Smart Media Journal / Vol.9, No.3 / ISSN:2287-1322
https://dx.doi.org/10.30693/SMJ.2020.9.3.80

Removing Out – Of – Distribution Samples on Classification Task

Thanh-Vu Dang*, Hoang-Trong Vo*, Gwang-Hyun Yu*,
Ju-Hwan Lee*, Huy-Toan Nguyen*, Jin-Young Kim*

Abstract

Out – of – distribution (OOD) samples are frequently encountered when deploying a classification model in plenty of real-world machine learning-based applications. Those samples are normally sampling far away from the training distribution, but many classifiers still assign them high reliability to belong to one of the training categories. In this study, we address the problem of removing OOD examples by estimating marginal density estimation using variational autoencoder (VAE). We also investigate other proper methods, such as temperature scaling, Gaussian discrimination analysis, and label smoothing. We use Chonnam National University (CNU) weeds dataset as the in – distribution dataset and CIFAR-10, CalTeach as the OOD datasets. Quantitative results show that the proposed framework can reject the OOD test samples with a suitable threshold.

Keywords : Out of distribution detection | Classification | Neural Networks | Statistical Modeling

I. INTRODUCTION

Image classification is a supervised machine learning problem where given an image, the learning model can output its identity. Particularly, vision-based classification is considerately important since it is the primary step for further operations, for example, biometric security based on face or fingerprint images. Object classification is also a core component of smart systems (e.g., smart farm, smart factory) where self-propelled robots and learning models are integrated to supervise a process. Over a decade, image classification methods have developed significantly to cope with the advancement of hardware and especially learning models.

Research on image recognition using handcrafted features and sallow learning has a long tradition. Well-known handcraft features such as HOG [1], SIFT/SURF [2], LBP [3] were frequently used as image descriptors and classified by single learning models (e.g support vector machine, logistic regression) or ensemble models (e.g random forest, AdaBoost). Previous studies have revealed that complex and large datasets are usually the most problematic to image classification, where simpler handcraft features are not enough to represent the visual structure of objects.

A series of recent studies have indicated that Deep Learning approaches have outperformed traditional methods in image recognition. Especially, Convolutional Neural Networks (CNNs) can automatically learn coarse to fine features via the hierarchical mechanism of stacked convolutional layers [4]. Furthermore, ImageNet [5], a large-scale dataset, and the corresponding classification challenge have promoted the development of Deep Learning [6]. Also, many studies have gradually proved that deeper models with appropriate architecture [7, 8] benefit to image classification.

However, Deep learning models usually behave wrongly with OOD samples. Precisely, given an OOD test example that is drawn far away from the in-distribution dataset, naïve models still assigned this sample the class probability with high confidence. Classifying an OOD sample into a known class is a considerable risk in practical applications. For example,

*This work was carried out with the support of "Cooperative Research Program for Agriculture Science and Technology Development (Project No. PJ01385501)" Rural Development Administration, Republic of Korea.
*Member, Department of ICT Convergence System Engineering, Chonnam National University , Republic of Korea

Manuscript: 2020.07.31
Confirmation of Publication: 2020.09.22
Corresponding Author: Jin-Young Kim
e-mail: beyondi@chonnam.ac.kr

a self-driving car mistakenly recognizing an unknown traffic signal as an existent signal and acting in a misguided way will cause tremendous troubles. Several methods have been brought to address the problem of OOD detection. Nguyen et al. revealed that Deep neural networks (DNNs) are easy to be fooled by adversarial visual attacks [9], such as pattern disturbance or gradient adjustment, when the samples are adjusted to be meaningless to human vision but entrusted with high class-probability from DNNs. Having been one of the simplest techniques, a study about label smoothing of Muller showed that training DNN models with smoothing labels would prevent the network from becoming overconfident [10]. Although the raw class posterior output from DNNs is usually overconfident, Hendrycks considered it as a measure to detect OOD samples [11]. The author also formulated a set of baseline evaluators that were used in this literature. Non-linear classifiers structured by fully-connected layers in DNNs tend to be ambiguous about the boundary between in and out of distribution regions. For that reason, to construct a reliable classifier, Lee et al. employed Gaussian discriminant analysis regarding the input of feature vectors extracted from trained models instead of end-to-end models alone trained with softmax neural classifiers [12]. Marginal likelihood estimation is a direct way to detect OOD samples since those samples have marginal probability lower than in-distribution samples. Ren suggested using PixelCNN to estimate the marginal likelihood of observation and calculated its likelihood ratios for OOD detection [13]. Likewise, Lee trained a confidence-calibrated classifier for detecting OOD samples. Their method based on the assumption that OOD samples distributed uniformly surround the real distribution, hence a modified loss function of the generative model could be employed to draw the OOD samples and train the classifier jointly [14].

