Benchmarking Robustness of Deep Learning Classifiers Using Two-Factor Perturbation

Wei Dai[1*] and Daniel Berleant[2]

[1]Southeast Missouri State University, Cape Girardeau, MO 6370,1 USA
[2]University of Arkansas at Little Rock, Little Rock, AR 72204, USA
[*]wdai@semo.edu

Abstract. The accuracy of deep learning (DL) classifiers is unstable in that it often changes significantly when retested on adversarial images, imperfect images, or perturbed images. This paper adds to the small but fundamental body of work on benchmarking the robustness of DL classifiers on defective images. Unlike existed single-factor digital perturbation work, we provide state-of-the-art two-factor perturbation that provides two natural perturbations on images applied in different sequences. The two-factor perturbation includes (1) two digital perturbations (Salt & pepper noise and Gaussian noise) applied in both sequences. (2) one digital perturbation (salt & pepper noise) and a geometric perturbation (rotation) applied in different sequences. To measure robust DL classifiers, previous scientists provided 15 types of single-factor corruption. We created 69 benchmarking image sets, including a clean set, sets with single factor perturbations, and sets with two-factor perturbation conditions. To be best of our knowledge, this is the first report that two-factor perturbed images improves both robustness and accuracy of DL classifiers. Previous research evaluating DL classifiers has often used top-1/top-5 accuracy, so researchers have usually offered tables, line diagrams, and bar charts to display accuracy of DL classifiers. But these existed approaches cannot quantitively evaluate robustness of DL classifiers. We innovate a new two-dimensional, statistical visualization tool, including mean accuracy and coefficient of variation (CV), to benchmark the robustness of DL classifiers. All source codes and related image sets are shared on websites (http://cslinux.semo.edu/david/data.html or https://github.com/daiweiworking/RobustDeepLearningUsingPerturbations) to support future academic research and industry projects.

Keywords: Robust Deep Learning, Imperfect images, Benchmark Metrics

1 Introduction

Computer scientists and engineers have innovated many benchmark tools to measure hardware devices and compare software algorithms. Well-known, non-profit benchmark organizations include Standard Performance Evaluation Corporation (SPEC) [1], Transaction Processing Performance Council (TPC) [2], Storage Performance Council (SPC) [3], and Machine Learning Performance (MLPerf) [4].

These evaluate CPUs, databases, storage, and machine learning respectively. MLPerf has measured the training and inference performance of ML hardware, software, and services. All these organizations aperiodically update their benchmark tools, retire obsolete benchmark programs, and officially publish on their websites.

Deep learning (DL) classifiers have been shown to work well on high quality image sets. DL classifiers have classified objects in high quality image sets, for example, at 97.3% accuracy, which is better than human capabilities [5]. However, low quality sets present a more challenging environment. If the data is imperfect, as real data so often are, how will results be affected? In [6], the researchers discovered that human visual systems are more robust than DL classifiers when images are manipulated by contrast reduction, additive noise, and eidolon distortions. In [7], DL classifier performance was lower than that of humans on recognizing images corrupted with Gaussian noise or Gaussian blur.

DL classifiers can make mistakes even on high quality images in the DL security field. For example, an adversarial image may be a high quality image modified by tiny perturbation s chosen to confuse DL classifiers. Even though humans may not notice these perturbation s, these deliberately modified images can confuse DL classifiers, reducing their accuracy [8] [9] [10]. Therefore, the robustness of DL classifiers across image perturbation conditions will continue to merit attention. To measure robustness of DL classifiers, we can expect a robust DL classifier to have both high accuracy as well as low variance across the perturbation conditions.

In this paper, our **contributions** are as follows:

Unlike previous single-factor perturbation work, we provide state-of-the-art two-factor perturbation that includes two digital perturbations (salt & pepper noise and Gaussian noise) applied in both sequences. Also, the two-factor perturbation may consist of one digital perturbation (salt & pepper noise) and a geometric perturbation (rotation) applied in different sequences. Briefly, two-factor perturbation provides two natural perturbations on images. For some examples see Figs. 7-11 at the end. To the best of our knowledge, this is a new approach to measuring robustness of deep learning classifiers.

To measure robust DL classifiers, previous scientists provided 15 types of single perturbation. We created 68 corrupted image test sets, each containing images corrupted in one of the 68 ways. Each way involved corrupting images twice, by two different perturbation s. These image sets and source codes are shared on a website (http://www.animpala.com/) to support additional research projects.

Compared with using only clean images, training with these two-factor corrupted image sets can improve the robustness of DL classifiers. To be best of our knowledge**,** this is the first report that two-factor perturbation improves both robustness and accuracy of DL classifiers. For details, see Table 3.

Previous research evaluating DL classifiers has often used top-1/top-5 accuracy. Previous researchers have usually offered tables, line diagrams, and bar charts to display accuracy of DL classifiers. However, there is not a quantitative tool to measure robust DL classifiers. In this work, we supplement those tools with a new statistical graphic, the mCV plot, that integrates two statistic metrics (mean accuracy and coefficient of variation (CV)) for enhanced understanding. It is the first time that

scientists visualize robust DL classifiers via such a statistical plot with better understand and less bias. For resulting mCV plots, see Figs. 2-6.

2 Related Work

Researchers have discovered that DL classifiers are fragile when faced with image perturbation s. In [11], the authors demonstrated that the Google Cloud Vision API could be misled after adding approximately 14.25% impulse noise density. In [12], DL classifiers had reduced accuracy of classification when testing five types of quality distortions, including JPEG/JPEG2000 compression, blur, Gaussian noise, and contrast. There have been various previously reported corruptions imposed on high quality ImageNet datasets. In [13] the authors created 15 types of single perturbation refers to modifying images using two types of perturbation , one after the other., including noise, blur, and brightness. In [14][7], the authors used Gaussian noise or Gaussian blur to change images. Previous research projects created defective images via single factor corruptions. Sixty-nine (69) benchmarking image sets, including a clean set, sets with single factor perturbation s, and sets with two-factor perturbation conditions, for benchmarking image sets were discussed in [15]. That study demonstrated that training DL classifiers on two-factor corrupted image sets could improve their robustness.

