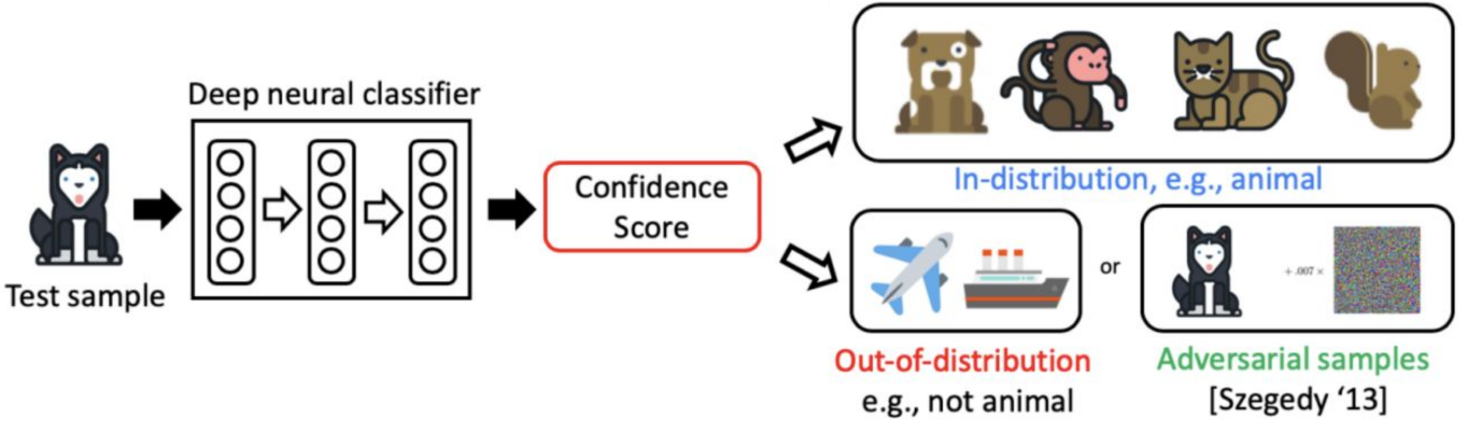
Out-of-Distribution Detection Methods and its application on Colorectal Pathology Image

Kangyeol Kim, 20191106

Problem of interest: Detecting Abnormal Samples

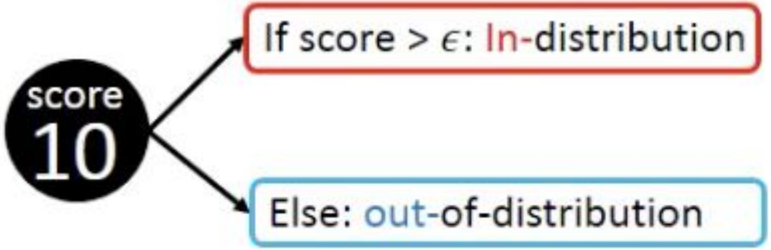
- **Detecting abnormal samples (a.k.a. novelty detection)**
  - Given a pre-trained (deep) classifier,
  - Detect whether a test sample is from the training distribution (In-distribution) or not (Out-of-distribution, Adversarial samples)



Problem of interest: Detecting Abnormal Samples

- **Detecting abnormal samples (a.k.a. novelty detection)**
  - Given a pre-trained (deep) classifier,
  - Detect whether a test sample is from the training distribution (In-distribution) or not (Out-of-distribution, Adversarial samples)
- **Softmax Threshold-based Detector** [Hendryck et al., 2017]

Maximum softmax value



[Hendryck et al., 2017] A baseline for detecting misclassified and out-of-distribution examples in neural networks, ICLR

Previous work: Out-of-DIstribution detector for Neural networks (ODIN)

- ODIN propose two components for detecting out-of-distribution
- **Temperature Scaling**

$$S_i(\boldsymbol{x};T) = \frac{\exp\left(f_i(\boldsymbol{x})/T\right)}{\sum_{j=1}^{N}\exp\left(f_j(\boldsymbol{x})/T\right)},$$

- **Input Preprocessing;** adding small perturbations to increase sx score

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} - \varepsilon \mathrm{sign}(-\nabla_{\boldsymbol{x}} \log S_{\hat{y}}(\boldsymbol{x};T)),$$

