OUT-OF-DISTRIBUTION CLASSIFICATION WITH ADAPTIVE LEARNING OF LOW-LEVEL CONTEXTUAL FEATURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Humans can inherently identify what is unknown to them, but the existing Neural Network (NN) is still lacking in this aspect. Out-of-Distribution (OOD) classification is an incredibly challenging problem for any NN model. In General, any model tries to predict the OOD samples from the labels used for training only, but that is not acceptable for AGI (Artificial General Intelligence) [Fjelland (2020)]. There are several kinds of research already done to avoid this issue and build a model to predict the OOD samples, existing baseline work like 1) Thresholding SoftMax, 2) train model by adding extra OOD class as a label, or 3) Mahalanobis distance-based approach. All existing approach uses the CNN to get the spatial feature information and channel-wise information within the local receptive field at each layer. Here in this paper, we have proposed a method to learn the features of In-class and OOD sample's features with global receptive field among channels to learn the spatial relationship with modified SEnet block. Broadly, our model learns the interdependencies between channels with adaptive recalibration of the weights of stacked channels at each layer. To give more weightage to the In-class samples, we uniformly normalized the OOD samples with the total number of known class samples and trained our model to suppress the OOD class probability with a simple and effective loss function. We did our experiments for our model with MNIST and F-MNIST as In-class samples and EMNIST, KMNIST, not-MNIST, Omniglot, Uniform Noise, Gaussian Noise as OOD samples.

1 INTRODUCTION

With the rising power and accessibility of graphics processing units(GPUs), Neural networks(NNs) have been applied in different tasks and purposes over the past decade. During development, any proposed NN model performs well when we train and test those models with the independent and identical data distribution. However, model may not give the desired output label(i.e. Out-of-Distribution dataset label) for the test data if the data distribution is different from the limited distribution of the training data.

Labeling the unknown classes(i.e., Out-of-Distribution) into known ones with high probability confidence, may cause the problem for the deployment of the model for industrial purposes like Autonomous cars, medical industries, etc. By these facts, detecting and labeling the OOD dataset is an important building block for explainable AI [amodei et al. (2016)].

Several other works have already been done to address this problem by defining the maximum/threshold SoftMax score [Hendrycks & Gimpel (2016)] or by temperature scaling the SoftMax score with added input perturbation [Liang et al. (2017)], [Guo et al. (2017)]. But the reliability and accuracy of detecting OOD class samples were saturated, due to the lack of additional features apart from spatial features and channel-wise information within the local receptive field of CNN (Convolution Neural Network) filters. Here in this paper, we go a little further. We observe that the spatial features from the CNN filter can't learn the global spatial relationship between channels, so we added Modified SEnet Block in the network and trained that model with In-Class and OOD samples. With the distribution plot in Figure 6, we found that the features for both In-class and OOD datasets are more robust to segregate the OOD samples at model test time and we show that

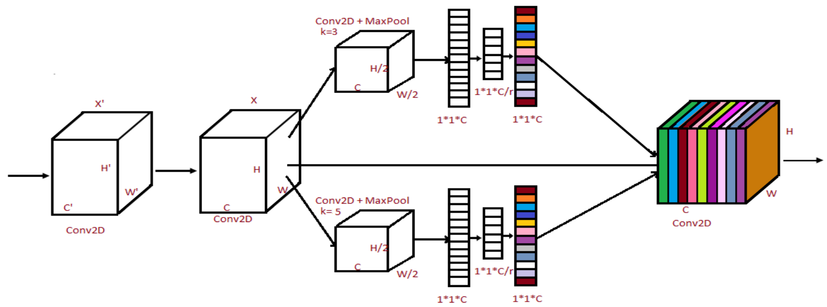


Figure 1: *Modified SEnet Block: The additional blocks in SEnet block i.e. Conv2D + MaxPooling with kernel size 3,5 that helps our model to learn better low level structural features along with high level features in each layer. Then squeeze and excitation layer, extracted the stacked channel wise information with global receptive field.*

this additional block can lead to better OOD classification performance along with In-class detection performance. For example, pre-trained Alexnet [Krizhevsky et al. (2012)] model with Modified SEnet block on MNIST dataset (In-class samples), we test against images from EMNIST, KMNIST, FMNIST, not-MNIST, Omniglot, Uniform & Gaussian Noise dataset (OOD samples). Our approach reduces the Detection error of OOD samples from 13.6% to 0.69%, in comparison with the baseline method [Lee et al. (2018b)] (when 95% of in-distribution images are correctly classified) for the EMNIST OOD sample. We summarize the main contributions of this paper as the following:

