Confident Deep Learning

Kimin Lee

Ph.D. student at KAIST

NAVER Tech Talk

Outline

- Introduction
  - Predictive uncertainty of deep neural networks
  - Summary
- How to train confident neural networks
  - Training Confidence-Calibrated Classifiers for Detecting Out-of-Distribution Samples [Lee' 18a]
- Applications
  - Confident Multiple Choice Learning [Lee' 17]
  - Hierarchical novelty detection [Lee' 18b]
- Conclusion

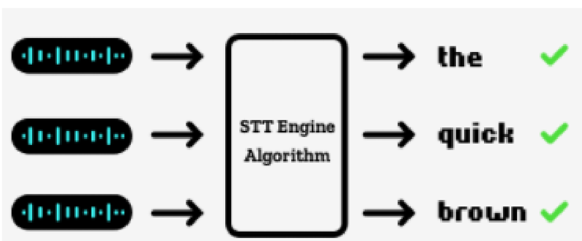
[Lee' 18a] Lee, K., Lee, H., Lee, K. and Shin, J. Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples. I n ICLR, 2018.
[Lee' 17] Lee, K., Hwang, C., Park, K. and Shin, J. Confident Multiple Choice Learning. *In ICML, 2017.*
[Lee' 18b] Lee, K., Lee, Min. K, Zhang, Y. Shin. J, Lee, H. Hierarchical Novelty Detection for Visual Object Recognition, In CVPR, 2018.

Introduction: Predictive uncertainty of deep neural networks (DNNs)

- Supervised learning (e.g., regression and classification)
  - Objective: finding an unknown target distribution, i.e., P(Y|X)

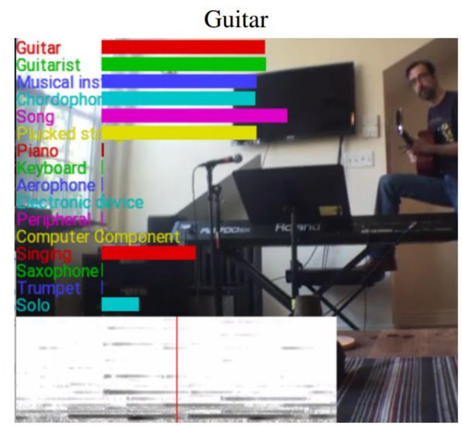
Input space  Output space

  - Recent advances in deep learning have dramatically improved accuracy on several supervised learning tasks

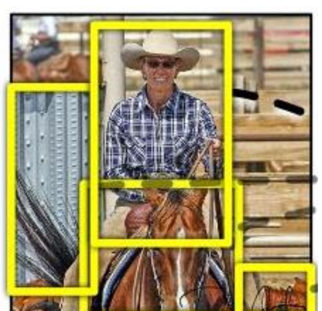


Speech recognition [Amodei’ 16]

Image classification [He’ 16]



Audio recognition [Hershey’ 17]



Objective detection [Girshick’ 15]

[Amodei’ 16] Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G. and Chen, J. Deep speech 2: End-to-end speech recognition in english and mandarin. In *ICML, 2016.*

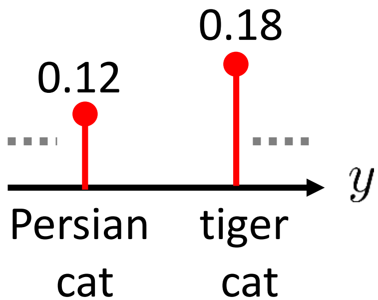
[He’ 16] He, K., Zhang, X., Ren, S. and Sun, J. Deep residual learning for image recognition. In *CVPR, 2016.*

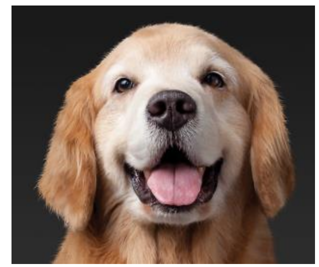
[Hershey’ 17] Hershey, S., Chaudhuri, S., Ellis, D.P., Gemmeke, J.F., Jansen, A., Moore, R.C., Plakal, M., Platt, D., Saurous, R.A., Seybold, B. and Slaney, M. CNN architectures for large-scale audio classification. In *ICASSP, 2017.*

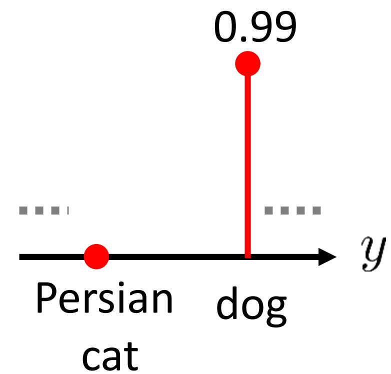
[Girshick’ 15] Girshick, Ross. Fast r-cnn. In ICCV, pp. 1440–1448, 2015

Introduction: Predictive uncertainty of deep neural networks (DNNs)

- Uncertainty of predictive distribution is important in DNN's applications
  - What is predictive uncertainty?
    - As a example, consider classification task







    - It represents a confidence about prediction!
  - For example, it can be measured as follows:
    - Entropy of predictive distribution [Lakshminarayanan' 17]

$$\sum_y -P(y|\mathbf{x}) \log P(y|\mathbf{x})$$

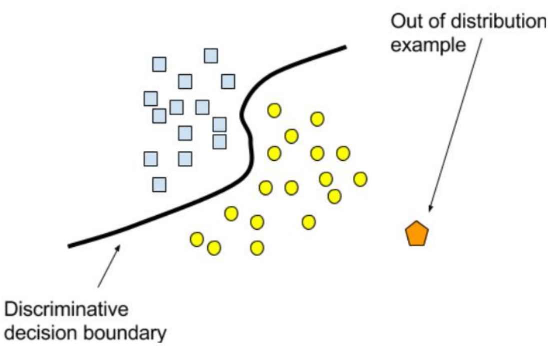
    - Maximum value of predictive distribution [Hendrycks' 17]

$$\max_y P(y|\mathbf{x})$$

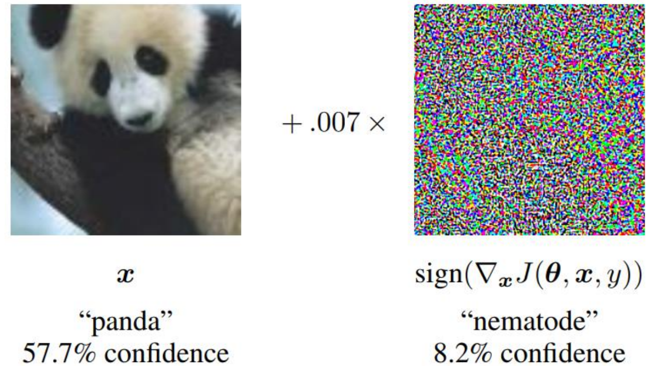
[Lakshminarayanan' 17] Lakshminarayanan, B., Pritzel, A. and Blundell, C., Simple and scalable predictive uncertainty estimation using deep ensembles. In *NIPS, 2017.*
[Henderycks' 17] Hendrycks, D. and Gimpel, K., A baseline for detecting misclassified and out-of-distribution examples in neural networks. *In ICLR 2017.*

Introduction: Predictive uncertainty of deep neural networks (DNNs)

- Predictive uncertainty is related to many machine learning problems:



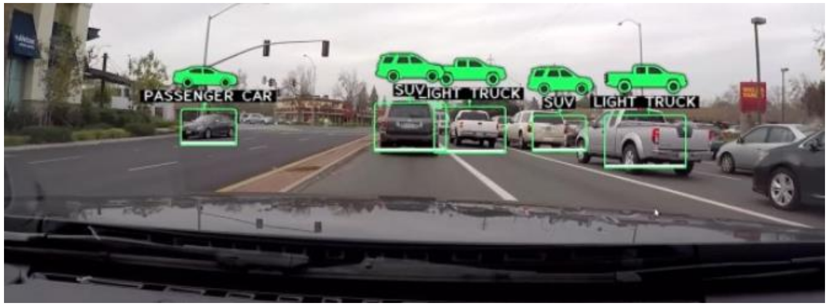
Novelty detection [Hendrycks' 17]



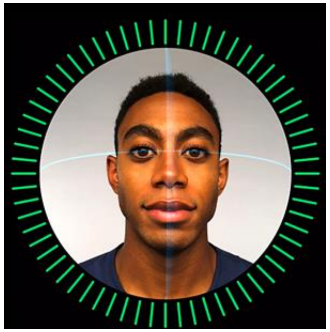
Adversarial detection [Song' 18]

Ensemble learning [Lee' 17]

- Predictive uncertainty is also indispensable when deploying DNNs in real-world systems [Dario' 16]



Autonomous drive

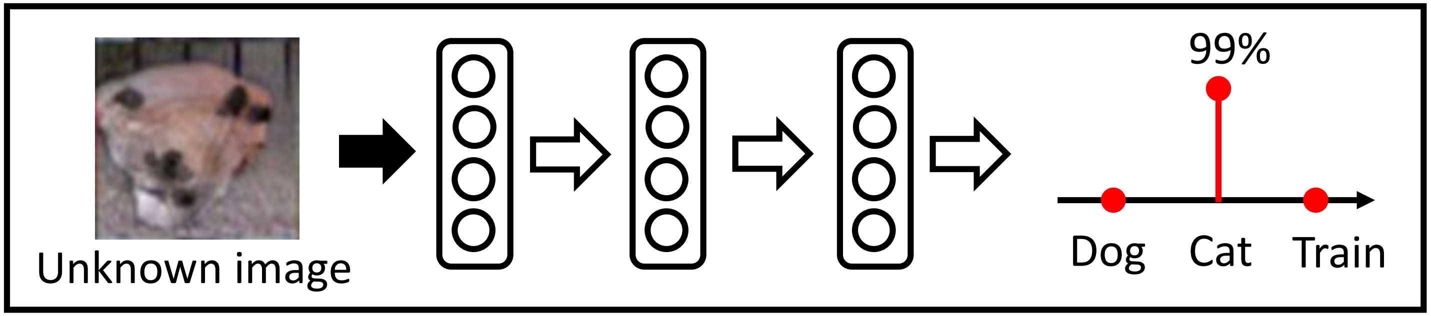


Secure authentication system

[Dario' 16] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.
[Henderycks' 17] Hendrycks, D. and Gimpel, K., A baseline for detecting misclassified and out-of-distribution examples in neural networks. *In ICLR 2017.*
[Guo' 17] Guo, C., Pleiss, G., Sun, Y. and Weinberger, K.Q., 2017. On Calibration of Modern Neural Networks. *In ICML 2017.*
[Goodfellow' 14] Goodfellow, I.J., Shlens, J. and Szegedy, C., 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.
[Srivastava' 14] Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., Dropout: a simple way to prevent neural networks from overfitting. JMLR. 2014.

