Generalization of MSP based out-of-distribution detection to intermediate convolutional layers

Praveen Annamalai Nathan

August, 2021

TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

Department of Computer Science

This thesis is submitted in fulfillment for the degree of Master of Science

Generalization of MSP based out-of-distribution detection to intermediate convolutional layers

Praveen Annamalai Nathan

1. Reviewer Prof. Dr. Didier Stricker
Augmented Vision
DFKI, Kaiserslautern

2. Reviewer Prof. Dr. Bernt Schiele
Computer Vision and Machine Learning
Max-Planck-Institut für Informatik, Saarbrücken

Supervisors Max Maria Losch

August, 2021

Praveen Annamalai Nathan

Generalization of MSP based out-of-distribution detection to intermediate convolutional layers

This thesis is submitted in fulfillment for the degree of Master of Science, August, 2021

Reviewers: Prof. Dr. Didier Stricker and Prof. Dr. Bernt Schiele

Supervisors: Max Maria Losch

Technical University Kaiserslautern

Department of Computer Science
Erwin-Schrödinger-Straße 52
67663 and Kaiserslautern

Abstract

Deep neural networks have achieved impressive performance on vision tasks. However, they have an overwhelming weakness - they do not have a fail-safe mechanism to avoid catastrophic outputs when confronted with inputs far from the training data distribution. This can result in a false sense of security when deep neural networks are deployed in real-life applications like autonomous driving or medical applications. This necessitates the need for an out-of-distribution detector which can raise the alarm when a neural network encounters samples far from the training in-distribution.

Many traditional approaches use the final layer softmax outputs to predict in- or out-of-distribution, while all intermediate layer outputs remain unused. Recent work has found that it is possible to introduce softmax activation to various intermediate layers of the network without impairing performance. We build upon these results and investigate whether the utilization of softmax outputs at intermediate layers can improve the prediction of out-of-distribution samples.

We find it is possible to apply the softmax activation at nearly every layer in a deep residual network without losing performance and discover that its application enables out-of-distribution detection at the intermediate layer. We extend the current final layer softmax-based out-of-distribution evaluation to softmax outputs at intermediate layers. We can leverage these methods for out-of-distribution detection without any need to train an additional out-distribution detector. Furthermore, we find that early layers of the network are more beneficial to detect out-distribution samples, which are farther to in-distribution, and the last layers of the network for closer out-distribution samples.

Acknowledgement

I have received a great deal of support and assistance throughout my thesis. I would first like to thank my supervisor Max Maria Losch, whose expertise and support was invaluable. His insightful feedback on my methodology and experiments was crucial for structuring the work in this thesis. I would take this opportunity to thank professor Bernt Schiele for reviewing the work and providing guidance to focus on the critical aspects of this thesis. I would also like to thank professor Didier Stricker and his office team at DFKI for timely support at different stages of completing this thesis.

Last but not least, I owe my deepest gratitude to my family and friends for the support and motivation provided to me throughout my master's studies.

Contents

1 Introduction

Deep neural networks (DNNs) have been shown to achieve state-of-the-art results in various domains, including computer vision, natural language processing, and audio processing. This has resulted in DNNs being widely used in various sectors ranging from applied to fundamental research, e.g., physics, healthcare, bioinformatics, finance, autonomous driving cars, and many others. As a result, DNNs are increasingly being used in real-life applications where safety is critical, e.g., in the case of medical diagnosis or autonomous driving [Eyk+18], where making an erroneous decision can have fatal consequences.

In the normal development of a DNN architecture, the performance guarantee is provided by testing with a held-out test data that is analogous to the training data. Although DNNs perform well when trained and tested on data from the same distribution. However, DNNs have an overwhelming weakness. They are vulnerable to making overconfident predictions under dataset shift [NYC15; GSS15; Amo+16; HAB19; Gaw+21]. E.g., see figure 1.1, where a cat-dog image classifier trained with a limited domain of light-colored cats and dark-colored dogs, when encountered with a light-colored dog, predicts as a cat with very high confidence.

This is problematic, especially while deploying these DNNs in real-world applications. The high confidence predictions for data different from the expected input domain, so-called out-of-distribution(OOD) inputs, could lead to performing an inappropriate and dangerous action. Consider the case of an autonomous driving system that deploys an image segmentation network for pixel-wise classification of the scene. Suppose the segmentation network encounters an out-distribution object in the scene and predicts it as a different class with high confidence. In that case, the decision-making system of the car is compromised and could fail to lead to an accident. (see figure 1.2 for an illustration of segmentation results from a methodology from the literature).

To have a fail-safe mechanism for safety-critical applications and to alert the system for manual intervention in the case of unexpected input, it is necessary to have an OOD detector for these DNNs to deploy them in real-world applications. Many methods have been proposed in the literature that aims to mitigate this problem of

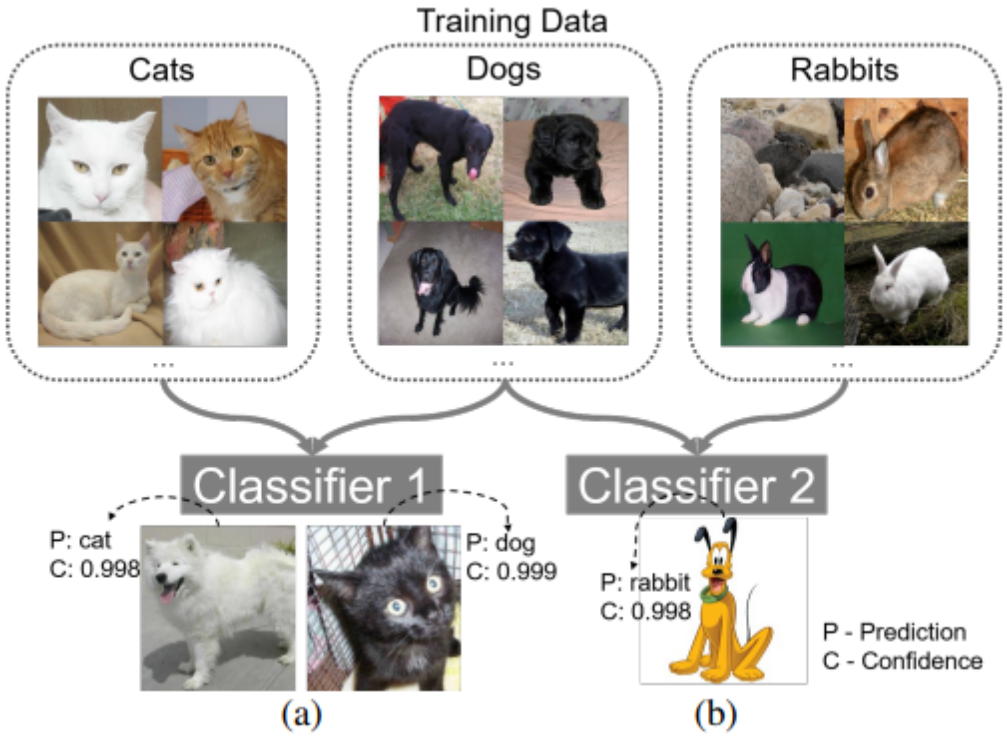


Fig. 1.1.: Classifiers when encountering OOD samples. (a) a white dog and a black cat (OOD samples) are incorrectly predicted with high confidence by a classifier trained on a dataset only containing dark-colored dogs and light-colored cats. (b) When tested with a cartoon dog, a classifier trained on dogs and rabbits gives the wrong prediction as a rabbit with very high confidence. Image taken from [Che+20].

detecting input data far from the training and testing distribution. These methods are collectively known as OOD detection.

In this thesis, we mainly experimented with one sub-problem of computer vision which is image classification. The convolutional neural network (CNN), a deep neural network architecture, has shown that it is tremendously beneficial for image classification tasks. Various methods have been proposed in the literature for OOD detection for image classifiers. In general, these methods can be classified into two categories: (i) post-hoc OOD detection and (ii)OOD-finetuned detection methods. In post-hoc OOD detection, CNN classifiers are trained till convergence only with access to in-distribution data. Then post-training, they devise an OOD scoring technique. In OOD-fine-tuned detection methods, additional auxiliary out-distribution data is used to fine-tune the classifiers, enabling them to learn features to distinguish the in- and out-distribution samples effectively. Many of these fine-tuning methods are an extensible addition to the post-hoc OOD detection methods for which the OOD

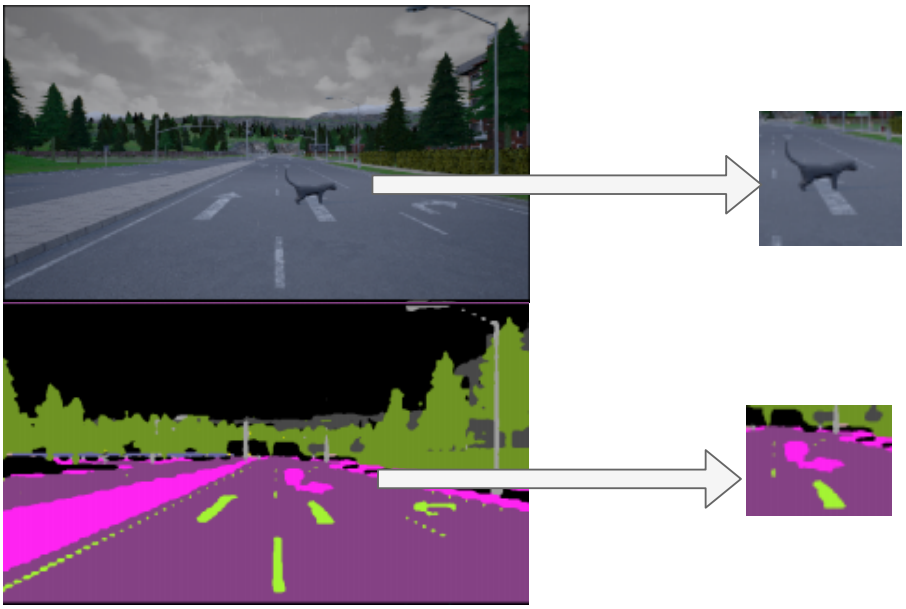


Fig. 1.2.: A sample scene is shown in the top image with an out-distribution animal being part of the image, and the segmentation network trained without any knowledge of the out-distribution data predicts the animal with the same class as the footpath class. This could lead to an accident if an autonomous car with such a decision-making deep neural network is deployed. This illustrates the necessity for an out-of-distribution sample detector. Image from [Hen+20].

scoring technique remains the same. Once the OOD score for an input sample is computed, then based on this score, a threshold-based OOD detector is proposed.

Considering the post-hoc and OOD-finetuned methods, many of these methods build upon softmax probabilities of the final layer of the classifier to devise an OOD score [HG18; LLS20; HMD19]. Most of this method considers the softmax application at the final layer of the network. Often, the intermediate features of the network remain unused. Recent work in [LFS20] has shown that it is possible to introduce softmax activations for intermediate layers of a deep image segmentation network without impairing its performance. Building upon these results, we apply the softmax activation to the intermediate layers of the classifier architecture and study its benefits for OOD detection at the intermediate layer level. Further, instead of using the bounded softmax activation, we use the log-softmax activation, which is unbounded. This is motivated by the work in [GBB11], where they have argued that unbounded activation promotes sparse representation and potentially prevents gradient saturation usually encountered with bounded activations.

We, in this thesis, find that the image classifiers integrated with log-softmax activations after various intermediate layers retain classification performance. Furthermore, log-softmax activation at the intermediate block enables a cheap generalization

of Maximum Softmax Probability(MSP introduced in [HG18]) based OOD scoring method utilized at the output layer to the intermediate layers. We show that this extended version of the MSP method for intermediate layers reaches competitive performance for OOD detection compared to state-of-the-art methods. E.g., our method has competing performance with the Mahalanobis-distance-based method [Lee+18], which requires additional OOD classifier training with features from multiple layers of the network.

Further, we observe that with the generalized MSP evaluation for intermediate layers, different layers enable different OOD detection performances depending on the data distributional shift. We observe that similar images from another distribution(e.g., CIFAR-10 vs. CIFAR-100) require a late layer from the network. In contrast, an image from an entirely different distribution(e.g., CIFAR-10 vs. SVHN) can be detected very efficiently at an early layer already. Contrary to the observation from literature ([HG18],[LLS20]) where MSP of the classifier at the final layer is used as OOD score. We empirically observe that the Minimum Softmax probability(MiSP) score is more suitable as an OOD score than MSP.

The organization of the rest of the thesis is as follows. Chapter 2 discusses some related work and relevant literature from which we have adopted the evaluation for our experiments. Chapter 3 discusses our experimental setup, evaluation methodologies, and the classifier network and the datasets experimented. In chapter 4, we go into details of the experimental results supporting the contributions of this thesis. In chapter 5, we discuss the conclusions.Finally, in chapter 6 we discuss the challenges in adopting our methodology for open world setting and in chapter 7 we discuss the potential future works of this thesis.

2 Related Work

Several approaches are proposed in the literature for OOD detection for image classifiers. This chapter covers some of the methods we can leverage for OOD evaluations for our classifier with intermediate log-softmax activation. The two general categorizations of methods we see from the literature are (i) post-hoc OOD detection methods [HG18; HMD19; LLS20; Lee+18; Win+20; LPB17]. Here, the classifiers are trained with only in-distribution data. Post-training, they devise an OOD sample detection technique that may be as simple as using the softmax probabilities as OOD score or train an additional detector with features from the classifier network. In these methods, the classifiers learn features to discriminate between the classes in the in-distribution dataset. In order to learn features of the out-distribution datasets to discriminate between the in- and out-distribution data samples more effectively, the (ii) OOD-finetuned detection methods were introduced [Liu+21; HMD19; Pap+21; Bev+18; GE19; Moh+20]. These methods use an additional auxiliary out-distribution dataset that differs from the test time out-distribution dataset to fine-tune the classifiers.

Some successful methods from both these categories have proposed adopting the final layer softmax score for deriving an OOD score. In this thesis, our proposed approach integrates the log-softmax activation for the intermediate layers of the classifier network. The OOD evaluations for this log-softmax outputs are derived as an extension of existing final layer softmax-based methods from literature. Here in this chapter, we discuss some softmax-based approaches from the literature. Also, we discuss the Mahalanobis-distance-based method [Lee+18] which uses intermediate features of the classifier to derive an OOD score. Also, it serves as the baseline for intermediate layer OOD evaluation.

In the following sections, the common evaluation metrics used for OOD detector performance in the literature are described.Then in post-hoc OOD detection methods ,considering the softmax score based methods we briefly describe the MSP [HG18] and ODIN methods [LLS20]. Then we introduce the Mahalanobis-distance-based method [Lee+18]. Also, we cover some methods with OOD fine-tuning. Here we briefly introduce some of the successful methods based on softmax scores, Outlier Exposure (OE) [HMD19] and Outlier Exposure with confidence control

(OECC) [Pap+21]. The OOD-finetuned methods are often an extension of the post-hoc methods with additional usage of auxiliary OOD datasets. We select the out-distribution test datasets based on the data distributional shift to the in-distribution. For this, we adopt the OOD dataset spectrum introduced in [Win+20] based on the Confusion Log Probability(CLP) metric, which we briefly cover. Then we discuss the baseline methods selected for the OOD evaluations of our models.

2.1 Evaluation metrics for OOD detection

Several metrics are adopted in the literature to measure the effectiveness of the OOD detector for distinguishing in- and out-distribution samples. Many of the methods in the literature propose OOD detectors by defining an OOD score that they generate. It could be Softmax probability-based [HG18; LLS20; Hsu+20], Mahalanobis distance-based [Lee+18] or log-likelihood based [Ren+19]. These scores generated are continuous outputs, so a threshold needs to be defined for the OOD detector to distinguish in- and out-distribution samples. To measure the performance of threshold-based OOD detectors, we briefly cover some of the most common metrics seen in OOD literature and conclude why we choose AUROC as the metrics we evaluated to measure the performance of the OOD detector in our experiments.

2.1.1 AUROC

Considering the binary decision problem of OOD detection based on an OOD score, the detector labels the samples as positive for in-distribution or negative for out-distribution based on a threshold value. The accuracy of this detector can be measured given a predefined threshold. However, selecting a predefined threshold should again consider the trade-off between false positive and false negative detections allowed for the OOD detector. Consider the figure 2.1 which shows the distribution of log-probability based OOD scores for in- and out-distribution samples. It shows the cut-off for three threshold values for OOD detection where any value above the cut-off is considered in-distribution and lower as out-distribution. Considering threshold 3 allows minimal false positive detection, i.e., the out-distribution sample being classified as in-distribution. However, it allows higher false-negative detections, i.e., in-distribution being classified as out-distribution. In the case of threshold 1, it allows a higher false-positive detection but minimal false negative detections. The threshold value for an OOD detector can be selected based on

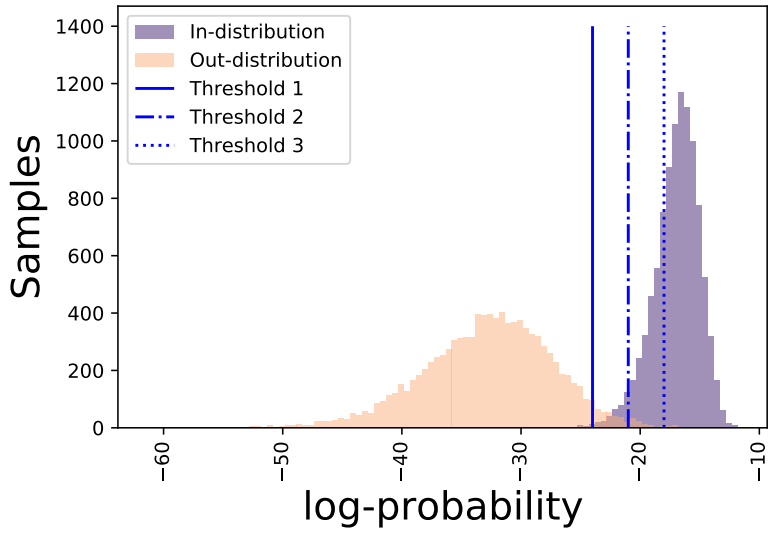


Fig. 2.1.: The figure shows the histogram of log-probability based OOD score distribution for in- and out-distribution samples. In-distribution is considered as positive class and out-distribution as a negative class. It shows three cut-off values for the OOD detection threshold. Any value above the threshold is considered in-distribution and below as out-distribution for the OOD detector. Threshold 1 allows more false positive detections but minimal false negative detections. Threshold 3 allows minimal false positive detections but higher false-negative detection.

this trade-off between false positive and false negative detection allowed for the application where the OOD detector is deployed.

But, a threshold independent metric should be considered to measure the effectiveness of an OOD detection methodology and to compare the detection performance with other methods. The Area Under the Receiver Operating Characteristic curve(AUROC), which is a threshold independent metric ([DG06]), was introduced as the evaluation metric to sidestep the issue of threshold selection to measure the OOD detector performance. The AUROC graph (see figure 2.2) shows the relationship between the True Positive Rate(TPR) of the in-distribution samples and the false positive rate(FPR) of the out-distribution samples measured at all possible thresholds. A higher AUROC value means a better detection rate. For a perfect detector, the AUROC score will be 100, and for a chance detector, 50. This metric is heavily adopted in the OOD literature [HG18; LLS20; Lee+18; Moh+20; Abd+19; SO20; Win+20].

