Rotation 1: Report

Konstantin, Kueffner

Abstract The $KS(conf)$ procedure, introduced by (Sun and Lampert 2019), is designed to detect whether a convolutional neural network operates in an out-of-specification environment. This is accomplished by estimating the distribution of confidence scores corresponding to a within-specification environment, and comparing it with the distribution of confidence scores observed during operation using the Kolmogorov–Smirnov goodness-of-fit test. This text presents two adaptations of this procedure. The first, $KS(\tau\text{-}conf)$, generalises $KS(conf)$ by introducing the function τ, making the method of confidence score aggregation variable. The second, $KS(T\text{-}conf)$, is best understood as a procedure utilising multiple $KS(\tau\text{-}conf)$ procedures by combining their verdicts. Extensive experimentation on three neural network architectures suggest that some instances of τ and T improve upon $KS(conf)$.

1 Introduction

The application of deep convolutional neural networks (cNet) in computer vision has resulted in the emergence of systems that can operate autonomously over long periods of time, while retaining a high level of accuracy. This remarkable performance, however, relies on the assumption that the distribution generating the data observed during operation is representative of the distribution from which the trainings data was sampled. Yet, given enough time it may be possible that this assumption fails to be satisfied, i.e. the operational data distribution shifts. Hence, nullifying any established performance guarantees and thus discouraging their deployment in safety-critical applications. This, however, provides ample motivation for the development of procedures that can reliably detect said shifts in the underlying distribution (Sun and Lampert 2019).

One such procedure is the $KS(conf)$ procedure. Leveraging the Kolmogorov–Smirnov goodness-of-fit test (KS-test), compares the distribution of confidence scores observed during operation against an estimate of the within-spec distribution of confidence scores constructed during the calibration phase of this procedure. Two possible avenues for improvement are discussed here. Firstly, while cNet produce a vector of confidence scores, $KS(conf)$ utilises only its maximum entry, thus selecting more inclusive methods of aggregation may lead to improvements in performance, leading to the introduction of $KS(\tau\text{-}conf)$. Secondly, considering that different aggregation methods emphasise different aspects of the confidence score vector, the conjecture arises that some aggregation functions may complement each other. Therefore, it may be possible to obtain a more robust procedure for out-of-spec detection, by combining several $KS(\tau\text{-}conf)$ procedures into a single one, called $KS(T\text{-}conf)$. Robust in this case refers to sustained high performance over various out-of-spec distributions and cNet architectures.

2 Preliminaries

This section introduced notational convention used through the remainder of this text. Starting with the notion of a "vectorised" function, i.e. given some function $f : A \to B$, let $f(\mathbf{a}) := (f(a_1), \ldots, f(a_n))$ for $\mathbf{a} \in A^n$. In addition, for some set X let $\text{sample}_n(X)$ be some randomly sampled subset of X with size $n \in \mathbb{N}$. Moreover, c will indicate some trained multi-class classifier mapping from $\mathcal{X}$ to $\mathcal{Z} \subset [0,1]^K$. $\mathcal{Z}$ being the set of vectors representing confidence scores sorted in descending order, i.e. $(z_1, \ldots, z_k) \in \mathcal{Z}$ satisfies $\sum_{i=1}^{K} z_i = 1$ and $z_i \geq z_{i+1}$, where $K := |\mathcal{Y}|$ with $\mathcal{Y}$ being the set of classes on which c was trained. By vectorisation c can be applied to multiple inputs, i.e. $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathcal{X}^n$, $c(\mathbf{X}) = (c(\mathbf{x}_1), \ldots, c(\mathbf{x}_n)) = (\mathbf{z}_1, \ldots, \mathbf{z}_n) = \mathbf{Z}$. Furthermore, let $\tau : \mathbb{R}^K \to \mathbb{R}$ be some aggregation function, s.t. for all $\mathbf{z} \in \mathcal{Z}$ $\tau(\mathbf{z}) = s^\tau \in [0,1]$ and that $\exists \mathbf{z} \in \mathcal{Z}$ $\tau(\mathbf{z}) = 1$. Again, $\mathbf{s}^\tau = \tau(\mathbf{Z}) = (s_1^\tau, \ldots, s_n^\tau) := (\tau(\mathbf{z}_1), \ldots, \tau(\mathbf{z}_n))$. For simplicity, assume that $\mathbf{s}^\tau$ is already sorted in ascending order, i.e. $s_i \leq s_{i+1}$. A set of such aggregation functions, will be referred to as $T := (\tau_1, \ldots, \tau_H)$. This text distinguishes between two kinds of input sets. Namely, $\mathbf{X}^{\text{cal}} := (\mathbf{x}_1^{\text{cal}}, \ldots, \mathbf{x}_N^{\text{cal}})$ and $\mathbf{X}^{\text{op}} := (\mathbf{x}_1^{\text{op}}, \ldots, \mathbf{x}_M^{\text{op}})$, with $\mathbf{X}^{\text{cal}}$ representing a sample of the in-specification distribution, while $\mathbf{X}^{\text{op}}$ contains inputs encountered during operation. Subsequently, M will be referred to as batch size. Moreover, for $\mathbf{X}^{\text{cal}}$, the vector $\mathbf{s}^{\text{cal},\tau}$ is adjusted such that each entry is unique, i.e. minuscule amounts of noise is added to ensure this condition (Sun and Lampert 2019).

All procedures introduced below use a restricted form of linear interpolation, as well as the Kolmogorov–Smirnov goodness-of-fit test. Staring with the prior.

