

# Test case prioritization with z-score based neuron coverage

Hyekyoung Hwang\*, Jitae Shin\*

\* Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, 16419, Korea

ristar1234@skku.edu, jtshin@skku.edu

**Abstract**— Deep neural networks (DNNs) have been widely used in various applications, such as autonomous driving, healthcare, etc. However, despite achieving high accuracy, DNNs have exhibited quality issues in various aspects such as vulnerability to data corruptions, adversarial attacks, and data dependencies. To ensure the integrity and reliability of these systems, the use of systematic verification and validation methodologies before the deployment of DNN is considered an indispensable technique. Test case prioritization techniques reduce the cost of DNN verification by prioritizing test cases that could induce mispredictions of DNN. In this paper, we propose a neuron coverage-based test case prioritization technique for the DNN classifier that assigns sample priorities based on the ratio of outlier-valued neurons among total neurons in DNN. We evaluate the proposed method with three publicly accessible datasets with different sizes of DNN. The experimental results demonstrate that the proposed method outperforms the existing state-of-the-art neuron coverage-based approach both in error-inducing sample prioritization effectiveness and inference time efficiency.

**Keywords**— Test case prioritization, Neuron coverage, Deep neural network

## I. INTRODUCTION

Deep learning (DL) has achieved remarkable accomplishments, showing its potential capabilities in tackling intricate tasks in diverse advanced real-world applications. These include tasks such as image classification [1], natural language processing [2]. In addition, there is a growing demand for the integration of DL models into safety-critical domains such as autonomous driving [3], healthcare [4].

However, despite achieving high accuracy, deep neural networks (DNNs) have exhibited quality issues in various aspects such as vulnerability to data corruptions [5], adversarial attacks [6] and data dependencies (e.g., bias [7]). Real-world incidents, such as autonomous car accidents and incorrect clinical diagnosis, have already emphasized the need for these quality assessment issues. Thus, revealing system problems and fine-tuning are necessities for model deployment in the aspect of DL safety [8].

The conventional DNN evaluation uses a train-test split; one always has corresponding labels to the test set. However, one may attempt to evaluate a DNN with new collections of test sets, which may not have labels, so that it requires additional costs for annotating them.

To mitigate this limitation, researchers have proposed test case prioritization (TCP) techniques [9 - 16], which aim to identify and prioritize instances in an unlabelled test set that could reveal the weakness of the DL model. Since TCP approaches prioritize error-inducing instances within a predefined number of annotation budgets, they effectively reduce the manual cost of labelling. In general, TCP approaches extract features of a DNN model on a training set and define patterns from the features that represent the common behaviour of the trained model. Then, they calculate the similarity between the patterns and a feature of the DNN model on an unlabelled test instance. Some structural coverage criteria are proposed to represent patterns from features of a training set, for example, neuron coverage (NC) [9] and its variants [10]-[14], surprise adequacy [15], mutation analysis [16], and so on.

NC-based TCP methods [9]-[14] are inspired by traditional white-box software tests. They extract the patterns of a training set in terms of neuron activation. From a training set, statistics (e.g., interval of neuron activation values) for all neurons or a threshold for each neuron activation are regarded as the patterns. The patterns are then applied to neuron activation of each unlabelled test input to provide the score for a sample prioritization. In general, TCP techniques give higher priority to samples with a higher score.

FD+ [14] achieves state-of-the-art performance in the NC-based TCP approach by extracting the most specified patterns based on a training set. It collects the output of all neurons over a whole training set and builds the pattern of each neuron for all classes based on exact percentile values. Thus, it requires larger time and storage for the pattern extraction and slower inference time due to its complexity.

To moderate this problem, we propose a neuron-wise threshold based on z-score. Unlike FD+ which uses exact percentile values, we approximate the percentile values based on the z-score. We refer to the proposed method as z-score neuron coverage (ZSNC). The experimental result shows that the proposed ZSNC outperforms the current state-of-the-art NC-based TCP, FD+ [14] in both effectiveness and total time consumption. Specifically, the proposed ZSNC is 7.18 times faster in pattern extraction from a training set and 2.39 times faster in total TCP inference.