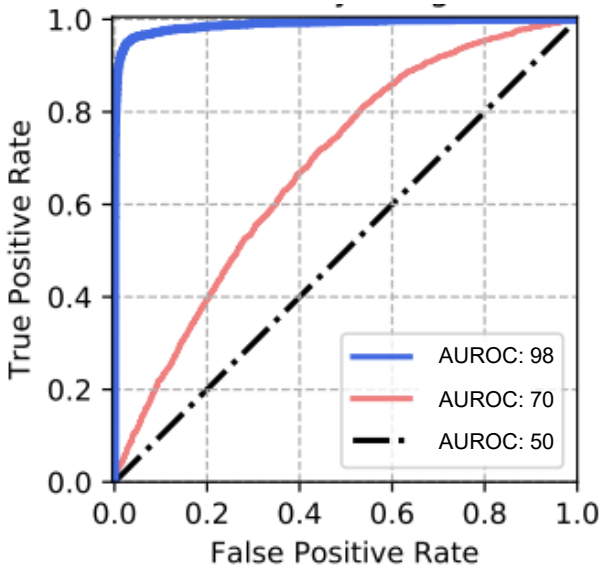


Fig. 2.2.: AUROC is the area under the curve when true positive rate of in-distribution samples is plotted against false positive rate of out-distribution samples at all possible threshold. The line corresponding to AUROC 50 is the chance detector. The curve corresponding to AUROC 98 can be considered as an excellent detector. Image taken from [HMD19].

2.1.2 FPR at 95% TPR

This metric measures one instance of the AUROC curve. This metric can be used to make a comparison between the detectors strictly at one threshold value. When the False positive rate(FPR), i.e., the probability that the out-of-distribution sample (negative sample) is misclassified as in-distribution (positive) when the threshold is such that the true positive rate of the in-distribution is 95%. Another metric that is measured when the TPR threshold is 95% is the detection error. It measures the percentage of in-distribution samples classified as out-distribution and out-distribution samples being classified as in-distribution.

For the empirical results presented in this thesis, the metric reported for all the OOD evaluations is the AUROC metric. Since it is threshold-independent, it gives an unbiased evaluation of the OOD detector.

2.2 OOD Methods without fine-tuning

2.2.1 Maximum Softmax Probability(MSP)

This method [HG18] proposes to use the final layer maximum Softmax probability of a pre-trained classifier as an OOD detection score. Though it has been noticed in the literature that OOD samples are assigned high prediction probabilities, here the authors point out that these scores are still lower than for correctly classified samples from in-distribution datasets. Based on this maximum softmax score, they define a threshold-based OOD detector. This method is a hyperparameter-free method applied for OOD detection without any knowledge of out-distribution data. This method serves as a simple baseline for softmax-based methods. The maximum softmax score distributions for an in-distribution dataset and out-distribution dataset is shown in figure 2.5 (a).

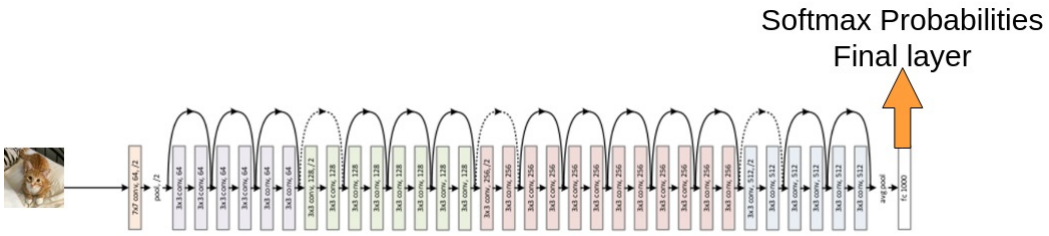


Fig. 2.3.: MSP evaluation is performed on the probability values in the final softmax layer of the classifier.

2.2.2 ODIN

This method [LLS20] is an improved version of MSP. Here, post-training the classifiers, they apply softmax temperature scaling and input preprocessing techniques. Then the maximum softmax probability is used as an OOD score. Empirically they observe that the max softmax probabilities of in-distribution images and out-distribution images are more separable after applying these post-training steps.

Temperature Scaling : For a pre-trained classifier the logits of each class is scaled with a temperature parameter $T \in R^+$ before applying the softmax. For a neural network $\mathbf{f} = (f_1, ..., f_N)$ which is trained to classify N classes. The softmax output for input $\mathbf{x}$ with temperature scaling for i^{th} class is given by 2.1.

$$S_i(\mathbf{x};T) = \frac{exp(f_i(\mathbf{x})/T)}{\sum_{j=1}^{N} exp(f_j(\mathbf{x})/T)} \tag{2.1}$$

The $max_i S_i(\mathbf{x};T)$, which is the maximum softmax probability among all the classes, is the softmax score that is used to define the threshold-based detector for in-distribution and out-distribution samples.

Input Preprocessing: Inspired by the idea of generating adversarial examples by adding small perturbations to decrease the softmax score for the true label, thereby forcing the network to make an incorrect prediction from [GSS15]. Here they add small perturbations to increase the softmax score. For an input image, $\mathbf{x}$ the perturbations are computed by back-propagating the gradient of the cross-entropy loss w.r.t the input. The computation is given in the equation 2.2 where ϵ is the perturbation magnitude. Empirically they have noted that after performing input processing, for an in-distribution sample, the maximum softmax score is higher than that for an out-distribution sample, thus making their distributions more separable. For the input processing technique, two passes of the sample images are required through the network, first, for computing the gradient to be added as the perturbation to the input image and second pass for computing the maximum softmax score passing the perturbed image $\widetilde{\mathbf{x}}$ as computed in equation 2.2.

$$\widetilde{\mathbf{x}} = \mathbf{x} - \epsilon sign(-\nabla_x log(max_i S_i(\mathbf{x};T)) \tag{2.2}$$

The temperature parameter T and the perturbation magnitude ϵ are hyper-parameters tuned with an OOD dataset different from test time OOD datasets. The effectiveness of applying the temperature scaling and input preprocessing of the ODIN method over the MSP method can be seen in table 2.1 where it is observed that the ODIN method has performance improvements over the MSP method.

2.2.3 Mahalanobis distance-based.

Improving on the MSP, ODIN method which proposed to use the classifier softmax score, in this method they proposed using the pre-trained features of the penultimate layer of the classifier for effective OOD detection [Lee+18]. They assume that a class conditional Gaussian distribution can be fitted with these pre-trained features. To estimate the class-wise Gaussian distribution parameters, first, they compute the class-wise mean of the penultimate features given by equation 2.3. Next, they assume all the classes have a tied covariance matrix computed as in equation 2.4.

Then for the OOD score, the minimum Mahalanobis distance for a test sample to a class-conditional Gaussian distribution is used.

Given N training image, label pairs $\{(x_1, y_1)...(x_N, y_N)\}$,where y_i represents the class label $c \in 1, ..., C$, where C is the number of classes in the training dataset. $\hat{\mu_c} \in \mathcal{R}^n$ where $n > 1$ is the mean of the multivariate Gaussian distribution which is given by equation 2.3. Where N_c is the number of training sample with class label c and $f(.)$ denote the penultimate layer of the neural network. The tied covariance matrix is given in equation 2.4.

$$\hat{\mu}_c = \frac{1}{N_c} \sum_{i=1}^{N_c} f\left(x_i\right) \tag{2.3}$$

$$\hat{\Sigma} = \frac{1}{N} \sum_{c} \sum_{i=1}^{N_c} \left(f\left(x_i\right) - \hat{\mu}_c\right) \left(f\left(x_i\right) - \hat{\mu}_c\right)^T \tag{2.4}$$

For generating an OOD score, the Mahalanobis distance is computed as in equation 2.5. Furthermore, the Mahalanobis distance to the closest class is used as the OOD score.

$$M(x) = \max_{c} - \left(f(x) - \hat{\mu}_c\right)^T \hat{\Sigma}^{-1} \left(f(x) - \hat{\mu}_c\right) \tag{2.5}$$

Additionally, to improve results, they propose to use two techniques. First, similar to the input preprocessing introduced in ODIN (section 2.2.2), they also add small controlled noise to the test sample to improve the Mahalanobis score distribution separability between in-distribution and out-distribution samples. The computation for the noise calculation is shown in equation 2.6. Here, ϵ is the magnitude of noise, and $\hat{c}$ is the index of the closest class for the input sample when computing the Mahalanobis distance to all classes. Here they add the noise to the input to further reduce the Mahalanobis distance to the closest class. Thus increasing the Mahalanobis score computed as in equation 2.5.

$$\hat{\mathbf{x}} = \mathbf{x} - \varepsilon sign\left(\nabla_{\mathbf{x}} \left(f(\mathbf{x}) - \widehat{\mu_{\widehat{c}}}\right)^{\top} \widehat{\Sigma}^{-1} \left(f(\mathbf{x}) - \widehat{\mu_{\widehat{c}}}\right)\right) \tag{2.6}$$

Another technique they introduce to improve the OOD detection is to use low-level features from multiple intermediate blocks of the network. For this, they extract features from multiple intermediate blocks. In their experiments, they choose the intermediate layers at the end of Residual or Dense blocks based on the architecture.

In order to reduce the number of features from the intermediate feature maps, they propose to perform average pooling. Then for each of them, compute the Mahalanobis score individually as introduced for the penultimate layer. Then these scores from multiple layers are integrated to produce a single OOD score by doing a weighted average. The weights for Mahalanobis score from multiple layers are determined by training a logistic regression detector. This prevents degradation of overall performance even if some layers are not very effective.

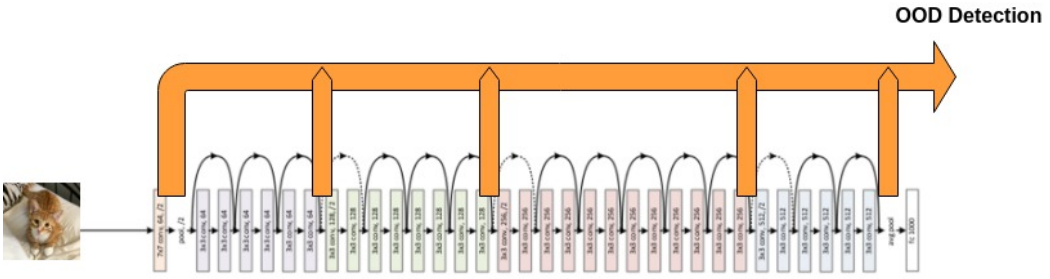


Fig. 2.4.: The figure shows the layers from which the pre-trained features are used to derive the Mahalanobis-based OOD score. Features are used from the first convolutional layer and the last layer of all residual blocks for a residual connection-based architecture.

The Mahalanobis method with both the input preprocessing and ensemble features from multiple layers offered the best results. This method needs knowledge of out-distribution data to train the logistic regression detector and set the amount of noise added to the input sample. Again, similar to the ODIN input processing this method also needs two passes for the image samples through the network. The performance improvements of the Mahalanobis distance-based OOD technique with additional improvements can be seen in table 2.1.

2.3 OOD Methods with fine-tuning

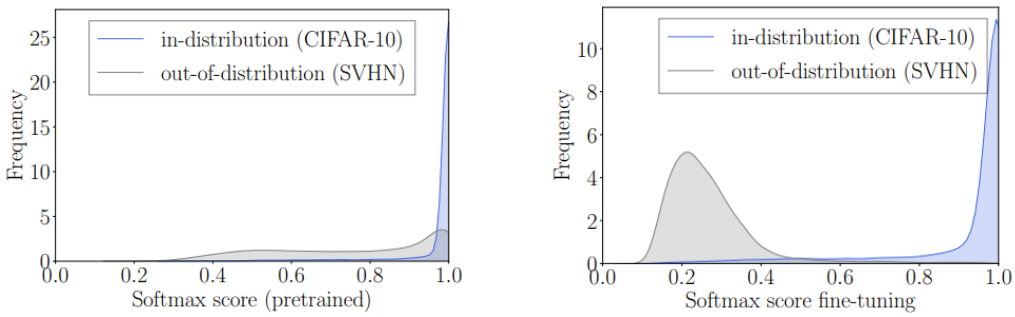
2.3.1 Outlier Exposure

This method introduces a complementary way to improve existing out-of-distribution detection methods. They achieve this by exposing the network to additional outlier datasets different from test time OOD datasets. With this outlier exposure, the network learns effective heuristics for detecting OOD samples. They implement the outlier exposure by introducing an additional optimization term in the learning objective function. E.g., they extend the MSP method by fine-tuning with the

outlier dataset by forcing its predictions to be a uniform distribution. For this, they introduce an additional cross-entropy loss term for the outlier predictions and uniform distribution. The benefits of this fine-tuning with the outlier dataset can be seen in the softmax score distribution of in-distribution and out-distribution datasets as shown in figure 2.5. The figure in 2.5(a) is the MSP distribution before applying outlier exposure, and in 2.5(b) is the MSP distribution after applying outlier exposure where the in- and out-distribution MSP scores are more separable.

In-dist (model)	OOD dataset	Validation on OOD samples AUROC(↑) MSP/ODIN/Mahalanobis
CIFAR-10 (ResNet-34)	SVHN	89.9/96.7/**99.1**
	TinyImageNet	91.0/94.0/**99.5**
	LSUN	91.0/94.1/**99.7**
CIFAR-100 (ResNet-34)	SVHN	79.5/93.9/**98.4**
	TinyImageNet	77.2/87.6/**98.2**
	LSUN	75.8/85.6/**98.2**

Tab. 2.1.: Table showing the comparison of the OOD evaluation for different OOD methods discussed. It is observed that the Mahalanobis distance metric performs the best in comparison to other techniques. Comparing MSP and ODIN, we see that the temperature scaling and input perturbation applied in ODIN improve the baseline MSP method. Table taken from [Lee+18]. The reported metric is AUROC, ↑ indicates higher the value, the better.



(a) (b)

Fig. 2.5.: Figure (a) shows the maximum softmax probability score distribution of in-distribution CIFAR-10 [Kri09] and out-distribution SVHN[Net+11] when a pre-trained classifier is used without fine-tuning. Figure (b) shows the improved separation in the distribution of in- and out-distribution softmax scores after fine-tuning the classifier with a held-out out-distribution dataset. Image taken from [Liu+21].

2.3.2 Outlier Exposure with Confidence control

In addition to the regularization term introduced for the outlier datasets to make their predictions uniform distribution in the Outlier Exposure method. In this method, they introduce another regularization term for confidence calibration [Guo+17] of the classifier network. This makes sure that the classifiers predict with average confidence close to the classification accuracy for the in-distribution dataset. They have empirically shown that adding this regularizer for confidence calibration has improved OOD performance over the Outlier Exposure method.

2.4 Confusion Log Probability(CLP)

We adopt the in- and out- distribution datasets for OOD evaluations for our classifiers based on the data distributional shift between in- and out-distribution datasets which was quantitatively measured in [Win+20] with the confusion log probability(CLP) metric. With this we can evaluate whether introducing the log-softmax activation for intermediate layers have benefits for OOD detection based on dataset distribution shift between in- and out- distribution.

CLP is based on the likelihood that a classifier that has access to out-distribution samples during training may confuse out-distribution with in-distribution. Consider the datasets $\mathcal{D}_{in}$ of in-distribution with label set $\mathcal{C}_{in}$ and out-distribution dataset $\mathcal{D}_{out}$ with label set $\mathcal{C}_{out}$. For the joint dataset $\mathcal{D} = \mathcal{D}_{in} \cup \mathcal{D}_{out}$ and for extended label set $\mathcal{C} = \mathcal{C}_{in} \cup \mathcal{C}_{out}$, first an ensemble of N_e classifiers $\{\hat{p}^j\}_{j=1}^{N_e}$ are trained. Given the N_e classifiers, the expected probability of a test sample $\mathbf{x}$ to be predicted as class k is given by equation 2.7. With this equation the estimate of confusion between different classes for an input sample can be computed. Then for a held out test out-distribution dataset $\mathcal{D}_{test}$, the measure, log of estimate of confusion of these test OOD samples with inlier classes $\mathcal{C}_{in}$ is the confusion log probability, which is given by equation by 2.8.

$$c_k(\mathbf{x}) = \frac{1}{N_e} \sum_{j=1}^{N_e} \hat{p}^j(\hat{y} = k \mid \mathbf{x}) \tag{2.7}$$

$$\mathrm{CLP}_{\mathcal{C}_{in}}(\mathcal{D}_{test}) = \log \left(\frac{1}{|\mathcal{D}_{test}|} \sum_{\mathbf{x} \in \mathcal{D}_{lest}} \sum_{k \in \mathcal{C}_{in}} c_k(\mathbf{x}) \right) \tag{2.8}$$

	Near OOD CLP= [-4.5 to -2.6]	Near and Far OOD CLP=[-7.4 to -0.8]	Far OOD CLP=[-12.1 to -7.6]
In-distribution	CIFAR100	CIFAR10	CIFAR10
Out-distribution	CIFAR10	CIFAR100	SVHN

Tab. 2.2.: The table shows the in- and out-distribution dataset configuration for different OOD detection difficulty categories as introduced in [Win+20]. The categorization is based on the CLP metric. Here they deem the 'Near OOD' setting to be the most challenging for OOD detectors.

With this confusion estimate for out-distribution samples, the CLP captures the similarity of in- and out-distribution datasets to assess the difficulty of the OOD detection task. Based on equation 2.8, a low CLP score indicates that the test out-distribution samples are far out-distribution datasets, i.e., the out-distribution samples have less similar features to confuse as in-distribution, and a high CLP indicates that they are near out-distribution datasets. Based on this measure, they introduced the categorization of out-distribution datasets as 'near,' 'near and far,' and 'far.' The in- and out-distribution datasets for these categories are shown in table 2.2. According to this categorization, the 'near' OOD is the most challenging setting for an OOD detector.

2.5 Choosing the baselines for our experiments.

The focus of the experiments in this thesis was to study whether log-softmax outputs from intermediate layers can be utilized for OOD detection. In the previous sections many softmax based and method which leverages intermediate features were introduced. Here in this section, we discuss the suitable methods selected for our experiments.

Broadly categorizing our experiments, we perform two types of OOD evaluations for our models. (i) Test how beneficial are the log-softmax integrated models when evaluated with existing OOD evaluation techniques. (ii) Test how well each intermediate layer enables OOD detection for log-softmax integrated models. The log-softmax model OOD evaluations are compared with OOD evaluation for a baseline architecture without log-softmax activation for both cases.

For (i), we adapt the MSP method as introduced in the literature directly since it serves as simple to apply baseline technique. Also, it could help understand whether introducing the intermediate layer log-softmax activation is beneficial for the final layer softmax-based MSP method. Also, we evaluate our models with the Mahalanobis-distance-based method as proposed in the literature with input

preprocessing and ensemble of features from intermediate layers. This method could help understand whether the log-softmax models have intermediate features that have the competitive performance to baseline for OOD detection. Considering the techniques without further OOD fine-tuning, the Mahalanobis-distance method has competitive performance. See table 2.3 for the comparison between some of the post-hoc OOD methods and OOD fine-tuned methods. Note that for some of the methods, the values are -NA-(Not Available). This is primarily due to inconsistent usage of network architecture and combinations of in- and out-distribution datasets by the methods in the literature.