Introduction: Predictive uncertainty of deep neural networks (DNNs)

- However, DNNs do not capture their predictive uncertainty



  - E.g., DNNs trained to classify MNIST images often produce high confident probability 91% even for random noise [Henderycks' 17]
  - Challenge arises in improving the quality of the predictive uncertainty!
- Main topic of this presentation
  - How to train confident neural networks?
    - Training confidence-calibrated classifiers for detecting out-of-distribution samples [Lee' 18a]
  - Applications
    - Confident multiple choice learning [Lee' 17]
    - Hierarchical novelty detection [Lee' 18b]

[Henderycks' 17] Hendrycks, D. and Gimpel, K., A baseline for detecting misclassified and out-of-distribution examples in neural networks. *In ICLR 2017.*
[Lee' 18a] Lee, K., Lee, H., Lee, K. and Shin, J. Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples. In ICLR 2018.
[Lee' 17] Lee, K., Hwang, C., Park, K. and Shin, J. Confident Multiple Choice Learning. *In ICML, 2017.*
[Lee' 18b] Lee, K., Lee, Min. K, Zhang, Y. Shin. J, Lee, H. Hierarchical Novelty Detection for Visual Object Recognition, In CVPR, 2018.

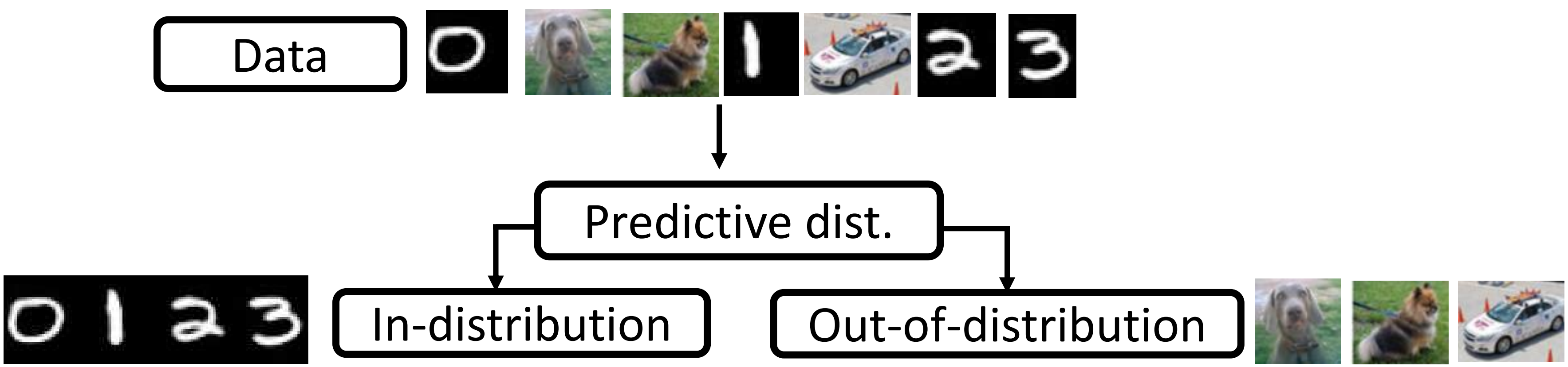
Outline

- Introduction
  - Predictive uncertainty of deep neural networks
  - Summary
- How to train confident neural networks
  - Training Confidence-Calibrated Classifiers for Detecting Out-of-Distribution Samples [Lee’ 18a]
- Applications
  - Confident Multiple Choice Learning [Lee’ 17]
  - Hierarchical novelty detection [Lee’ 18b]
- Conclusion

[Lee’ 18a] Lee, K., Lee, H., Lee, K. and Shin, J. Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples. In ICLR, 2018.
[Lee’ 17] Lee, K., Hwang, C., Park, K. and Shin, J. Confident Multiple Choice Learning. *In ICML, 2017.*
[Lee’ 18b] Lee, K., Lee, Min. K, Zhang, Y. Shin. J, Lee, H. Hierarchical Novelty Detection for Visual Object Recognition, In CVPR, 2018.

How to Train Confident Neural Networks?

- Related problem
  - Detecting out-of-distribution [Hendrycks’ 17]
    - Detect whether a test sample is from in-distribution (i.e., training distribution by classifier) or out-of-distribution
  - E.g., image classification
    - Assume a classifier trains handwritten digits (denoted as in-distribution)
    - Detecting out-of-distribution

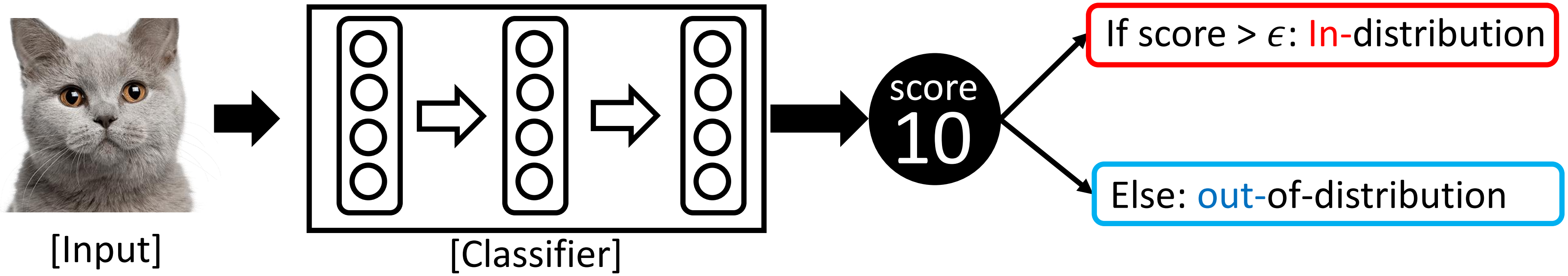


  - Performance of detector reflects confidence of predictive distribution!

[Henderycks’ 17] Hendrycks, D. and Gimpel, K., A baseline for detecting misclassified and out-of-distribution examples in neural networks. *In ICLR 2017.*
[Guo’ 17] Guo, C., Pleiss, G., Sun, Y. and Weinberger, K.Q., 2017. On Calibration of Modern Neural Networks. *In ICML 2017.*
[Liang’ 17] Liang, S., Li, Y. and Srikant, R., 2017. Principled Detection of Out-of-Distribution Examples in Neural Networks. *arXiv preprint arXiv:1706.02690*.

Related Work

- Threshold-based Detector [Guo’ 17, Hendrycks’17, Liang’ 18]



  - How to define the score?
    - Baseline detector [Hendrycks’17]
      - Confidence score = maximum value of predictive distribution
    - Temperature scaling [Guo’ 17]
      - Confidence score = maximum value of scaled predictive distribution

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

Output of neural networks

- Limitations
  - Performance of prior works highly depends on how to train the classifiers

[Henderycks’ 17] Hendrycks, D. and Gimpel, K., A baseline for detecting misclassified and out-of-distribution examples in neural networks. *In ICLR 2017.*
[Guo’ 17] Guo, C., Pleiss, G., Sun, Y. and Weinberger, K.Q., 2017. On Calibration of Modern Neural Networks. *In ICML 2017.*
[Liang’ 17] Liang, S., Li, Y. and Srikant, R., 2017. Principled Detection of Out-of-Distribution Examples in Neural Networks. *In ICLR, 2018.*

Our Contributions

- Main components of our contribution
  - New loss
    - Confident loss for confident classifier
  - New generative adversarial network (GAN)
    - GAN for generating out-of-distribution samples
  - New training method
    - Joint training of classifier and GAN
- Experimental results
  - Our method drastically improves the detection performance
  - VGGNet trained by our method improves TPR compared to the baseline:
    - 14.0%→39.1% and 46.3% → 98.9% on CIFAR-10 and SVHN, respectively
  - Providing visual understandings on the proposed method

Contribution 1: Confident Loss

- Confident loss
  - Minimize the KL divergence on data from out-of-distribution

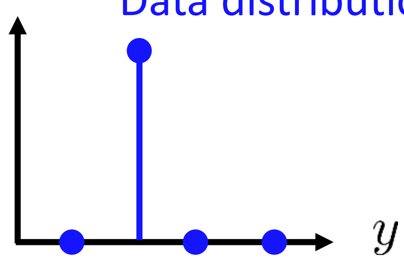
$$\boxed{\min_{\theta} \ \mathbb{E}_{P_{\text{in}}(\widehat{\mathbf{x}},\widehat{y})}\Big[-\log P_{\theta}\left(y=\widehat{y}|\widehat{\mathbf{x}}\right)\Big] + \beta \ \mathbb{E}_{P_{\text{out}}(\mathbf{x})}\Big[KL\left(\mathcal{U}\left(y\right) \parallel P_{\theta}\left(y|\mathbf{x}\right)\right)\Big],}$$

Data from in-dist　　　　Data from out-of-dist

- Interpretation
  - Assigning higher maximum prediction values to in-distribution samples than o ut-of-distribution ones

$P_\theta(y|\mathbf{x}) \rightarrow P(y|\mathbf{x})$

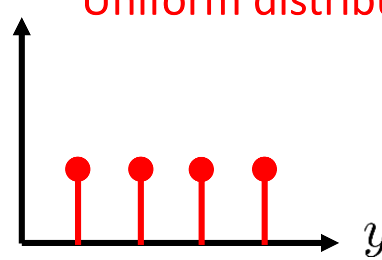
Data distribution



[In-distribution data]

$P_\theta(y|\mathbf{x}) \rightarrow \mathcal{U}(y)$

Uniform distribution



[Out-of-distribution data]

"Zero confidence"