[Shiyu Liang., 2017] Enhancing the Reliability of Out-of-Distribution Image Detection in Neural Networks, ICLR

Previous work: Out-of-DIstribution detector for Neural networks (ODIN)

- **Temperature Scaling**

$$S_{\hat{y}}(\boldsymbol{x};T) = \frac{\exp\left(f_{\hat{y}}(\boldsymbol{x})/T\right)}{\sum_{i=1}^{N}\exp(f_i(\boldsymbol{x})/T)}$$

$$= \frac{1}{\sum_{i=1}^{N}\exp\left(\frac{f_i(\boldsymbol{x})-f_{\hat{y}}(\boldsymbol{x})}{T}\right)}$$

$$= \frac{1}{\sum_{i=1}^{N}\left[1+\frac{f_i(\boldsymbol{x})-f_{\hat{y}}(\boldsymbol{x})}{T}+\frac{1}{2!}\frac{(f_i(\boldsymbol{x})-f_{\hat{y}}(\boldsymbol{x}))^2}{T^2}+o\left(\frac{1}{T^2}\right)\right]}$$

by Taylor expansion

$$\approx \frac{1}{N-\frac{1}{T}\underline{\sum_{i=1}^{N}[f_{\hat{y}}(\boldsymbol{x})-f_i(\boldsymbol{x})]}+\frac{1}{2T^2}\underline{\sum_{i=1}^{N}[f_i(\boldsymbol{x})-f_{\hat{y}}(\boldsymbol{x})]^2}}$$

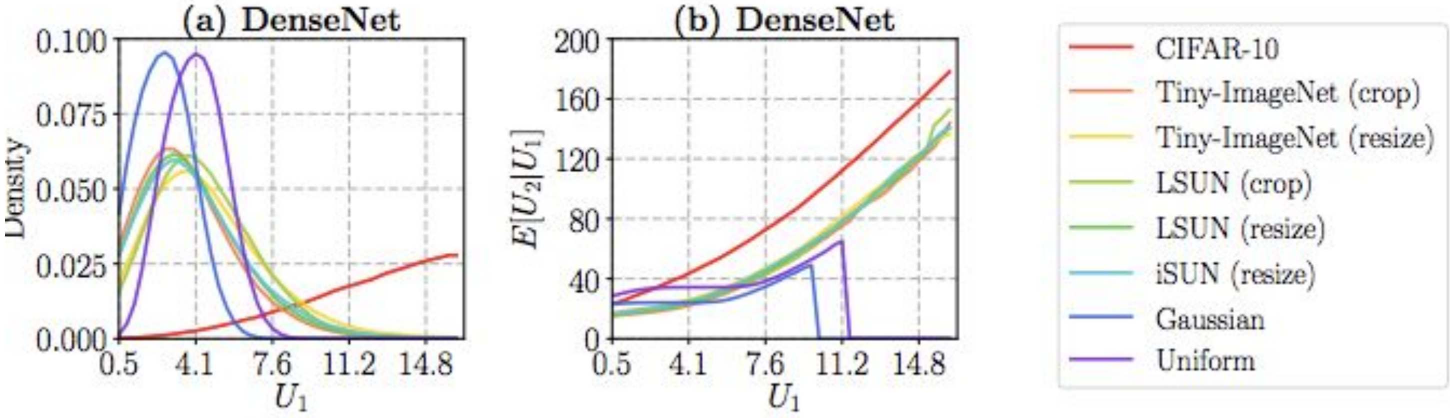
$$\underline{U_1(\boldsymbol{x}) = \frac{1}{N-1}\sum_{i\neq\hat{y}}[f_{\hat{y}}(\boldsymbol{x})-f_i(\boldsymbol{x})]}$$

$$\underline{U_2(\boldsymbol{x}) = \frac{1}{N-1}\sum_{i\neq\hat{y}}[f_{\hat{y}}(\boldsymbol{x})-f_i(\boldsymbol{x})]^2.}$$

[Shiyu Liang., 2017] Enhancing the Reliability of Out-of-Distribution Image Detection in Neural Networks, ICLR

Previous work: Out-of-DIstribution detector for Neural networks (ODIN)