- We propose a simple and effective method to detect out-of-distribution samples in neural networks. Our Method requires both In-class and OOD-class samples to train our model and learn the efficient features along with the spatial and stacked channel-wise, global feature information at each layer. Model is learning the features with an additional layer in the block of [Hu et al. (2018)], which helps us to understand better and reliable global feature parameters for classification purposes.
- Our main insight is that learning the global structural information along with spatial, color, and texture information for In and OOD class samples, helps our model to learn better features for In-class samples.
- We propose an algorithm with a modified SEnet block, that substantially improves the OOD sample detection with maintaining the In-class classification accuracy.
- We analyze our method with different dataset like MNIST, Fashion-MNIST [Xiao et al. (2017)], EMNIST [Cohen et al. (2017)], KMNIST [Prabhu (2019)], not-MNIST, Omniglot [Lake et al. (2015)], Uniform Noise[Vyas et al. (2018)], Gaussian Noise[Vyas et al. (2018)] and provide effective results with our approach.

Figure 1 is a single block of our modified SEnet, where we added the extra layer Con2D + MaxPooling, to get the much stable low-level global contextual as well as spatial features. Figure 2 & Figure 3 represent the data distribution from our model for In-class and OOD classes after adding some addition noises. Figure 5 shows the Alex-Net model 2D representation of features extracted from In-class and OOD class samples, while Figure 6 shows the 2D representation of the extracted features for In-class(Colored) and OOD class(Black) from the modified SEnet block algorithm model.

The outline of this paper is as follows. In section 2, we briefly go over the background and related work. We present our Modified SEnet approach and institution behind the loss function insection 3. Then, we introduce visualization for OOD detection in section 4. section 5 is for discussion about the results, and final conclusion and future works of our method are in section 6.

2 BACKGROUND AND RELATED WORK

OOD Classification is about classifying those samples whose distribution doesn't match the distribution of the dataset on which the model is trained for classification. The samples with predefined labels used for model trains are representative of the in-distribution class. There are several methods to identify OOD class samples as follows:

- Probabilistic approaches: Several papers have reported about OOD detection with a probabilistic approach.They make the maximum softmax probability score discriminate between In-class and OOD samples Here in this method model trained with In-class samples, will give a low probability score for OOD samples. recent work like Corbière et al. (2019) also proposed the detection of an overconfident incorrect prediction, temperature scaling was studied by Guo et al. (2017) for OOD class detection. Hendrycks & Gimpel (2016) proposed a baseline is for OOD class probability will be lower than the In-class probability distribution. Lee et al. (2018a) used the GAN method to reduce the probability confidence of OOD classes. Hendrycks et al. (2019) use the OE (Outlier exposure) method to extract the diverse dataset from the real one. Malinin & Gales (2018), Bevandic et al. (2018) uses the model which can make an uncertain prediction for OOD classes.
- Distance-based approaches: Lee et al. (2018b), proposed the distance-based approach to calculate the confidence score for the samples with existing sample cluster if it is significantly far than sample belongs to OOD class.
- Reconstruction-oriented approaches: Nalisnick et al. (2019), Ren et al. (2019), Alemi et al. (2018), Burda et al. (2016) is the method that uses the data compression technique Here in this approach model compress the In-class data sample to get the latent information and Compare it with OOD class samples to get the effective knowledge about which class it will belong.
- Domain-based approaches: Lamrini et al. (2018) try to fit the domain for the In-class data samples and the data samples which will not fit in those domains will be OOD class.
- Information-theoretic techniques: Here the model train with In-class samples and put any sample, if it will be OOD samples the entropy increases drastically meanwhile for In class it will not vary much.

3 ADAPTIVE FEATURE LEARNING FOR OOD CLASSIFICATION

In this section, we perform will perform experiments to answer the following questions:

- Which factor of the model is affecting the OOD classification most?
- How to better extract the In-class sample features along with the OOD sample features?
- How to suppress the OOD sample magnitude without dropping the in-class classification accuracy?