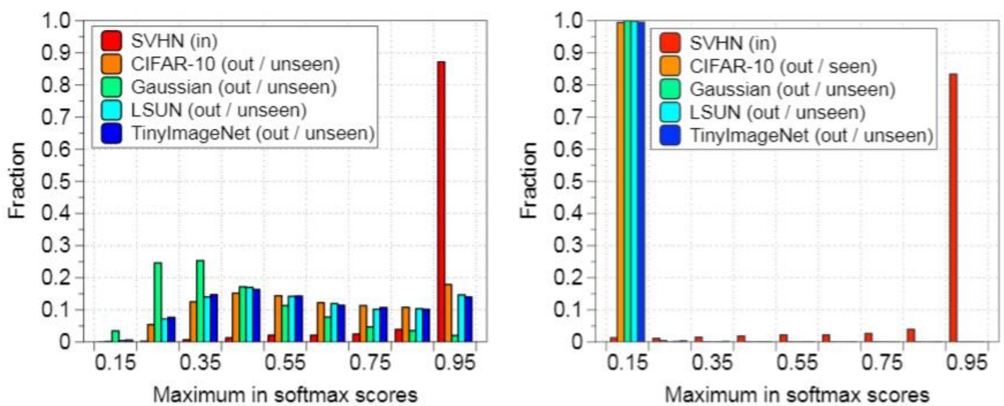
Contribution 1: Confident Loss

- Confident loss
  - Minimize the KL divergence on data from out-of-distribution

$$\boxed{\min_{\theta} \ \underbrace{\mathbb{E}_{P_{\text{in}}(\widehat{\mathbf{x}},\widehat{y})}}_{} \left[-\log P_{\theta}\left(y=\widehat{y}|\widehat{\mathbf{x}}\right) \right] + \beta \ \underbrace{\mathbb{E}_{P_{\text{out}}(\mathbf{x})}}_{} \left[KL\left(\mathcal{U}\left(y\right) \parallel P_{\theta}\left(y|\mathbf{x}\right)\right) \right],}$$

Data from in-dist　　　　　Data from out-of-dist

  - Interpretation
    - Assigning higher maximum prediction values to in-distribution samples than o ut-of-distribution ones
  - Effects of confidence loss
    - Fraction of the maximum prediction value from simple CNNs (2 Conv + 3 FC)
    - KL divergence term is optimized using CIFAR-10 training data



(a) Cross entropy loss　　(b) Confidence loss in (1)

Contribution 2. GAN for Generating Out-of-Distribution Samples

- Main issues of confidence loss
  - How to optimize the KL divergence loss?
    - The number of out-of-distribution samples might be almost infinite to cover the entire space
- Our intuition
  - Samples close to in-distribution could be more effective in improving the detection performance

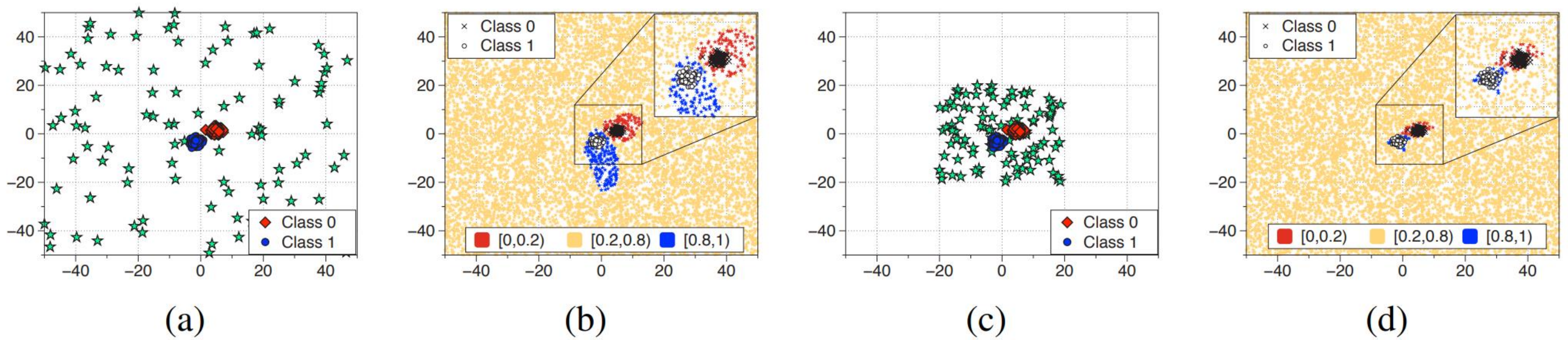


Figure 2: Illustrating the behavior of classifier under different datasets. We generate the out-of-distribution samples from (a) 2D box $[-50, 50]^2$, and show (b) the corresponding decision boundary of classifier. We also generate the out-of-distribution samples from (c) 2D box $[-20, 20]^2$, and show (d) the corresponding decision boundary of classifier.

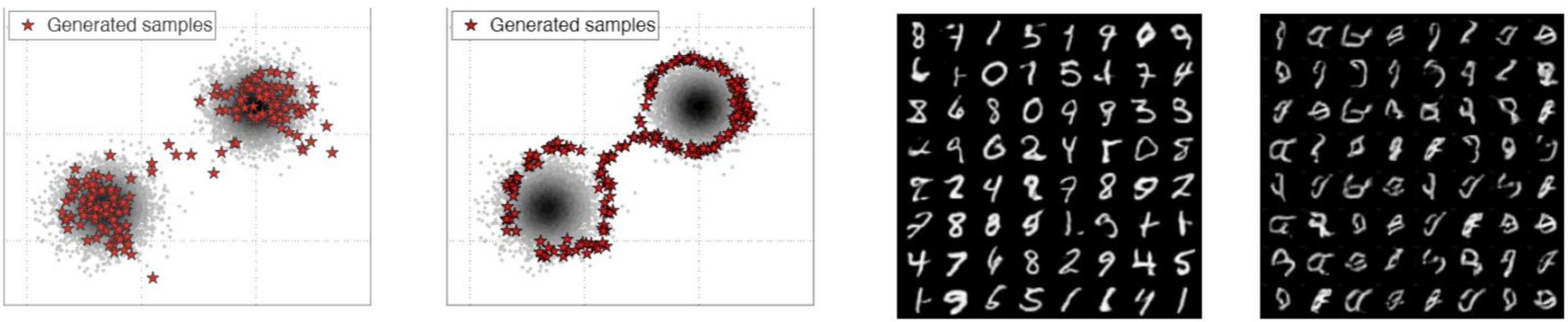
Contribution 2. GAN for Generating Out-of-Distribution Samples

- New GAN objective

$$\min_{G}\max_{D} \quad \beta \underbrace{\mathbb{E}_{P_G(\mathbf{x})}\Big[KL\left(\mathcal{U}\left(y\right) \parallel P_\theta\left(y|\mathbf{x}\right)\right)\Big]}_{\text{(a)}} + \underbrace{\mathbb{E}_{P_{\text{in}}(\mathbf{x})}\Big[\log D\left(\mathbf{x}\right)\Big] + \mathbb{E}_{P_G(\mathbf{x})}\Big[\log\left(1 - D\left(\mathbf{x}\right)\right)\Big]}_{\text{(b)}},$$

  - Term (a) forces the generator to generate low-density samples
    - (approximately) minimizing the log negative likelihood of in-distribution
  - Term (b) corresponds to the original GAN loss
    - Generating out-of-distribution samples close to in-distribution

- Experimental results on toy example and MNIST



(a) (b) (c) (d)

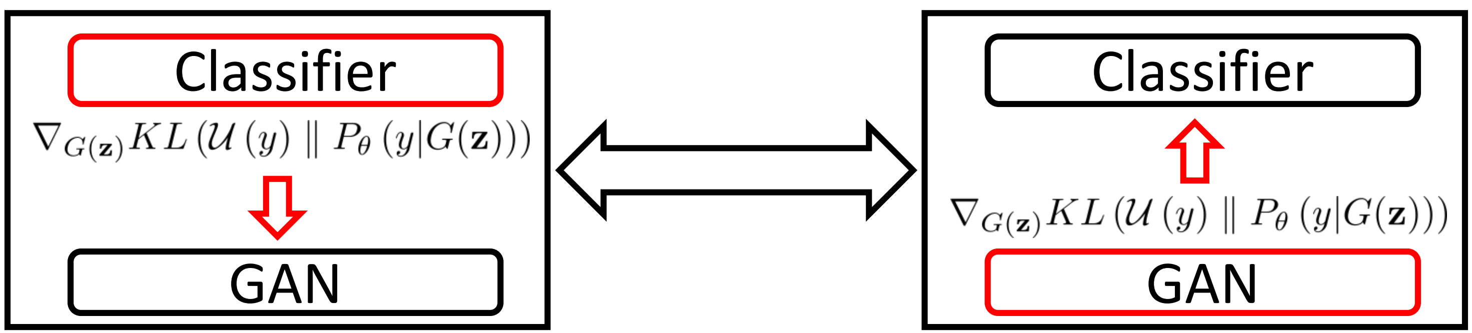
Figure 3: The generated samples from original GAN (a)/(c) and proposed GAN (b)/(d).

Contribution 3. Joint Confidence Loss

- We suggest training the proposed GAN using a confident classifier
  - Converse is also possible
- We propose a joint confidence loss

$$\min_{G}\max_{D}\min_{\theta} \quad \underbrace{\mathbb{E}_{P_{\text{in}}(\widehat{\mathbf{x}},\widehat{y})}\Big[-\log P_\theta\left(y=\widehat{y}|\widehat{\mathbf{x}}\right)\Big]}_{\text{(c)}} + \beta\, \underbrace{\mathbb{E}_{P_G(\mathbf{x})}\Big[KL\left(\mathcal{U}\left(y\right) \parallel P_\theta\left(y|\mathbf{x}\right)\right)\Big]}_{\text{(d)}}$$
$$\underbrace{+\mathbb{E}_{P_{\text{in}}(\widehat{\mathbf{x}})}\Big[\log D\left(\widehat{\mathbf{x}}\right)\Big] + \mathbb{E}_{P_G(\mathbf{x})}\Big[\log\left(1-D\left(\mathbf{x}\right)\right)\Big].}_{\text{(e)}}$$

  - Classifier's confidence loss: (c) + (d)
  - GAN loss: (d) + (e)
- Alternating algorithm for optimizing the joint confidence loss



Step 1. update GAN Step 2. update classifier

Experimental Results - Metric

- TP = true positive
- FN = false negative
- TN = true negative
- FP = false positive
- **FPR at 95% TPR**
  - FPR = FP/(FP + TN), TPR = TP/(TP + FN)
- **AUROC (Area Under the Receiver Operating Characteristic curve)**
  - ROC curve = relationship between TPR and FPR
- **Detection Error**
  - Minimum misclassification probability over all thresholds

$$\boxed{\min_{\delta} \left\{ H\left(g\left(\mathbf{x};\sigma\right) \neq 1 | z=1\right) H\left(z=1\right) + H\left(g\left(\mathbf{x};\sigma\right) \neq 0 | z=0\right) H\left(z=0\right) \right\}}$$

- **AUPR (Area under the Precision-Recall curve)**
  - PR curve = relationship between precision=TP/(TP+FP) and recall=TP/(TP+FN)