Definition 1. *Consider $\mathbf{v} \in [0,1]^n$ and some $\mathbf{w} \in \mathbb{R}^n$ such that $\mathbf{v}$ contains no duplicate entries and that it is already sorted in ascending order, i.e. $v_i < v_{i+1}$. For some $p \in [0,1]$, $F_{\mathbf{v},\mathbf{w}}(p)$ is computed by finding $v_i \leq p \leq v_{i+1}$ and using the piecewise linear interpolation function*

$$F_{\mathbf{v},\mathbf{w}}(p) := w_i + (w_{i+1} - w_i)\,\frac{p - v_i}{v_{i+1} - v_i}$$

Furthermore, $F_{\mathbf{v}}(p) := F_{\mathbf{v}',\mathbf{w}'}(p)$ with $\mathbf{v}' := (0, v_1, \ldots, v_n, 1)$ and $\mathbf{w}' := (\frac{i}{n+1})_{i \in \{0,\ldots,n+1\}}$.

Moving on to the one-sample Kolmogorov–Smirnov goodness-of-fit test against the cdf of a uniform reference distribution. This test requires the computation of the KS-statistic.

Definition 2 (Sun and Lampert 2019)**.** *For some vector $\mathbf{v} \in [0,1]^n$ s.t. $v_i \leq v_{i+1}$, let*

$$KS(\mathbf{v}) := \max\left(\max_{i=1,\ldots,n}\left\{v_i - \frac{i-1}{n}\right\}, \max_{i=1,\ldots,n}\left\{\frac{i}{n} - v_i\right\}\right)$$

This measure captures the largest absolute difference between the empirical cumulative distribution function (ecdf) of the sample and the uniform reference cdf. Observe that for almost identical distribution KS-statistic will be close to 0. Using this measure one can formulate the KS-test.

Definition 3 (Sun and Lampert 2019)**.** *For some vector $\mathbf{v} \in [0,1]^n$ s.t. $v_i \leq v_{i+1}$ and for some $\alpha \in [0,1]$. The KS-test rejects the null hypothesis, i.e. the* ecdf *of $\mathbf{v}$ differs from the* cdf *of $U(0,1)$, with a significance of α if $KS(\mathbf{v}) > \Theta_{\alpha,n}$, with $\theta_{\alpha,n}$ being computed numerically (Marsaglia, Tsang, Wang, et al. 2003)*

For the procedures to come, α is best understood as providing a bound on the false-positive rate (Sun and Lampert 2019). Furthermore, it must be reiterated that this variant of the KS-test already assumes a uniform reference distribution, for more on the actual KS-test please consult (Massey Jr 1951).

3 KS(τ-conf)

KS(τ-*conf*) generalises *KS*(*conf*) by introducing the aggregation function τ, resulting in the following procedure.

Similar to *KS*(*conf*), *KS*(τ-*conf*) has a calibration phase, during which the cdf of the aggregated confidence scores, given input $\mathbf{X}^{\text{cal}}$, is estimated. This is accomplished as follows. First, the classifier c is used to extract the confidence scores $\mathbf{Z}^{\text{cal}}$ from $\mathbf{X}^{\text{cal}}$. Second, the extracted confidence scores are aggregated into $\mathbf{s}^{\text{cal},\tau}$ using the aggregation function τ. Third, using $\mathbf{s}^{\text{cal},\tau}$, it is possible to estimate the cdf by means of piecewise linear interpolation, finally obtaining $F_{\mathbf{s}^{\text{cal},\tau}}$, which is used to construct the corresponding test $\mathcal{A}_\tau^{c,\mathbf{X}^{\text{cal}}}$.

The batch testing phase, follows at first a similar structure as the calibration phase. Namely, given $\mathbf{X}^{\text{op}}$ and some $\alpha \in [0,1]$ the procedure uses c to obtain $\mathbf{Z}^{\text{op}}$, which is aggregated by τ resulting in $\mathbf{s}^{\text{op},\tau}$. Now differing from the calibration phase, one applies the cdf obtained during calibration, $F_{\mathbf{s}^{\text{cal},\tau}}$ to $\mathbf{s}^{\text{op},\tau}$. Since $F_{\mathbf{s}^{\text{cal},\tau}}(\mathbf{s}^{\text{cal},\tau})$ approximates the cdf of the uniform distribution $U(0,1)$, the *KS*-test introduced in Definition 3 can be applied w.r.t. to $\Theta_{\alpha,M}$. This, however, is only possible because the *KS*-test is in fact invariant under the reparameterisation of the sample space (Sun and Lampert 2019).

Definition 4. *For some classifier c, some calibration data $\mathbf{X}^{\text{cal}}$, some operation data $\mathbf{X}^{\text{op}}$ and for some aggregation function τ, KS(τ-conf) is defined as a two phase process using the calibration function* cal_τ *to construct the test $\mathcal{A}_\tau^{c,\mathbf{X}^{\text{cal}}}$, i.e. $\text{cal}_\tau(\mathbf{X}^{\text{cal}}, c) = \mathcal{A}_\tau^{c,\mathbf{X}^{\text{cal}}}$. Moreover, $\mathcal{A}_\tau^{c,\mathbf{X}^{\text{cal}}}$ is defined as $\mathcal{A}_\tau^{c,\mathbf{X}^{\text{cal}}}(\mathbf{X}^{\text{op}}, \alpha) := KS\Big(F_\tau^{c,\mathbf{X}^{\text{cal}}}\big(\tau(c(\mathbf{X}^{\text{op}}))\big)\Big)$, where $F_\tau^{c,\mathbf{X}^{\text{cal}}} := F_{\tau(c(\mathbf{X}^{\text{cal}}))}$ is established during the calibration phase.*

If clear from the context or irrelevant, both c and $\mathbf{X}^{\text{cal}}$ will be suppressed.

