Li and Andrej Karpathy and Justin Johnson, `http://cs231n.stanford.edu/`

Convolutional layer

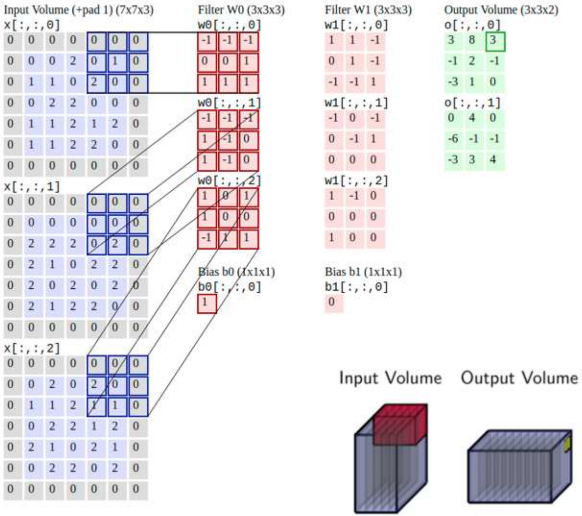


Image source CS231n: Convolutional Neural Networks for Visual Recognition 2020, Fei-Fei Li and Andrej Karpathy and Justin Johnson, `http://cs231n.stanford.edu/`

Convolutional layer

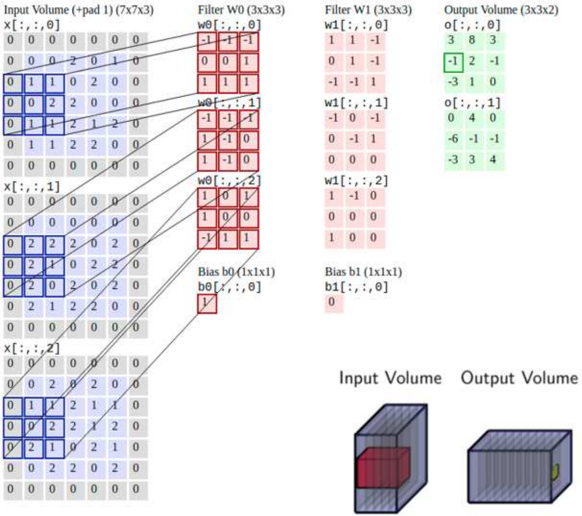


Image source CS231n: Convolutional Neural Networks for Visual Recognition 2020, Fei-Fei Li and Andrej Karpathy and Justin Johnson, http://cs231n.stanford.edu/

Convolutional layer

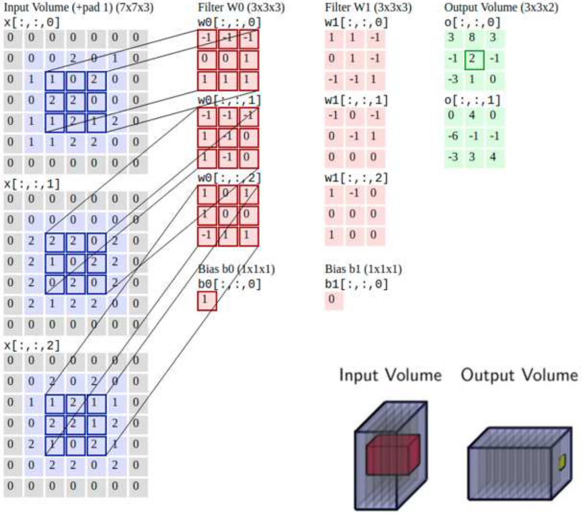


Image source CS231n: Convolutional Neural Networks for Visual Recognition 2020, Fei-Fei Li and Andrej Karpathy and Justin Johnson, http://cs231n.stanford.edu/

Convolutional layer

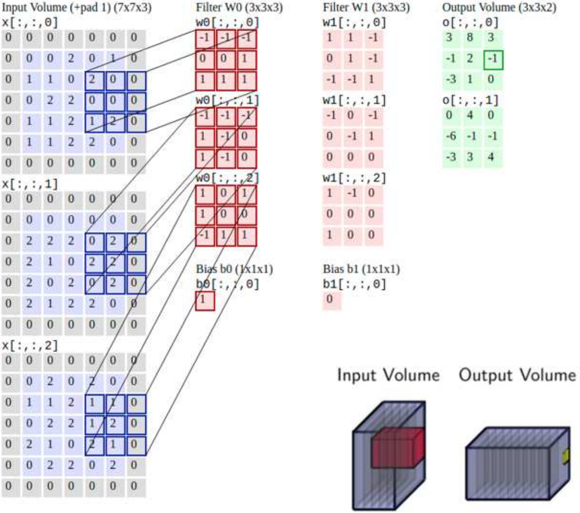


Image source CS231n: Convolutional Neural Networks for Visual Recognition 2020, Fei-Fei Li and Andrej Karpathy and Justin Johnson, `http://cs231n.stanford.edu/`

