

Highlights

Recognition Awareness: New Probabilistic Interpretation of Softmax Inference and Its Application

T Katanyukul,P Nakjai

- Research highlights item 1
- Research highlights item 2
- Research highlights item 3

Recognition Awareness: New Probabilistic Interpretation of Softmax Inference and Its Application

T Katanyukul^{a,*}, P Nakjai^{b,**}

^aComputer Engineering, Khon Kaen University, Thailand

^bUttaradit Rajabhat University, Thailand

ARTICLE INFO

Keywords:

softmax
probabilistic interpretation
open-set recognition
object recognition
recognition awareness
out-of-context identification
latent cognizance
novelty detection

ABSTRACT

This study investigates a new probabilistic interpretation of a softmax output from an inference model and its application on Open-Set Recognition (OSR). Softmax is a mechanism widely used in classification and object recognition. However, despite being credited as a part of numerous recognition achievements, a softmax mechanism forces a model to operate under a closed-set paradigm, i.e., to predict an object class out of a set of pre-defined labels. This characteristic contributes to both efficacy in classification and a risk of an adverse effect in object recognition.

Object recognition is often operated under a real world condition, which is immeasurably diverse and incomprehensibly dynamic. Balancing efficacy to accomplish a narrow task and awareness of a possibility of an encounter with a larger context is naturally undertaken by human intelligence. Yet for machine intelligence, this issue—studying under OSR—is still in its infancy state. A recent re-interpretation of softmax inference leads to a new approach allowing an out-of-context inference. It results in a mechanism, arguably more similar to natural intelligence than other OSR approaches in the sense of homogeneity with its functionality in a closed-set recognition.

Our study investigates the validity of the new interpretation using traceable problems and examines the application to OSR under various scenarios, using Imagenet 2012 dataset and fooling and open-set images. Our findings reveal the **validity of the new interpretation and the viability of its application on OSR**. In addition, new OSR evaluation metrics are proposed. These new metrics reflect a more comprehensive view of OSR performance. [I may remove this.] A complementary study is discussed and suggested for discerning the factors behind the perceiving results: here it exposes a huge contributing effect of a base recognizer and a potential improvement of 27% investing on a base recognizer alone. Applicability of OSR methods on detecting adversarial images is also investigated.


1. Introduction

Object recognition is a task identifying a type of a dominant object in an image. Its current state of the art achieves an impressive performance on thousands of categories ???. Its applications are extensive, including robot vision?, autonomous driving?, scene description?, customer behavior analysis?, and hand-sign recognition?. However, a deep network still operates under a closed-set paradigm. It employs a softmax mechanism to force prediction over a set of pre-defined class labels. Despite being highly effective for classification, this closed-set paradigm is often associated with an un-humanly fooled-able property of an object-recognition network that an image can be altered to appear perceptibly the same to human eyes, but it will cause a well-trained network to change its prediction, with high confidence, to a wrong class label??.

Real world is diverse, ever-evolving, and full of the unexpected. For example, in hand-sign recognition, an image of a signing hand is automatically translated to its corresponding alphabet. However, practical automatic hand-sign transcription needs to correctly identify a valid sign, as well as to be able to identify any non-sign posture when it happens. A non-sign posture might have appeared unintentionally during sign transition or from an unaware posture. Open-Set Recognition (OSR) is addressing this issue, generally with an ability to recognize an input that is beyond training experience. The implication of OSR could lead to a robust practical object recognition application. In addition, awareness of an input being beyond the interpolation range or beyond the expertise context can serve as a cornerstone for active learning?, transfer learning?, universal machine intelligence??, and publicly concerned AI safety?.

*Corresponding author

**Principal corresponding author

 tatpong@kku.ac.th (T. Katanyukul); mynameisbee@gmail.com (P. Nakjai)

ORCID(s): 0000-0003-3586-475X (T. Katanyukul)

This article investigates OSR. Given a well-trained inference deep network capable of recognizing images of multiple categories, the challenge is how we can exploit and enhance a well-trained network to tell when an input image does not belong to any categories it has been trained on.

OSR is different from object detection???. As ? have pointed out, an object-detection model is usually trained on images containing both positive and negative examples, while OSR is to recognize a category of an image as well as to identify if an image is of any category unseen at all in the training process. For conciseness, an “unseen” image will be referred to an image of any class that has not been included in the training process. Similarly, a “seen” image will be referred to an image of any class prepared in the training process.

The OSR has been actively studied. OpenMax? uses both thresholding and tail statistics. Tail statistics through Extreme Value Theory is to estimate the probability that an input image is unseen. OpenMax is reported with a marginal success?, while preparing tail statistics demands a considerable effort.

Observing that empirical evidence does not agree with a conventional interpretation of a deep network output, a new interpretation is proposed?. With the new interpretation and Bayes’ theorem, a relation between penultimate values and posterior probabilities is discovered. The discovery then founds a ground for a new method, Latent Cognizance (LC)?. LC exploits what has been learned in a deep network to estimate a probability if an image is seen. Our study investigates the effectiveness of both OpenMax and LC to OSR under various scenarios. Our study also revises OSR performance indices and an evaluation process. The revised indices account for every possible outcome. The revised process emphasizes a complementary study providing an insight into the underlying factors. In addition, a potential benefit of OSR methods as adversarial-image detectors is investigated.

Section 2 provides a general background and related approaches. Section 3 discusses Latent Cognizance in details. Section 4 describes our experiments and results and discusses evaluation metrics, a complementary study, and OSR on detecting adversarial images. Section 5 provides conclusions.

2. Background

Open-Set Recognition (OSR) can be viewed as a mapping task, $F^{OSR} : \mathbf{x} \mapsto \{0, 1, \dots, K\}$, where F^{OSR} is an inference model, e.g., a deep network??; an input image $\mathbf{x} \in \mathbb{B}^{H \times W \times C}$ is of height H , width W and C color channels, and pixel intensity $\mathbb{B} \in \{0, \dots, 255\}$; $1, \dots, K$ are indices of K pre-defined classes used in model preparation/training process, called “seen classes”; index 0 represents any class not used in the training process, collectively called an “unseen class”. With this outline, OSR can be viewed as a conventional object recognition with a novelty detection capability. Novelty detection is to identify whether an input is unseen or novel, i.e., an input is significantly different from samples used in the training process. An extensive review? outlines a general approach in novelty detection that (1) training samples are used to build a detector model; (2) given an input \mathbf{x} its score $s(\mathbf{x})$ is then computed from the model or its related function; and (3) a judgement is decided by thresholding.