By varying τ new tests can be instantiated. Section 5 investigates the following additional variants:

- *KS*(max-*conf*), where τ computes the maximum confidence score, i.e. for some $\mathbf{z} \in \mathcal{Z}$ $\tau(\mathbf{z}) = \max(\mathbf{z}) := z_1$;
- *KS*(margin-*conf*), where τ is the difference between the highest and second highest confidence score, i.e. for some $\mathbf{z} \in \mathcal{Z}$ $\tau(\mathbf{z}) = \text{margin}(\mathbf{z}) := z_1 - z_2$;
- *KS*(L^2-*conf*), where τ is the L^2-norm over the confidence scores, i.e. for some $\mathbf{z} \in \mathcal{Z}$ $\tau(\mathbf{z}) = \|\mathbf{z}\| := \sqrt{\sum_{i=1}^K z_i^2}$;
- *KS*(GM-*conf*),where τ is the normalised geometric mean over the confidence scores, i.e. for some $\mathbf{z} \in \mathcal{Z}$ $\tau(\mathbf{z}) = K \cdot \text{GM}(\mathbf{z}) := K \cdot \left(\prod_{i=1}^K z_i\right)^{\frac{1}{K}}$;
- *KS*(*H*-*conf*), where τ is the normalised entropy function (Shannon 1948) over the confidence scores, i.e. for some $\mathbf{z} \in \mathcal{Z}$ $\tau(\mathbf{z}) = \frac{H(\mathbf{z})}{\log(K)} := \frac{-\sum_{i=1}^K z_i \log(z_i)}{\log(K)}$.

4 KS(T-conf)

KS(*T*-*conf*) aggregates the decisions of multiple *KS*(τ-*conf*) procedures together into a single verdict. In the construction of such a method there are three major design decision.

Firstly, the method for aggregating the $KS(\tau\text{-}conf)$ decisions, which can be accomplished using various voting mechanisms. Secondly, the adjustment of the fpr, which is necessary due to the dependence between the employed hypothesis tests. Thirdly, the selection of $KS(\tau\text{-}conf)$ procedures. Therefore, this procedure is relative to some voting mechanism ν, some correction function g and some set of aggregation function $T := \{\tau_1, \dots \tau_H\}$.

The calibration phase of $KS(T\text{-}conf)$ has two objectives. The first one, is to calibrate all $KS(\tau\text{-}conf)$ sub-procedures. That is, for each $\tau_i \in T$ the procedure $KS(\tau_i\text{-}conf)$ has to be calibrated, resulting in $\big(\text{cal}_{\tau_1}(\mathbf{X}^{\text{cal}}, c), \dots, \text{cal}_{\tau_H}(\mathbf{X}^{\text{cal}}, c)\big) = \big(\mathcal{A}_{\tau_1}^{c,\mathbf{X}^{\text{cal}}}, \dots, \mathcal{A}_{\tau_H}^{c,\mathbf{X}^{\text{cal}}}\big)$. The second objective is to calibrate g such that it can properly adjust α to prevent the observed fpr from exceeding the set fpr. This, however, is obviously dependent on the specific correction function.

The batch testing phase, is constructed as follows. First, the parameter α is passed through g to obtain the corrected set fpr α'. Then both the input data $\mathbf{X}^{\text{op}}$ and α' are used to construct solution vector $\mathbf{r} \in \{0,1\}^H$ by running it through each configured sub-procedure, i.e. $\mathbf{r} = \big(\mathcal{A}_{\tau_1}^{c,\mathbf{X}^{\text{cal}}}(\mathbf{X}^{\text{op}}, \alpha'), \dots, \mathcal{A}_{\tau_H}^{c,\mathbf{X}^{\text{cal}}}(\mathbf{X}^{\text{op}}, \alpha')\big)$. Finally, this vector is aggregated using some voting mechanism to obtain a final verdict, i.e. $\nu(\mathbf{r}) \in \{0,1\}$.

Definition 5. *For some classifier c, some calibration data $\mathbf{X}^{\text{cal}}$, some operation data $\mathbf{X}^{\text{op}}$, for some aggregation function τ, some correction function g and some voting mechanism ν, $KS(\tau\text{-}conf)$ is defined as a two phase process, using cal_T to construct $\mathcal{A}_\tau^{c,g,\nu,\mathbf{X}^{\text{cal}}}$, i.e. $\text{cal}_T(\mathbf{X}^{\text{cal}}, c, \nu) := \mathcal{A}_\tau^{c,g,\nu,\mathbf{X}^{\text{cal}}}$. Moreover, $\mathcal{A}_\tau^{c,g,\nu,\mathbf{X}^{\text{cal}}}$ is defined as*

$$\mathcal{A}_T^{c,g,\nu,\mathbf{X}^{\text{cal}}}(\mathbf{X}^{\text{op}}, \alpha) := \nu\Big(\mathcal{A}_{\tau_1}^{c,\mathbf{X}^{\text{cal}}}\big(\mathbf{X}^{\text{op}}, g^{\text{cal}}(\mathbf{X}^{\text{op}}, \alpha)\big), \dots, \mathcal{A}_{\tau_H}^{c,\mathbf{X}^{\text{cal}}}\big(\mathbf{X}^{\text{op}}, g^{\text{cal}}(\mathbf{X}^{\text{op}}, \alpha)\big)\Big)$$

where $\mathcal{A}_{\tau_i} = \text{cal}_{\tau_i}(\mathbf{X}^{\text{cal}}, c)$ for some $i \in \{1, \dots, H\}$ and $g^{\text{cal}} := \text{conf}_T^g(\mathbf{X}^{\text{cal}}, c, \nu)$ being the calibrated correction function computed using the function specific calibration function.