Despite many studies, the research in the field of OOD detection remains limited to large-scale datasets. Therefore, this study focuses on the large-scale weeds classification, where a reliable classifier greatly requires a mechanism to detect samples unlikely sampled from the training distribution, while recent frameworks only focus on classification accuracy. To address this problem, our study examined possible techniques that can eliminate the OOD samples in real-time. Furthermore, we introduce VAE as a straightforward estimator for marginal likelihood and effective to reduce the statistical error. The rest of the paper is structured as follow:

- Section 2 gives insight into the problem of OOD detection, where valid techniques are introduced.
- Section 3 describes our framework and VAE method to detect OOD samples.
- Section 4 gives a brief description of the datasets and shows experimental results.
- Finally, the conclusion is given in section 5.

II. PRELIMINARIES

A. Problem definition

OOD detection is a problem that arises when a classifier fails to eliminate OOD samples from the in-distribution. In other words, the classifier behaves over confidently and assigns a high class-probability to the OOD examples. Formally, it can be formulated from a supervised learning problem. Let $\boldsymbol{X_{in}} \in \boldsymbol{\Omega}$ is a random variable distributed as $P_{in}(\boldsymbol{X})$, and $Y \in \{1,..,K\}$ is the corresponding random variable associating to the label of $\boldsymbol{X} \in \boldsymbol{\Omega}$, distributed as $P(Y|\boldsymbol{X})$. To estimate the joint data distribution $P_{in}(\boldsymbol{X},Y) = P(Y|\boldsymbol{X})P_{in}(\boldsymbol{X})$, ones sampling $\boldsymbol{X}_i$ independently identical from $P_{in}(\boldsymbol{X})$ and the corresponding label Y_i from $P(Y|\boldsymbol{X})$. A question arises when a new $\boldsymbol{X}$ is sampled by $P(\boldsymbol{X},Y) = P(Y|\boldsymbol{X})P(\boldsymbol{X})$, is it belongs to $P_{in}(\boldsymbol{X})$? If not, we denote it as $\boldsymbol{X}_{out}$, which is unlikely distributed as $P_{in}(\boldsymbol{X})$.

B. Label smoothing

Label smoothing is a training technique introduced by Szegedy [7] to increase classification accuracy. In contrast to "hard" targets encoded by one-hot encoding, label smoothing approaches with "soft" target:

$$p_k = P(Y = k|\boldsymbol{X} = \boldsymbol{x}) \quad (1)$$
$$= \begin{cases} \alpha & where\ x\ belongs\ to\ class\ k, \\ \dfrac{1-\alpha}{K-1} & otherwise \end{cases}$$

where p_k is the class probability of belonging to class k given an image x. In typical hard encoding, $p_k = 1$ if x belongs to class k, since $\sum_{k=1}^{K} p_k = 1$, the rest probabilities are "0", $p_{k\neq k} = 0$. As a result, the hard encoding is overconfident to the correct class and makes DDNs miscalibrated. A direct consequence of hard encoding is to squash out the probability of incorrect classes. Precisely, when minimizing the

cross-entropy of a variable X, $H(\boldsymbol{p},\hat{\boldsymbol{p}}) = \sum_{k=1}^{K} -p_k \log \hat{p}_k$, between the true probability p_k and the network's output $\hat{p}_k$; $H_{hard}(\boldsymbol{p},\hat{\boldsymbol{p}}) = -p_k \log \hat{p}_k$ with hard labels contains only $\hat{p}_k$, while $H_{soft}(\boldsymbol{p},\hat{\boldsymbol{p}}) = -\alpha \log \hat{p}_k - (1-\alpha)\sum_{h\neq k}^{K} \log \hat{p}_k$ with soft labels involves all $\hat{p}_k$ for backpropagation. Therefore, training a network with hard labels usually causes the correct logit to be larger than any incorrect logits. While label smoothing drives the difference between the correct logit and incorrect logits not overwhelming.

C. Temperature scaling

Temperature scaling is the simplest version of Platt scaling [15] when only one single scalar parameter $T > 0$ is used to scale all the class logits before being fed through the softmax layer.

$$\hat{p}_k = P\big(\hat{Y} = k \big| \boldsymbol{X} = \boldsymbol{x}\big) = \frac{\exp_k(f_\theta(\boldsymbol{x})/T)}{\sum_{k'=1}^{K} \exp_{k'}(f_\theta(\boldsymbol{x})/T)}, \quad (2)$$

where f_θ is the classification model parameterized by θ taking an image $\boldsymbol{x}$ as the input and producing the score vector $\boldsymbol{z} = f_\theta(\boldsymbol{x})$, also known as logits. The scaling factor T has a function that flattens the density $P\big(\hat{Y}\big|\boldsymbol{X}\big) = \frac{1}{K}, \forall k$ when $T \to \infty$ or collapsed $P\big(\hat{Y}\big|\boldsymbol{X}\big)$ to one point when $T \to 0$. Naturally, temperature scaling is a unique solution when finding the most uncertainty distribution that satisfied unbias estimation for $f_\theta(x_i), \forall i$. Another advantage of temperature scaling is keeping predictions of the model unchanged since the mode of density $P\big(\hat{Y}\big|\boldsymbol{X}\big)$ is consistent when scaling. Note that temperature scaling is only applied to the testing phase, meanings that $T = 1$ during the training process.