For (ii), very few works from the literature study the benefits of individual intermediate layer features for OOD evaluation. So, a suitable baseline has to be selected for the intermediate layer performance comparison. We again consider the Mahalanobis-distance-based method evaluated per layer as a baseline OOD evaluation for the baseline model without log-softmax activation. Then for the log-softmax models, we propose a cheap generalization of the MSP method applied at the final softmax layer to the intermediate layer log-softmax outputs to derive an OOD score.

Further, the ODIN method and other fine-tuning methods like Outlier Exposure can be applied to improve the MSP-based OOD detection method. The benefits of OOD fine-tuning methods are illustrated in table 2.3. E.g., for WideResNet as the network architecture trained on in-distribution dataset CIFAR10 and for out-distribution dataset LSUN-resize [Yu+16] the MSP method has an AUROC of 91.37. However, the same setting with Outlier Exposure fine-tuning has an AUROC of 98.94.

Network Achitecture (In-Dist: CIFAR10)	OOD dataset	AUROC(↑)					
		MSP	ODIN	Mahalanobis	Energy Based	MSP+ Outlier Exposure (Fine tuning)	Energy Based (Fine tuning)
WideResNet	SVHN	91.89[1]	91.96[1]	97.62[1]*	90.96[1]	98.6[1]	**99.41**[1]
	LSUN-crop	95.65[1]	97.04[1]	94.15[1]*	98.35[1]	**99.49**[1]	99.32[1]
	LSUN-resize	91.37[1]	94.57[1]	93.23[1]*	94.24[1]	98.94[1]	**99.39**[1]
ResNet-34	SVHN	89.9[2]	96.7[2]	**99.1**[2]	-NA-	-NA-	-NA-
	LSUN-crop	-NA-	99.2[3]	**99.3**[3]	-NA-	-NA-	-NA-
	LSUN-resize	91.0[2]	94.1[2]	**99.7**[2]	-NA-	-NA-	-NA-

Tab. 2.3.: Table comparing the results of different methodologies for OOD detection. MSP,ODIN,Mahalanobis and Energy based [Liu+21] are post-hoc OOD methods and the rest are OOD fine tuned methods. -NA- indicates the value is not available this is due to the configuration not being tested for the respective methodology in the literature. The following are the literature correspondence: [1] [Liu+21], [2] [Lee+18],[3] [Hsu+20]. $*$ - Mahalanobis score calculated using only features of penultimate layer.

3 Methods and Experimental Setup

The primary focus of this thesis is to explore how the intermediate features of the CNN classifier can be used for OOD detection. For which we propose the introduction of log-softmax activation at intermediate layers of the network. Furthermore, we propose extending existing softmax-based OOD evaluation techniques introduced at the final classification layer to the intermediate layers evaluations. This chapter introduces the experimental setup we follow to verify the intermediate log-softmax benefits for OOD detection.

First, we explain how the log-softmax activations are introduced in the classifier architecture. Further, one of the benefits of the log-softmax activation at the intermediate layers is the possibility of applying entropy regularization. This is mainly motivated by the cross-entropy loss applied at the final layer for the softmax scores and the class labels. We propose a similar entropy minimization for the intermediate layer log-softmax outputs. Then this chapter covers how we apply the existing OOD evaluations for our models. Then we introduce the newly developed OOD detection methods, the Minimum Softmax Probability(MiSP) method, and the generalization of the MSP method for intermediate log-softmax outputs.

Then in the experimental setup section, we briefly cover the classifier architecture experimented with and details of the datasets we test as in-distribution and out-distribution.

3.1 Log-softmax for layer-wise OOD detection

In the work of [LFS20] in order to construct inherently interpretable models, one of the critical architectural changes adopted was the softmax activation introduction after multiple layers of an image segmentation network. Moreover, they have successfully shown that it is possible to have softmax activation after multiple intermediate layers without impairing the segmentation performance. Methods like MSP, ODIN from OOD literature build upon the final layer softmax probabilities

to derive an OOD score. Further, many methods like Outlier Exposure show that this softmax score distribution is further separable between in- and out-distribution with appropriate fine-tuning with auxiliary OOD datasets. In [Hen+20] they show the utility of extending the MSP method for pixel-wise OOD detection for an image segmentation use-case. However, we often observe that the softmax-based methods from the literature are developed based on the final classification layer softmax probabilities. The intermediate features of the network mostly remain unused for OOD detection.

Introducing the log-softmax activation for intermediate layers.

To utilize the intermediate features of the network, we, in this work, propose to extend the softmax applicability to intermediate layers as introduced in [LFS20] to the classifier architecture. This modification for the classifier architecture helps to verify whether any intermediate layer softmax outputs are beneficial for OOD sample detection. Further, the work in [GBB11] has discussed the suitability of using an unbounded activation for training the neural networks. They have argued that unbounded activations promote sparse representation and potentially prevents gradient saturation. The bounded softmax activation produces outputs in the range $[0, 1]$. In order to introduce an unbounded activation for our classifiers, we propose to use the log-softmax activation for which the outputs are in the range $[-\infty, 0]$.

Generalization of log-softmax activation

In [LFS20] their work, the log-softmax activation was introduced only after certain layers of the network. We, in our work, propose a generalized introduction of the log-softmax activation for the classifier architectures. Standard state-of-the-art networks based on residual connections or dense connections-based architectures relied on ReLU [NH10] as the intermediate layer non-linear activation. We, in our work, for a residual connection-based architecture(e.g., ResNet-34 [He+16]), we introduce the log-softmax activation for the residual layers after the residual skip connection replacing the ReLU activation. The log-softmax activation introduction for a single residual layer is depicted in figure 3.1. Considering an intermediate layer feature map after convolutional layer of shape $H \times W \times C$ where H is the height of the feature map,W its width, and C is the number of channels. We apply the log-softmax activation along the channel depth C for each spatial location in $H \times W$. So, in effect, the outputs of the log-softmax activations are also of the shape $H \times W \times C$.

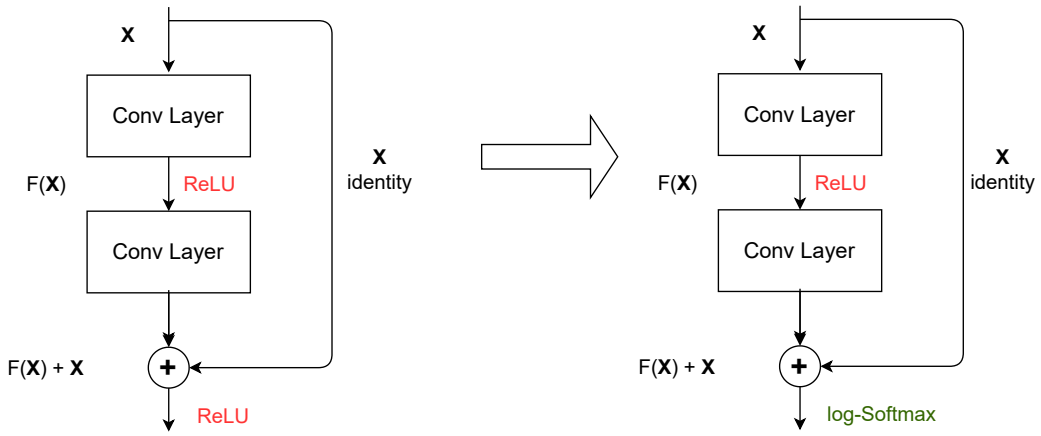


Fig. 3.1.: The figure on the left is one single residual layer as introduced in [He+16] with the ReLU activation. As shown in the right figure, we introduce the log-softmax activation replacing the ReLU activation after the residual skip connection.

Entropy Minimization

Cross entropy loss function has been very crucial for training a classifier that predicts softmax probabilities as outputs. Given a true label distribution and softmax predictions distribution of a classifier, the cross-entropy loss reduces the entropy on the prediction softmax distribution. It pushes the predicted softmax probability corresponding to the true label towards the maximum during the training process. This is one of the aspects leading to the development of the Maximum Softmax Probability(MSP) based method for OOD detection. The formulation of the cross-entropy loss for a M class classification is given in equation 3.1. Where y_c is the truth label for class c and $\hat{y}_c$ is the softmax probability predicted by the classifier for class c.

$$CE_{loss} = -\sum_{c=1}^{M} (y_c \cdot \log \hat{y}_c) \tag{3.1}$$

We, in our experiments proposed the introduction of intermediate log-softmax activation and have proposed an extension of MSP-based evaluation for these intermediate log-softmax outputs(see section 3.2.4 for the MSP generalization for intermediate layers). This log-softmax intermediate output distribution can also be subjected to such an entropy minimization as cross-entropy loss does for the classifier's final softmax outputs, which was beneficial for the MSP method. However, these log-softmax intermediate outputs are multidimensional. Since we apply the log-softmax along the channel depth, this depth might not directly correlate with the number of classes in the training distribution. Thus, restricting the direct usability of true label distribution for introducing cross-entropy loss for each spatial location. To this

extent, to achieve a similar effect as the cross-entropy loss, we propose introducing entropy minimization for this intermediate log-softmax outputs during the classifier's training by introducing an additional loss. For layers with log-Softmax activation, the entropy of the outputs calculated per spatial location is our loss. This loss is summed over all spatial locations at the given layer and again summed over all the layers with log-softmax outputs. This final loss is jointly optimized with the classification loss of the classifier.

The final loss $\mathcal{L}$ formulation for our log-softmax integrated classifier with the additional term for intermediate layer entropy minimization is shown in equation 3.2. The first term is the cross-entropy loss applied at the final layer, and the second term is the entropy loss introduced for the intermediate log-softmax layers. Here, N is the number of intermediate layers for which the log-softmax activation is introduced.$H_i \times W_i \times C_i$ is the log-softmax output dimension at layer i. x_{ijk} , is the input value , and $log(s(x_{ijk}))$ is its corresponding log-softmax output for the value at layer i, in j^{th} position of the feature map of dimension $H_i \times W_i$ and in k^{th} position along the channel of size C_i. To jointly minimize entropy for log-Softmax activation outputs and to retain the classification performance at the same time, we experiment adding the entropy loss of the layers with different entropy loss factors λ(in equation 3.2) to the total loss.

$$\mathcal{L} = CE_{loss} + \lambda \sum_{i=1}^{N} \sum_{j=1}^{H_i \times W_i} \sum_{k=1}^{C_i} -s(x_{ijk}) log(s(x_{ijk})) \tag{3.2}$$

3.2 How do we evaluate OOD?

In this section, we discuss the details of the OOD evaluations performed for our experiments. In order to have a baseline to compare the benefits of the OOD evaluations performed for the log-softmax models, the OOD evaluations are performed on a standard residual architecture without log-softmax activation or entropy minimization. For each OOD method described in this section, we discuss its applicability for the baseline and log-softmax models.

We discuss the MSP and Mahalanobis-method adoption for our experiments. Then we discuss the newly introduced Minimum Softmax Probability(MiSP) based OOD detection method for the final softmax layer. Then we introduce the MSP generalization we propose for deriving the OOD score for the log-softmax intermediate layer outputs.

3.2.1 Maximum Softmax Probability(MSP)

Exactly as the Maximum Softmax probability introduced in section 2.2.1, we perform the OOD evaluations with the Maximum Softmax probability score of the classifier as the OOD score. This evaluation is performed for the baseline model and the models with log-softmax activation with different factors of entropy minimization. This experiment is performed to verify how the log-softmax models perform compared to the baseline model for an existing softmax-based evaluation technique from the literature. Also, we can evaluate whether introducing intermediate log-softmax activations have any effect on the final softmax-based OOD scoring. MSP serves as a baseline for softmax-based OOD evaluation methods, which is further extensible with ODIN and Outlier Exposure methods. We measure the performance of the models based on the AUROC metric of the softmax-score-based OOD detector.

3.2.2 Minimum Softmax Probability(MiSP) and Average Softmax Probability(ASP)

In [HG18; LLS20] explored the usage of maximum softmax probability as the OOD score. This choice was intuitive as the classifier networks are trained to produce maximum softmax probability for the true class label with cross-entropy loss. From the literature related to softmax-based methods, we see that the lower softmax spectrum is often ignored. In order to test the benefits of the lower softmax value spectrum for OOD detection, we, in this work, explore the Minimum Softmax Probability(MiSP) of the final softmax layer of the classifier as an OOD score.

We define the MiSP score as follows, for a neural network $\mathbf{f} = (f_1, ..., f_N)$ which is trained to classify N classes. The softmax output for input $\mathbf{x}$ for i^{th} class is given by equation 3.3. The MiSP OOD score is defined by equation 3.4 which is the minimum softmax probability among the N classes.We perform this experiment for the baseline model as well as the models with log-Softmax activation. The performance of the OOD detector with MiSP as the scoring technique is measured with the AUROC metric.

Further, another possibility for the OOD scoring with the softmax probabilities is the Average Softmax Probability(ASP), which we define as the average of softmax probabilities of all the classes. We define the ASP OOD score for an N class classifier as given in 3.5. Since the sum of softmax probabilities at the final layer is 1 and for an N class classifier, the ASP always results in a constant. We do not apply this method for the final softmax layer. However, this can be applied for the intermediate

log-softmax outputs since its sum is not constant. We discuss how the ASP is applied for intermediate log-softmax outputs in section 3.2.4.

$$S_i(\mathbf{x}) = \frac{exp(f_i(\mathbf{x}))}{\sum_{j=1}^{N} exp(f_j(\mathbf{x}))} \tag{3.3}$$

$$MiSP_{score} = min_i S_i(\mathbf{x}) \tag{3.4}$$

$$ASP_{score} = \frac{1}{N} \sum_{i=1}^{N} S_i(\mathbf{x}) \tag{3.5}$$

3.2.3 Mahalanobis distance-based

Similar to the Mahalanobis distance-based evaluation introduced in section 2.2.3. Where feature ensemble from intermediate blocks and input pre-processing was proposed. For feature ensemble, the features from the end of residual blocks and the first convolutional layers were used for a residual skip connection-based architecture. We perform the same evaluation on our baseline and log-softmax models. The figure 2.4 represents from which layers the features are used for Mahalanobis-based OOD detection. We consider the Mahalanobis-distance-based evaluation since it leverages the intermediate features to derive an OOD score. We apply this method for the log-softmax models and the baseline. This method helps to evaluate how well the intermediate features are beneficial for OOD detection for the log-softmax-based models compared to baseline.

3.2.4 Intermediate layer OOD evaluation

To evaluate whether any intermediate layer with log-softmax activations enables OOD detection, we propose a post-hoc OOD detection method to extend the final layer softmax-based methods from the literature. To compare OOD detection for the newly developed softmax-based methods for intermediate outputs, we need to define a baseline OOD performance for the intermediate layer outputs. Since the Mahalanobis-distance-based method proposed using an ensemble of intermediate outputs, we propose performing the Mahalanobis-distance method per intermediate layer as the baseline OOD performance. This layerwise Mahalanobis-distance method is performed for the baseline architecture without log-softmax activation. This

evaluation is considered for performance comparison for the newly developed softmax-based extension for intermediate layer OOD evaluation.

MSP generalization for intermediate layer OOD evaluation

In order to generate an OOD score for this intermediate log-softmax outputs, we propose to extend the MSP method, which used the final maximum softmax score of the classifier to derive an OOD score. However, the MSP at the final layer was computed on 1-dimensional softmax probability scores corresponding to different classes of the dataset with which the network was trained. Here, the intermediate log-softmax activation produces 3-dimensional outputs corresponding to the intermediate feature map's channel, height, and width. (more details of output shape after each layer can be found in table 3.1). We propose an extension of the MSP evaluation for the intermediate layers by aggregating the log-softmax outputs for these intermediate feature outputs.

To obtain an OOD score for the 3-dimensional log-softmax outputs, we aggregate the outputs as represented in figure 3.2. First, considering an intermediate layer output of shape $H \times W \times C$ where H is the height of the output feature map,W its width, and C is the number of channels,. For each spatial location, the channel reduction is performed. For this, we compute the maximum, minimum, or average along the channel depth and obtain a 2-dimensional output of shape $H \times W$. Then to obtain a single scalar OOD score further, these 2-dimensional results are aggregated by taking maximum, minimum, or the average of these values. Post-training the classifier, these methods introduced above are performed for the in- and out-distribution test samples. The benefit of using this aggregation is that we can use this as an OOD score for the intermediate layers with log-softmax outputs, which is cheap to evaluate and does not introduce any hyperparameters or the need to train an additional OOD detector.

Additionally, before aggregating the results after the channel reduction with maximum/minimum/average of log-softmax along the channel, these values can be leveraged to evaluate pixel-wise OOD detection. This is equivalent to performing MSP/MiSP/ASP per pixel for the intermediate log-softmax outputs. We performed these experiments for the intermediate layers log-softmax outputs, and since we do not make a significant observation, these results are presented in the appendix section A.3.

In effect, we introduce nine types of evaluations for the intermediate layers. Below we introduce the naming convention we adopt for these nine evaluations.

- Maximum Softmax Probability for Intermediate layer.

  - max/min/avg of MSP - Maximum/Minimum/Average of Maximum Softmax Probability

- Minimum Softmax Probability for Intermediate layer.
  - max/min/avg of MiSP - Maximum/Minimum/Average of Minimum Softmax Probability

- Average Softmax Probability for Intermediate layer.
  - max/min/avg of ASP - Maximum/Minimum/Average of Average Softmax Probability

We perform all these nine evaluations for models only with log-softmax activations. Since we derive these evaluations based on the log-softmax scores, we do not evaluate the baseline classifier architecture with the ReLU activation.

To summarize our evaluations, to test the performance of log-softmax integrated models with the baseline model, we adopt the MSP and Mahalanobis-distance-based methods as introduced in the literature. Additionally, we introduce the MiSP evaluation. For a baseline for the intermediate layerwise OOD evaluations, we perform the Mahalanobis-distance-based evaluation for these intermediate layers individually. For the intermediate layer evaluations of the log-softmax models, we introduce the nine aggregation evaluation as an extension of MSP for the log-softmax intermediate outputs.