There are two correction functions considered in Section 5. The first is the Bonferroni correction for α (Wasserman 2013, p. 166).

Definition 6. *The Bonferroni correction function* bf^{cal} *is* $\text{conf}_T^{\text{bf}}(\mathbf{X}^{\text{cal}}, c, \nu) := \frac{1}{|T|}$.

The disadvantage of the Bonferroni correction is that it is a worst case estimate, possibly leading to an over-correction of α. To combat such a behaviour an alternative method is proposed. This approach attempts to empirically estimate a better correction of α, by constructing a relationship between the observed-fpr and the set-fpr α during the calibration phase. That is, it constructs $\mathcal{A}_T^{c,\text{id},\nu,\mathbf{X}^{\text{cal}}}$, where id^{cal} simply returns α and runs $\mathcal{A}_T^{c,\text{id},\nu,\mathbf{X}^{\text{cal}}}(\mathbf{X}^{\text{op}}, \alpha)$ for various α, n-times, sampling randomly from $\mathbf{X}^{\text{cal}}$ to obtain $\mathbf{X}^{\text{op}}$ at each iteration, to collect observed fpr. Which subsequently can be used in conjunction with linear interpolation to construct a function that adjust α based on the observed-fpr.

Definition 7. *The empiric correction function is defined as* $\text{em}^{\text{cal}} = \text{conf}_T^{\text{em}}(\mathbf{X}^{\text{cal}}, c, \nu) := L_{\mathbf{w},\mathbf{v}}$ *for $\mathbf{v} \in [0,1]^n$ s.t. $v_i < v_{i+1}$ and with $\text{id}^{\text{cal}}(\mathbf{X}, \alpha) = \alpha$ for*

$$\mathbf{w} := \left(\frac{1}{n}\sum_{i=1}^{n} \mathcal{A}_T^{c,\text{id},\nu,\mathbf{X}^{\text{cal}}}\big(\text{sample}_M(\mathbf{X}^{\text{cal}}), v_i\big)\right)_{i \in \{1,\dots,n\}}$$

Section 5 explores two variants of $KS(T\text{-}conf)$, differing only in g with T being fixed to $T_{all} := \{\max, \text{margin}, L^2, \text{GM}, H\}$ and ν to max. Hence, for simplicity let $\text{cal}_{T_{all}}^{\text{bf}}$ be

the calibration function of $KS(T_{all}\text{-}bconf)$ where g is fixed to bf. Moreover, $\text{cal}^{\text{em}}_{T_{all}}$ is the same, but for g begin em with the sum running until $n = 10000$ and the vector set to $(0, 0.001, 0.005, 0.01, 0.015, 0.02, 0.03, 0.04, 0.05, 0.075, 0.1, 0.5, 1)$. Resulting in $KS(T_{all}\text{-}econf)$.

5 Experiments

The performance of the procedures was evaluated using three different cNet architectures. Namely, the medium sized *ResNet50* (He, Zhang, Ren, and Sun 2016) - abbreviated as *Res* - which is a network architecture commonly used in computer vision, as well as the *SqueezeNet* (Iandola, Han, Moskewicz, Ashraf, Dally, and Keutzer 2016) and the *MobileNet25* (Howard, Zhu, Chen, Kalenichenko, Wang, Weyand, Andreetto, and Adam 2017). Both of them, *Squ* and *Mob* respectively, are considered to be small and thus suitable for mobile applications. All networks were accessed in a pre-trained state using the publicly available python package tensornets[1]. Since, those network were trained on the training part of the *ImageNet ILSVRC 2012* dataset (Russakovsky, Deng, Su, Krause, Satheesh, Ma, Huang, Karpathy, Khosla, Bernstein, et al. 2015), the ImageNet ILSVRC 2012 test dataset (*Cal*) was chosen to be the calibration data set $\mathbf{X}^{\text{cal}}$. Hence, $\mathbf{x} \in \mathcal{X} := [0, 255]^{224\times224\times3}$ is an image, where the first dimension represents the hight, the second the width and the third the colour channels of the image. A single pixel will be addressed using $\mathbf{x}[i, j, k]$ for $(i, j, k) \in I := \{1, \ldots, 224\} \times \{1, \ldots, 224\} \times \{1, 2, 3\}$. The dimensions of 224×224 are required by each of the networks in question. Hence, all images used during the experiments had to be rescaled accordingly, this was accomplished using Ubuntu's (v20.10) `convert` program (Sun and Lampert 2019).

The calibration data set remains fixed across all experiments. By contrast, to simulate various operational scenarios $\mathbf{X}^{\text{op}}$ was sampled from multiple data sets. The first two are the *ImageNet ILSVRC 2012 validation dataset* (*Val*), which was included in order to observe the true fpr, and *AwA* abbreviating the *Animals with Attributes 2 dataset* (Xian, Lampert, Schiele, and Akata 2018). The remaining are collections of data sets obtained by distorting the images of *Cal*. One transformation simulates sensor noise, which is achieved by adding Gaussian noise to each pixel. That is, for $\sigma \in \{5, 10, 15, 20, 30, 50, 100\}$

$$N_\sigma := \left\{ \Big(\max\big(0, \min(255, \mathbf{x}[i, j, k] + \sigma \cdot \text{rnd}()\big)\Big)_{(i,j,k)\in I} \,\middle|\, \mathbf{x} \in \mathbf{X}^{\text{cal}} \right\}$$

where rnd() generates samples from a standard Gaussian distribution (Sun and Lampert 2019).

