Transfer Learning Project

Chenyu Tian
Columbia University
ct3308@columbia.edu

Yilu Yang
Columbia University
yy3626@columbia.edu

1 Introduction

This project aims to solve a hierarchical classification problem by predicting both a superclass and its corresponding subclass for a given image. The problem has two primary complexities that test model robustness and generalization:

Open Set Recognition (OSR): The test set contains novel superclasses and subclasses not seen during training. The model must identify and reject these "unknown" samples as "novel".

Distribution Shift: The test set's class frequencies may differ from the training set. The model must generalize to this new statistical distribution of known classes.

2 Related Work

OSR contrasts with traditional closed-set recognition, which assumes all testing classes are known during training. The methods for solving OSR problems primarily fall into the following two groups [SD23].

Discriminative Models: Discriminative models aim to learn decision rules directly. This is often approached either by learning highly compact and discriminative representations for known classes (through score-based, distance-based or reconstruction-based methods) or by explicitly introducing "unknown" information into the training process. This unknown information can be synthesized from known classes.

Generative Models: The generative models is another approach which can be further divided into Instance and Non-Instance Generation-based methods [GHC20]. The former method focuses on generating useful new samples, while the second one focuses on learning the underlying distributions of the known-class data, often using Autoencoders (AEs) or Generative Adversarial Networks (GANs). The principle is that unknown samples will not fit the learned distributions, allowing for their detection based on deviation or high reconstruction error.

3 Method / Algorithm

3.1 Dataset Preparation

Simulate Open-Set: Simulate an open-set scenario by holding out specific subclasses to serve as novel samples in the validation set.

Distribution Shift

- **Data Augmentation:** Adopt the suggestion from the brief to use data augmentation (e.g., brightness, contrast, minor geometric transforms) to improve generalization.
- **Validation Set:** Build a validation set for hyperparameter tuning.

3.2 Framework

Since it is a hierarchical OSR problem, we propose different hierarchical models.

- **Model Structure:** Use a pre-trained feature extractor (e.g., CLIP [RKH+21]) followed by a superclass classification head and a subclass classification head, and train the network jointly using a combined loss from both heads.
- **Hierarchical Constraints:** To ensure the superclass classification impacts the subclass classification, we apply a soft constraint (SE-style Feature Gating [HSA+19]) during training and a hard constraint (Masking) during inference.

Training with SE-style Feature Gating: During Training, we apply SE-style Feature Gating to recalibrate features based on their importance for superclass classification.

Given input features $\mathbf{x} \in \mathbb{R}^d$ from CLIP, we define:

Superclass Head:

$$\mathbf{z}^{\text{super}} = W^{\text{super}}\mathbf{x} + \mathbf{b}^{\text{super}} \tag{1}$$

SE Feature Gating (Squeeze-Excitation):

$$\mathbf{s} = \sigma(W_2 \cdot \text{ReLU}(W_1\mathbf{x})) \qquad \text{(Squeeze \& Excitation)} \tag{2}$$

$$\tilde{\mathbf{x}} = \mathbf{s} \odot \mathbf{x} \qquad \text{(Feature Recalibration)} \tag{3}$$

where $W_1 \in \mathbb{R}^{(d/r)\times d}$, $W_2 \in \mathbb{R}^{d\times(d/r)}$, $r = 4$ is the reduction ratio, and $\odot$ denotes element-wise multiplication.

Subclass Head (with attended features):

$$\mathbf{z}^{\text{sub}} = W^{\text{sub}}\tilde{\mathbf{x}} + \mathbf{b}^{\text{sub}} \tag{4}$$

Joint Cross-Entropy Loss:

$$\mathcal{L} = \mathcal{L}_{\text{CE}}(\mathbf{z}^{\text{super}}, y^{\text{super}}) + \mathcal{L}_{\text{CE}}(\mathbf{z}^{\text{sub}}, y^{\text{sub}}) \tag{5}$$

Inference with Hierarchical Masking: During inference, we apply hierarchical masking to enforce consistency between superclass and subclass predictions.

Let $\hat{c}^{\text{super}}$ be the predicted superclass and $\mathcal{S}_c$ be the set of valid subclass indices for superclass c. The masked subclass logits are:

$$\tilde{z}_i^{\text{sub}} = \begin{cases} z_i^{\text{sub}} & \text{if } i \in \mathcal{S}_{\hat{c}^{\text{super}}} \\ -\infty & \text{otherwise} \end{cases} \tag{6}$$

3.3 Models

We categorize the approaches we explored into post-hoc methods applied to standard trained models, and methods that require modifying the training process or architecture.

3.3.1 Post-hoc Score-based Methods

These methods define a scoring function based on the outputs (logits or softmax probabilities) of a pre-trained closed-set classifier. A threshold is applied to this score to reject novel samples.

Threshold Methods: For all score-based methods, a sample is classified as novel if its score falls below (or above, for Energy) a threshold τ calibrated on the validation set to achieve a target recall rate.

- Maximum Softmax Probability (MSP) [HG18]: The baseline approach. It uses the maximum value of the softmax distribution as the confidence score. Logit $\rightarrow$ Softmax $\rightarrow$ Threshold Gate.