The remainder of the paper is organized as follows. Section 2 introduces NC-based TCP techniques and the current state-of-the-art NC-based approach. Section 3 provides the

motivation and details of the proposed ZSNC. Section 4 demonstrates the experimental results. Finally, we conclude the paper in Section 5.

## II. RELATED WORKS

Recently, software engineering researchers have proposed several NC-based TCP techniques. The goal of NC-based TCP technique is to effectively identify test instances that reveal errors of a trained DNNs and prioritize the error-inducing samples in predefined number of labelling budget.

### A. Test case prioritization based on neuron coverage

Inspired by the white box testing of traditional software coverage testing, [9] was the first to propose NC as a testing metric for DNN. Given a test input  $x$ , they define the NC of  $x$  as the ratio of the number of "activated" neurons to the total number of neurons, denoted as  $NC(x, \varepsilon)$ . The parameter  $\varepsilon$  is a threshold that determines the conditions under which neurons in a DNN are considered "activated" or not. In general,  $\varepsilon$  is defined by users and is commonly set to 0. Thus,  $NC(x, \varepsilon)$  represents the ratio of the number of neurons whose output values exceed  $\varepsilon$  to the total number of neurons. The authors in [9] assume that a sample with a higher  $NC(x, \varepsilon)$  value is more difficult compared to the lower one. The assumption stems from the fact that a higher NC-valued sample requires the greater number of neuron explorations in a DNN.

Inspired by this work, many DL testing research focus on designing coverage metrics, such as coverage based on different  $\varepsilon$  settings [10, 11], coverage based on layer level [12], and class-wise familiarity degree [14].

### B. Familiarity degree-based Neuron Coverage

The authors of [14] propose an NC-based TCP approach, FD+, that calculates the degree of familiarity based on the valuation pattern of each neuron. FD+ is motivated by the observation that the output values of the neurons in error-inducing samples deviate from the distribution of those of correctly predicted samples. Thus, FD+ extracts class-specific patterns for each neuron as the common patterns of the DNN model for a training set.

Firstly, it collects the activation value of all neurons for all instances of a training set. Based on the collection of activation output, each neuron is classified as "active" or "inactive" for all classes. When a class is given, a neuron is classified as "active" if the majority (e.g., 95%) of the collected neuron output for the given class are greater than zero, otherwise "inactive". Given  $N$  number of neurons,  $N_{a,k}$  represents the set of "active" neurons and  $N_{i,k}$  is the set of "inactive" neurons for class  $k$ .

In addition, they introduce different thresholds to  $N_{a,k}$  and  $N_{i,k}$  to classify the neurons in each set more specifically. A maximal lower bound,  $l_{n,k}$ , is provided to each of "active" neuron, which is the quantile value of 5% of the output distribution of the "active" neurons. A minimal upper bound,  $u_{n,k}$ , is provided to each of "inactive" neuron, which is the quantile value of 95% of the output distribution of the "inactive" neurons. When an unlabelled test input and its prediction class  $k$  are given, each neuron is classified into one

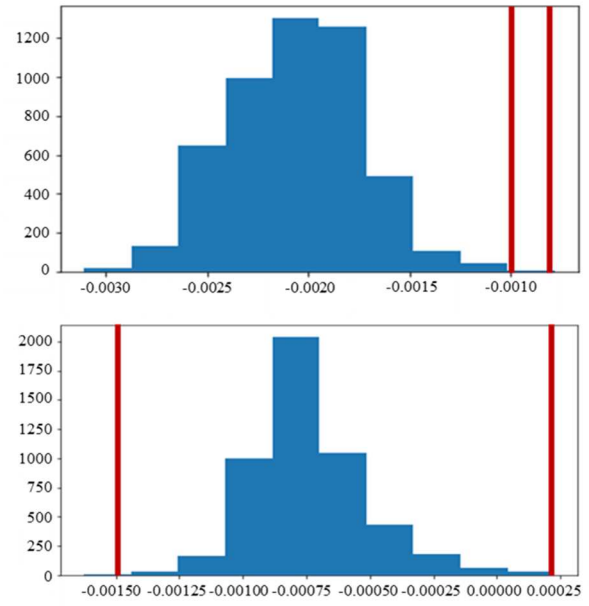


**Figure 1** Feature distributions of class "airplane" for 1<sup>st</sup>, 2<sup>nd</sup> neurons in 12-th layer of VGG16, trained on CIFAR-10, respectively. The red lines represent the neuron output of samples with its ground truth label "airplane", while their predictions are not "airplane".