- **Temperature Scaling** $U_1(\boldsymbol{x}) = \frac{1}{N-1}\sum_{i\neq\hat{y}}[f_{\hat{y}}(\boldsymbol{x}) - f_i(\boldsymbol{x})]$ $U_2(\boldsymbol{x}) = \frac{1}{N-1}\sum_{i\neq\hat{y}}[f_{\hat{y}}(\boldsymbol{x}) - f_i(\boldsymbol{x})]^2.$



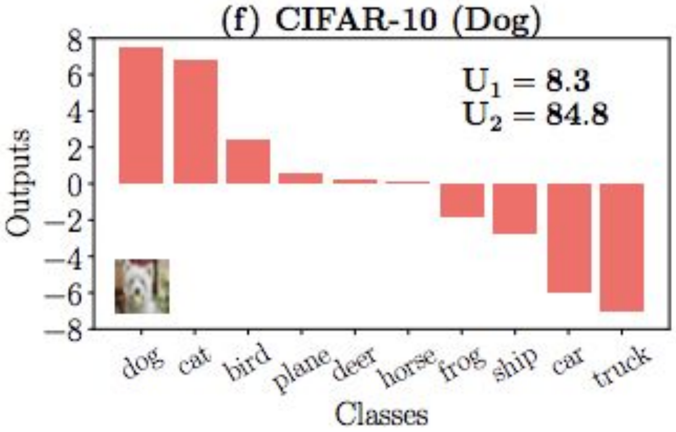
in
out
out
out
out
out
out
out

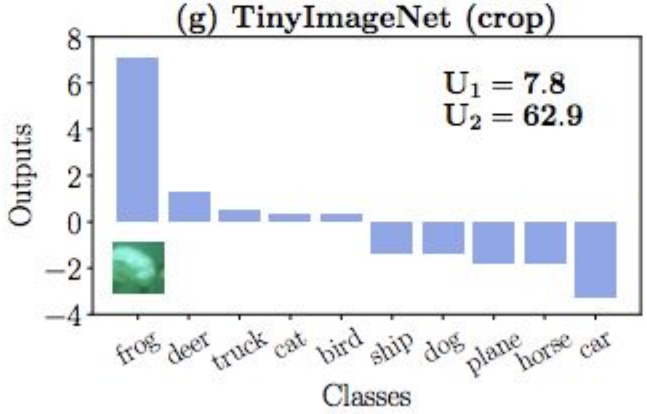
- Expectation of U1 of in-distribution is larger than that of out-of-distribution
- When U1 is similar, U2 of in-distribution is larger than that of out-of-distribution!

[Shiyu Liang., 2017] Enhancing the Reliability of Out-of-Distribution Image Detection in Neural Networks, ICLR

Previous work: Out-of-DIstribution detector for Neural networks (ODIN)

- **Temperature Scaling** $U_1(\boldsymbol{x}) = \frac{1}{N-1}\sum_{i\neq\hat{y}}[f_{\hat{y}}(\boldsymbol{x}) - f_i(\boldsymbol{x})]$ $U_2(\boldsymbol{x}) = \frac{1}{N-1}\sum_{i\neq\hat{y}}[f_{\hat{y}}(\boldsymbol{x}) - f_i(\boldsymbol{x})]^2.$





[Shiyu Liang., 2017] Enhancing the Reliability of Out-of-Distribution Image Detection in Neural Networks, ICLR

Previous work: Out-of-DIstribution detector for Neural networks (ODIN)

- **Temperature Scaling** $U_1(\boldsymbol{x}) = \frac{1}{N-1}\sum_{i \neq \hat{y}}[f_{\hat{y}}(\boldsymbol{x}) - f_i(\boldsymbol{x})]$ $U_2(\boldsymbol{x}) = \frac{1}{N-1}\sum_{i \neq \hat{y}}[f_{\hat{y}}(\boldsymbol{x}) - f_i(\boldsymbol{x})]^2.$

$$S_{\hat{y}}(\boldsymbol{x};T) = \frac{\exp\left(f_{\hat{y}}(\boldsymbol{x})/T\right)}{\sum_{i=1}^{N}\exp(f_i(\boldsymbol{x})/T)} \approx \frac{1}{N - \frac{1}{T}\sum_{i=1}^{N}[f_{\hat{y}}(\boldsymbol{x}) - f_i(\boldsymbol{x})] + \frac{1}{2T^2}\sum_{i=1}^{N}[f_i(\boldsymbol{x}) - f_{\hat{y}}(\boldsymbol{x})]^2}$$

$$\propto (U_1 - U_2/2T)T \quad \left(f(x) \propto g(x) \Leftrightarrow f(x) \propto -\frac{1}{g(x)}\right)$$