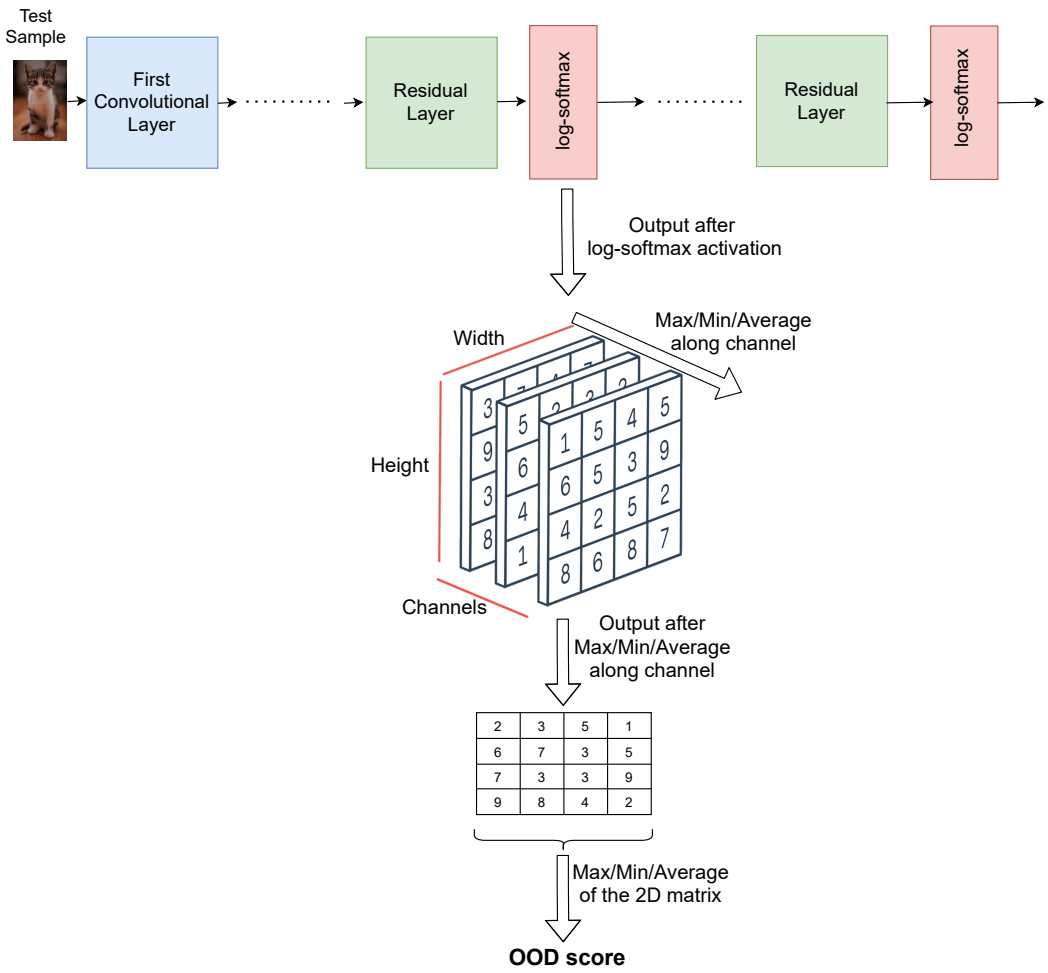


Fig. 3.2.: The figure shows how we generalize the MSP method to derive an OOD score from the classifier's intermediate layer log-softmax outputs. The aggregation methods are performed for in- and out-distribution test datasets post-training the classifiers with log-softmax activation.

3.3 Experimental Setup

3.3.1 Classifier Architecture

As already proposed in section 3.1 the generalized introduction of the log-softmax activation for the intermediate layers of the classifier architecture. We, for our experiments, choose the residual skip connection-based architecture, especially the ResNet-34 architecture [He+16]. The summary of the architecture is described in table 3.1. It provides more details on the convolutional kernel size, the number of kernels at each layer, and the output shape after each residual layer.

Layer Name	ResNet-34	Output
Conv0	3×3 conv, 64, stride=2 3×3 max pool, stride=2	32×32×64
ResidualBlock1_x	$\begin{bmatrix} 3\times3 \text{ conv, } 64 \\ 3\times3 \text{ conv, } 64 \end{bmatrix} \times 3$	32×32×64
ResidualBlock2_x	$\begin{bmatrix} 3\times3 \text{ conv, } 128 \\ 3\times3 \text{ conv, } 128 \end{bmatrix} \times 4$	16×16×128
ResidualBlock3_x	$\begin{bmatrix} 3\times3 \text{ conv, } 256 \\ 3\times3 \text{ conv, } 256 \end{bmatrix} \times 6$	8×8×256
ResidualBlock4_x	$\begin{bmatrix} 3\times3 \text{ conv, } 512 \\ 3\times3 \text{ conv, } 512 \end{bmatrix} \times 3$	4×4×512
Average pool, FC layer , Softmax		

Tab. 3.1.: Table describing the architecture of the ResNet-34 classifier. In the brackets under the ResNet-34 column are the building blocks showing kernel size and the number of kernels in the residual layers and the multiplied number shows the number of residual layers stacked in that residual block. The output column shows the output sizes after each convolutional block for an input image with height ×width ×channel as 32 ×32 ×3.

The ResNet-34 network architecture takes the input image having height, width as multiples of 32, and 3 as channel width. First, the images are passed through a convolutional layer(Conv0 in table 3.1), then through four residual Blocks with [3,4,6,3] residual layers with skip connection in the respective blocks. At the end of each block is a downsampling layer. After the residual blocks, the network has an Average Pooling layer followed by a fully connected layer which gives logit outputs equivalent to the number of classes in the training dataset. Then comes the softmax layer, which converts the logits to class probabilities.

We, for our experiments, introduce the log-Softmax activation for the residual layers after the residual skip connection(as discussed previously in section 3.1). Thus, for the ResNet-34 architecture, it will be 16(sum of all residual layers in all the residual blocks) log-Softmax activation. Also, we perform entropy minimization for all of these 16 layer outputs.

3.3.2 Datasets

In-distribution Datasets

For the in-Distribution dataset or the training dataset, we train the classifier with CIFAR-10, and CIFAR-100 datasets [Kri09]. This choice of datasets for in-distribution is what is frequently tested in the OOD detection literature. Another aspect we consider while choosing the in-distribution datasets is the data distributional shift. As discussed earlier in section 2.4 this is based on the CLP metric introduced in [Win+20]. The in-and out-distribution datasets for different dataset distance categories are shown in table 2.2. For all the categories, the in-distribution selected was either CIFAR-10 or CIFAR-100.

Fig. 3.3.: Images from each of the 10 classes of CIFAR10.Image from [@Ost19].

CIFAR10: This dataset consists of 60000 color images in 10 classes with 6000 images per class. Each of these images has a resolution of 32×32. For training, 50000 images are used, and for testing 10000 images. See figure 3.3 for some samples.

CIFAR100: This dataset is similar to CIFAR-10 except that it has a total of 100 classes containing 600 images each. There are 500 training images and 100 testing images per class. They are broadly categorized into 20 superclasses. Some of them are aquatic mammals, fish, flowers, fruit and vegetables, vehicles, and others. A sample collection of images from CIFAR100 dataset is shown in figure 3.4.

Out-Distribution Datasets

For out-distribution detection testing we experiment on some of the common datasets experimented in the literature TinyImageNet [Den+09], LSUN [Yu+16], SVHN [Net+11]. Also considering the data distributional shift (see table 2.2) to experiment with different OOD categories we also experiment with CIFAR100 when CIFAR10 is in-distribution and vice versa.

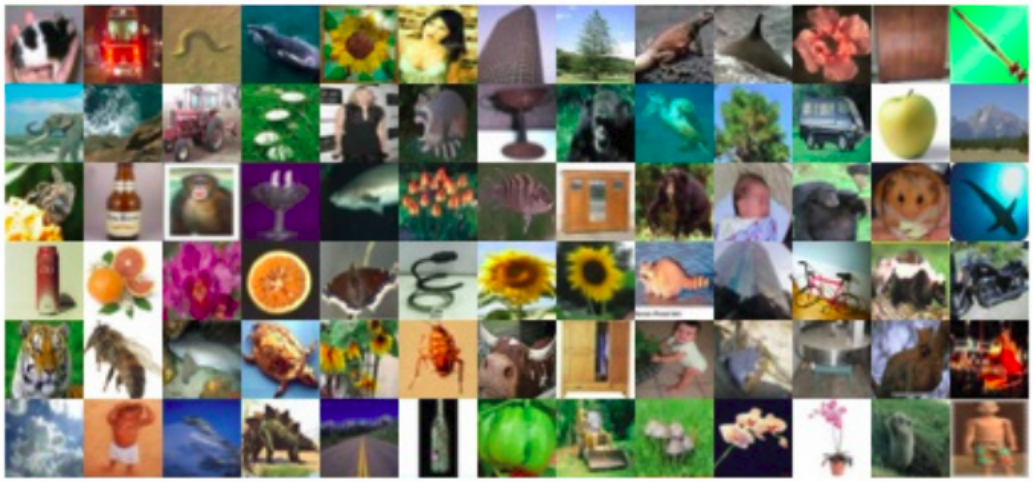


Fig. 3.4.: Some sample images from the CIFAR-100 dataset.There are images from 100 classes which are under the superclasses of flowers,fish,people,trees,reptiles etc. Image from [@Gar20].

LSUN: The Large-Scale Scene UNderstanding(LSUN) dataset was initially introduced to address the scarcity of large-scale image datasets for visual recognition benchmarking tasks. The images in this dataset are broadly classified into 10 scene categories (e.g., bedroom, kitchen, church, etc.). For our experiments, we adopt the dataset configurations similar to [Lee+18], where 10000 images, 1000 images from each of the 10 scene categories are used as the out-distribution dataset. The images are downsampled to 32×32 so that the image resolution equals the in-distribution datasets CIFAR10 and CIFAR100. Though there is no categorization for the LSUN dataset in terms of out-distribution distance from in-distribution as introduced in [Win+20]. Intuitively, based on the visual appearance and the image class differences, we consider LSUN as a 'far OOD' dataset when the in-distribution datasets are CIFAR-10/CIFAR-100. Some sample images from four different scenes from the LSUN dataset are shown in figure 3.5.

TinyImageNet : It is a dataset constructed as a subset of the ImageNet [Den+09] dataset. For out-distribution testing, 10000 images from 200 different classes are used. Each of these images is downsampled to 32×32. Similar to the dataset distance categorization of LSUN, we do not have a quantification from [Win+20]. We consider the TinyImageNet also a 'far OOD' dataset since there is minimal overlap of classes compared to in-distribution CIFAR-10/CIFAR-100 datasets. Some sample images from this dataset is shown in figure 3.6.

SVHN: The Street View House Numbers (SVHN) dataset consists of images of digits obtained from house numbers in Google Street View images. The out-distribution

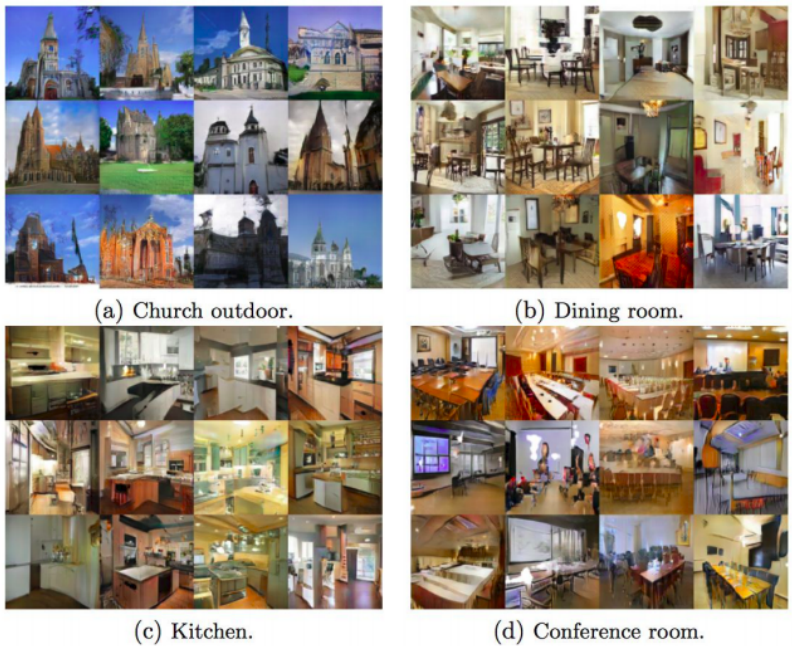


Fig. 3.5.: Sample images from four different scenes from LSUN dataset. Image taken from [Bon17].

test-set consists of 26032 images from 10 classes, one for each digit. The resolution of the images is 32×32. According to the out-distribution distance calculation in [Win+20] for in-distribution dataset CIFAR-10, the SVHN dataset is considered as 'far OOD'. Some sample images from the dataset are shown in figure 3.7.

Introduction of the mixture datasets : We introduce three more datasets to study whether any particular intermediate layer with log-softmax activation is beneficial for OOD detection based on data distributional distances. According to [Win+20] for CIFAR-10 as in-distribution, CIFAR-100 is considered as 'near and far OOD' regime

Fig. 3.6.: Sample images from TinyImageNet dataset.Image from [CT18].

Fig. 3.7.: Sample images from SVHN dataset. Image taken from [Net+11].

and SVHN dataset as 'far OOD'. Table 2.2 shows all the dataset setting introduced in [Win+20].

In order to experiment with how the OOD detection varies per intermediate layer with dataset distances for datasets that lie between the CIFAR-100 and SVHN, we mix CIFAR-100 and SVHN datasets in 3 different configurations. Image samples are randomly drawn from the CIFAR-100 and SVHN datasets in a fixed percentage for each dataset. The percentage of samples drawn from CIFAR-100 and SVHN to create the new datasets is described in table 3.2. This way of mixing the datasets is motivated by how the confusion log probability(CLP) is computed as discussed in section 2.4. More the samples mixed from 'far OOD' dataset SVHN with the 'near and far OOD' dataset CIFAR-100, the CLP score decreases, and the new dataset moves away from the 'near and far OOD' spectrum to 'far OOD.' We do not measure the CLP metric for these newly created datasets. We deem them to lie between the CIFAR-100 distribution and SVHN, depending on the configurations in which the datasets are mixed. i.e., more the percentage of CIFAR-100 samples in the mixture datasets, the dataset lies close to CIFAR-100 and similar for the SVHN dataset. All the newly created mixture datasets have 10000 images each.

Dataset	CIFAR-100	SVHN
CIFAR100-75-SVHN-25	75%	25%
CIFAR100-50-SVHN-50	50%	50%
CIFAR100-25-SVHN-75	25%	75%

Tab. 3.2.: Table showing the percentages in which we mix the samples from CIFAR-100 and SVHN datasets to create new datasets which intuitively should lie between the 'near and far OOD' regime i.e., when CIFAR-10 is the in-distribution, and CIFAR-100 is the out-distribution and 'far OOD' when CIFAR-10 is in-distribution and SVHN the out-distribution.

4 Results

This chapter discusses the empirical results of our experiments to study the benefits of the log-softmax introduction to the classifier network for OOD detection. The experimental section is divided broadly into two sections. In the first section, we perform the experiments with CIFAR10 as the in-distribution. In the next section, we show the experimental results with CIFAR100 as the in-distribution dataset.

For the CIFAR-10, the experimental results are structured as follows. First, we discuss the details of training the classifier with log-softmax integration and entropy minimization and show that these models retain classification performance. Then we discuss the experimental results of evaluating the log-softmax models with existing softmax-based baseline method MSP and the Mahalanobis-distance based method, which uses the intermediate features of the classifier. Also, we present the results of MiSP based OOD evaluations, which we surprisingly find to work better than the MSP method. Next, going into the intermediate layer OOD evaluations, we present the layerwise OOD evaluation comparison with the newly introduced generalized MSP method for intermediate log-softmax outputs. Then show that with this cheap generalization of the MSP method, competitive performance can be achieved compared to the Mahalanobis-distance method, which needs additional training of an OOD detector. Then we discuss the observations made for the generalized MSP method based on the dataset distances. We observe early layers of the network are beneficial for detecting sample from far distribution and later layers for nearby distribution.

In the case of the CIFAR-100 experiments, we do not perform extensive experiments as performed for CIFAR-10. We only perform enough experiments to verify whether the observations made for CIFAR-10 generalize to CIFAR-100.

4.1 Results for in-distribution dataset: CIFAR-10

This section covers the experimental results when the CIFAR-10 dataset is used as the in-distribution(training dataset). We use the ResNet-34 classifier architecture as the base architecture for our training. First, we train the baseline ResNet-34 with the

ReLU activation and then the ResNet-34 models with the log-softmax activation, with different factors of entropy minimization. Then the OOD evaluations are performed for all the models. When CIFAR10 is the in-distribution, the out-distribution datasets chosen for OOD evaluations are CIFAR100, LSUN, TinyImageNet, and SVHN. Further considering the intermediate layerwise OOD evaluations, we perform the OOD evaluation for the newly introduced datasets CIFAR100-75-SVHN-25, CIFAR100-50-SVHN-50, CIFAR100-25-SVHN-75. Here, we study how each layer gets beneficial for OOD sample detection as the OOD dataset tested is introduced between the 'near and far OOD' to 'far OOD' categories.

4.1.1 Classification performance of log-softmax models

Implementation Details

We train the baseline ResNet-34 architecture with ReLU activation with cross-entropy loss as introduced in equation 3.1. For the ResNet-34 models with log-softmax activation, the adopted loss is introduced in equation 3.2. Thus, the total loss considered for training is the sum of cross-entropy loss and the entropy loss with regularizer λ. We adopted the training configurations similar for the baseline ResNet-34 with ReLU activations and the log-softmax activations except for minor changes in the learning rates.

We tried different optimizers and learning rate schedulers and found that AdamW introduced in [LH19] along with One-Cycle learning rate scheduler [ST18] helped retain classification accuracy as well as faster convergence of the training. The AdamW optimizer was trained with a base learning rate of 10^{-4}, with a weight decay of $5 \cdot 10^{-4}$, and with a momentum of 0.9. For the One-Cycle learning rate scheduler for the baseline ReLU model, we use a maximum learning rate of 10^{-2}. For log-softmax models, we use the maximum learning rate of 10^{-3} and use the linear anneal strategy for both models. With the linear anneal strategy at the starting of the training, the learning rate would be closer to the base learning rate of 10^{-4} and it will be linearly increased to the max learning rate 10^{-2} around the 30% of the total epochs, and then it will be linearly decreased to the base learning rate by the end of the training. For both the baseline and log-softmax models, we train for 95 epochs and with a batch size of 256. While training, we also perform image normalization and data augmentations like random horizontal flip and random cropping.

For the models with log-softmax activations, we experimented with different entropy loss factors λ ranging from 0 to $1e{-}1$. The details of the entropy loss values experimented with can be seen under the Entropy Loss factor column in table 4.1.

Classification performance comparison

With the implementation details mentioned above, we train our classifier models with the CIFAR-10 dataset with 50000 training samples, and the test accuracy is measured for 10000 test samples. We report the classification performance, the top-1 accuracy of the different models in table 4.1. It is clear from the results that the baseline ResNet-34 with ReLU activation has the best top-1 accuracy of 94.31%. However, the model with the log-softmax activation also retains very close by top-1 accuracy of 93.5%. Instead of log-softmax, we also experimented with the softmax activation. Using the same training setup for log-softmax models, we observed that softmax activation could only retain top-1 accuracy of 89.75% compared to 93.5% for the log-softmax model. As noted in [GBB11] the benefits of unbounded activation over bounded ones, due to the promotion of sparse representation and can potentially prevent gradient saturation, could have been the reason the log-softmax models retained better accuracy than the softmax activation model.