The other transformation simulates dead pixels by adding a percentage of salt and pepper noise (in a 50:50 ratio) to the existing images. That is, for some percentage $p \in \{1, 5, 10, 20, 40, 60, 80, 100\}$

$$D_\sigma := \left\{ \big(\text{dead}(\mathbf{x}[i, j, k], p)\big)_{(i,j,k)\in I} \,\middle|\, \mathbf{x} \in \mathbf{X}^{\text{cal}} \right\}$$

for

$$\text{dead}(x, p) := \begin{cases} 0 & \text{if } (i, j) \in J_b \\ 255 & \text{if } (i, j) \in J_w \\ x & \text{otw.} \end{cases}$$

[1] see https://github.com/taehoonlee/tensornets

where J_b and J_w are random subsets of $\{1, \ldots, 224\}^2$, both having the size $\lfloor \frac{p}{200} \cdot 224^2 \rfloor$ (Sun and Lampert 2019).

Given this, an experimental setting $E := (e_c, e_{\mathbf{X}^{\text{cal}}}, e_{\mathbf{X}^{\text{op}}}, e_{\text{cal}}, e_M, e_\alpha)$ is defined by the following parameters:

- cNet architecture, $e_c \in \{Mob, Res, Squ\}$;
- the calibration data, $e_{\mathbf{X}^{\text{cal}}} \in \{Cal\}$;
- the test data, $e_{\mathbf{X}^{\text{op}}} \in \{AwA, Val\} \cup \{N_\sigma \mid \sigma \in \{5, 10, 15, 20, 30, 50, 100\}\} \cup \{D_p \mid p \in \{1, 5, 10, 20, 40, 60, 80, 100\}\}$;
- the procedure, $e_{\text{cal}} \in \{\text{cal}_{\max}, \text{cal}_{\text{margin}}, \text{cal}_{L^2}, \text{cal}_{\text{GM}}, \text{cal}_H, \text{cal}^{bf}_{T_{all}}, \text{cal}^{em}_{T_{all}}\}$;
- the batch size (bs), $e_M \in \{10, 30, 50, 100, 300, 500, 1000, 3000, 5000, 10000\}$;
- the set fpr (α), $e_\alpha \in \{0.005, 0.01, 0.05, 0.1\}$.

Let E_{tpr} be the settings where $e_{\mathbf{X}^{\text{op}}} \neq Val$ and let E_{fpr} be the settings where $e_{\mathbf{X}^{\text{op}}} = Val$. To obtain fairly accurate results an experiment is conducted by running an experimental setting 10000 times, summing over the output of the experiments and dividing it by the number of iterations to establish the observed-fpr in the case of $e_{\mathbf{X}^{\text{op}}}$ being Val and the tpr for any other assignment of $e_{\mathbf{X}^{\text{op}}}$. That is, for some experimental setting E one computes $\mathcal{A}_E := e_{\text{cal}}(e_{\mathbf{X}^{\text{cal}}}, e_c)$ and runs $\frac{1}{10000} \sum_{i=1}^{10000} \mathcal{A}_E(\text{sample}_{e_M}(e_{\mathbf{X}^{\text{op}}}, e_\alpha))$.

5.1 KS(τ-conf) Results

There are two primary objective driving the analysis of the $KS(\tau\text{-}conf)$-procedures. The first, is to identify whether $KS(\tau\text{-}conf)$ for $\tau \in T_{new} := \{\text{margin}, L^2, \text{GM}, H\}$ improves upon the existing $KS(conf)$-procedure overall. The second, is to explore whether there are some observable differences between the various experimental settings. To gauge the performance, it is sufficient to solely compute the tpr by executing all E_{tpr}, as the underlying KS-test ensures a stable fpr.

Figure 1 provides an overview of all the results generated during the experiments, by depicting the tpr of each $\mathcal{A}_\tau$ for $\tau \in T_{new}$, relative to the tpr of $\mathcal{A}_{\max}$ for each experiment. This Using Figure 1 one can observe that $\mathcal{A}_H$ and $\mathcal{A}_{L^2}$ outperform $\mathcal{A}_{\max}$ in most settings, showing only some variation at low noise distortions for the Res architecture. Moreover, $\mathcal{A}_H$ tends to perform better than $\mathcal{A}_{L^2}$. Furthermore, $\mathcal{A}_{\text{margin}}$ consistently under-performs, and $\mathcal{A}_{\text{GM}}$ is subject to the greatest fluctuations. That is, not only are there great variations between cNet architectures, but also within distortion types, i.e. noise and dead pixel, dramatically flipping its position relative to $\mathcal{A}_{\max}$ by increasing or decreasing the amount of distortion. Figure 2a aggregates all tpr into box-plots, discriminating between the cNet architecture and the type of test datasets, i.e. AwA, the collection of N_σ datasets and the collection of D_p datasets and Figure 2b provides the same overview, but for batch sizes smaller or equal to 100. Further highlighting the behaviour of the geometric average, which significantly outperforms all other procedures on the AwA dataset on smaller network architectures. To summarise, $\mathcal{A}_H$ seems to consistently improve on $\mathcal{A}_{max}$. Moreover, $\mathcal{A}_{\text{margin}}$ is the only procedure that fails to do so. $\mathcal{A}_{\text{GM}}$ fluctuates, having both the highest peaks and deepest valleys, while $\mathcal{A}_{L^2}$ falls somewhere in-between $\mathcal{A}_H$ and $\mathcal{A}_{\max}$. Therefore, $\mathcal{A}_H$, $\mathcal{A}_{\text{GM}}$ will be carried over for the analysis of $KS(T\text{-}conf)$, while $\mathcal{A}_{\max}$ will also be retained serving as reference.