Convolutional layer

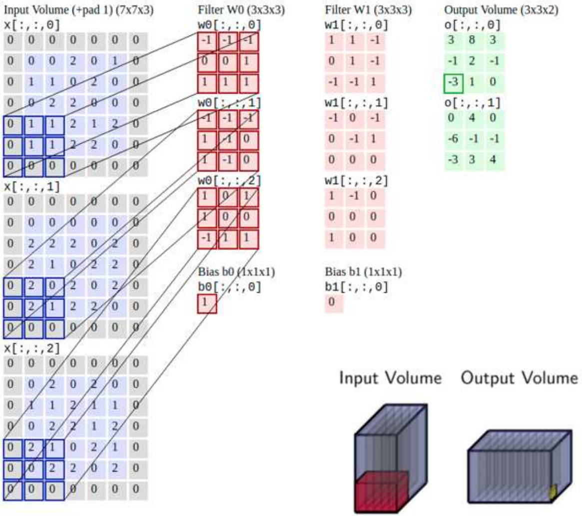
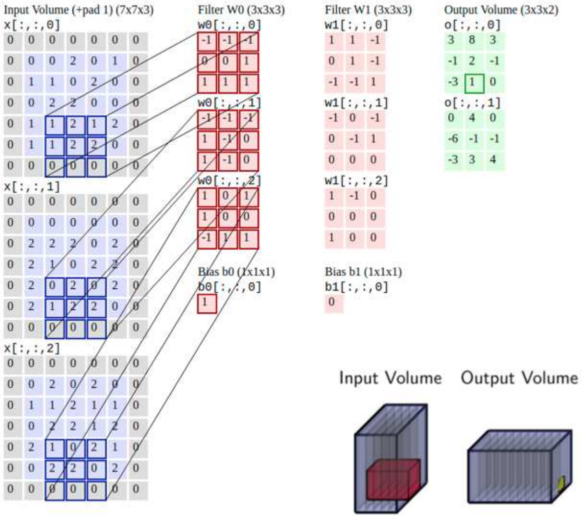


Image source CS231n: Convolutional Neural Networks for Visual Recognition 2020, Fei-Fei Li and Andrej Karpathy and Justin Johnson, http://cs231n.stanford.edu/

Convolutional layer



[Image source](http://cs231n.stanford.edu/) CS231n: Convolutional Neural Networks for Visual Recognition 2020, Fei-Fei Li and Andrej Karpathy and Justin Johnson, `http://cs231n.stanford.edu/`

Convolutional layer

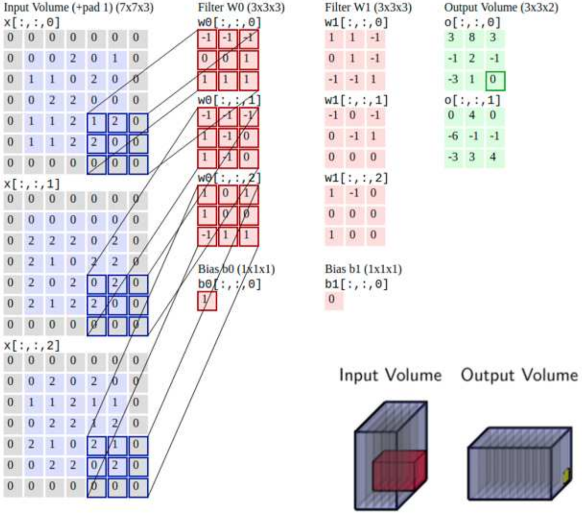


Image source CS231n: Convolutional Neural Networks for Visual Recognition 2020, Fei-Fei Li and Andrej Karpathy and Justin Johnson, http://cs231n.stanford.edu/

Convolutional layer

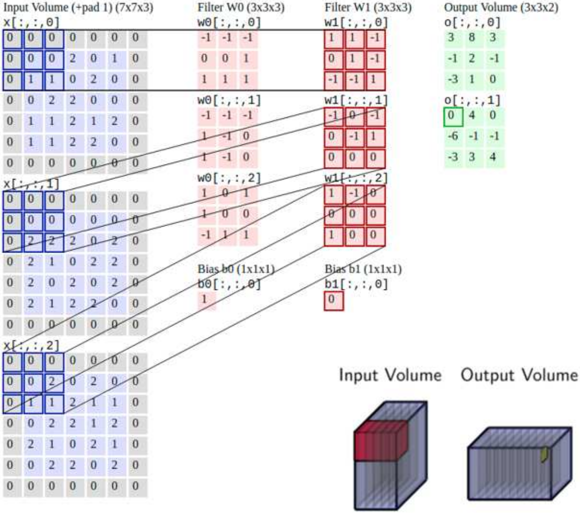


Image source CS231n: Convolutional Neural Networks for Visual Recognition 2020, Fei-Fei Li and Andrej Karpathy and Justin Johnson, `http://cs231n.stanford.edu/`

Convolutional layer

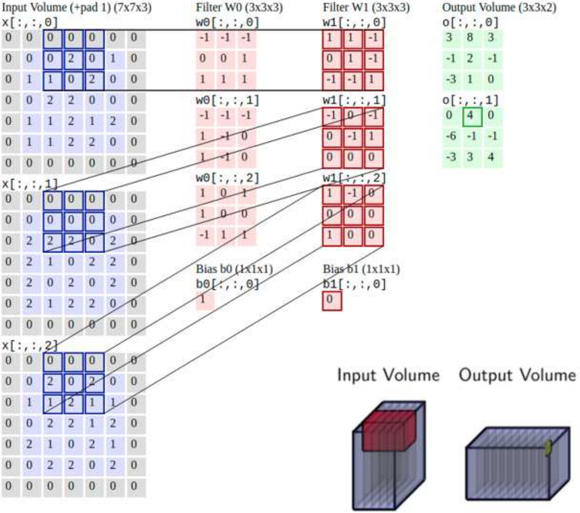


Image source CS231n: Convolutional Neural Networks for Visual Recognition 2020, Fei-Fei Li and Andrej Karpathy and Justin Johnson, http://cs231n.stanford.edu/

Convolutional layer

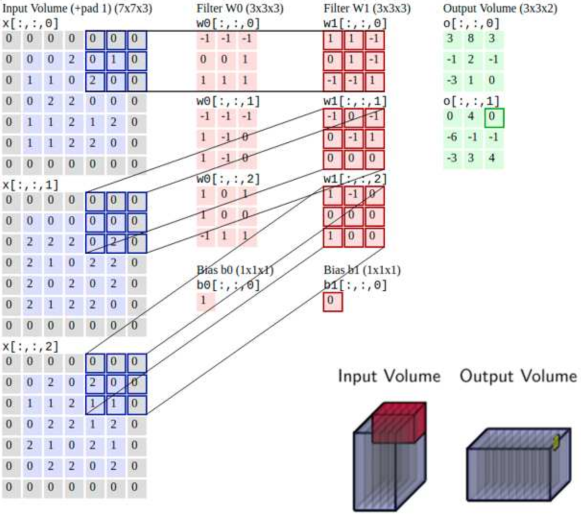


Image source CS231n: Convolutional Neural Networks for Visual Recognition 2020, Fei-Fei Li and Andrej Karpathy and Justin Johnson, http://cs231n.stanford.edu/