Effects of entropy loss factors on the classification accuracy: For the models with log-softmax activation, we experiment with different factors for entropy minimization. We observe that the lower the entropy minimization factor higher the accuracy. As it can be seen in table 4.1 with increasing the entropy minimization factor, the top-1 classification accuracy drops.

Observations from results in table 4.1 show that it is possible to train classifiers with log-softmax activation, which can produce a comparable performance as a ReLU activation-based classifier. Moreover, with careful selection of the entropy loss factor, it is possible to introduce the log-softmax activation for intermediate layers of the classifiers and retain the classification accuracy.

4.1.2 OOD performance evaluation of log-softmax models

We have shown that the log-softmax models retain classification performance. Now we evaluate how beneficial these models are for out-of-distribution detection. To test whether the intermediate log-softmax introduction affects the final softmax-based evaluation, we perform the MSP and the newly introduced MiSP methods.

Network Architecture: ResNet34 In-Dist: CIFAR-10		
log-Softmax	Entropy Loss factor(λ)	Top-1 accuracy(%)
-NA-	-NA-	**94.31**
yes	0	93.5
yes	1e-3	93.37
yes	5e-3	93.09
yes	1e-2	92.64
yes	5e-2	92.48
yes	1e-1	91.92

Tab. 4.1.: Table showing the classification performance of our models.The Top-1 accuracy is calculated for the held out CIFAR-10 test set. The baseline is the Resnet34 with both log-softmax and Entropy Loss factor row entry as -NA- . Note that higher the entropy loss factor lower the classification top-1 accuracy.

Finally, to test how well the log-softmax-based models perform compared to baseline when considering an intermediate feature-based OOD evaluation, we perform the Mahalanobis-distance-based evaluation. All the evaluations are performed for the vanilla model with ReLU activation, which is our baseline for comparing the OOD evaluation performance of log-softmax models. We report the results for out-distribution datasets CIFAR-100, LSUN, TinyImageNet, and SVHN. Note that CIFAR-100 is considered a 'near and far OOD dataset' and all others as 'far OOD' datasets.

Implementation Details.

For the MSP and MiSP methods, there is no additional processing step for calculating the OOD score. As discussed earlier in section 3.2.2 we do not perform the ASP method for the final layer softmax outputs. For MSP, the maximum softmax probability is used as the score, and for MiSP, the minimum softmax probability of the final softmax layer.

For the Mahalanobis method, the features from different intermediate layers are used for deriving an OOD score as discussed in 2.2.3. To apply the Mahalanobis-distance method for the ResNet-34 architecture, we choose the layers after the end of each residual block and after the first convolution block, which brings to a total of 5 layers(4 after residual blocks + 1 after the first convolution layer). Then for the features from the 5 layers, the class conditional Gaussian distribution fitting is performed per layer, and the Mahalanobis score is computed per layer for 1000 images from the in- and out-of-distribution datasets. Additionally, to improve the Mahalanobis score, we experiment by adding a small noise to the input image as

discussed in 2.2.3. And for the ϵ (in equation 2.6) which is the noise magnitude factor we experiment with values [0.0, 0.01, 0.005, 0.002, 0.0014, 0.001, 0.0005] and choose the one which gives the best Mahalanobis score. Then with these scores from these 5 layers, a logistic regression detector is trained to derive the final OOD score for in- and out-distribution samples. This methodology is the same as introduced in the Mahalanobis-distance-based method [Lee+18].

Once a score is derived from MSP, MiSP, or Mahalanobis distance-based method. The scores for in- and out distributions are sorted. The True Positive Rate(TPR) and False Positive Rate(FPR) are computed for each possible threshold. Then, plotting the relationship between TPR and FPR for all possible thresholds, the area under this curve is the required AUROC metric that we use to compare the performance of the detectors. (more details in section 2.1).

OOD evaluation performance comparison

First, we verify whether we can reproduce the OOD evaluation performance for baseline ResNet-34 with ReLU activation with our training methodology since we adopted a different training strategy than one proposed in [Lee+18]. Where they used SGD optimizer and Step learning rate decay scheduler and trained for 200 epochs. In table 4.2 the first two rows compare the baselines results reported in [Lee+18] (first row) and the baseline ResNet-34 trained by us (second row). We observe similar or better OOD detection performance for all datasets for both MSP and Mahalanobis methods. Now for comparing the OOD evaluations of the log-softmax models, our model's OOD evaluation results are considered the baseline.

log-softmax models OOD performance for MSP: Considering the MSP-based OOD evaluation (see table 4.2), for CIFAR100 as an out-distribution dataset, the performance is comparable for baseline and log-softmax models. However, for TinyImageNet, there is a drop in performance of more than 5% for the log-softmax model with an entropy loss factor of 0 and $1e{-}2$. However, it is not that worse for the model with a higher entropy factor of $1e{-}1$. For LSUN and SVHN, we notice a slight performance drop. With MSP evaluation, there is no clear best-performing model considering all datasets. For the 'near and far OOD dataset' CIFAR100, the OOD evaluation performance with log-softmax models are retained, and for 'far OOD' datasets(SVHN, TinyImageNet, and LSUN), the OOD evaluation drops. To conclude, for models with log-softmax activation at intermediate layers, with and without entropy minimization, the softmax-based MSP evaluation at the final softmax layer

does not improve over the baseline ReLU architecture. This shows there is no direct benefit for the MSP method with the intermediate log-softmax introduction.

log-softmax models OOD performance for Mahalanobis-distance based: For Mahalanobis-distance-based evaluation, the results are comparable for all models and all datasets(see table 4.2). Another general observation is considering the average performance for all datasets, the model with a higher entropy loss factor of $1e{-}1$ had poor performance compared to the other models. An interesting observation is for the CIFAR-100 out-distribution dataset. The model with log-softmax activation performs better than the baseline. However, the MSP evaluation technique is better than the Mahalanobis-distance-based evaluation for the CIFAR-100 dataset in all cases. Again no one architecture is better in performance comparing other models for all datasets. With this, we conclude that for an intermediate feature-based OOD evaluation method, the log-softmax models perform as competitive as the baseline model with ReLU activation. This also indicates that the intermediate features of the log-softmax models also enable OOD detection as for the ReLU based model. Also, we observe that the Mahalanobis method performs worse for the 'near and far OOD' dataset CIFAR-100.

MSP vs MiSP

In the MSP and the ODIN method, the intuition to use the maximum softmax probability as an OOD score was that the CNN classifiers assign higher softmax score values for samples from in-distribution than a sample from out-distribution. Surprisingly we find with empirical experimentation that the minimum softmax probability(MiSP) could also be used as an OOD score. The results are presented for the MiSP evaluation in table 4.2. We observe that for the in-distribution as CIFAR-10 setting, the MiSP consistently performs better than the MSP. We have not performed a further study on why the MiSP based scoring is better for out-of-distribution detection.

For some out-distribution datasets and entropy loss factor settings, the MiSP based OOD evaluation is comparable to the Mahalanobis-distance-based method. E.g., for the model with entropy loss factor $1e{-}2$ and out-distribution dataset SVHN, the MiSP evaluation gives the AUROC measure of 97.41. On the other hand, with the Mahalanobis distance-based evaluation, the AUROC is 98.19.

To conclude the OOD evaluations for the log-softmax models, for intermediate feature-based evaluation technique the Mahalanobis-distance based OOD evaluations, the performances are comparable to the baseline architecture OOD evaluations.

Network Architecture: ResNet34 In-distribution: CIFAR-10					
log-Softmax	Entropy Loss factor(λ)	Out-distribution dataset	AUROC(↑)		
			MSP	Mahalanobis Distance	MiSP
-NA- (from [Lee+18])		CIFAR100	-NA-	-NA-	-NA-
		TinyImageNet	91.0	99.50	-NA-
		LSUN	91.0	99.70	-NA-
		SVHN	89.90	99.10	-NA-
-NA-(Ours)		CIFAR100	84.89	66.06	88.19
		TinyImageNet	92.50	99.46	95.22
		LSUN	93.48	99.75	95.66
		SVHN	92.08	98.62	96.54
yes	0	CIFAR100	85.20	73.51	88.39
		TinyImageNet	83.82	99.04	89.80
		LSUN	88.48	99.46	93.53
		SVHN	90.27	98.85	94.70
yes	1e-2	CIFAR100	83.26	74.07	86.23
		TinyImageNet	83.35	98.90	88.43
		LSUN	89.89	99.45	95.32
		SVHN	92.43	98.19	97.41
yes	1e-1	CIFAR100	84.68	65.85	85.78
		TinyImageNet	88.43	97.85	93.07
		LSUN	90.16	98.53	94.31
		SVHN	89.89	98.34	91.41

Tab. 4.2.: Table showing the MSP,Mahalanobis distance based and MiSP OOD evaluation results for our models in comparison to the baseline model reported in the paper [Lee+18]. The models with log-softmax and entropy loss factors as -NA- are the baseline models with ReLU activation. The first row is the model from the paper [Lee+18]. ↑ indicates higher the AUROC value the performance is better.

However, there is a performance drop for the log-softmax models for softmax-score-based MSP evaluations compared to the baseline. However, the surprising observation we make is the OOD performance evaluation with the MiSP based OOD scoring, which is observed to work better than the MSP method. See appendix table A.1 for detailed OOD evaluation results for all the entropy loss factors experimented.

4.1.3 Intermediate Layerwise OOD evaluation

We perform the OOD evaluations with intermediate layers outputs to verify whether the log-softmax activation enables out-of-distribution detection at the intermediate

layer level. Considering the ResNet-34 architecture, we perform this evaluation for 16 residual layers, after which the log-softmax activations were introduced. In this section, we first discuss the baseline OOD evaluations at the intermediate layers. Then we discuss why the MSP generalization introduced for intermediate layer log-softmax outputs, which is based on aggregating the log-softmax outputs as discussed in section 3.2.4 can be used as a cheap OOD scoring technique. Then we compare the layerwise OOD detection performance of the baseline performed with the layerwise Mahalanobis-distance method and the layerwise log-softmax aggregation based OOD evaluation for the intermediate layer outputs of the log-softmax models.

Mahalanobis distance method as a baseline.

We use the Mahalanobis-distance score computed per layer as the OOD score to create the baseline for intermediate layer outputs for the vanilla ReLU based ResNet-34 model. We perform this layerwise Mahalanobis-distance method for the same 16 layers for the baseline ReLU based model as the layers after which the log-softmax activations are introduced for the log-softmax-based ResNet-34 model. For computing the Mahalanobis score as introduced in equation 2.5, first, the Gaussian distribution fitting is performed individually for features from each of the 16 intermediate layers. For this, 1000 images each from in- and out-distribution datasets are used. Then the Mahalanobis distance to the closest class is computed as in equation 2.5 for an input image from in- or out-distribution. Then to improve the Mahalanobis-distance score a small controlled noise is added to the input image with noise magnitude ϵ (in equation 2.6) which is selected from [0.0, 0.01, 0.005, 0.002, 0.0014, 0.001,0.0005]. Then for this input preprocessed sample, the Mahalanobis-distance to the closest class is computed again as in equation 2.5. Then this distance is used as the OOD score to evaluate the OOD detection performance of the detector at a given intermediate layer.

Figure 4.2 summarizes the OOD performance per layer for the vanilla ResNet-34 architecture, the model with log-softmax activation with entropy loss as 0 and $1e^{-2}$. For the vanilla ResNet-34 architecture, the layerwise OOD evaluation performed is the Mahalanobis-distance evaluated per layer. For the log-softmax models, it is the log-softmax aggregation method min of MiSP. Considering only the Mahalanobis-distance evaluated per layer for the baseline ReLU model(baseline in figure 4.2), we note that not all layers are equally beneficial for OOD detection. This is why in [Lee+18] proposed usage of multiple layer outputs to generate an OOD score. Generally, we observe that layers from 2^{nd} residual block (from layer2.1 to layer2.4, notation: '2' represents the 2^{nd} residual block and 1, and 4 represents the 1^{st} and

4^{th} residual layer in that block) of the ResNet-34 to be more beneficial for OOD detection in comparison to the other residual blocks. Layers from the last blocks are not that beneficial as the layers from 2^{nd} and 3^{rd} residual block. We observe that for the 'near and far OOD' setting, i.e., in-distribution CIFAR-10 and out-distribution CIFAR-100, the performance of the intermediate layers is poorer than for the other far datasets. E.g., for CIFAR-100 as out distribution, the best performing layer is layer2.4. The AUROC metric for this layer is 64.62. Compared with SVHN, a 'far OOD' dataset, the best performing layer is layer2.3 with AUROC 97.95. However, we observe that for CIFAR-100, the layers from residual blocks 1 and 2 have better OOD detection performance than those from later blocks.

Additionally, the Mahalanobis-distance-based evaluation performed per layer can be performed for the intermediate layers of the log-softmax models. This evaluation comparison for the vanilla ResNet-34 model with the models with log-softmax intermediate activation is presented in appendix section A.2. Since we observed similar patterns in OOD performance per layer for the vanilla ReLU based model and log-softmax. We did not notice a significant improvement for the log-softmax model with and without entropy minimization except for few pattern differences in the case of CIFAR-100 as the out-distribution dataset. We do not discuss these results further here.

Aggregation of intermediate layer log-softmax outputs as an OOD score

The log-softmax as intermediate layer activation benefits leveraging these intermediate outputs for devising an OOD score as a generalization to the MSP score. As introduced in 3.2.4 the nine ways of deriving the OOD score, we see some of this generalization to be beneficial for use as an OOD score.

Especially we find that the minimum of Minimum Softmax Probability (min of MiSP) to be a good way of aggregating the log-softmax outputs at intermediate layers and using this score for the OOD detection. The benefits are illustrated in figure 4.1. The figure shows how the aggregated min of MiSP log-softmax value distributions are for in- and out- datasets. We observe that the distributions are highly separable when the in-distribution and the out-distribution datasets are far away. E.g., in the case of out-distribution datasets SVHN, LSUN, and TinyImageNet, the OOD score distributions are highly separable with AUROC of the OOD detector with the min of MiSP scoring being above 98. Moreover, for the 'near and far OOD dataset' CIFAR-100, min of MiSP values are not as separable as the other datasets. The AUROC value of 86.06 observed is better than what was observed with the MSP and

Mahalanobis-distance-based evaluation results shown for different models in table 4.2. Also, we observe the average of Average Softmax Probability (avg of ASP) to be beneficial for some cases. The detailed results are discussed in table 4.3.

Additionally, this method does not need any access to OOD datasets for hyperparameter tuning. Further, these evaluations are cheap to compute and do not have the overhead of training additional OOD detectors. Though we see benefits in the min of MiSP aggregation methods for OOD detection at the intermediate layer, we also observe that the min of MiSP log-softmax value spectrum is not consistent for in- and out-distribution dataset configurations. For example, see figure 4.1, when CIFAR-10 is the in-distribution, and for CIFAR-100 and SVHN as out-distribution datasets, the min of MiSP values are in the higher ranges for these datasets when compared to min of MiSP values for in-distribution CIFAR-10. On the other hand, for the out-distribution datasets TinyImageNet and LSUN the min of MiSP values are in the lower ranges. So the min of MiSP OOD score sign has to be adjusted while computing the AUROC metric depending on the dataset.

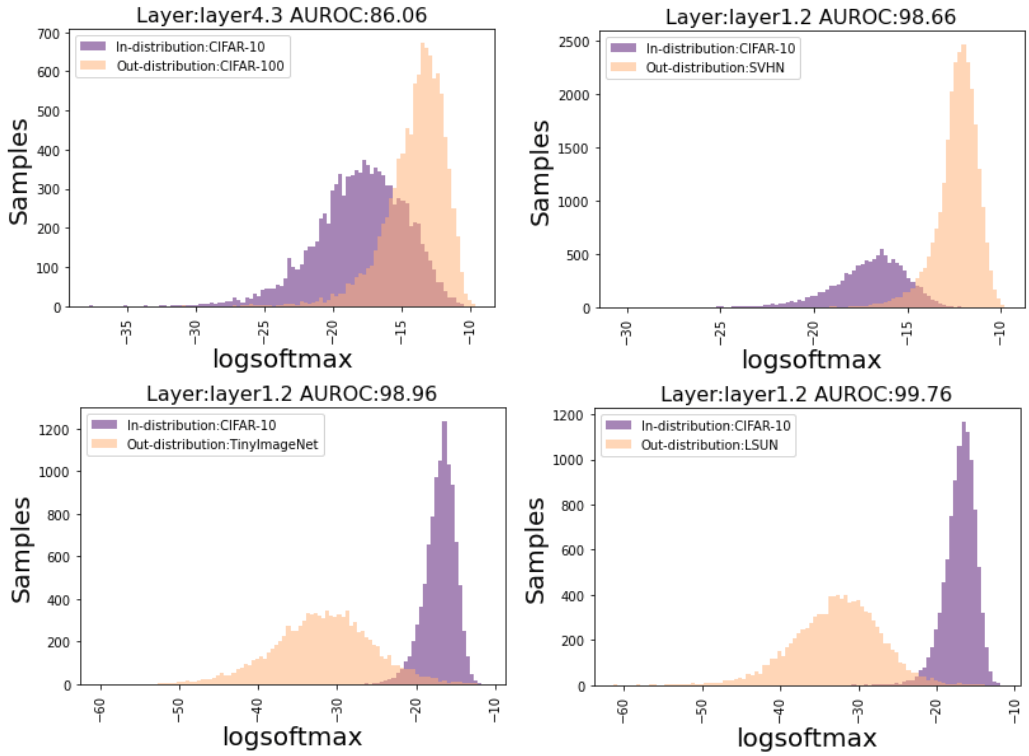


Fig. 4.1.: Figure shows the minimum of Minimum Softmax Probability(min of MiSP) aggregation of intermediate log-softmax outputs for in- and out-distribution datasets. The in-distribution dataset for all case is CIFAR-10 and the out-distribution CIFAR-100,SVHN,LSUN,TinyImageNet. Top of every figure shows the layer from which the log-softmax outputs are used and the AUROC score for the OOD detector with min of MiSP evaluation at this layer.

Intermediate layer OOD evaluations of log-softmax models.

Comparison of min of MiSP with Mahalanobis-layerwise Now we have established the utility of aggregation of intermediate-layer log-softmax output as an OOD score. Here, we discuss how each layer's OOD detection performance compares with the Mahalanobis-layerwise baseline for the aggregation evaluation.

For the intermediate softmax aggregation evaluation, we find the min of MiSP to be more beneficial than the other aggregation methods. In figure 4.2 we show the layerwise evaluation comparison for the baseline ResNet-34 model with ReLU activation evaluated with layerwise Mahalanobis distance and the min of MiSP layerwise evaluation performed for the log-softmax model with entropy loss 0 and $1\mathrm{e}{-2}$. Again, as the similar observation made for Mahalanobis-layerwise evaluation, we observe that different layers enable different OOD detection performance for the min of MiSP layerwise evaluation. Generally, we see the OOD detection performance for layers in residual block 2 and 3 is lower than the other layers. This is contrary to the observation made for Mahalanobis-layerwise evaluation, where it was more beneficial for these layers.