We perform extensive experiments to answer these questions. In summary, we have the following findings:

- Adding low-level global features at each layer affect the OOD classification.
- OOD sample normalization helps our model to learn their sample distribution.
- The loss function will suppress the OOD class feature magnitude, help our model to distinguish between In-class and OOD class samples.

3.1 MODIFIED SENET APPROACH

Our approach of learning low level contextual features along with high level spatial features at each layer with modified SEnet block Figure 1, helps to get better In-class as well as OOD-class sample feature information. Additional layers of convolutions with kernel size 3,5 along with max-pooling layer with SEnet block helps our model to understand better In-class samples features. The weighted fully connected layer will learn the stacked channel-wise global feature information with adaptive learning, helping our model to get the local spatial as well as global contextual information of the In-class as well as OOD class samples.

3.2 OOD SAMPLE NORMALIZATION

Normalization of OOD samples during model training makes our model efficient to distinct the In-class and OOD class sample features. Before normalization, we assume that the distribution of every class feature will be uniform De Stefano et al. (2000), Fumera & Roli (2002) in hidden layers for model training. The normalization of OOD samples has been done by dividing the OOD-classes samples by the total number of In-class samples. The cost function after normalization for OOD samples is in Equation 1 & Equation 2.

3.3 INTUITION BEHIND THE LOSS FUNCTIONS

While training our model, there are two loss function used for feature extraction: 1) Weightage cross-entropy loss, to learn the classification between various In-classes and OOD-class sample. While training we had normalized our entropy loss function for the OOD class samples by the total number of known classes, which help our model to give less weightage Figure 6c for OOD samples. Our Entropy loss $\mathcal{L}_E$ is defined as:

$$\mathcal{L}_E = \begin{cases} \textit{balanced-xcent}(\hat{\mathbf{Y}}, \mathbf{Y}^*), & \text{if } \hat{\mathbf{Y}} \in D_{in} \\ \frac{1}{N} Entropy(\hat{\mathbf{Y}}, \mathbf{Y}^*), & \text{if } \hat{\mathbf{Y}} \in D_{out} \end{cases} \tag{1}$$

$$\mathcal{L}_E = \begin{cases} -\beta \mathbf{Y}^* \log \hat{\mathbf{Y}} - (1-\beta)(1-\mathbf{Y}^*) \log(1-\hat{\mathbf{Y}}), & \text{if } \hat{\mathbf{Y}} \in D_{in} \\ \frac{1}{N}(-\mathbf{Y}^* \log \hat{\mathbf{Y}} - (1-\mathbf{Y}^*) \log(1-\hat{\mathbf{Y}})), & \text{if } \hat{\mathbf{Y}} \in D_{out} \end{cases} \tag{2}$$

Where $\hat{\mathbf{Y}}$ is the prediction score, $\mathbf{Y}^*$ is the ground truth, the parameter β is the weightage factor for In-class samples and N is the total number of known classes for the given In-class dataset (D_{in}) and OOD class (D_{out}).

2) Objectosphere loss function, to learn and maximize the distance between the In-class and OOD-class sample distributions.

$$\mathcal{L}_{Obj} = \begin{cases} max(\gamma - \|\text{F(x)}\|, 0)^2 & \text{if } \hat{\mathbf{Y}} \in D_{in} \\ \|\text{F(x)}\|^2 & \text{if } \hat{\mathbf{Y}} \in D_{out} \end{cases} \tag{3}$$

where, γ is the lowest probability score for the known data samples, $\|F(x)\|$ is the norm of softmax output help to minimize the feature-length and entropy for the OOD samples.

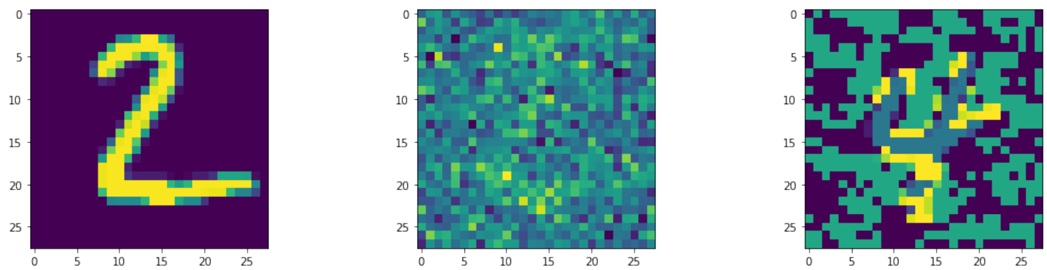
The overall loss to train our model is the combination of entropy loss and objectosphere loss [Dhamija et al. (2018)]. Here entropy will learns the features for classification and objectosphere loss function scale the In-class samples towards higher probability and OOD samples to low probability score, help the samples to maximize the margin between In-class and OOD samples. Total loss function $\mathcal{L}_T$ is defined as:

$$\mathcal{L}_T = \mathcal{L}_E + \lambda \mathcal{L}_{Obj} \tag{4}$$

where λ is the scaling factor defined as the probability score

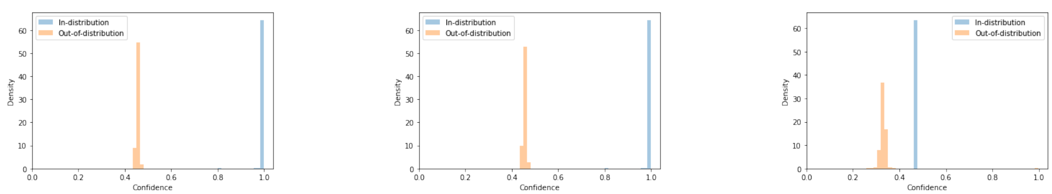
3.4 ADVERSARIAL EFFECT ON OUR MODEL

To test our model robustness against adversarial effect we added some white noise, Gaussian noise as in Figure 2b and FGSM with an epsilon 0.1 (Fast Gradient Sign Method) [Goodfellow et al. (2015)] in Figure 2c respectively and we found that with white and Gaussian noise there is no drastic change is classification accuracy Figure 3b at all, because our model has learned the efficient low-level contextual features along with high-level features which are preserved even after adding noise. Meanwhile, by added the adversarial attack FGSM with an epsilon value of more than 0.1 we are losing in-class accuracy Figure 3c but even, in this case, our model can differentiate between in-class and OOD samples. For further value of epsilon, model will be not robust enough due to loss of global structural feature information.



(a) (b) (c)

Figure 2: *Input MNIST dataset for our model with a) Input MNIST digit with grayscale form b) Input MNIST digit with added Gaussian noise of zero mean and unit variance c) Input MNIST FGSM adversarial digit image for 0.1 epsilon.*



(a) (b) (c)

Figure 3: *In-class and OOD class samples density vs confidence with probability score for a) Input MNIST digit with grayscale form b) Input MNIST digit with added Gaussian noise of zero mean and unit variance c) Input MNIST FGSM adversarial digit image for 0.1 epsilon.*

3.5 Limitation of our approach

We had tested our approach with a different dataset as mentioned in subsection 5.1, but there are few limitations of our approach is that our approach is effective with a dataset with a fixed background. Dataset with different backgrounds causes a problem to learn the better and efficient features due to varying color features. Our approach is effective for adversarial images at a certain limit and we had tested it by adding Gaussian noise and FGSM with various epsilon values as shown in Figure 2 and the corresponding probability score in Figure 3 for In-class and OOD samples.

3.6 Setup

3.6.1 In-distribution Datasets.

To proof our method, we train and evaluate our model on various datasets. In our experiment, we used different OOD datasets for training and test purpose. For example, with MNIST or Fashion-MNIST dataset as D_{in}, we have used the EMNIST and MNIST as D_{out} respectively. The images are normalized to be in the range[0,1] before training. we evaluate our model with unseen D^{out}_{test} like K-MNIST, not-MNIST, and F-MNIST, Omniglot, Uniform Noise, and Gaussian Noise dataset to validate our model performance.

3.6.2 Architectures and Training Configurations.

We use the modified version of the SENet model as shown in Figure 4. Losses for the training are weightage cross-entropy loss plus the objectosphere loss as in subsection 3.3, to suppress the OOD samples probability score and maximize the margin between In-class and OOD classes. We trained our model with 150 epochs and 128 batch sizes.