of two classes: "triggered", "inhibited". A neuron is classified as "triggered" when the neuron is "active" for class  $k$  and its output value from a given test input is greater than or equal to  $l_{n,k}$ . A neuron is classified as "inhibited" when the neuron is "inactive" for class  $k$  and its output value from a given test input is smaller than or equal to  $u_{n,k}$ .

Finally, the degree of familiarity (FD+) of an unlabelled test instance is the ratio of "inhibited" or "triggered" neurons in the total  $N$  neurons,  $(N_t + N_i)/N$ , where  $N_t$  is the number of "triggered" neuron and  $N_i$  is the number of "inhibited" neuron.

However, FD+ requires a higher cost than other NC-based methods. Saving outputs of all neurons for all training instances are required to classify the activeness of each neuron for all classes and calculate the lower/upper bounds. This slows down the pattern extraction process of FD+. Also, comparing each neuron twice makes a longer inference time.

To moderate this challenge, we propose a TCP approach with z-score based interval for each neuron. The proposed method observes the ratio of neurons that do not fall within an interval for a given class. The upper and lower bounds of the interval consist of z-scores of neuron output values of a specific class.

## III. PROPOSED METHOD

### A. Motivation

Since various activation functions have been employed, it is inappropriate to distinguish the state of each neuron based on whether the output value of the neuron exceeds 0. Furthermore, similarly to FD+ [14], we observed that the output values of a neuron for incorrect predictions deviate from the distribution of those of correctly predicted samples. Figure 1 displays examples of our observations.

A z-score is the number of standard deviations that is from the mean of a given distribution. Negative z-score indicates the value lies below the mean. Positive z-score indicates the value lies above the mean. Anomaly detection based on z-scores [18], one defines the anomalous sample as the one does not fall within the z-score based interval, is a classic approach and has been used in deep learning to detect outliers in the output of specific layers [19]. Based on the aforementioned observation and fact, we set a common pattern of each neuron for a training set as an interval based on the z-scores.

### B. Z-Score based Neuron Coverage (ZSNC)

We assume that there are  $N$  neurons in a DNN classifier  $f(\cdot)$  with total  $C$  classes. We denote a set of outputs of  $N$  neurons comprising  $f(\cdot)$  for an input  $x$  as  $N_{f(x)}$ , and we have a total of  $T$  training instances. The total  $T$  instances are divided into  $C$  number of subsets,  $T = \{T_1, T_2, \dots, T_C\}$ , where each  $T_i$  is a subset of the training set based on the ground truth class  $i$ .

The pattern extraction process of FD+ [14] can be formulated as follows: 1) Collecting the  $N_{f(T)}$  and separate it to  $C$  subsets based on the ground truth classes  $N_{f(T)} = \{N_{f(T_1)}, \dots, N_{f(T_C)}\}$ . 2) Classifying each neuron into “active” or “inactive” based on  $N_{f(T_i)}$ . 3) Extracting values at the  $a\%$  and  $(100-a)\%$  percentile from each of  $N_{f(T_i)}$  to determine class-specific lower and upper bounds.

However, the pattern extraction process of the proposed method does not require the first two stages of that of FD+. Also, we approximate the class-specific lower and upper bounds for each neuron based on the mean and standard deviation of neuron outputs for each class.