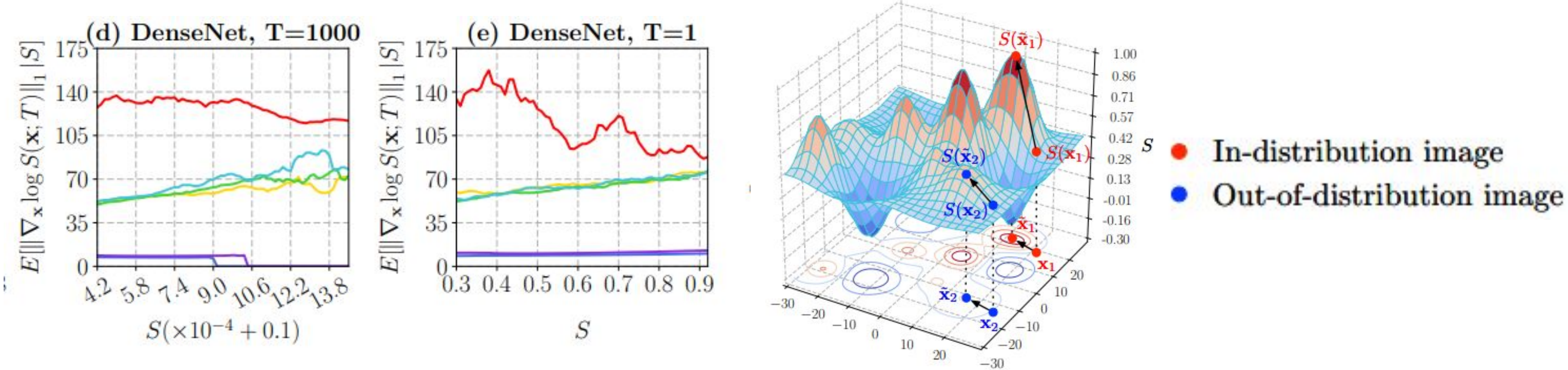
- **Problem without T -** Large U2 value of in-distribution decreases maximum softmax output (less confident outputs)
- **Solution with T -** Sufficient T can alleviate above problem
- **Effect of T** - Appropriate selection process is necessary

[Shiyu Liang., 2017] Enhancing the Reliability of Out-of-Distribution Image Detection in Neural Networks, ICLR

Previous work: Out-of-DIstribution detector for Neural networks (ODIN)

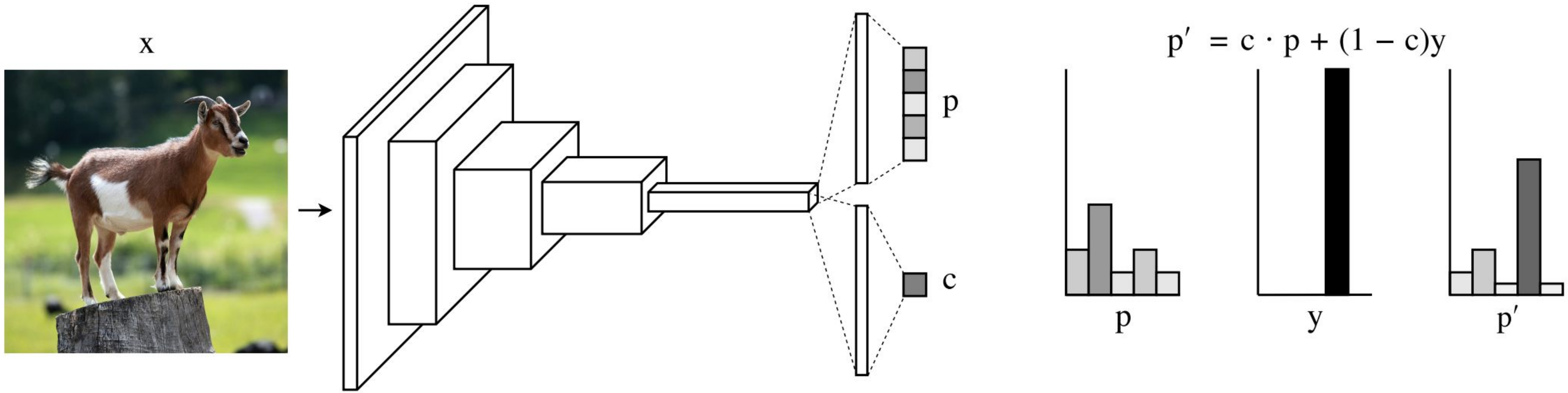
- **Input Preprocessing** $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \varepsilon \text{sign}(-\nabla_{\boldsymbol{x}} \log S_{\hat{y}}(\boldsymbol{x}; T)),$
- **Analysis,** first order Taylor expansion of perturbed image:

