Anomaly Detection with Normalizing Flows

곽재호
Vision and Display System Lab.
Sogang University

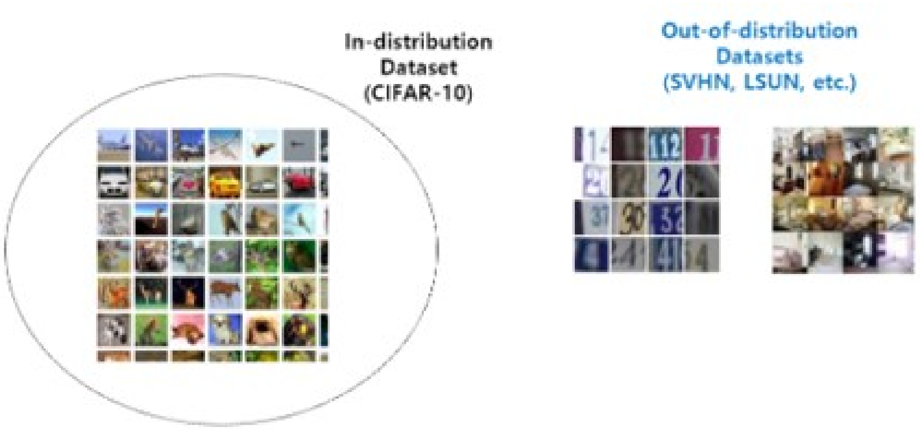
Outline

- Background
  - Anomaly Detection (AD)
    - Anomaly detection datasets
    - AUROC
    - Methodologies
  - [1] Normalizing Flows (Real-NVP)

- Anomaly Detection with Normalizing Flows
  - [2] Same Same But DifferNet: Semi-Supervised Defect Detection with Normalizing Flows (WACV 2021)
  - [3] CFLOW-AD: Real-Time Unsupervised Anomaly Detection with Localization via Conditional Normalizing Flows (WACV 2022)

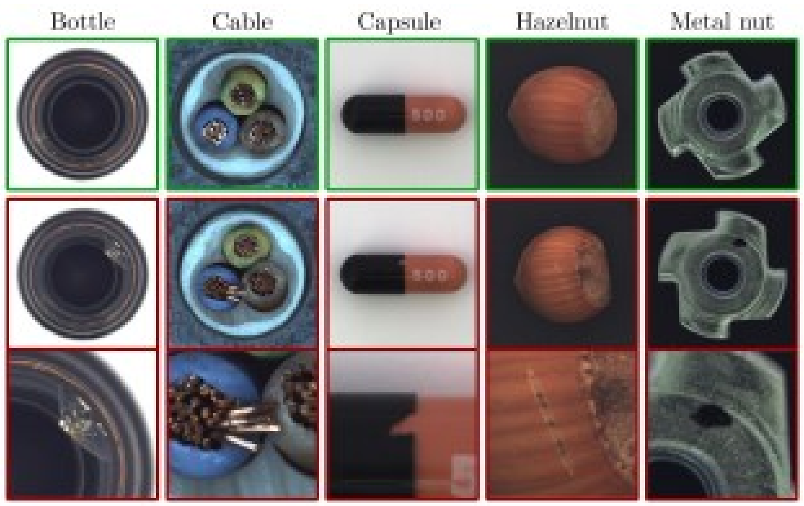
Background

- Anomaly Detection (AD)
  - Anomaly detection datasets
    - CIFAR-10 vs SVHN (or LSUN, CelebA, etc.)
    - [1] ShanghaiTech-AD dataset

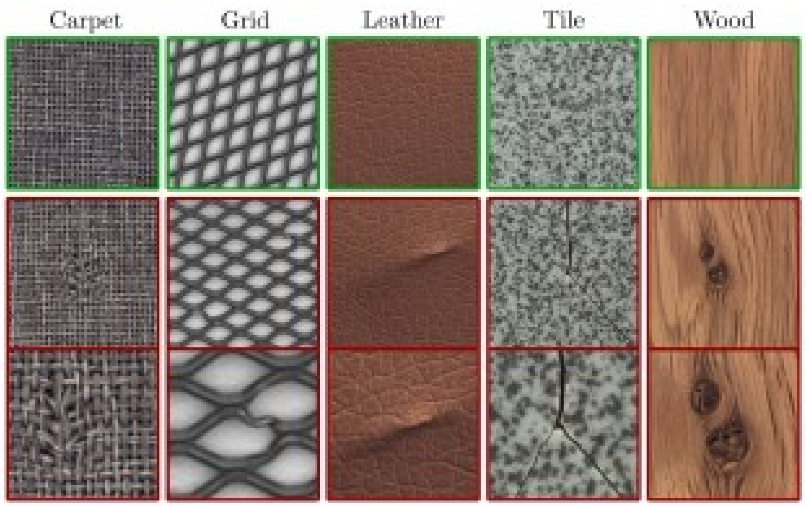


Background

- Anomaly Detection (AD)
  - Anomaly detection datasets
    - [1] MVTec-AD dataset



Object classes

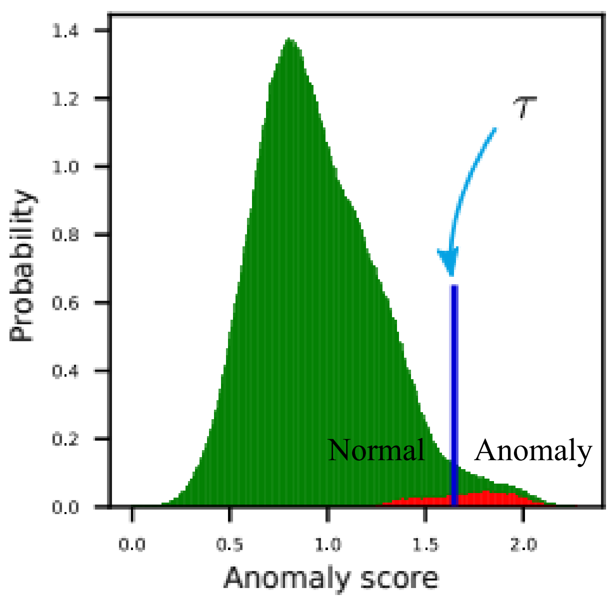


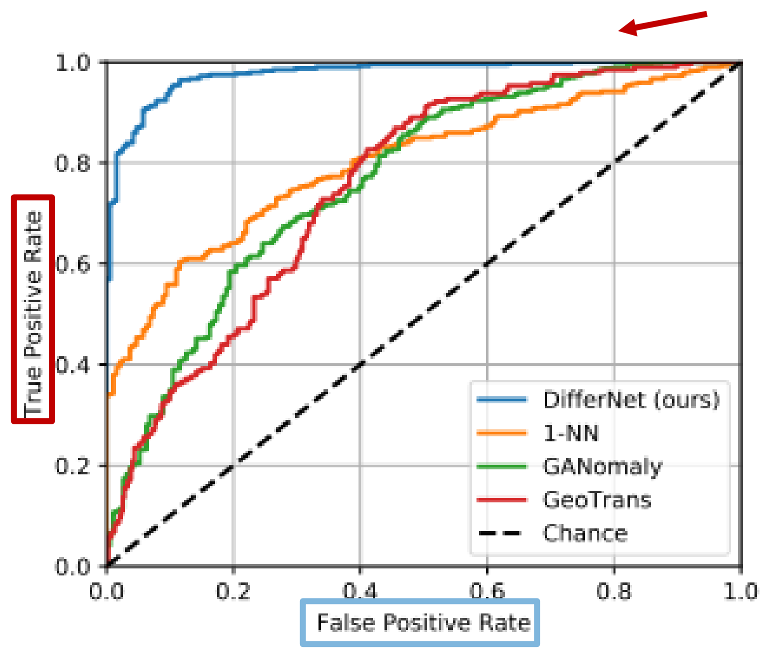
Texture classes

Background

	Positive	Negative
Positive	True Positive	False Positive
Negative	False Negative	True Negative

