life.augmented

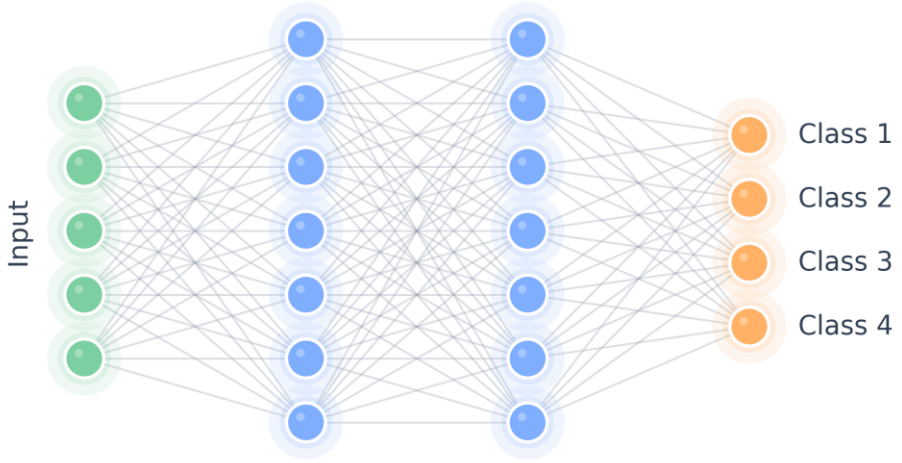
Uncertainty and Out of Distribution Detection in Neural Networks

Angelo Bosco
STMicroelectronics - Artificial Intelligence Software & Tools Group
Seminar at Catania University - May 22, 2026

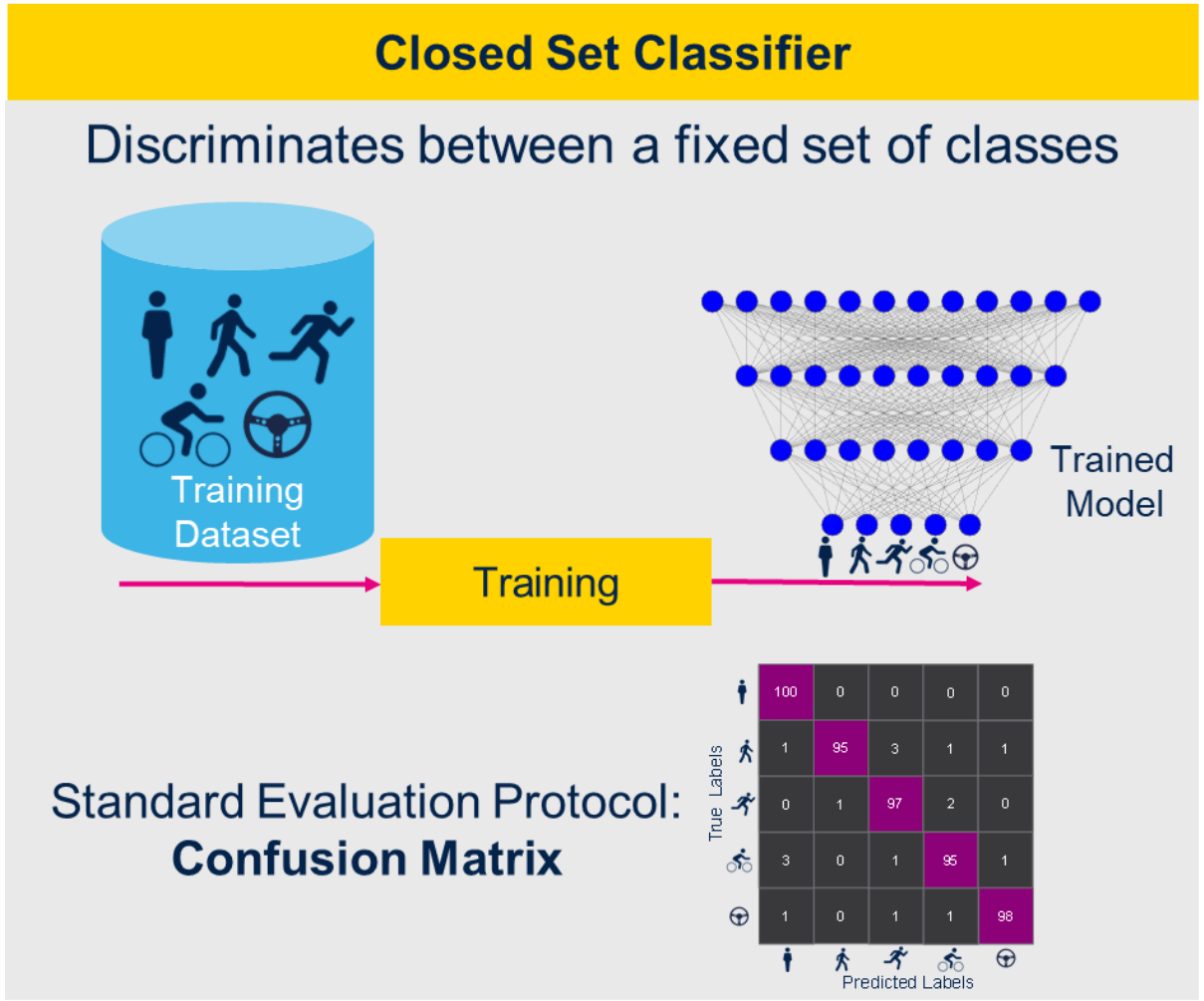
Closed Set Classifiers and OOD Data

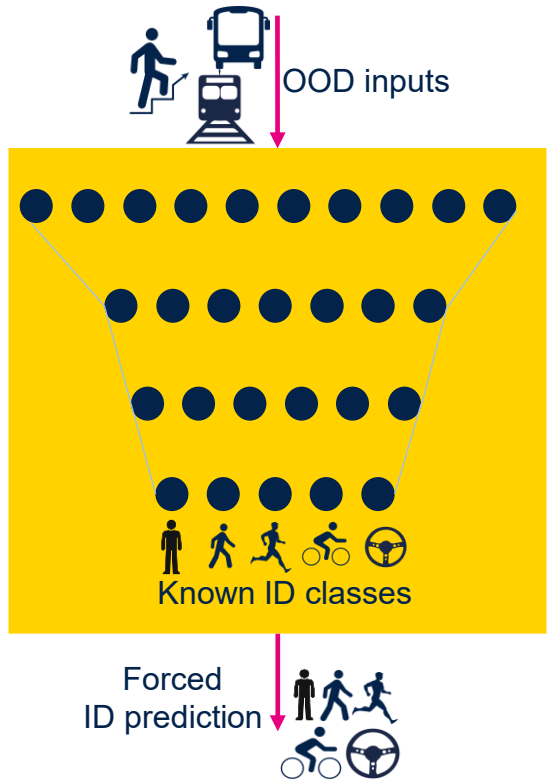
Closed Set Classification

- Standard classifiers are usually trained on **in-distribution (ID)** data and predict among a **fixed set of known classes**.
- This is a **closed-set** setting: all test inputs are implicitly assumed to belong to one of the known classes.



Closed Set Classifiers and OOD Data





- At deployment, the model may receive inputs that do not match the training distribution. These are called **Out-Of-Distribution (OOD)** inputs
- E.g.: **novel defects** in visual inspection images, **noisy/drifted** sensor **data**, etc.

life.augmented

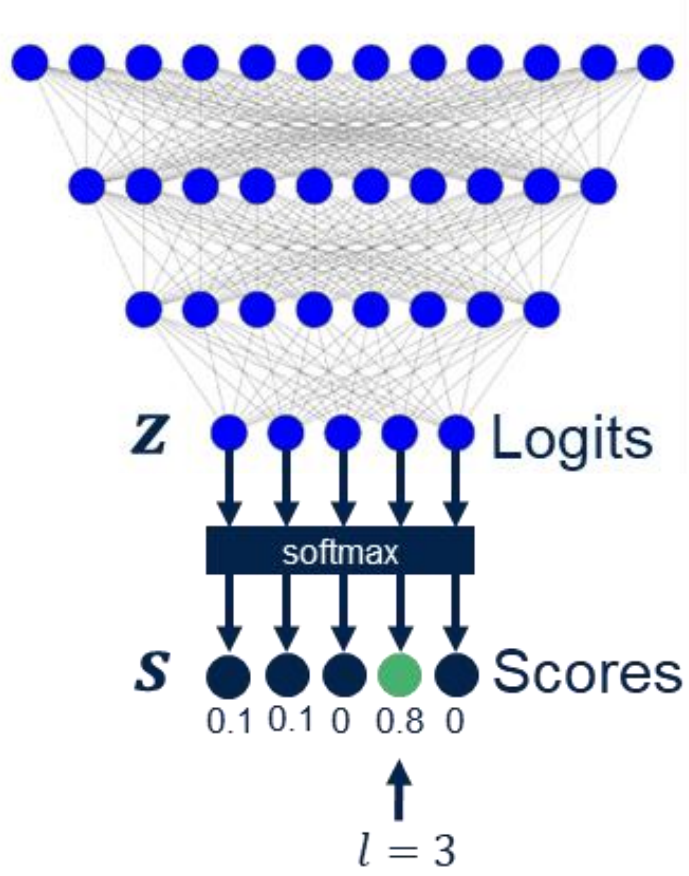
Closed Set Setting vs Real World

- **Training and test data** are assumed to come from the **same distribution**
- The classifier is optimized for **accuracy on known ID classes**
- However, **after deployment**, these assumptions often break

Closed-Set Assumption	Real-world Deployment
All classes are «known»	Unknown classes may appear
The environment is stable	Distribution shifts is common
ID Accuracy is the main objective	Reliability and uncertainty matter

OOD detection addresses the gap between closed-world assumptions and open-world deployment.

Softmax ≠ calibrated confidence



$$softmax_T(z_i) = s_i = \frac{e^{z_i/T}}{\sum_{i=1}^{N} e^{z_i/T}}$$

Softmax output sums to 1.

$$l = \underset{i}{\operatorname{argmax}}\, s_i$$

$$confidence_{bl} = \max(\boldsymbol{s})$$

Ideally:
"High" score → High **confidence**
"Low" score → **Uncertainty**.

- $\max(\boldsymbol{s})$ is treated as baseline confidence score, but it is uncalibrated probability.
- It is not the true likelihood of correctness.

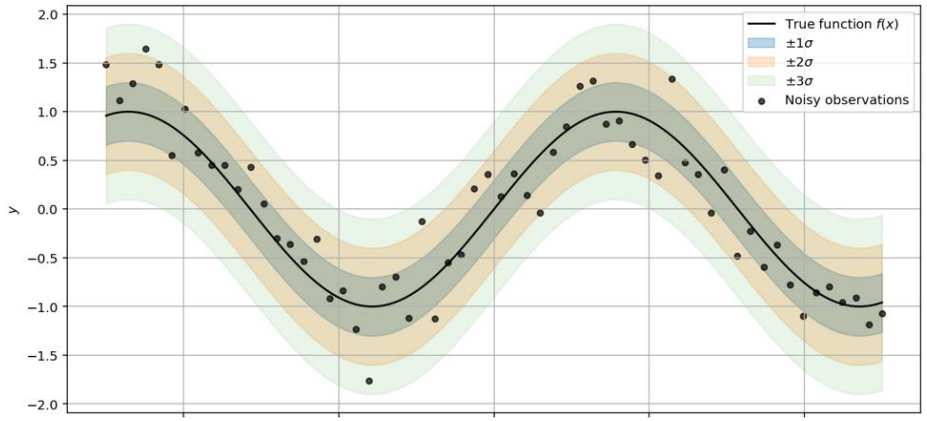
life.augmented

Aleatoric and Epistemic Uncertainty

Aleatoric and Epistemic Uncertainty

There are two main types of uncertainty:

- Aleatoric → noise in the data
- Epistemic → lack of knowledge



OOD inputs are mostly related to high epistemic uncertainty

Sources of Aleatoric Uncertainty

Measurement / Acquisition Variability

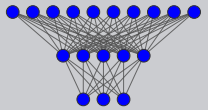
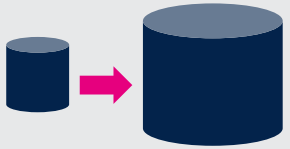
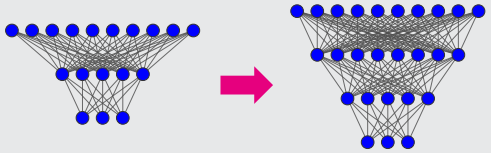
- **Sensor Limitations**:
  - resolution, quantization, thermal noise,…
- **Partially reducible** (better hardware, preprocessing)

Physical Process Variability

- **Intrinsic variability** in the physical phenomenon (turbulence in fluid flow, **photon shot noise** in images, **thermal fluctuations**, …)
- Cannot be removed, only modeled more accurately.

