OOD Detection and Open World

Knowing what you don't know

Antonio Carta

antonio.carta@unipi.it

Today's Lecture

- **OOD Detection**:
  - Anomaly detection
  - Estimating confidence and uncertainty
- **Open World**:
  - Learning in an open world
  - How to deal with unseen data at test time
  - Knowing what you don't know

Motivation

- In practical problem, we don’t always have a training data that covers the entire test distribution
- We assume the model is unreliable for out-of-distribution (OOD) samples
- We would like the model to be able to estimate the uncertainty (of the data and its predictions) correctly

Motivation

- This is a fundamental issue for many applications
- Safety-critical domains
  - A self-driving car should (safely) stop in uncertain conditions and give control to the passenger
  - Medical systems should give uncertainty estimate so that an expert can determine the correct course of action
- Today, this is still an open issue
  - Example: ChatGPT is often extremely confident, regardless of the actual correctness of the output

Anomaly Detection / OOD Detection

Anomaly Detection

- Distinguish normal and abnormal behavior
- Examples: spam classification, intrusion detection
- Usually modeled as an highly imbalanced binary classification problem

OOD Detection

- Detect a train/test drift and reject uncertain outputs
- We don't necessarily know how the uncertain output looks like

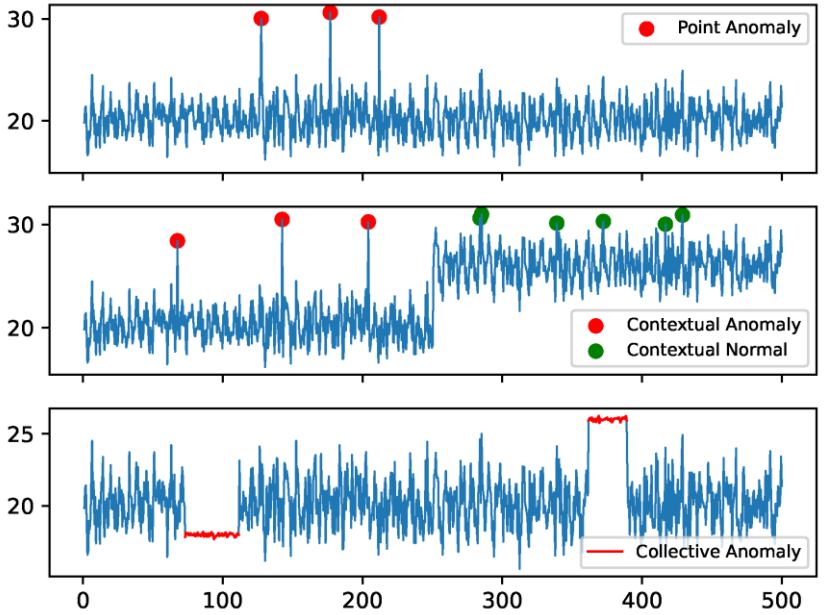


image from https://www.mdpi.com/1424-8220/23/5/2844

Open Set Recognition

- All the methods we have seen up to now assume a closed world, even when we had a train/test drift
  - In domain adaptation, we still had access to the unlabeled target domain
  - In meta-learning we know that we have a new task and we get a small training set for the new task
- What happens when the model encounters unseen classes during testing?
- What happens if the model doesn't even know how the unseen classes look like?

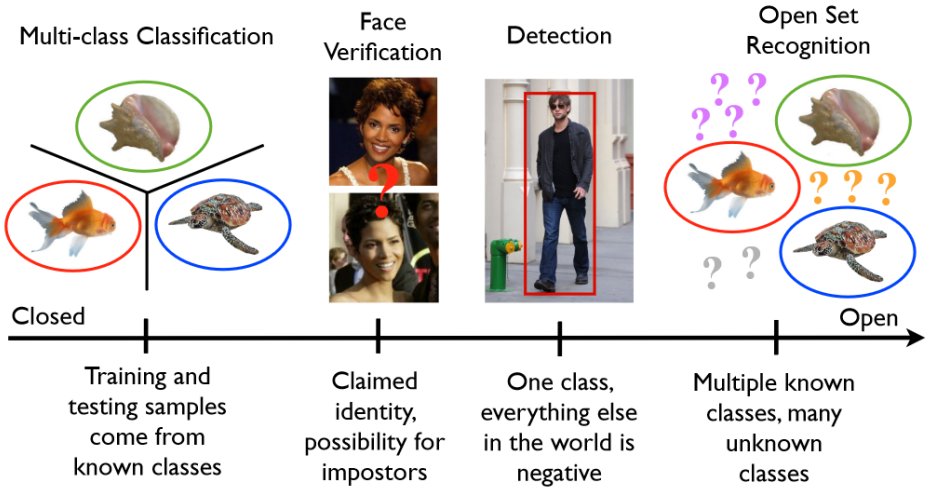


Fig. 1. Vision problems arranged in order of "openness". For some problems, we do not have knowledge of the entire set of possible classes during training, and must account for unknowns during testing. In this article, we develop a deeper understanding of those open cases.