Some novelty detection approaches are strongly probabilistic, i.e., the score function $s(\cdot)$ is associated with the most likely seen class probability or probability density function (pdf) of the seen data. However, estimating a pdf in practice is challenging, especially under a context of object recognition whose data are images— structured, locally related, and highly dimensional. It requires a powerful generative model along with an efficient mechanism to train it. For benefit much beyond novelty detection, a computationally feasible generative model is currently a subject of active research???

Many novel detection methods rely on some kinds of a distance-based scheme, using a distance between the input and the representatives of training samples to define a score function. Representation learning, e.g., ??, can facilitate this approach by providing a more relevant and concise data space to work on. This distance-based novelty detection approach is often characterized by a requirement of a search to identify seen representatives, which are either the nearest training samples or centroids of the nearest clusters of the training samples. This search is computationally expensive under a context of a large-scale object recognition. However, some studies also use distance measure as a score function, but manage to avoid an expensive search mechanism. An early work in facial recognition? uses Principle Component Analysis (PCA) to map an original facial image onto a reduced feature space. Since the feature space is crafted specifically for facial images, a facial image can be mapped onto a feature space and reconstructed back with little information loss, compared to a reconstruction of a non-facial image. Thus, a distance between an original image and the reconstruction can be used as a score function. The pivotal factor of this approach is an efficient reconstruction mechanism. Linearity of PCA and specificity of a facial domain allow a computationally convenient and effective reconstruction. Another notable novelty detection approach takes a holistic view of the problem. One-

class support vector machine (SVM)? models a boundary of seen samples on the projection space. A score function can be determined by a distance from the boundary to the input.

Designed solely for novelty detection, none of these approaches exploits a well-trained object recognizer, which is available under OSR. In addition, most of them do not scale well to OSR, which involves images and thousands of categories.

To address the issue from OSR perspective, ? formalize OSR definition and propose a concept of an open-space risk as well as a SVM-based one-vs-set machine. Instead of using a single hyperplane (as in One-class SVM?), the one-vs-set machine uses two hyperplanes to bound the seen samples, minimizing the open-space risk. Later, OpenMax? is developed and claimed to apparently outperform the one-vs-set machine on Imagenet dataset?.

OpenMax extends a well-adopted softmax object recognizer to account for the unseen and adjusts class probabilities based on an open-space risk and Extreme Value Theory. OpenMax is built on the following rationales. (1) Given a softmax output vector $\mathbf{y} = [y_1, y_2, \dots, y_K]$, summation of its components $\sum_{k=1}^K y_k$ should be less than one to keep some space left for the unknown. (2) The top ranking known classes (top highest y_k values), which in combination take most of the probability summation, should be more responsible for leaving some space for the unseen. Noted, each y_k was conventionally perceived as a class probability. (3) The longer distance between a class average and a penultimate value is, the more probability the class should sacrifice. (4) A very long distance is also a cue that the input may not belong to the class.

These rationales reflect to OpenMax two-phase scheme. (Phase 1: meta-recognition calibration.) OpenMax uses Extreme Value Theory—specifically Weibull distribution—to estimate intra-distance distributions of all seen classes. Specifically, given a K -class recognizer $f(\mathbf{x}) = \text{softmax}(f'(\mathbf{x}))$, and training input and output $(\mathbf{x}_1, \hat{y}_1), \dots, (\mathbf{x}_N, \hat{y}_N)$, the penultimate vectors $\mathbf{a}_n = f'(\mathbf{x}_n)$ for $n = 1, \dots, N$. For each class k , a class centroid along with Weibull parameters are computed, $\boldsymbol{\mu}_k = \text{mean}_{n \in C_k} \mathbf{a}_n$ and $(\tau_k, \beta_k, \lambda_k) = \text{fit}^{\text{Weibull}}(\{\|\mathbf{a}_n - \boldsymbol{\mu}_k\|\}_{n \in C_k}, \eta)$, where C_k is a set of indices of correctly classified samples belonging to class k ; $\text{fit}^{\text{Weibull}}$ is a function to fit the data to Weibull distribution; η is a user-specific tail size to specify that only η largest distances will be fitted to Weibull distribution; and τ_k 's, β_k 's, λ_k 's are Weibull parameters, i.e., locations, shapes, and scales. This phase appears as an extra learning process and requires all training samples in the process.

(Phase 2: class probability re-adjustment.) Then, at test time given an input \mathbf{x} , a penultimate vector, \mathbf{a} , is obtained through a base recognizer. (2a) OpenMax computes intra-distances of M top classes: $d_j = \|\mathbf{a} - \boldsymbol{\mu}_j\|$ for $j \in \{k : r_k \leq M\}_{k=1, \dots, K}$ when r_k is a class rank of class k corresponding to penultimate vector \mathbf{a} . (2b) Then, OpenMax re-adjusts M top-class probabilities: $y_j = \text{softmax}'(a_j^{(new)})$ if $j \in \{k : r_k \leq M\}$, where $\text{softmax}'(\cdot)$ is a softmax calculation including class 0 and $a_j^{(new)} = a_j \cdot (1 - \alpha_j \omega_j)$ when a_j is the j^{th} component of \mathbf{a} ; $\alpha_j = \frac{M - r_j}{M}$; and $\omega_j = 1 - \exp(-(\frac{d_j - \tau_j}{\lambda_j})^{\beta_j})$. (2c) A probability for an unseen class is computed: $y_0 = \text{softmax}'(a_0)$ when $a_0 = \sum_{j \in \{k : r_k \leq M\}} (a_j - a_j^{(new)})$. (2d) Finally, OpenMax thresholds out any seen class prediction if its probability is too small: predict seen class k if $k = \arg \max_i y_i$ and $y_k \geq \epsilon$ when ϵ is a user-specific threshold, otherwise predict unseen (i.e., either $0 = \arg \max_i y_i$ or $\max_{1 \leq i \leq K} y_i < \epsilon$). Note that OpenMax implicitly implies a uni-modal distribution of intra-class distances.