Previous researchers have measured the robustness of DL classifiers via top-1 or top-5 precision analyses [16]. For example [13] chose the average rates of correct ("Top-1") and almost correct ("Top-5") classifications, the authors setting the Top-1 error rate of AlexNet as the reference error rate. In contrast, we provide a two-dimensional metric, consisting of mean accuracy and coefficient of variation (CV). This supports characterizing the stability of DL classifiers as having high mean accuracy along with small coefficient variance of accuracies across perturbation conditions.

Note that the coefficient of variation defines the ratio of the standard deviation to the mean. The coefficient of variation is used in different fields, including physics, medicine, chemistry, and engineering [17] [18]. In agriculture, Francis and Kannenberg [19] used the mean yield and coefficient of variation to analyzing yield stability. The CV offers a way to help differentiate the variation from the mean because, compared to using variance, CV controls for the mean. So, when judging a DL classifier, we can measure the average accuracy, $\overline{x}$, and the coefficient of variation, CV, as they say different things. A high accuracy and small CV for a DL classifier is better than a low accuracy with a large CV.

Spearman's rank correlation coefficient is a statistic used for calculating the strength of a monotonic relationship between paired data (X,Y), in a sample N. Assume that a value of X has rank K and its corresponding value of Y has rank L. As shown in Eq. 1, Spearman's rank r_{xy} can be calculated for the sample data N.

$$r_{xy} = 1 - \frac{6 \sum d_{xy}^2}{n(n^2-1)} \tag{1}$$

where d is the difference between K and L, and $d_{xy}^2 = (K - L)^2$. The number of data in N is n.

The Pearson correlation coefficient is used for evaluating the strength of a linear relationship between two random variables. The Pearson correlation coefficient, r_{xy}, can be calculated as in Eq. 2.

$$r_{xy} = \frac{\sum_i^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i^n (x_i - \bar{x})^2} \sqrt{\sum_i^n (y_i - \bar{y})^2}} \quad (2)$$

where x_i, y_i are pared datasets indexed with *i* and $\bar{x}, \bar{y}$ are means. The sample size is *n*.

3 Research Design and Methodology

A. Two-factor Perturbations

Previous research projects usually used single-factor corruptions on images [13] [20] for benchmarking DL classifiers. For example, in [13], the authors created simple imperfect images through single-factor perturbations. However, images are impacted by two factors in the real world. For example, images captured by drones could be impacted by the drones' vibrations and environments such as fog or strong light. As another example, traffic signs could be faded, dirty, and/or rotated 5 or 10 degrees.

We use defective images with two-factor perturbation. Two-factor perturbation refers to modifying images using two types of perturbation, one after the other. The two-factor perturbation may include the two digital perturbations of salt & pepper noise and Gaussian noise in different sequences. Also, the two-factor perturbation may consist of a geometric perturbation (rotation) with different sequences.

For instance, SP-Gaussian perturbation means we firstly use salt & pepper noise for corrupting images, then use Gaussian noise for corrupting more. On the other hand, Gaussian-SP perturbation denotes changing images through Gaussian perturbation first, then applying salt & pepper perturbation. Examples of SP-Gaussian perturbation and Gaussian-SP perturbation are shown at the end in Figs. 8 and 10. Examples of SP-Rotation perturbation and Rotation-SP perturbation are shown at the end in Figs. 9 and 11.

The two-factor perturbation occurs in two stages. We use perturbation level and rotation ranges of [0.1, 0.2] and [-60, +60] degrees, respectively. Zero degrees denotes that we do not rotate images, negative degrees mean we rotate images counterclockwise, and positive degrees mean we rotate images clockwise. Specifically, two-factor perturbation types SP-Gaussian, Gaussian-SP, and SP-Rotation and their ranges will be applied as shown in Table 1.

Table 1. Perturbation Sequence, Perturbation Types, and Noise Ranges.

Perturbed Sequence	Respective Noise Strengths
SP-Gaussian	[0.1, 0.15, 0.2],[0.1, 0.15, 0.2]
Gaussian-SP	[0.1, 0.15, 0.2],[0.1, 0.15, 0.2]
SP-Rotation	[0.1, 0.15, 0.2],[-60^0, -30^0, 0^0, 30^0, 60]

Rotation-SP	[-60^0, -30^0, 0^0, 30^0,60^0],[0.1, 0.15, 0.2]

For measuring DL classifiers, we designed three steps: train stage, inference stage, and analyzing stage. For details, see Fig. 1. We used the Image Processing Toolbox of MATLAB 2019b for changing three image datasets: CIFAR10, CIFAR100, and MNIST. All source codes and related image sets are shared on websites (https://github.com/daiweiworking/RobustDeepLearningUsingPerturbations or http://cslinux.semo.edu/david/data.html) to support future academic research and industry projects.

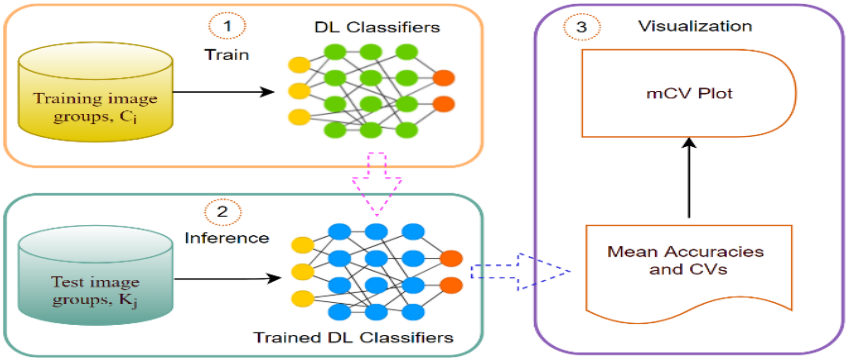

Fig. 1. An architecture for benchmarking DL classifiers. There are three stages when measuring the robustness of DL classifiers. Step 1 trains DL classifiers. Step 2 performs inference using the trained DL classifiers. Step 3 depicts the average accuracies and coefficients of variation (CV for short) through the mCV plot.