$$\log S_{\hat{y}}(\tilde{\boldsymbol{x}}; T) = \log S_{\hat{y}}(\boldsymbol{x}; T) + \varepsilon \left\| \nabla_{\boldsymbol{x}} \log S_{\hat{y}}(\boldsymbol{x}; T) \right\|_1 + o(\varepsilon),$$



[Shiyu Liang., 2017] Enhancing the Reliability of Out-of-Distribution Image Detection in Neural Networks, ICLR

Previous work: Learning Confidence for Out-of-Distribution in Neural Networks



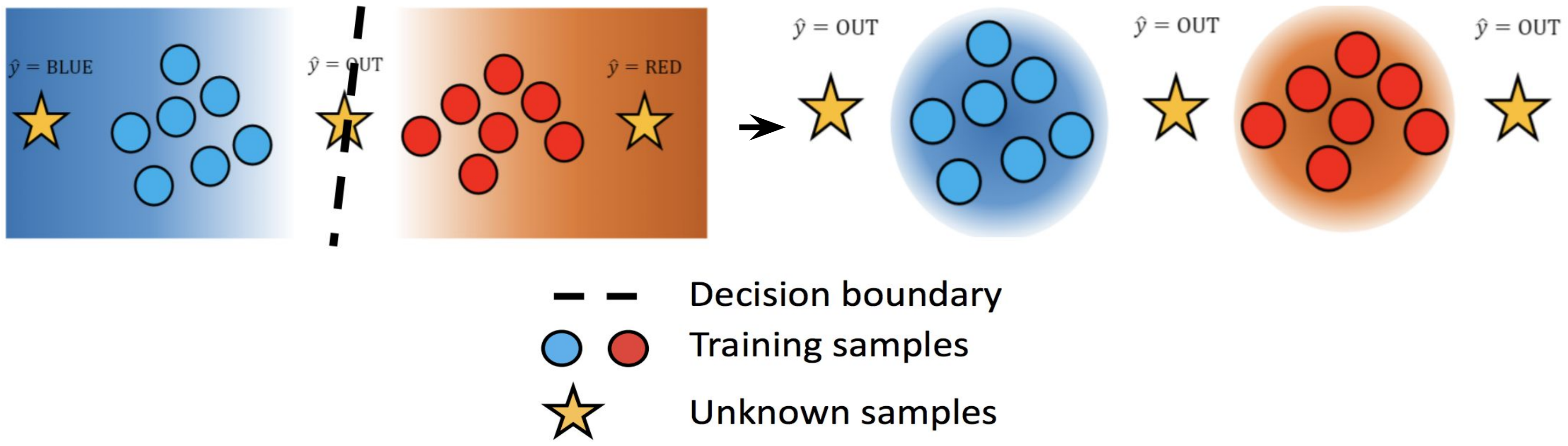
- During training, we trust neural networks output in the degree of confidence score (c). And take a hint from ground truth label if c is low.
- Penalty term to prevent the networks from easily taking c as 0 to minimize loss:

$$\mathcal{L}_c = -\log(c).$$

[Terrance DeVries and Graham W. Taylor, 2018] Learning Confidence for Out-of-Distribution in Neural Networks, arXiv