- Anomaly Detection (AD)
  - AUROC (Area Under Receiver Operating Characteristic curve)
    - As threshold grows along the anomaly score axis, TPR (TP/TP+FN) and FPR (FP/FP+TN) decrease
      - AUROC is the area under the ROC curve
    - The more two classes are separated, higher AUROC

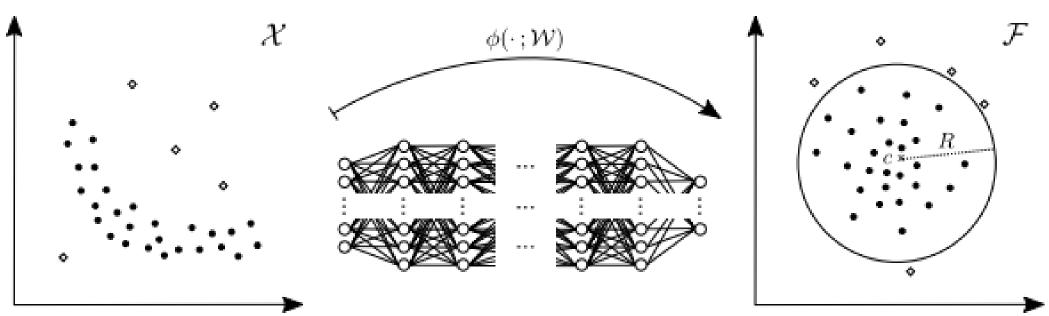




Background

- Anomaly Detection (AD)
  - Methodologies
    - [1] Deep-SVDD (Support Vector Data Description)
      - Jointly learns the network parameters W together with minimizing the volume of a data-enclosing hypersphere in output space F
      - Deep SVDD was trained to extract a data-dependent representation, removing the need to choose an appropriate kernel function by hand
      - At test time, the distance between the representation of the input and the center is used as an anomaly score

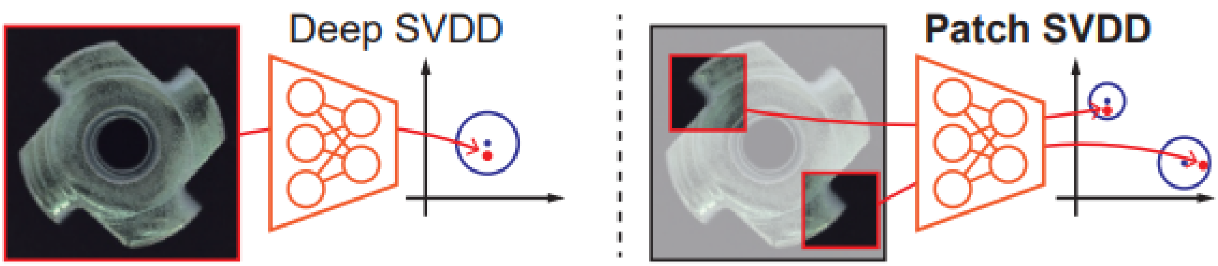
$$\mathcal{L}_{\text{SVDD}} = \sum_i \|f_\theta(\mathbf{x}_i) - \mathbf{c}\|_2$$



Background

- Anomaly Detection (AD)
  - Methodologies
    - [1] Patch-SVDD (Support Vector Data Description)
      - Patch-SVDD extends Deep-SVDD to a patch-wise detection method
        - ✓ Patch-SVDD performs inspection on every patch to localize a defect
      - Mapping all the features of dissimilar patches to a single center and imposing a uni-modal cluster weaken the connection between the representation and the content
      - To deal with this, the encoder was trained to gather semantically similar patches by itself

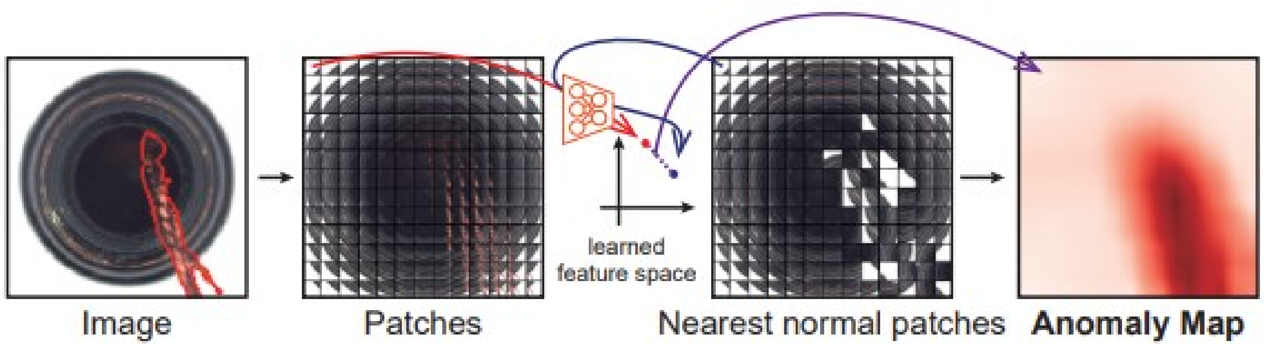
$$\mathcal{L}_{\text{SVDD'}} = \sum_{i,i'} \| f_\theta(\mathbf{p}_i) - f_\theta(\mathbf{p}_{i'}) \|_2$$



Background

- Anomaly Detection (AD)
  - Methodologies
    - [1] Patch-SVDD (Support Vector Data Description)
      - Patch-SVDD stores all the representation of the galleries' (training dataset)
      - At test time for every patch p with a stride S within test image x, **the L2 distance to the nearest normal patch** in the feature space is then defined to be its anomaly score