Given a  $t$ -th training instance  $x_t$  with corresponding label  $c$ , we iteratively calculate the mean and standard deviation of  $N$  neurons for class  $c$  as follows:

$$\mu_t^c = \mu_{t-1}^c + \frac{1}{t} (N_{f(x_t)} - \mu_{t-1}^c), \quad (1)$$

$$(\sigma_t^c)^2 = \frac{t-2}{t-1} (\sigma_{t-1}^c)^2 + \frac{1}{t} (N_{f(x_t)} - \mu_{t-1}^c)^2. \quad (2)$$

Here,  $\mu_t^c$  is a set that consists of the mean of  $t$ -th iteration for class  $c$  and  $\sigma_t^c$  is a set that contains the standard deviation of  $t$ -th iteration for class  $c$ . Both  $\mu_t^c$  and  $\sigma_t^c$  consists of  $N$  elements that each of them implies the mean and standard deviation of each neuron in  $t$ -th iteration for class  $c$ , respectively.

When the iteration is completed for all classes,  $C$  sets of mean  $\{\mu^1, \dots, \mu^C\}$  and standard deviation  $\{\sigma^1, \dots, \sigma^C\}$  are collected. For simplicity, we omit the number of iterations for each class. The mean and standard deviation of a  $n$ -th neuron for class  $i$  is denoted as  $\mu_n^i$  and  $\sigma_n^i$ , respectively. Then, we set an interval for  $n$ -th neuron for a class  $i$  as  $[\mu_n^i - \epsilon \cdot \sigma_n^i, \mu_n^i + \epsilon \cdot \sigma_n^i]$ , where  $\epsilon$  is a scaling factor and selected from the z-score table.

Finally, when an unlabelled test input and its prediction result  $k$  is given, we calculate the ratio of neurons whose output value does not fall within the range of the interval  $[\mu_n^k - \epsilon \cdot \sigma_n^k, \mu_n^k + \epsilon \cdot \sigma_n^k]$ . The proposed z-score-based neuron coverage is

$$ZSNC(x) = \frac{|\{n \mid n \in N, f_n(x) \notin [\mu_n^k - \epsilon \cdot \sigma_n^k, \mu_n^k + \epsilon \cdot \sigma_n^k]\}|}{N} \quad (3)$$

where  $f_n(x)$  is the output of  $n$ -th neuron on a test input  $x$  and  $|\cdot|$  represents a cardinality of a set.

For training set pattern extraction, [14] have to store  $N \times T$  values and then classify the active status and calculate the bounds of intervals. In contrast, our proposed method calculates the mean and standard deviations iteratively so that it does not require separate storage spaces. During an inference, the proposed method does not classify each neuron with ‘active’/‘inactive’ status so that it has leading to reduced inference time.

## IV. EXPERIMENTAL RESULTS

In this section, we describe the experimental setups and analyse the experimental results.

### A. Datasets and Models

To evaluate our method, we adopt three datasets for image classification benchmark: SVHN [20], CIFAR10 and 100 [21]. Additionally, to evaluate the performance of TCP in real-world scenarios, we employ real-world corruptions [22] to CIFAR10 dataset, which is notated with ‘C’ after the dataset. We choose 19 different corruptions such as motion blur, brightness, shot noise, etc.

We tested the proposed method on different sizes of models: VGG16 [23], ResNet50 [24], WideResNet50\_2 (WRN50\_2) [25], MobileNetV2 (MobileV2) [26], and EfficientNetV2\_S (EfficientV2S) [27].

Table 1 represents details about our evaluation setups. The dataset and models, classification accuracy of the model on training and test data, and the number of training and test data are shown. For simplicity, we describe each experiment as their ID. The experiment IDs F to H represents CIFAR10 with real-world corruptions, while the trained model is same with the experiment IDs B to D.

TABLE 1 EXPERIMENTAL SETUPS FOR EVALUATION

| ID | Data-Model            | Train Acc. (%) | Test Acc. (%) | # Train | # Test  |
|----|-----------------------|----------------|---------------|---------|---------|
| A  | SVHN-WRN50_2          | 94.01          | 92.38         | 73,257  | 26,032  |
| B  | CIFAR10-VGG16         | 99.97          | 93.81         | 60,000  | 10,000  |
| C  | CIFAR10-MobileV2      | 99.51          | 93.38         | 60,000  | 10,000  |
| D  | CIFAR10-ResNet50      | 98.69          | 93.78         | 60,000  | 10,000  |
| E  | CIFAR100-EfficientV2S | 99.52          | 88.20         | 60,000  | 10,000  |
| F  | CIFAR10C - VGG16      | -              | 29.56         | -       | 190,000 |
| G  | CIFAR10C - MobileV2   | -              | 26.91         | -       | 190,000 |
| H  | CIFAR10C - ResNet50   | -              | 29.71         | -       | 190,000 |