Uncertainty

Statistical Dispersion

- In statistics, dispersion is a property of a distribution measuring how «stretched» it is
- Some measures: variance, standard deviation, IQR
- Example: in a unimodal distribution we can use the mean and dispersion to identify uncertain samples

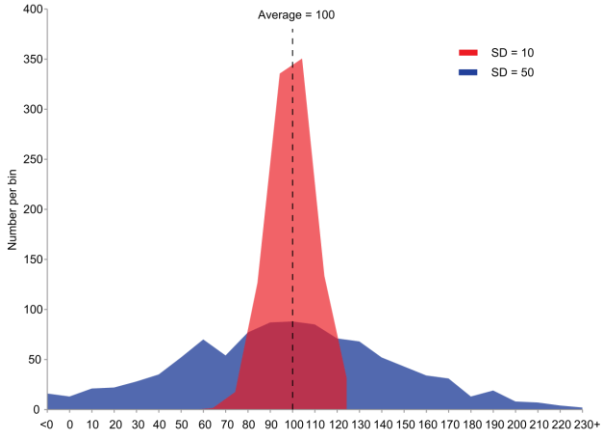


image from https://en.wikipedia.org/wiki/Statistical_dispersion

Aleatoric vs Epistemic Uncertainty

- **Aleatoric Uncertainty** of the data generating process
  - Data distribution
  - Noise in the measurements
  - Irreducible
- **Epistemic Uncertainty** of the model
  - How much the model is uncertain in its predictions
  - e.g. how much variance do we have in the output distribution
  - Can be reduced with more data

How do we detect OOD examples?

- **OBSERVATION**: Correctly classified examples tend to have greater maximum softmax probabilities than erroneously classified and out-of-distribution examples, allowing for their detection (Hendrycks, 2017)
- **SOLUTION**:
  - order data by max softmax probability
  - Use a threshold to separate ID/OOD

- Only works in very simple settings

Hendrycks, Dan, and Kevin Gimpel. 2018. “A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks.”

ODIN

- **OBSERVATION**: *temperature scaling and small perturbations to the input can separate the softmax score distributions between in- and out-of-distribution images.* Improves (Hendrycks, 2017)
- **Small perturbations** decrease the softmax score
  - If the input is in a “flat region” of the input space a small perturbation shouldn’t decrease the probability too much
  - We expect OOD samples to be more sensitive to small perturbations
- **Temperature scales** changing the softmax output distribution towards more uniform (high T) or more peaked (low T) distributions
- **ADVANTAGES**: simple method and it can be added to any pretrained supervised model

ODIN

ODIN:

- Input perturbation: $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \varepsilon \text{sign}\left(-\nabla_{\boldsymbol{x}} \log S_{\hat{y}}(\boldsymbol{x}; T)\right)$
  - Make a small step away from the high probability inputs
- Temperature scaling: $S_i(\boldsymbol{x}; T) = \frac{\exp(f_i(x)/T)}{\sum_{j=1}^{N} \exp\left(f_j(x)/T\right)}$
  - Make the distribution more or less peaked to control the True Positive Rate

S. Liang et al. 2020. "Enhancing The Reliability of Out-of-Distribution Image Detection in Neural Networks." arXiv:1706.02690

ODIN

- Softmax score: $S_{\hat{y}}(\boldsymbol{x}; T) = \max_i S_i(x; T)$
  - Select the max probability
- Detector: $g(\boldsymbol{x}; \delta, T, \varepsilon) = \begin{cases} 1 & \text{if } S_{\hat{y}}(\tilde{\boldsymbol{x}}; T) \leq \delta \\ 0 & \text{if } S_{\hat{y}}(\tilde{\boldsymbol{x}}; T) > \delta \end{cases}$
  - Use a threshold on the score to discriminate ID/OOD samples

S. Liang et al. 2020. “Enhancing The Reliability of Out-of-Distribution Image Detection in Neural Networks.” arXiv:1706.02690

ODIN

ODIN:

- Input perturbation: $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \varepsilon \text{sign}\left(-\nabla_{\boldsymbol{x}} \log S_{\hat{y}}(\boldsymbol{x}; T)\right)$
- Temperature scaling: $S_i(\boldsymbol{x}; T) = \frac{\exp(f_i(x)/T)}{\sum_{j=1}^{N} \exp\left(f_j(x)/T\right)}$
- Softmax score: $S_{\hat{y}}(\boldsymbol{x}; T) = \max_i S_i(x; T)$
- Detector: $g(\boldsymbol{x}; \delta, T, \varepsilon) = \begin{cases} 1 & \text{if } S_{\hat{y}}(\tilde{\boldsymbol{x}}; T) \leq \delta \\ 0 & \text{if } S_{\hat{y}}(\tilde{\boldsymbol{x}}; T) > \delta \end{cases}$
- T, δ selected via model selected to achieve a desired true positive rate

S. Liang et al. 2020. “Enhancing The Reliability of Out-of-Distribution Image Detection in Neural Networks.” arXiv:1706.02690

ODIN

Published as a conference paper at ICLR 2018

	Out-of-distribution dataset	FPR (95% TPR) ↓	Detection Error ↓	AUROC ↑	AUPR In ↑	AUPR Out ↑
		Baseline (Hendrycks & Gimpel, 2017) / ODIN				
Dense-BC CIFAR-10	TinyImageNet (crop)	34.7/**4.3**	10.0/**4.7**	95.3/**99.1**	96.4/**99.1**	93.8/**99.1**
	TinyImageNet (resize)	40.8/**7.5**	11.5/**6.1**	94.1/**98.5**	95.1/**98.6**	92.4/**98.5**
	LSUN (crop)	39.3/**11.4**	10.2/**7.2**	94.8/**97.9**	96.0/**98.0**	93.1/**97.9**
	LSUN (resize)	33.6/**3.8**	9.8/**4.4**	95.4/**99.2**	96.4/**99.3**	94.0/**99.2**
	Uniform	23.5/**0.0**	5.3/**0.5**	96.5/**99.0**	97.8/**100.0**	93.0/**99.0**
	Gaussian	12.3/**0.0**	4.7/**0.2**	97.5/**100.0**	98.3/**100.0**	95.9/**100.0**
Dense-BC CIFAR-100	TinyImageNet (crop)	67.8/**26.9**	36.4/**12.9**	83.0/**94.5**	85.3/**94.7**	80.8/**94.5**
	TinyImageNet (resize)	82.2/**57.0**	43.6/**22.7**	70.4/**85.5**	71.4/**86.0**	68.6/**84.8**
	LSUN (crop)	69.4/**18.6**	37.2/**9.7**	83.7/**96.6**	86.2/**96.8**	80.9/**96.5**
	LSUN (resize)	83.3/**58.0**	44.1/**22.3**	70.6/**86.0**	72.5/**87.1**	68.0/**84.8**
	Uniform	100.0/**100.0**	35.86/**17.9**	43.1/**99.5**	63.2/**87.5**	41.9/**65.1**
	Gaussian	100.0/**100.0**	41.2/**38.0**	30.6/**40.5**	53.4/**60.5**	37.6/**40.9**

Table 2: Distinguishing in- and out-of-distribution test set data for image classification. All values are percentages. ↑ indicates larger value is better, and ↓ indicates lower value is better. We use $T = 1000$ for all experiments. The noise magnitude ε was selected on a **separate validation dataset**, which is different from the out-of-distribution test sets. On CIFAR-10 pretrained model, we use $\varepsilon = 0.0014$ for all OOD test datasets; and $\varepsilon = 0.002$ for CIFAR-100 pretrained model.

S. Liang et al. 2020. "Enhancing The Reliability of Out-of-Distribution Image Detection in Neural Networks." arXiv:1706.02690

Limits of Uncertainty Estimation

- DNN are overconfident
- Supervised and generative models are overconfident
- Ideally, we would like to use bayesian models to estimate uncertainty
  - Often expensive and heavily approximated
- A simpler solution: Ensembles