Convolutional layer

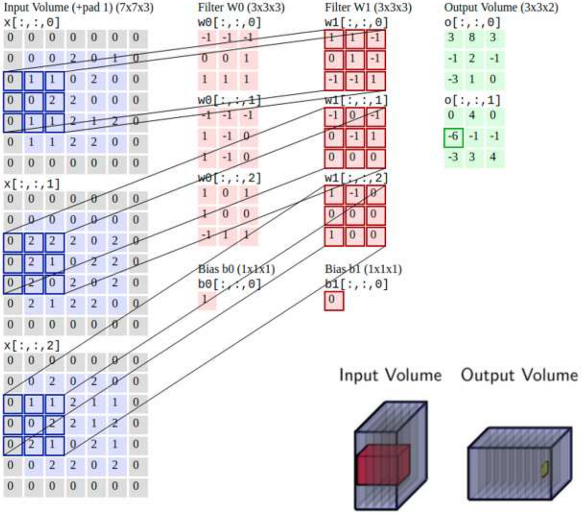


Image source CS231n: Convolutional Neural Networks for Visual Recognition 2020, Fei-Fei Li and Andrej Karpathy and Justin Johnson, http://cs231n.stanford.edu/

Convolutional layer

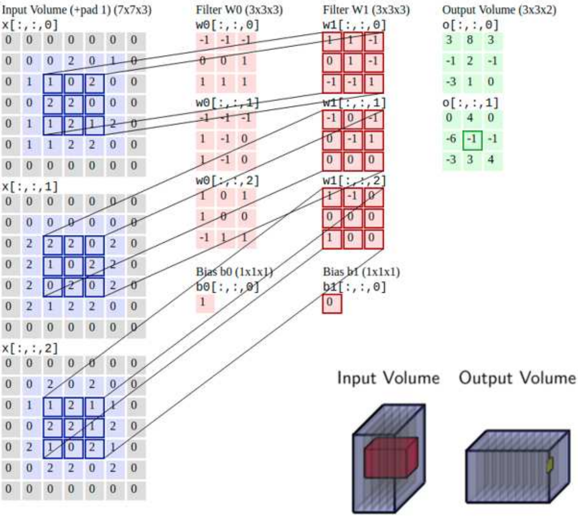


Image source CS231n: Convolutional Neural Networks for Visual Recognition 2020, Fei-Fei Li and Andrej Karpathy and Justin Johnson, http://cs231n.stanford.edu/

Convolutional layer

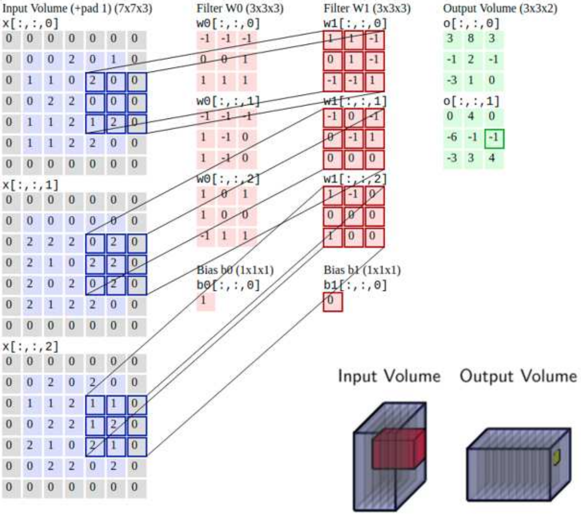


Image source CS231n: Convolutional Neural Networks for Visual Recognition 2020, Fei-Fei Li and Andrej Karpathy and Justin Johnson, http://cs231n.stanford.edu/

Convolutional layer

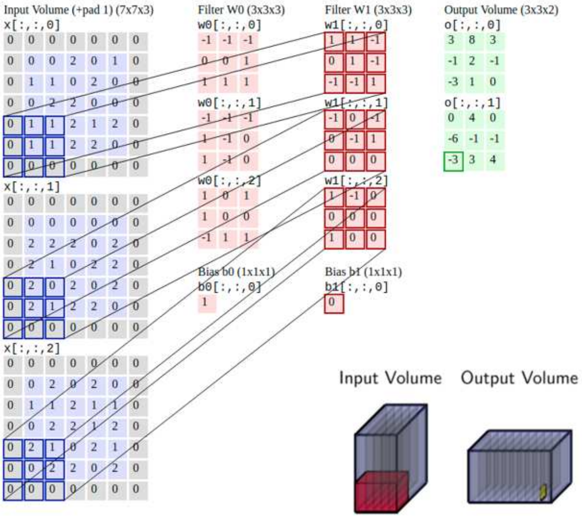


Image source CS231n: Convolutional Neural Networks for Visual Recognition 2020, Fei-Fei Li and Andrej Karpathy and Justin Johnson, http://cs231n.stanford.edu/