Extending further from merely identifying an unseen, Extreme Value Machine (EVM)? offers to learn an unseen class when found. EVM relies on building Ψ -models for every non-redundant input sample. Given an input \mathbf{x} , EVM makes a prediction by $y^* = \arg \max_{k \in \{1, \dots, K\}} y_k$ if $y_k \geq \epsilon$, otherwise $y^* = 0$, where $y_k = \arg \max_{n \in \mathcal{T}_k} \Psi(\mathbf{x}, \mathbf{x}_n, \beta_n, \lambda_n)$ for $k = 1, \dots, K$. A set \mathcal{T}_k contains indices of non-redundant training input samples of class k . Each Ψ -model is computed by, $\Psi(\mathbf{x}, \mathbf{x}_n, \beta_n, \lambda_n) = \exp(-(\frac{\|\mathbf{x} - \mathbf{x}_n\|}{\lambda_n})^{\beta_n})$, where β_n and λ_n are Weibull shape and scale parameters. Values of all β_n 's and λ_n 's are obtained using training data. Apparently, EVM comes with substantial computing costs: an extensive search through all training samples \mathbf{x}_n 's in every class, the need to compute values of Ψ -models and distances to all training samples, and requirement to prepare sets \mathcal{T}_k 's for all classes. EVM? try to mitigate this issue using redundancy removal along with Alexnet? as a potent feature extraction to digest original Imagenet input to 4096 features. In spite of these approaches, OSR remains greatly challenging?.

Related but having slightly different objectives, many studies??? investigate mechanisms to quantify uncertainty of a model inference. Inference uncertainty is a measure quantifying a degree of confidence or a level of expertise in making a particular prediction. ? have discussed that inference uncertainty is different from a model confidence, which

is conventionally taken as a value of each softmax output. A value of the softmax output can be shown to be very high (close to one) even when the input lies far beyond vicinity of the training samples in the input space. In this respect, quantifying inference uncertainty is similar to quantifying a degree of an unseen (in our context). However, a striking distinction between identifying an unseen and quantifying uncertainty is at the difficult classification or ambiguity among seen classes. Difficulty in distinguishing between seen classes is well encompassed by inference uncertainty, but this should not be recognized as an unseen.

While working on active learning criteria—how to select unlabeled samples from a data pool to ask experts in the most efficient manner—, ? propose a soft selection strategy along with approximation using Dirichlet process. The strategy is that samples with high approximate misclassification probabilities are more likely to be selected. ? choose a misclassification probability over an entropy—commonly used in active learning—for that an inclusion of a new class poses a great challenge for an efficient application of an entropy. Given an input \mathbf{x} , approximate misclassification probability $p(\text{wrong}|\mathbf{x}) = 1 - P_n(k'|\mathbf{x})$, where $P_n(k'|\mathbf{x}) \equiv p(y = k'|\mathbf{x})$; $k' = \arg \max_{k \in C} P_c(k|\mathbf{x})$ when C is a set of all seen classes and $P_c(k|\mathbf{x})$ is a probability calculated by a classifier, i.e., a softmax output. Based on Dirichlet process—specifically a chinese restaurant process—, $P_n(k|\mathbf{x})$ can be obtained through normalizing ?'s deduction: $P_n(k|\mathbf{x}) \propto \frac{M_k}{\alpha + \sum_{i \in C} M_i} P_c(\mathbf{x}|k)$ if $k \in C$ and $P_n(k|\mathbf{x}) \propto \frac{\alpha}{\alpha + \sum_{i \in C} M_i} P(\mathbf{x})$ if k is a new class, where M_k is a number of instances labelled with class k . ? obtain $P_c(\mathbf{x}|k) \equiv p(\mathbf{x}|k)$ and $P(\mathbf{x}) \equiv p(\mathbf{x})$ through kernel density estimation. Parameter α is a concentration coefficient of Dirichlet process. It can be user-specific, but ? use prior $\Gamma(1, 1)$ and Gibbs sampling to estimate its value. ? work on criteria to select data from a data pool. Therefore, they have all data available for $p(\mathbf{x}|k)$ and $p(\mathbf{x})$. This is generally a different situation from OSR. In addition, resorting to density estimation, a scalability aspect of this approach remains highly challenging.

Another interesting extension to object recognition, zero-shot learning, e.g. ?, takes a bold approach. Instead of directly mapping an input to a class label, which results in a fixed set up of pre-specified class labels, zero-shot learning focuses on mapping from input to a set of representative features. Its underlying assumption is that a set of representative features describes a true nature of a class, only without an explicit class label. This allows an ability to classify an instance of a new class, given sufficiently representing features. Despite potential and enthusiasm, this approach may not completely solve an issue of a rigid nature of an inference model. It merely shifts a challenge from extending a set of classes to extending a set of features. In addition, regarding OSR whose emphasis is on identifying an unseen, this approach can only provide a good feature extraction or dimension reduction and may have to resort to a distance-based approach to complete the task. Another recently active direction, one-shot learning? addresses an issue of learning with one or a few examples of a new concept, while avoiding catastrophic forgetting?—situation that a deep network loses its prediction ability of previously learned examples, after being trained with newly acquired examples. Its approaches are often resorted to a dedicated memory or an ability to augment its resources, including accessing to an extra memory or instantiating a new computing model. Similar to zero-shot learning, to recognize a new concept, it may resort to a distance-based approach.

Notably, many literature have somehow commented on a nature of classification inference that often found uncorrelated to class probability, especially when an input is unseen. Based on this observation, a new interpretation of a deep network output is proposed? to emphasize a limited nature of classification inference. This new interpretation has led to an invention of Latent Cognizance.

3. Latent Cognizance

Artificial neural networks including a deep network employ a softmax structure for a multi-class classification task. Softmax computation (1) is used at the last step of an inference. Given a task to predict one of K classes, the final prediction output $y = \arg \max_k y_k$, where softmax output $\mathbf{y} = [y_1, \dots, y_K]^T$ and, for $k = 1, \dots, K$,

$$y_k = \text{softmax}(\mathbf{a}) = \frac{\exp(a_k)}{\sum_{i=1}^K \exp(a_i)}. \quad (1)$$

Penultimate output $\mathbf{a} = [a_1, \dots, a_K]^T = f'(\mathbf{x}, \mathbf{w})$, when \mathbf{x} is an input, \mathbf{w} represents network parameters, f' is a network computation before the softmax. The realization of f' depends on a specific network configuration, while values of \mathbf{w} are obtained through a training process, minimizing cross-entropy loss.