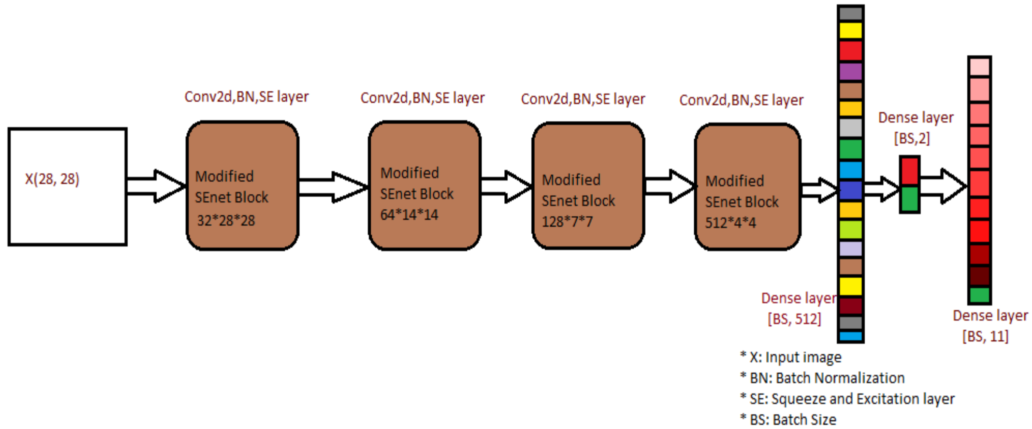


Figure 4: *Modified SEnet Model: Trained for MNIST and Fashion-MNIST dataset with 10 classes as In-distribution and one for OOD sample distribution with overall Alexnet.*

3.6.3 Hyper-parameters.

To get the desired results, we randomly select 60,000 samples from D_{in} and 6000 samples from D_{out} to train our proposed model and set the batch size of 128. for efficient and stable results we use the SGD optimizer with learning rate 0.01, zero momentum and the optimal parameters are chosen to minimize the FPR at TPR 95%. While training the model we had taken then all possible layers which can generalized our model like Dropout Srivastava et al. (2014), Batch Normalization Ioffe & Szegedy (2015) and By using simple SGD optimizer Zhou et al. (2020). Generalization helps the model, not to over-fit the training dataset and learn the most generalized features of given samples.

3.7 Evaluation Metrics

Our OOD classification method is similar to the Maximum softmax probability detection method Hendrycks & Gimpel (2016) so we here also adapt the evaluation metrics used in Hendrycks et al. (2019). Defining as binary classification with OOD samples as +ve class and In-distribution sample as a -ve class.

- **FPR at 95% TPR.** Liang et al. (2017), Balntas et al. (2016), Kumar BG et al. (2016) This performance metric calculates the false positive rate (FPR) on out-of-distribution examples when the true positive rate (TPR) is 95%.
- **Detection Error.** This metric corresponds to the minimum misdetection probability over all possible thresholds γ, which is $min_{\gamma} L(\,PX, QX; G(\,x;\gamma)\,)$.
- **AUROC.** Area Under the Receiver Operating Characteristic curve is a threshold-independent metric Davis & Goadrich (2006). It can be interpreted as the probability that a positive example is assigned a higher detection score than a negative example Fawcett (2006). A perfect detector corresponds to an AUROC score of 100%.
- **AUPR.** Area under the Precision-Recall curve, which is another threshold independent metric Manning & Schutze (1999). The PR curve is a graph showing the precision=TP/(TP+FP) and recall=TP/(TP+FN) against each other. The metric AUPR-In and AUPR-Out in Table 1 denotes the area under the precision-recall curve where in-distribution and out-of-distribution images are specified as positives, respectively.