Convolutional layer

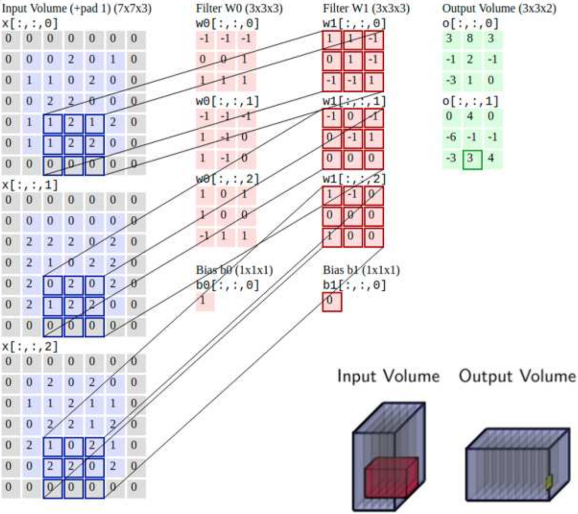


Image source CS231n: Convolutional Neural Networks for Visual Recognition 2020, Fei-Fei Li and Andrej Karpathy and Justin Johnson, http://cs231n.stanford.edu/

Convolutional layer

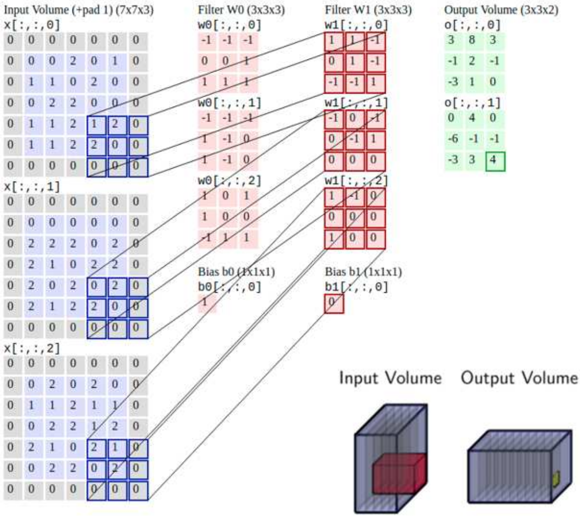
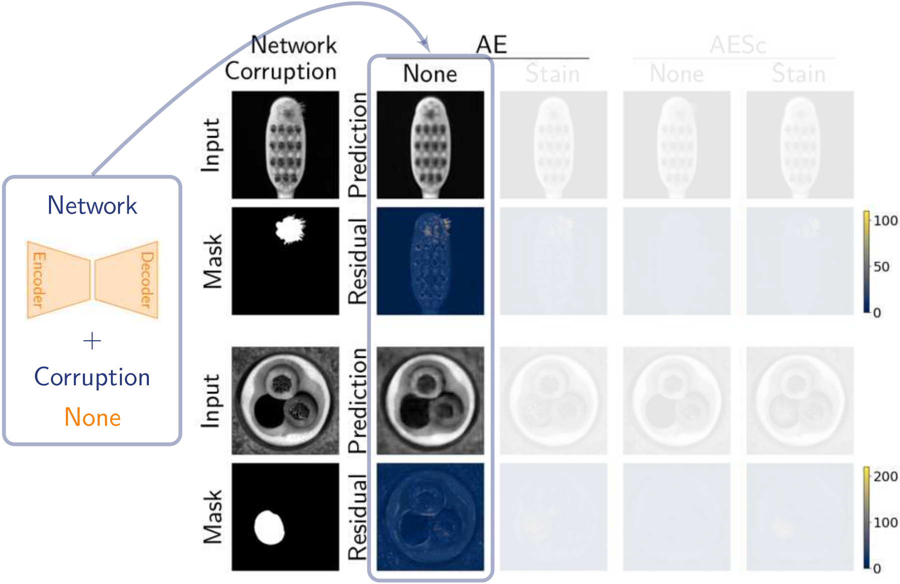
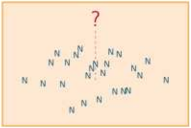


Image source CS231n: Convolutional Neural Networks for Visual Recognition 2020, Fei-Fei Li and Andrej Karpathy and Justin Johnson, http://cs231n.stanford.edu/

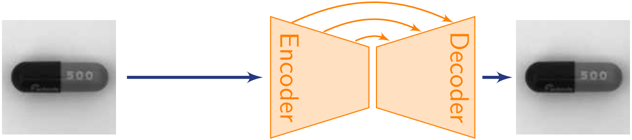

Network
Encoder
Decoder
+
Corruption
None
Network
Corruption
AE
AESc
None
Stain
None
Stain
Input
Mask
Prediction
Residual
Input
Mask
Prediction
Residual
100
50
0
200
100
0

Reconstruct a clean image (2)