**TABLE 2** COMPARISON OF ATPF (%) WITH DIFFERENT NC-BASED TCP TECHNIQUES

| ID   | NBC [10] | NLC [12] | FD+ [14] | ZSNC   |
|------|----------|----------|----------|--------|
| A    | 23.381   | 23.296   | 72.693   | 71.121 |
| B    | 59.579   | 69.21    | 48.962   | 83.631 |
| C    | 51.835   | 34.818   | 70.376   | 78.265 |
| D    | 56.071   | 40.109   | 35.464   | 86.905 |
| E    | 10.616   | 12.264   | 28.81    | 30.447 |
| F    | 72.677   | 59.827   | 73.664   | 84.343 |
| G    | 76.231   | 77.296   | 85.98    | 83.412 |
| H    | 71.18    | 70.158   | 66.831   | 86.96  |
| Avg. | 52.696   | 48.372   | 60.348   | 75.511 |

### B. Compared Methods

We compared the proposed ZSNC approach with the three existing Neuron Coverage-based TCP methods.

**Neuron boundary coverage (NBC)** [11] targets to observe the number of neurons that does not lies between lower and upper bound of the outputs of a neuron upon training set. The lower/upper bound is set to be the minimum/maximum values of the total neuron outputs upon the training set. The proposed ZSNC differs from NBC in that it measures class-specific boundaries for each neuron and replaces lower/upper boundaries with z-scores.

**Neural coverage (NLC)** [12] defines the coverage criteria as the covariance of layer output. It supposes that a sample with higher covariance between the layer outputs leads more diversity to test set. It does not requires pattern extraction from training set.

**Familiarity degree (FD+)** [14] is described in Section 2, one requires two steps to determine the status (inhibited, triggered) of each neuron. FD+ requires saving neuron outputs for the entire training data separately for setting the lower/upper bound thresholds to distinguish the status of neuron. In contrast, the proposed ZSNC is single-step and does not require storing neuron outputs for the entire training data. According to the setup in [14], we set maximal lower bound for active neuron as the 5%, while the minimal upper bound is 95% of the values.

We set the scaling factor for proposed method as 1.96, which implies the 95% of confidence intervals in z-score table.

### C. Evaluation Metric

For evaluation of **effectiveness**, we used the Average Test Percentage of Fault (ATPF) [17] to measure the error-revealing identification performance of TCPs. ATPF is the average of the ratios of the number of erroneous instances detected within the  $n$ -th labelling budget:

$$\text{ATPF}(\%) = 100 \cdot \frac{1}{N_{fail}} \sum_{n=1}^{N_{fail}} \frac{N_{err,n}}{n} \quad (4)$$

where  $N_{fail}$  is the total number of erroneous instances in a test set and  $N_{err,n}$  is the number of error-inducing instances detected in the labelling budget,  $n$ .

To evaluate an **efficiency** of proposed method, we measure the time required for the pattern extraction from and the total inference time of NC-based TCP techniques.

### D. Evaluation Result: Effectiveness

Table 2 shows the comparison of ATPF with different NC-based TCP techniques. The final row displays the average ATPF across all experiments. A higher ATPF value implies a more effective identification of error-inducing samples within the same labelling budget. The proposed ZSNC takes the first place for five among seven experiments.

Even though the NLC is the most recent approach, the performance of NLC is less than the others. It is because the initial purpose of NLC is to compare the diversity between the different sets, not to find error-inducing samples. The comparison between NBC and the proposed ZSNC shows that employing class-specific threshold for each neuron can lead the meaningful performance enhancement based on the more detailed patterns from the training set. A comparison of FD+ with the proposed ZSNC reveals the superiority of single staged TCP and z-score-based intervals over the sampling of exact quantile samples.