D. Gaussian discrimination analysis

The vanilla classifier integrated into DNNs is constructed by fully connected layers to form a non linear mapping before forging separable hyperplanes between classes. In terms of the discriminative model, training the network with cross-entropy loss function is analogous to assign the multinominal distribution explicitly to the posterior $P(Y|\boldsymbol{X}) = M_{(p_1,..,p_n)}(g(\boldsymbol{X}))$ to solve the inference problem of the class probability. Otherwise, in the generative approach, multivariate Gassausian distribution is assumed for the class conditional $P(g(\boldsymbol{X})|Y = c) = \mathcal{N}(g(\boldsymbol{X})|\boldsymbol{\mu_c}, \boldsymbol{\Sigma})$, where

$$\hat{\boldsymbol{\mu}}_c = \frac{1}{N_c}\sum_{x_i \in C} g(\boldsymbol{x}_i), \quad (3)$$

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{C}\sum_{x_i \in C}(g(\boldsymbol{x}_i) - \hat{\boldsymbol{\mu}}_c)(g(\boldsymbol{x}_i) - \hat{\boldsymbol{\mu}}_c)^T, \quad (4)$$

$\hat{\boldsymbol{\mu}}_c$ is the samples mean calculated over samples of class C, while $\hat{\boldsymbol{\Sigma}}$ is the tied covariance shared across all classes. $g(\boldsymbol{x})$ denotes for the feature extraction part inside a DNN. To obtain the feature extractor $g(\boldsymbol{x})$, we trained the network f_θ with vanilla softmax classifier and keep only the feature extraction part, as depicted in Figure 1. The experimental results show that those features in the representation space could still be separable. Meanwhile, the space output from fully connected layers in the softmax-based approach is usually overfitted to the labels. Therefore, solving the inference problem of class probability $P(Y|\boldsymbol{X})$ is not sufficient to gain insight into the separated boundary between in distribution and OOD region.

Furthermore, Gaussian discrimination analysis supplies an uncertainty metric based on estimating the covariances of class conditional densities. OOD samples can be detected by calculating the maximum value of Mahalanobis distance [12]

$$m(\boldsymbol{x}) = \max_c -(g(\boldsymbol{x}) - \hat{\boldsymbol{\mu}}_c)^T \hat{\Sigma}^{-1}(g(\boldsymbol{x}) - \hat{\boldsymbol{\mu}}_c) \quad (5)$$

In the representation space, the Mahalanobis distance from OOD samples to the center of classes will be larger than that of in distribution samples. Mahalanobis distance takes spreads of class density into account, which is not emphasized in the discriminative model. As a result, OOD samples can be eliminated by determining a threshold based on the Mahalanobis distance of in distribution samples.

III. PROPOSED FRAMEWORK

A. Classification model

In this work, we consider a deep neural network (DNN) as a classification model and particularly experiment with ResNet18 [8]. ResNet is a well-known architecture on image understanding tasks, which has been successfully employed or a back-bone model in plenty of research such as image classification, object detection, and image segmentation. For a brief explanation, ResNet architecture consists of two core parts, feature extraction and class probability calculation, as shown in Figure 1. This network organizes information flowing from one layer to the next with the use of shortcuts so that the gradient does not vanish when traveling through a deep process. Besides, the classification mechanism in ResNet is vanilla, in which a series of fully connected layers is used to model a non linear classifier with softmax class probability $\hat{p}_k = P\big(\hat{Y} = k\big|\,\boldsymbol{X} = \boldsymbol{x}\big) = \frac{\exp_k(f_\theta(\boldsymbol{x}))}{\sum_{k'=1}^{K}\exp_{k'}(f_\theta(\boldsymbol{x}))}$,

where f_θ is the classification model parameterized by θ taking an image x as the input and producing the score (logit) vector $\boldsymbol{z} = f_\theta(\boldsymbol{x})$. Although increasing the width and depth of a network will enhance its capability to capture rich features, the network tends to be overconfident. Moreover, techniques widely used in training DNN, such as batch normalization and weight decay, make the model miscalibrated [15]. To show that DNNs trained with softmax could not detect OOD samples, we trained ResNet on the CNU weed dataset and tested on CIFAR dataset, the results are illustrated in Figure 2. Details will be given in section 4. Although only experimenting with ResNet, we note that our study can be expanded to other architectures.

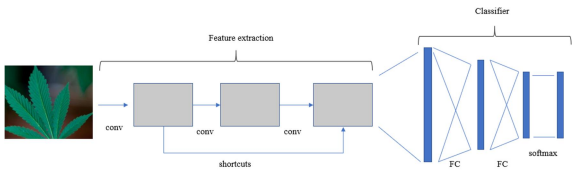

Figure 1. A brief overview of ResNet: architecture involves two parts, feature extraction and classifier, the feature extraction part consists of convolutional layers and shortcuts path to extract features from an image, the classifier is made from fully connected layers and softmax at the last layer.