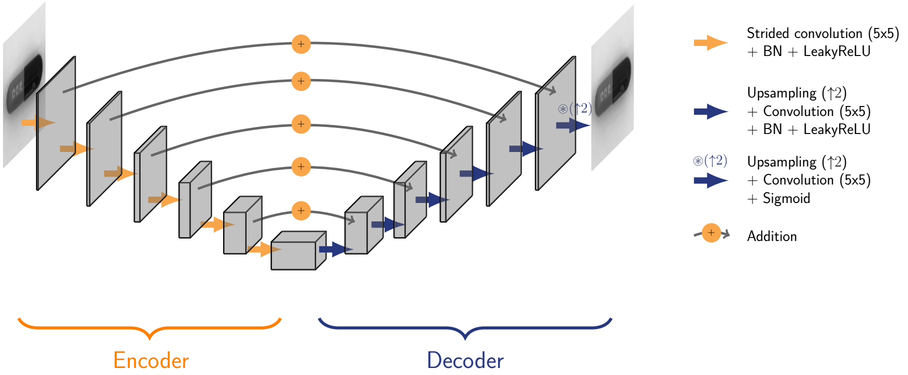


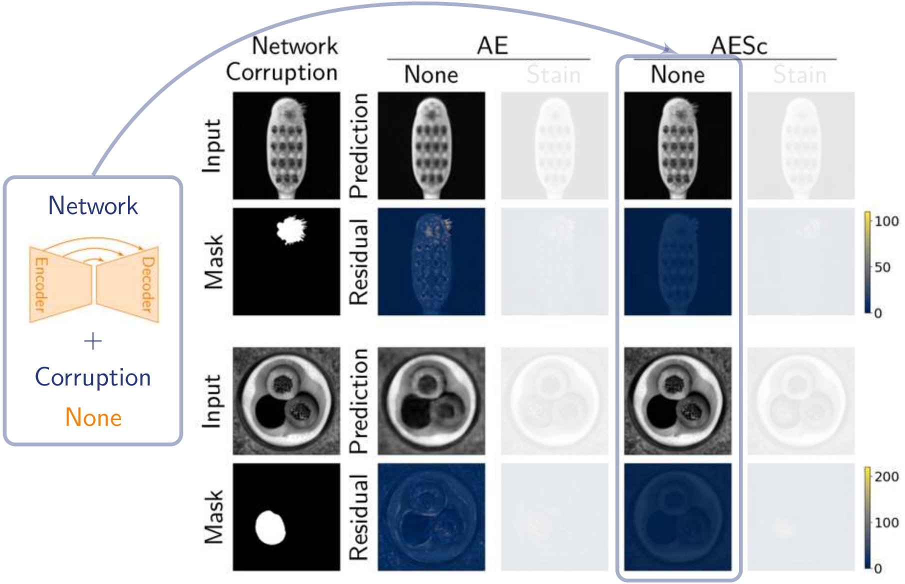
Model 2 :

Train an Autoencoder with Skip connection (AESc) to perform an identity mapping

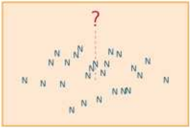


Architecture of the autoencoder with skip connections (AESc)

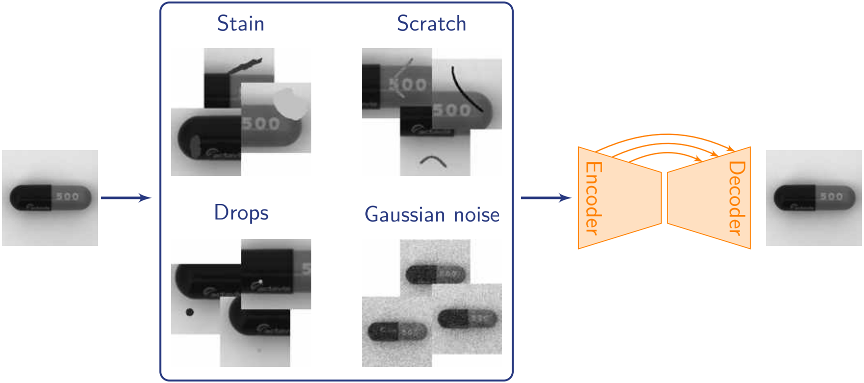



Network
Encoder
Decoder
+
Corruption
None
Network
Corruption
AE
None
Stain
AESc
None
Stain
Input
Mask
Prediction
Residual
Input
Mask
Prediction
Residual
100
50
0
200
100
0

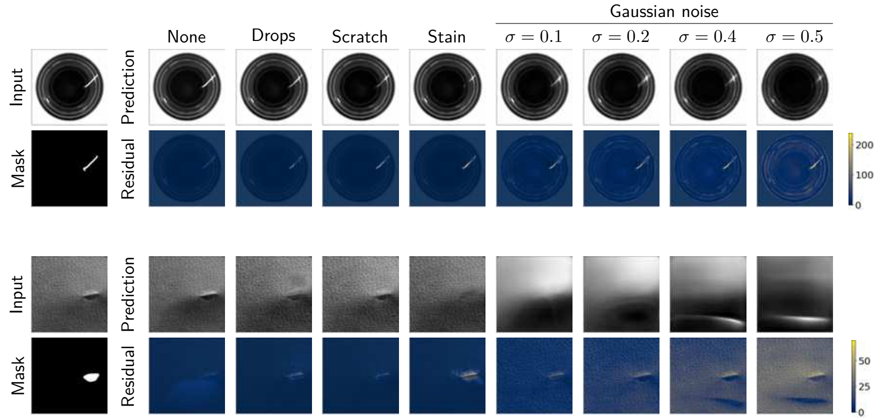
Reconstruct a clean image (3)

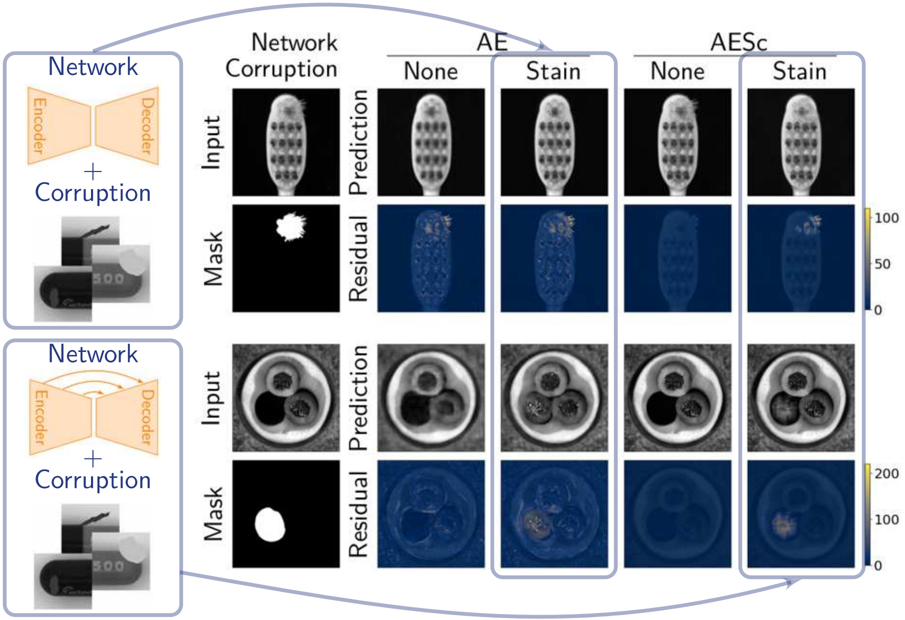


Corrupt training images with synthetic noise to improve the reconstruction



Comparison of the corruption models

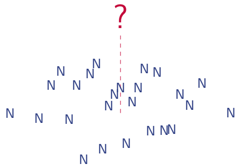



Network
Encoder
Decoder
+
Corruption
Network
Encoder
Decoder
+
Corruption
Network
Corruption
AE
None
Stain
AESc
None
Stain
Input
Mask
Prediction
Residual
100
50
0
Input
Mask
Prediction
Residual
200
100
0