Softmax computation normalizes penultimate values and regulates the final output such that $y_k \in [0, 1]$ and $\sum_{k=1}^K y_k = 1$ for all k 's. Since softmax output could provide a decent class prediction and all y_k 's agree with probabilis-

tic properties, a convention perception is that a well-trained network has its softmax output converged to the posterior probability: $y_k \equiv p(y = k|\mathbf{x})$. Softmax is effective and allows numerous successful classification applications.

Despite its efficacy, softmax output of a well-trained network is found unrelated to class probabilities when an input is unseen: there will be class k whose y_k value is large, although the input belongs to an unseen class, not class k . This observation contradicts the conventional interpretation: $y_k \equiv p(y = k|\mathbf{x})$. Taken this as evidence against the conventional view, ? have reinterpreted the softmax output as a conditional probability that the given seen input \mathbf{x} belongs to class k , i.e.,

$$y_k \equiv p(y = k|\mathbf{x}, s), \quad (2)$$

where s indicates the validity of the context, that \mathbf{x} is seen or $s \equiv (y = 1|y = 2|y = 3 \dots |y = K)$. This interpretation emphasizes the condition s , which has been subtle and overlooked. Based on (2) and Bayes' theorem, the softmax output can be written as:

$$y_k = p(y = k|\mathbf{x}, s) = \frac{p(y = k, s|\mathbf{x})}{\sum_{i=1}^K p(y = i, s|\mathbf{x})}. \quad (3)$$

Conferring (1) to (3), the relation:

$$\frac{\exp(a_k)}{\sum_i \exp(a_i)} = \frac{p(y = k, s|\mathbf{x})}{\sum_i p(y = i, s|\mathbf{x})} \quad (4)$$

is found. Consider similar patterns on both sides of (4), there may be a relation between penultimate values on the left side and the probabilities on the right side. Given a well-trained network, we assume that the penultimate vector \mathbf{a} relates to posterior probability $p(y = k, s|\mathbf{x})$ through function $\tilde{h}_k(\mathbf{a}) = p(y = k, s|\mathbf{x})$. Rather than working with $\tilde{h}_k(\mathbf{a})$, it is more convenient to work with a function whose value just correlates to the probability, because it is sufficient for identifying an unseen and it lessens a burden on enforcing probabilistic properties. Assume that there exists a monotonic function $g(\cdot)$ that $g(a_k(\mathbf{x})) \propto p(y = k, s|\mathbf{x})$, where $a_k(\mathbf{x})$ represents the k^{th} penultimate value corresponding to input \mathbf{x} . Thus, with marginalization, an unseen input \mathbf{x} can be identified by a low value of

$$\sum_{k=1}^K g(a_k(\mathbf{x})) \propto p(s|\mathbf{x}). \quad (5)$$

The function $g(\cdot)$ is called a cognizance function and the approach is called Latent Cognizance (LC). LC utilizes an already-trained network without requirement for extra training (c.f. meta-recognition calibration phase in OpenMax). Most other approaches including OpenMax determine a degree of being unseen of input \mathbf{x} by how unlikely input \mathbf{x} belongs to any seen class. Thus, they have to examine $p(y = i|\mathbf{x})$ for all classes $i = 1, \dots, K$, where K is a number of all seen classes. LC approach follows a new softmax interpretation, i.e., $y_k = p(y = k|\mathbf{x}, s)$, where s represents a state of being a seen class. Utilizing what softmax calculation has provided, how likely sample \mathbf{x} is unseen then can be directly deduced with minimal calculation (5).

4. Methodology

OpenMax and our proposed Latent Cognizance (LC) are investigated on Open-Set Recognition (OSR). Fig. 1 illustrates structural differences between OpenMax and LC. Conventional object recognition (OR) takes an image and predicts a class label out of pre-defined class labels. An internal structure of a conventional OR network resorts to a softmax layer at the end. OpenMax replaces a softmax layer with OpenMax computation. LC extends a conventional OR with LC computation for unseen identification, but still keeps a softmax layer for class recognition.

Instead of directly estimating a seen class probability $p(y = k, s|\mathbf{x})$, ? have proposed a cognizance function whose value correlates to a seen class probability $g(a_k) \sim p(y = k, s|\mathbf{x})$. They have empirically explored various candidates for $g(\cdot)$ and found cubic and exponential functions viable.

Our investigation here evaluates cubic and exponential cognizances along with state-of-the-art OpenMax on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) version 2012 dataset, as described in Table 1. A seen test dataset has 50000 images of all 1000 classes from ILSVRC 2012 validation set. An unseen test dataset is a

Recognition Awareness

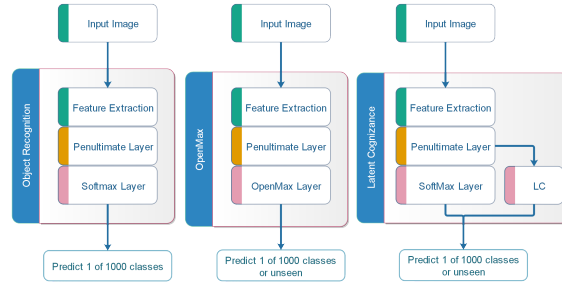


Figure 1: Architectures of conventional object recognition, OpenMax, and Latent Cognizance.

Table 1
Datasets for evaluation.

	Dataset	Remark
Learning data	2012 training set	OpenMax only
Test seen data	2012 validation set ¹	
Test fooling data	newly generated	
Test unseen data	2010 training set	only classes not in 2012

combination of 15000 fooling images and selections from 360 classes out of 1000 classes in ILSVRC 2010. All of the selected 360 classes are not in ILSVRC 2012. All of 15000 fooling images are generated based on an adversarial image generation algorithm loosely implemented ℓ_2 's formulation is $\min_r \|r\|_2$ s.t. $f(x+r) = k$; $x+r \in [0, 255]^D$, where r is an input perturbation, x is a base input of D dimensions, k is a target class label. $f(\cdot)$ is a classifier function. However, our implementation relaxes ℓ_2 's, i.e., find $r : f_k(x+r) > \alpha$ s.t. $x+r \in [0, 255]^D$, where $f_k(\cdot)$ is referred to the k^{th} output of the classifier. Parameter α is user specific. Our experiment sets α to 0.9. All fooling images have been inspected. They all are unrecognizable by human.