Effects of entropy minimization for log-softmax aggregation based evaluation: Though initially, the motivation to introduce the entropy minimization for the intermediate layer outputs was to mimic the cross-entropy loss at the final layer. Since cross-entropy was crucial for the final layer MSP evaluation, we deemed the entropy minimization could bring the same effect for the max of MSP evaluation for the intermediate layers. However, in most cases, the min of MiSP evaluation is a better OOD scoring method than the other aggregations evaluated. So we discuss only the effects observed with entropy minimization for the min of MiSP evaluation.

In figure 4.2 we compare the layerwise OOD detection performance with the min of MiSP OOD scoring of the log-softmax model with no entropy minimization (Soft+EL0) and with entropy minimization factor of $1\mathrm{e}{-2}$ (Soft+EL1e-2). We observe that entropy minimization in the initial layers of the network does not bring much improvement over the log-softmax model with no entropy minimization. However, in final layers, especially in the 4^{th} residual block, layers 4.1-4.3, the OOD detection performance improves with entropy minimization. E.g., for CIFAR-100 out-distribution dataset at layer4.2, the log-softmax model has an AUROC of 81.18 and the model with entropy minimization factor $1e-2$ has an AUROC of 87.16. It is not clear why the entropy minimization does not significantly affect the intermediate log-softmax outputs.

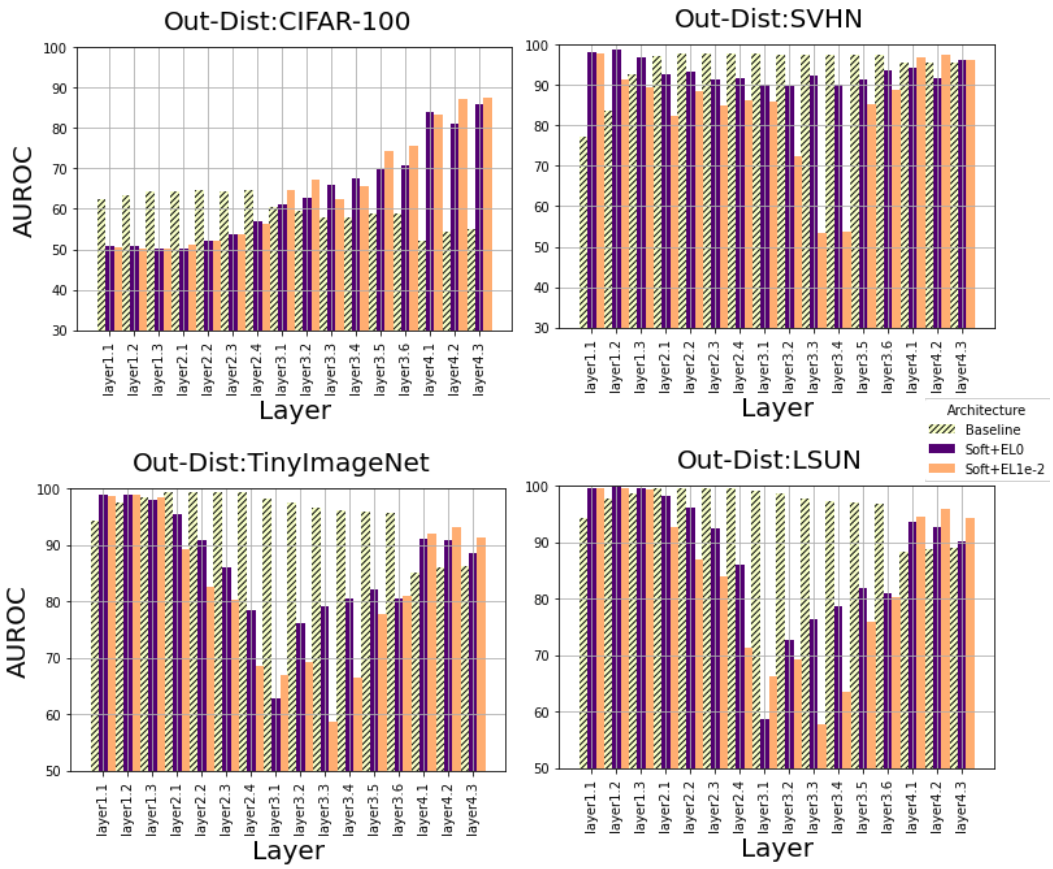


Fig. 4.2.: Figure showing the comparison of Mahalanobis layerwise evaluation for the baseline model and the min of MiSP aggregation based evaluation for log-softmax models with entropy loss factor 0 and $1\mathrm{e}{-2}$. The base architecture is ResNet-34. The in-distribution dataset is CIFAR-10.

Intermediate layer OOD detection performance based on OOD dataset distances.

Another observation we make for the min of MiSP based OOD scoring is that different layers in the network enable different OOD sample detection performance based on the out-distribution dataset distributional shift to the in-distribution. We observe that for a difficult OOD detection setting, i.e., for the CIFAR-100 as out-distribution, classified under the 'near and far OOD' dataset when the in-distribution is CIFAR-10, the initial layers perform very poorly. E.g.,see figure 4.2, the min of MiSP OOD detection performance for layers 1.1-1.3 for the log-softmax models, the AUROC values are close to 50. This indicates that these layers could only perform as well as a chance detector. Furthermore, as the layer depth increases, the OOD detection performance improves. However, for the 'far OOD' datasets SVHN, TinyImageNet, and LSUN, we observe that initial layers (layers 1.1 - 1.3) are more beneficial. As depth increases, the OOD detection performance drops in the residual block 2 and

in the initial layers of residual block 3 and again improves with further depth. The layers in the 4^{th} residual block reach comparable performance to the initial layers. This trend observed for the 'far and near OOD' dataset CIFAR-100 and 'far OOD' datasets SVHN,TinyImageNet, LSUN are illustrated in figure 4.3. Further, in figure 4.3 we compare the intermediate layer min of MiSP evaluation with MiSP evaluation performed for the softmax probability outputs of the final layer. We observe that for CIFAR-100 out-distribution dataset, the MiSP at the final layer is better than the min of MiSP at any intermediate layer. But, for other out-distribution datasets there is more than one intermediate layer usually initial layers for which the min of MiSP evaluation is better than the MiSP at final layer.

Layerwise OOD detection performance for mixture datasets.

We observed that with the min of MiSP OOD scoring, features from an early layer in the network are enough to detect samples from 'far OOD' distributions, and features from the later layers of the network offer the best OOD detection for samples from 'near and far OOD' distribution. To further study how the network layers enable OOD detection when encountering data from the distribution between the 'near and far OOD' and 'far OOD' distributions, we evaluate the min of MiSP per layer for the newly introduced mixture datasets of CIFAR-100 and SVHN as described in table 3.2.

The min of MiSP evaluations per layer is compared in figure 4.4 for CIFAR100, SVHN, and the three newly introduced mixture datasets. We observe that as the dataset distance increase from the CIFAR-100 distribution to SVHN, the initial layers start getting better for OOD detection performance as observed for 'far OOD' datasets. For the mixture datasets, the final layers are in general, more beneficial than the initial layers. However, we observe that the rate at which the initial layers(layer1.1,layer1.2) benefit OOD detection as the dataset distribution moves from CIFAR-100 to SVHN is higher than the OOD detection gain for layer4.3.

This experiment further shows that the final layer features are generally suitable for the min of MiSP based OOD evaluation for any dataset. For a dataset that is near to the in-distribution, the features of the initial layers are not beneficial for the min of MiSP based OOD evaluation. However, for 'far OOD' datasets, the initial layer features are more beneficial than the final layer features for the min of MiSP OOD evaluation which was illustrated in figure 4.3.

4.1.4 Performance comparison of log-softmax aggregation methods for intermediate layer with existing methods from literature

This section compares how competitive the newly introduced nine intermediate log-softmax-based aggregation methods are in comparison to the MSP and the Mahalanobis-distance-based method from literature.

Table 4.3 compares the OOD evaluations for methods from literature and the results of intermediate log-softmax aggregations. We compare only the best results from the methods from literature and the best result from the intermediate layer aggregation for simplicity. For the baseline ReLU based architecture, there is no intermediate layer aggregation evaluation. For the 'near and far OOD' dataset CIFAR-100, we already observed in table 4.2 that MSP is a better method than the Mahalanobis-distance based. Here we observe that for all the models, the intermediate layer aggregation has better results than MSP. E.g., For the log-softmax model with the entropy minimization factor of $1e{-}2$, the AUROC with the MSP method is 83.26. With aggregation methods, we have AUROC of 87.44 for the min of MiSP aggregation.

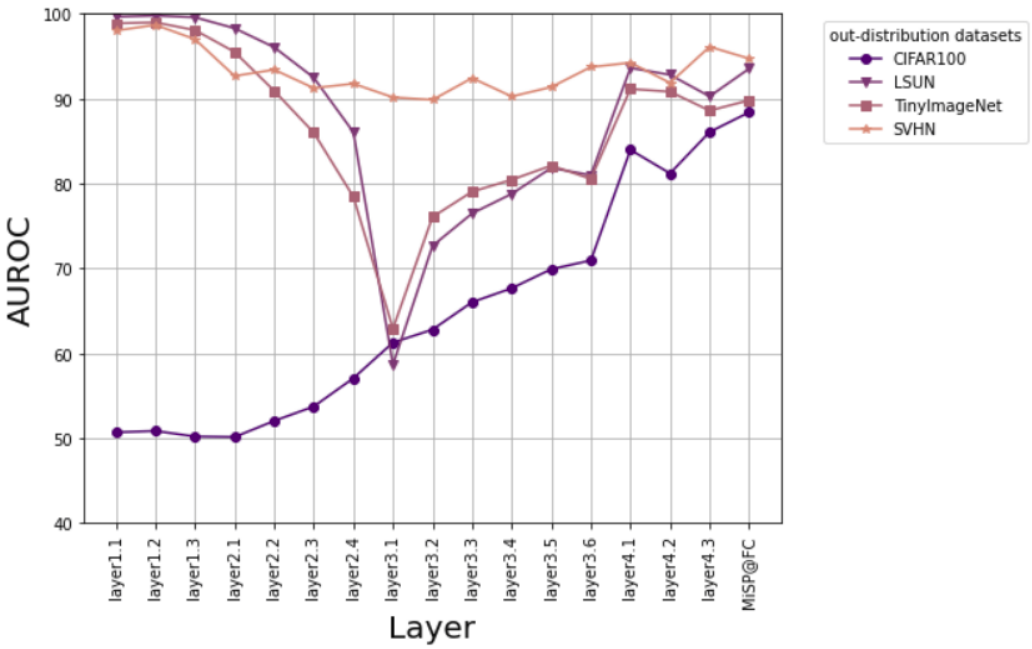


Fig. 4.3.: Figure shows the min of MiSP evaluations performed per intermediate layer and MiSP evaluation performed for the final classification layer for the log-softmax model with entropy loss 0. The in-distribution dataset is CIFAR-10. In out-distribution datasets CIFAR-100 is 'near and far OOD' dataset and all others are 'far OOD' dataset.

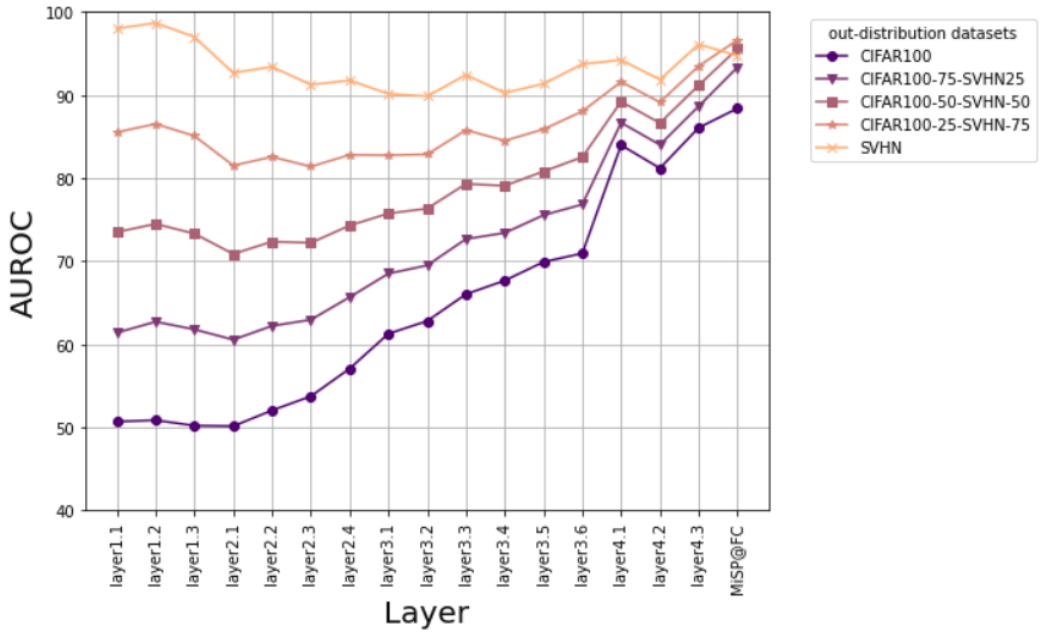


Fig. 4.4.: Figure shows the min of MiSP evaluation performed per intermediate layer and MiSP evaluation performed for the final classification layer for the log-softmax model with entropy loss 0. The in-distribution dataset is CIFAR-10. The out-distribution datasets are CIFAR-100, SVHN and the three newly introduced mixture datasets. As the data distribution moves away from CIFAR-100 the initial layers gets better for OOD detection.

Considering the 'far OOD' datasets, we observe that the aggregation-based methods are equally competitive as the Mahalanobis-distance method. In some cases, the OOD detection performance with the aggregation methods is slightly better. For example, for the log-softmax models with entropy loss 0 and $1\mathrm{e}{-2}$ and out-distribution dataset LSUN, we observe the aggregation evaluation to be better than the Mahalanobis-method.

Considering the intermediate layer aggregation methods, we observe that the best results are mainly from the min of MiSP evaluation, except for the log-softmax model with entropy loss factor 0 and $1\mathrm{e}{-1}$ CIFAR-100 as out-distribution, the best aggregation method was avg of ASP. Though the best evaluation for CIFAR-100 for the other models is the min of MiSP, the layer at which the best results were observed remains the same. Generally, for 'near and far OOD datasets, ' layer4.3 of the network is beneficial for OOD, and 'far OOD' datasets its either layer1.1 or layer1.2. To conclude the aggregation methods, we find that applying these OOD scoring methods is cheap, does not need additional training, and is hyperparameter free. However, it can be as competitive as the Mahalanobis-distance method, which

uses features from multiple intermediate layers and requires an additional logistic regression detector for the OOD score computation.

4.2 Results for in-distribution dataset: CIFAR-100

So far, we have discussed the results with the in-distribution as CIFAR-10. According to out-distribution dataset distributional shift categorization we discussed the cases of 'near and far OOD' and 'far OOD' dataset spectrum(more details in table 2.2). Now in this section, we discuss the results with training distribution as CIFAR-100. We cover the 'near OOD'(e.g., CIFAR-10) out-distribution category, which is deemed the most difficult setting for an OOD detector according to [Win+20]. Further, for the experimental results in this section, we discuss how generalizable the observations made for the CIFAR-10 in-distribution to the experimental results of CIFAR-100.

Like the experimental setup for CIFAR-10, we use the ResNet-34 classifier architecture to train our model with CIFAR-100. Moreover, for performance comparison, a baseline ReLU activation-based architecture is trained and models with log-softmax with entropy loss factors 0 and $1\mathrm{e}{-2}$. Since we did not observe a significant benefit of performing the entropy minimization for OOD detection for CIFAR-10 experiments. We do not perform extensive experiments with different entropy minimizations for CIFAR-100. For the out-distribution datasets for measuring the OOD detection performance, we use CIFAR-10, TinyImageNet, LSUN, SVHN.

4.2.1 Classification performance of log-softmax models

Implementation Details

We train the baseline ResNet-34 architecture with ReLU activation with cross-entropy loss as in equation 2.4. For the ResNet-34 models with log-softmax activation, the adopted loss is introduced in equation 2.5. We observe that a similar training setup followed for CIFAR-10 can be adapted to train the ResNet-34 classifier with the CIFAR-100 dataset. We train the models with AdamW optimizer with a base learning rate of 10^{-4} for the baseline model and log-softmax model 10^{-4}. Moreover, we use One-Cycle learning rate schedulers with linear anneal strategy and train for 95 epochs and a batch size of 256. For the One-Cycle learning rate scheduler, we use the maximum learning rate of 10^{-2} for the baseline model and learning rate of

Network Architecture: ResNet34 In-distribution: CIFAR10				
log-Softmax	Entropy Loss Factor(λ)	Out-distribution dataset	AUROC(↑)	
			MSP/Mahalanobis (Best)	Intermediate Layer Aggregation (Best)
-NA-		CIFAR100	84.89 (MSP)	-NA-
		TinyImageNet	99.46 (Maha)	-NA-
		LSUN	99.75 (Maha)	-NA-
		SVHN	98.62 (Maha)	-NA-
yes	0	CIFAR100	85.20 (MSP)	86.72 (avg of ASP layer4.3)
		TinyImageNet	99.04 (Maha)	98.96 (min of MiSP layer1.2)
		LSUN	99.46 (Maha)	99.76 (min of MiSP layer1.2)
		SVHN	98.85 (Maha)	98.66 (min of MiSP layer1.2)
yes	1e-2	CIFAR100	83.26 (MSP)	87.44 (min of MiSP layer4.3)
		TinyImageNet	98.90 (Maha)	98.85 (min of MiSP layer1.2)
		LSUN	99.45 (Maha)	99.66 (min of MiSP layer1.2)
		SVHN	98.19 (Maha)	97.89 (min of MiSP layer1.1)
yes	1e-1	CIFAR100	84.68 (MSP)	85.51 (avg of MiSP layer4.3)
		TinyImageNet	97.85 (Maha)	94.27 (min of MiSP layer1.2)
		LSUN	98.53 (Maha)	97.59 (min of MiSP layer1.2)
		SVHN	98.34 (Maha)	96.42 (min of MiSP layer1.1)

Tab. 4.3.: MSP/Mahalanobis-ensemble method(Maha) vs intermediate layer aggregation evaluations. The aggregation results are not available for the baseline ReLU model(indicated by -NA-).

10^{-3} for the log-softmax models. We also use the same data augmentation strategy, random horizontal flip, and random cropping as performed for CIFAR-10.

Classification performance

The classifier models are trained with 50000 training images of CIFAR-100, and the top-1 accuracy is measured for 10000 test samples. The top-1 accuracy of the models is presented in table 4.4. We note that the log-softmax models retain classification performance. The best log-softmax model has a top-1 accuracy of 74.25% while the baseline model has an accuracy of 75.93%. As observed for the CIFAR-10 in-distribution dataset with higher entropy loss, the top-1 accuracy decreases.

4.2.2 OOD performance for methods from literature

For all the OOD evaluation methods we follow the same implementation methodology as followed for the CIFAR-10 as in-distribution dataset.