Detect anomalies

Multi-class:
- Classification Based

One-class:
- Reconstruction Based
- Out-of-Distribution Based

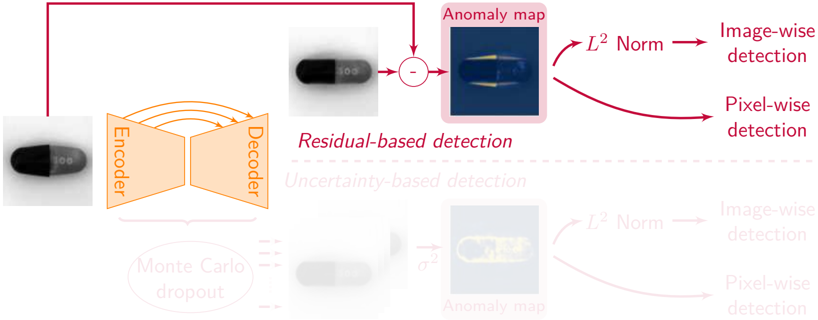


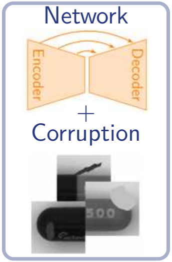
$$\boxed{||? - \text{proj}_{\mathcal{N}}?|| < \tau}$$

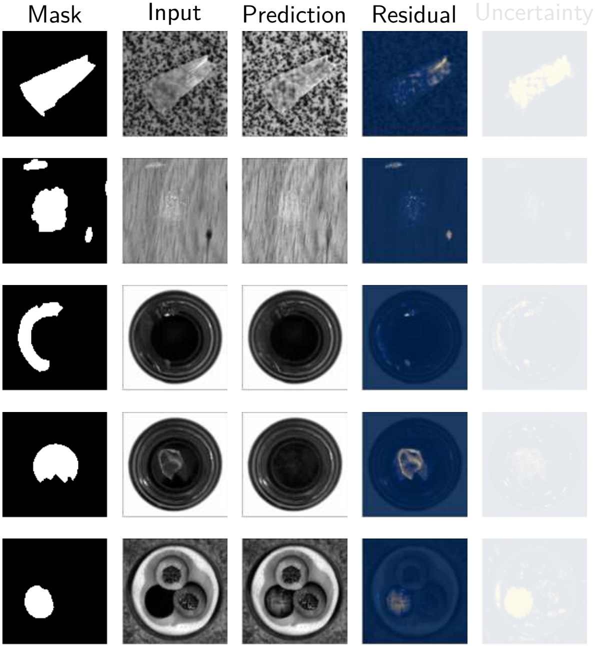
$||? - \text{proj}_{\mathcal{N}}?|| < \tau$

Residual-based approach

Hypohtesis: Residual correlates with defective areas



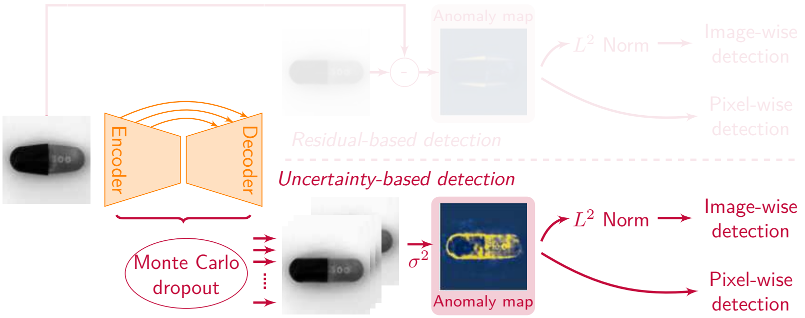

Network
Encoder
Decoder
+
Corruption


Mask
Input
Prediction
Residual
Uncertainty

$||? - \mathrm{proj}_{\mathcal{N}}?|| < \tau$

Uncertainty-based approach

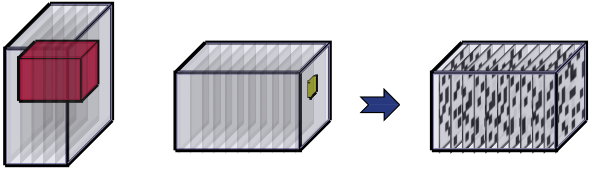
Hypohtesis: Uncertainty correlates with structural deviations from a normal training set