- Out-of-Distribution Detector (ODIN) [LLS20]: Improves MSP by applying temperature scaling to the logits before softmax and adding small perturbations to the input image. Logit $\rightarrow$ Temperature Scaling $\rightarrow$ Softmax $\rightarrow$ Threshold Gate.
- Energy-based Out-of-distribution Detection [LWOL21]: Instead of softmax, this method uses the free energy derived from the logits as the scoring function. Lower energy indicates known data, while higher energy indicates novel data. Logit $\rightarrow$ Energy $\rightarrow$ Threshold Gate.

Maximum Softmax Probability (MSP): Given an input $\mathbf{x}$, the MSP score is defined as the maximum class probability:

$$S_{\text{MSP}}(\mathbf{x}) = \max_i \frac{\exp(z_i(\mathbf{x}))}{\sum_{j=1}^{C} \exp(z_j(\mathbf{x}))} \tag{7}$$

ODIN (with Temperature Scaling): ODIN applies temperature scaling $T > 1$ to the logits to separate the softmax score distributions of ID and OOD samples:

$$S_{\text{ODIN}}(\mathbf{x}; T) = \max_i \frac{\exp(z_i(\mathbf{x})/T)}{\sum_{j=1}^{C} \exp(z_j(\mathbf{x})/T)} \tag{8}$$

Energy Score (with Temperature Scaling): The free energy function $E(\mathbf{x}; T)$ using temperature T is defined as:

$$E(\mathbf{x}; T) = -T \cdot \log \sum_{i=1}^{C} \exp(z_i(\mathbf{x})/T) \tag{9}$$

Lower energy values indicate higher confidence that the sample belongs to a known class. To maintain consistency with MSP and ODIN (where higher scores indicate known classes), we define the Energy score as the negated energy:

$$S_{\text{Energy}}(\mathbf{x}; T) = -E(\mathbf{x}; T) = T \cdot \log \sum_{i=1}^{C} \exp(z_i(\mathbf{x})/T) \tag{10}$$

Temperature Limit Analysis: The effect of temperature can be understood by examining the limiting cases:

ODIN as $T \rightarrow \infty$: As temperature increases, the softmax distribution becomes uniform, and the score approaches a linear function of the logit gap:

$$\lim_{T \to \infty} S_{\text{ODIN}}(\mathbf{x}; T) \approx \frac{1}{C} + \frac{1}{C \cdot T} \left(z_{\max}(\mathbf{x}) - \bar{z}(\mathbf{x}) \right) \tag{11}$$

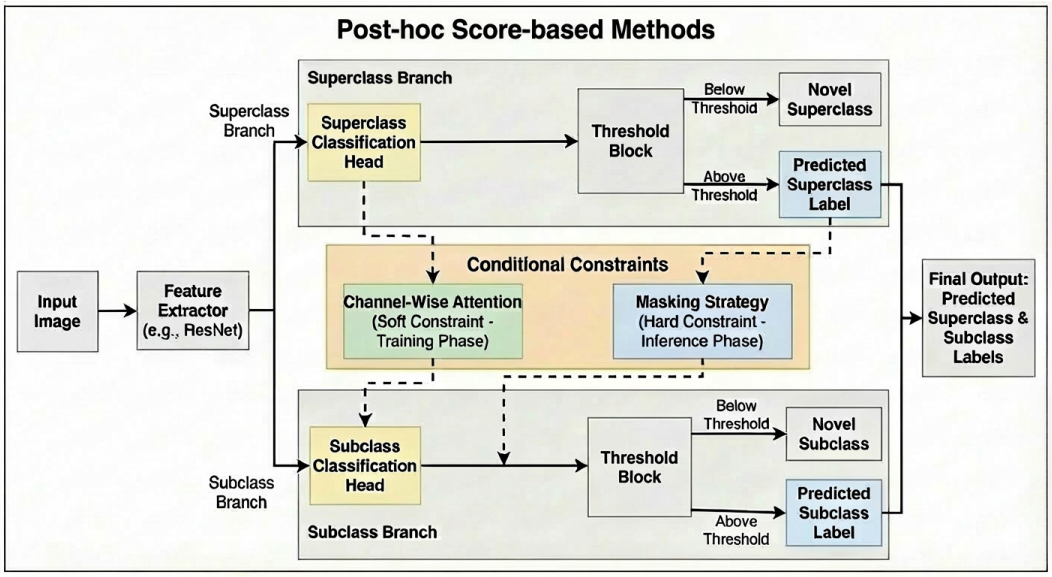
where $z_{\max}$ is the maximum logit and $\bar{z}$ is the mean logit. This reveals that high-temperature ODIN measures the **Relative Sharpness** of the logit distribution.

Energy as $T \rightarrow 0$: As temperature decreases, the LogSumExp converges to the maximum:

$$\lim_{T \to 0} E(\mathbf{x}; T) = -\max_i z_i(\mathbf{x}) \tag{12}$$

This shows that low-temperature Energy reduces to the negative **MaxLogit**, measuring the **Absolute Magnitude** of the strongest prediction.