MSP: As noted for the CIFAR-10 as an in-distribution, the MSP performance degrades for the models with log-softmax activation compared to the baseline ReLU model. The results are presented in table 4.5. This shows that log-softmax as intermediate activation is not beneficial for the final layer MSP method.

Mahalanobis-distance based: With this evaluation for the log-softmax models in comparison to the baseline model, we observe that the OOD detection performance is retained(see table 4.5). We conclude that the intermediate features of log-softmax models are as beneficial as the ReLU based models for OOD detection. We note that the performance is poor with the Mahalanobis distance method for the 'near OOD' setting, i.e. when tested with CIFAR-10 as out-distribution. E.g., for the baseline ReLU architecture for CIFAR-10 as the OOD dataset, the AUROC is only 57.71 and for the log-softmax model 61.29. Which is very close to a chance detector. A similar observation was made for the 'near and far OOD' setting where for the in-distribution CIFAR-10 and out-distribution as CIFAR-100, the Mahalanobis-distance performance was poor in comparison to the 'far OOD' setting (see table 4.2).

MSP vs. MiSP: The Minimum Softmax probability(MiSP) as an OOD score was found to be a better OOD scoring technique in comparison to the MSP for in-distribution CIFAR-10(see table 4.2). Here for the CIFAR-100 as the in-distribution, we find that the observation is generalizable. For most of the models, the MiSP is equally good or better than the MSP method.

Network Architecture: ResNet34 In-distribution:CIFAR100		
log-softmax	Entropy Loss factor(λ)	Top-1 accuracy(%)
-NA-	-NA-	75.93
yes	0	74.25
yes	1e-2	73.39

Tab. 4.4.: Table showing the classification performance of baseline model, model with log-softmax activation and model with entropy minimization when trained with CIFAR100.

4.2.3 Intermediate layerwise OOD evaluations

For the log-softmax models trained with CIFAR-100, all the nine intermediate log-softmax aggregations methods were performed. The implementation adopted is the same for the CIFAR-10 models. The general observation made here is again that the min of MiSP is a better aggregation method than the others. In table 4.5 we summarize the best aggregation method for each of the log-softmax models and the out-distribution datasets.

We find that for the 'near OOD' out-distribution dataset CIFAR-10, the aggregation performance is poor with an AUROC of 62.5 and 63.77 for the log-softmax model with Entropy loss 0 and $1e{-}2$ respectively. However, when compared with the Mahalanobis-distance method, the performances of the aggregation methods are slightly better. Still, it is poorer in comparison to the MSP method. Nevertheless, another interesting observation is that for this difficult 'near OOD' setting, the best performing layer was layer4.3. A similar observation was made for the CIFAR-10 as the in-distribution case for the 'near and far OOD' setting, i.e., CIFAR-100 as out-distribution. Thus, the later layers of the network were more beneficial for OOD sample detection.

But considering the 'far OOD' datasets SVHN, TinyImageNet, and LSUN. We observe that the log-softmax aggregation-based OOD evaluation performs as well or even better than the Mahalanobis distance-based method. Moreover, it performs much better than the MSP method. Another general observation is that similar to what was observed for the CIFAR-10 as the in-distribution, the initial layers (layers1.1 and layers1.2) are more beneficial for OOD detection for far away out-distribution datasets.

Network Architecture: ResNet34 In-distribution: CIFAR100						
log-Softmax	Entropy Loss factor(λ)	Out-distribution dataset	AUROC(↑)			
			MSP	Mahalanobis	MiSP	Intermediate Aggregation (Best)
-NA-		CIFAR10	73.31	57.71	72.43	-NA-
		TinyImageNet	74.92	98.57	75.8	-NA-
		LSUN	75.15	99.21	78.3	-NA-
		SVHN	81.50	96.73	94.28	-NA-
yes	0	CIFAR10	70.56	61.29	70.49	62.5(avg of MiSP layer4.3)
		TinyImageNet	70.11	97.87	70.79	99.3(min of MiSP layer1.1)
		LSUN	71.4	99.05	73.69	99.84(min of MiSP layer1.1)
		SVHN	73.95	97.32	86.59	98.41(min of MiSP layer1.1)
yes	1e-2	CIFAR10	71.25	57.49	71.44	63.77(min of MiSP layer4.3)
		TinyImageNet	62.68	98.12	60.93	98.82(min of MiSP layer1.2)
		LSUN	69.27	98.88	69.36	99.65(min of MiSP layer1.2)
		SVHN	75.15	96.63	85.53	96.82(min of MiSP layer1.2)

Tab. 4.5.: Table showing OOD evaluation performance comparison for MSP,Mahalanobis,MiSP and aggregation methods.

5 Conclusion

In this thesis, we proposed introducing log-softmax activation to the intermediate layers of the image classifier. With such an architectural change, the classifiers can retain classification performance and at the same time enable to development of cheap out-of-distribution detection methods with these intermediate log-softmax outputs.

We find it possible to introduce the log-softmax activations after multiple intermediate layers of a residual skip connection-based architecture. Furthermore, we show that it is possible to retain classification accuracy. Once we have the classification performance guarantee for log-softmax models, we evaluate how well these models perform for a final layer softmax score-based MSP method and the Mahalanobis-distance-based method that leverages features from multiple intermediate layers of the classifier.

With the MSP method, we observe that the performance degrades for the log-softmax models. This indicates that the introduction of log-softmax activations for intermediate layers does not benefit the final layer softmax-based MSP method. And for the Mahalanobis-distance-based method, we observe that the log-softmax models perform equally competitively with the baseline ReLU based model for OOD detection. This indicates that the intermediate features of the log-softmax models enable OOD detection with the same competence as the ReLU based classifiers. Contrary to the MSP method and its extensions proposed in literature which builds upon the maximum probability score of the final layer, we explore the minimum softmax probability(MiSP) as an OOD score for the final layer softmax-based evaluation. We surprisingly find that this way of OOD scoring performs even better than the MSP method.

For leveraging the intermediate layer log-softmax outputs for OOD detection, we introduce different aggregation methods for this multidimensional log-softmax outputs to derive an OOD score. This was inspired by the MSP method, which used the maximum softmax probability at the final layer as an OOD score. Similarly, for this multidimensional log-softmax outputs, we develop nine aggregation methods based on computing the maximum/minimum/average of these outputs. Finally, we evaluated OOD detection with these newly introduced aggregation methods per

intermediate layer and find that some of these methods are beneficial to use as an intermediate layer OOD scoring method with log-softmax activation.

We find that the min of MiSP to be a very beneficial log-softmax score aggregation method for layerwise OOD detection. With this method, different layers enable different OOD performances for out-distribution datasets depending on the amount of data distributional shift. We find that for the 'near OOD'(in-distribution: CIFAR-100 vs. out-distribution: CIFAR-10) and 'near and far OOD' dataset (in-distribution: CIFAR-10 vs. out-distribution: CIFAR-100), later network layers to be more beneficial. The network's early layers are more useful for the 'far OOD' dataset (in-distribution: CIFAR-10 vs. out-distribution: SVHN). However, with the min of MiSP based scoring, the 'near OOD' setting is still a challenging task.

Further, we observe that with a min of MiSP aggregation method, OOD evaluation performed at a single intermediate layer is equally competitive as the Mahalanobis-distance method, which uses an ensemble of features from multiple layers. Another benefit of this aggregation method is that it does not need further training to derive an OOD score, as was the case in the Mahalanobis-distance method or access to the OOD dataset to tune hyperparameters. Thus, the aggregation methods serve as a cheap OOD detection method for models integrated with log-softmax intermediate activation.

6 Discussion

This work showed that log-softmax as an intermediate activation for the classifiers introduces a way to leverage the intermediate layer outputs to develop cheap OOD detection methods. This section discusses some of the challenges in adopting this methodology for deploying such a classifier for open-world OOD detection.

Many of the state-of-art classifiers rely on ReLU activation for their architecture development. But in our case, to make the intermediate log-softmax aggregation methods useable for OOD detection, the network architecture should be integrated with log-softmax activation after multiple intermediate layers. Further, this log-softmax integration for the classifier architecture comes with the additional overhead of training the network again with finding the suitable combination of optimizers and other hyperparameters. However, in contrast to the existing methods MSP, ODIN, or the Mahalanobis-distance-based methods are post-training techniques that are easily extensible to any architecture.

Another challenge in deploying the intermediate log-softmax aggregation-based OOD evaluation methods is, it is not generalizable across the datasets. Depending on the out-distribution dataset distributional distance to the in-distribution, different layers in the network are beneficial for different datasets. This can be overcome by training an additional logistic regression detector to generate an OOD score with features from multiple intermediate layers similar to the Mahalanobis-distance method introduced in [Lee+18]. Furthermore, we also noted in section 4.1.3 that the min of MiSP distribution spectrum is not consistent for all the in- and out-distribution dataset configurations. We observed that when CIFAR-10 is the in-distribution, and for CIFAR-100 and SVHN as out-distribution datasets, the min of MiSP values are in the higher ranges than the min of MiSP values of CIFAR-10 samples. For the out-distribution datasets, TinyImageNet and LSUN, the min of MiSP values are in the lower ranges. This poses a challenge, especially when considering an open-world setting with no knowledge of the type of distribution for an input sample. In this case, it isn't easy to predetermine the threshold for the OOD score to identify as in- and out-of-distribution input.

7 Future Work

The introduction of log-softmax activation for intermediate layers opens multiple avenues for enhancing the OOD detection performance. In OOD literature, various methods have been proposed which develop on the final layer softmax probabilities. We deem many of these methods could be extended to the intermediate layer log-softmax outputs.

The MSP method served as the baseline for softmax-based OOD methods. A simple extension proposed was the ODIN method, which performs the post-training processing of temperature scaling the softmax and adds small noise to the input image and showed that it could improve the in- and out-distribution MSP score separation. These techniques could be applied for the intermediate log-softmax outputs to enhance the performance. Further another successful method was the Outlier Exposure which used an additional auxiliary out-distribution dataset and fine-tuned the network to make the outlier distribution uniform. We think this Outlier Exposure fine-tuning objective introduced for the intermediate log-softmax layers and entropy minimization for the in-distribution dataset could further help separate the in- and out-distribution. We observed with aggregation of log-softmax evaluation for intermediate layers OOD performance evaluated at a single layer had a competitive performance as Mahalanobis-distance method. So, we propose to ensemble the scores from multiple intermediate layers and derive an OOD score. Still, it could be cheaper than the Mahalanobis-distance method as we do not have to fit the class-conditional Gaussian distribution.

Through our experiments, another interesting observation made was that the minimum softmax probabilities-based scoring works as a better OOD scoring technique than the maximum softmax probabilities. Considering the final layer softmax method, we observed the MiSP be a better scoring technique than MSP. Moreover, for the intermediate layers aggregation evaluation, we observed min of MiSP to be a better scoring technique. So, it is not clear yet why the minimum softmax spectrum is beneficial for OOD. This is an area that needs further studies to confirm why MiSP works.

We find that the log-softmax intermediate activations for a classifier can retain performance and enable OOD detection for intermediate layers by introducing cheap

techniques. There are numerous possibilities to study whether these observations made are robust across varying network architectures. We studied only for a residual connection-based ResNet-34 architecture. The benefits of log-softmax observed here can be studied for a dense connection-based architecture, e.g. DesnseNet121 [Hua+18] or wide residual network architecture, e.g., WideResNet-28-10 [ZK17]. Further, the ResNet-34 does not have bottleneck blocks in the residual layers. So studies can also be done based on this. E.g., the ResNet-50 has the same number of residual layers as ResNet-34 but has an additional bottleneck layer in each residual layer. Another possibility is exploring the depth of the network for OOD detection benefits with the log-softmax introduction, e.g., ResNet-18 vs. ResNet-101.

This work explored the entropy minimization for the intermediate log-softmax outputs in an unsupervised way. However, this does not seem to have significant benefits for OOD detection. Another possibility is to apply the supervised entropy minimization. This can be applied per spatial location for the intermediate log-softmax outputs, which is applied along the channel given that we have channel depth the same as the number of classes. In [LFS20] proposed introduction of supervised bottleneck layer of drastically reduced dimensionality after certain blocks in image segmentation network and have shown that it is possible to retain segmentation performance. For the image classifier networks, such a bottleneck layer of lower dimensionality can be introduced after certain blocks and verified whether the classifier retains performance. Furthermore, the OOD detection benefits of these layers can be studied.

Bibliography

[Abd+19] Vahdat Abdelzad, Krzysztof Czarnecki, Rick Salay, et al. *Detecting Out-of-Distribution Inputs in Deep Neural Networks Using an Early-Layer Output*. 2019. arXiv: 1910.10307 [cs.LG] (cit. on p. 7).

[Amo+16] Dario Amodei, Chris Olah, Jacob Steinhardt, et al. *Concrete Problems in AI Safety*. 2016. arXiv: 1606.06565 [cs.AI] (cit. on p. 1).

[Bev+18] Petra Bevandić, Ivan Krešo, Marin Oršić, and Siniša Šegvić. *Discriminative out-of-distribution detection for semantic segmentation*. 2018. arXiv: 1808.07703 [cs.CV] (cit. on p. 5).

[Bon17] Rodolfo Bonnin. *Machine Learning for Developers*. 2017 (cit. on p. 31).

[CT18] Jaison Chacko and B. Tulasi. "Semantic image annotation using convolutional neural network and WordNet ontology". In: *International Journal of Engineering and Technology(UAE)* 7 (Aug. 2018), pp. 56–60 (cit. on p. 31).

[Che+20] Changjian Chen, Jun Yuan, Yafeng Lu, et al. *OoDAnalyzer: Interactive Analysis of Out-of-Distribution Samples*. 2020. arXiv: 2002.03103 [cs.HC] (cit. on p. 2).

[DG06] Jesse Davis and Mark Goadrich. "The Relationship between Precision-Recall and ROC Curves". In: *Proceedings of the 23rd International Conference on Machine Learning*. ICML '06. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2006, pp. 233–240 (cit. on p. 7).

[Den+09] Jia Deng, Wei Dong, Richard Socher, et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255 (cit. on pp. 29, 30).

[Eyk+18] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, et al. *Robust Physical-World Attacks on Deep Learning Models*. 2018. arXiv: 1707.08945 [cs.CR] (cit. on p. 1).

[Gaw+21] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, et al. *A Survey of Uncertainty in Deep Neural Networks*. 2021. arXiv: 2107.03342 [cs.LG] (cit. on p. 1).

[GE19] Yonatan Geifman and Ran El-Yaniv. *SelectiveNet: A Deep Neural Network with an Integrated Reject Option*. 2019. arXiv: 1901.09192 [cs.LG] (cit. on p. 5).

[GBB11] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. "Deep Sparse Rectifier Neural Networks." In: *AISTATS*. Ed. by Geoffrey J. Gordon, David B. Dunson, and Miroslav Dudík. Vol. 15. JMLR Proceedings. JMLR.org, 2011, pp. 315–323 (cit. on pp. 3, 20, 35).

[GSS15] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. *Explaining and Harnessing Adversarial Examples*. 2015. arXiv: 1412.6572 [stat.ML] (cit. on pp. 1, 10).

[Guo+17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. *On Calibration of Modern Neural Networks*. 2017. arXiv: 1706.04599 [cs.LG] (cit. on p. 14).

[He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016 (cit. on pp. 20, 21, 27).

[HAB19] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. *Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem*. 2019. arXiv: 1812.05720 [cs.LG] (cit. on p. 1).

[Hen+20] Dan Hendrycks, Steven Basart, Mantas Mazeika, et al. *Scaling Out-of-Distribution Detection for Real-World Settings*. 2020. arXiv: 1911.11132 [cs.CV] (cit. on pp. 3, 20, 71).

[HG18] Dan Hendrycks and Kevin Gimpel. *A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks*. 2018. arXiv: 1610.02136 [cs.NE] (cit. on pp. 3–7, 9, 23).

[HMD19] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. *Deep Anomaly Detection with Outlier Exposure*. 2019. arXiv: 1812.04606 [cs.LG] (cit. on pp. 3, 5, 8).

[Hsu+20] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. *Generalized ODIN: Detecting Out-of-distribution Image without Learning from Out-of-distribution Data*. 2020. arXiv: 2002.11297 [cs.CV] (cit. on pp. 6, 17).

[Hua+18] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. *Densely Connected Convolutional Networks*. 2018. arXiv: 1608.06993 [cs.CV] (cit. on p. 58).

[Kri09] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. Tech. rep. 2009 (cit. on pp. 13, 29).

[LPB17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. *Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles*. 2017. arXiv: 1612.01474 [stat.ML] (cit. on p. 5).

[Lee+18] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. *A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks*. 2018. arXiv: 1807.03888 [stat.ML] (cit. on pp. 4–7, 10, 13, 17, 30, 37, 39, 40, 55, 70).

[LLS20] Shiyu Liang, Yixuan Li, and R. Srikant. *Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks*. 2020. arXiv: 1706.02690 [cs.LG] (cit. on pp. 3–7, 9, 23).

[Liu+21] Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. *Energy-based Out-of-distribution Detection*. 2021. arXiv: 2010.03759 [cs.LG] (cit. on pp. 5, 13, 17).

[LFS20] Max Maria Losch, Mario Fritz, and Bernt Schiele. "Semantic Bottlenecks: Quantifying and Improving Inspectability of Deep Representations". In: *Pattern Recognition - 42nd DAGM German Conference, DAGM GCPR 2020, Tübingen, Germany, September 28 - October 1, 2020, Proceedings*. Ed. by Zeynep Akata, Andreas Geiger, and Torsten Sattler. Vol. 12544. Lecture Notes in Computer Science. Springer, 2020, pp. 15–29 (cit. on pp. 3, 19, 20, 58).

[LH19] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2019. arXiv: 1711.05101 [cs.LG] (cit. on p. 34).

[Moh+20] Sina Mohseni, Mandar Pitale, Jbs Yadawa, and Zhangyang Wang. "Self-Supervised Learning for Generalizable Out-of-Distribution Detection". In: *AAAI*. 2020 (cit. on pp. 5, 7).

[NH10] Vinod Nair and Geoffrey E. Hinton. "Rectified Linear Units Improve Restricted Boltzmann Machines". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML'10. Haifa, Israel: Omnipress, 2010, pp. 807–814 (cit. on p. 20).

[Net+11] Yuval Netzer, T. Wang, A. Coates, et al. "Reading Digits in Natural Images with Unsupervised Feature Learning". In: 2011 (cit. on pp. 13, 29, 32).

[NYC15] Anh Nguyen, Jason Yosinski, and Jeff Clune. *Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images*. 2015. arXiv: 1412.1897 [cs.CV] (cit. on p. 1).

[Pap+21] Aristotelis-Angelos Papadopoulos, Mohammad Reza Rajati, Nazim Shaikh, and Jiamian Wang. "Outlier exposure with confidence control for out-of-distribution detection". In: *Neurocomputing* 441 (2021), pp. 138–150 (cit. on pp. 5, 6).

[Ren+19] Jie Ren, Peter J. Liu, Emily Fertig, et al. *Likelihood Ratios for Out-of-Distribution Detection*. 2019. arXiv: 1906.02845 [stat.ML] (cit. on p. 6).