B. Evaluating robust deep learning classifiers using the mCV plot

For evaluating the robustness of DL classifiers, we start with the fundamental assumption that one DL classifier is more robust than another if it has higher accuracy and smaller coefficient of variation. The coefficient of variation is derived from the standard deviation (σ), which is a measure of spread of distributions in statistics. A larger σ denotes that the population of the sample data is further dispersed from the mean. A smaller σ indicates the population of sample data is more clumped together.

In statistics, the formula for standard deviation (σ) is shown in Eq. 3.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n}} \tag{3}$$

where x_i, n, and μ are a data value, the number of data, and the mean, respectively.

When testing performance across a group of data sets, we can collect the mean of the accuracies, μ, and the standard deviation of the accuracies, σ. We should compare the σ of two or more DL classifiers when their μ are the same or similar, because if two DL classifiers have the same mean accuracies, a small variation (σ) for a DL classifier shows better stability in its performance across image quality conditions than a large

variation. We can say that a smaller variation (σ) indicates that the DL classifier performs more conistently. In brief, we want a DL classifier to have high accuracy with small standard deviation.

The coefficient of variation (or CV) denotes the standard deviation as a percentage of the mean as shown in (4).

$$CV\ \% = 100 \times \frac{\sigma}{\mu} \tag{4}$$

where σ is the population standard deviation, and the mean, μ, is the average of the data sets.

Unlike accuracy or top-1/top-5 accuracies of DL classifiers, we developed the mCV statistical visualization for comparing the robustnesses of DL classifiers. Suppose that we wish to compare performances of DL classifiers. We propose a four-quadrant statistical plot approach which we name mCV (for mean accuracy and coefficient of variation) plot. In this statistical plot, the Y-axis and X-axis indicate mean accuracy and coefficient of variation, respectively. Then, the mean accuracy and CV of a reference point splits the figure into four groups, as shown in Figure 2.

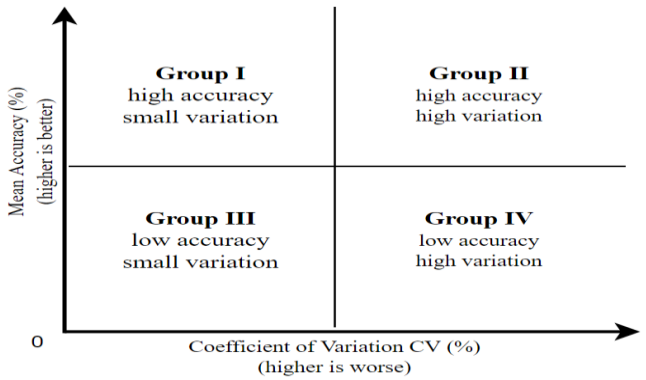

Fig. 2. An mCV plot. The Y-axis and X-axis indicate mean accuracy and coefficient of variation. The mean accuracy and CV of a reference point divides the above plot into four groups: Group I, Group II, Group III, and Group IV. The ideal DL classifier is in Group I.

In Figure 2, clockwise from top left, are Group I, Group II, Group III, and Group IV. Group I has high accuracy and low variation, Group II has high accuracy and high variation, Group III has low accuracy and low variation and Group IV has low accuracy and high variation.

How to split DL classifiers into four groups is shown in Algorithm 1.

Algorithm 1: Place a DL classifier into a group in the mCV graph.
Input 1: MA (deep learning **M**ean **A**ccuracy)
Input 2: CV (deep learning **C**oefficient of **V**ariation)
Output: group number
IdentifyGroup (MA, CV)

```
{
  rMA ← reference value for mean accuracy
  rCV ← reference value for coefficient of variation
  if (MA ≥ rMA and CV ≤ rCV) return "Group I"
  if (MA ≥ rMA and CV > rCV) return "Group II"
  if (MA < rMA and CV ≤ rCV) return "Group III"
  if (MA < rMA and CV > rCV) return "Group IV"
}
```

A trained DL classifier is notated as $\boldsymbol{D}_{trainsets}^{testsets}$, where D refers to the DL classifier's name, testsets refers to a group of test sets, and trainsets refers to a group of training sets. Then, we test the accuracy rate on every perturbation type *c*. Different *c* values are different perturbation types, and level of severity is *s*, with $s \in \{0.1, 0.15, 0.2\}$ when doing SP (salt and pepper) or GA (Gaussian) perturbation s, and $s \in \{-60^{o}, -30^{o}, 0^{o}, 30^{o}, 60^{o}\}$ when doing rotation perturbation s. The accuracy rate is written $Accu(\boldsymbol{D})$ as shown in Eq. 5.

$$Accu(\boldsymbol{D})\% = 100 \times \frac{\sum_{k=1}^{n} Accu(\boldsymbol{D}_{trainset_c}^{testset_k})}{n-1}, \qquad (5)$$

where both *c* and *k* mean a perturbation type, respectively. Each possible value of *k*, from type 1 to type 69, is specified in Table 4. Values for *c* were type 1 (clean images), type 5 (SP0.1), type 6 (SP0.1GA0.1), type 24 (GA0.1), type 25 (GA0.1SP0.1), type 43 (SP0.1RL30), type 34 (SP0.1RR30), type 40 (RL30) and type 32 (RR30). SP0.1GA0.1 means SP perturbation was applied at the 0.1 level followed by GA (Gaussian) perturbation at the 0.1 level, respectively.

For example, we can define $AlexNet_{clean}^{alltests}$ to denote the AlexNet classifier trained on a clean image set, and tested on all 69 different test image sets. $AlexNet_{SP0.1RL30}^{alltests}$ will denote the AlexNet classifier trained on an image set corrupted by SP0.1 then corrupted more by RL30 (rotate left 30^{o}), and the classifier is tested on all 69 test image sets.

$$CV(\boldsymbol{D})\% = 100 \times \frac{\sigma}{\mu} = 100 \times \frac{\sqrt{\frac{\sum_{k=1}^{69}(Accu(\boldsymbol{D})_k - \overline{Accu(\boldsymbol{D})})^2}{n}}}{\overline{Accu(\boldsymbol{D})}} \qquad (6)$$

where $CV(\boldsymbol{D})$ indicates the coefficient of variation exhibited by a DL classifier. $\overline{Accu(\boldsymbol{D})}$ denotes the mean accuracy of the DL classifier. $Accu(\boldsymbol{D})_k$ means the accuracy of DL classifier on test set *k*. Variable *k*, $k \in [1, 69]$, is the testing image group (for more information, see attached Table 4).