OpenMax and both LCs use Alexnet² with pre-trained weights as their base classifier. The pre-trained weights were obtained from Caffe software package², which has trained Alexnet on ILSVRC-2012 dataset to achieve 80.1% top-5 accuracy (57.1% top-1 accuracy) on ILSVRC-2012 validation set. Alexnet is used with these pre-trained weights and there is no fine tuning on Alexnet weights.

OpenMax requires learning of what have been seen, in its meta-calibration phase. Our experiment follows default values of ℓ_2 's implementation (Weibull tail size $\eta = 20$, a number of top classes $M = 10$, and using Euclidean Cosine method²). Cubic and exponential LCs do not require any extra learning process. OpenMax and both LCs employ feature information from a penultimate vector. All techniques are evaluated with performance indices $Q1 = \frac{F_s + F_u}{2}$ and $Q2 = \sqrt{F_s \cdot F_u}$, where F_s is an F-score of seen samples and F_u is an F-score of unseen and fooling samples. Specifically, F_s is an arithmetic mean of class F-scores, i.e., $F_s = \frac{1}{K} \sum_{i=1}^K F_i$, where F_i is an F-score of the i^{th} class and $\{1, \dots, K\}$ is a set of seen-class indices. F-scores F_i and F_u are defined as $F_i = \frac{2P_i \cdot R_i}{P_i + R_i + \epsilon}$ for $i \in \{1, \dots, K\}$ and $F_u = \frac{2P_u \cdot R_u}{P_u + R_u + \epsilon}$, respectively, where ϵ is a small number (set to 0.0001 in our experiment). Precisions and Recalls are defined as $P_i = \frac{TP_i}{TP_i + FP_i + \epsilon}$ and $R_i = \frac{TP_i}{TP_i + FN_i}$ for $i \in \{1, \dots, K\}$ and $P_u = \frac{TP_u + TP_f}{TP_u + TP_f + FP_u + \epsilon}$ and $R_u = \frac{TP_u + TP_f}{TP_u + TP_f + FN_u + FN_f}$. True positives TP 's, false positives FP 's, and false negatives FN 's are defined as shown in Table 2.

Note that ℓ_2 use a simpler metric. Their metric counts for only three cases: correct seen-class prediction (counted as TP), wrong seen-class prediction (counted as FP), and predicting a seen class on an unseen sample (counted as FN). Their metric does not account for predicting unseen on either an unseen sample or a seen sample. That consequently leaves predicting unseen on a seen sample un-penalized.

Although both OpenMax and LC rely on penultimate values. OpenMax estimates distributions of every class penultimate a_k . Thus it requires penultimate vectors of all training samples, as well as the penultimate vector $\mathbf{a} = [a_k]_{k=1, \dots, K}$ corresponding to the test sample. On the other hand, LC uses probability marginalization and establishes

¹Notably, ILSVRC 2012 validation set is chosen over the test set for its available ground truth.

²<https://github.com/abhihitbendale/OSDN>

Table 2

Performance metric for evaluating open-set recognition. Symbols i , i' , f , and u represent a seen sample of class i , a seen sample of class $i' (\neq i)$, a fooling sample, and an unseen sample, respectively. The evaluated systems do not have f output. Thus, predicting u on f is counted as TP_f .

Ground Truth	Prediction	Metric Count
i	i	TP_i
i	i'	FN_i and $FP_{i'}$
i	u	FN_i and FP_u
u	i	FN_u and FP_i
u	u	TP_u
f	i	FN_f and FP_i
f	u	TP_f

Table 3

Test cases. A different number of images per class is chosen to control a size of unseen data, which in turn reflects in a comfort ratio. Numbers of fooling and seen images are held fixed.

Test Case	I	II	III	IV
Comfort ratio	0.625	0.500	0.333	0.288
#images/class	42	97	236	300
Unseen data	15120	34920	84960	108000
Fooling data	15000	15000	15000	15000
Seen data	50000	50000	50000	50000

heuristic $\sum_k g(a_k)$ requiring only a penultimate vector corresponding to the test sample.

Diversity of data, or diverseness, can be quantified by a number of all classes used in a test dataset. The open ratio (o/r) is defined as, $o/r = \frac{\# \text{unseen classes}}{\# \text{seen classes} + \# \text{unseen classes}}$. The fooling ratio (f/r) is also defined in a similar manner. Therefore, given 1000 seen categories, 360 unseen categories, and 1000 fooling-image categories, our dataset has diverseness of 2360 with o/r of 0.2647 and f/r of 0.5. Previously proposed openness[?], formulated as $\text{openness} = 1 - \sqrt{\frac{2 \cdot |\text{training classes}|}{|\text{testing classes}| + |\text{target classes}|}}$, has a strong tendency toward a small classifier. For example, a small face verification system[?] trained with 12 classes to target 12 classes and tested on 50 classes has openness of 0.3778, while A larger system trained with 1200 classes to target 1200 classes and tested on 5000 classes has the same openness. Building a 1200-class verification system is more challenging than building a 12-class system, yet this aspect is not accounted at all in openness measure.

Thus, providing diverseness, o/r, and f/r would bring a more complete picture of the test scenario. In addition, the proportion of data is another thing to consider. Although numbers of classes may reveal the diversity of the data, proportion among seen, fooling, and unseen data also play an important role on the evaluation metrics. The ratio between a number of seen samples and a number of all samples, including seen, fooling and unseen classes, will be called a comfort ratio. Our investigation experiments 4 scenarios of different comfort ratios, as shown in Table 3.