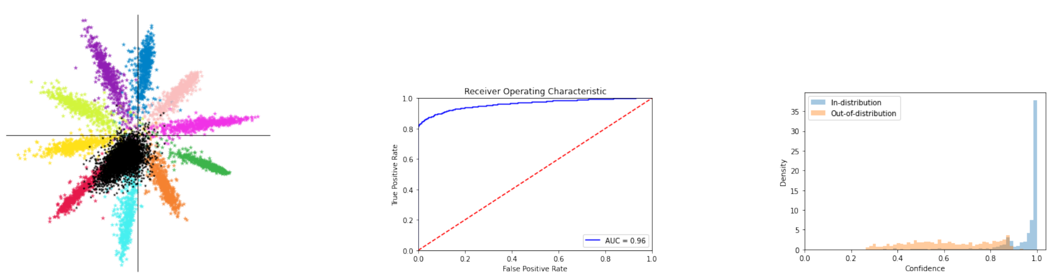
	Out-of-distribution dataset	FPR at 95% TPR ↓	Detection-Error ↓	AUROC ↑	AUPR-In ↑	AUPR-Out ↑
		MD / ODIN / Confident-Classifier /Ours				
	EMNIST	31.2/25.7/31.0/**0.0**	13.6/94.4/93.0/**0.69**	93.2/94.4/93.0/**99.88**	92.7/94.3/93.0/**99.93**	93.2/94.1/92.4/**99.42**
Modified	K-MNIST	NA/0.03/NA/**0.0**	NA/NA/NA/**1.15**	NA/97.60/NA/**99.83**	NA/97.05/NA/**99.88**	NA/NA/NA/**99.58**
SEnet	F-MNIST	94.2/0.4/7.9/**0.0**	11.9/1.8/5.6/**0.4**	86.6/99.8/98.5/**99.89**	92.0/99.8/98.8//**99.94**	74.0/**99.8**/98.4/99.39
(MNIST)	not-MNIST	34.8/11.3/26.5/**0.0**	16.3/6.9/12.3/**0.1**	91.7/97.8/94.0/**99.81**	91.3/97.7/93.8/**99.94**	92.3/97.7/93.8/**98.33**
	Omniglot	98.5/0.0/0.0/**0.0**	46.9/0.2/1.0/**0.1**	19.8/100/**100**/99.80	40.8/100/**100**/99.90	35.0/100/**100**/98.82
	Uniform Noise	82.6/0.0/0.0/**0.0**	26.4/0.0/**0.0**/0.1	65.0/100/**100**/99.80	76.0/100/**100**/99.90	63.9/100/**100**/98.80
	Gaussian Noise	99.9/0.0/0.0/**0.0**	24.6/0.0/**0.0**/0.1	50.9/100/**100**/99.80	71.8/100/**100**/99.90	35.1/100/**100**/98.80
	EMNIST	10.1/83.5/87.3/**0.3**	7.3/13.6/41.8/**1.0**	98.1/62.0/60.0/**99.94**	98.3/62.0/60.0/**99.94**	98.1/66.6/61.6/**99.95**
Modified	K-MNIST	NA/NA/NA/**1.3**	NA/NA/NA/**2.5**	NA/NA/NA/**99.74**	NA/NA/NA/**99.74**	NA/NA/NA/**99.75**
SEnet	MNIST	2.4/70.2/87.4/**0.0**	3.6/76.7/67.0/**0.0**	99.5/76.7/67.0/**100**	99.5/73.2/65.2/**100**	99.4/77.3/64.8/**100**
(Fashion	not-MNIST	7.2/80.2/78.9/**0.0**	5.8/33.9/32.2/**0.95**	97.8/69.3/73.7/**98.94**	97.4/63.0/73.0/**99.64**	**98.2**/70.5/72.4/94.53
MNIST)	Omniglot	58.4/9.6/59.8/**0.0**	26.8/7.1/22.1/**0.95**	83.2/97.9/85.6/**98.65**	84.9/97.6/85.8/**99.27**	83.4/**98.2**/85.1/95.71
	Uniform Noise	1.7/99.4/71.0/**0.0**	3.3/24.7/16.4/**0.85**	**98.9**/74.7/88.6/98.84	99.2/82.9/91.8/**99.38**	**97.9**/61.6/82.9/95.84
	Gaussian Noise	99.7/4.5/32.2/**0.0**	19.9/3.8/9.6/**0.85**	80.0/98.0/95.8/**99.09**	87.0/96.7/96.7/**99.48**	66.3/95.6/94.7/**97.43**

Table 1: EXPERIMENTAL RESULTS, *Modified SEnet evaluation metrics tested on different datasets. For each experiments and dataset the best best performance is in bold.*

4 FEATURE VISUALIZATION FOR OOD SAMPLES

Here in our approach, we are bringing the high dimensional dataset into 2D then bringing it into total number of training classes. With 2D features, we can visualize how the OOD samples are equally distributed for all know classes and giving less magnitude for probability value as shown in Figure 6c.