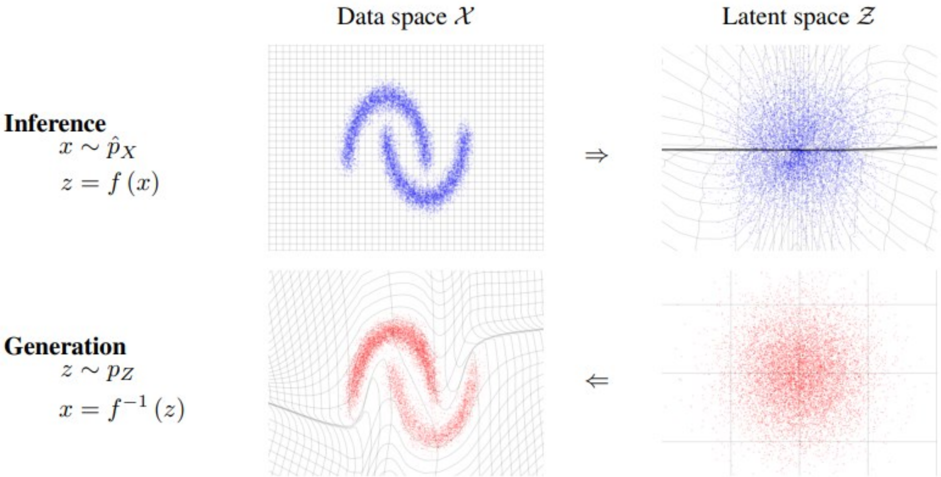
$$\mathcal{A}_\theta^{\text{patch}}(\mathbf{p}) \doteq \min_{\mathbf{p}_{\text{normal}}} \| f_\theta(\mathbf{p}) - f_\theta(\mathbf{p}_{\text{normal}}) \|_2$$



Background

- [1] Normalizing Flows (Real-NVP)
  - Basic Idea
    - [2] *"a good representation is one in which the distribution of the data is easy to model"*
    - Find transformation h = f(x) of the data into a new space such that the resulting distribution factorizes
      - Each distribution H_d should be **independent**, and could be any **parametric** distribution (ex. Gaussian, Poisson)
      - The transformation f is **easily invertible**, and the dimension of h is same as the dimension of x

$$p_H(h) = \prod_d p_{H_d}(h_d).$$



Background

- [1] Normalizing Flows (Real-NVP)
  - Change of variable formula
    - The likelihood of a datapoint at the feature space could be written with the following **change of variables formula**

$$p_X(x) = p_H(f(x))\left|\det \frac{\partial f(x)}{\partial x}\right|. \qquad \begin{aligned} h &\sim p_H(h) \\ x &= f^{-1}(h) \end{aligned}$$

Given an arbitrary distribution $p_X(x)$, likelihood at a datapoint x' is $p_X(x')$

$$\log p_X(x) = \log p_H(f(x)) + \boxed{log \left|\det \frac{\partial f(x)}{\partial x}\right|}$$

    - If a random variable H follows a Standard Gaussian distribution, $h \sim \mathcal{N}(0, 1)$

$$p_H(h) = (2\pi)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}h^2\right)$$

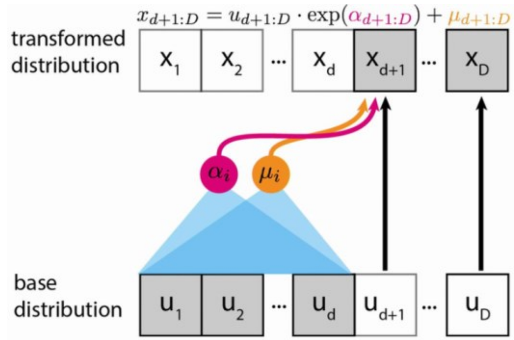
    - Log-likelihood of $p_H(h)$ is just a L2 distance of the datapoint on the latent space
    - Now, all that matter is the **log determinant of the Jacobian (red box)**

Background

- [1] Normalizing Flows (Real-NVP)
  - Computational complexity of determinant of the Jacobian
    - However, the complexity of computing the **determinant of the Jacobian** is a major drawback
  - Determinant of the triangular matrix
    - The determinant of an upper (or lower) triangular matrix is the **product of the main diagonal entries**

$$J_f = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$$

$$det\ \ U = \begin{vmatrix} d_1 & * & * & * \\ 0 & d_2 & * & * \\ 0 & 0 & \ddots & * \\ 0 & 0 & 0 & d_n \end{vmatrix} = d_1 \times d_2 \times \cdots \times d_n$$



Background

- [1] Normalizing Flows (Real-NVP)
  - Affine coupling layer
    - For the sake of **easy determinant of the Jacobian**, a special transform was proposed, coupling layer

$$y_{1:d} = x_{1:d}$$
$$y_{d+1:D} = x_{d+1:D} \odot \exp\left(s(x_{1:d})\right) + t(x_{1:d})$$

      - s and t stand for scale and translation, and are functions from $R^d \rightarrow R^{D-d}$ (MLP)
    - With this type of transformation, the determinant of the Jacobian will be a lower triangular matrix

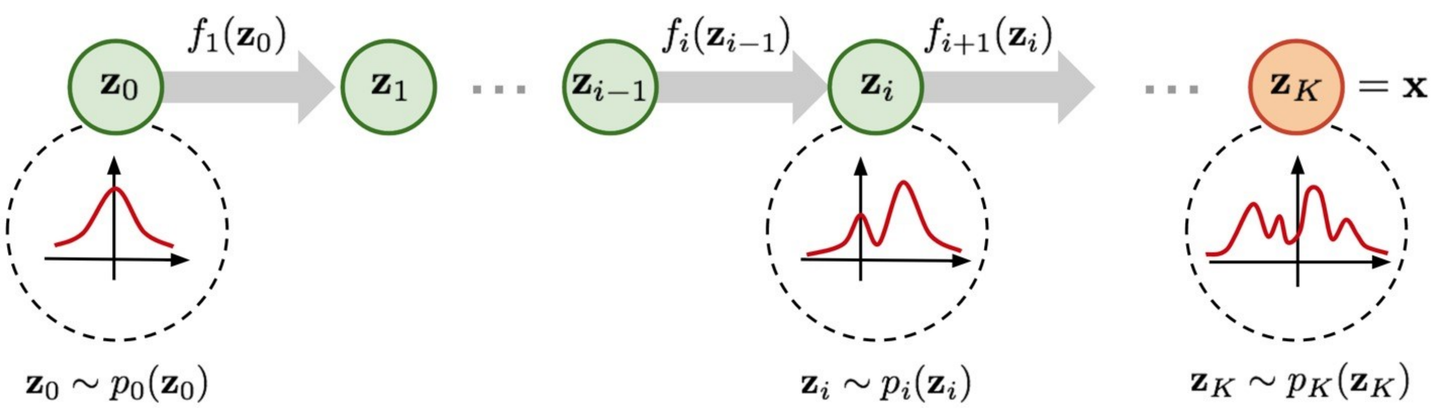
$$\frac{\partial y}{\partial x^T} = \begin{bmatrix} \mathbb{I}_d & 0 \\ \frac{\partial y_{d+1:D}}{\partial x^T_{1:d}} & \operatorname{diag}\left(\exp\left[s\left(x_{1:d}\right)\right]\right) \end{bmatrix}$$

    - And eventually, log-determinant of the Jacobian is $\sum_j s(x_{1:d})$

$$\log p_X(x) = \log p_H(f(x)) + \boxed{log\left|\det\frac{\partial f(x)}{\partial x}\right|}$$