Aleatoric Uncertainty → variability in observations, not lack of knowledge

Aleatoric vs Epistemic Uncertainty

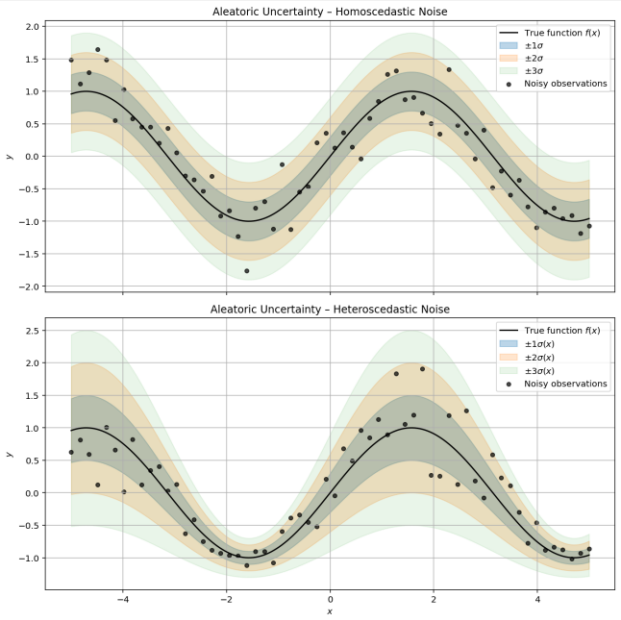
	Aleatoric Uncertainty	Epistemic Uncertainty
Root Cause	**Intrinsic Randomness** in the data generation process (e.g. sensor noise, stochastic processes)	 Data scarcity  *weak* model
Can be reduced?	Usually No. Not with more data of the same kind.	  Often **yes**, by collecting more data, or using a better-suited model.
Relevance to OOD	May be present even for in-distribution inputs	It increases because of unseen classes, anomalies, shift, or extrapolation regime

life.augmented

Aleatoric Uncertainty in Regression and Classification

Regression

Aleatoric uncertainty comes from **noise in the observations**



Homoscedastic: constant noise level

Heteroscedastic: input-dependent noise level

Example: in imaging, some noise sources are signal-dependent

Classification

Aleatoric uncertainty comes from **ambiguous or degraded inputs:**

Blur, low resolution, occlusion,…

It can be present even for **in-distribution** samples

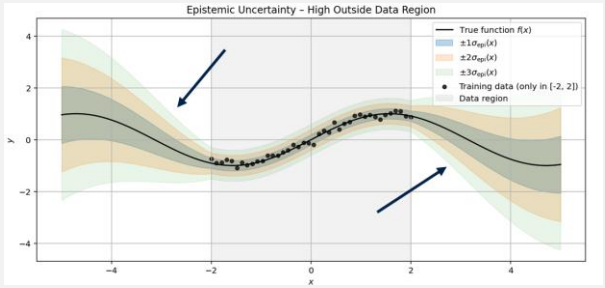
life.augmented

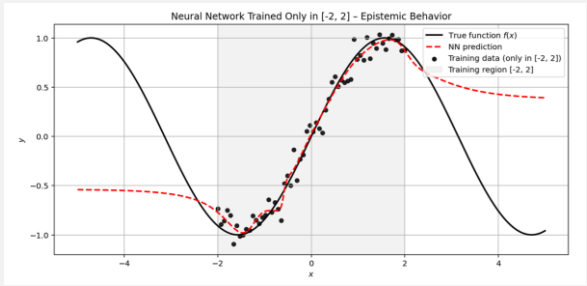
Aleatoric uncertainty comes from the data itself: the input is hard, noisy, or ambiguous.

Epistemic Uncertainty in Regression and Classification

Regression

- Epistemic uncertainty is high where **training coverage is poor**





- In regions with few or no observations, It can often be reduced with more data, a better model

Classification

- Epistemic uncertainty comes from limited model knowledge

Examples:

- **rarely** sampled patterns
- **unbalanced** datasets
- **unseen** classes
- inputs **far** from the training distribution

This is the uncertainty type most closely linked to **OOD detection**

Hard ID Samples : Covariate Shift

Covariate Shift:

- Sample distribution $\boldsymbol{x}$ *shifts* between training and testing.
- Label dependency $y \mid \boldsymbol{x}$ remains unchanged.
- often considered a **shifted ID**

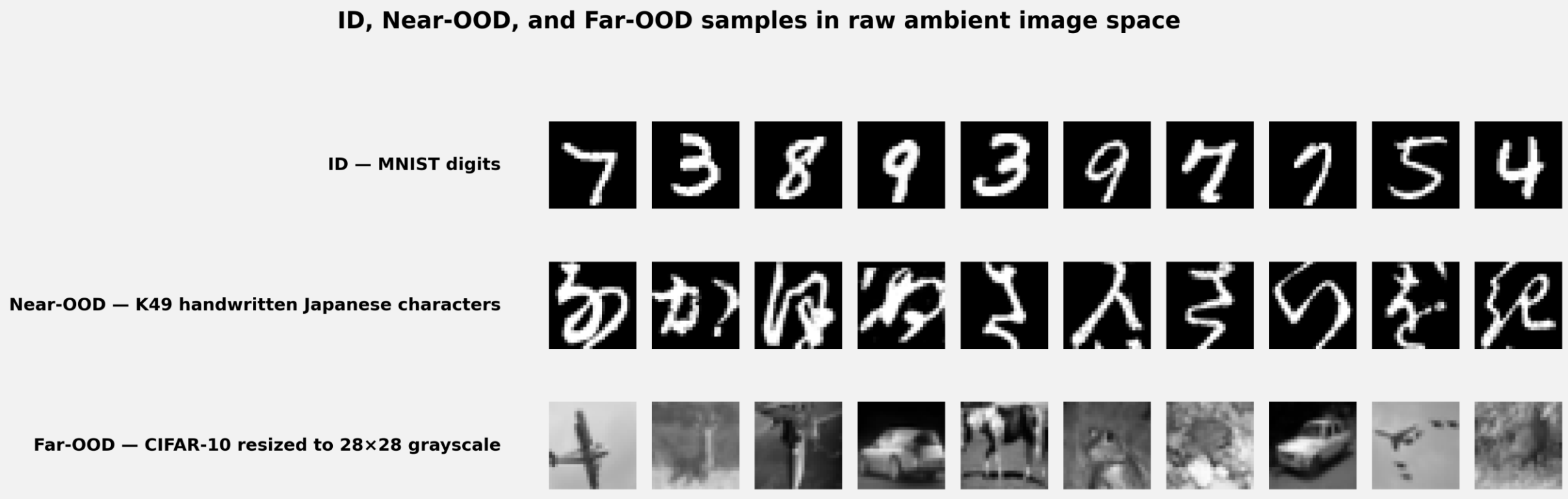
Informally: same task, different appearance/conditions

A robust classifier should be able to classify the drifted samples as well.

Common mitigation: data augmentation during training

Near-OOD, Far-OOD & Overconfidence

Near-OOD and Far-OOD

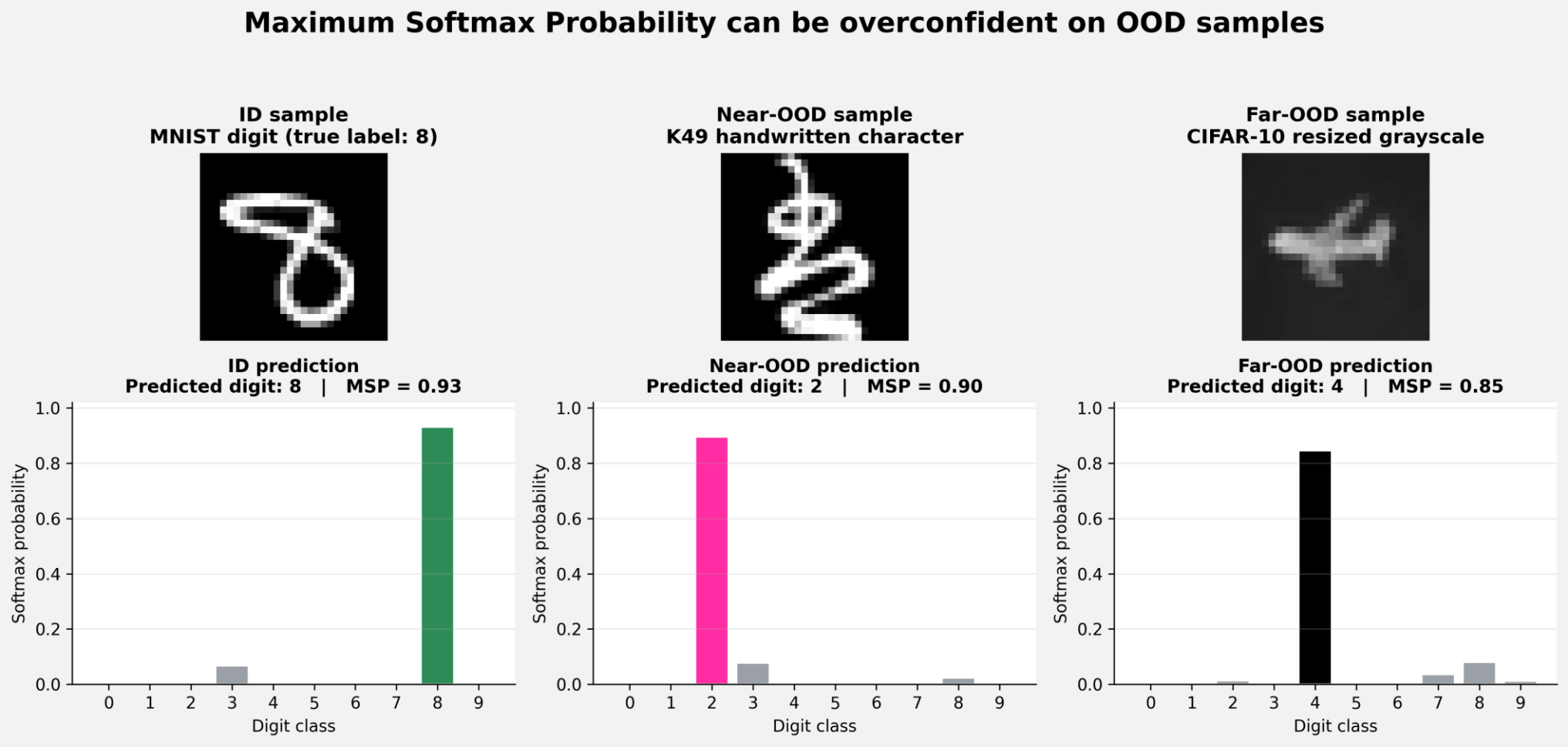


- The classifier is trained only on MNIST digits and is never exposed during training to K49, CIFAR-10, or other OOD samples.