### E. Evaluation Result: Efficiency

Table 3 displays the time for pattern extraction for each neuron from a training set, in seconds. The bottom row displays the average time across all experiments. Experiment IDs F to H are omitted because they are designed to observe inferences on real-world corruptions, using the same model with experiment IDs B to D.

NLC is marked with '-' in the table because it does not require the pattern extraction from the training set. Among the other NC-based TCP methods, NBC takes the shortest pattern extraction time because it extracts only the minimum and maximum values for each neuron from the entire training set. But, the ATPF value of NBC is always smaller than that of the proposed method and FD+.

The proposed ZSNC takes the second fastest pattern extraction as it extracts class-specific thresholds for each neuron iteratively. FD+ takes the longest time since it requires the storage for saving the activation values of all neurons on a whole training set and extracts exact percentile values as their bounds of intervals. The proposed method takes about three times longer on average than NBC but is approximately 7.19 times faster compared to the latest algorithm, FD+.

**TABLE 3** COMPARISON OF TIME (SEC.) REQUIRED FOR PATTERN EXTRACTION OF TRAINING SET.

| ID   | NBC [10] | NLC [12] | FD+ [14] | ZSNC    |
|------|----------|----------|----------|---------|
| A    | 47.61    | -        | 1463.162 | 149.39  |
| B    | 7.604    | -        | 244.659  | 23.95   |
| C    | 17.423   | -        | 402.778  | 51.266  |
| D    | 20.094   | -        | 561.45   | 59.297  |
| E    | 135.924  | -        | 2311.569 | 409.675 |
| Avg. | 45.731   | -        | 996.724  | 138.716 |

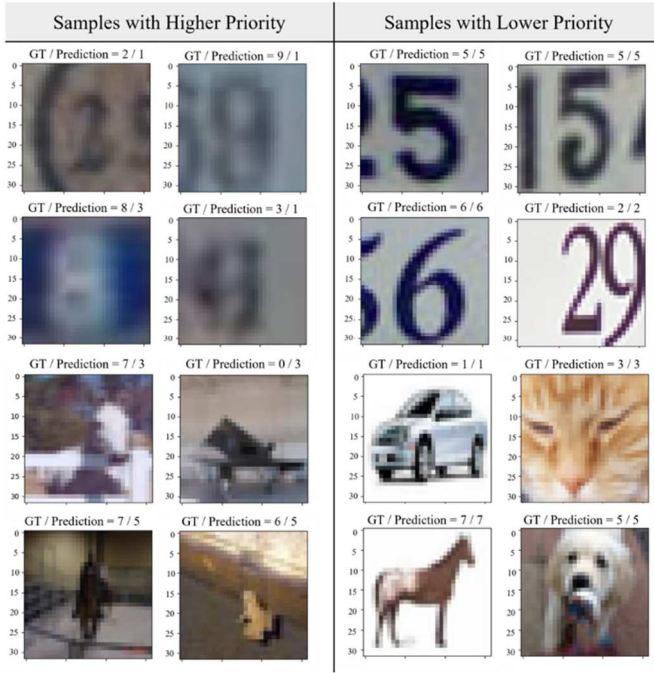


**Figure 2** Sample visualizations with higher/lower priority based on proposed ZSNC. (Left) samples with high priority based on ZSNC, (Right) samples with low priority based on ZSNC.

**TABLE 4** COMPARISON OF THE TOTAL INFERENCE TIME (SEC.)

| ID   | NBC [10] | NLC [12] | FD+ [14] | ZSNC    |
|------|----------|----------|----------|---------|
| A    | 15.946   | 43.762   | 73.357   | 26.345  |
| B    | 1.756    | 12.019   | 9.855    | 5.523   |
| C    | 2.801    | 5.965    | 23.839   | 5.547   |
| D    | 3.809    | 11.856   | 25.763   | 7.329   |
| E    | 27.156   | 66.669   | 90.85    | 36.869  |
| F    | 19.705   | 118.228  | 80.027   | 45.69   |
| G    | 44.741   | 71.141   | 215.173  | 72.655  |
| H    | 55.218   | 167.037  | 235.27   | 106.947 |
| Avg. | 21.392   | 62.085   | 94.267   | 38.363  |

Table 4 represents a comparison of total inference time of different NC-based TCP techniques for seven different classification experiments in seconds. The bottom row displays the average across all experiments.