The training data sets consists of nine training sets: a clean image set and eight corrupted image sets. Each training group has 500 images. The clean image set contains the original images. The eight corrupted image sets are corrupted by SP0.1, GA0.1, SP0.1GA0.1, GA0.1SP0.1, SP0.1RL30, SP0.1RR30, RL30 and RR30.

Each testing set contains one clean image group and 68 corrupted image groups. The clean image group contains original images. Each testing group has 500 images. The training sets and testing sets have no overlap. In other words, if an image exists in a training group, the image will not be found in any testing groups.

4 Experimental Results

A. Benchmarking three DL classifiers on clean images

In this experiment, the average of the CVs of the AlexNet, ResNet50, and VGG19 classifiers and the average of their mean accuracies defined the reference point.

Because abbreviated notations might not be easily read in Figure 3, we wrote labels instead. For example, the label "AlexNet(clean)" denotes the AlexNet network trained on the unmodified CIFAR-10 image set. In Table 1, the training data sets of CIFAR-10 are different for each row, but the testing was on all 69 data sets including the clean CIFAR-10 image set and 68 corrupted versions, which we may denote as $AlexNet\ {}^{alltests}_{clean}$.

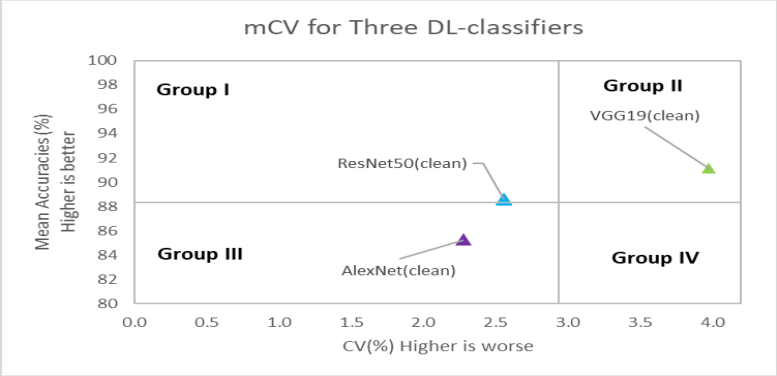

Fig. 3. The mCV plot for three DL classifiers on the CFIAR-10 dataset and tested on all 69 test sets. The reference value, (CV, Accuracy)=(2.94%, 88.31%), is the average of ResNet(clean), AlexNet(clean), and VGG(clean).

As shown in Figure 3, the mean accuracy of the ResNet50 classifier is better than AlexNet, but the CV of the ResNet-50 is higher (thus worse) than for AlexNet. The VGG19 has the worst CV and the best mean accuracy of those DL classifiers. Thus the mCV plot visualizes a comparison of different DL classifiers on the same test protocol.

B. Benchmarking the AlexNet classifier on corrupted images

AlexNet(clean) represents the AlexNet classifier trained on clean data. The purple points represent the AlexNet classifiers trained on corrupted image sets. From Figure 4, we can see the following.

- The training datasets affected prediction CVs of the AlexNet classifiers. Notably, after training AlexNet on corrupted images, the CVs are less than when training

is limited to clean images.

- Training AlexNet on corrupted images can improve the mean accuracy during testing. Specifically, most Accu($AlexNet_{corrupt}^{alltests}$) values are higher than Accu($AlexNet_{clean}^{alltests}$). That is, tested accuracies tend to be better when training sets contain images that have been corrupted by degradation with common forms of noise and distortion.

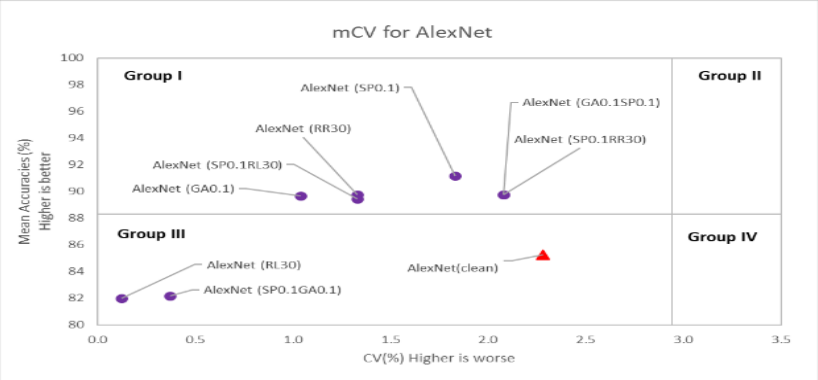

Fig. 4. The mCV plot for AlexNet after corrupted image groups were used to training the DL classifier on the CIFAR-10 dataset.

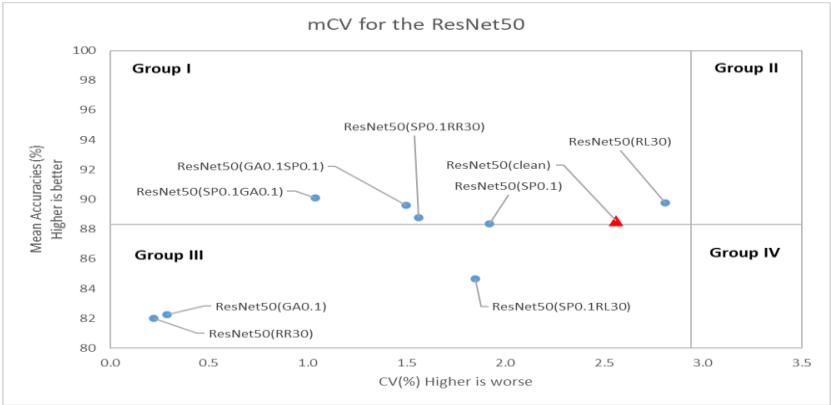

Fig. 5. The mCV plot for the ResNet50 DL classifier after corrupted image sets were used to train it on the CIFAR-10 dataset. The red triangle represents Accu$\left(ResNet50_{clean}^{alltests}\right)$, and blue points represent training on different corrupted image sets, i.e., examples of Accu$\left(ResNet50_{corrupt}^{alltests}\right)$.