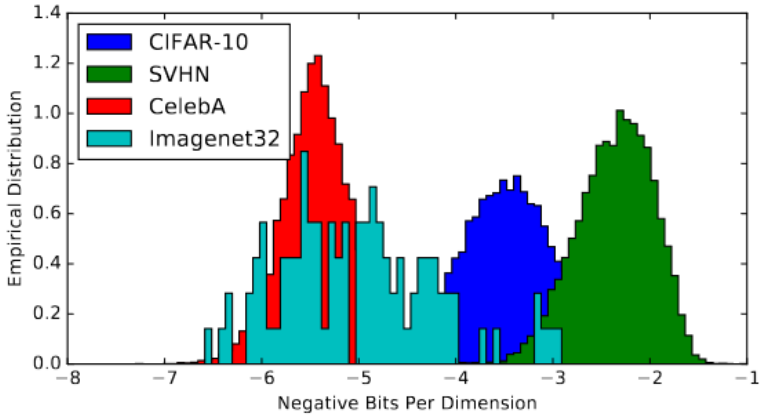
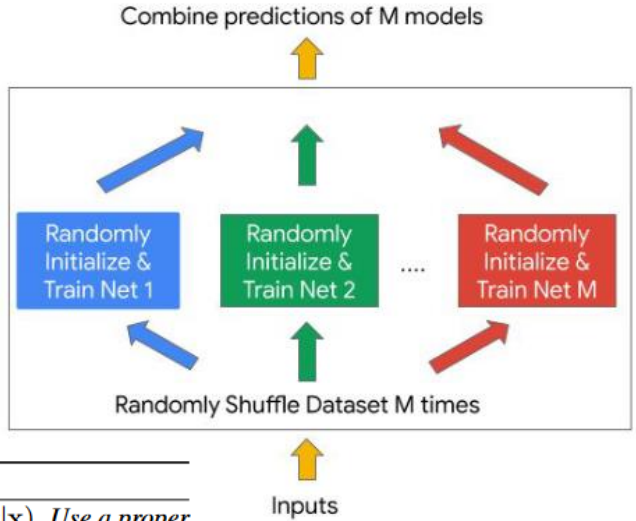


Figure 1. Density estimation models are not robust to OoD inputs. A GLOW model (Kingma & Dhariwal, 2018) trained on CIFAR-10 assigns much higher likelihoods to samples from SVHN than samples from CIFAR-10. .

H. Choi et al. 2019. "WAIC, but Why? Generative Ensembles for Robust Anomaly Detection."

Deep Ensembles

- **Deep Ensembles**
  - (1) use a proper scoring rule as the training criterion,
  - (2) use adversarial training to smooth the predictive distributions, and
  - (3) train an ensemble



Algorithm 1 Pseudocode of the training procedure for our method

1: ▷ *Let each neural network parametrize a distribution over the outputs, i.e.* $p_\theta(y|\mathbf{x})$. *Use a proper scoring rule as the training criterion* $\ell(\theta, \mathbf{x}, y)$. *Recommended default values are* $M = 5$ *and* $\epsilon = 1\%$ *of the input range of the corresponding dimension (e.g 2.55 if input range is [0,255]).*
2: Initialize $\theta_1, \theta_2, \ldots, \theta_M$ randomly
3: **for** $m = 1 : M$ **do** ▷ *train networks independently in parallel*
4: Sample data point n_m randomly for each net ▷ *single* n_m *for clarity, minibatch in practice*
5: Generate adversarial example using $\mathbf{x}'_{n_m} = \mathbf{x}_{n_m} + \epsilon \operatorname{sign}\left(\nabla_{\mathbf{x}_{n_m}} \ell(\theta_m, \mathbf{x}_{n_m}, y_{n_m})\right)$
6: Minimize $\ell(\theta_m, \mathbf{x}_{n_m}, y_{n_m}) + \ell(\theta_m, \mathbf{x}'_{n_m}, y_{n_m})$ w.r.t. θ_m ▷ *adversarial training (optional)*

Method: B. Lakshminarayanan. 2017. “Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles.”
image from https://www.gatsby.ucl.ac.uk/~balaji/balaji-uncertainty-talk-cifar-dlrl.pdf

Uncertainty with Ensembles

Empirical observations:

- **post-hoc calibration** often fails
- **marginalize over models** (i.e. ensembles!) give surprisingly strong results across a broad spectrum of tasks.

Y. Ovadia et al. 2019. “Can You Trust Your Model’s Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift.”

Results

Expected Calibration Error (ECE) ↓ Measures the correspondence between predicted probabilities and empirical accuracy