Background

- [1] Normalizing Flows (Real-NVP)
  - Combining coupling layer
    - Although coupling layers can be powerful, their forward transformation leaves some components unchanged
    - By sequentially attaching the coupling layers, overall network's representation power grows



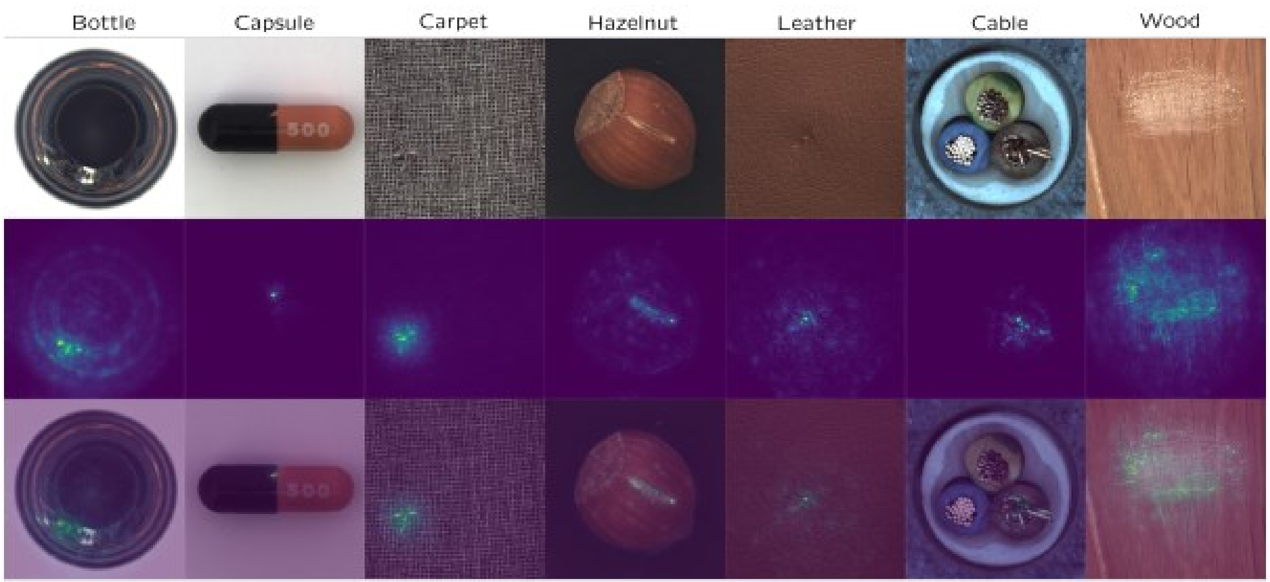
Anomaly detection with Normalizing Flows

- [1] Same Same But DifferNet: Semi-Supervised Defect Detection with Normalizing Flows (WACV 2021)
  - Detection of anomalies via the usage of likelihoods provided by a **normalizing flow** on multi-scale image features with multi-transform evaluation
  - **Anomaly localization** without training labels, the necessity of any pixel-wise optimization and sub-image detection

- [2] CFLOW-AD: Real-Time Unsupervised Anomaly Detection with Localization via Conditional Normalizing Flows (WACV 2022)
  - Proposed to use **conditional normalizing flows for unsupervised anomaly detection with localization** using computational and memory-efficient architecture

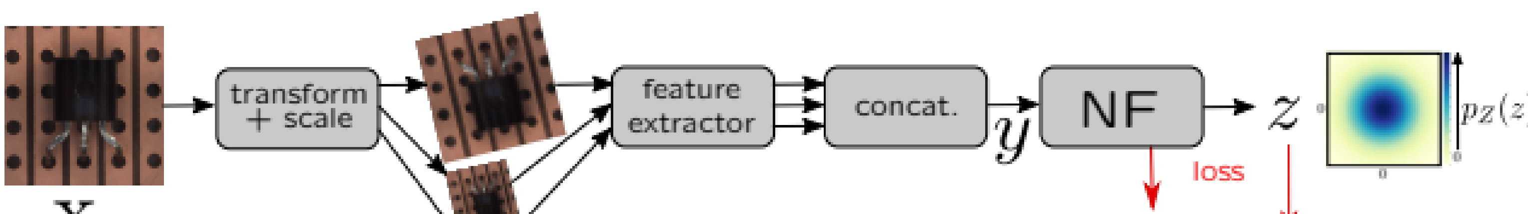
Anomaly detection with Normalizing Flows

- [1] Same Same But DifferNet: Semi-Supervised Defect Detection with Normalizing Flows (WACV 2021)
  - Detection of anomalies via the usage of likelihoods provided by a **normalizing flow** on multi-scale image features with multi-transform evaluation
  - **Anomaly localization** without training labels, the necessity of any pixel-wise optimization and sub-image detection



Anomaly detection with Normalizing Flows

- [1] Same Same But DifferNet: Semi-Supervised Defect Detection with Normalizing Flows (WACV 2021)
  - DifferNet is a density estimation-based network using [2] Real-NVP
    - **Density estimation of image features y** $\in$ Y from the anomaly-free training images x $\in$ X
  - DifferNet uses an ImageNet pretrained feature extractor which is not further optimized
    - There have been [3] studies showing that it cannot be effectively applied to data of high dimensionality
    - As there are many defects with variable scales in MVTec-AD dataset, DifferNet uses a multi-scale feature extractor
    - DifferNet also used image transforms such as rotation.



$$\mathcal{L}(y) = -\log\left|\det\frac{\partial z}{\partial y}\right| - \log p_Z(z)$$

Anomaly detection with Normalizing Flows

- [1] Same Same But DifferNet: Semi-Supervised Defect Detection with Normalizing Flows (WACV 2021)
  - What is learned from the network is a **transformation** from the feature space to the latent space
    - Training is done in a way that maximizes the likelihood for extracted features y
    - According to the change-of-variable formula, this problem could be described as maximizing

Simple MLE!

$$p_Y(y) = p_Z(z)\left|\det \frac{\partial z}{\partial y}\right|.$$

Log-likelihood

$$\log p_Y(y) = \log p_Z(z) + \log\left|\det \frac{\partial z}{\partial y}\right|$$

Negative log-likelihood & MVG modeling $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I})$

$$\mathcal{L}(y) = \boxed{\frac{\|z\|_2^2}{2}} \boxed{- \log\left|\det \frac{\partial z}{\partial y}\right|}.$$

    - The first term (in the red box) of the loss function forces NF to map all y as close as possible to z=0
    - The latter term (in the blue box) penalizes trivial solutions (all points are mapped into z=0)
  - Thanks to the **tractability** of the determinant of the Jacobian, it is easy to calculate the exact **log-likelihoods in the feature space** ($log p_y(y)$)

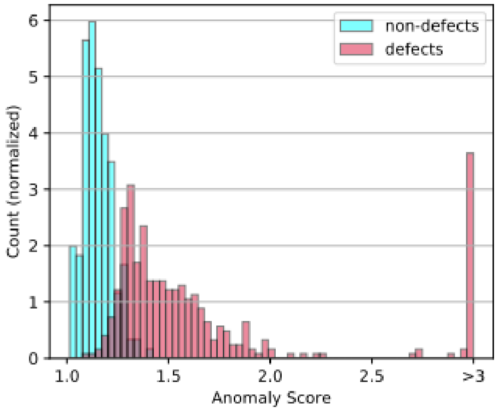
Anomaly detection with Normalizing Flows