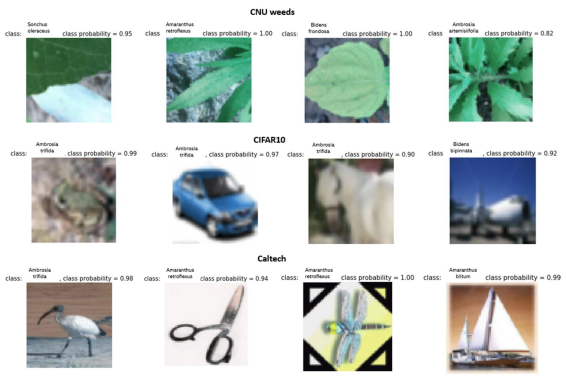

Figure 2. The vanilla softmax classifier (with ResNet18) trained on the CNU weeds dataset failed to detect OOD samples from CIFAR10 and Caltech dataset.

B. Variational Autoencoder

It has been proven that a generative model might be sufficient to detect OOD samples [16, 13, 14]. In the generative approach, the generative distribution $P_{in}(\boldsymbol{X})$ is estimated by the training dataset; hence the ODD samples could be eliminated by determining the low probability region. Noticeably, $P_{in}(\boldsymbol{X})$ is generally intractable because of the curse of dimension and lacking an infinite dataset. Instead of formulating $P_{in}(\boldsymbol{X})$ in the close form, one might want to approximate it by a parametric model. There are a number of researches that address the problem of estimating the distribution of a dataset, for example, k nearest neighbor, Pazen windows or Gaussian mixture model are well-known methods. To expand the representation capability of parametric models, density functions are approximated by more complicated forms such as Masked Autoencoder [16], PixelCNN [13]. Recently, generative adversarial networks (GANs) have also been noticed as an efficient way to model a sampling model as well as applied for OOD detection [14].

In this work, we employed Variational Autoencoder (VAE) [17] to approximate the data distribution $P_{in}(\boldsymbol{X})$. Our approach is partly similar to the work in [16], where autoencoder was used to generate perturbed samples. Compared to the estimators introduced in [13], VAE based estimator presents a latent variable $\boldsymbol{Z}$ to express perturbations of $\boldsymbol{X}$, instead of being dependent only on raw samples $\boldsymbol{X}$, which is considerably high dimension. The former method of VAE is variational Bayes inference, which presented variational lower bound, equation 6, to measure the capability of the encoder model $Q(\boldsymbol{Z}|\boldsymbol{X})$.

$$L(\boldsymbol{X}) = \mathbb{E}_{Q(\boldsymbol{Z}|\boldsymbol{X})}[-\log Q(\boldsymbol{Z}|\boldsymbol{X}) + \log P(\boldsymbol{X},\boldsymbol{Z})], \quad (6)$$

where $Q(\boldsymbol{Z}|\boldsymbol{X})$ is the encoder model, an approximation for the intractable posterior $P(\boldsymbol{Z}|\boldsymbol{X})$, $P(\boldsymbol{X},\boldsymbol{Z})$ is the joint distribution. We note that the class variable Y is not considered in this approach to avoid turbulence. Equation 6 can be expanded to the criterion of VAE, as given in equation 7.

$$L(\boldsymbol{X}) = \mathbb{E}_{Q(\boldsymbol{Z}|\boldsymbol{X})}[log\, P(\boldsymbol{X}|\boldsymbol{Z})] - KL(Q(\boldsymbol{Z}|\boldsymbol{X})||P(\boldsymbol{Z})), \quad (7)$$

where $P(\boldsymbol{X}|\boldsymbol{Z})$ is the decoder used to reconstruct $\mathbf{X}$ from the latent $\mathbf{Z}$, $P(\boldsymbol{Z})$ is prior distribution of $\boldsymbol{Z}$. In variational Bayes inference, the variational lower bound $L(\boldsymbol{X})$ arises naturally from log marginal likelihood.

$$\log P(\boldsymbol{X}) = KL\big(Q(\boldsymbol{Z}|\boldsymbol{X})||P(\boldsymbol{Z}|\boldsymbol{X})\big) + L(\boldsymbol{X}), \quad (8)$$

Trivially, maximizing the log marginal likelihood $\log(P(\boldsymbol{X}))$ and minimizing the KL divergence between the true posterior $P(\boldsymbol{Z}|\boldsymbol{X})$ and its approximation $Q(\boldsymbol{Z}|\boldsymbol{X})$ is equivalent to maximize the variational lower bound $L(\boldsymbol{X})$. In DNN perspective, Furthermore, VAE is the consequence of parameterizing $Q(\boldsymbol{Z}|\boldsymbol{X})$ and $P(\boldsymbol{X}|\boldsymbol{Z})$ to encoder network and decoder network (Figure 3) jointly trained with variational reconstruction loss $-L(\boldsymbol{X})$.