Consistent with the results in Table 3, NBC enables the fastest inference. The proposed method achieves the second fastest total inference time, by exhibiting 1.79 times longer than NBC in average. However, the most recent algorithm [12] and the state-of-the-art in NC-based TCP [14] takes 1.58 times and 2.46 times longer than the proposed method, respectively. Furthermore, the proposed method outperforms the other NC-based TCP methods in error-inducing sample detection performance, making it the most proper option among the NC-based TCP methods.

## F. Sample Visualization

Figure 2 shows the samples with higher/lower priority based on the proposed ZSNC, respectively. Samples that received a high priority by the proposed method tend to exhibit hard conditions such as indistinct differentiation from the background, low lighting conditions, or misalignment of objects. On the other hand, samples with lower priorities present easy conditions such as clear distinction between the background and the object, distinctive characteristics of prediction class, such as animal faces, etc.

## V. CONCLUSIONS

As the importance of thorough testing before deploying DL models has increased due to higher safety standards, test case prioritization has emerged as an effective solution by prioritizing error-inducing samples for testing, thereby reducing testing costs. This paper introduces a Z-score-based outlier detection approach to existing neuron coverage-based test case prioritization. The proposed method demonstrates superior error-inducing sample detection capabilities compared to the state-of-the-art NC-based methods across various experimental settings, and it also significantly reduces both pattern extraction and total inference time, performing approximately 7.19 and 2.46 times faster, respectively.

## ACKNOWLEDGMENT

This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government Ministry of Science and ICT (MSIT) (NRF-2020R1F1A1065626), in part by the MSIT under the Information Technology Research Center (ITRC) support program (IITP-2023-2018-0-01798) supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP), and partly supported by the BK21 FOUR Project.

## REFERENCES

- [1] K. B. Obaid, S. Zeebaree & O. M. Ahmed, "Deep learning models based on image classification: a review". *International Journal of Science and Business*, vol. 4(11), pp. 75-81, October 2020.
- [2] D. W. Otter, J.R. Medina, & J.K. Kalita, "A survey of the usages of deep learning for natural language processing". *IEEE Trans. on neural networks and learning systems*, vol. 32(2), pp. 604-624, April 2020.
- [3] Almalioglu, Y., Turan, M., Trigoni, N., & Markham, A. "Deep learning-based robust positioning for all-weather autonomous driving." *Nature Machine Intelligence*, vol. 4(9), pp. 749-760, September 2022.
- [4] A. Esteva, K. Chou, S. Yeung, N. Naik, et al. Deep learning-enabled medical computer vision. *NPJ digital medicine*, vol. 4(1), 5, January 2021.
- [5] E. Rojas, D. Pérez & E. Meneses. "Exploring the effects of silent data corruption in distributed deep learning training". in Proc. SBAC-PAD'22, 2022, pp. 21-30)
- [6] N. Akhtar, & A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey". *Ieee Access*, vol. 6, pp. 14410-14430, February 2018.
- [7] G. Vardi, G. "On the implicit bias in deep-learning algorithms". *Communications of the ACM*, vol. 66(6), pp. 86-93, June 2023.
- [8] F. Tambon, F. G. Laberge, L. An, A. Nikanjam, P. S. N Mindom, Y. Pequignot, ... & F. Laviolette, "How to certify machine learning based safety-critical systems? A systematic literature review." *Automated Software Engineering*, vol. 29(2), pp. 38, April 2022.