C. Benchmarking the ResNet50 Classifier on corrupted images

In Figure 5, the red triangle shows the ResNet50 classifier trained on clean data. The blue points are from the ResNet50 classifiers trained on different corrupted image sets. From Figure 5, we see the following.

- The training datasets affected prediction CVs of the ResNet50 classifiers. After training ResNet50 classifiers on corrupted images, the highest CVs

(with the exception of Accu($Resnet50^{alltests}_{RL30}$) are less than for the DL classifier trained on clean images.

- Training ResNet50 on corrupted images can improve the mean accuracy during testing. Accu($Resnet50^{alltests}_{RL30}$), Accu($Resnet50^{alltests}_{GA0.1SP0.1}$), and Accu($Resnet50^{alltests}_{SP0.1R30}$), and Accu($Resnet50^{alltests}_{SP0.1GA0.1}$) are greater than Accu($Resnet50^{alltests}_{clean}$).

D. Benchmarking the VGG-19 Classifier on corrupted images

As shown in Figure 6, the red triangle represents the VGG19 classifier trained on clean data. The green points are for the ResNet50 classifiers trained on corrupted image sets. From Figure 6, CV($VGG19^{alltests}_{clean}$) is greater than any CV($VGG19^{alltests}_{corrupt}$). This illustrates the effect of training on corrupted images to boost DL robustness.

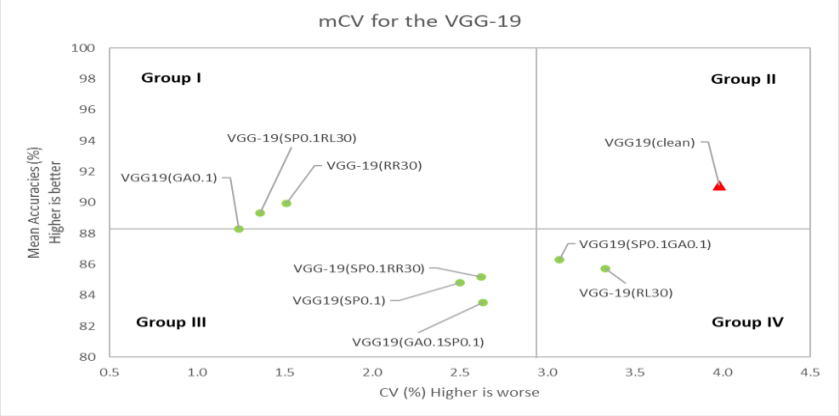

Fig. 6. The mCV plot for VGG19 after corrupted image groups trained the DL classifier on the CIFAR-10 dataset. The red triangle is Accu($VGG19^{alltests}_{clean}$), and green points are Accu($VGG19^{alltests}_{corrupt}$).

5 Discussion

Our tests demonstrate that high accuracy of DL classifiers on high quality image sets does not ensure high stability of DL classifiers when used on images of lowered quality, which are commonly encountered in real world applications. From the merged performance data of three DL classifiers in Table 1, we conclude as follows.

- Most CV($\boldsymbol{D}^{alltests}_{corrupt}$) values are less than the corresponding CV($\boldsymbol{D}^{alltests}_{clean}$) values. In other words, corrupted images tend to improve the robustness of DL classifiers.
- In different DL classifiers, the same corrupted image training sets may reduce or improve the accuracies of DL classifiers. For the VGG19 classifier, Accu($VGG19\ ^{alltests}_{SP0.1RR30}$) is less than Accu($VGG19\ ^{alltests}_{clean}$), which means that training on corrupted image groups reduced the accuracy of VGG19. However, the

ResNet50 and AlexNet classifiers show different results. For example, Accu($Alexnet\ ^{alltests}_{SP0.1RR30}$) is greater than Accu($Alexnet\ ^{alltests}_{clean}$).

- When training sets can be either the clean image set or a corrupted image set, Accu($\boldsymbol{D}^{clean}_{clean}$) is greater than Accu($\boldsymbol{D}^{clean}_{corrupt}$) for most types of perturbation within training sets.

Table 2. Spearman's Rank & Pearson Correlation Coefficient.

Data Sets	Spearman's Rank	Pearson Correlation Coefficient
CV & Mean Accuracy	0.202	0.096
CV & Accuracy (clean)	0.345	0.365
Mean Accuracy & Accuracy (clean)	0.057	0.052

To measure the correlation coefficient, mean accuracy, and accuracy (clean), Spearman's rank correlation coefficient and Pearson correlation coefficient were used, as shown in Table 2.

From Table 2, the values of Pearson correlation coefficients are all less than 0.5. This means that the correlation coefficient values between CV, mean accuracy, and accuracy(clean) are not significant. In other words, these three variables are at least somewhat independent.

A robust DL classifier [16] [13] should produce results with high accuracies and small CVs. While accuracy and CV are, individually, one-dimensional metrics, we integrated them into a two-dimensional metric by designing and using the mCV plot for understanding DL classifier robustness. Thus we show how to use the mean accuracy and CV together to visualize robust DL classifiers on a statistical plot. Another advantage of the mCV plot is that it provides a visualization tool for measuring robust DL classifiers with a low learning curve. This complements the standard tools used in related work like tables, line diagrams, and bar charts. This is the first study that focuses on using two-factor perturbations on images for benchmarking DL classifiers, an objective supported by the mCV plot graphic.

Third, we created 69 benchmarking image sets, including a clean set, sets with single factor perturbation s, and sets with two-factor perturbation conditions. The datasets and source codes are publicly available as a resource for future investigations.

Finally, previous researchers have indicated that corrupted image sets reduce the accuracy of classifiers [12] or trigger underfit [16]. We have found that this appears to be a result of using only high quality image sets for training/testing. Our results indicate that it is advantageous to include corrupted image sets during training. However, our results were partially consistent with [16] and [12]. When testing on clean images, our results were consistent with previous researchers' reports. Nevertheless, when choosing 69 testing image sets, our results confirmed that DL classifiers can enhance their robustness by training these classifiers on defective images degraded by two-factor perturbation s.