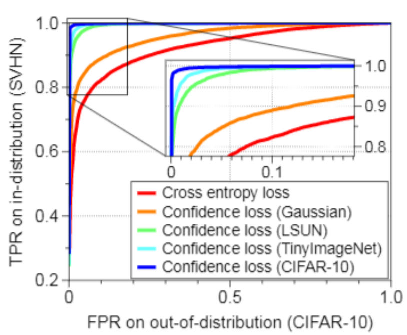
Experimental Results

- Measure the detection performance of threshold-based detectors
- Confidence loss with some explicit out-of-distribution dataset

In-dist	Out-of-dist	Classification accuracy	TNR at TPR 95%	AUROC	Detection accuracy	AUPR in	AUPR out
		Cross entropy loss / Confidence loss					
SVHN	CIFAR-10 (seen)	93.82 / **94.23**	47.4 / **99.9**	62.6 / **99.9**	78.6 / **99.9**	71.6 / **99.9**	91.2 / **99.4**
	TinyImageNet (unseen)		49.0 / **100.0**	64.6 / **100.0**	79.6 / **100.0**	72.7 / **100.0**	91.6 / **99.4**
	LSUN (unseen)		46.3 / **100.0**	61.8 / **100.0**	78.2 / **100.0**	71.1 / **100.0**	90.8 / **99.4**
	Gaussian (unseen)		56.1 / **100.0**	72.0 / **100.0**	83.4 / **100.0**	77.2 / **100.0**	92.8 / **99.4**
CIFAR-10	SVHN (seen)	80.14 / **80.56**	13.7 / **99.8**	46.6 / **99.9**	66.6 / **99.8**	61.4 / **99.9**	73.5 / **99.8**
	TinyImageNet (unseen)		**13.6** / 9.9	**39.6** / 31.8	**62.6** / 58.6	**58.3** / 55.3	**71.0** / 66.1
	LSUN (unseen)		**14.0** / 10.5	**40.7** / 34.8	**63.2** / 60.2	**58.7** / 56.4	**71.5** / 68.0
	Gaussian (unseen)		2.8 / **3.3**	10.2 / **14.1**	50.0 / 50.0	48.1 / **49.4**	39.9 / **47.0**

Table 1: Performance of the baseline detector (Hendrycks & Gimpel, 2016) using VGGNet. All values are percentages and boldface values indicate relative the better results. For each in-distribution, we minimize the KL divergence term in (1) using training samples from an out-of-distribution dataset denoted by "seen", where other "unseen" out-of-distributions were only used for testing.

- Classifier trained by our method drastically improves the detection performance across all out-of-distributions

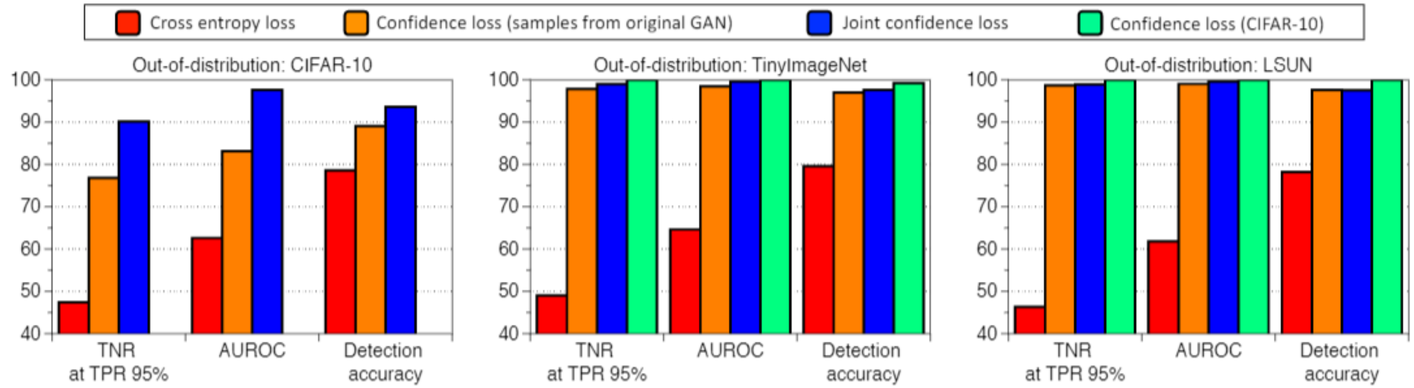


(c) ROC curve

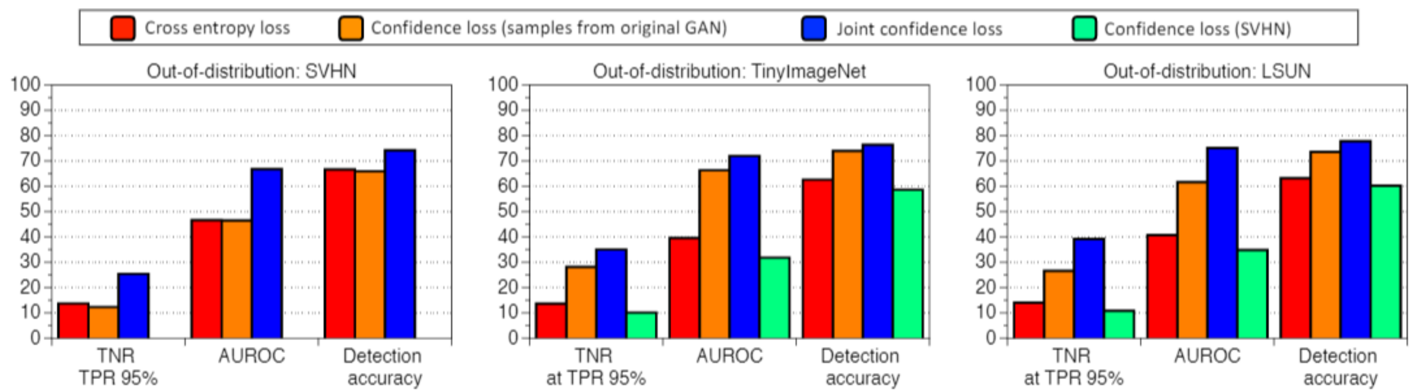
⇨ Realistic images such as TinyImageNet (aqua line) and LSUN(green line) are more useful than synthetic datasets (orange line) for improving the detection perfor-mance

Experimental Results

- Joint confidence loss



(a) In-distribution: SVHN

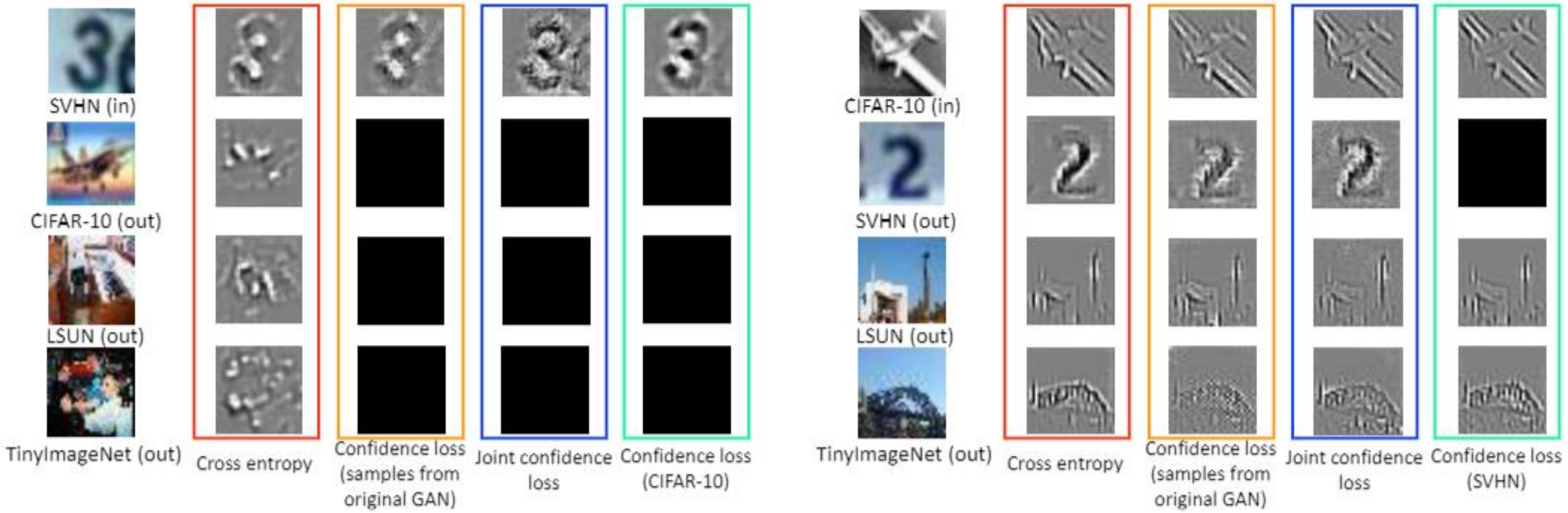


(b) In-distribution: CIFAR-10

- Confidence loss with the original GAN (orange bar) is often useful for improving the detection performance
- Joint confidence loss (bluebar) still outperforms all baseline it in all cases

Experimental Results

- Interpretability of trained classifier



(a) In-distribution: SVHN (b) In-distribution: CIFAR-10

Figure 5: Guided gradient (sensitivity) maps of the top-1 predicted class with respect to the input image under various training losses.

- Classifier trained by cross entropy loss shows sharp gradient maps for both samples from in- and out-of-distributions
- Classifiers trained by the confidence losses do only on samples from in-distribution.