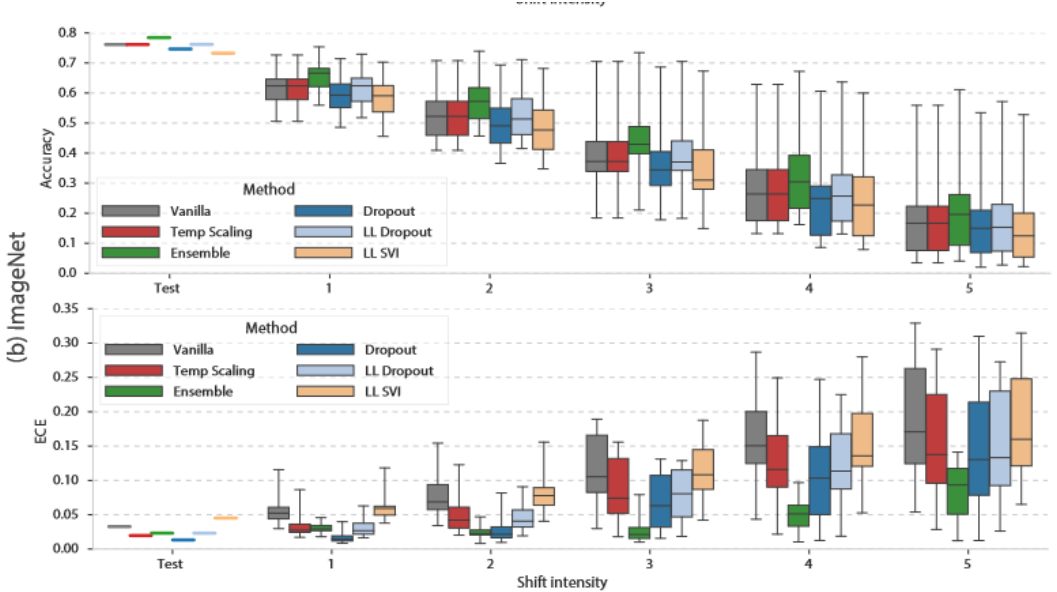


Figure 2: Calibration under distributional shift: a detailed comparison of accuracy and ECE under all types of corruptions on (a) CIFAR-10 and (b) ImageNet. For each method we show the mean on the test set and summarize the results on each intensity of shift with a box plot. Each box shows the quartiles summarizing the results across all (16) types of shift while the error bars indicate the min and max across different shift types. Figures showing additional metrics are provided in Figures S4 (CIFAR-10) and S5 (ImageNet). Tables for numerical comparisons are provided in Appendix G.

Y. Ovadia et al. 2019. "Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift."

Summing up

- We need to track aleatoric uncertainty (data) and epistemic uncertainty (model)
- DNN are overconfident and their confidence estimates cannot be trusted
- Post-hoc calibration can mitigate the issue but only in simple settings
- Ensembles show much more consistent results
  - At the price of increased computational cost (both training and inference)

Open World

Knowing in an Open World

- **Known**: in-distribution samples and predicted correctly with high confidence (correct predictions)
- **Known Unknowns**: low-confidence samples, such as successfully recognized anomalies (low confidence)
- **Unknown Unknowns**: out-of-distribution samples with highly confident predictions (the model shouldn't have high confidence here)
  - In general, if the test distribution drifts, it may become something completely new. Effectively, the model doesn't know what it doesn't know.
- **PROBLEM**: How do we identify unknown unknowns?

Open World Assumption

- **Closed world assumption**: the model «knows what it needs to know» such as which classes are present in the data distribution
- **Open world assumption**: the model may encounter new data at test time
  - Example: unknown classes
  - We don’t expect the model to generalize to unseen classes
  - Therefore, we would like to model to be able to recognize what it doesn’t know
- Did we see already some examples of open world assumptions?

Open World Assumption

- Did we see already some examples of open-world assumptions? NO.

Examples:

- **Continual learning**: closed world. First, we train on new experiences, then, we evaluate on them.
  - We expect the model to be incrementally adaptable
  - We don't expect the model to recognize unknown unknowns
- **Meta-learning**: closed world. we meta-train and then we meta-test on new tasks
  - But again, the model trains on every task
  - We expect the model to generalize (possibly few-shot) to novel tasks
  - We don't expect it to recognize unknown unknowns

OWA in Formal Logic

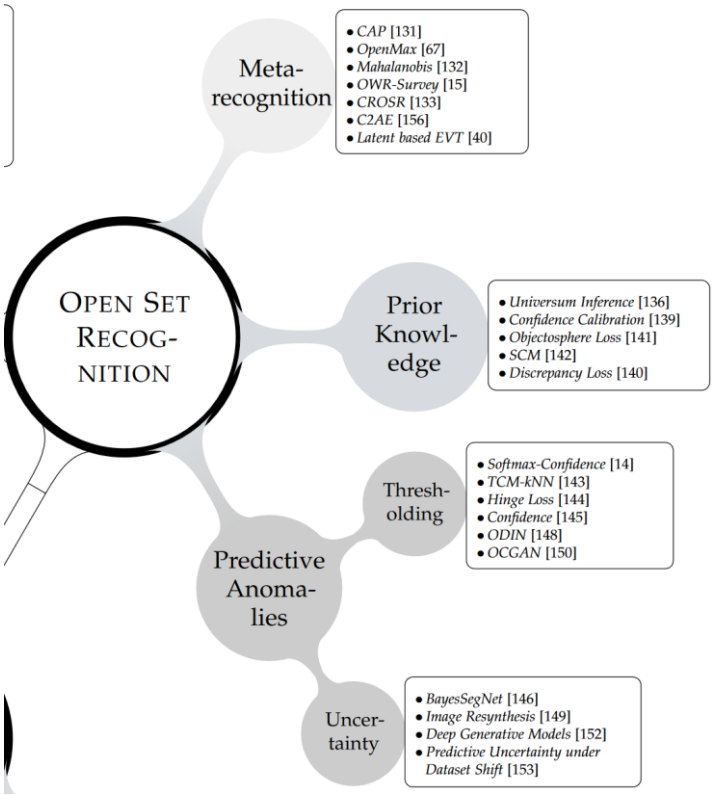
An example, in formal logic systems in classical AI:

- **Closed-world** means that any true statement is known to be true
- **Open-world** means that a statement may be true but yet unknown
  - The logic needs to be adapted to deal with unknown truth values

Probably the first example of open world assumption in AI and something you may have seen in previous courses (AIF)

Approaches

- **Anomalies**: detect out-of-distribution samples via anomaly detection
- **Prior Knowledge**: train the model to recognize a «background» class
- **Open Set Recognition**: model the space of «things you know» and never predict outside of it



Prior Knowledge

Background Class

Sometimes, we have access to a large set of «unknown examples»

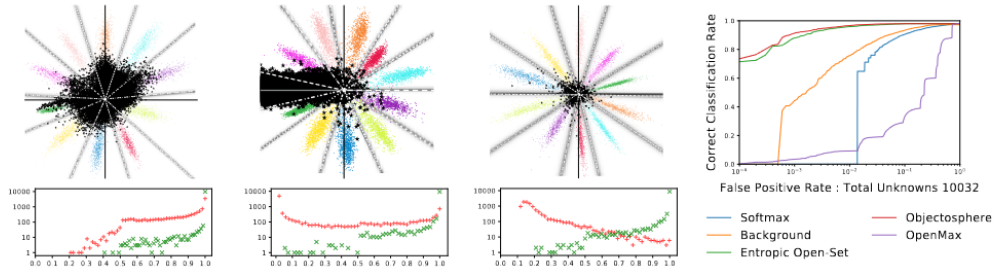
- Example: let's say we have a supervised problem
- We have a small subset of classes we are interested in / have already labeled
- The rest are «background classes», and they represent what the model doesn't know
- We can use them for training as an additional explicit «background class»
- Convert an open world problem into a closed world one

Objectosphere

- Train with a background class
- **OBSERVATION**: magnitudes of features for unknown samples in deep feature space are often lower than those of known samples.
- **Objectosphere loss** explicitly optimize this objective
  - *known samples* should have a magnitude above a specified minimum
  - *background samples* should have magnitude of the features close to zero

A. R. Dhamija et al. 2018. “Reducing Network Agnostophobia.” NIPS

Objectosphere



(a) Softmax (b) Background (c) Objectosphere (d) Open-Set Recognition Curve

Figure 1: LeNet++ Responses To Knowns and Unknowns. *The network in (a) was only trained to classify the 10 MNIST classes ($\mathcal{D}'_c$) using softmax, while the networks in (b) and (c) added NIST letters [13] as known unknowns ($\mathcal{D}'_b$) trained with softmax or our novel Objectosphere loss. In the feature representation plots on top, colored dots represent test samples from the ten MNIST classes ($\mathcal{D}_c$), while black dots represent samples from the Devanagari[28] dataset ($\mathcal{D}_a$), and the dashed gray-white lines indicate class borders where softmax scores for neighboring classes are equal. This paper addresses how to improve recognition by reducing the overlap of network features from known samples $\mathcal{D}_c$ with features from unknown samples $\mathcal{D}_u$. The figures in the bottom are histograms of softmax probability values for samples of $\mathcal{D}_c$ and $\mathcal{D}_a$ with a logarithmic vertical axis. For known samples $\mathcal{D}_c$, the probability of the correct class is used, while for samples of $\mathcal{D}_a$ the maximum probability of any known class is displayed. In an application, a score threshold θ should be chosen to optimally separate unknown from known samples. Unfortunately, such a threshold is difficult to find for either (a) or (b), a better separation is achievable with the Objectosphere loss (c). The proposed Open-Set Classification Rate (OSCR) curve in (d) depicts the high accuracy of our approach even at a low false positive rate.*

A. R. Dhamija et al. 2018. "Reducing Network Agnostophobia." NIPS

Limitations

- We can train the model to recognize the unknown
  - The open world problem becomes a supervised learning problem
- But only because it is known (we have the background class)
- What if we don’t have background classes?