Magnitude Information: A known issue with MSP and ODIN is their reliance on the Softmax function. Softmax forces output probabilities to sum to 1, losing the magnitude information of the underlying logits, which often leads to **overconfidence** on unseen data. Energy-based methods mitigate this by utilizing the raw logit magnitudes.



3.3.2 Class-wise Sigmoid with BCE Loss

Consistency between Threshold Method and Training Objective: In the MSP framework, the thresholding method (based on Softmax probabilities) and the training objective (Cross-Entropy loss based on Softmax) are consistent, as neither preserves magnitude information. However, for Energy-based OOD detection, there is a theoretical inconsistency: the thresholding method relies on Logits and their magnitude information, whereas the training objective (CE loss with Softmax) discards this information. Therefore, to ensure consistency and preserve magnitude information, we propose replacing Softmax with **Class-wise Sigmoids** [SXL17].

Formulation: Instead of the standard Softmax normalization, we apply independent sigmoid activations to each class logit, treating the multi-class classification problem as multiple one-vs-rest binary classification problems:

$$p_i = \sigma(z_i) = \frac{1}{1 + e^{-z_i}}, \quad i = 1, \ldots, C \tag{13}$$

where z_i is the logit for class i, and C is the number of classes.

The training objective becomes Binary Cross-Entropy (BCE) loss with one-hot encoded targets:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{C} \sum_{i=1}^{C} [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \tag{14}$$

where $y_i \in \{0, 1\}$ is the one-hot encoded ground truth for class i.

Maximum Sigmoid (MaxSigmoid) Score: Since we have replaced Softmax with Sigmoid in the training objective, we can define a consistent inference method that also uses Sigmoid probabilities. The OOD score is computed as the maximum sigmoid probability:

$$S_{\text{MaxSigmoid}} = \max_i \sigma(z_i) \tag{15}$$

3.3.3 Trainable Confidence Estimation Methods

Unlike post-hoc methods, these approaches modify the model architecture during training to explicitly learn representations better suited for distinguishing known from unknown.

- **Auxiliary Confidence Gating [DT18]:** Add a parallel confidence branch alongside the main classification branch. The network is trained to output both a class prediction and a scalar confidence score via a sigmoid output. The model learns to output low confidence for samples that are difficult to classify or likely out-of-distribution.

3.3.4 OpenMax Hierarchical Classifier

OpenMax [BB16] is a classic discriminative method in OSR, which is based on **Feature Distribution** (Weibull Distribution).

- **Stage 1 - OpenMax for Superclass:** Use OpenMax at the superclass level, it will either classify a sample as known superclass or novel superclass.
- **Stage 2 - OpenMax for Subclass:** If the sample is identified as novel at the previous level, it will also be classified as novel subclass. Otherwise, we train a separate OpenMax classifier on its corresponding set of known subclasses.

3.3.5 Hierarchical Class Anchor Clustering (CAC)

CAC is a Discriminative method that is based on **Metric Learning**.

- **Stage 1 - Hierarchical Anchor Design:** Design structured anchors [MSMD21] by first defining a unique base vector for each superclass. Subclass anchors will then be created by adding small offset vectors to their corresponding superclass base vector.
- **Stage 2 - Training:** Train the network using CAC loss function with hierarchical anchors forcing the network to learn a semantically meaningful logit space where the cluster layout directly mirrors problem's hierarchy.
- **Stage 3 - Inference:** Use the distance-based rejection process to classify known subclasses, while distinguishing rejected samples as either "Unknown Superclass" or "Unknown Subclass".

3.4 Evaluation

- **Primary Metric:** Our top priority is to significantly improve upon the CLIP baseline's performance on "Unseen Accuracy".
- **Secondary Metrics:** We will monitor all official metrics, including "Overall Accuracy", "Seen Accuracy", and "Categorical Cross-entropy" for both class levels.
- **Threshold Tuning:** To determine the gatekeeper's probability threshold and the expert's confidence threshold, we will simulate an open-set scenario (e.g., by holding out subclasses during training) and tune on our local validation set to maximize "Unseen Accuracy".

References

[BB16] Abhijit Bendale and Terrance E. Boult. Towards Open Set Deep Networks . In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1563–1572, Los Alamitos, CA, USA, June 2016. IEEE Computer Society.

[DT18] Terrance DeVries and Graham W. Taylor. Learning confidence for out-of-distribution detection in neural networks, 2018.

[GHC20] Chuanxing Geng, Sheng-jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3614–3631, 2020.

[HG18] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks, 2018.

[HSA+19] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2019.

[LLS20] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks, 2020.

[LWOL21] Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection, 2021.

[MSMD21] Dimity Miller, Niko Sunderhauf, Michael Milford, and Feras Dayoub. Class anchor clustering: A loss for distance-based open set recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3570–3578, 2021.

[RKH+21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[SD23] Jiayin Sun and Qiulei Dong. A survey on open-set image recognition. *arXiv preprint arXiv:2312.15571*, 2023.

[SXL17] Lei Shu, Hu Xu, and Bing Liu. Doc: Deep open classification of text documents, 2017.