[SO20] Chandramouli Shama Sastry and Sageev Oore. *Detecting Out-of-Distribution Examples with In-distribution Examples and Gram Matrices*. 2020. arXiv: 1912.12510 [cs.LG] (cit. on p. 7).

[ST18] Leslie N. Smith and Nicholay Topin. *Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates*. 2018. arXiv: 1708.07120 [cs.LG] (cit. on p. 34).

[Win+20] Jim Winkens, Rudy Bunel, Abhijit Guha Roy, et al. *Contrastive Training for Improved Out-of-Distribution Detection*. 2020. arXiv: 2007.05566 [cs.LG] (cit. on pp. 5–7, 14, 15, 29–32, 48).

[Yu+16] Fisher Yu, Ari Seff, Yinda Zhang, et al. *LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop*. 2016. arXiv: 1506.03365 [cs.CV] (cit. on pp. 16, 29).

[ZK17] Sergey Zagoruyko and Nikos Komodakis. *Wide Residual Networks*. 2017. arXiv: 1605.07146 [cs.CV] (cit. on p. 58).

Webpages

[@Gar20] Cristian Garcia. *Sample images from CIFAR-100 dataset*. 2020. URL: https://cgarciae.github.io/dataget/datasets/image/cifar100/#info (cit. on p. 30).

[@Ost19] Stephan Osterburg. *Images of CIFAR-10 all classes*. 2019. URL: https://stephan-osterburg.gitbook.io/coding/coding/ml-dl/tensorfow/untitled-2/load-and-explore-cifar10-dataset (cit. on p. 29).

List of Figures

List of Tables

Appendix

A

A.1 OOD evaluation performance

Since there was no significant effect of the entropy minimization for OOD detection, in table 4.2 limited results were presented for the MSP, MiSP, and Mahalanobis distance evaluation for the log-softmax models with different entropy loss factors. Here, in table A.1 the detailed results of all the entropy loss factors experimented for the log-softmax models are presented.

A.2 Mahalanobis layerwise evaluation.

Additionally, the Mahalanobis-distance-based evaluation performed per layer for the baseline ReLU based model can be applied to the log-softmax models' intermediate layers. This evaluation comparison for the vanilla ResNet-34 model with the models with log-softmax intermediate activation is presented in figure A.1.

We observe that for the SVHN, TinyImageNet and LSUN as the out-distribution datasets, the Mahalanobis layerwise evaluation performs similarly as for the baseline model and the log-softmax models in the 1^{st},2^{nd} and 3^{rd} residual blocks. But for the TinyImageNet and LSUN datasets, the layers in 4^{th} block have better performance for the log-softmax models in comparison to the baseline models. However, this OOD detection performance at the 4^{th} block for the log-softmax models is not that significant in comparison to the performance in the layers from 2^{nd} and 3^{rd} residual blocks.

Considering the CIFAR-100 out-distribution dataset, the log-softmax models perform significantly better than the baseline model in the residual block 3 and 4. Still, the min of MiSP evaluated at the 4^{th} residual block have better OOD detection performance (see figure 4.2) in comparison to the Mahalanobis-distance based layerwise evaluation.

Network Architecture: ResNet34 In-distribution: CIFAR10					
log-Softmax	Entropy Loss factor(λ)	Out-distribution dataset	AUROC		
			MSP	Mahalanobis Distance	MiSP
-NA- (from [Lee+18])		CIFAR100	-NA-	-NA-	-NA-
		TinyImageNet	91.0	99.50	-NA-
		LSUN	91.0	99.70	-NA-
		SVHN	89.90	99.10	-NA-
-NA- (Ours)		CIFAR100	84.89	66.06	88.19
		TinyImageNet	92.50	99.46	95.22
		LSUN	93.48	99.75	95.66
		SVHN	92.08	98.62	96.54
yes	0	CIFAR100	85.20	73.51	88.39
		TinyImageNet	83.82	99.04	89.80
		LSUN	88.48	99.46	93.53
		SVHN	90.27	98.85	94.70
yes	1e-3	CIFAR100	84.49	72.59	86.88
		TinyImageNet	88.88	98.42	93.21
		LSUN	91.48	99.34	94.43
		SVHN	91.48	98.59	95.20
yes	5e-3	CIFAR100	84.68	74.70	85.10
		TinyImageNet	88.43	98.88	86.36
		LSUN	90.40	99.46	91.05
		SVHN	89.90	98.36	96.15
yes	1e-2	CIFAR100	83.26	74.07	86.23
		TinyImageNet	83.35	98.90	88.43
		LSUN	89.89	99.45	95.32
		SVHN	92.43	98.19	97.41
yes	5e-2	CIFAR100	84.42	66.73	84.78
		TinyImageNet	85.76	98.88	88.43
		LSUN	88.42	99.42	90.85
		SVHN	88.55	98.15	90.62
yes	1e-1	CIFAR100	84.68	65.85	85.78
		TinyImageNet	88.43	97.85	93.07
		LSUN	90.16	98.53	94.31
		SVHN	89.89	98.34	91.41

Tab. A.1.: Table showing the MSP,Mahalanobis distance based and MiSP OOD evaluation results for our models in comparison to the baseline model reported in the paper [Lee+18]. ↑ indicates higher the value the performance is better.

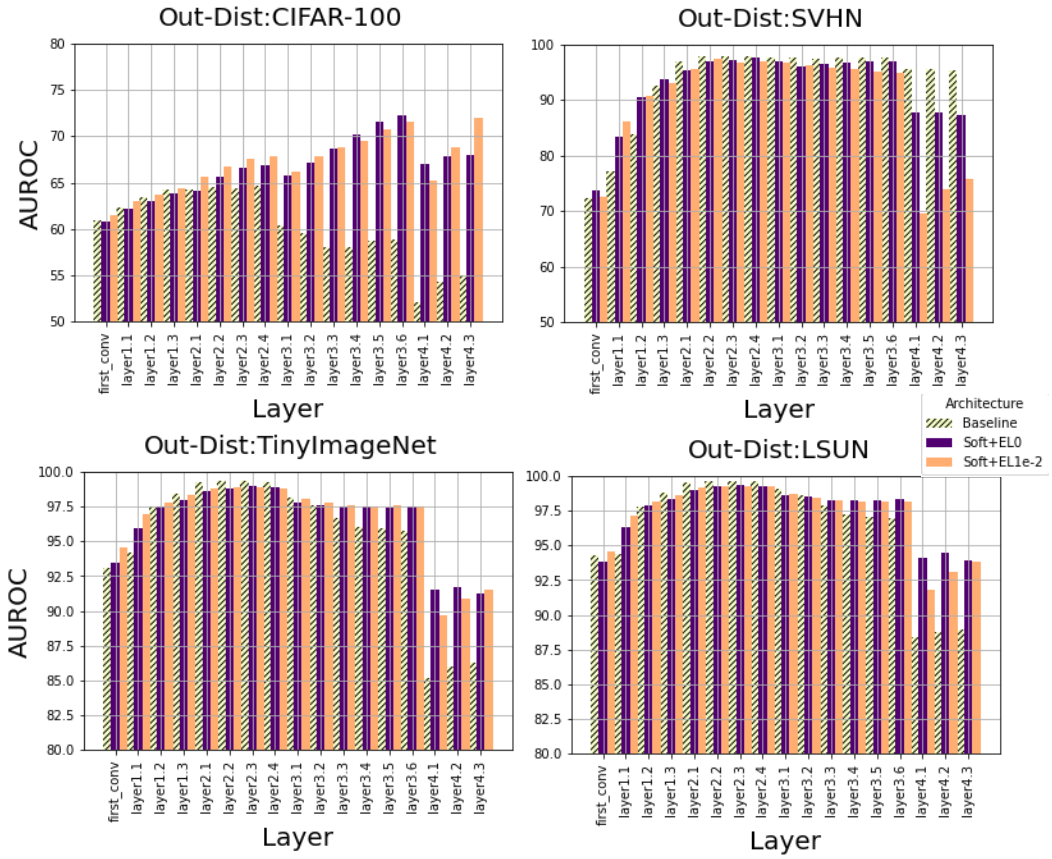


Fig. A.1.: Figure showing the comparison of Mahalanobis layerwise evaluation for the baseline model and log-softmax models with entropy loss factor 0 and $1e{-}2$. The base architecture is ResNet-34. The in-distribution dataset is CIFAR-10.

A.3 Pixel-wise OOD evaluation for intermediate layer log-softmax outputs.

In [Hen+20] the authors introduced the utility of performing MSP per pixel for pixel-wise OOD detection for an image segmentation output. In section 3.2.4 we discussed the applicability of performing the pixel-wise MSP/MiSP/ASP for the intermediate log-softmax outputs. We perform this experiments for the log-softmax models with entropy loss factor 0 and $1e{-}2$. The comparison for MSP per pixel for the CIFAR-10 as in-distribution and CIFAR-100 as the out-distribution is shown in figure A.2 and for SVHN as out-distribution is shown in figure A.3.

Considering the CIFAR-100 as an out-distribution dataset observing the AUROC values for the MSP per pixel for both the log-softmax model with entropy loss 0 and $1e{-}2$, till layer3.6 we observe that the values are not that significant. Most of the

AUROC values are close to the chance value of 50. But we see that in layer3.6 for the model with entropy loss factor $1e{-}2$, the pixel which is most significant for OOD detection is concentrated around the center. Considering layer4.3, we see that the pixel with the best AUROC for the entropy loss factor $1e{-}2$ has an AUROC value around 87, and for the model with entropy loss factor 0, we have the best AUROC value around 67. This is a significant improvement for the model with entropy loss factor $1e{-}2$.

Considering the SVHN as an out-distribution dataset, we observe a similar pattern as for CIFAR-100 out-distribution dataset. Till layer3.6, the values are not that significant for both the log-softmax models. The MSP per pixel AUROC values are around the 60-70 range. But in layer4.3, the top per pixel AUROC values are high in comparison to the other layers. Similar to the observation made in the case of CIFAR-100, we observe that the model with entropy loss factor $1e{-}2$ have a pixel with the best AUROC of around 94, which is better in comparison to the model with entropy loss factor 0 for which the best AUROC is approximately 87. This improvement might be due to the entropy minimization, which might have more effect in the final layers of the network than the early layers.

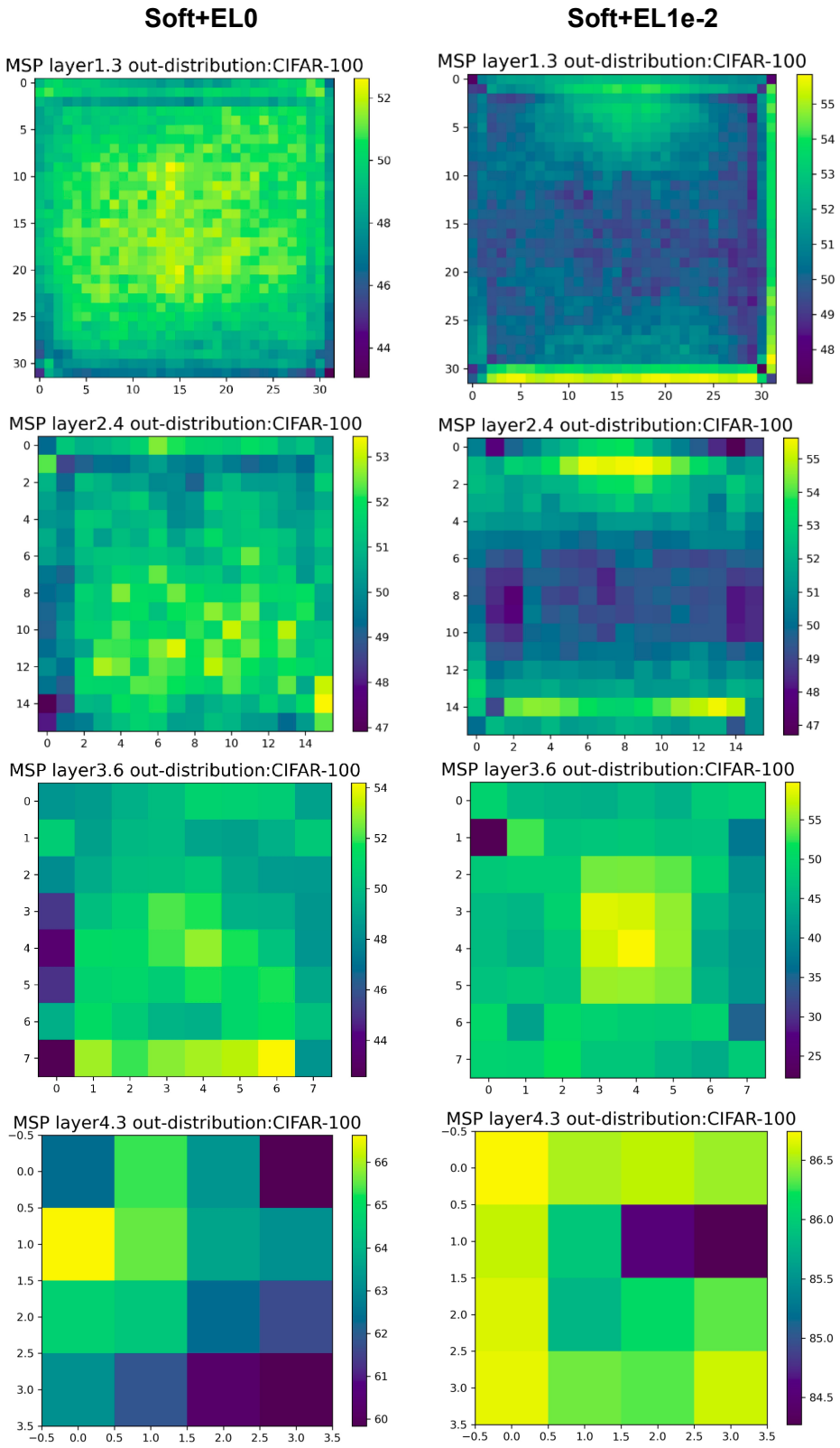


Fig. A.2.: Figure shows heatmap of AUROC values when the MSP per pixel OOD evaluation is performed. The scale on the right of each heatmap shows the ranges of AUROC values. The comparison is presented for the ResNet-34 model with log-softmax activation with entropy loss factor 0(Soft+EL0) and $1e{-}2$(Soft+1e-2). For this comparison, only the last layers from each of the residual blocks in the ResNet-34 architecture is considered. The following is the dimension of the heatmap [layer1.3: 32×32,layer2.4: 16×16,layer3.6: 8×8,layer4.3:4×4]. The in-distribution dataset is CIFAR-10 and the out distribution is CIFAR-100.

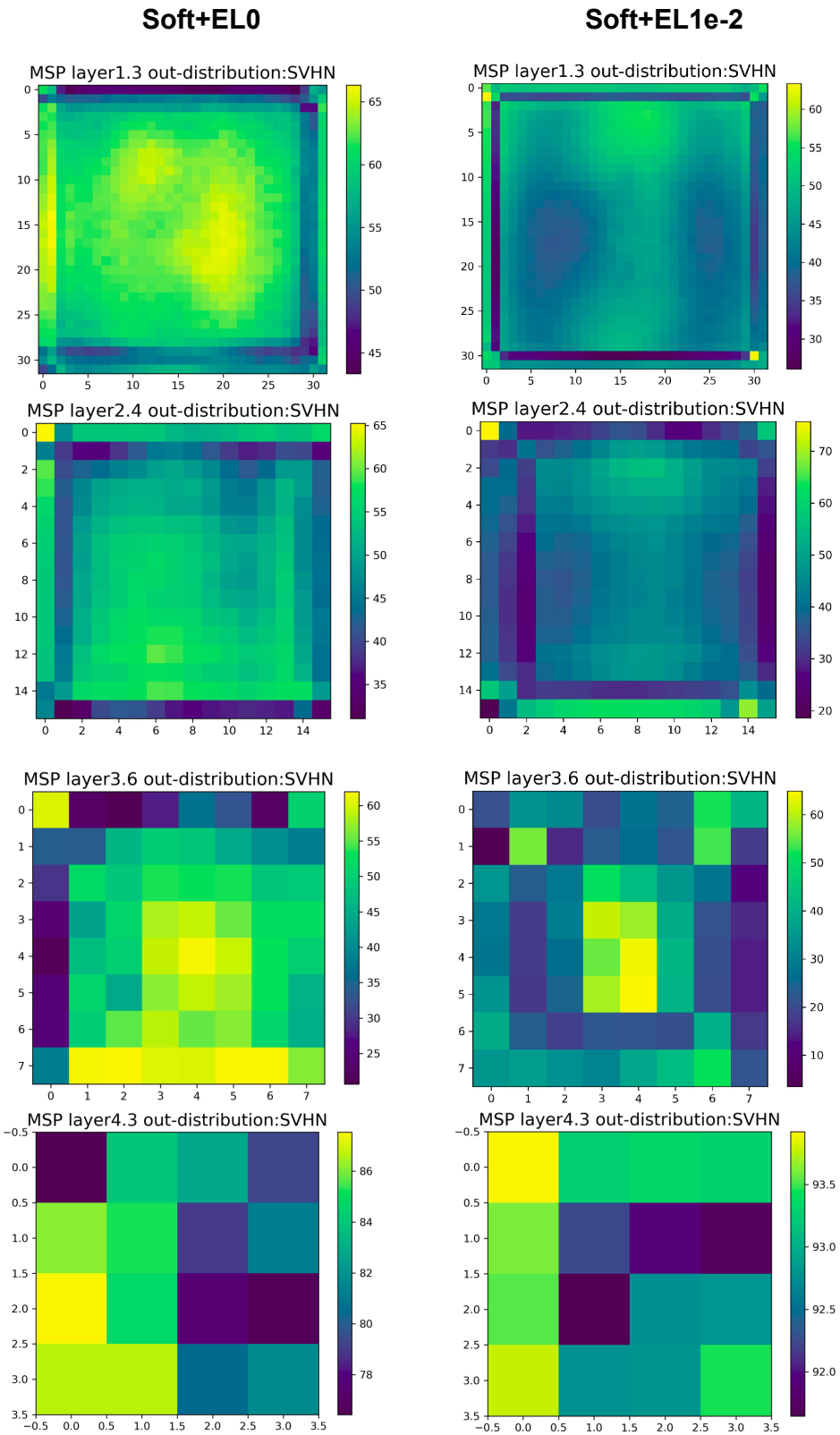


Fig. A.3.: Figure shows heatmap of AUROC values when the MSP per pixel OOD evaluation is performed. The scale on the right of each heatmap shows the ranges of AUROC values. The comparison is presented for the ResNet-34 model with log-softmax activation with entropy loss factor 0(Soft+EL0) and $1e{-}2$(Soft+EL1e-2). For this comparison, only the last layers from each of the residual blocks in the ResNet-34 architecture is considered. The following is the dimension of the heatmap [layer1.3: 32×32,layer2.4: 16×16,layer3.6: 8×8,layer4.3:4×4]. The in-distribution dataset is CIFAR-10 and the out distribution is SVHN.

Declaration

I, Praveen Annamalai Nathan, declare that this thesis titled, "Generalization of MSP based out-of-distribution detection to intermediate convolutional layers", and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

_____

Date: August,2021, Kaiserslautern

_____