Open Set Recognition

Open Set Recognition

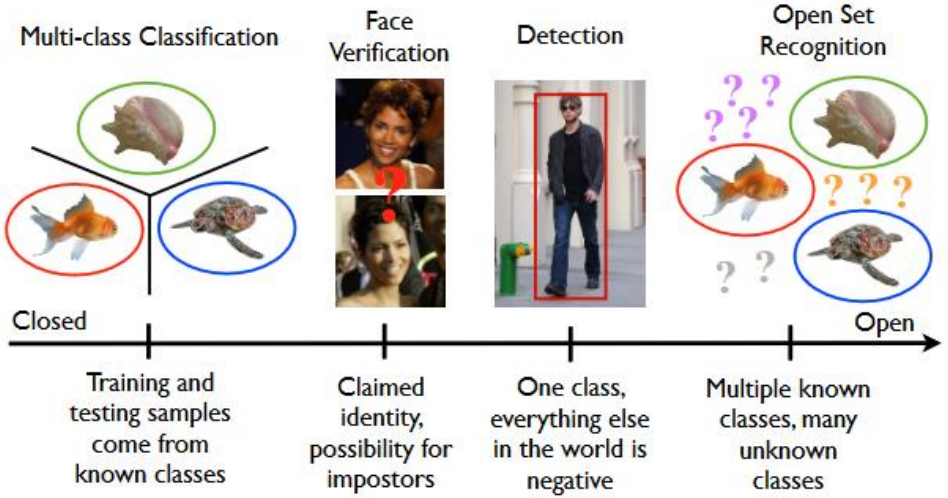


Fig. 1. Vision problems arranged in order of "openness". For some problems, we do not have knowledge of the entire set of possible classes during training, and must account for unknowns during testing. In this article, we develop a deeper understanding of those open cases.

W. J. Scheirer et al. "Towards Open Set Recognition." TPAMI, 2012.

Unbounded Decision Boundaries

- All the classification models that we used have unbounded decision boundaries
- Example: a binary linear model
  - Splits the space into two regions
  - One side is positive, the other negative
  - **Regardless of how far they are from the boundary or the training data distribution**
- In an open world setting, we need to allow a reject option

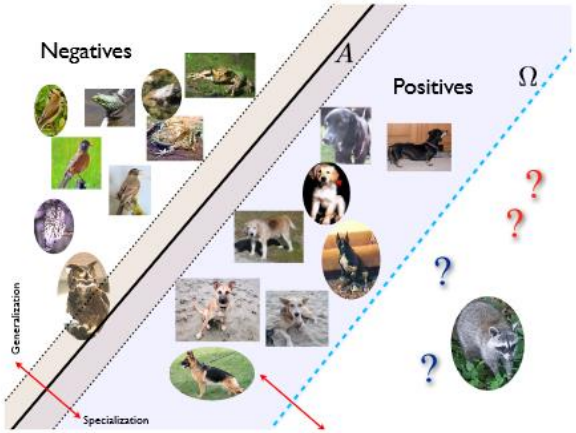


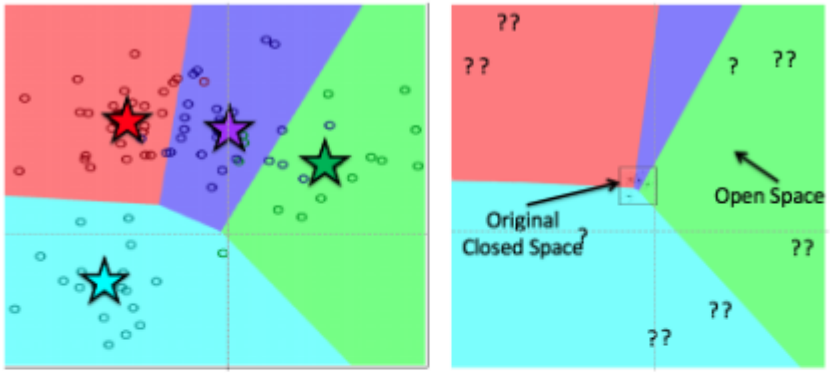
Fig. 2. The Open Set Recognition Problem explicitly assumes not all classes are known *a priori*. Square images are from training, oval images are from testing. The class of interest ("dog") is surrounded by other classes, which can be known ("frog", "birds"), or unknown ("owl","raccoon", "?"). Plane A maximizes the SVM margin making "dog" a half-space – which is mostly open space. The 1-vs-Set machine adds a second plane Ω and defines an optimization to adjust A and Ω to balance empirical and open space risk.

W. J. Scheirer et al. "Towards Open Set Recognition." TPAMI, 2012.

Unbounded Decision Boundaries

Example:

- (left) decision boundary close to training examples
- (right) zoomed out, the decision boundaries discriminate OOD examples (often with high confidence!)