Previous work: A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks

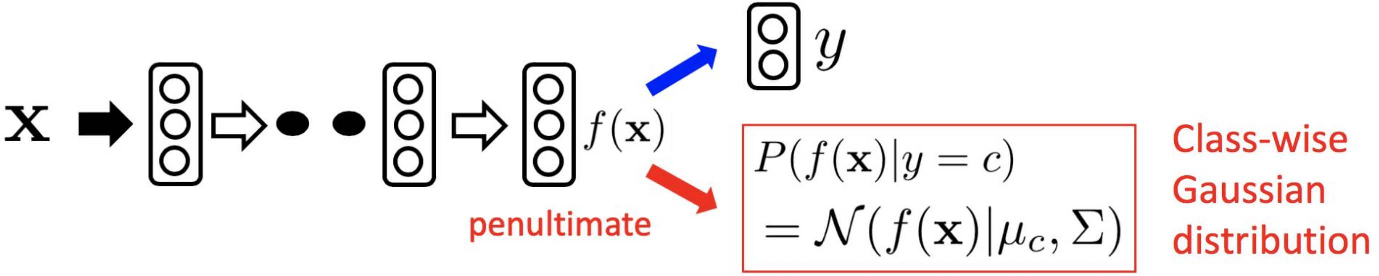
- This paper considers data distribution P(x|y) rather posterior distribution P(y|x) to find out-of-distribution samples.



[Lee et al., 2018] A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks, Neurips

Previous work: A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks

- **Main idea: Post-processing a generative classifier -** Given a pre-trained softmax classifier, the paper post-process a simple generative classifier on hidden feature spaces:



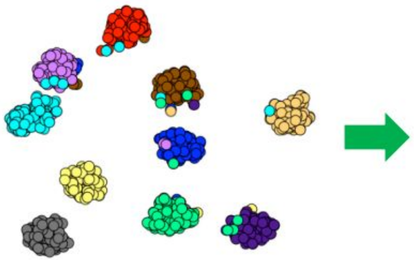
- **How to estimate parameters? -** Empirical class mean and covariance matrix via training set

$$\widehat{\mu}_c = \frac{1}{N_c}\sum_{i:y_i=c} f(\mathbf{x}_i), \quad \widehat{\mathbf{\Sigma}} = \frac{1}{N}\sum_c \sum_{i:y_i=c} \left(f(\mathbf{x}_i) - \widehat{\mu}_c\right)\left(f(\mathbf{x}_i) - \widehat{\mu}_c\right)^\top$$

[Lee et al., 2018] A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks, Neurips

Previous work: A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks

- **Why Gaussian?** - the posterior distribution of the generative classifier (with tied covariance) is equivalent to softmax classifier



[T-SNE of penultimate features]

- **Empirical observation**
  - ResNet-34 trained on CIFAR-10
  - Hidden features follow class-conditional unimodal distributions