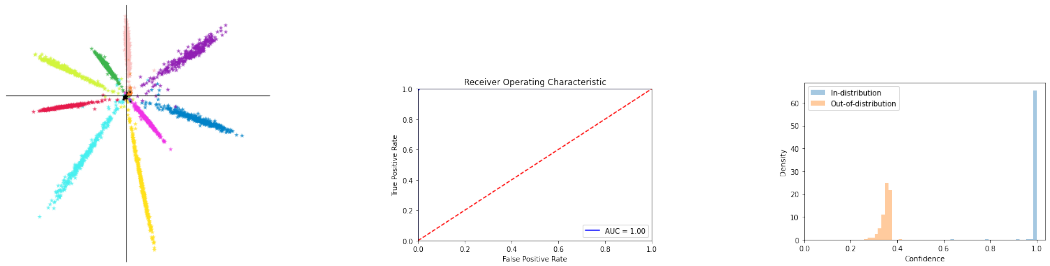
To get a broad understanding of our model response for In-class and OOD samples, we have to visualize the features and probability distribution coming from the end layer, which helps us to explain our approach even better. We train an Alexnet model without considering the low-level structural features at each layer and visualize the distribution of each class with Equation 4 and including one class for OOD samples as shown in Figure 5, here in this case model can classify each class but it can't proper discriminate the OOD samples from In-class samples. Meanwhile, after adding the low-level structural feature information with modified SEnet block Figure 1, the model can classify every In-class accurately and it can even discriminate OOD samples from In-class samples efficiently. We had trained our model with MNIST and Fashion MNIST D_{in} and then test with OOD sample D_{out} like EMNIST, KMNIST, FMNIST, Omniglot, Gaussian noise, Uniform noise. In Figure 5 we sampled OOD samples(Black Color sample) from the EMNIST dataset(colored sample) which is trained for MNIST classification and visualizes the output from Alex-Net, while Figure 5c is the histogram of the softmax score for In-class and OOD samples.



(a) (b) (c)

Figure 5: *Alex-Net model with IN-Class(MNIST, colored samples) and OOD class(EMNIST, black color samples) a) Data distribution from model b) AUROC evaluation and c) plot of data density with probability confidence*

In Figure 6a is the output samples from our model with In-class and OOD class samples, there it is quite clear that the OOD samples are having quite less probability score for OOD samples than In-class while Figure 6c shows the histogram for softmax score for In-class and OOD samples.



(a) (b) (c)

Figure 6: *Our Modified SEnet model with IN-Class(MNIST, colored samples) and OOD class(EMNIST, black color samples) a) Data distribution from model b) AUROC evaluation and c) plot of data density with probability confidence*

Apart from the confidence score for each class, we had visualized the second last layer output distribution of our model that helps us to understand how our loss function can maximize the margin between In-class and OOD samples and how the OOD samples normalization affects the magnitude of distribution.

5 Experiments & Results

We trained our model with MNIST and Fashion MNIST as an in-class sample and several other OOD samples. We compared our method with confident-classifier which is a classifier-based method for OOD detection like ODIN and Mahalanobis distance-based approach.

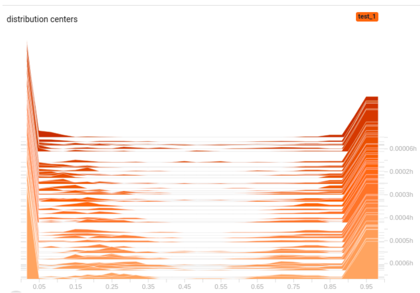
5.1 OOD Datasets

- **EMNIST-letters** [Cohen et al. (2017)] contains hand written english alphabets from a to z.
- **NotMNIST** [Bulatov (2011)] contains engilish character from A through I with different fonts.
- **KMNIST** [Prabhu (2019)] contains Japanese hand written alphabet has 49 classes (28×28 grayscale, 270,912 images) similar to MNIST.
- **Omniglot** [Lake et al. (2015)] contains handwritten characters from 50 different alphabets, to test with our model we downsampled to 28×28.
- **Uniform noise** contains gray-scale images where each pixel is sampled from independent uniform distribution within the range [0,1].
- **Gaussian noise** contains gray-scale images where each pixel is sampled from independent gaussian distribution with 0.5 mean and unit variance.