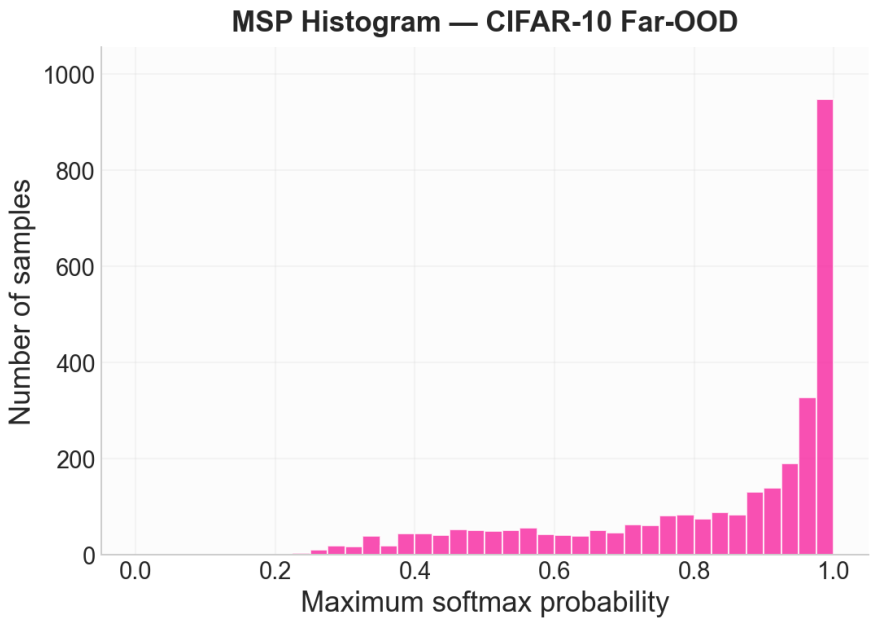
- When K49, CIFAR-10, or even pure noise images are presented at test time, the classifier assigns them to one of the 10 known digit classes.

life.augmented

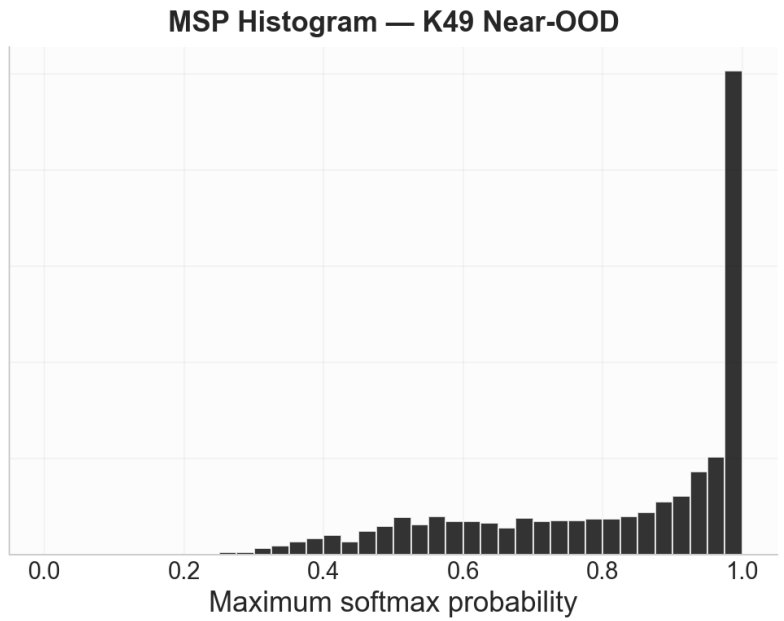
Overconfidence on OOD Data



Maximum Softmax Probability on OOD Datasets



Far-OOD: CIFAR-10 images evaluated by an MNIST classifier



Near-OOD: K49 images evaluated by an MNIST classifier

- Both histograms are **skewed** towards 1.0 (**confidence**): most **OOD** inputs are receiving **strong confidence scores**.

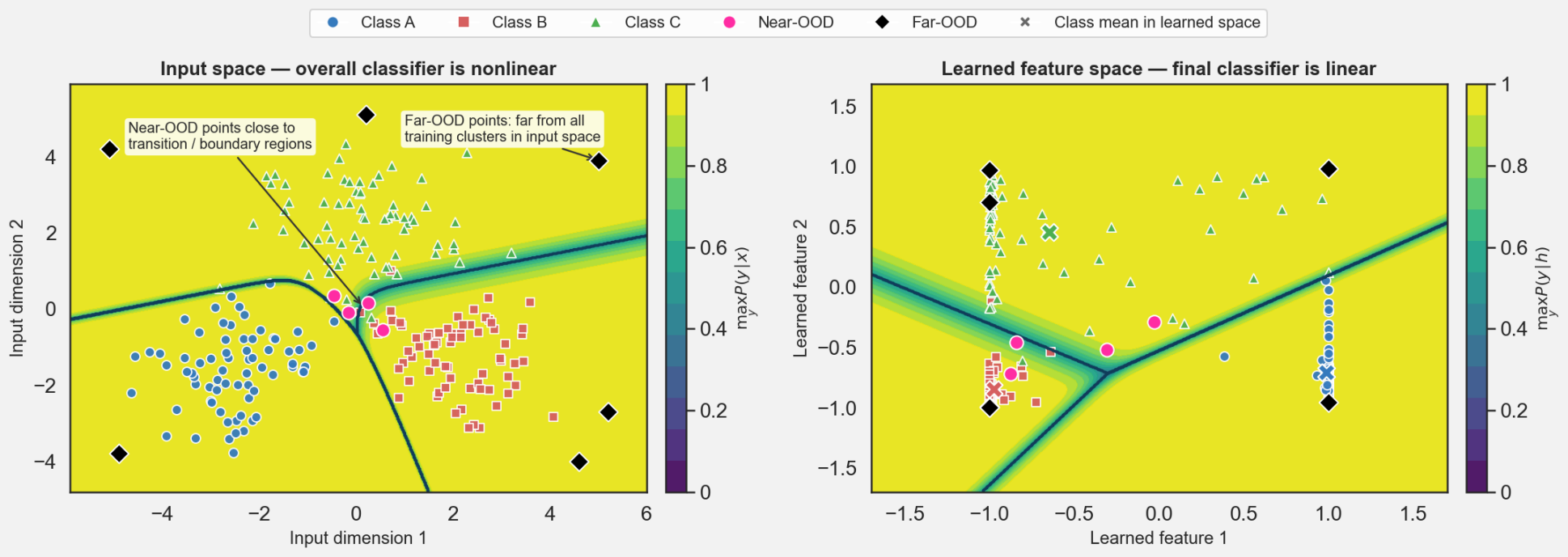
- MSP basically answers "**which known class wins**?", not "**is this input *familiar*?**"
- We need more robust solutions.

life.augmented

Discriminative and Generative Models

Discriminative Models

Neural Classifiers are **discriminative models**: they learn decision boundaries that separate samples from different classes.

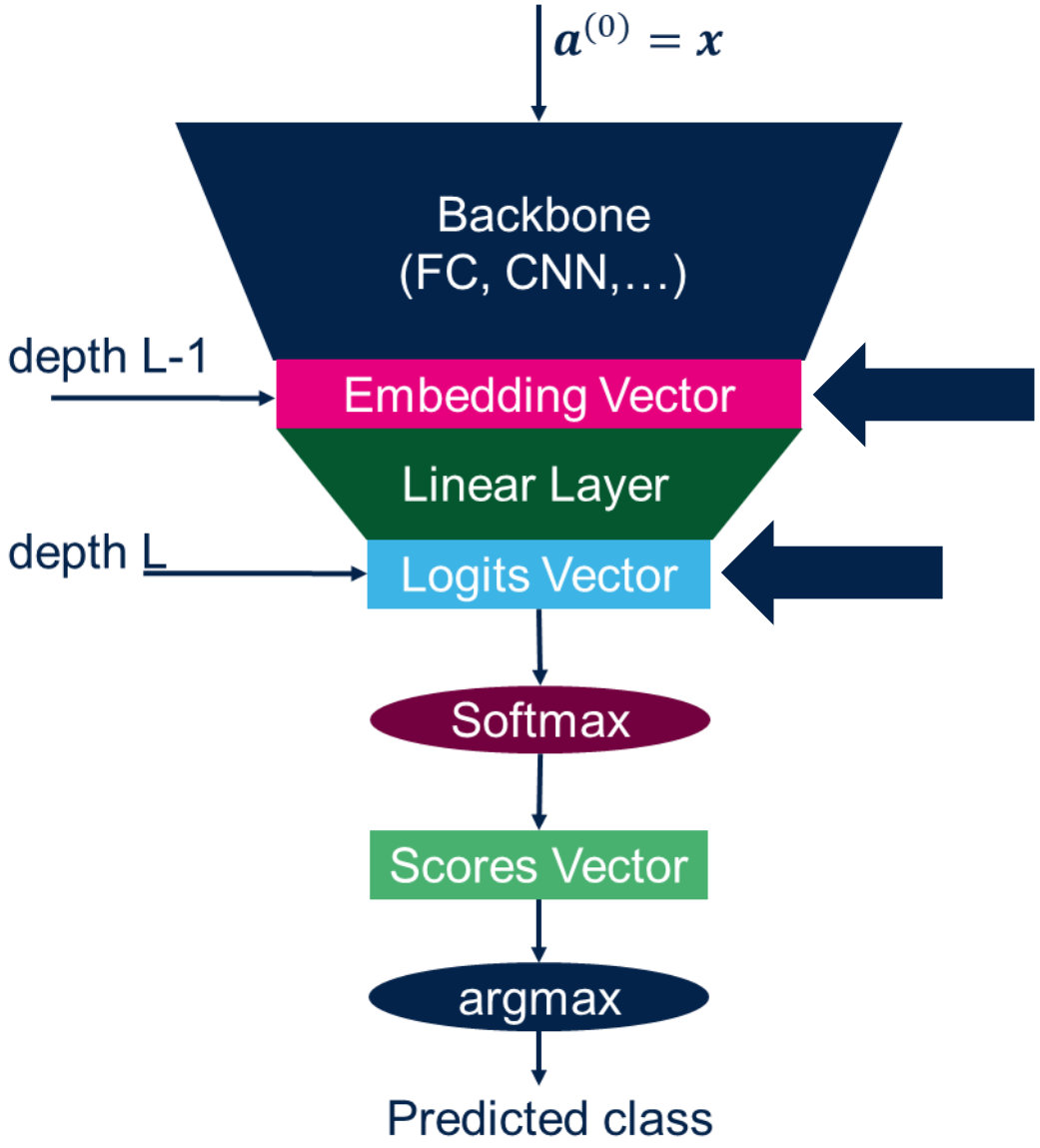


Closed Set Neural Classifiers

- ***Closed Set* Neural Networks** can achieve very **high accuracy,** however they can be **overconfident** on ***wrong*** predictions.

- Confidence via **softmax** outputs is **not calibrated**.

- Softmax does not represent the true likelihood that the prediction is correct. It should be treated as a **score**.



life.augmented

OOD Detection: Integrated & Post-Hoc Methods

Integrated and Post-Hoc Methods for OOD Detection

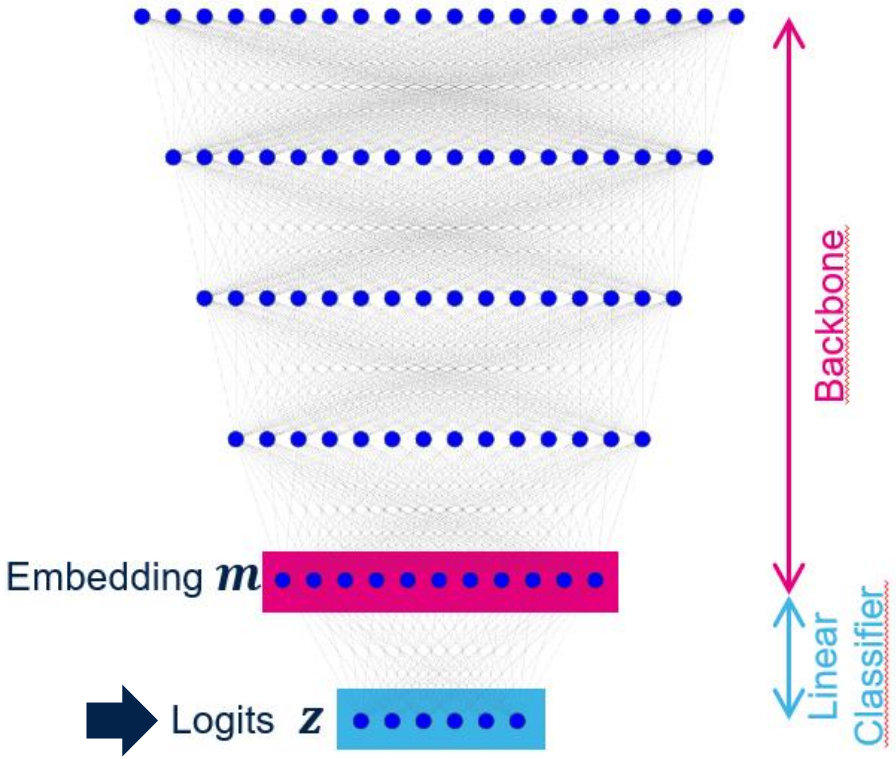
Approach	Main idea	Pros	Cons	Edge-AI Suitability
Integrated	**Modify the model** Build OOD awareness into the model **during training**	Better uncertainty management	**Requires retraining** / Multiple **Inferences** / model **architecture changes**	Very Limited 
Post-hoc	**Interpret the model** Add OOD detection **after training**. • **Extra** scoring/**processing** on the **existing** model outputs or **features** • or using an **auxiliary side-model**	Easier to deploy on edge-devices (post-processing or side-model)	**Inherits** blind-spots and **weaknesses** of the **pre-trained model**.	Depends on detector complexity, but generally good 

life.augmented

Confidence-based methods (Post-hoc on logits)