- [1] Same Same But DifferNet: Semi-Supervised Defect Detection with Normalizing Flows (WACV 2021)
  - Likelihoods are directly used as a criterion to classify a sample as anomalous or normal

$$\tau(x) = \mathbb{E}_{T_i \in \mathcal{T}} \left[-\log p_Z(f_{\mathrm{NF}}(f_{\mathrm{ex}}(T_i(x)))) \right].$$

    - At test time, multiple transformations are used to get a robust anomaly score $\tau(x)$
      - Rotations or manipulations of brightness and contrast
    - An image is classified as anomalous if the anomaly score $\tau(x)$ is above the threshold value θ where $\mathcal{A}(x) = 1$ indicates an anomaly
      - θ is varied to calculate the AUROC

$$\mathcal{A}(x) = \begin{cases} 1 & \text{for } \tau(x) \geq \theta \\ 0 & \text{for } \tau(x) < \theta \end{cases},$$



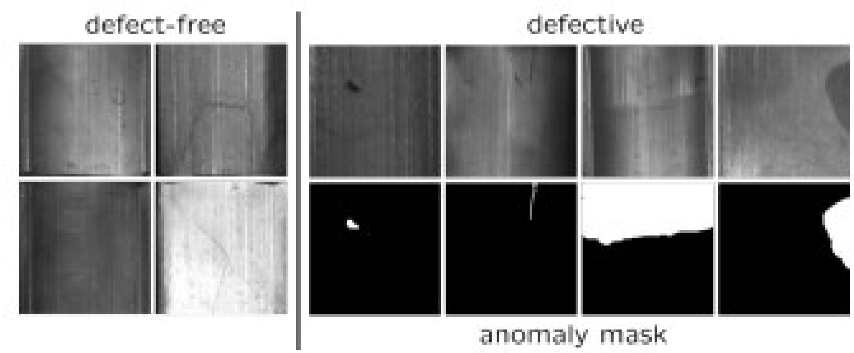
Anomaly detection with Normalizing Flows

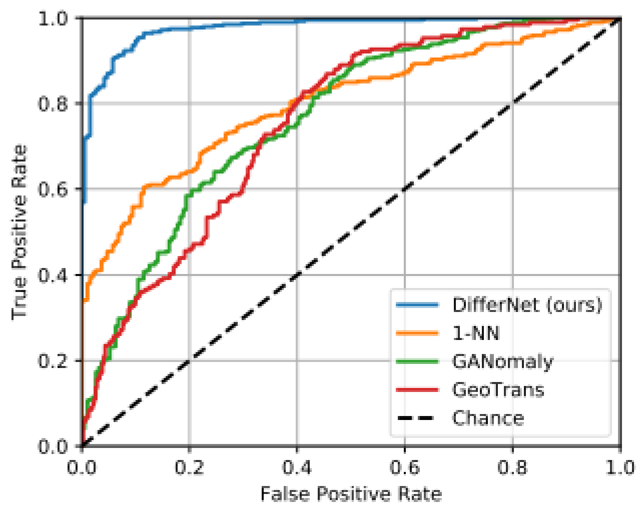
- [1] Same Same But DifferNet: Semi-Supervised Defect Detection with Normalizing Flows (WACV 2021)
  - Image-level AUROC in % for detected anomalies of all categories of MVTec AD

	Category	GeoTrans [10]	GANomaly [1]	DSEBM [30]	OCSVM [2]	1-NN [21]	**DifferNet (ours)**
Textures	Grid	61.9	70.8	71.7	41.0	55.7	**84.0**
	Leather	84.1	84.2	41.6	88.0	90.3	**97.1**
	Tile	41.7	79.4	69.0	87.6	96.9	**99.4**
	Carpet	43.7	69.9	41.3	62.7	81.1	**92.9**
	Wood	61.1	83.4	95.2	95.3	93.4	**99.8**
Objects	Bottle	74.4	89.2	81.8	**99.0**	98.7	**99,0**
	Capsule	67.0	73.2	59.4	54.4	71.1	**86.9**
	Pill	63.0	74.3	80.6	72.9	83.7	**88.8**
	Transistor	86.9	79.2	74.1	56.7	75.6	**91.1**
	Zipper	82.0	74.5	58.4	51.7	88.6	**95.1**
	Cable	78.3	75.7	68.5	80.3	88.5	**95.9**
	Hazelnut	35.9	78.5	76.2	91.1	97.9	**99.3**
	Metal Nut	81.3	70.0	67.9	61.1	76.7	**96.1**
	Screw	50.0	74.6	**99.9**	74.7	67.0	96.3
	Toothbrush	97.2	65.3	78.1	61.9	91.9	**98.6**
	Average	67.2	76.2	70.9	71.9	83.9	**94.9**

Anomaly detection with Normalizing Flows

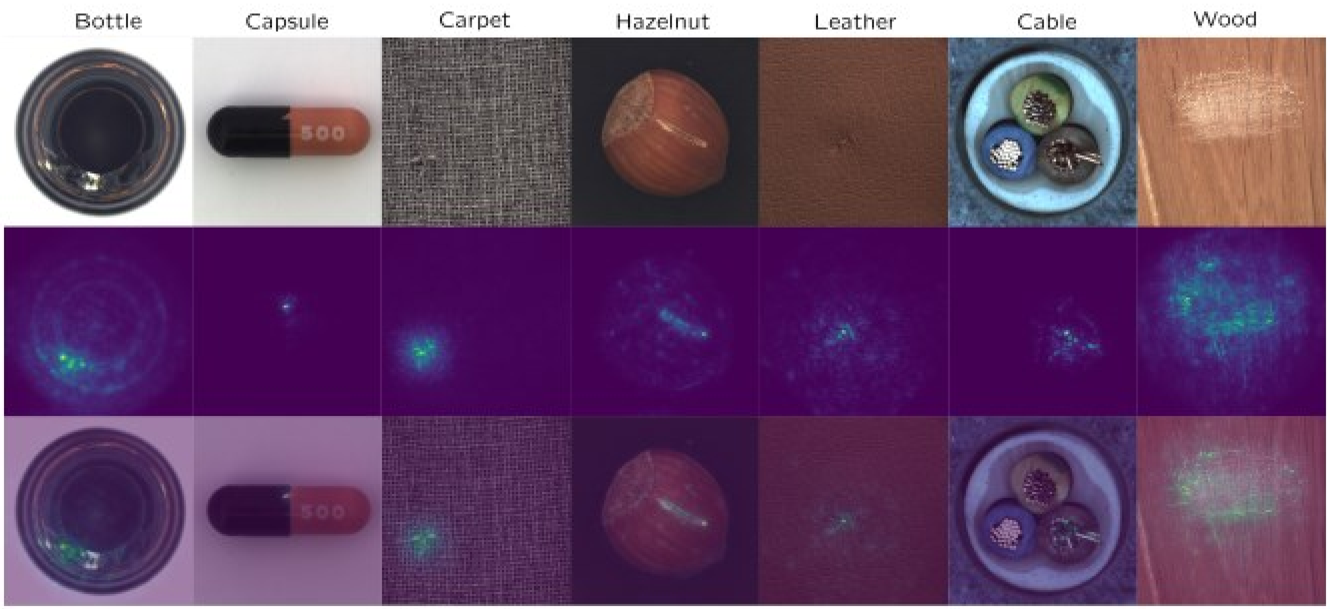
- [1] Same Same But DifferNet: Semi-Supervised Defect Detection with Normalizing Flows (WACV 2021)
  - Anomaly detection results on [2] Magnetic Tile Defects dataset