Fig. 2 illustrates performance of all three OSR methods on the four scenarios. The y-axis represents performance measure: maximal Q1 (left plot) and maximal Q2 (right plot). The x-axis represents the comfort ratio. Details of our experimental results are shown in Tables 4 and 5. Table 4 shows performance indices—maximal Q1 and Q2. The last row shows a normalized time per image of each method. The normalized time per image is an average time spent to identify whether an image is unseen and is normalized by classification time per image. Alexnet—the base recognizer—is put into perspective. Alexnet does not have unseen identification and Q2 ostensibly reflects this. Table 5 reports all time durations spent in each operation. Classification time per image is a time Alexnet spent to classify an image into one of the seen classes. It is measured through an average time over classification of randomly chosen 50000 images. All three methods use Alexnet and are subject to the same classification time per image. Seen learning time reports a time spent to fine tune the open-set capability. A number in parenthesis represents a normalized time

Table 4
Performing results

	Comfort ratio(%)	OpenMax	Exponential LC $g(a) = \exp(a)$	Cubic LC $g(a) = a^3$	Alexnet
Q1	62.5	0.553	0.578	0.566	0.253
	50	0.579	0.575	0.535	0.231
	33.3	0.602	0.591	0.539	0.197
	28.8	0.606	0.595	0.542	0.186
Q2	62.5	0.549	0.574	0.562	0
	50	0.568	0.564	0.526	0
	33.3	0.578	0.563	0.508	0
	28.8	0.577	0.562	0.504	0
Normalized time		13.7	9.58×10^{-4}	1.24×10^{-3}	N/A

Table 5

Execution time. All methods are subject to the same classification time per image, which is 7.52×10^{-2} sec. A number in parenthesis represents a normalized time (over classification time per image).

	Seen Learning in sec.	Unseen Identification time/image in sec.
OpenMax	4.1×10^4 (5.5×10^5)	1.03 (13.7)
Exponential LC	0	7.2×10^{-5} (9.6×10^{-4})
Cubic LC	0	9.3×10^{-5} (1.2×10^{-3})

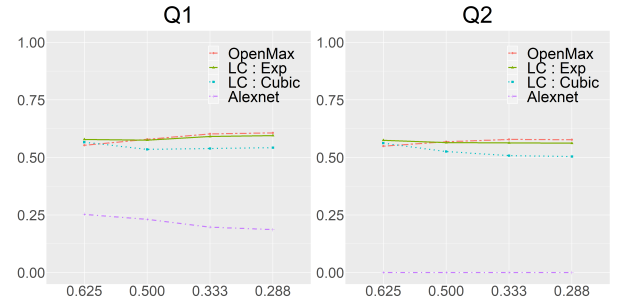


Figure 2: Open-set recognition performance over different comfort ratios. Legends “OpenMax”, “LC:Exp”, “LC:Cubic”, and “Alexnet” represent OpenMax, Exponential LC, Cubic LC, and Alexnet, respectively. Alexnet is a base recognizer and does not have an unseen-identification capability.

(over the classification time). OpenMax requires this fine tuning (in its meta-calibration phase). Exponential and cubic LCs do not require this learning time. Unseen identification time is a time spent to identify whether an image is seen or unseen given penultimate values. A number in parenthesis represents a normalized unseen identification time (over the classification time).

Performances of all three methods are comparable, but LC methods spent considerably less time than OpenMax did. In addition, OpenMax requires significant seen learning time, while both cubic and exponential LCs can work right off the shelf. All three methods seem to be robust against various comfort ratios.

Error analysis. Tables 6 and 7 show confusion matrices of OpenMax and exponential LC with thresholds at maximal Q1’s. The data has three distinct groups while prediction is only limited to either seen or unseen. The tables differentiate predicting seen on seen samples with correct classification (CR) and incorrect classification (IC), since this classification is more attribution of the underlying classifier than unseen identification capability itself. Table entries are obtained from $\sum_{i=1}^K TP_i$ (predicting a correct class on seen samples), $\sum_{i=1}^K FP_i - FN_u - FN_f$ (predicting an incorrect class on seen samples), FN_f (predicting seen on fooling samples), FN_u (predicting seen on unseen samples), FP_u (predicting unseen on seen samples), TP_f (predicting unseen on fooling samples), and TP_u (predicting unseen on unseen samples).

Since OSR performance incorporates both classification and unseen identification aspects, Table 8 shows separate

Table 6

OpenMax confusion matrices.

Comfort ratio(%)	Prediction	Data		
		Seen	Fooling	Unseen
62.5%	Seen	CR: 19981 IC: 4526	1315	3957
	Unseen	25493	13685	11163
50%	Seen	CR: 19755 IC: 4322	1090	8862
	Unseen	25923	13910	26058
33.3%	Seen	CR: 18043 IC: 3164	297	16987
	Unseen	28793	14703	67973
28.8%	Seen	CR: 17778 IC: 3010	224	20941
	Unseen	29212	14776	87059

Table 7

Exponential LC confusion matrices.

Comfort ratio(%)	Prediction	Data		
		Seen	Fooling	Unseen
62.5%	Seen	CR: 24527 IC: 11572	646	8518
	Unseen	13901	14354	6602
50%	Seen	CR: 20936 IC: 7561	4	12226
	Unseen	21503	14996	22694
33.3%	Seen	CR: 18365 IC: 5576	0	20251
	Unseen	26060	15000	64709
28.8%	Seen	CR: 17320 IC: 4931	0	22132
	Unseen	27749	15000	85868

accuracies, i.e., accuracies of unseen identification (denoted “U ACC”) and accuracies of classification (denoted “C ACC”). For example, U ACC of OpenMax at comfort 62.5% is 0.616 ($\# \text{correctly identified as seen or unseen} / \# \text{total test samples} = (19981 + 4526 + 13685 + 11163) / 80120$), while C ACC is 0.815 = $19981 / (19981 + 4526)$.

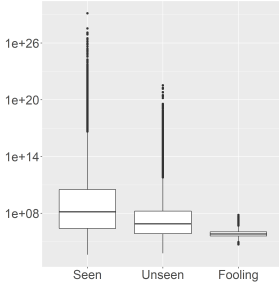
Fig. 3 shows boxplots of exponential LC values, $\sum_k \exp(a_k)$, on different sample groups at comfort 28.8%. On the left, exponential LC values of seen samples (including both correctly and incorrectly classified seen samples, denoted “Seen”), unseen samples (denoted “Unseen”), and fooling samples (denoted “Fooling”) are shown. On the right, values of correctly classified seen samples (denoted “Correct”) and ones of incorrectly classified seen samples (denoted “Incorrect”) are shown separately.