In terms of implementation, we used the reparameterization trick [17] to sample from the prior

$P(\boldsymbol{Z})$, the variational reconstruction loss $-L(\boldsymbol{X})$ is given in equation 9.

$$-L(\boldsymbol{x}) = -\frac{1}{2}\sum_{j=1}^{K}\left(1 + log(\sigma_j^2) - (\mu_j)^2 - (\sigma_j)^2\right) + \frac{\beta}{N}\sum_{i=1}^{N}\|\boldsymbol{x} - \boldsymbol{y_i}\|^2, \quad (9)$$

where μ_j, σ_j are output from the encoder network, which are parameters of the Gaussian distribution assumed for $Q(\boldsymbol{Z}|\boldsymbol{X})$. $\boldsymbol{y_i}$ is a reconstruction of $\boldsymbol{x}$ output from the decoder network, where $P(\boldsymbol{X}|\boldsymbol{Z})$ is also assumed to distributed as isotropic Gaussian. Finally, the OOD samples are detected by calculating the approximate (Monter Carlo and Jensen inequality) log marginal likelihood of $\boldsymbol{X}$.

$$\log(P(\boldsymbol{x})) = \log\left(\sum P(\boldsymbol{x}|\boldsymbol{z})P(\boldsymbol{z})\right) \approx \log\left(\sum P(\boldsymbol{x}|\boldsymbol{z})Q(\boldsymbol{z}|\boldsymbol{x})\right) \propto -\sum_{i=1}^{N}\|\boldsymbol{x} - \boldsymbol{y_i}\|^2, \quad (10)$$

where $\boldsymbol{y_i} = Decoder(\boldsymbol{x}, \boldsymbol{z_i})$ is the i^{th} reconstruction of $\boldsymbol{x}$.

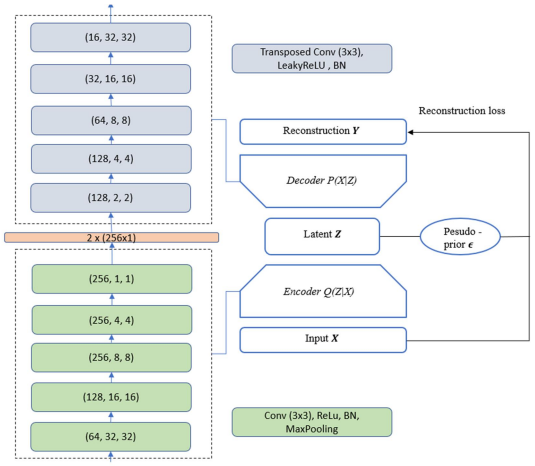

Figure 3. Variational Auto Encoder: The input is an image with a size of $64 \times 64 \times 3$, the encoder includes 5 ResNet blocks and 2 additional FC layers to encode mean and covariance of $\boldsymbol{Z}$. The decoder part is formed by a series of 5 transposed convolutions to reconstruct the original image.

IV. EXPERIMENT

A. Dataset

We undertake the empirical analysis using three datasets, CNU weed dataset served as the in distribution dataset, while CIFAR10 and Caltech were considered as the OOD datasets.

- CNU weed dataset: the CNU weed dataset [18] comprises of images from 21 categories of weeds growing up in Korea. Those samples were collected by high-resolution cameras in various scenarios. The dataset used in this study had already been manually preprocessed by experts, where the weeds (or part of weeds: leaf, flower, branch, bud) were cropped from the noisy background and centered in the cropped images. After cropping, the number of samples per category is highly imbalance, when the largest amount is 11%, and the smallest amount is 3% out of 210k images. We used the CNU weeds dataset in this study because it is a real word large-scale dataset, not only diverse in pieces but also diverse in family. The dataset has been labeled carefully, and the amount of data has been statistically organized. Besides, this dataset has also been successfully used for plenty of tasks [18, 19].
- CIFAR10 dataset[1]: The CIFAR10 dataset consists of 60K color images with a size of 32×32 belonging to 10 classes. There are two main categories of the label, vehicle (airplane, automobile, ship, truck), and animal (bird, cat, deer, dog, frog, horse). The dataset is suitable to serve as the OOD dataset since there not exists any class related to plant category. In this study, we used 10K images in the testing dataset to evaluate our framework.
- Caltech dataset[2]: The Caltech dataset contains around 9.2k color images in total; the original size of each image is 300×200. It is a diverse dataset when there are 101 categories. The number of images in each category is not equal, which varied from 40 to 800 images per class, but most categories have around 50 images. Although there are some classes related to the category of plants, they are completely different from samples in the CNU weeds dataset. In this work, all images were used to evaluate the OOD detection models.

B. Evaluation metrics

[1]. Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, 2009.

[2]. L. Fei-Fei, R. Fergus and P. Perona. One-Shot learning of object categories. IEEE Trans. Pattern Recognition and Machine Intelligence. In press.

Threshold-based detection: the purpose of constructing the above detection methods is that finally, an OOD sample can be rejected by a threshold, where the threshold value varied by criterion required in a particular method. Details are given in equation 11 and Table 1.