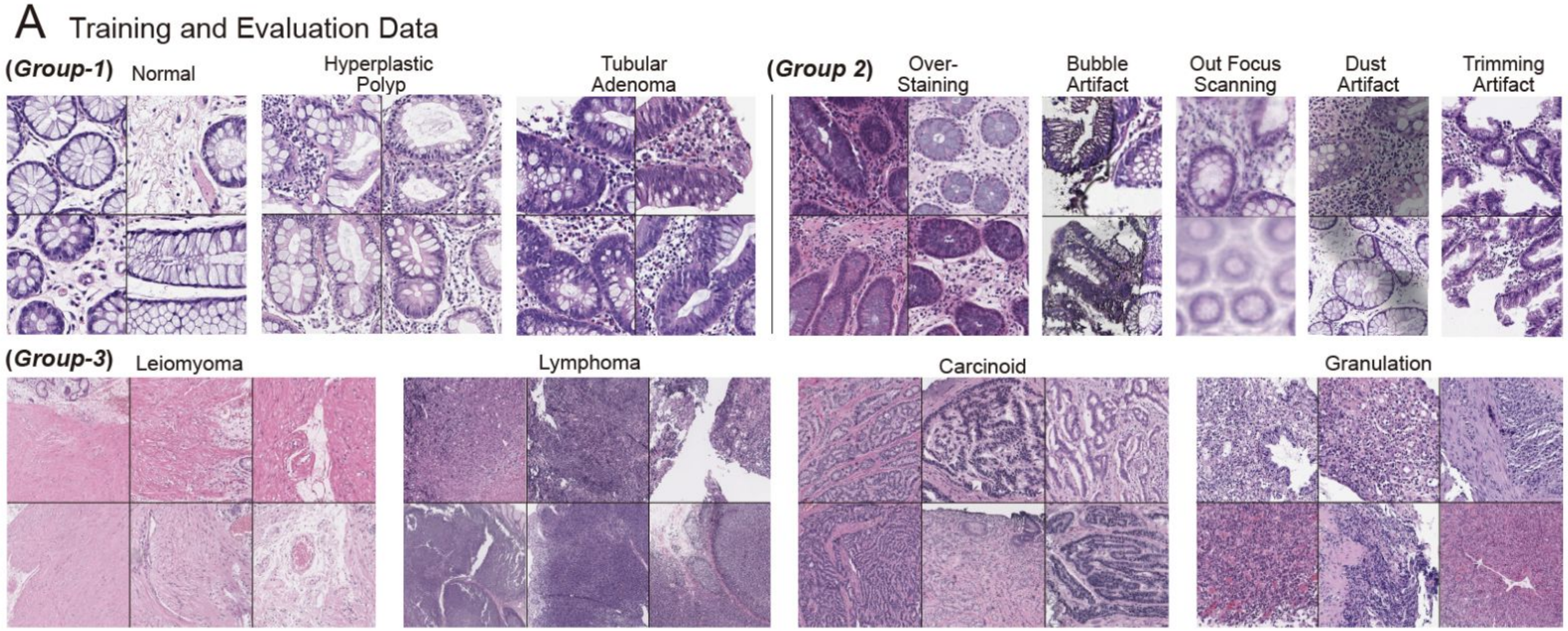
- **Computing confidence score: Mahalanobis distance** between a test sample and a closest class Gaussin

$$M(\mathbf{x}) = \max_c \log P(f(\mathbf{x})|y=c)$$
$$= \max_c \; -(f(\mathbf{x}) - \widehat{\mu}_c)^\top \widehat{\Sigma}(f(\mathbf{x}) - \widehat{\mu}_c)$$

[Lee et al., 2018] A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks, Neurips

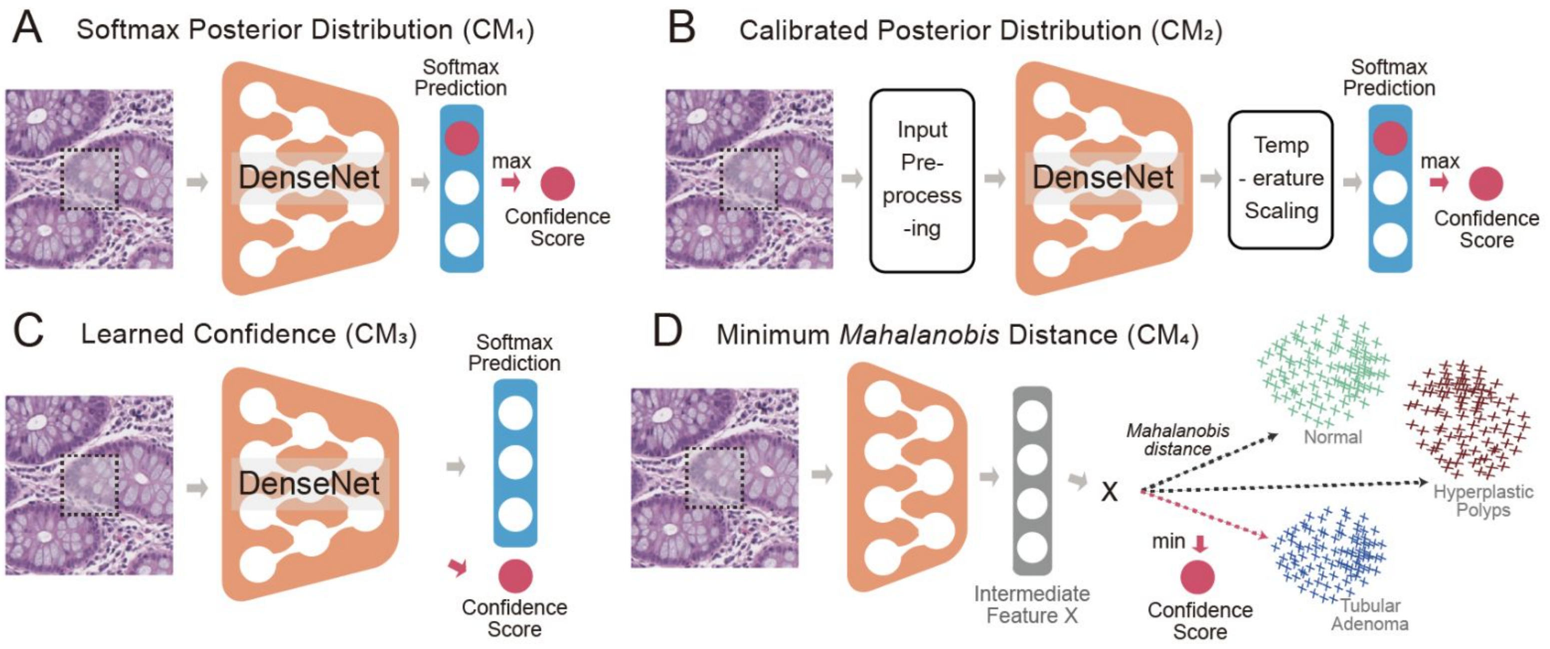
My work: Colorectal Pathology Image Classification via Uncertainty-Aware Deep Neural Networks