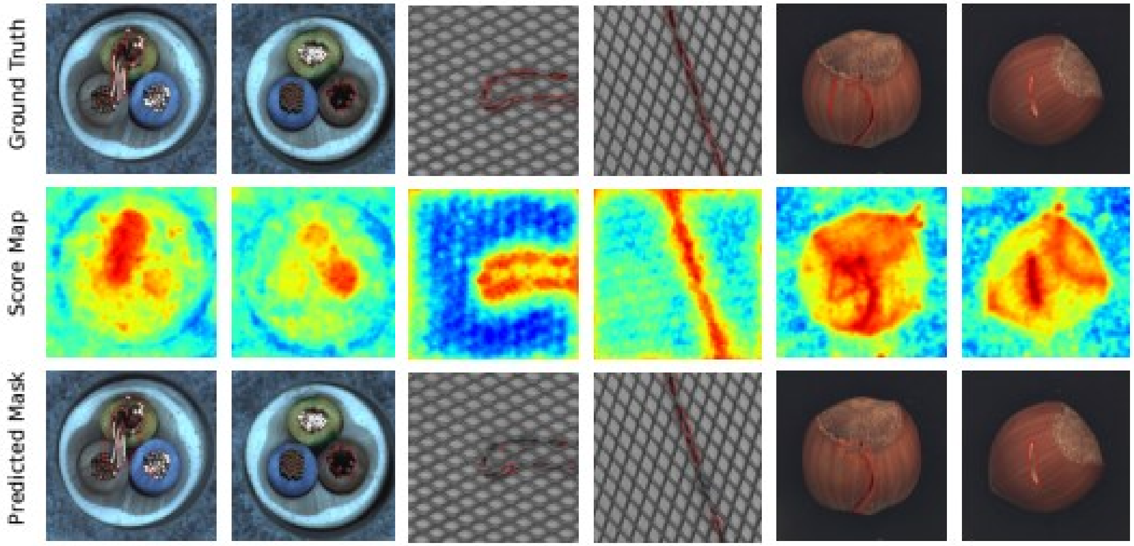
Anomaly detection with Normalizing Flows

- [1] Same Same But DifferNet: Semi-Supervised Defect Detection with Normalizing Flows (WACV 2021)
  - Anomaly localization could be done with propagating the negative log-likelihood back to the input image x



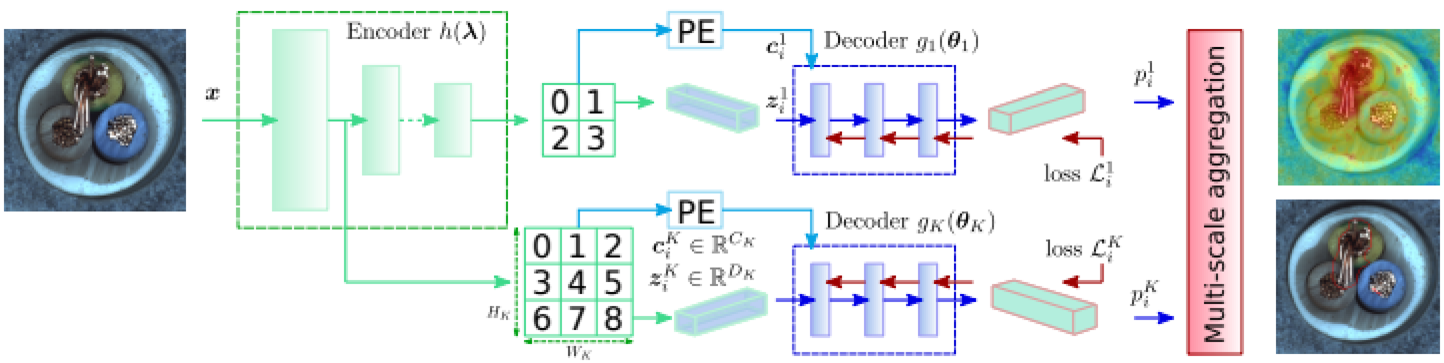
Anomaly detection with Normalizing Flows

- [1] CFLOW-AD: Real-Time Unsupervised Anomaly Detection with Localization via Conditional Normalizing Flows (WACV 2022)
  - Proposed to use **conditional normalizing flows for unsupervised anomaly detection with localization** using computational and memory-efficient architecture



Anomaly detection with Normalizing Flows

- [1] CFLOW-AD: Real-Time Unsupervised Anomaly Detection with Localization via Conditional Normalizing Flows (WACV 2022)
  - CFlow also uses an ImageNet pretrained feature extractor which is not further optimized
    - One difference with DifferNet is that CFlow uses multi-scale features from the encoder's pyramid pooling layers
  - Cflow performs anomaly localization with a variation of the [2] Real-NVP, namely, conditional flow
    - Conditional vector c_i is a 2D form of conventional positional encoding (sinusoidal)
      - Conditional vector gives spatial information of each feature vector to the CFlow decoder



Anomaly detection with Normalizing Flows

- [1] CFLOW-AD: Real-Time Unsupervised Anomaly Detection with Localization via Conditional Normalizing Flows (WACV 2022)
  - Training procedure is same as DifferNet
    - Maximizing the likelihoods of the features in the feature space

$$\mathcal{L}(\theta) = \mathbb{E}_{\hat{p}_Z(\boldsymbol{z},\boldsymbol{\theta})}[D_{KL}[p_Z(\boldsymbol{z})\|\hat{p}_Z(\boldsymbol{z},\boldsymbol{\theta})\ ]]$$

Injecting conditional vector

$$\mathcal{L}(\theta) = \mathbb{E}_{\hat{p}_Z(\boldsymbol{z},\boldsymbol{c},\boldsymbol{\theta})}[D_{KL}[p_Z(\boldsymbol{z})\|\hat{p}_Z(\boldsymbol{z},\boldsymbol{c},\boldsymbol{\theta})]]$$

$$\approx \frac{1}{N}\sum_{i=1}^{N}\left[\frac{\|\boldsymbol{u}_i\|_2^2}{2} - \log|\det \boldsymbol{J}_i|\right] + \text{const}$$