- [9] K. Pei, Y. Cao, J. Yang, & S. Jana., “Deepxplore: Automated whitebox testing of deep learning systems”. in Proc. SOSP’17, 2017, pp. 1-18.
- [10] Y. Tian, K. Pei, K., S. Jana, & B. Ray, “Deeptest: Automated testing of deep-neural-network-driven autonomous cars”. in Proc. ICSE’18, 2018, pp. 303-314.
- [11] L. Ma, F. Juefei-Xu, F. Zhang, J. Sun, M. Xue, B. Li & Y. Wang, (2018, September). “Deepgauge: Multi-granularity testing criteria for deep learning systems”. in Proc. ASE’18, 2018, pp. 120-131.
- [12] Y. Yuan, Q. Pang, & S. Wang, “Revisiting neuron coverage for dnn testing: A layer-wise and distribution-aware criterion”. in Proc. ICSE’23, 2023, pp. 1200-1212
- [13] X. Xie, L. Ma, F. Juefei-Xu, M. Xue, H. Chen, Y. Liu, ... & S. See, “Deephunter: a coverage-guided fuzz testing framework for deep neural networks”. in Proc. ISSTA’19, 2019, pp. 146-157.
- [14] R. Yan, Y. Chen, H. Gao, & J. Yan, “Test case prioritization with neuron valuation based pattern”. *Science of Computer Programming*, vol. 215, pp. 102761, March 2022.
- [15] S. Kim, & S. Yoo, “Multimodal surprise adequacy analysis of inputs for natural language processing DNN models.” in Proc. AST’21, 2021, pp. 80-89.
- [16] Z. Wang, H. You, J. Chen, Y. Zhang, X. Dong, & W. Zhang, “Prioritizing test inputs for deep neural networks via mutation analysis”. in Proc. ICSE’21, 2021, pp. 397-409.
- [17] Y. Li, M. Li, Q. Lai, Y. Liu, & Q. Xu, “Testrank: Bringing order into unlabeled test instances for deep learning tasks”. in Proc. NIPS’21, 2021 pp. 20874-20886.
- [18] P. R. Bushel, H. K. Hamadeh, L. Bennett, J. Green, A. Ableson, S. Misener, ... & R. S. Paules, “Computational selection of distinct class- and subclass-specific gene expression signatures”. *Journal of biomedical informatics*, vol. 35(3), pp. 160-170, June 2002.
- [19] V. Aggarwal, V. Gupta, P. Singh, K. Sharma & N. Sharma, (2019, April). “Detection of spatial outlier by using improved Z-score test”. in Proc. ICOEI’19, 2019, pp. 788-790.
- [20] N. Yuval, W. Tao, C. Adam, B. Alessandro, W. Bo, Y. Ng. Andrew, “Reading Digits in Natural Images with Unsupervised Feature Learning” *NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011*.
- [21] A. Krizhevsky, “Learning multiple layers of features from tiny images”, M. Eng. thesis, University of Tront, 2009.
- [22] D. Hendrycks, & T. Dietterich. “Benchmarking Neural Network Robustness to Common Corruptions and Perturbations”. in Proc. ICLR’18, 2018
- [23] K. Simonyan & A. Zisserman, “Very deep convolutional networks for large-scale image recognition.” ICLR’15, 2015
- [24] K. He, X. Zhang, S. Ren, J. Sun, “Identity mappings in deep residual networks”. in Proc. ECCV’16, 2016, pp. 630-645.
- [25] S. Zagoruyko, & N. Komodakis, “Wide Residual Networks”. in Proc. BMVC’16, 2016
- [26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks”. in Proc. CVPR’18, 2018, pp. 4510-4520.
- [27] M. Tan, Q. Le. “Efficientnetv2: Smaller models and faster training”. in Proc. PMLR’21, 2021, pp. 10096-10106.



**Hyekyoung Hwang** is a Ph.D student and she received her BS degree from the Department of Electronic and Electrical Engineering, College of Information and Communication Engineering and mathematics, College of Natural Science, Sungkyunkwan University, Republic of Korea in 2018. Her research interests include image processing and deep learning, focused on explainable AI with uncertainty estimation.



**Jitae Shin** has been a professor from 2002 with the School of Electronic and Electrical Engineering, Sungkyunkwan University, Suwon, South Korea. He received the B.S. degree from Seoul National University, in 1986, the M.S. degree from the Korea Advanced Institute of Science and Technology, in 1988, and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, USA, in 1998 and 2001, respectively. For former industrial experiences, he worked with Korea Electric Power Corporation and the Korea Atomic Energy Research Institute from 1988 to 1996. His current research interests include image/video signal processing using deep learning, medical image processing, and video transmission over wireless/mobile communication systems.