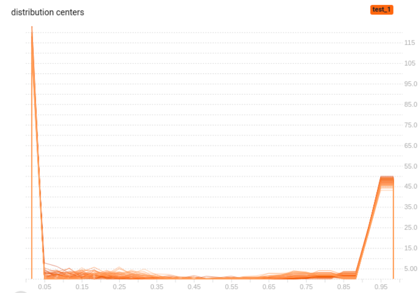
5.2 Results

We present our results in Table 1. The table having a combination of two rows containing various OOD dataset classification problems derived from the same In-class dataset. For example, row 1 having the results for an OOD classification problem for MNIST data as In-class distribution and EMNIST, K-MNIST, F-MNIST, and not-MNIST as an OOD dataset, while row2 having the results for an OOD classification problem for Fashion-MNIST data as In-class distribution and EMNIST, K-MNIST, MNIST, and not-MNIST as an OOD dataset. The evaluation metrics we calculated are in subsection 3.7 with state-of-the-art results in bold.

Here in Figure 7, we illustrate the effect of uniform normalization for the OOD dataset, distribution with higher magnitude belongs to the In-class image while lower one is from OOD data distribution. We can visualize it too that after processing through our proposed approach, the OOD samples are having lesser SoftMax score than In-distribution samples.



(a)



(b)

Figure 7: *Data distribution from 2D features from our model a)For entire batch b) overlay view*

6 Conclusion and Future Work

We presented a robust approach for OOD dataset, classification method without much drop in In-class classification accuracy. Here our model learns the low-level contextual features along with high level features at each layer with modified SEnet block. The loss functions had played the crucial roles for maximizing the margin between the In-class and OOD class data distribution. We also achieved the state-of-the-art (S.O.T.A.) results for few evaluation metrics as in Table 1. Our approach can give better and more reliable classification, AI system for real word application.

The future work for our approach will be to get better and robust object detection and segmentation algorithm for real time systems.

References

Alexander A. Alemi, Ian Fischer, and Joshua V. Dillon. Uncertainty in the variational information bottleneck. *UAI*, abs/1807.00906, 2018.

Dario amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *CoRR*, abs/1606.06565, 2016.

Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Bmvc*, pp. 3, 2016.

Petra Bevandic, Ivan Kreso, Marin Orsic, and Sinisa Segvic. Discriminative out-of-distribution detection for semantic segmentation. *CoRR*, abs/1808.07703, 2018.

Yaroslav Bulatov. Notmnist dataset. *Google (Books/OCR), Tech. Rep.[Online]. Available: http://yaroslavvb. blogspot. it/2011/09/notmnist-dataset. html*, 2, 2011.

Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In Yoshua Bengio and Yann LeCun (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. *in NeurIPS 2019*, 2019.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pp. 233–240, 2006.

C. De Stefano, C. Sansone, and M. Vento. To reject or not to reject: that is the question-an answer in case of neural classifiers. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 30(1):84–94, 2000. doi: 10.1109/5326.827457.

Akshay Raj Dhamija, Manuel Günther, and Terrance E Boult. Reducing network agnostophobia. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 9175–9186, 2018.

Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

Ragnar Fjelland. Why general artificial intelligence will not be realized. *Humanities and Social Sciences Communications*, 7(1):1–9, 2020.

Giorgio Fumera and Fabio Roli. Support vector machines with embedded reject option. *Springer-Verlag*, pp. 68–82, 2002.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *in ICLR*, 2016.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *in ICLR*, 2019.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5385–5394, 2016.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Bouchra Lamrini, Augustin Gjini, Simon Daudin, Pascal Pratmarty, François Armando, and Louise Travé-Massuyès. Anomaly detection using similarity-based one-class svm for network traffic characterization. In *DX@ Safeprocess*, 2018.

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *in ICLR*, 2018a.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *in NIPS*, 2018b.

Shiyu Liang, Yixuan Li, and R Srikant. Principled detection of out-of-distribution examples in neural networks. *CoRR*, pp. 655–662, 2017.

Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. *in NIPS*, 2018.

Christopher Manning and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? *in ICLR*, 2019.

Vinay Uday Prabhu. Kannada-mnist: A new handwritten digits dataset for the kannada language. *arXiv preprint arXiv:1908.01242*, 2019.

Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A DePristo, Joshua V Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. *in NeurIPS 2019*, 2019.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L Willke. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 550–564, 2018.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

Pan Zhou, Jiashi Feng, Chao Ma, Caiming Xiong, Steven HOI, et al. Towards theoretically understanding why sgd generalizes better than adam in deep learning. *in NIPS*, 2020.