- **Post-hoc** methods that exploit the classifier's **logits** to detect OOD inputs
- Easy to add to an existing model, cheap to compute
- Typical scores:
  - **Maximum Softmax Probability (MSP)**
  - **Energy score**

- **Limitations**
  - softmax confidence is often **overconfident on OOD**
  - performance can degrade on **near-OOD** samples



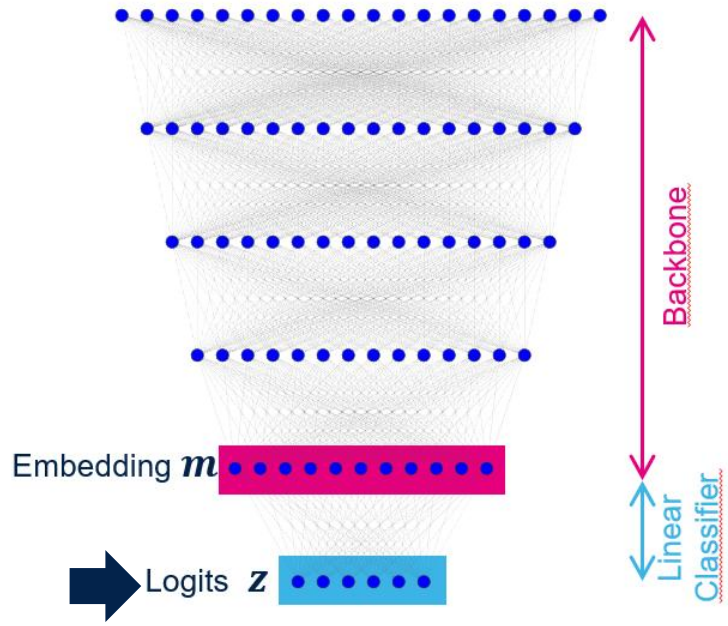
Energy Based Methods (Post-hoc on logits)

Probability that a physical system in state x	Softmax probability that the label is c given sample x
$P(\boldsymbol{x}) = exp(-E(\boldsymbol{x})/kT) / \sum_{\boldsymbol{x}} exp(-E(\boldsymbol{x})/kT)$	$p(c\|\boldsymbol{x}) = exp(z_c) / \sum_i exp(z_i)$

- We can compute the overall Free Energy for sample $\boldsymbol{x}$ as:

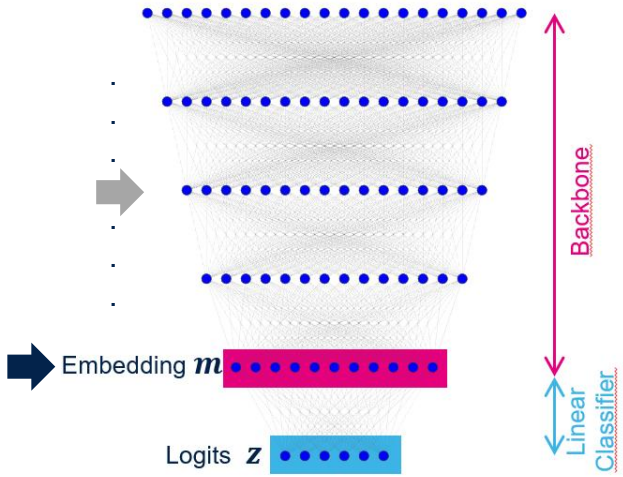
$$E(\boldsymbol{x}) = -log \sum_i exp(z_i)$$

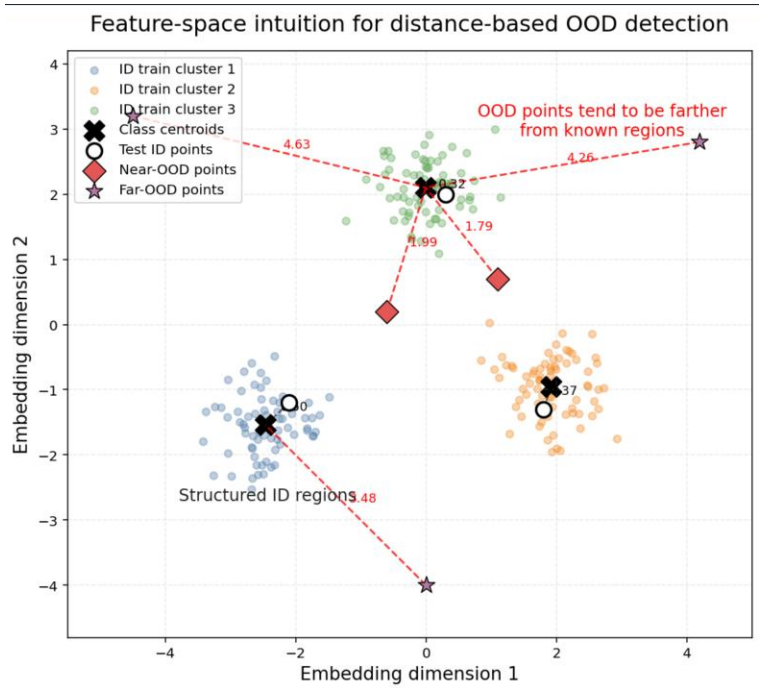
- **Low energy for ID data**, **high energy for OOD data**.
- However, it is **not permutation invariant**, different logits vectors can produce similar energy.



Feature-space / distance-based methods

- **Detect OOD samples by measuring how far a test sample is from known ID regions in feature space.**
- Typical methods:
  - **Mahalanobis distance, k-NN distance in embedding space, Distance to class centroids**
- **Main Limitations**
  - depend on the quality of learned features
  - Cluster shapes are not always compact and unimodal
  - distance metric and layer choice matter

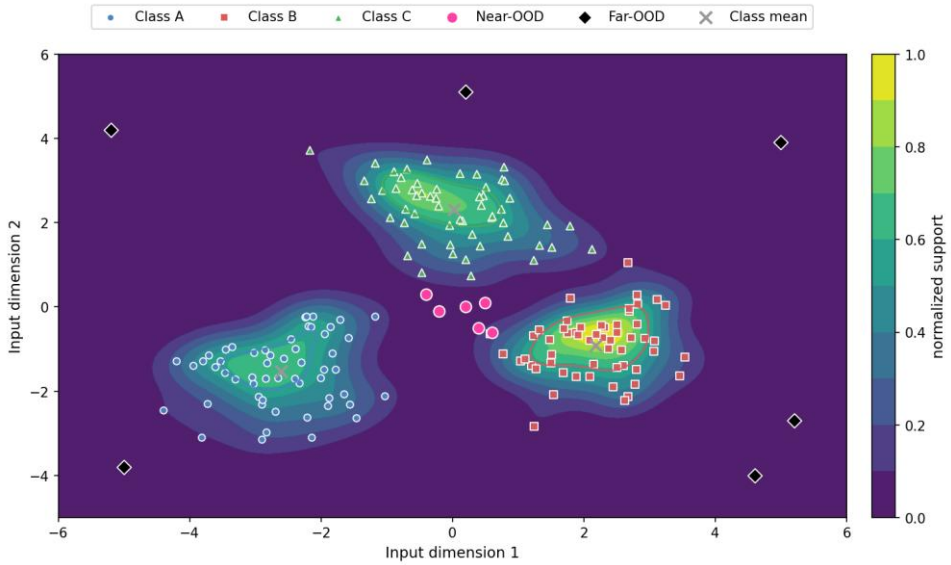




Density / generative methods

- Learn a model of the **training data distribution and flag inputs with low likelihood as potentially OOD.**

- ID inputs → **high likelihood,** OOD inputs → **low likelihood**

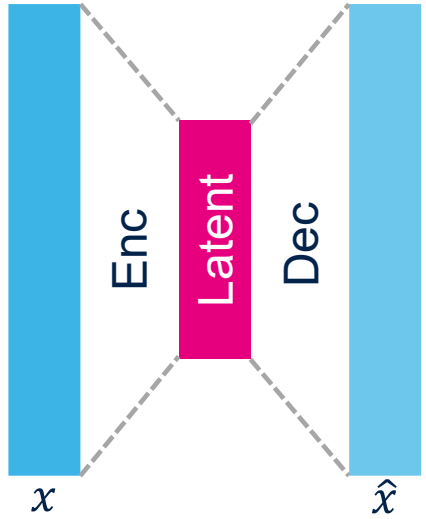


- Typical models:
  - **Normalizing flows, Autoregressive models, Variational Autoencoders (VAEs)**

- **Limitations**
  - **likelihood can be misleading** as some generative models assign high likelihood to OOD data

Reconstruction-based methods

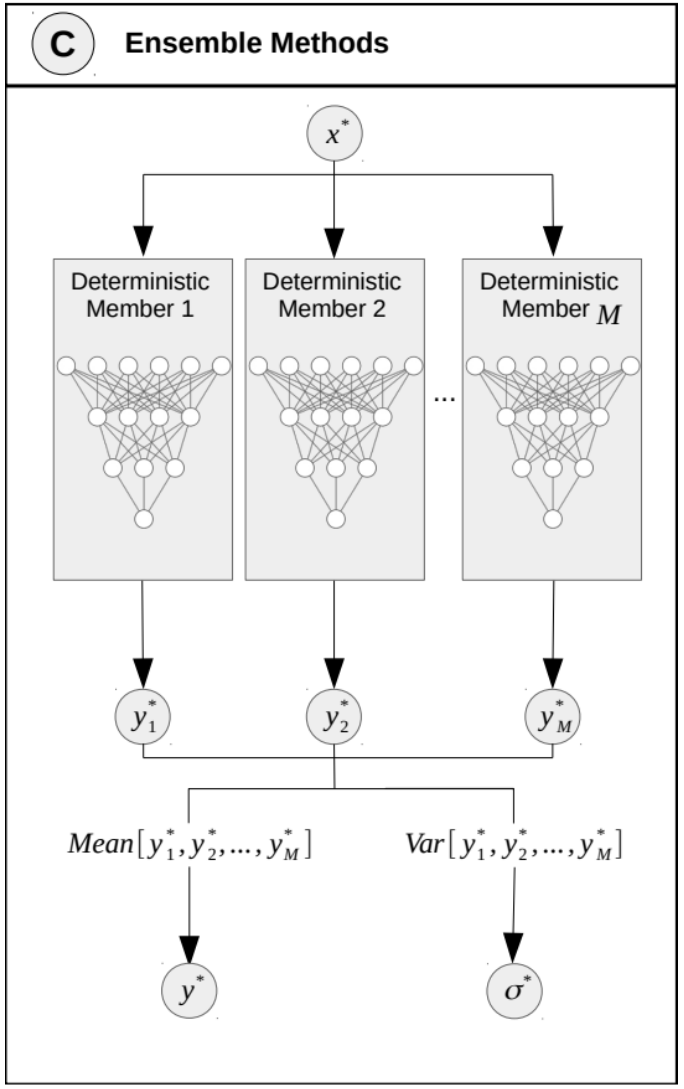
- **Learn to reconstruct in-distribution samples and use high reconstruction error as an OOD signal.**
- ID inputs should be reconstructed well, OOD inputs should be reconstructed poorly
- Typical methods: **Autoencoders**
- **Strengths**: useful in anomaly detection and industrial inspection
- **Limitations**
  - autoencoders may reconstruct OOD samples surprisingly well



$$Error = \|x - \hat{x}\|$$

Deep Ensembles

- **Train multiple models (ensemble)** and look at how much they **agree** or **disagree**.
- The goal is : **better generalization**, under the assumption that a group of models tend to make better decisions.
- Variety into ensembles achieved by:
  - Random Initialization and Data Shuffle, Data Augmentation, Ensemble of different Network Architectures
- Not suitable for embedded systems:
  - Multiple models in memory, Multiple inferences