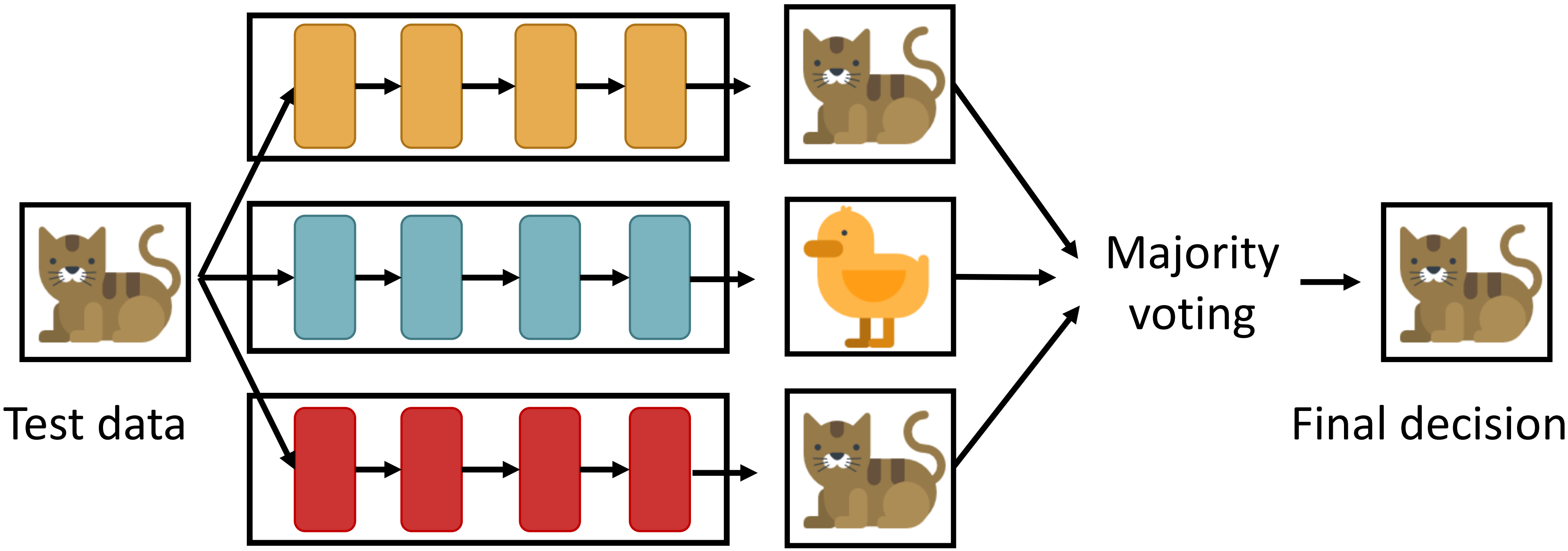
Outline

- Introduction
  - Predictive uncertainty of deep neural networks
  - Summary
- How to train confident neural networks
  - Training Confidence-Calibrated Classifiers for Detecting Out-of-Distribution Samples [Lee' 18a]
- **Applications**
  - **Confident Multiple Choice Learning [Lee' 17]**
  - Hierarchical novelty detection [Lee' 18b]
- Conclusion

[Lee' 18a] Lee, K., Lee, H., Lee, K. and Shin, J. Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples. In ICLR, 2018.
[Lee' 17] Lee, K., Hwang, C., Park, K. and Shin, J. Confident Multiple Choice Learning. *In ICML, 2017.*
[Lee' 18b] Lee, K., Lee, Min. K, Zhang, Y. Shin. J, Lee, H. Hierarchical Novelty Detection for Visual Object Recognition, In CVPR, 2018.

Application: Ensemble Learning using Deep Neural Networks

- Ensemble learning
  - Train multiple models to try and solve the same problem
  - Combine the outputs of them to obtain the final decision



- Bagging [Breiman' 96], boosting [Freund' 99] and mixture of experts [Jacobs' 91]

[Freund' 99] Freund, Yoav, Schapire, Robert, and Abe, N. A short introduction to boosting. Journal-Japanese Society For Arti- ficial Intelligence, 14(771-780):1612, 1999.
[Breiman' 96] Breiman, Leo. Bagging predictors. Machine learning, 24 (2):123–140, 1996.
[Jacobs' 91] Jacobs, Robert A, Jordan, Michael I, Nowlan, Steven J, and Hinton, Geoffrey E. Adaptive mixtures of local experts. Neural computation, 1991.

Ensemble Methods for Deep Neural Networks

- Independent Ensemble (IE) [Ciregan' 12]
  - Independently train each model with random initialization

$$L_E(\mathcal{D}) = \sum_{i=1}^{N} \sum_{m \in [M]} \ell\left(y_i, f_m(\mathbf{x}_i)\right).$$

Var	Definition
$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$	training data
$(f_1, \ldots, f_M)$	M models
$\ell(y_i, f(\mathbf{x}))$	task-specific loss

  - IE generally improves the performance by reducing the variance
- Multiple choice learning (MCL) [Guzman' 12]
  - Making each model specialized for certain subset of data

$$L_O(\mathcal{D}) = \sum_{i=1}^{N} \min_{m \in [M]} \ell\left(y_i, f_m(\mathbf{x}_i)\right),$$

  - MCL can produce diverse solutions
- Image classification on CIFAR-10 using 5 CNNs

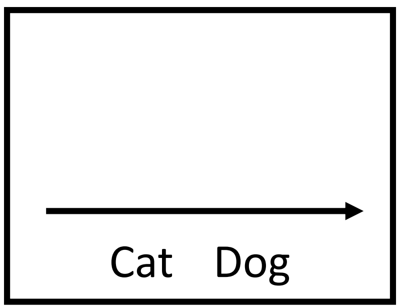
Ensemble Method	Ensemble Size $M = 5$	
	Oracle Error Rate	Top-1 Error Rate
IE	10.65%	15.34%
MCL	4.40%	60.40%

Ensemble Methods for Deep Neural Networks

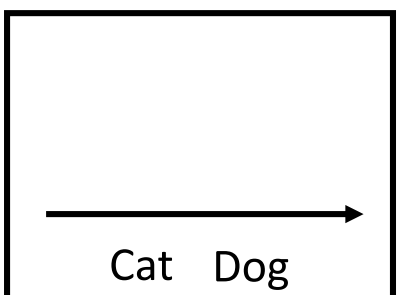
- Multiple choice learning (MCL) [Guzman' 12]
  - Making each model specialized for certain subset of data

$$L_O(\mathcal{D}) = \sum_{i=1}^{N} \min_{m \in [M]} \ell\left(y_i, f_m\left(\mathbf{x}_i\right)\right),$$

  - Overconfidence issues of MCL



Model 1 (specialized in "Cat" image)



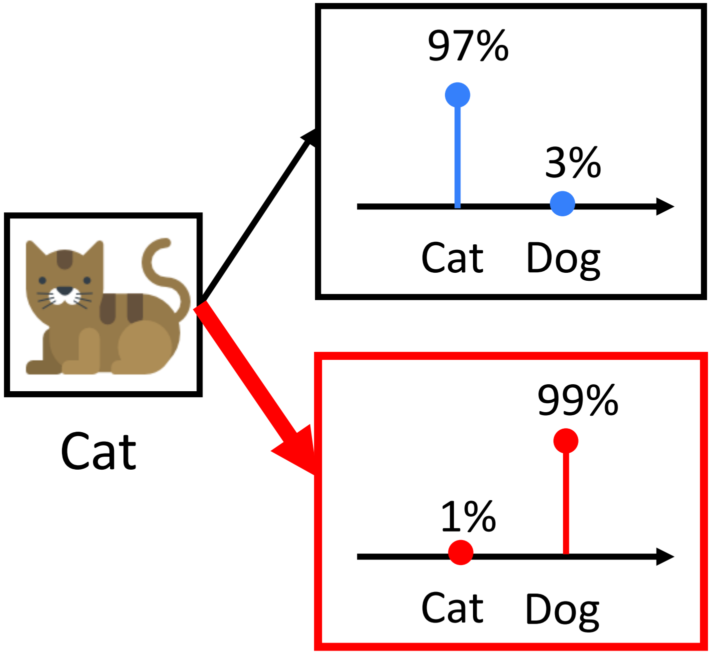
Model 2 (specialized in "Dog" image)

Ensemble Methods for Deep Neural Networks

- Multiple choice learning (MCL) [Guzman' 12]
  - Making each model specialized for certain subset of data

$$L_O(\mathcal{D}) = \sum_{i=1}^{N} \min_{m \in [M]} \ell\left(y_i, f_m\left(\mathbf{x}_i\right)\right),$$

  - Overconfidence issues of MCL



Model 1 (specialized in "Cat" image)

Overconfident

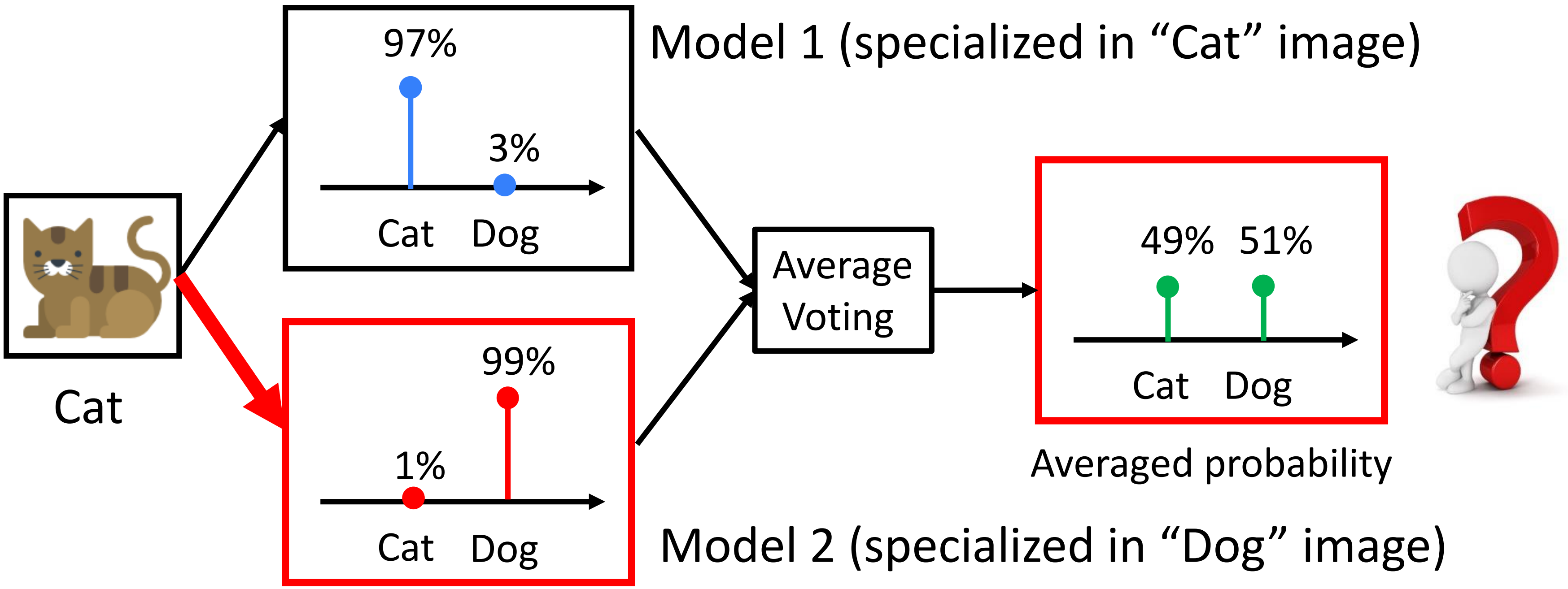
Model 2 (specialized in "Dog" image)