- **Motivation; Colorectal Pathology out-of-distribution detection**



My work: Colorectal Pathology Image Classification via Uncertainty-Aware Deep Neural Networks

- **Confidence Methods**



My work: Colorectal Pathology Image Classification via Uncertainty-Aware Deep Neural Networks

Evaluated Set	**Type**	**FPR at 95% TPR(%)**				**Detection Error(%)**			
		CM_1	CM_2	CM_3	CM_4	CM_1	CM_2	CM_3	CM_4
Group-2	Over-staining	95.2	97.2	76.0	**0.0**	49.7	46.2	38.0	**0.2**
	Bubble	84.6	98.8	82.7	**13.4**	31.4	37.3	29.7	**8.9**
	Dust	96.2	98.6	94.8	**6.7**	42.8	33.2	31.5	**5.5**
	Out-focused	91.3	98.0	95.9	**0.0**	45.5	39.6	41.5	**0.7**
	Trimming	74.3	100.0	97.1	**44.2**	16.2	30.2	33.0	**19.6**
Group-3	Lymphoma	90.7	91.2	93.6	**2.1**	42.9	33.3	46.2	**2.8**
	Leiomyoma	86.7	85.9	94.3	**1.6**	40.0	38.0	48.5	**2.3**
	Carcinoid	80.9	86.7	83.7	**1.9**	36.0	31.2	38.8	**2.8**
	Granulation	85.2	72.5	88.8	**5.7**	35.3	29.3	34.9	**4.8**
Evaluated Set	**Type**	**AUROC**				**AUPR**			
		CM_1	CM_2	CM_3	CM_4	CM_1	CM_2	CM_3	CM_4
Group-2	Over-staining	0.729	0.500	0.685	**1.00**	0.478	0.502	0.594	**1.00**
	Bubble	0.812	0.635	0.760	**0.968**	0.658	0.669	0.766	**0.967**
	Dust	0.752	0.627	0.680	**0.988**	0.546	0.668	0.698	**0.987**
	Out-focused	0.753	0.521	0.561	**1.00**	0.529	0.561	0.557	**1.00**
	Trimming	**0.887**	0.587	0.713	0.877	**0.900**	0.732	0.755	0.870
Group-3	Lymphoma	0.766	0.627	0.585	**0.996**	0.549	0.721	0.527	**0.996**
	Leiomyoma	0.786	0.650	0.566	**0.995**	0.575	0.646	0.486	**0.998**
	Carcinoid	0.809	0.676	0.682	**0.995**	0.611	0.720	0.611	**0.995**
	Granulation	0.800	0.762	0.671	**0.978**	0.616	0.784	0.683	**0.962**

Table 2. Patch-level Out-of-distribution Detection Comparisons