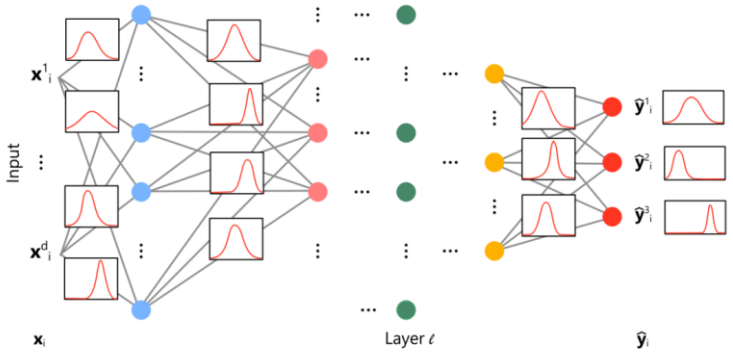


Reference:
2107.03342 "A Survey of Uncertainty in Deep Neural Networks", Gawlikowski et al (2022)

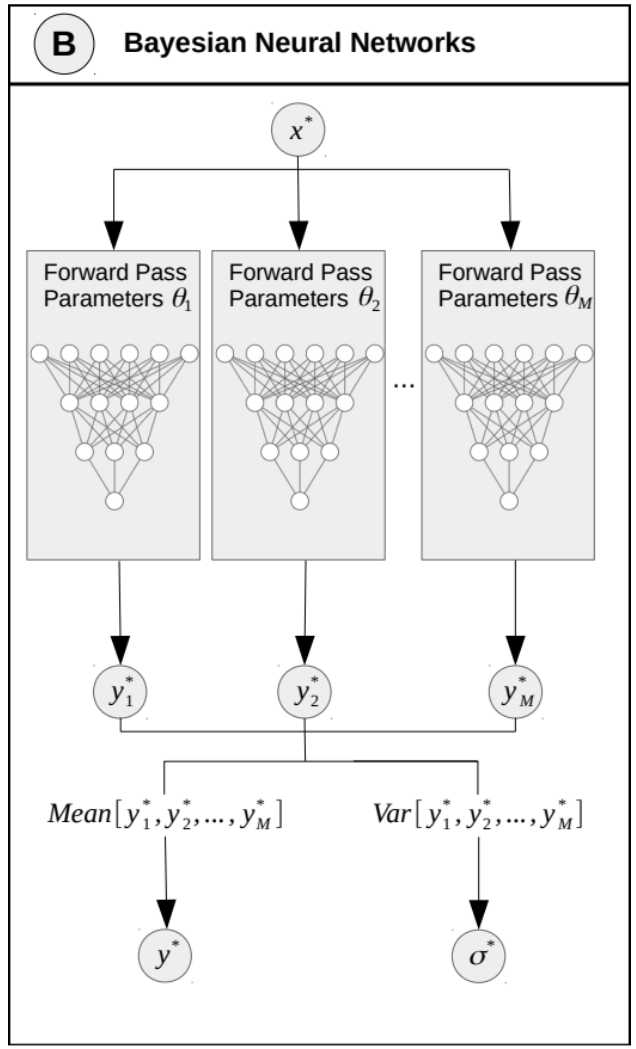
life.augmented

Bayesian Approach

- Bayesian Modeling of the network, its **weights are random variables** with a distribution, **inference is not deterministic**, requires sampling the weights.
- **Weights are sampled before inference**, producing stochastic results.



- In general, not suitable for embedded systems:
  - Sampling of weights, complex inference, multiple inferences, …



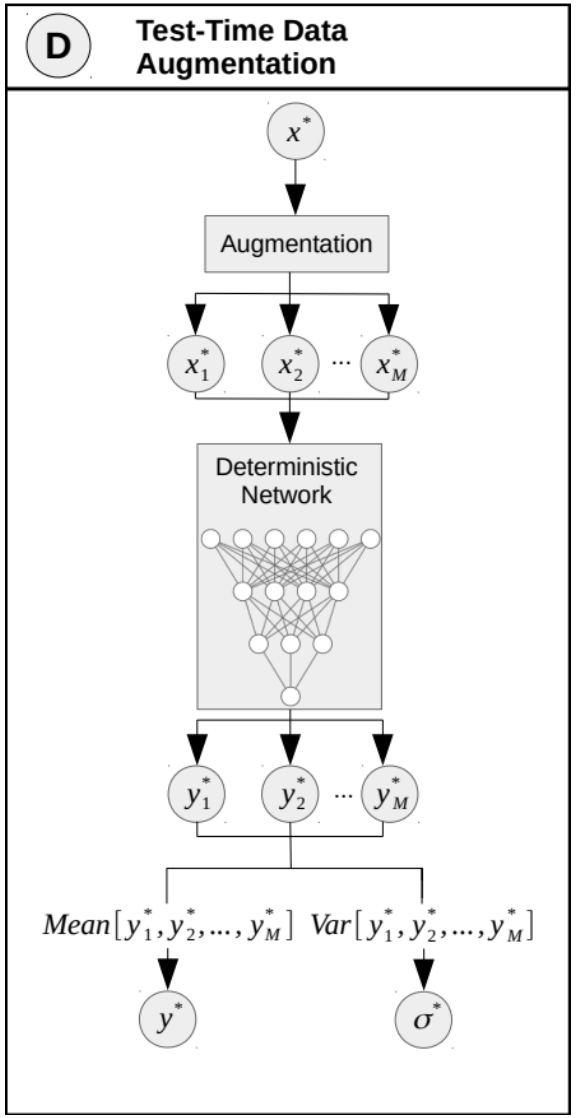
Memory footprint is similar to a standard network
But we pay the multiple inference passes cost.

Reference:
2107.03342 "A Survey of Uncertainty in Deep Neural Networks", Gawlikowski et al (2022)

life.augmented

Test Time Augumentation

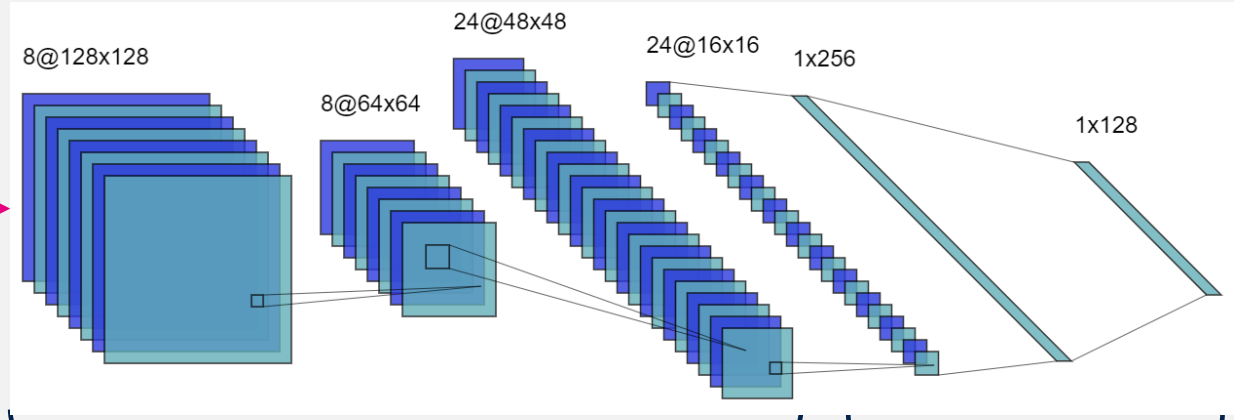
- Relatively Simple Approach
- **Create multiple test samples** from each test sample by **applying data augmentation**
- Test all the augmented samples to compute a predictive distribution in order to measure uncertainty.
- Not ideal for embedded systems:
  - Depends on data augmentation techniques, hard to achieve near and far OOD, multiple inferences potentially, …
- Limited effectiveness.



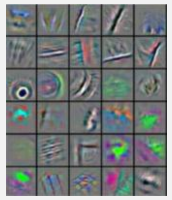
Reference:
2107.03342 "A Survey of Uncertainty in Deep Neural Networks", Gawlikowski et al (2022)

life.augmented

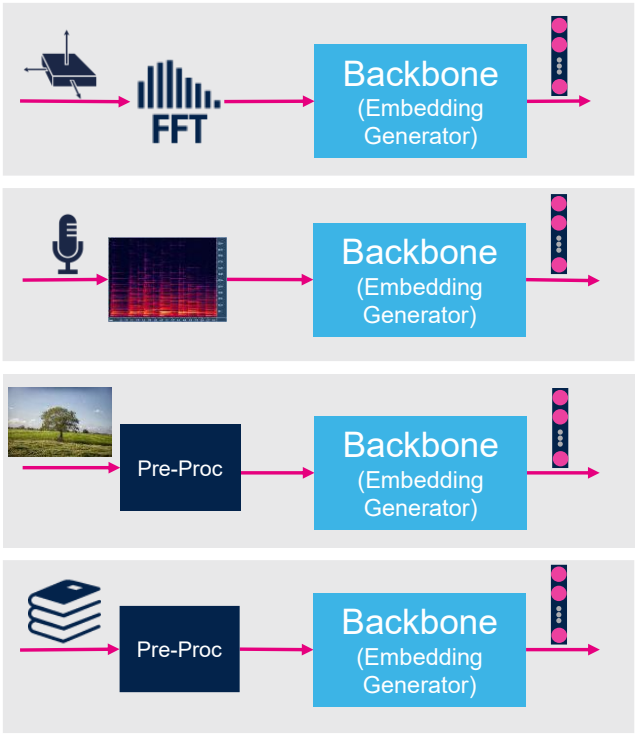
Deep Feature Vectors as Embeddings



- High Dimensionality
- Layers for **“Low” level features**, likely **shared** among **IND** and **OOD**



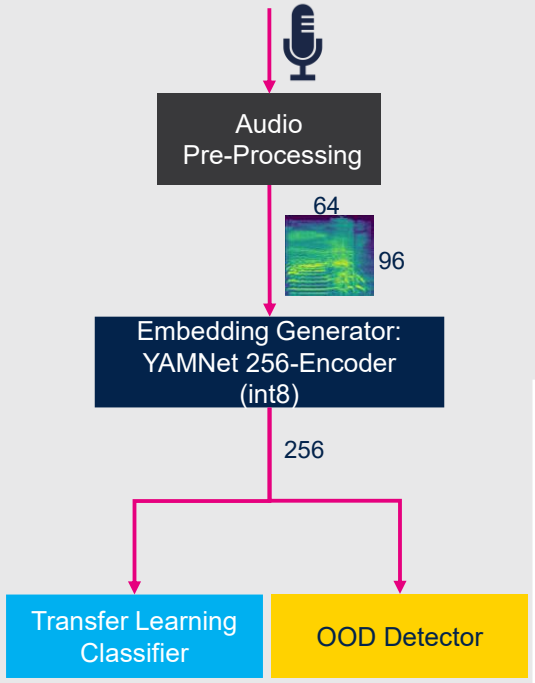
- **“Good” lower dimensional layers** conveying more information **peculiar to In-distribution data**.
- Thus, **useful** for **OOD** Detection



life.augmented

OOD Detection

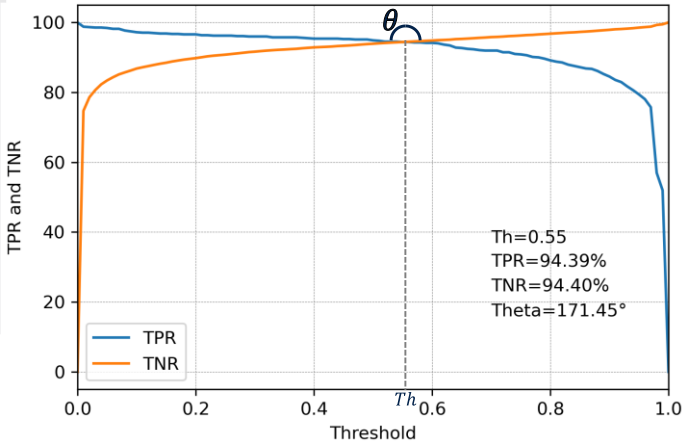
Processing Pipeline



YAMNet: based on MobileNetV1. Pre-trained on AudioSet (521 classes from >2M YouTube audio clips).