Ensemble Methods for Deep Neural Networks

- Multiple choice learning (MCL) [Guzman' 12]
  - Making each model specialized for certain subset of data

$$L_O(\mathcal{D}) = \sum_{i=1}^{N} \min_{m \in [M]} \ell\left(y_i, f_m\left(\mathbf{x}_i\right)\right),$$

  - Overconfidence issues of MCL



Confident Multiple Choice Learning (CMCL)

- Making the specialized models with confident predictions
- Main components of our contributions

New loss: confident oracle loss

New architecture: feature sharing

New training method: random labeling

- Experiments on CIFAR-10 using 5 CNNs (2 Conv + 2 FC)

Ensemble Method	Feature Sharing	Stochastic Labeling	Oracle Error Rate	Top-1 Error Rate
IE	-	-	10.65%	15.34%
MCL	-	-	4.40%	60.40%
CMCL	-	-	4.49%	15.65%
	✓	-	5.12%	14.83%
	✓	✓	**3.32%**	**14.78%**

Confident Oracle Loss

- Confident oracle loss

$$L_C(\mathcal{D}) = \min_{v_i^m} \sum_{i=1}^{N} \sum_{m=1}^{M} \left(v_i^m \ell \left(y_i, P_{\theta_m} \left(y_i \mid \mathbf{x}_i \right) \right) + \beta \left(1 - v_i^m \right) D_{KL} \left(\mathcal{U} \left(y \right) \parallel P_{\theta_m} \left(y \mid \mathbf{x}_i \right) \right) \right) \tag{1a}$$

$$\text{subject to} \quad \sum_{m=1}^{M} v_i^m = 1, \quad \forall i, \tag{1b}$$

$$v_i^m \in \{0, 1\}, \quad \forall i, m \tag{1c}$$

- Generating confident predictions by minimizing the KL divergence

D_{KL}: the KullbackLeibler (KL) divergence
$\mathcal{U}(y)$: the uniform distribution
v_i^m: a flag variable to decide the assignment of $\mathbf{x}_i$ to the m-th model
β: a penalty parameter
θ_m : model parameters
$P_{\theta_m}(y \mid \mathbf{x})$: Predictive distribution of m-th model

Confident Oracle Loss

- Confident oracle loss

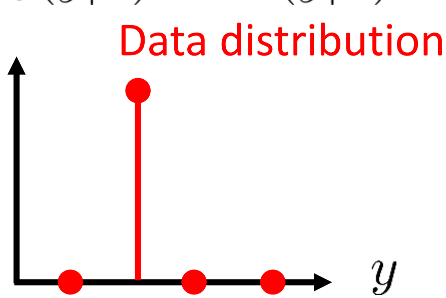
$$L_C(\mathcal{D}) = \min_{v_i^m} \sum_{i=1}^{N} \sum_{m=1}^{M} \left(\boxed{v_i^m \ell\left(y_i, P_{\theta_m}\left(y_i \mid \mathbf{x}_i\right)\right)} + \beta\left(1 - v_i^m\right) D_{KL}\left(\mathcal{U}\left(y\right) \parallel P_{\theta_m}\left(y \mid \mathbf{x}_i\right)\right) \right) \tag{1a}$$

$$\text{subject to} \quad \sum_{m=1}^{M} v_i^m = 1, \quad \forall i, \tag{1b}$$

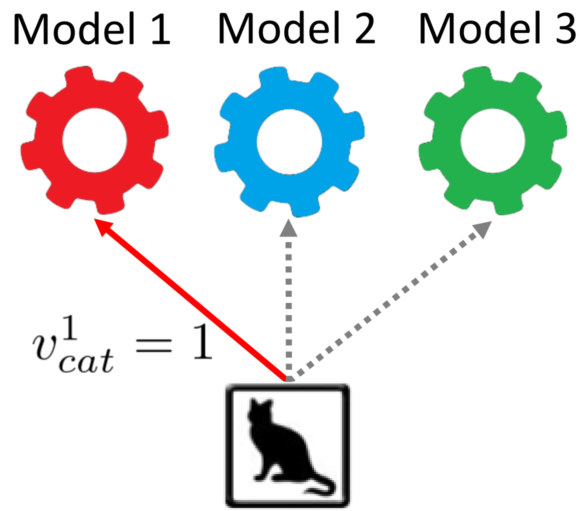
$$v_i^m \in \{0, 1\}, \quad \forall i, m \tag{1c}$$

  - Generating confident predictions by minimizing the KL divergence

$P_\theta(y|\mathbf{x}) \rightarrow P(y|\mathbf{x})$



$[\text{Target data } (v_i = 1)]$



Confident Oracle Loss

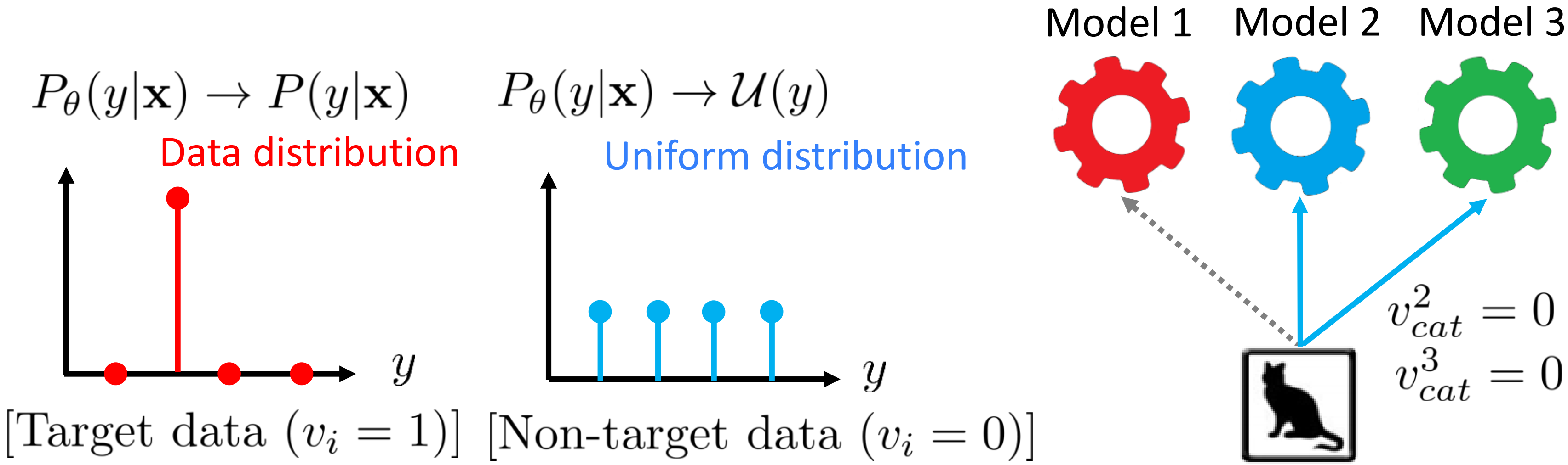
- Confident oracle loss

$$L_C(\mathcal{D}) = \min_{v_i^m} \sum_{i=1}^{N} \sum_{m=1}^{M} \left(v_i^m \ell\left(y_i, P_{\theta_m}\left(y_i \mid \mathbf{x}_i\right)\right) + \boxed{\beta\left(1 - v_i^m\right) D_{KL}\left(\mathcal{U}\left(y\right) \parallel P_{\theta_m}\left(y \mid \mathbf{x}_i\right)\right)} \right) \tag{1a}$$

$$\text{subject to} \quad \sum_{m=1}^{M} v_i^m = 1, \quad \forall i, \tag{1b}$$

$$v_i^m \in \{0, 1\}, \quad \forall i, m \tag{1c}$$

  - Generating confident predictions by minimizing the KL divergence



Experimental Results: Image Classification

- Classification test set error rates on CIFAR-10 and SVHN

CIFAR-10 [Krizhevsky' 09]

- 32 × 32 RGB
- 10 classes
- 50,000 training set
- 10,000 test set

SVHN [Netzer' 11]

- 32 × 32 RGB
- 10 classes
- 73,257 training set
- 26,032 test set

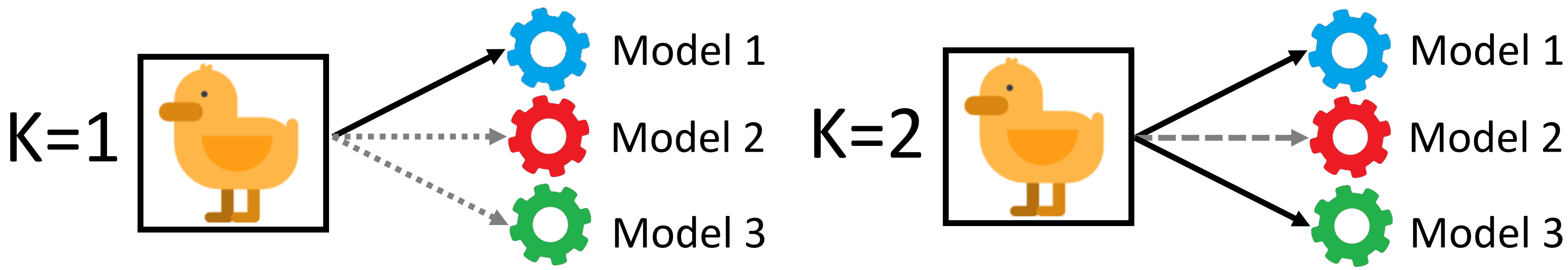
- Top-1 error
  - Select the class from averaged probability
- Oracle error
  - Measuring whether none of the members predict the correct class
- We use both feature sharing and random labeling for all experiments

Experimental Results: Image Classification

- Ensemble of small-scale CNNs (2 Conv + 2 FC)

Ensemble Method	K	Ensemble Size $M = 5$		Ensemble Size $M = 10$	
		Oracle Error Rate	Top-1 Error Rate	Oracle Error Rate	Top-1 Error Rate
IE	-	10.65%	15.34%	9.26%	15.34%
MCL	1	4.40%	60.40%	**0.00%**	76.88%
	2	3.75%	20.66%	1.46%	49.31%
	3	4.73%	16.24%	1.52%	22.63%
	4	5.83%	15.65%	1.82%	17.61%
CMCL	1	**3.32%**	14.78%	1.96%	14.28%
	2	3.69%	**14.25% (-7.11%)**	1.22%	13.95%
	3	4.38%	14.38%	1.53%	14.00%
	4	5.82%	14.49%	1.73%	**13.94% (-9.13%)**

"Picking K specialized models"