Where the OOD sample will be recognized If its value $q(\boldsymbol{x})$ is smaller than a threshold δ_q, note that δ_q is dependent on the detection method q. Tables 1 summarizes the criterion q of each method represented in this study.

$$OOD(\boldsymbol{x}) = \begin{cases} 1 & if\ q(\boldsymbol{x}) \le \delta_q, \\ 0 & otherwise \end{cases} \quad (11)$$

Table 1. Summary of detection criterions

Method	Criterion	Formula
Softmax class probability	Softmax class probability	$q(\boldsymbol{x}) = \max_c P(\hat{Y} = c \mid \boldsymbol{X} = \boldsymbol{x})$
Temperature scaling	Softmax class probability with temperature scaling T	$q(\boldsymbol{x}) = \max_c P_T(\hat{Y} = c \mid \boldsymbol{X} = \boldsymbol{x})$
Label smoothing	Softmax class probability trained with "soft" label α	$q(\boldsymbol{x}) = \max_c P_\alpha(\hat{Y} = c \mid \boldsymbol{X} = \boldsymbol{x})$
Gaussian discrimination analysis	Mahalanobis distance	$q(\boldsymbol{x}) = \max_c -(g(\boldsymbol{x}) - \hat{\mu}_c)^T \hat{\Sigma}^{-1} (g(\boldsymbol{x}) - \hat{\mu}_c)$
Variation autoencoder	Negative log marginal likelihood	$q(\boldsymbol{x}) = \sum_{i=1}^{N} \|\boldsymbol{x} - \boldsymbol{y}_i\|^2$

C. Experimental results

First of all, we trained a classification model on the CNU weeds dataset, the model used here was ResNes18 with the default configuration [8], and being pre-trained on ImageNet dataset. We trained the network for 200 epochs, the batch size was set to 64, the initial learning rate was 0.01 and decreased haft after 40 epochs to ensure the convergence of network, and the Adam optimizer was used to minimize the cross-entropy loss. In this study, we set the resolution of an image to 64×64 despite the fact that the image sizes varied by different datasets. We note that the OOD detection methods presented in this study do not require OOD samples for training. The training process was completely blind to the OOD dataset. Which is the practical scenario when the number of OOD samples is infinite, and ones normally do not own the sampling model. After training 200 epochs, the best parameters on the validation set will be used to evaluate the test set, where the portion of training, validation, and test dataset was 60% − 20% − 20%, respectively. The classification accuracy on the test set was 99.22%. We showed that even though the model got high accuracy in the classification task, vanilla soft max could not detect OOD samples.

Regarding the label smoothing approach, the same experimental scenario was used except for that "hard" labels were replaced by "soft" labels with $\alpha = 0.9$. The classification accuracy on the test set was 99.44% when training with label smoothing techniques. Not only achieving higher classification accuracy, but the model trained with label smoothing could also avoid overconfidence. Figure 5 shows reliability diagrams [15] and histogram of softmax probability scores, which illustrates that model trained with "soft" label alleviated miscalibrated.

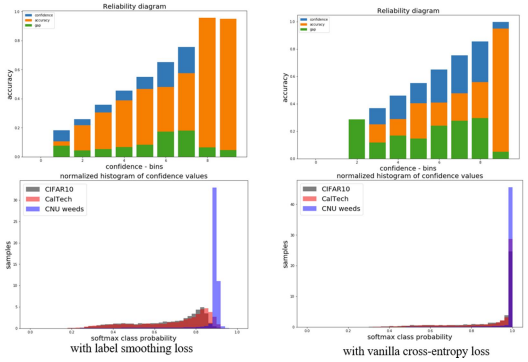

Figure 5. Reliability diagrams and histogram of softmax probabilities on CNU weeds, CIFAR10, Caltech dataset of the model trained with vanilla cross-entropy loss and model trained with label smoothing - cross-entropy loss. The gaps between confidence and probability on each column of the model trained without label smoothing are larger than that of the model trained without label smoothing, which tells that the model in the right is overconfident.

In the temperature scaling approach, we used the pre-trained model with vanilla cross-entropy as stated above and extract the logits layer (right before the softmax layer) to calculate new probabilities with

temperature scaling value T, as given in Equation 2. Figure 6 shows the histograms of softmax class probability when varying some values of T. when $T < 1$, the histograms are spread to value 1, make the model predict all of the samples in 3 datasets approximately 100%. While $T > 1$, the OOD datasets tend to reach zero confidence faster than in distribution, which suggested that a threshold could be used to distinguish OOD samples from in distribution samples.

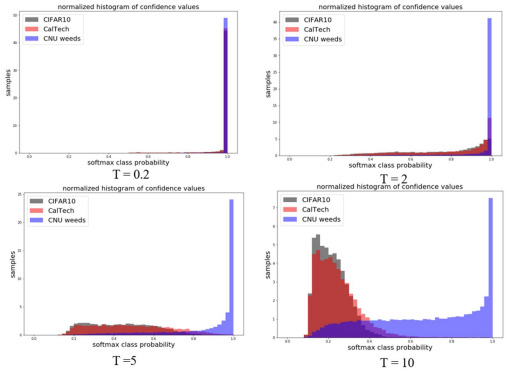

Figure 6. Histogram of softmax probabilities with varied values of temperature scaling factor.