MCMC approximation

- i is the index of each training dataset

    - All K scale CFlow decoders are trained

Anomaly detection with Normalizing Flows

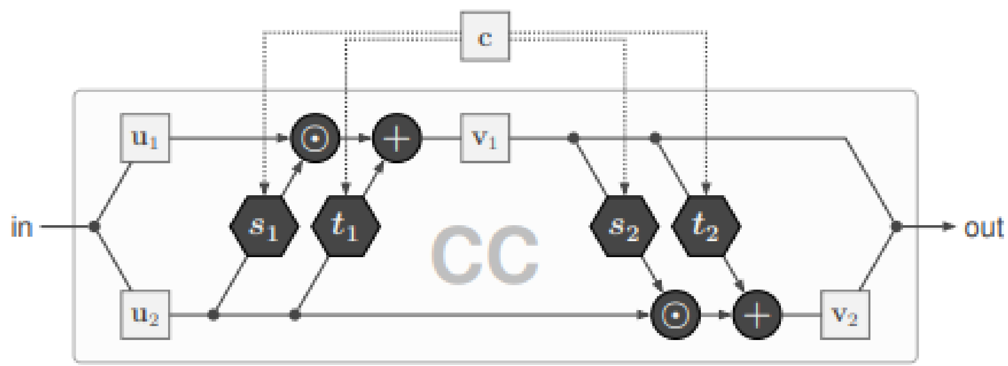
- [1] CFLOW-AD: Real-Time Unsupervised Anomaly Detection with Localization via Conditional Normalizing Flows (WACV 2022)
  - Anomaly score for each pixel is also directly computed with the likelihoods at the latent space

$$\log \hat{p}_Z(\boldsymbol{z}_i, \boldsymbol{c}_i, \hat{\boldsymbol{\theta}}) = -\frac{\|\boldsymbol{u}_i\|_2^2 + D\log(2\pi)}{2} + \log|\det \boldsymbol{J}_i|$$

    - Here, $\hat{p}_Z(\boldsymbol{z}, \boldsymbol{c}, \widehat{\boldsymbol{\theta}})$ is the likelihoods of transformed feature vectors ($\mathcal{F} \rightarrow \mathcal{Z}$)
      - $\hat{p}_Z(\boldsymbol{z}, \boldsymbol{c}, \widehat{\boldsymbol{\theta}}) = p_Z(\boldsymbol{z})|\det \boldsymbol{J}_i|$
    - After calculating all the log-likelihoods for each pixel, normalize them to be in [0 : 1] range
      - Constructing anomaly maps for each scale is finished
  - Up-sample all the anomaly maps into the input image resolution using bilinear interpolation
  - The final anomaly score map is made by aggregating all the anomaly score maps

Anomaly detection with Normalizing Flows

- [1] CFLOW-AD: Real-Time Unsupervised Anomaly Detection with Localization via Conditional Normalizing Flows (WACV 2022)
  - CFlow learns every patch's (effective receptive field in the input image) distribution
    - Which is like the Patch-SVDD
    - However, Patch-SVDD stores every trained representation, this makes difference in speed, memory usage
  - CFlow followed [2] a study that concatenates the intermediate vectors inside decoder coupling layers with the conditional vectors
    - Conditional vector gives spatial information of each feature vector to the CFlow decoder



Anomaly detection with Normalizing Flows

- [1] CFLOW-AD: Real-Time Unsupervised Anomaly Detection with Localization via Conditional Normalizing Flows (WACV 2022)
  - Experimental results (pixel level AD AUROC on the MVTec-AD dataset)

Encoder	WRN50	WRN50	WRN50	WRN50	WRN50	R18	R18	MNetV3	MNetV3
# of CL	4 →	8	8	8	8	8	8	8	8
# of PL	2	2 →	3	3	3	3	3	3	3
H×W	256	256	256 →	512	512	256 →	512	256 →	512
Type	CFLOW	CFLOW	CFLOW	CFLOW →	UFLOW	CFLOW	CFLOW	CFLOW	CFLOW
Bottle	97.28$\pm$0.03	97.24$\pm$0.03	98.76$\pm$0.01	**98.98**$\pm$0.01	98.83$\pm$0.01	98.47$\pm$0.03	**98.64**$\pm$0.01	98.74	**98.92**
Cable	95.71$\pm$0.01	96.17$\pm$0.07	**97.64**$\pm$0.04	97.12$\pm$0.06	95.29$\pm$0.04	**96.75**$\pm$0.04	96.07$\pm$0.06	**97.62**	97.49
Capsule	98.17$\pm$0.02	98.19$\pm$0.05	**98.98**$\pm$0.00	98.64$\pm$0.02	98.40$\pm$0.12	**98.62**$\pm$0.02	98.28$\pm$0.05	**98.89**	98.75
Carpet	98.50$\pm$0.01	98.55$\pm$0.01	99.23$\pm$0.01	**99.25**$\pm$0.01	99.24$\pm$0.00	99.00$\pm$0.01	**99.29**$\pm$0.00	98.64	**99.00**
Grid	93.77$\pm$0.05	93.88$\pm$0.16	96.89$\pm$0.02	**98.99**$\pm$0.02	98.74$\pm$0.00	93.95$\pm$0.04	**98.53**$\pm$0.01	94.75	**98.81**
Hazelnut	98.08$\pm$0.01	98.13$\pm$0.02	98.82$\pm$0.01	**98.89**$\pm$0.01	98.88$\pm$0.01	**98.81**$\pm$0.01	98.41$\pm$0.01	98.88	**99.00**
Leather	98.92$\pm$0.02	99.00$\pm$0.06	99.61$\pm$0.01	**99.66**$\pm$0.00	99.65$\pm$0.00	99.45$\pm$0.01	**99.51**$\pm$0.02	99.50	**99.64**
Metal Nut	96.72$\pm$0.03	96.72$\pm$0.06	**98.56**$\pm$0.03	98.25$\pm$0.04	98.16$\pm$0.03	**97.59**$\pm$0.05	96.42$\pm$0.03	98.36	**98.78**
Pill	98.46$\pm$0.02	98.46$\pm$0.01	**98.95**$\pm$0.00	98.52$\pm$0.05	98.20$\pm$0.08	**98.34**$\pm$0.02	97.80$\pm$0.05	**98.69**	98.44
Screw	94.98$\pm$0.06	95.28$\pm$0.06	98.10$\pm$0.05	**98.86**$\pm$0.02	98.78$\pm$0.01	97.38$\pm$0.03	**98.40**$\pm$0.03	98.04	**99.09**
Tile	95.52$\pm$0.02	95.66$\pm$0.06	97.71$\pm$0.02	**98.01**$\pm$0.01	97.98$\pm$0.02	95.10$\pm$0.02	**95.80**$\pm$0.10	96.07	**96.48**
Toothbrush	98.02$\pm$0.03	97.98$\pm$0.00	98.56$\pm$0.02	**98.93**$\pm$0.00	98.89$\pm$0.00	98.44$\pm$0.02	**99.00**$\pm$0.01	98.09	**98.80**
Transistor	93.09$\pm$0.28	94.05$\pm$0.11	**93.28**$\pm$0.40	80.52$\pm$0.13	76.28$\pm$0.14	**92.71**$\pm$0.23	83.34$\pm$0.46	**97.79**	95.22
Wood	90.65$\pm$0.10	90.59$\pm$0.07	94.49$\pm$0.03	**96.65**$\pm$0.01	96.56$\pm$0.02	93.51$\pm$0.03	**95.00**$\pm$0.04	92.24	**94.96**
Zipper	96.80$\pm$0.02	97.01$\pm$0.05	98.41$\pm$0.09	**99.08**$\pm$0.02	99.06$\pm$0.01	97.71$\pm$0.06	**98.98**$\pm$0.01	97.50	**99.07**
Average	96.31	96.46	97.87	97.36	96.86	97.06	96.90	97.59	98.16