Experimental Results: Image Classification

- Ensemble of small-scale CNNs (2 Conv + 2 FC)

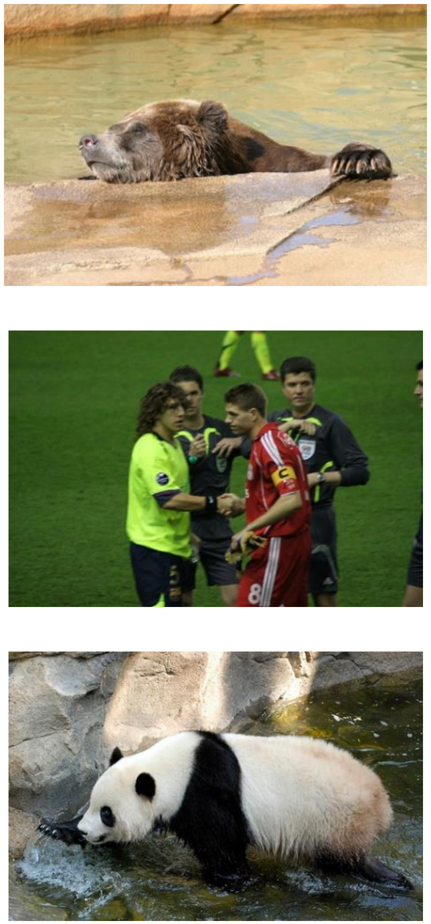
Ensemble Method	K	Ensemble Size $M=5$ Oracle Error Rate	Ensemble Size $M=5$ Top-1 Error Rate	Ensemble Size $M=10$ Oracle Error Rate	Ensemble Size $M=10$ Top-1 Error Rate
IE	-	10.65%	15.34%	9.26%	15.34%
MCL	1	4.40%	60.40%	**0.00%**	76.88%
	2	3.75%	20.66%	1.46%	49.31%
	3	4.73%	16.24%	1.52%	22.63%
	4	5.83%	15.65%	1.82%	17.61%
CMCL	1	**3.32%**	14.78%	1.96%	14.28%
	2	3.69%	**14.25% (-7.11%)**	1.22%	13.95%
	3	4.38%	14.38%	1.53%	14.00%
	4	5.82%	14.49%	1.73%	**13.94% (-9.13%)**

- Ensemble of 5 large-scale CNNs

Model Name	Ensemble Method	CIFAR-10 Oracle Error Rate	CIFAR-10 Top-1 Error Rate	SVHN Oracle Error Rate	SVHN Top-1 Error Rate
VGGNet-17	- (single)	10.65%	10.65%	5.22%	5.22%
	IE	3.27%	8.21%	1.99%	4.10%
	MCL	**2.52%**	45.58%	**1.45%**	45.30%
	CMCL	2.95%	**7.83% (-4.63%)**	1.65%	**3.92% (-4.39%)**
GoogLeNet-18	- (single)	10.15%	10.15%	4.59%	4.59%
	IE	3.37%	7.97%	1.78%	3.60%
	MCL	**2.41%**	52.03%	1.39%	37.92%
	CMCL	2.78%	**7.51% (-5.77%)**	**1.36%**	**3.44% (-4.44%)**
ResNet-20	- (single)	14.03%	14.03%	5.31%	5.31%
	IE	3.83%	10.18%	1.82%	3.94%
	MCL	**2.47%**	53.37%	1.29%	40.91%
	CMCL	2.79%	**8.75% (-14.05%)**	**1.42%**	**3.68% (-6.60%)**

Experimental Results: Image Segmentation

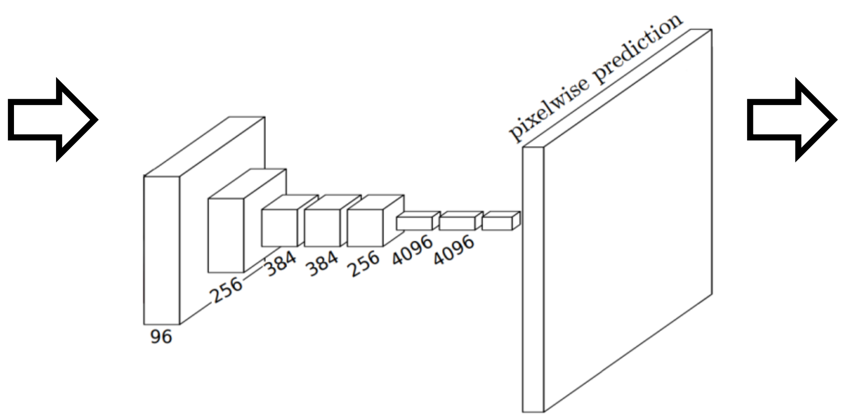
- iCoseg dataset



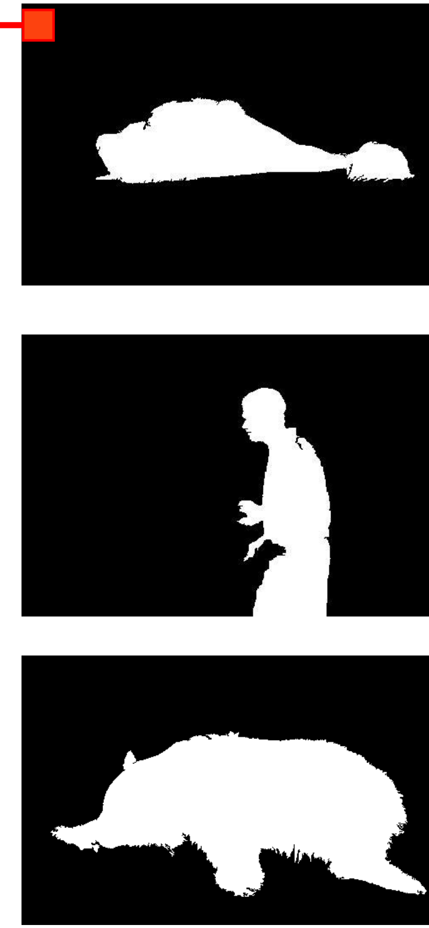
1(foreground) and 0 (background)

=

Pixel-level classification problem with 2 classes



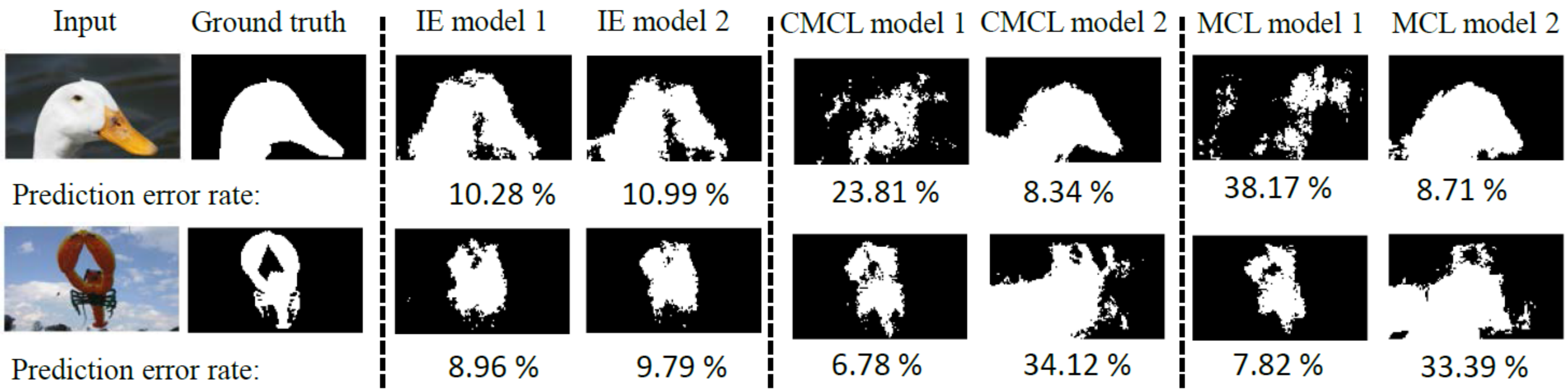
Fully convolutional neural networks (FCNs) [Long' 15]



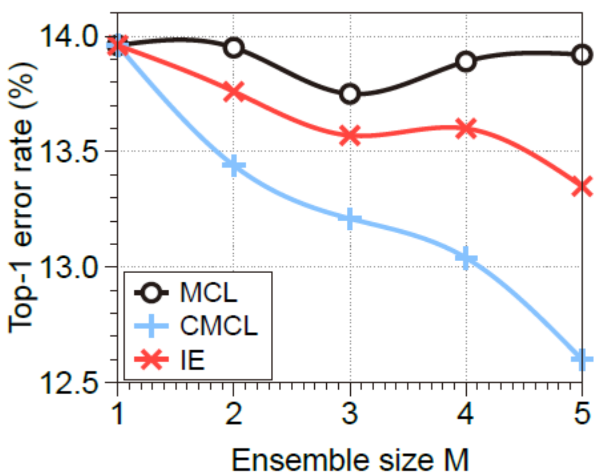
[Long' 15] Long, J., Shelhamer, E. and Darrell, T. Fully convolutional networks for semantic segmentation. In *CVPR, 2015*.

Experimental Results: Image Segmentation

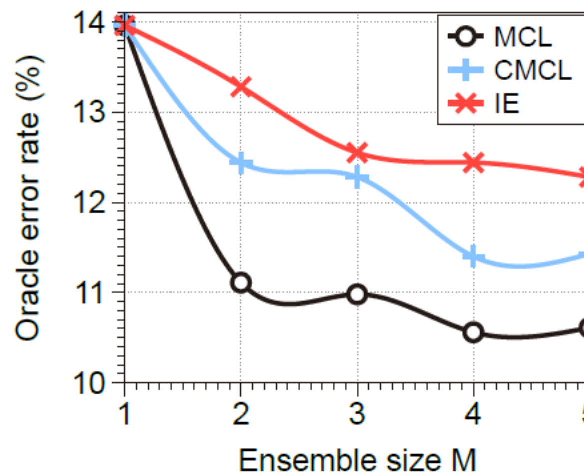
- Prediction results of segmentation for few sample images



  - MCL and CMCL generate high-quality predictions



- 6.77% relative reduction



  - CMCL only outperforms IE in terms of the top-1 error

Outline

- Introduction
  - Predictive uncertainty of deep neural networks
  - Summary
- How to train confident neural networks
  - Training Confidence-Calibrated Classifiers for Detecting Out-of-Distribution Samples [Lee’ 18a]
- Applications
  - Confident Multiple Choice Learning [Lee’ 17]
  - Hierarchical novelty detection [Lee’ 18b]
- Conclusion

[Lee’ 18a] Lee, K., Lee, H., Lee, K. and Shin, J. Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples. In ICLR, 2018.
[Lee’ 17] Lee, K., Hwang, C., Park, K. and Shin, J. Confident Multiple Choice Learning. *In ICML, 2017.*
[Lee’ 18b] Lee, K., Lee, Min. K, Zhang, Y. Shin. J, Lee, H. Hierarchical Novelty Detection for Visual Object Recognition, In CVPR, 2018.