In-distribution: **4 classes** from **Alert Sounds** : ***Baby Crying***, ***Door Knock***, ***Clock Alarm***, ***Glass Breaking***.

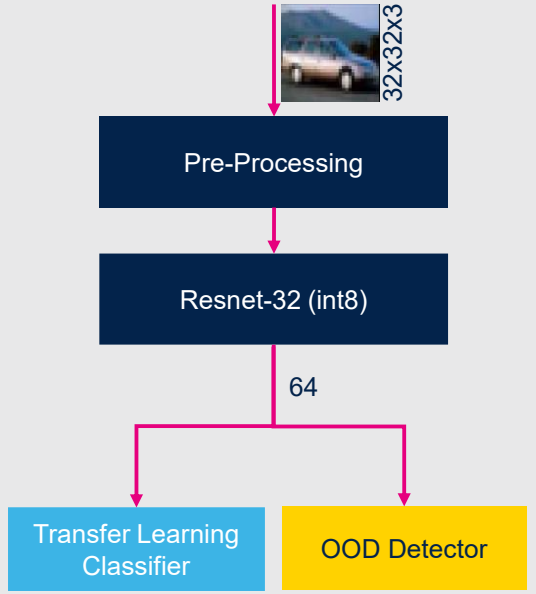
OOD: **remaining 46 classes** from **ESC-50**



OOD Detection (Audio Domain): Acoustic Event Detection

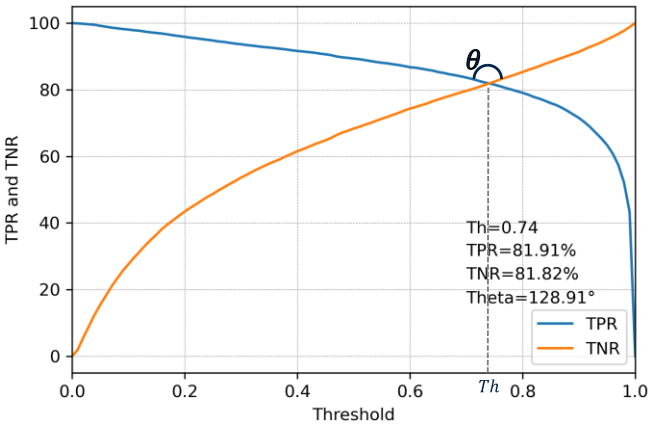
Method	Bal. Acc. (%)	Theta (°)
Proposed Method	94.40	171.4
Baseline (MSP)	84.57	-

Processing Pipeline



OOD Detection (Image Domain): CIFAR-10 vs CIFAR-100

Cifar-100 contains 100 classes.
Small overlap with Cifar-10.

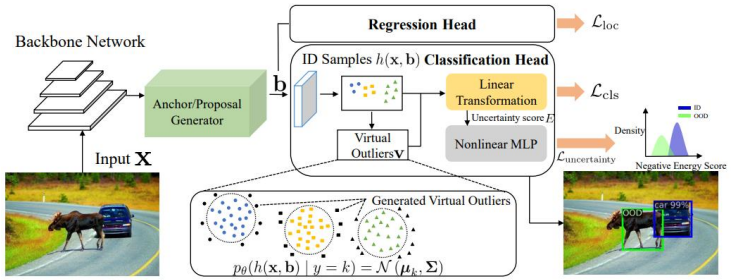


Method	Bal. Acc. (%)	Theta (°)
Proposed Method	81.87	128.9
Baseline (MSP)	66.57	-

life.augmented

Outlier Exposure (Real and Synthetic)

- Core idea: Train the model with explicit OOD examples to learn rejection.
- Problem: **Real OOD** data is often **hard or impossible to collect**
- OE approximates the unknown OOD distribution using surrogate samples
- Solution: **Synthetic Outliers** (in ambient or latent space):
  - **Provide corrupted** hard-ID samples or samples from **complementary** datasets or **genAI** synthetic samples (high cost)
  - Or generate samples in the latent space:



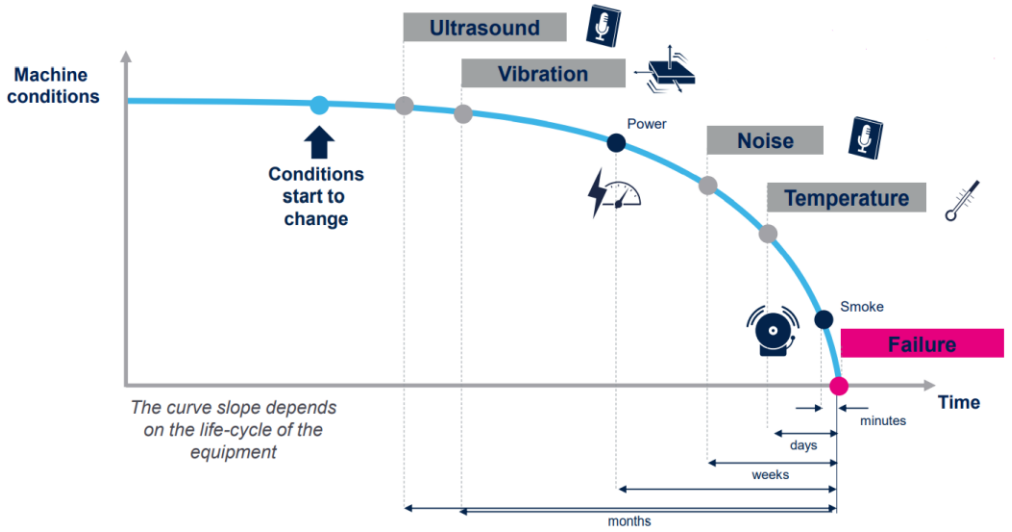
From: 2202.01197 (VOS: LEARNING WHAT YOU DON'T KNOW BY VIRTUAL OUTLIER SYNTHESIS 2022)

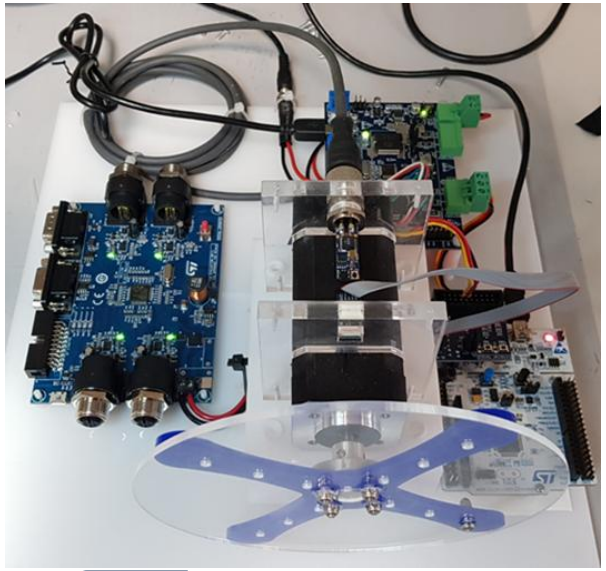
- Limitation: The model learns what we show as OOD and may not generalize to truly unknown data

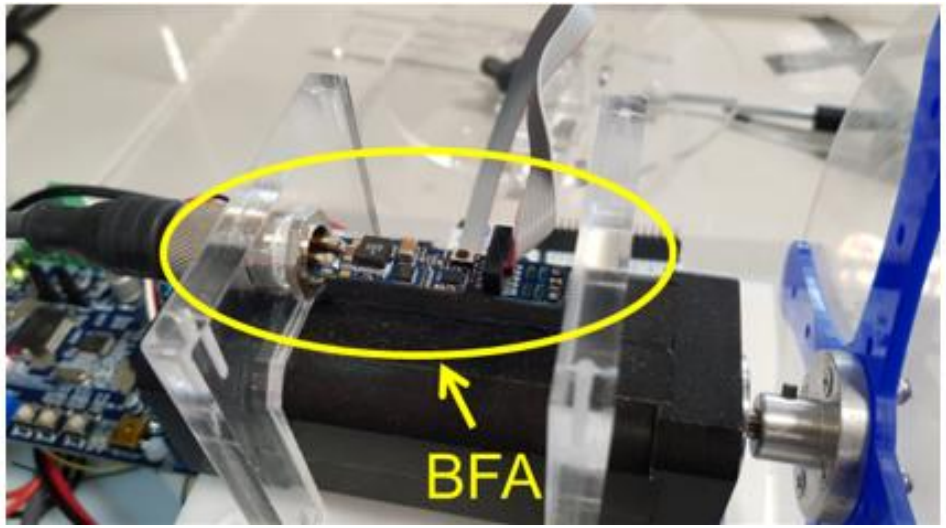
Example: Anomaly Detection in Predictive Maintenance (Inertial Domain)

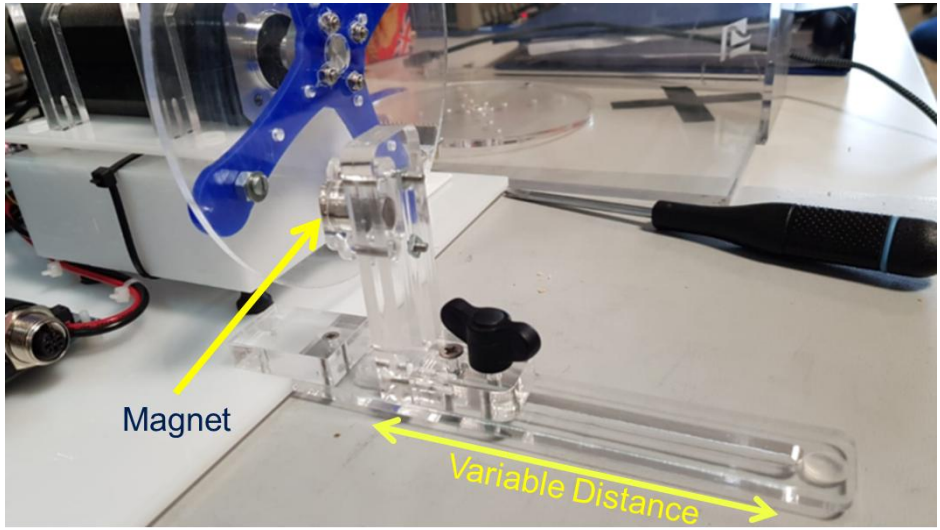
Motor Vibration Anomalies with Autoencoder

- **Goal** : detect vibration anomalies on a using an embedded device.
- **Injected Anomalies**: Unbalancing (screws in the disc), Misalignment (magnet close to the disc)
- **Testing Conditions and Acquisition Parameters**:
  - 1800, 2160, 2520, 2880, 3240, 3600, 3960 RPMs (i.e. from 30Hz to 66Hz, step 6Hz)
  - Accelerometer ODR=1,6KHz, FFT 1024 points, overlap 75%, FFT averaging