There may be some potential limitations in this research. The first is the datasets. the CIFAR-10 image sets are small datasets that do not present all images. Also, the CIFAR-10 data does not have any grayscale pictures. The second limitation concerns the 68 corrupted image sets that were degraded by SP perturbation s, GA perturbation

s, and rotations. These corrupted image sets do not constitute all types of imperfections that may distort images.

6 Conclusion

In this research, we provided a two-dimensional metric that integrates accuracy and coefficient of variation and, to display it, the mCV plot. This approach is useful for benchmarking robustness of DL classifiers. We recommended a view of robustness based on training and testing using image sets that have been deliberately corrupted to model the imperfections of images inherent in many real world application problems. Based on the foregoing, our studies serve as a proof-of-concept that mean accuracies and CVs can be used for quantitatively benchmarking robustness in the performance of DL classifiers.

In this study, we tested three DL classifiers on images degraded by two-factor perturbation. The research demonstrated that using two-factor corrupted images can increase the dependability of DL classifiers and, more generally, that imperfect images should certainly be considered in training, testing and benchmarking DL classifiers.

7 Acknowledgement

The authors are grateful to Google for awarding $5,000 in cloud credits for this research.

8 Appendix

Table 3. Comparing accuracy of DL classifiers after training on clean vs. corrupted data. Training on original, uncorrupted data is denoted as "(clean)". "(SP0.1GA01)" indicates training on an image set corrupted first by salt-and-pepper noise at the 0.1 level followed by Gaussian noise at the 0.1 level. Testing is on the 69 test sets

DL Classifier (training set)	CV (%)	Mean Accuracy (%)	Accuracy (clean) (%)
AlexNet (clean)	2.28	85.25	92.08
AlexNet (SP0.1GA0.1)	0.37	82.17	91.30
AlexNet (GA0.1)	1.04	89.67	90.90
AlexNet (GA0.1SP0.1)	2.08	89.77	90.82
AlexNet (SP0.1)	1.83	91.18	90.78
AlexNet (RR30)	1.33	89.75	90.74
AlexNet (RL30)	0.12	81.97	90.72
AlexNet (SP0.1RR30)	2.08	89.77	90.52
AlexNet (SP0.1RL30)	1.33	89.44	89.96
ResNet50 (clean)	2.56	88.56	91.18

ResNet50 (SP0.1RL30)	1.85	84.66	84.16
ResNet50 (RL30)	2.81	89.78	83.80
ResNet50 (RR30)	0.22	81.99	83.68
ResNet50 (GA0.1SP0.1)	1.50	89.60	82.56
ResNet50 (SP0.1)	1.92	88.36	82.40
ResNet50 (GA0.1)	0.29	82.27	82.40
ResNet50 (SP0.1RR30)	1.56	88.79	82.34
ResNet50 (SP0.1GA0.1)	1.04	90.08	81.90
VGG19 (clean)	3.98	91.13	94.92
VGG19 (GA0.1SP0.1)	2.63	83.54	92.82
VGG19 (GA0.1)	1.24	88.30	92.48
VGG19 (SP0.1GA0.1)	3.07	86.33	92.16
VGG19 (SP0.1)	2.50	84.83	92.16
VGG-19 (SP0.1RL30)	1.36	89.34	90.48
VGG-19 (RL30)	3.33	85.75	90.32
VGG-19 (RR30)	1.51	89.94	90.04
VGG-19 (SP0.1RR30)	2.62	85.20	88.72

Table 4. Perturbation types. Type 1 or "CLEAN" represents clean images group or original images group, whereas Type 8 or "(SP0.1GA0.2)" indicates an image set corrupted first by salt-and-pepper noise at a level of 0.1 followed by Gaussian noise at the 0.2 level.

1	2	3	4	5	6	7	8	9
CLEAN	SP0GA0.1	SP0GA0.15	SP0GA0.2	SP0.1GA0	SP0.1GA0.1	SP0.1GA0.15	SP0.1GA0.2	SP0.15GA0
10	11	12	13	14	15	16	17	18
SP0.15GA0.1	SP0.15GA0.15	SP0.15GA0.2	SP0.2GA0	SP0.2GA0.1	SP0.2GA0.15	SP0.2GA0.2	GA0SP0.1	GA0SP0.15
19	20	21	22	23	24	25	26	27
GA0SP0.2	GA0.15SP0	GA0.15SP0.1	GA0.15SP0.15	GA0.15SP0.2	GA0.1SP0	GA0.1SP0.1	GA0.1SP0.15	GA0.1SP0.2
28	29	30	31	32	33	34	35	36
GA0.2SP0	GA0.2SP0.1	GA0.2SP0.15	GA0.2SP0.2	SP0RR30	SP0RR60	SP0.1RR30	SP0.1RR60	SP0.15RR30
37	38	39	40	41	42	43	44	45
SP0.15RR60	SP0.2RR30	SP0.2RR60	SP0RL30	SP0RL60	SP0.1RO0	SP0.1RL30	SP0.1RL60	SP0.15RO0
46	47	48	49	50	51	52	53	54
SP0.15RL30	SP0.15RL60	SP0.2RO0	SP0.2RL30	SP0.2RL60	RR30SP0.1	RR30SP0.15	RR30SP0.2	RR30SP0

55	56	57	58	59	60	61	62	63
RR60SP0.1	RR60SP0.15	RR60SP0.2	RR60SP0	RO0SP0.1	RO0SP0.15	RO0SP0.2	RL30SP0	RL30SP0.1
64	65	66	67	68	69			
RL30SP0.15	RL30SP0.2	RL60SP0	RL60SP0.1	RL60SP0.15	RL60SP0.2			

Fig. 7. Results of SP-Gaussian perturbation . Note: the original picture is from CIFAR-10 as shown on the top left corner. (SP, GA) means Salt & Pepper noise perturbation was applied first, and then Gaussian noise was applied. (SP, GA) = (0, 0.1) indicates Salt & Pepper noise density is zero, but Gaussian noise density is 0.10.