Hierarchical Novelty Detection

- Objective
  - 1. Find the closest known (super-)category in taxonomy
  - 2. Find fine-grained classification for novel categories (i.e., out-of-distribution samples)

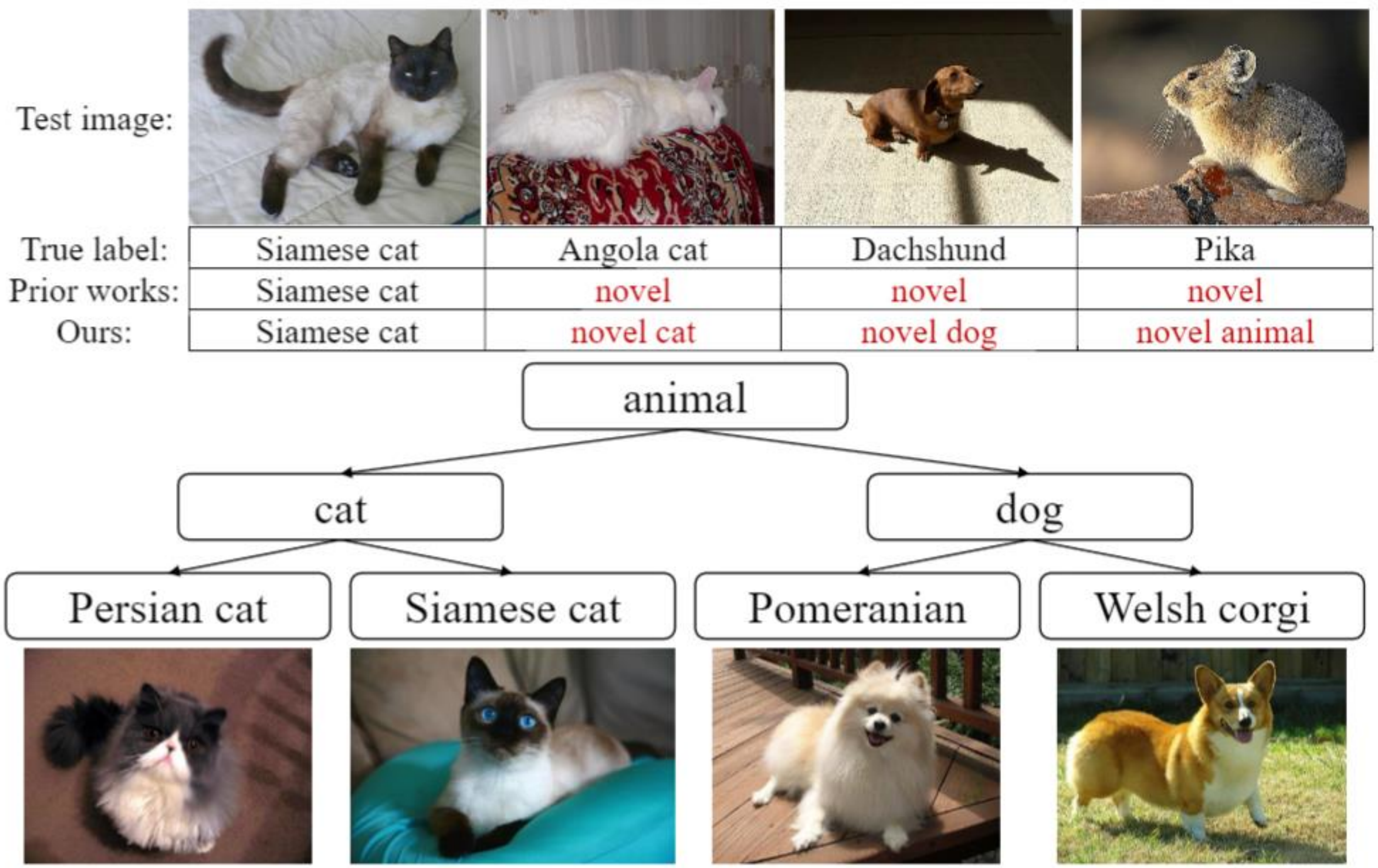
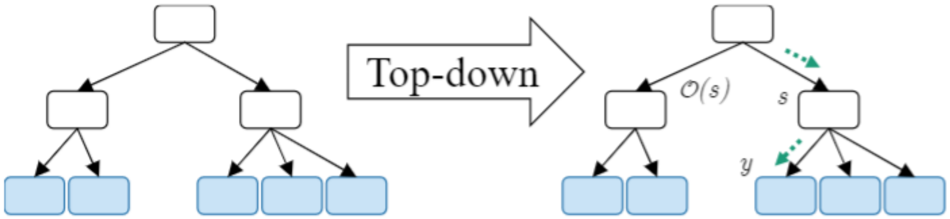


Figure 1. An illustration of our hierarchical novelty detection task

Two Main Approaches

- Top-down method (TD)
  - p(child) = $\sum_{super}$ p(child | super) p(super)



  - Objective

$$\boxed{\begin{aligned}\min_{\theta_s}\quad & \mathbb{E}_{Pr(x,y|s)}\left[-\log Pr(y|x,s;\theta_s)\right] \\ & + \mathbb{E}_{Pr(x,y|\mathcal{O}(s))}\left[D_{KL}\left(U(y|s)\parallel Pr(y|x,s;\theta_s)\right)\right],\end{aligned}}$$

$Pr(x,y|\mathcal{O}(s))$ denotes the data distribution of all exclusive classes from s

  - Inference

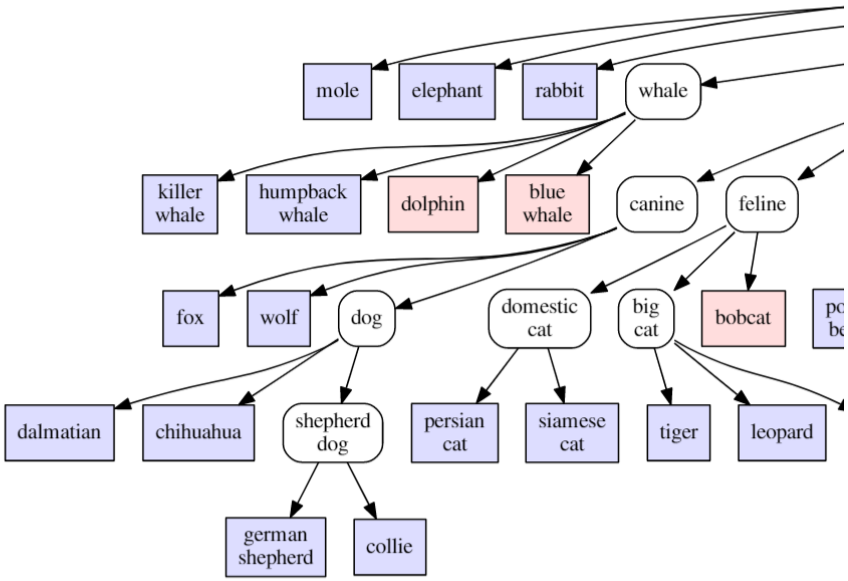
$$\hat{y} = \begin{cases} \arg\max_{y'} \; Pr(y'|x,s;\theta_s) & \text{if confident,} \\ \mathcal{N}(s) & \text{otherwise,} \end{cases}$$

Novel class

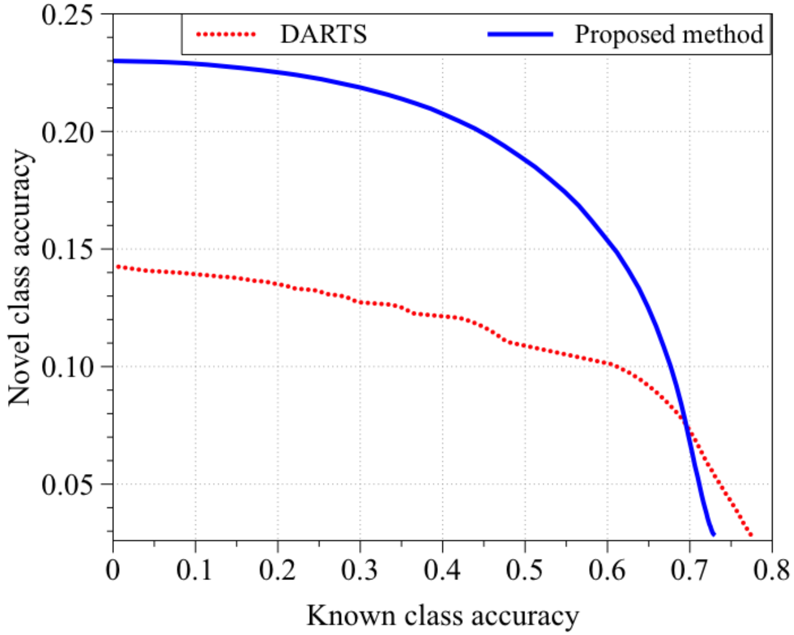
    - Definition of confidence: $D_{KL}(U(y|s) \parallel Pr(y|x,s;\theta_s)) \geq \lambda_s,$

Experimental Results on ImageNet Dataset

- ImageNet dataset
  - 22K classes
  - Taxonomy
    - 396 super classes of 1K known leaf classes
    - Rest of 21K classes can be used as novel class
  - Example



- Hierarchical novelty detection performance
  - Baseline: DARTS [Deng' 12]



- One can note that our methods have higher novel class accuracy than DARTS to have a same known class accuracy in most regions

[Deng' 12] J. Deng, J. Krause, A. C. Berg, and L. Fei-Fei. Hedging your bets: Optimizing accuracy-specificity trade offs in large scale visual recognition. In CVPR , pages 3450–3457. IEEE, 2012.

Conclusion

- We propose a new method for training confident deep neural networks
  - It produce the uniform distribution when the input is not from target distribution

- We show that it can be applied to many machine learning problems:
  - Detecting out-of-distribution problem
  - Ensemble learning using deep neural networks
  - Hierarchical novelty detection

- We believe that our new approach brings a refreshing angle for developing confident deep networks in many related applications:
  - Network calibration
  - Adversarial example detection
  - Bayesian probabilistic models
  - Semi-supervised learning