Breaking down performance into unseen identification and classification reveals that exponential LC performs pretty well on unseen identification. Its relatively constant U ACC reaffirms its robustness. The classification aspect is mostly attributed to the base classifier, Alexnet in our experiment. Although all methods employ the same Alexnet as their base classifier, classification accuracies are shown to be varied greatly. The explanation may be that unseen identification changes a number of seen samples to be evaluated for classification performance. For example, when difficult seen samples get incorrectly identified as unseen, this may hurt U ACC, but it may help C ACC as this may decrease a number of incorrectly classified samples. In addition to tail statistics and compact abating probability, OpenMax has thresholding on the maximal class probability. This mechanism filters out too low class probability and

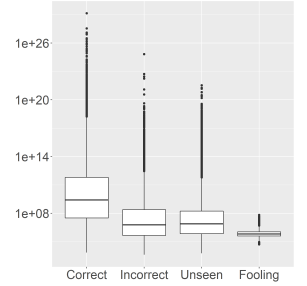
Table 8

Accuracies of unseen identification (U ACC) and accuracies of classification (C ACC).

	Comfort ratio(%)	OpenMax	Exponential LC	Cubic LC
U ACC	62.5	0.616	0.712	0.730
	50	0.641	0.662	0.639
	33.3	0.693	0.612	0.643
	28.8	0.709	0.712	0.670
C ACC	62.5	0.815	0.679	0.629
	50	0.820	0.735	0.658
	33.3	0.851	0.767	0.696
	28.8	0.855	0.778	0.708



(a) Seen, Unseen and Fooling groups.



(b) A seen group is broken down to correct and incorrect classifications.

Figure 3: Boxplots of exponential LC values on different groups at 28.8% comfort.

Table 9

OSR performing results after removing base-classifier weakness. Average difference and percentage increase are conferred to Table 4.

	Comfort ratio(%)	OpenMax	Exponential LC $g(a) = \exp(a)$	Cubic LC $g(a) = a^3$	Alexnet
Q1	62.5	0.758	0.783	0.757	0.415
	50	0.750	0.744	0.690	0.360
	33.3	0.733	0.720	0.651	0.288
	28.8	0.726	0.714	0.645	0.267
Q2	62.5	0.757	0.783	0.756	0
	50	0.748	0.742	0.690	0
	33.3	0.725	0.711	0.640	0
	28.8	0.716	0.702	0.629	0
Avg. Difference		0.163	0.162	0.147	
% Increase		28.4	28.2	27.4	

leads to OpenMax tendency toward predicting unseen. OpenMax thresholding mechanism may have provided a boost on OpenMax classification accuracies, as it could bring C ACC to reach 81.5%, conferring to its base classifier Alexnet top-1 accuracy of 57.1%.

Fig. 3 exposes an important aspect for evaluating OSR. While it is difficult to threshold for well separating seen and unseen as shown in the left boxplot, the use of cognizance heuristic values can well distinguish the correctly-classified seen samples from unseen samples as shown in the right boxplot. This may reveal that the true challenge of OSR is actually in differentiation between difficult classifying and unseen samples. This breakdown examination can expose clues to the underlying factors behind the final OSR performance and, as pointed out in ?, should be commonly practiced in an OSR study.

Since misclassification is more associated with classifying seen samples than identifying a novel pattern, then — rather than relying solely on a final OSR performance— examining OSR performance without incorrectly classified

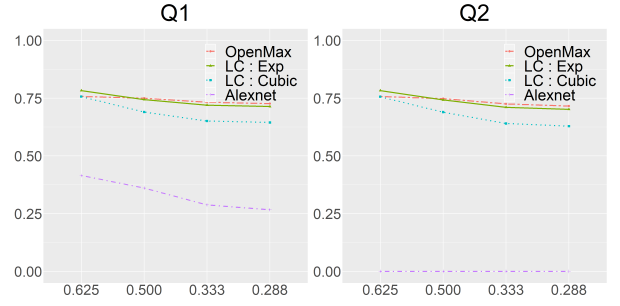


Figure 4: OSR performance over different comfort ratios after removing base-classifier weakness.

samples may allow more direct observation to the OSR extension capabilities. Table 9 and Fig. 4 show OSR performances of the three methods after removing the base-classifier weakness. The evaluation was conducted in a similar manner as described earlier, but all seen test samples that Alexnet—the base classifier— misclassified were discarded. All results seem much more promising: all Q1 and Q2 measures are over 0.6.

With average percentage increases more than 27%, these results remind and roughly quantify the significance of a base classifier contribution to the final OSR results. This revelation may help better planning and prioritizing for the improvement.

Accentuating a role of a base classifier is revived through this complementary study. Therefore, conducting this complementary study—observing correctly- and incorrectly-classified samples separately— allows an insight of contributing factors hidden in the final performance and is advised for a complete OSR investigation. In addition, a large number of incorrectly classified samples may reflect diversity and ambiguity in seen classes or immaturity of a classifier. Attention to this distinction may allow understanding of true causes of the perceiving results and a better chance to find an attainable solution.

Regarding performance indices, both Q1 and Q2 seem to well reflect OSR performances. Conferring to ?, OpenMax is reported on a case comparable to our 62.5%-comfort scenario with F-measure about 0.596, while Q1 and Q2 reflect to 0.553 and 0.549, respectively. Both Q1 and Q2 penalizes predicting unseen on seen samples, while ?'s F-measure does not. For three OSR methods, Q1 and Q2 come up with very similar numbers (the differences are no larger than 0.038). However, Q2 emphasizes Alexnet lack of unseen identification with 0, while Q1 simply reflects this with a low number. Therefore, both Q1 and Q2 are effective OSR performance measures, with Q2 being more sensitive to imbalance between classification and unseen identification.