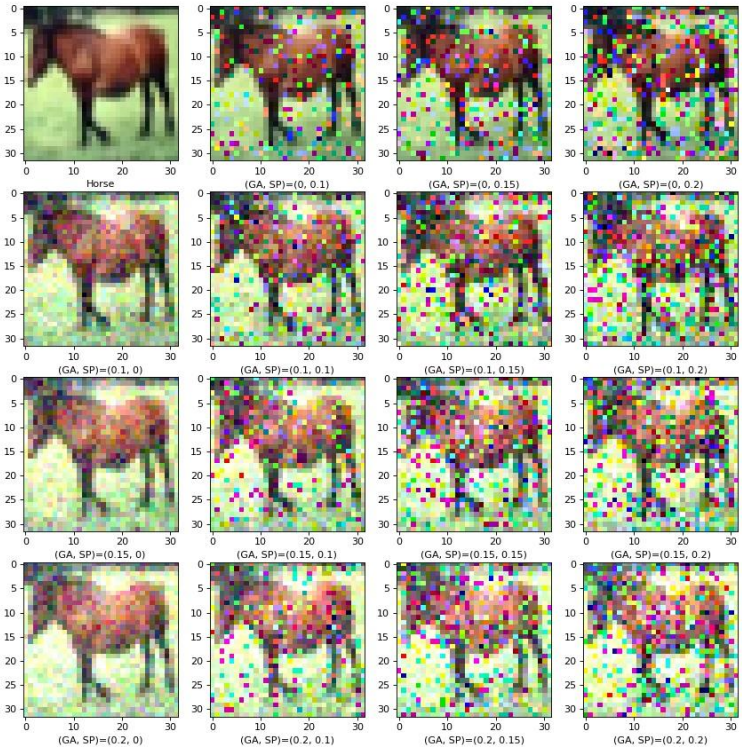

Fig. 8. Results of Gaussian-SP perturbation . Note: the original picture is located in the top left corner. (GA, SP) means Gaussian noise perturbation was applied first, and then Salt & Pepper noise was applied. (GA, SP) = (0.1, 0.15) indicates Gaussian noise density is 0.15, but Salt & Pepper noise density is 0.15.

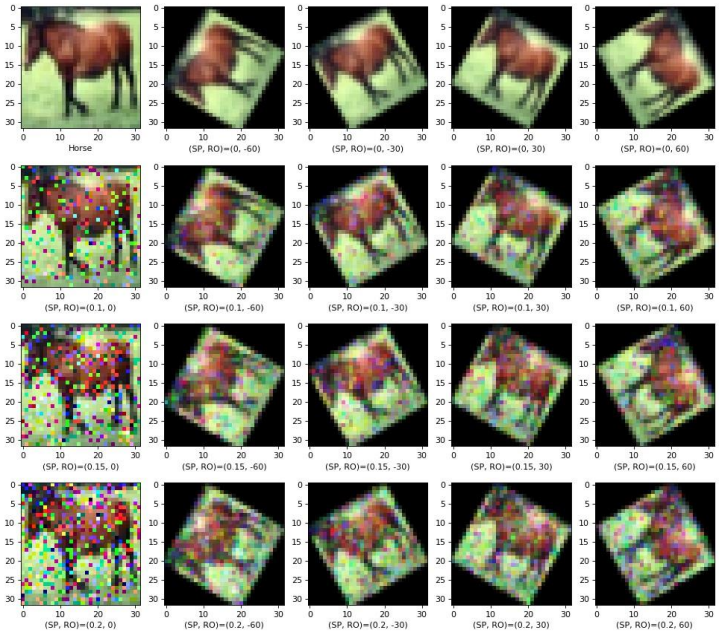

Fig. 9. Results of SP-Rotation perturbation . Note: the original picture locates at the top left corner. (SP, RO) means Salt & Pepper noise was corrupted first, and then rotation images. (SP, RO) =(0.1, -60) indicates Salt & Pepper noise density is 0.1, and the image counterclockwise rotation was 60 degrees. (SP, RO) = (0.2, 60) indicates Salt & Pepper noise density is 0.2, and the image clockwise rotate 60 degrees.

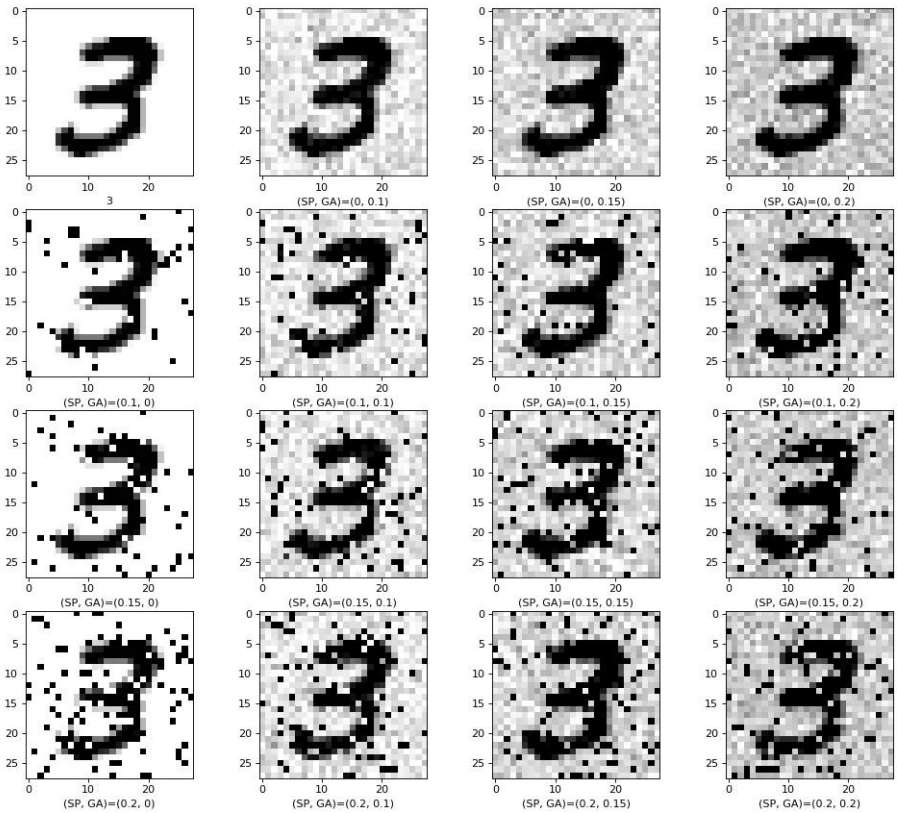

Fig. 10. Results of SP-Gaussian perturbation s. The original picture is at the top left corner. (SP, GA) means Salt & Pepper noise was applied first to corrupt images, and then Gaussian noise corrupted the images.