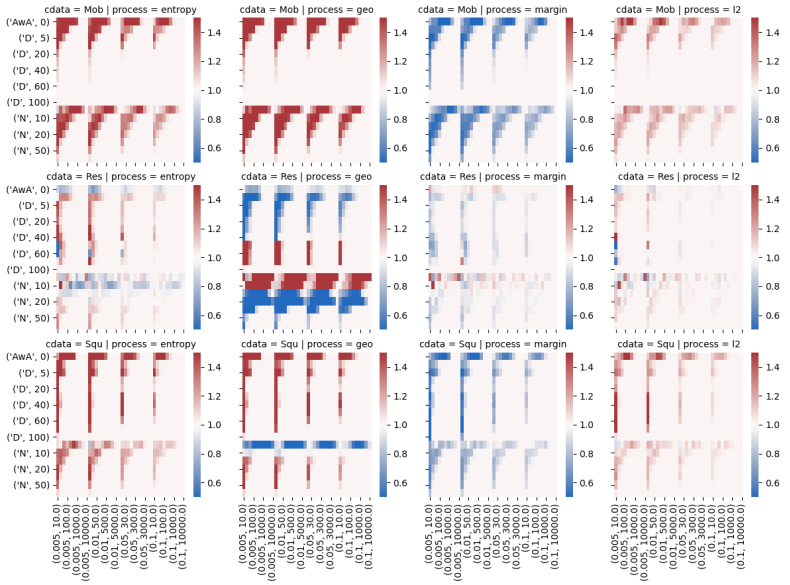
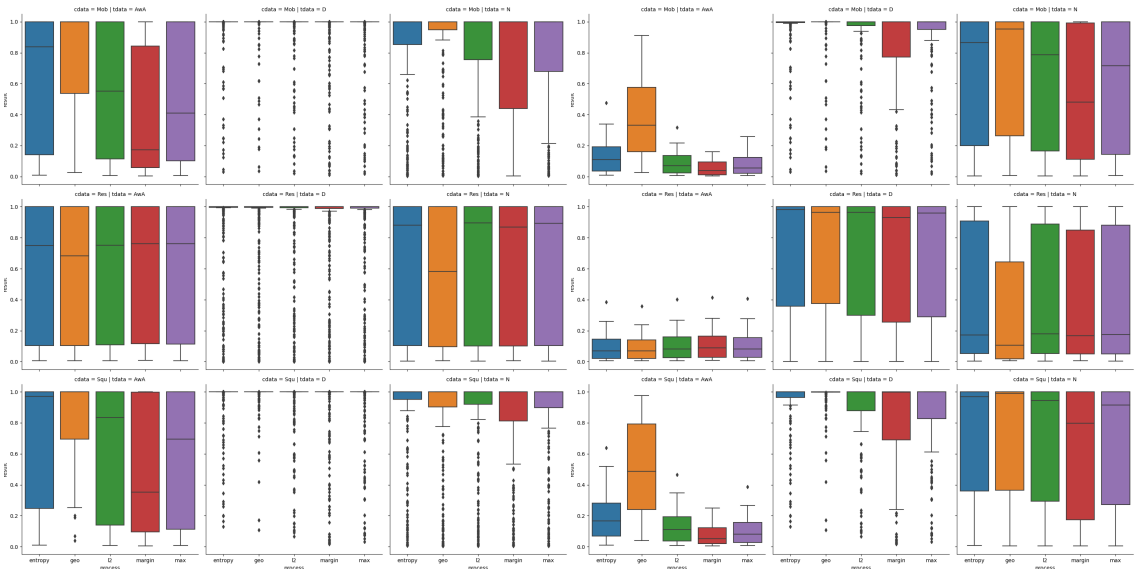
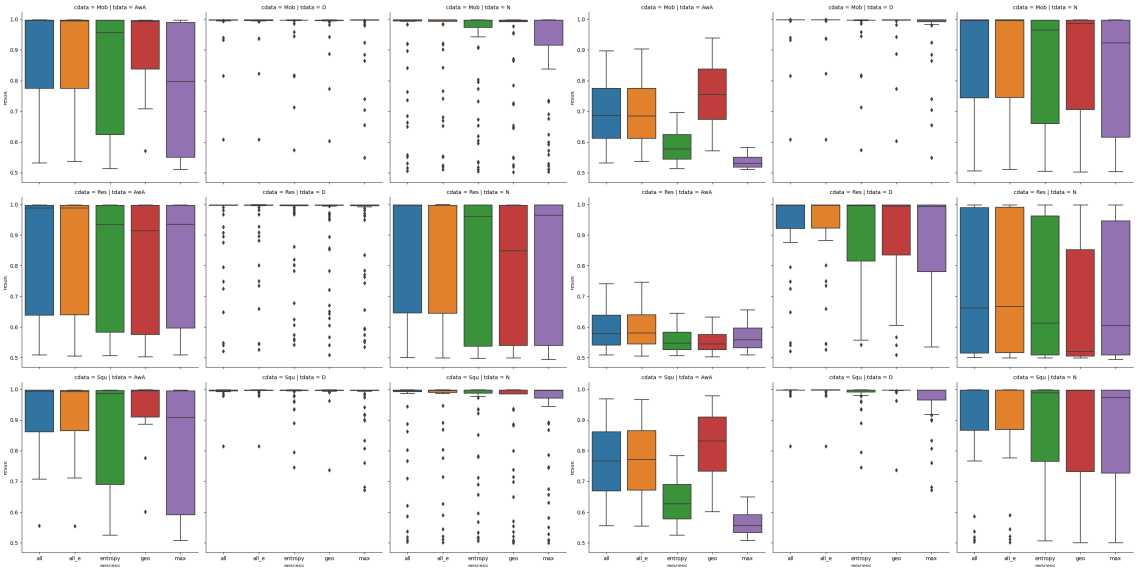