Regarding detecting adversarial images, the results (Tables 7) may appear as if LC can effectively address this issue. However, fooling images were generated using random noise as base images and this may give fooling images away much easier than actual adversarial images. To properly examine the issue, 15000 adversarial images were generated and tested against the 50000 seen images. The adversarial images were generated in the same process generating fooling images described earlier, but, instead of random noise, the base images were randomly chosen from images of other 999 classes. The resulting images were visually inspected. Fig. 5 shows boxplots of exponential LC values of various image types, including adversarial images. Fig. 6 shows Precision-Recall (PR) plots of adversarial-image detection: binary classification whose positive refers to an adversarial sample and negative refers to a seen (left plot) or a correctly-classified seen (right plot). Table 10 shows Area Under Curves (AUCs) of each method. For perspective, a random classifier was tested on adversarial-image detection and achieved AUC 0.317 on average (10 repeats).

Small AUCs (top row, Table 10) when tested against seen samples rule out a side benefit of any of these methods as an effective adversarial-image detector. However, better AUCs (bottom row) when tested against correctly-classified samples may disclose some potential of these approaches, but without a dedicated study it may be too optimistic to speculate this.

For practical implementation, since an exponential function is numerically susceptible³, it is better to work on $\log \sum_k \exp(a_k)$ rather than directly on $\sum_k \exp(a_k)$. With $\log \sum_k \exp(a_k) = a_{\max} + \log \sum_k \exp(a_k - a_{\max})$ when $a_{\max} = \max_k a_k$, the implementation of $a_{\max} + \log \sum_k \exp(a_k - a_{\max})$ can be safely computed.

³IEEE754 allows upto about $\exp(709)$ on a 64-bit computer, with any larger number taken as an infinity.

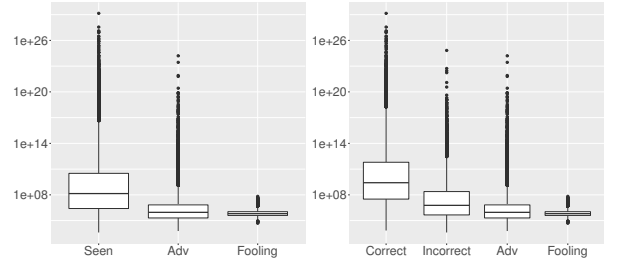


Figure 5: Boxplots of exponential LC values. Left: Seen, Adv (adversarial), and Fooling groups. Right: Correct and Incorrect groups are shown separately.

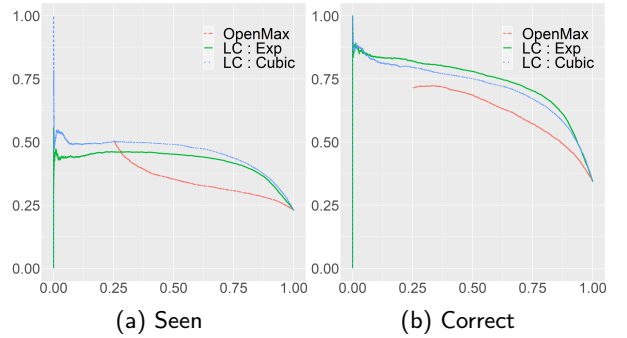


Figure 6: Precision-Recall plot of adversarial-image detection: positive refers to an adversarial sample.

Table 10

AUC: Detecting Adversarial Images.

	OpenMax	Exponential LC	Cubic LC
Seen	0.379	0.423	0.458
Correct	0.635	0.741	0.719

5. Conclusions

Our investigation has revealed viability of exponential Latent Cognizance for its performance on par with the state-of-the-art OpenMax, while its implementation and execution are much simpler and require far less resources.

In additions, our study introduces Open-Set Recognition (OSR) performance metrics Q1 and Q2, re-emphasizes a complementary OSR evaluation breaking down seen samples into correctly- and incorrectly-classified samples, highlights—as well as quantifies—a contributing effect of a base classifier to the final OSR results, and investigates OSR methods on adversarial images.

6. Bibliography

Two bibliographic style files (*.bst) are provided — model1-num-names.bst and model2-names.bst — the first one can be used for the numbered scheme. This can also be used for the numbered with new options of natbib.sty. The second one is for the author year scheme. When you use model2-names.bst, the citation commands will be like \citep, \citet, \citealt etc. However when you use model1-num-names.bst, you may use only \cite command.

thebibliography environment. Each reference is a \bibitem and each \bibitem is identified by a label, by which it can be cited in the text:

In connection with cross-referencing and possible future hyperlinking it is not a good idea to collect more than one

literature item in one `\bibitem`. The so-called Harvard or author-year style of referencing is enabled by the \LaTeX package `natbib`. With this package the literature can be cited as follows:

- Parenthetical: `\citep{WB96}` produces (Wettig & Brown, 1996).
- Textual: `\citet{ESG96}` produces Elson et al. (1996).
- An affix and part of a reference: `\citep[e.g.] [Ch. 2]{Gea97}` produces (e.g. Governato et al., 1997, Ch. 2).

In the numbered scheme of citation, `\cite{<label>}` is used, since `\citep` or `\citet` has no relevance in the numbered scheme. `natbib` package is loaded by `cas-sc` with `numbers` as default option. You can change this to author-year or harvard scheme by adding option `authoryear` in the class loading command. If you want to use more options of the `natbib` package, you can do so with the `\biboptions` command. For details of various options of the `natbib` package, please take a look at the `natbib` documentation, which is part of any standard \LaTeX installation.

A. My Appendix

Appendix sections are coded under `\appendix`.

`\printcredits` command is used after appendix sections to list author credit taxonomy contribution roles tagged using `\credit` in frontmatter.

CRedit authorship contribution statement

T Katanyukul: Conceptualization of this study, Methodology (examination of the new interpretation), Writing.
P Nakjai: Methodology (application to OSR), Data curation, Coding, Experimenting, Preparing result presentation.

T Katanyukul graduated B.Eng and M.Eng from King Mongkut's Institute of Technology Ladkrabang and Asian Institute of Technology, respectively. Both institutions are in Thailand. He earned his Ph.D. in 2010 from Colorado State University, USA. He currently works for Khon Kaen University. His research interests include pattern recognition and machine learning.

P Nakjai born in Chonburi, Thailand June 1985, graduated B.Eng (computer engineering) and M.Sci (computer science) from Naresuan University. He earned his Ph.D. in 2021 from Khon Kaen University and currently is serving Uttaradit Rajabhat University as a lecturer. His research interests are Convolution Neural Network, Long-Short-Term Memory and other machine learning techniques.