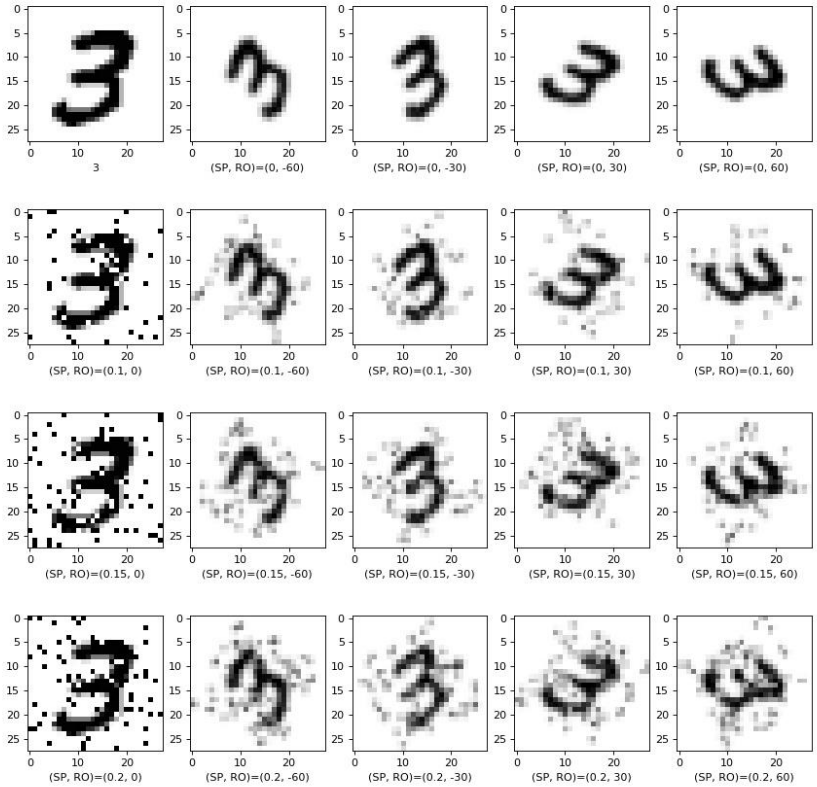

Fig. 11. Results of SP-Rotation perturbation . The original picture is at the top left corner. (SP, RO) means Sale &Pepper noise first was used to corrupt images, and then rotation. (SP, RO) =(0.1, -60) indicates Salt & Pepper noise density is 0.1, and the image counterclockwise rotation was 60 degrees. (SP, RO) = (0.2, 60) indicates Salt & Pepper noise density is 0.2, and the image clockwise rotate 60 degrees.

References

1. Standard Performance Evaluation Corporation, SPEC, http://www.spec.org, last accessed 2021/7/23.
2. Transaction Processing Performance Evaluation Corporation, TPC, http://www.tpc.org, last accessed 2021/7/23.
3. Storage Performance Corporation, SPC, http://www.spcresults.org, last accessed

2021/7/23.
4. Dai, W., Berleant, D.: Benchmarking contemporary deep learning hardware and frameworks: a survey of qualitative metrics. In: 2019 IEEE First Int. Conf. Cogn. Mach. Intell., pp. 148–155 (Dec. 2019).
5. Gershgorn, D.: The data that transformed AI research—and possibly the world. Quartz (July 26, 2017).
6. Geirhos, R., Janssen, D. H. J., Schütt, H. H., Rauber, J., Bethge, M., Wichmann, F. A.: Comparing deep neural networks against humans: object recognition when the signal gets weaker. arXiv 1706.06969 [cs.CV] (2017).
7. Dodge, S., Karam, L.: A study and comparison of human and deep learning recognition performance under visual distortions. In: 26th International Conference on Computer Communications and Networks, ICCCN (2017).
8. Carlini, N., Wagner, D.: Adversarial examples are not easily detected: bypassing ten detection methods. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security pp. 3–14 (2017)
9. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial machine learning at scale. arXiv 1611.01236 [cs.CV] (2016).
10. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv 1706.06083 [stat.ML] (2019].
11. Hosseini, H., Xiao, B., Poovendran, P.: Google's cloud vision API is not robust to noise. In: Proceedings – 16th IEEE International Conference on Machine Learning and Applications (ICMLA) (2017).
12. Dodge, S., Karam, L.: Understanding how image quality affects deep neural networks. In: 2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX), pp. 1–6 (2016).
13. Hendrycks, D., Dietterich, T. G.: Benchmarking Neural network robustness to common corruptions and surface variations. arXiv 1807.01697 [cs.LG] (2019).
14. Dodge, S., Karam, L.: Quality resilient deep neural networks. arXiv 1703.08119 [cs.CV] (2017).
15. Dai, W.: Benchmarking deep learning robustness on images with two-factor corruption. Dissertation, Dept. of Information Science, University of Arkansas at Little Rock (2020).
16. Zheng, S., Song, Y, Leung, T., Goodfellow, I.: Improving the robustness of deep neural networks via stability training. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2016).
17. Cohen, Y., Cohen, J. Y.: Statistics and data with R: an applied approach through examples (2008).
18. Reed, G. F., Lynn, F., Meade, B. D., Use of coefficient of variation in assessing variability of quantitative assays. Clin. Diagn. Lab. Immunol. 9(6), 1235–9 (2002).
19. Francis, T. R., Kannenberg, L. W.: Yield stability studies in short-season maize. I. A descriptive method for grouping genotypes. Can. J. Plant Sci. 58(4), Oct. (1978).
20. Vasiljevic, I., Chakrabarti, A., Shakhnarovich, G.: Examining the impact of blur on recognition by convolutional networks. arXiv 1611.05760 [cs.CV] (2016).