Figure 1: A heat map depicting the tpr of every $KS(\tau\text{-}conf)$ procedure divided by tpr of $\mathcal{A}_{\max}$ for every experimental setting, i.e. the x-axis ranges over $e_{\mathbf{X}^{\text{OP}}}$, the y-axis ranges over (e_α, e_M), the rows range over e_c and the columns range over e_{cal}. The colour read implies that the respective procedure outperforms $\mathcal{A}_{\max}$, while the colour blue indicates the opposite.



(a) (b)

Figure 2: Figure 2a is a box plot aggregating the tpr for each process, distinguishing between neural network architectures (rows) and out-of-spec dataset types, i.e. AwA, N_σ's, D_p's (columns); Figure 2b is identical to Figure 2a, but limited to experiments $e_M \leq 100$; $\mathcal{A}_H$ (Blue), $\mathcal{A}_{\text{GM}}$ (Orange), $\mathcal{A}_{L^2}$ (Green), $\mathcal{A}_{\text{margin}}$ (Red), $\mathcal{A}_{\max}$ (Violet).



(a) (b)

Figure 3: Identical to Figure 2, but rather than aggregating over the fpr the AUC was used instead. $\mathcal{A}^{\text{bf}}_{T_{all}}$ (Blue), $\mathcal{A}^{\text{em}}_{T_{all}}$ (Orange), $\mathcal{A}_H$ (Green), $\mathcal{A}_{\text{GM}}$ (Red), $\mathcal{A}_{\max}$ (Violet).

5.2 KS(T-conf) Results

Due to the heuristic adjustment of the fpr it is not possible to use the tpr as a measure of performance, e.g. consider some $KS(T\text{-}conf)$ procedure which has as voting mechanism the constant function 1. A better measure in this instance is the area under the receiver operating characteristic (ROC) curve, abbreviated as AUC. The ROC-curve, is established by varying α, observing both the tpr and the fpr, and relating them. The AUC is then a measure that summarises this relationship, allowing for the comparison of procedures, even if the fpr differs starkly (Sun and Lampert 2019). To observe the fpr all E_{fpr} were executed.

Figure 3, it is identical to Figure 2, but for the fact that the AUC, rather than the tpr is aggregated. As expected, this transformation does not have strong effects on the relative positioning between $\mathcal{A}_H$, $\mathcal{A}_{\text{GM}}$ and $\mathcal{A}_{\max}$. Moreover, one can observe that there is barely any difference between the $\mathcal{A}^{\text{em}}_{T_{all}}$ and $\mathcal{A}^{\text{bf}}_{T_{all}}$, which is reflected in the difference between the observed fpr, with $\mathcal{A}^{\text{em}}_{T_{all}}$ having on average an only 1.024 times greater fpr. However, by comparing the $KS(T\text{-}conf)$ procedures with the $KS(\tau\text{-}conf)$ procedures, it is quite apparent that the investigated $KS(T\text{-}conf)$ procedures consistently outperform $\mathcal{A}_H$, which more or less set the benchmark for the single-test procedures investigated above. Unfortunately, they fail to match $\mathcal{A}_{\text{GM}}$ on its highs. Furthermore, the comparison between Figure 3a and Figure 3b, suggest that the differences are more pronounced for smaller batch sizes. This is, why the ROC curves visible in Figure 5 were drawn for smaller batch sizes. Hence, they can be viewed as an exaggerated snapshot of the general trend sketched above. Lastly, Figure 4 suggests that the observed performance is relatively consistent across experiments, e.g. as compared to $\mathcal{A}_{\text{GM}}$.