In Gaussian discrimination analysis, we extracted features at the last layer of the feature extraction part and trained a linear classifier with Gaussian assumption for the class likelihood probability. The extracted feature vectors had a size of 512, which was the layer from trained ResNet before being fed through a series of FC layers to do classification. The Mahalanobis distance was then used as a confidence score to decide a threshold discriminating OOD and in-distribution dataset; the formula is given in equation 5. We experimented with feature vectors driven from the model trained with and without label smoothing. In the case of label smoothing, Figure 7 shows that the CNU weeds dataset has relatively low Mahalabonis distance, while samples in the OOD datasets are far away from the centroids. Otherwise, the classifier trained with feature vectors output from ResNet without label smoothing was incapable of rejecting OOD samples by Mahalobonis distance.

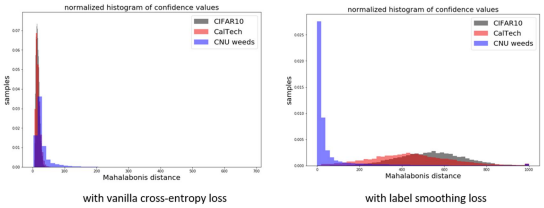

Figure 7. Histogram of Hahalonobis distances measured over CNU weeds, CIFAR10 and Caltech dataset in cases of with and without label smoothing.

Those above methods utilized the pre-trained model on an in-distribution dataset and manipulated the class probability to come up with a rejection criterion. Otherwise, VAE is a network that estimated the marginal likelihood of the in distribution dataset regardless of labels. In particular, we trained a VAE that consisted of an encoder to decompose an input image into latent vector and a decoder to reconstruct the input image. The encoder was the training from scratch - ResNet18 with the addition of 2 FC layers for encoding mean and covariance of distribution of corresponding latent vector. We set the dimension of latent vectors to 256. The decoder network involved 5 consecutive decoder blocks; each block had 1 transposed convolutional layer followed by LeakyReLu activation function and batch normalization. The last layer of the decoder network is the sigmoid function used to reconstruct the input image. We trained the network for 100 epochs with Adam optimizer, batch size of 128, and initial learning of 0.001. To distinguish OOD samples from in-distribution data, we calculated negative log marginal likelihood by equation 10. The histograms of negative log marginal likelihood are illustrated in Figure 8, where the negative log marginal likelihood estimated from CNU weeds dataset was closed to zero, while that of CIFAR10 and CalTech varies in a large range far from zero.

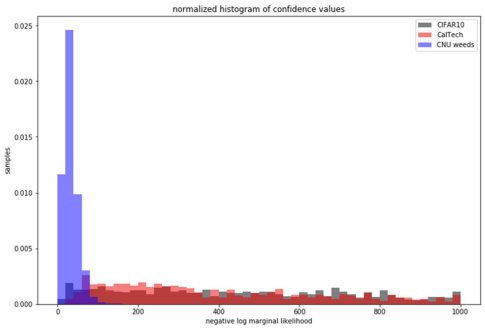

Figure 8. Histograms of negative log marginal likelihood output estimated by VAE.

We used AUPR, a threshold-independent metric, to evaluate a method used in this study. A receiver operating characteristic curve (ROC) is a graphical plot that illustrates trades of between type I and type II errors, which are normally encountered in binary classification (e.g., anomaly detection). Additionally, the AUPR score is a fair metric to make comparisons among methods where the impact of rejection

thresholds is discarded. Figure 9 shows that vanilla softmax has the smallest AUPR value (0.89), while the highest case is approximately 1 achieved by the marginal likelihood of VAE approach. Besides, methods such as Mahalanobis distance, temperature scaling and label smoothing also attained higher AUPR scores than of vanilla softmax, which states their efficiency in removing OOD samples by using only an appropriate threshold.

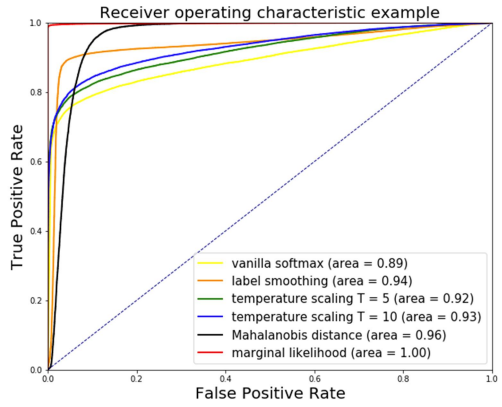

Figure 9. ROC curves and the corresponding AUPR values.

We conducted all experiments on Ubuntu Server equipped by a GPU of 12 GB RAM, and the main used framework was Pytorch. Public code is available at , which aims to the clear of reproducing results in this study. In the label smoothing and temperature scaling method, the inference time of classification with OOD detection is similar to the case without OOD detection because there is no extra operation included at the testing phase. In the GDA approach, an OOD sample is detected by Mahalabonis distance, which costs 0.0045 ms to calculate. Likewise, VAE requires a little extra time (0.005 ms) to decide if a sample belongs to in-distribution. Those evaluations guarantee that a weeds classification system equipped with an OOD detection mechanism still enable real-time processing

V. CONCLUSION

This study aims to solve the problem of Out of distribution detection on the classification task by examining four techniques: temperature scaling, label smoothing, Gaussian discrimination analysis and variational autoencoder. Experimenting with the weeds classification using a deep neural network, we have shown that the vanilla softmax class probability was not confident enough to remove samples that are unlikely drawn from the training set. Through various experiments, we concluded that OOD samples could be rejected based on marginal likelihood (VAE approach) or calibrated class probability (label smoothing temperature scaling and GDA approach). In conclusion, the OOD detection mechanism makes a classification model reliable with minor degradation in classification accuracy as well as inference time.