Monte Carlo Dropout

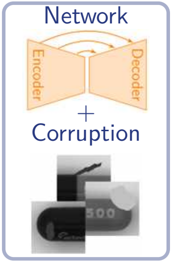
Dropout

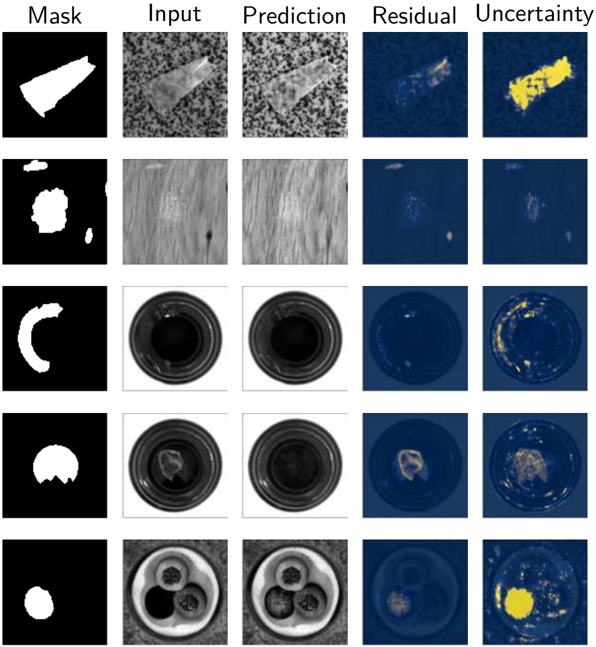
Randomly set values to 0

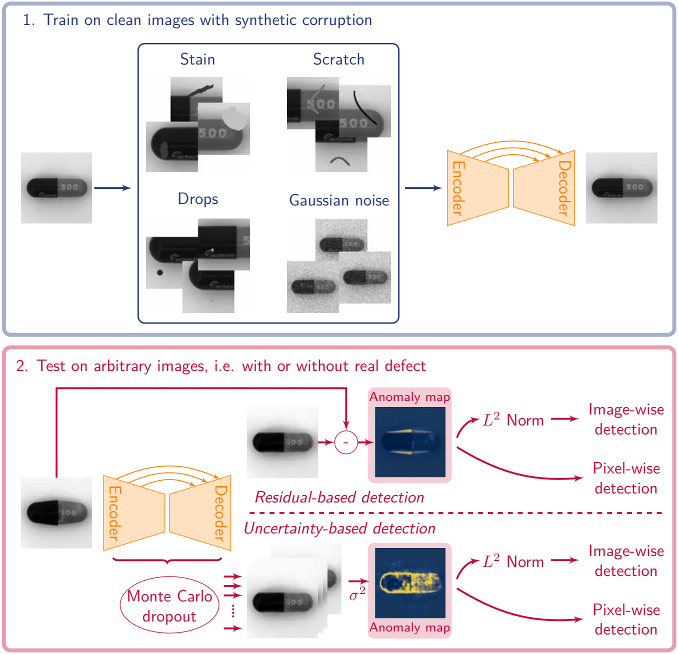


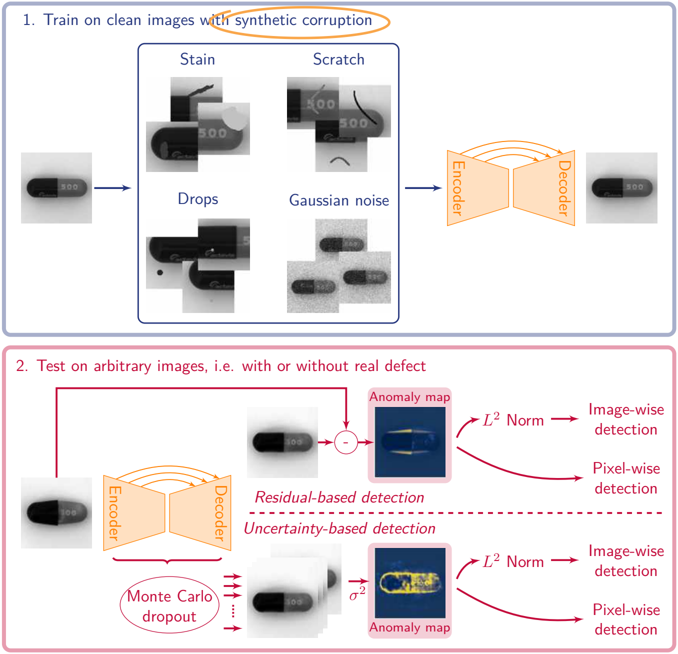
Monte Carlo Dropout

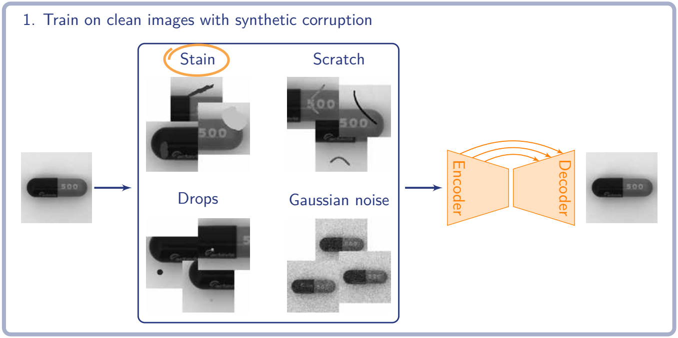
Run multiple forward passes through the model with a different dropout mask every time

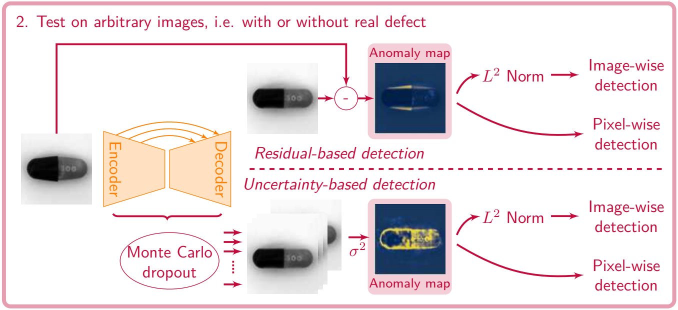

Network
Encoder
Decoder
+
Corruption


Mask
Input
Prediction
Residual
Uncertainty

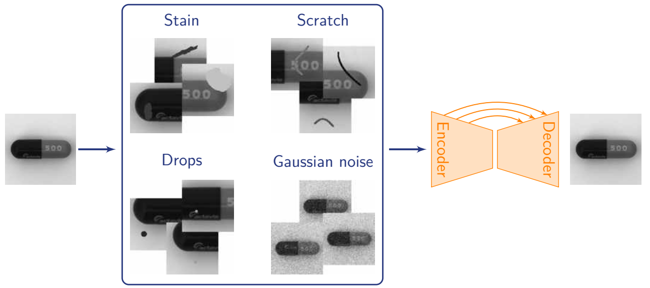

1. Train on clean images with synthetic corruption
Stain
Scratch
Drops
Gaussian noise
Encoder
Decoder
2. Test on arbitrary images, i.e. with or without real defect
Anomaly map
-
L² Norm
Image-wise detection
Pixel-wise detection
Residual-based detection
Uncertainty-based detection
Encoder
Decoder
Monte Carlo dropout
σ²
Anomaly map
L² Norm
Image-wise detection
Pixel-wise detection