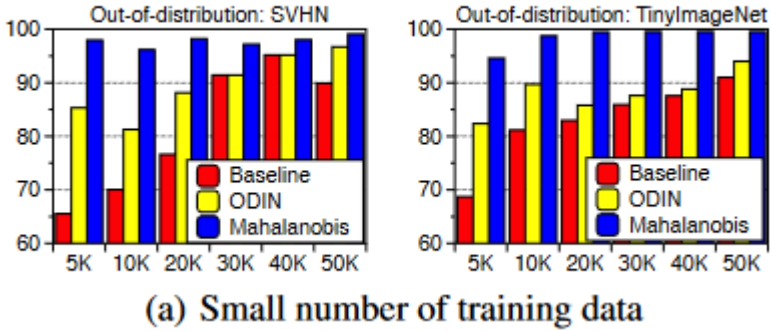
(a) Example four-class model from closed set point of view. (b) Zooming out to show some open space.

Figure 1: The issue of open space can be seen by zooming out from around the training data. Open space is the region far from training samples. A traditional classifier, e.g., NCM shown here, will label everything including the unknown "?" inputs. Even points infinitely far away are labeled.

T. E. Boult et al. 2019. "Learning and the Unknown: Surveying Steps toward Open World Recognition." AAAI.

Distance-based Rejection

- Reject samples that are too far
- We can change the softmax with
  A distance-based classifier
  And threshold the maximum distance



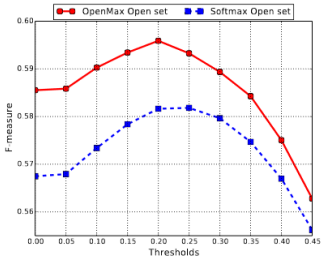
(a) Small number of training data

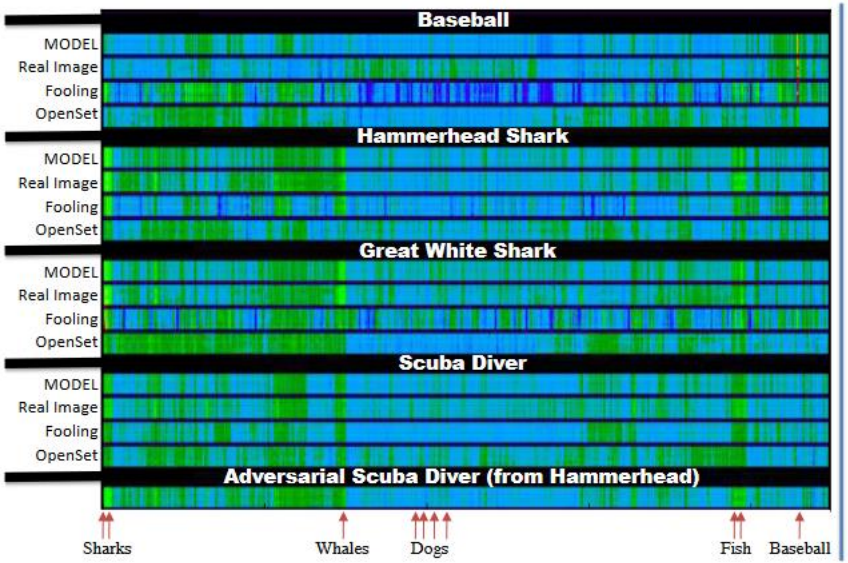
- **Mahalanobis Distance**
  - $M(\mathrm{x}) = \max_c -(f(\mathrm{x}) - \hat{\mu}_c)^\top \hat{\mathbf{\Sigma}}^{-1}(f(\mathrm{x}) - \hat{\mu}_c)$
  - $\hat{\mu}_c$ class mean
  - $\hat{\mathbf{\Sigma}}$ covariance matrix
  - **IDEA**: how many standard deviations of distance between x and the class mean

K. Lee et al. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks.” NIPS18

OpenMax – OSR with Deep Networks

- **IDEA**: activation vectors can be used to perform open-set recognition
- Recognition of unknown inputs





Take-Home Messages

- Many real-world problem are much more «open» than the toy examples we study and use to train our models
- **Known unknowns** can be accounted for during training (background classes) to improve the rejection rate of unknown objects
- **Unknown unknowns** require rethinking the training algorithm to account for the risk of confident predictions on unknown data
- Still a largely unsolved problem for DNN

References

- Papers in the footnotes
- Open World Lifelong Learning
  A Continual Machine Learning Course
  - https://owll-lab.com/teaching/cl_lecture/
  - Recordings are available
  - The organization of these slides is partly based on this course