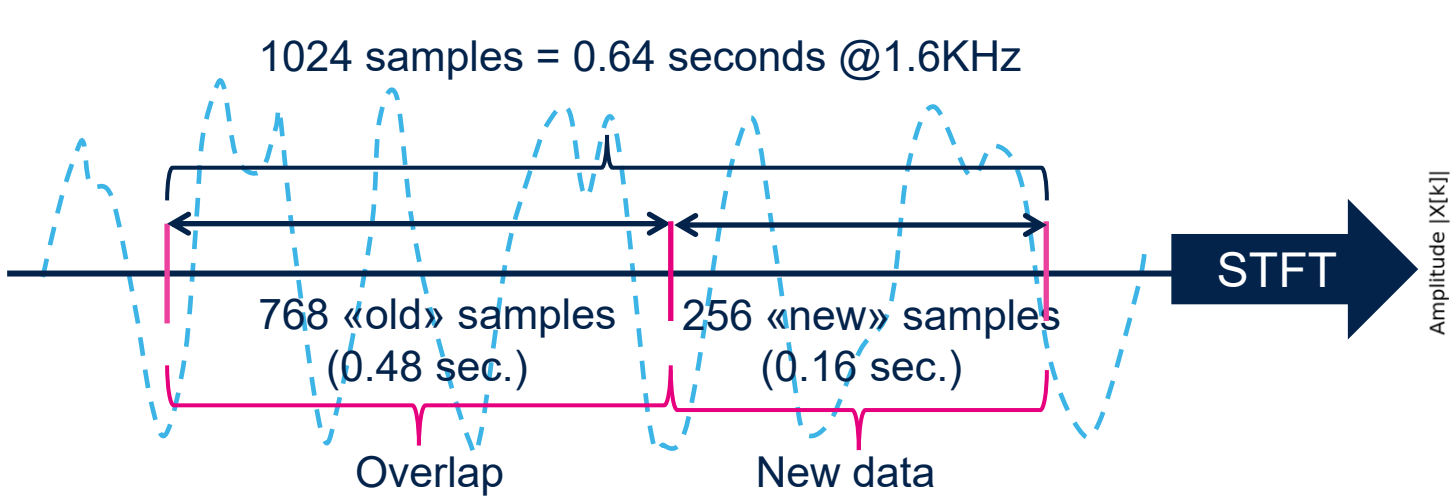




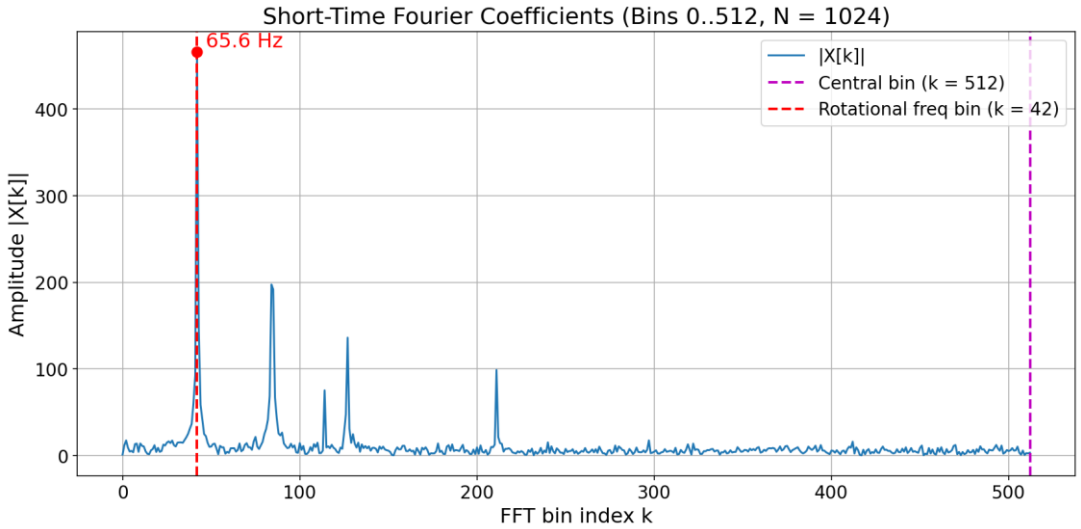
[STEVAL-BFA001V2B - Multi-sensor predictive maintenance kit with IO-Link stack v.1.1 - STMicroelectronics]

Overlapping Time Windows

- A 3-axes accelerometer attached to a dynamical system records the evolution of motion over time and generates three 1D time series.
- Processing time windows with 1024 samples each and with overlapping.
- Length is power of 2 to allow efficient FFT.



STFT

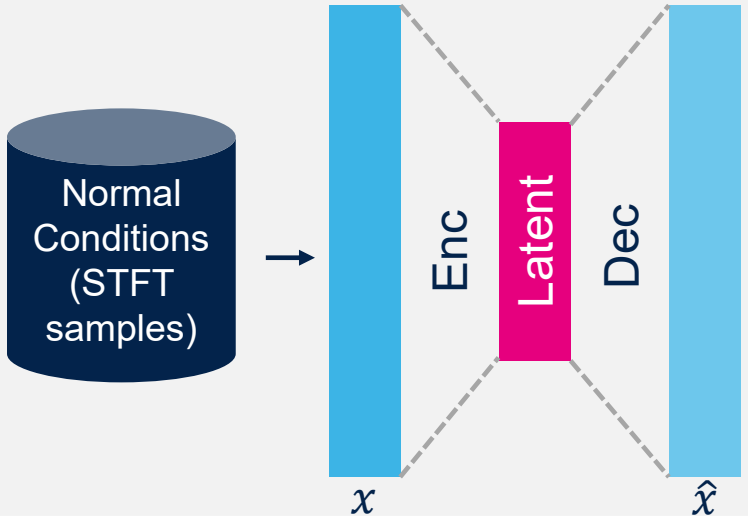


To further reduce noise, N noise signatures are averaged, e.g. N=8

life.augmented

Autoencoders for Anomaly Detection

Autoencoder



$$Error = \|x - \hat{x}\|$$

AE for feature extraction:
Only the Encoder is necessary

AE for anomaly detection:
Encoder and Decoder are necessary

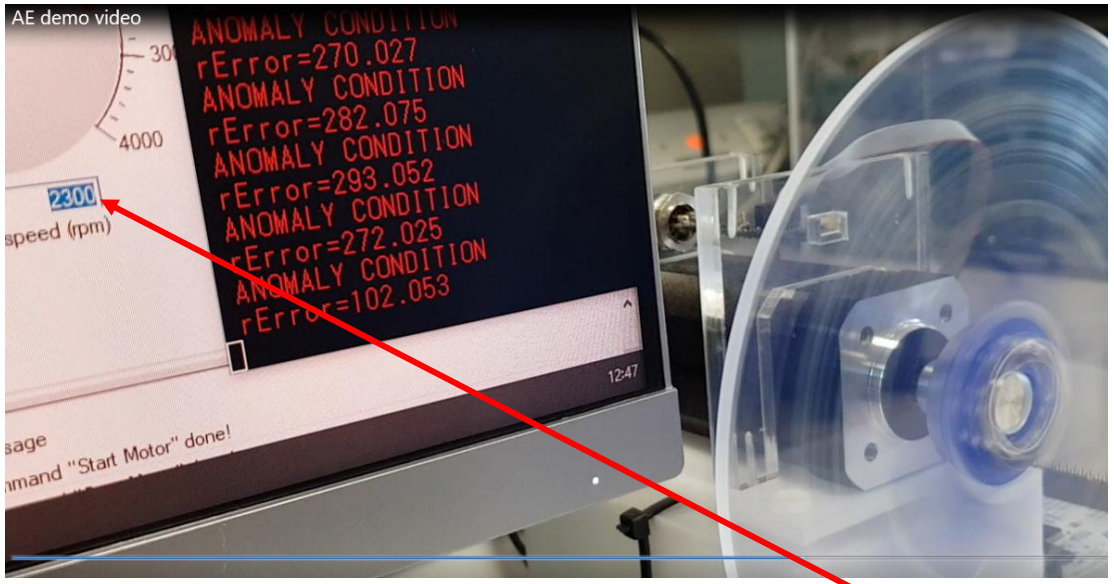
- **Train the Autoencoder on normal data only**. (STFT 512-dimensional vectors for each of the 7 speeds).
- **No need to acquire data about anomalies**.

- Compute the **difference** between the **decoded** vector $\hat{x}$ and the **original** input x according to a chosen **metric**, e.g., **MSE**.

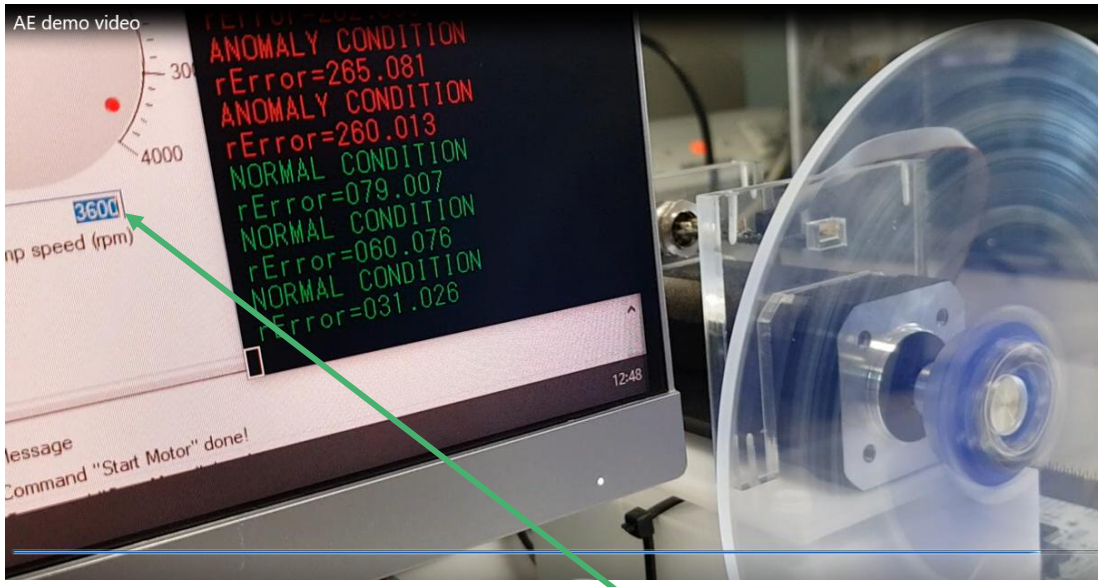
- If the reconstruction error E is «high» then its anomalous data → Raise Anomaly Flag

life.augmented

Autoencoder Anomaly Detection Demo



AE Anomaly Detector @ «unknown speed» 2300RPM



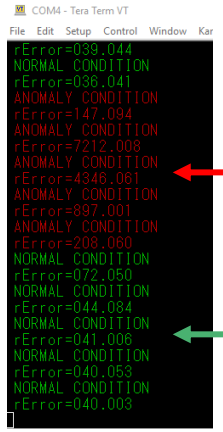
AE Anomaly Detector during speed sweep arrives at «known speed» 3600RPM

```
**********************************************************
* ACCELEROMETER & VIBRATION parameters values            *
**********************************************************

Accelerometer parameters are:
HpfCut =3       Acc_Odr=1660    FifoOdr=1660    Acc_Fs =2


MotionSP parameters:
size=1024       tau=50  wind=1  tacq=1400      ovl=75
```

Accelerometer settings



The systems passes through unknown speeds (transients)

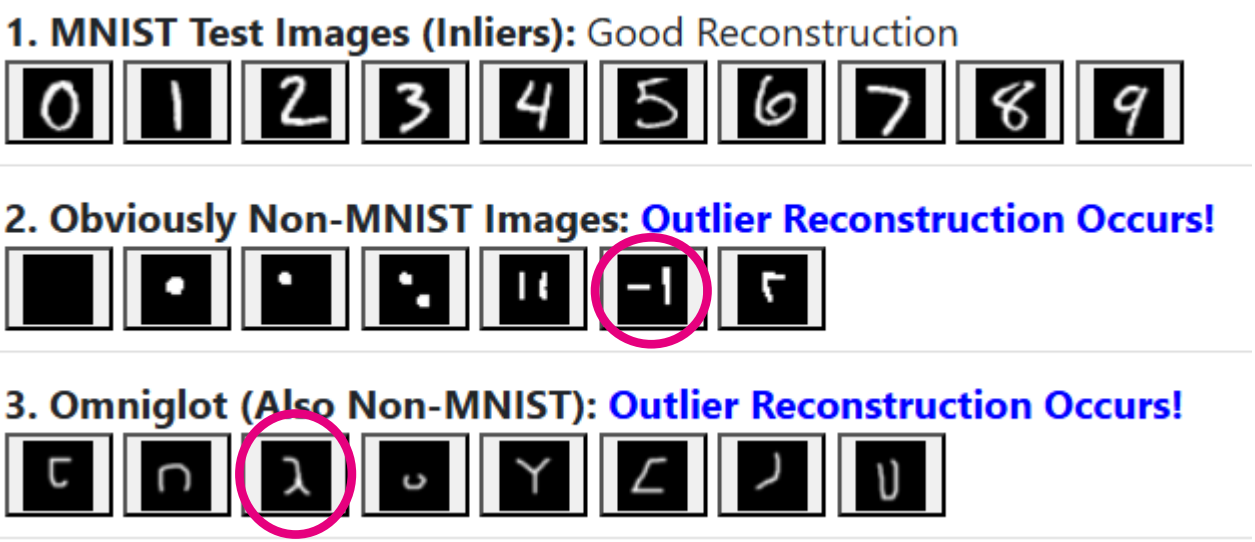
The systems is at a known speed. No disc unbalancing