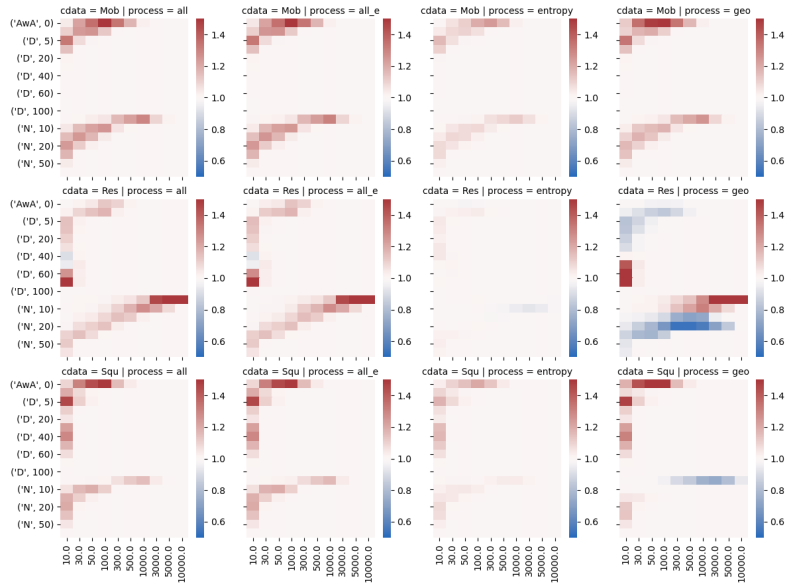
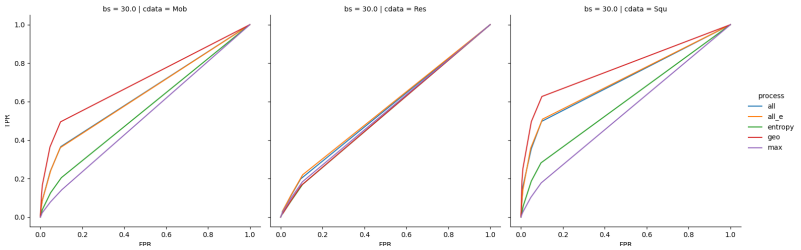
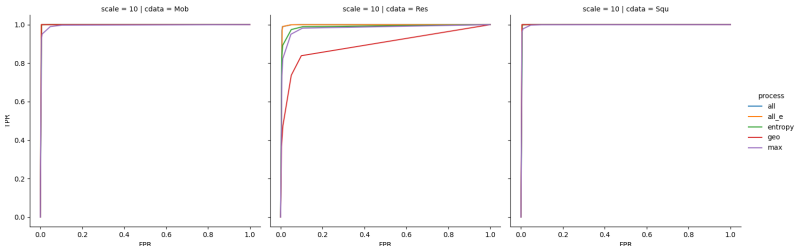


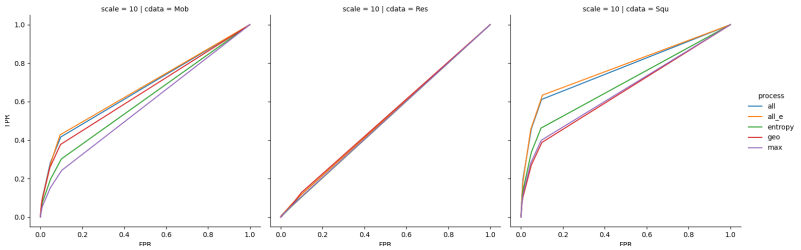
Figure 4: Similar to Figure 1, but for the fact that it depicts the relative performance based on the AUC measure.



(a) ROC curve for $e_{\mathbf{X}^{\text{OP}}} = AwA, e_M = 30$



(b) ROC curve for $e_{\mathbf{X}^{\text{OP}}} = D_5, e_M = 30$



(c) ROC curve for $e_{\mathbf{X}^{\text{OP}}} = N_{10}, e_M = 30$

Figure 5: ROC curves between the observed fpr and the tpr.

6 Conclusion

All procedures, but for the $\mathcal{A}_{\text{margin}}$ seem to be improvements over original $KS(conf)$ procedure. At the current state of testing $\mathcal{A}_H$ seems to be the best $KS(\tau\text{-}conf)$ procedure, generating consistently high tpr relative to $\mathcal{A}_{\max}$. Moreover, while $\mathcal{A}_{\text{GM}}$ can produce remarkable results in one setting, it may fail miserably in others. Although, impossible to verify given the small sample size, $\mathcal{A}_{\text{GM}}$ seems to perform quite well on smaller network architectures. Hence, further testing across a diverse array of architectures may be fruitful for better understanding the behaviour of those aggregation functions. The introduced $KS(T\text{-}conf)$ procedure shows promising results, further improving $\mathcal{A}_H$. It seems as the proposed correction function em, does not improve on the Bonferroni correction. However, as this may change with an increase in the size of T or with the employment of different voting mechanisms, discarding this approach may be pre-emptive. Lastly, the code used to generate those results can be accessed from the authors GitLab-page[2].

References

Shannon, Claude E (1948). "A mathematical theory of communication". In: *The Bell system technical journal* 27.3, pp. 379–423.

Massey Jr, Frank J (1951). "The Kolmogorov-Smirnov test for goodness of fit". In: *Journal of the American statistical Association* 46.253, pp. 68–78.

Marsaglia, George, Wai Wan Tsang, Jingbo Wang, et al. (2003). "Evaluating Kolmogorov's distribution". In: *Journal of Statistical Software* 8.18, pp. 1–4.

Wasserman, Larry (2013). *All of statistics: a concise course in statistical inference*. Springer Science & Business Media.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. (2015). "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115.3, pp. 211–252.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Iandola, Forrest N, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer (2016). "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 MB model size". In: *arXiv preprint arXiv:1602.07360*.

Howard, Andrew G, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam (2017). "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861*.

Xian, Yongqin, Christoph H Lampert, Bernt Schiele, and Zeynep Akata (2018). "Zero-shot learning—A comprehensive evaluation of the good, the bad and the ugly". In: *IEEE transactions on pattern analysis and machine intelligence* 41.9, pp. 2251–2265.

Sun, Rémy and Christoph H Lampert (2019). "KS (conf): A Light-Weight Test if a Multiclass Classifier Operates Outside of Its Specifications". In: *International Journal of Computer Vision*, pp. 1–26.

[2] https://git.ist.ac.at/kkueffne/ksconf_plus