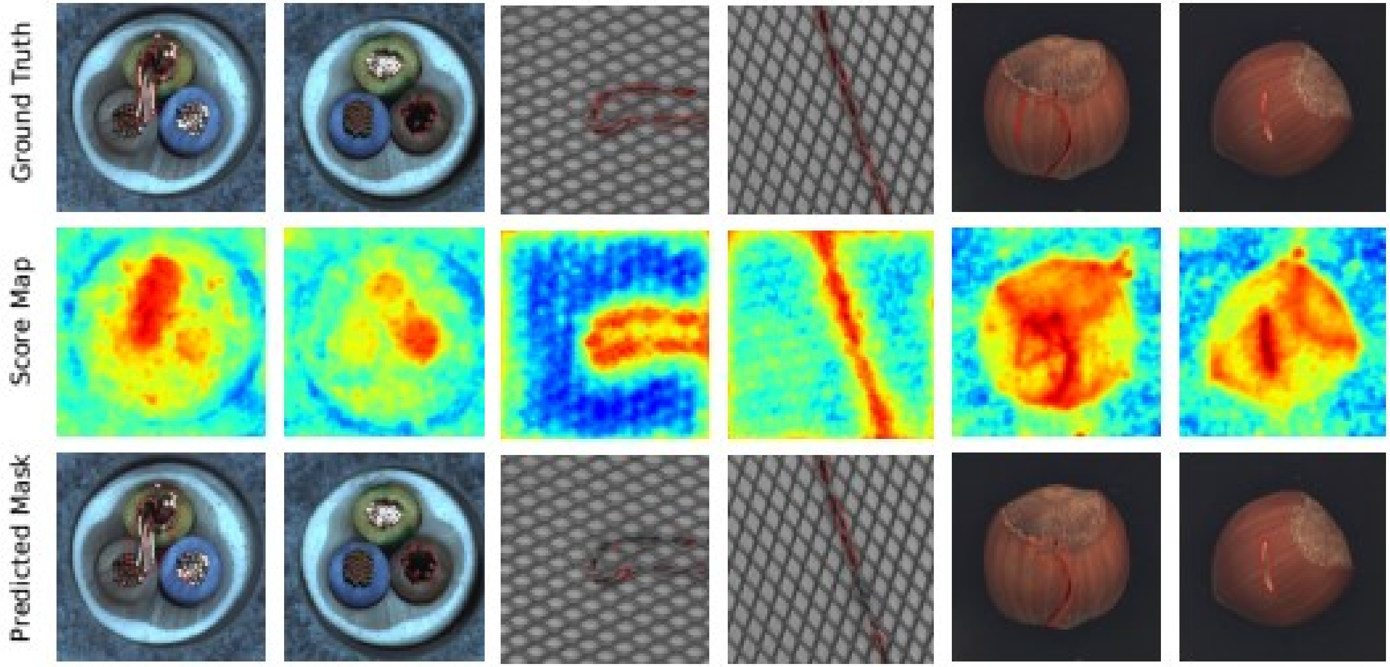
Anomaly detection with Normalizing Flows

- [1] CFLOW-AD: Real-Time Unsupervised Anomaly Detection with Localization via Conditional Normalizing Flows (WACV 2022)
  - Experimental results (comparison with other AD models (AUROC, AUPRO))

Task	Localization		Detection				Localization		
Encoder	ResNet-18				EffNetB4		WideResNet-50		
Class/Model	CutPaste	Ours	CutPaste	Ours	CutPaste	Ours	SPADE	PaDiM	Ours
Bottle	97.6	**98.64**	98.3	100.00	100.0	100.0	(98.4, 95.5)	(98.3, 94.8)	(**98.98**, **96.80**)
Cable	90.0	**96.75**	80.6	**97.62**	96.2	97.59	(97.2, 90.9)	(96.7, 88.8)	(**97.64**, **93.53**)
Capsule	97.4	**98.62**	96.2	93.15	95.4	**97.68**	(**99.0**, **93.7**)	(98.5, 93.5)	(98.98, 93.40)
Carpet	98.3	**99.29**	93.1	98.20	**100.0**	98.73	(97.5, 94.7)	(99.1, 96.2)	(**99.25**, **97.70**)
Grid	97.5	**98.53**	**99.9**	98.97	99.1	99.60	(93.7, 86.7)	(97.3, 94.6)	(**98.99**, **96.08**)
Hazelnut	97.3	**98.81**	97.3	99.91	99.9	**99.98**	(**99.1**, 95.4)	(98.2, 92.6)	(98.89, **96.68**)
Leather	99.5	**99.51**	100.0	100.00	100.0	100.0	(97.6, 97.2)	(98.9, 88.8)	(**99.66**, **99.35**)
Metal Nut	93.1	**97.59**	**99.3**	98.45	98.6	99.26	(98.1, **94.4**)	(97.2, 85.6)	(**98.56**, 91.65)
Pill	95.7	**98.34**	92.4	93.02	93.3	**96.82**	(96.5, 94.6)	(95.7, 92.7)	(**98.95**, **95.39**)
Screw	96.7	**98.40**	86.3	85.94	86.6	**91.89**	(**98.9**, **96.0**)	(98.5, 94.4)	(98.86, 95.30)
Tile	90.5	**95.80**	93.4	98.40	99.8	**99.88**	(87.4, 75.9)	(94.1, 86.0)	(**98.01**, **94.34**)
Toothbrush	98.1	**99.00**	98.3	**99.86**	90.7	99.65	(97.9, 93.5)	(98.8, 93.1)	(**98.93**, **95.06**)
Transistor	93.0	**97.69**	95.5	93.04	**97.5**	95.21	(94.1, **87.4**)	(97.5, 84.5)	(**97.99**, 81.40)
Wood	**95.5**	95.00	98.6	98.59	**99.8**	99.12	(88.5, **97.4**)	(94.9, 91.1)	(**96.65**, 95.79)
Zipper	**99.3**	98.98	99.4	96.15	**99.9**	98.48	(96.5, 92.6)	(98.5, 95.9)	(**99.08**, **96.60**)
Average	96.0	**98.06**	95.2	96.75	97.1	**98.26**	(96.0, 91.7)	(97.5, 92.1)	(**98.62**, **94.60**)

Anomaly detection with Normalizing Flows

- [1] CFLOW-AD: Real-Time Unsupervised Anomaly Detection with Localization via Conditional Normalizing Flows (WACV 2022)
  - Experimental results (anomaly localization results)



Thank you for listening