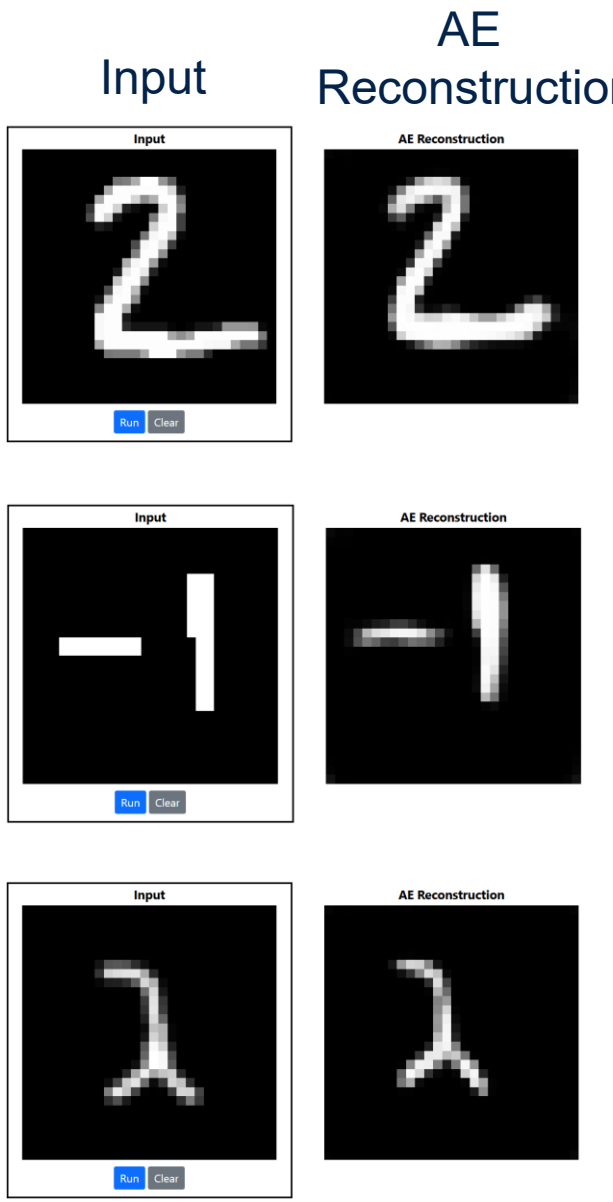
Anomaly Detector output during a speed sweep

life.augmented

Autoencoders could reconstruct anomalous data

- Outlier Reconstruction Web Demo





Reasons:

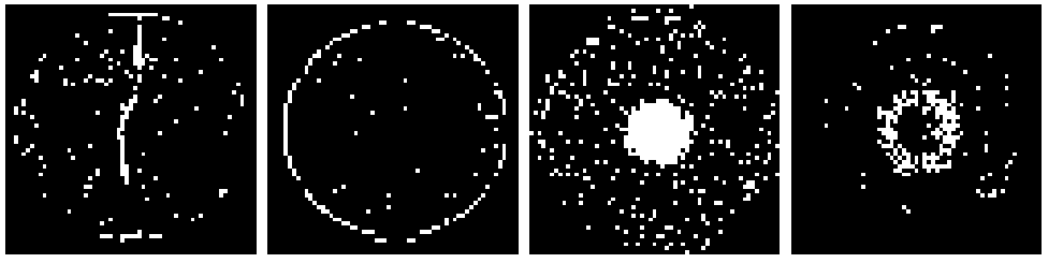
- If the AE bottleneck is not sufficiently constraining (overcomplete latent, strong skip connections, insufficient regularization)
- Near-OOD samples could be *easily* reconstructed.

life.augmented

OOD in Production

- Identify new types of defects in a production line can be useful to determine if some machinery is working properly or not.

- Useful to reduce the efforts of human in the loop supervision: only new defects are passed to the human supervision for deeper inspection.



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

Figure 1: EWS maps showing the following characteristic anomalies signatures: (a) Scratch, (b) Ring, (c) Spot, and (d) Wheel. White pixels identify defective dies.

«Semisupervised Classification of Anomalies Signatures in Electrical Wafer Sorting (EWS) Maps" Luigi C. Viagrande, Filippo L. M. Milotta, Paola Giuffre`, Giuseppe Bruno, Daniele Vinciguerra and Giovanni Gallo (STMicroelectronics and University of Catania) 89144.pdf

Open Vocabulary Object Detection

Out of Distribution Object Detection

- OOD in object detection tasks is even harder.

- **Open Vocabulary Object Detection (OVOD):**
  - detect objects from a broad or open-ended label space, often using text-image alignment.

- OVOD systems typically combine:
  - **Visual embeddings** (from CNNs, Transformers, etc.)
  - **Text embeddings** (e.g., word embeddings, **CLIP-like models**)
  - **Cross-modal alignment** between image regions and textual descriptions.

- Example models: YOLOE / YOLOE-26 for embedded devices

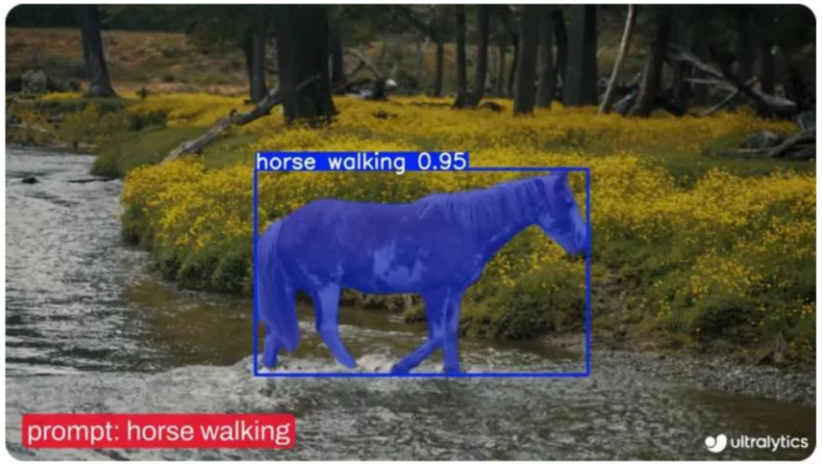


Fig 3. An example of using YOLOE to detect specific objects using a text prompt.

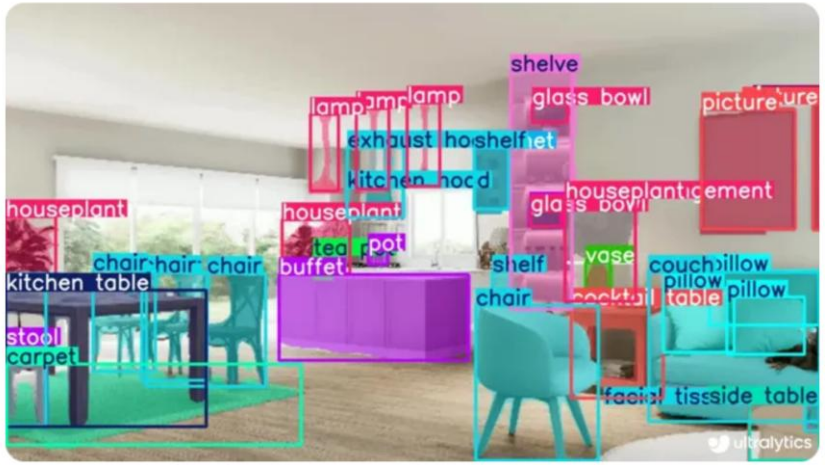


Fig 4. Using YOLOE in prompt-free mode.

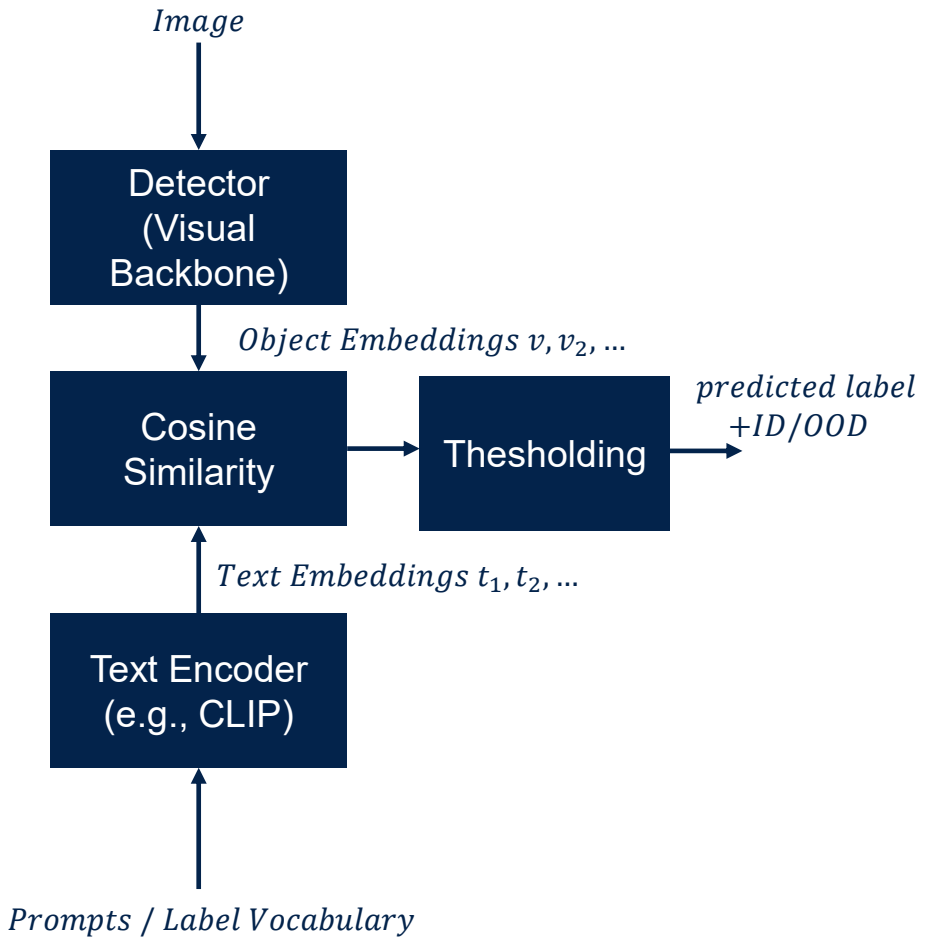
Source: What is YOLOE? Taking computer vision models further | Ultralytics

Open Vocabulary Object Detection (YOLOE)

- The detector extracts **visual embeddings** for detected objects
- A text encoder generates **text embeddings** from:
  - user prompts, or
  - a large vocabulary of candidate labels
- Because of **vision-language alignment during training**, visual and text embeddings live in a **shared embedding space**
- Each detected object is assigned the label with the **highest cosine similarity** to its visual embedding:

$$l = \underset{i \in \{1,\dots,K\}}{\boldsymbol{argmax}}\, \boldsymbol{cos\_sim}(v, \hat{t}_i), \qquad K = \#\boldsymbol{classes}$$

- If the best similarity is **below a threshold**, the object can be **rejected as unknown / OOD**



life.augmented