REFERENCES

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *2005 IEEE computer society conference on computer vision and pattern recognition*, 2005

[2] H. Bay, T. Tuytelaars and L. V. Gool, "Surf: Speeded up robust features," *European conference on computer vision*, Berlin, 2006

[3] T. Ahonen, A. Hadid and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *EEE transactions on pattern analysis and machine intelligence,* vol. 28, no. 12, pp. 2037-2041, 2006

[4] K. Alex, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems,* pp. 1097-1105, 2012

[5] O. Russakovsky, J. Deng, S. Hao , J. Krause, . S. Satheesh, S. Ma, Z. Huang, A. Karpathy and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International journal of computer vision,* vol. 115, no. 3, pp. 211-252, 2015

[6] N. Hoang, G. Lee, S. Kim, H. Yang, "Effective Hand Gesture Recognition by Key Frame Selection and 3D Neural Network," *Smart Media Journal,* vol. 9, no. 1, pp. 23-29, 2020

[7] C. Szegedy, . V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the inception architecture for computer vision," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016

[8] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016

[9] A. Nguyen, J. Yosinski and J. Clune, "Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images," *Computer Vision and Pattern Recognition*, 2015

[10] A. Nguyen, J. Yosinski and J. Clune, "Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images," *Computer Vision and Pattern Recognition*, 2015

[11] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *International Conference on Learning Representations*, 2017

[12] K. Lee, K. Lee, H. Lee and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," *Advances in Neural Information Processing Systems,* pp. 7167-7177, 2018

[13] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. A. DePristo, J. V. Dillon and B. Lakshminarayanan, "Likelihood ratios for out-of-distribution detection," *Advances in Neural Information Processing Systems,* pp. 14680-14691, 2019

[14] K. Lee, H. Lee, K. Lee and J. Shin, "Training confidence-calibrated classifiers for detecting out-of-distribution samples," *International Conference on Learning Representations*, 2018

[15] C. Guo, G. Pleiss, Y. Sun and K. Q. Weinberger, "On calibration of modern neural networks," *Proceedings of the 34th International Conference on Machine Learning*, 2017

[16] M. Germain, K. Gregor, I. Murray and H. L. Larochelle, "MADE: Masked Autoencoder for Distribution Estimation," *International Conference on Machine Learning*, 2015

[17] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114* , 2013

[18] T. H. Vo, H. G. Yu, V. T. Dang and Y. J. Kim, "Late fusion of multimodal deep neural networks for weeds classification," *Computers and Electronics in Agriculture,* vol. 175, pp. 105506, 2020

[19] T. H. Vo, G. H. Yu, H. T. Nguyen, J. H. Lee, T. V. Dang, J. Y. Kim,"Analyze weeds classification with visual explanation based on Convolutional Neural Networks," *Smart Media Journal,* vol. 8, no. 3, pp. 31-40, 2019

Authors

Thanh-Vu Dang

He is a student of M.S. degree in Department of Electronics Engineering at Chonnam National University. He received his B.S. degree in Mathematics and Computer Science at Vietnam National University-University of Science, Vietnam in 2018. His research interests are Digital Signal Processing, Image Processing, Speech Signal Processing, Machine Learning.

Hoang-Trong Vo

He is a student of Ph.D. degree in Department of Electronics Engineering from Chonnam National University. He received his M.S. degree in Electronics Engineering from Chonnam National University, Korea in 2019. His research interests are Object Classification, Neural Network, Deep Learning.

Gwang-Hyun Yu

He is a student of Ph.D. degree in Department of Electronics Engineering at Chonnam National University. He received his M.S. degree in Electronics Engineering from Chonnam National University, Korea in 2018. His research interests are Digital Signal Processing, Image Processing, Speech Signal Processing, ML, DL.

Ju-Hwan Lee

He is a student of M.S. degree in Department of Electronics Engineering at Chonnam National University. He received his B.S. degree in Oceanography from Chonnam National University, Korea in 2019. His research interests are Digital Signal Processing, Image Processing, Machine Learning.

Huy-Toan Nguyen

He is a Postdoctoral Researcher in Department of Electronics Engineering at Chonnam National University. He received his B.S. degree in Engineering from Thai Nguyen University of Technology, Vietnam in 2012 and Ph.D. degree in Electronics and Computer Engineering from Chonnam national University in 2020. His research interests are Computer Vision, Wearable Device, Microprocessor Based Systems, ML, DL.

Jin-Young Kim

He is a professor in Department of Electronics Engineering at Chonnam National University, Korea. He received his B.S., M.S. and Ph.D. degree in Electronics Engineering from Seoul National University, Korea in 1986, 1988 and 1994, respectively. His research interests are Digital Signal Processing, Image Processing, Speech Signal Processing, ML, DL.