1. Train on clean images with synthetic corruption
Stain
Scratch
Drops
Gaussian noise
Encoder
Decoder
2. Test on arbitrary images, i.e. with or without real defect
Anomaly map
L^2 Norm
Image-wise detection
Pixel-wise detection
Residual-based detection
Uncertainty-based detection
Encoder
Decoder
Monte Carlo dropout
σ^2
L^2 Norm
Image-wise detection
Pixel-wise detection
Anomaly map

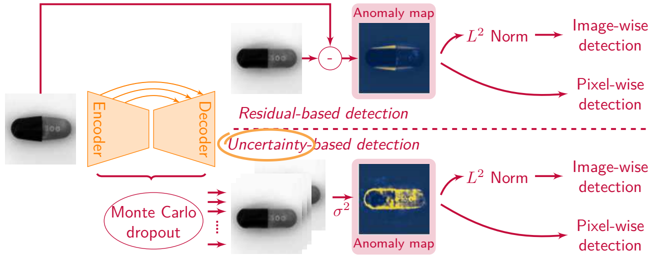

1. Train on clean images with synthetic corruption
Stain
Scratch
Drops
Gaussian noise
Encoder
Decoder


2. Test on arbitrary images, i.e. with or without real defect
Anomaly map
L^2 Norm
Image-wise detection
Pixel-wise detection
Residual-based detection
Uncertainty-based detection
Encoder
Decoder
Monte Carlo dropout
σ^2
L^2 Norm
Image-wise detection
Pixel-wise detection
Anomaly map

1. Train on clean images with synthetic corruption



2. Test on arbitrary images, i.e. with or without real defect



Perspectives

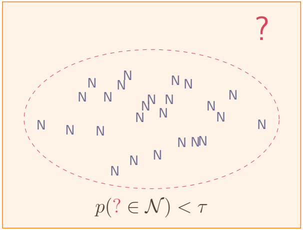
Classification Based

Multi-class

Reconstruction Based

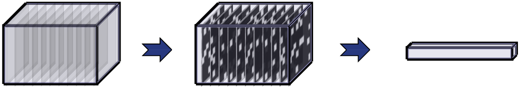
Out-of-Distribution Based

One-class



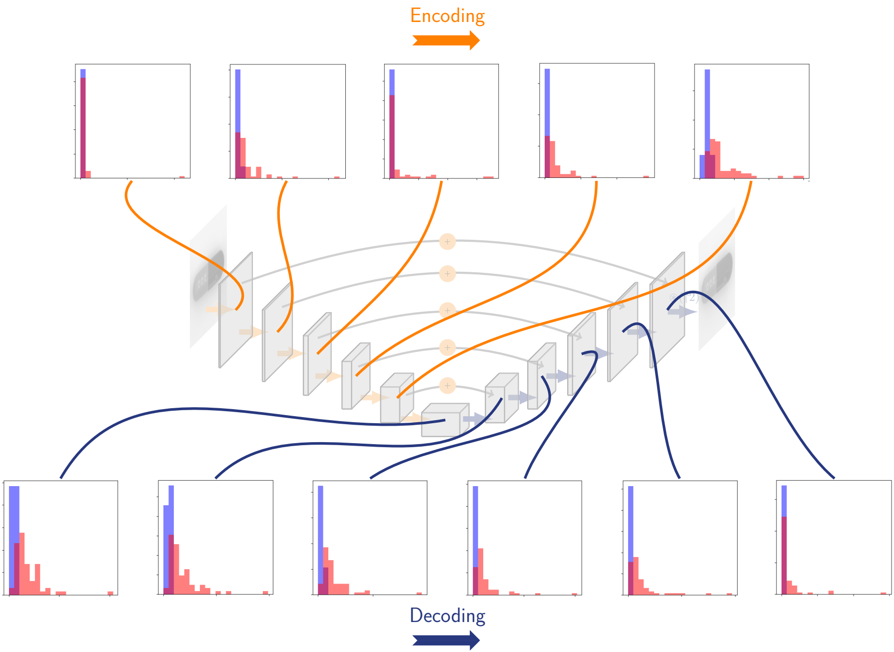
Interesting preliminary experiment...

For each image, construct 1×64 vectors by sampling randomly activation maps
$\rightarrow$ 1 mask/layer



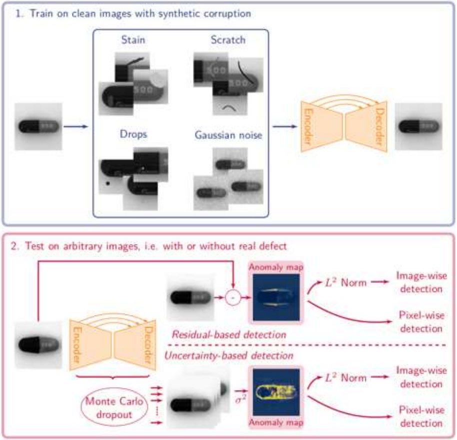
1- Characterize the clean distribution with a Gaussian distribution (training images)

2- Compute the distance between new images and the clean distribution (test images)


Encoding
Decoding

Synthetic corruption of images for anomaly detection using autoencoders

[ISPG seminar]



Any question?