Learning Deep Classifiers Consistent with Fine-Grained Novelty Detection

Jiacheng Cheng Nuno Vasconcelos
Department of Electrical and Computer Engineering
University of California, San Diego
{jicheng,nvasconcelos}@ucsd.edu

Abstract

The problem of novelty detection in fine-grained visual classification (FGVC) is considered. An integrated understanding of the probabilistic and distance-based approaches to novelty detection is developed within the framework of convolutional neural networks (CNNs). It is shown that softmax CNN classifiers are inconsistent with novelty detection, because their learned class-conditional distributions and associated distance metrics are unidentifiable. A new regularization constraint, the class-conditional Gaussianity loss, is then proposed to eliminate this unidentifiability, and enforce Gaussian class-conditional distributions. This enables training Novelty Detection Consistent Classifiers (NDCCs) that are jointly optimal for classification and novelty detection. Empirical evaluations show that NDCCs achieve significant improvements over the state-of-the-art on both small- and large-scale FGVC datasets.

1. Introduction

Deep convolutional neural networks (CNNs) enabled significant breakthroughs in image classification [25, 49, 19]. However, CNN classifiers are trained under the *closed-world* assumption that test examples belong to one of the classes on which the CNN was trained. These are referred to as *seen* or *known* classes. This assumption is violated in many practical settings, e.g. medical diagnosis [46] or autonomous driving [3], where CNNs can be exposed to images from both seen and *unseen* classes, i.e. classes that do not appear in the training set. In this setting, CNNs are well-known to assign examples from unseen classes to seen classes with high confidence [6, 56]. In fact, an entire literature on adversarial attacks [51, 16] has grown out of this observation. *Novelty detection* aims to thwart this problem, by identifying and rejecting examples from unseen classes.

Novelty detection can be divided into the one-class and multi-class settings depending on the number of known classes. In one-class novelty detection [43, 36, 38], which is also known as one-class classification (OCC), all training examples are assumed from the same class and have no labels. When seen and unseen classes are from different domains, novelty detection becomes out-of-distribution (OOD) detection [20, 28, 52, 12, 27]. For instance, a classifier of handwritten digit images is confronted with natural images. While OCC and OOD detection have gained significant attention, they are best suited when seen and unseen classes are fundamentally different.

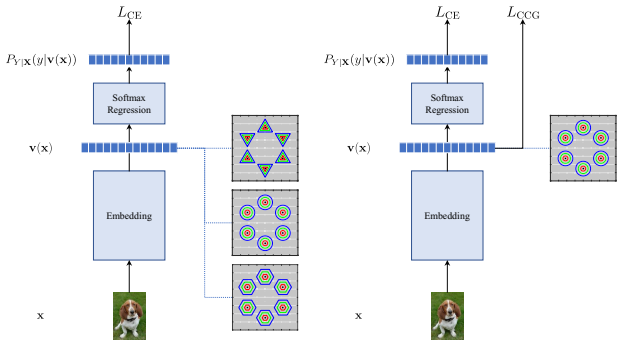
In this work, we address a different and more challenging setting where both seen and unseen classes are sub-classes (e.g., African hunting dog vs. Chesapeake Bay retriever) of a common category (e.g., dog). This is of very practical value for intelligent systems. For example, there might be cases in which it is necessary for surveillance systems deployed at wildlife sanctuaries to detect unseen animal species which might become alien-invasive species. Moreover, this is more frequent during the regular operation of vision systems. In applications such as autonomous driving, it is impossible to train for all object sub-classes that already exist, e.g. all road obstacles, or will be created after deployment of the classifier, e.g. new types of scooters or construction signs. Hence, sooner or later, the classifier will face unseen sub-classes. Since this type of novelty detection requires fine distinctions between seen and unseen classes, it is best addressed in the multi-class setting, where seen classes are modeled individually.

The core of a novelty detection algorithm is a *novelty score* or a measure of an example $\mathbf{x}$ not belonging to seen classes [9, 39]. This score can be computed by projecting $\mathbf{x}$ onto a feature space $\mathcal{V}$, usually the embedding learned by a CNN, and thresholded to produce a novelty detection decision. Two popular classes of novelty scores are probabilistic and metric-based [41]. The former estimates the probability of $\mathbf{x}$ under the distributions of seen classes. The latter estimates distances between $\mathbf{x}$ and seen class representatives. It can be shown that these two approaches are intrinsically connected for *exponential family* distributions [5], a family of probability densities that includes most parametric models in common use. Exponential family distributions are defined by a sufficient statistic, which can be seen as a feature

transformation or embedding, a set of canonical parameters, and a *cumulant* or log-partition function. The latter is a convex function of the canonical parameters and defines the geometry of the feature space: its derivatives are the moments of the distribution and its conjugate function defines the *Bregman divergence* [10] that underlies the geometry of $\mathcal{V}$ [4]. Hence, for exponential family features, probabilistic and metric scores are two faces of the same coin.

In this work, we leverage the fact that a CNN trained for *classification* induces exponential family class-conditional distributions on its embedding $\mathbf{v}(\mathbf{x})$, whose geometry thus follows the associated Bregman divergences. This enables novelty detection by simply thresholding the latter. The difficulty, however, is that the standard training by cross-entropy minimization produces class-conditional distributions and corresponding Bregman divergences, that are unknown. In fact, we show that both class-conditional distributions and Bregman divergences are *unidentifiable* from the CNN parameters. While seen classes are exponentially distributed, these parameters are compatible with many cumulant functions and, consequently, Bregman divergences. This is illustrated in Figure 1(a). Although novelty detection can be performed by assuming divergences of specific forms, e.g. Euclidean distances, this creates an *inconsistency* between classification and novelty detection that makes the latter suboptimal. It is thus important to consider alternate forms of CNN training that produce *classification CNNs consistent with novelty detection*. In this work, we propose to regularize CNN training so as to eliminate the unidentifiability of Figure 1(a). In particular, we seek regularization constraints that guarantee a desired (distribution, divergence) pair. While any pair could be chosen, we enforce multivariate Gaussian distributions and the associated Mahalanobis distances, for simplicity. However, given the high-dimensional nature of modern CNN embeddings, even covariances constraints are difficult to enforce. We show, however, that it is possible to leverage insights gained from the analysis of embedding geometry to derive a new *Class-Conditional Gaussianity* (CCG) regularization loss.

As shown in Figure 1(b), the combination of the CCG loss L_{CCG} with the standard cross-entropy loss L_{CE} can be seen as a loss function that operates on the two sides of softmax regression layer. On one hand, L_{CE} shapes the class-posterior probabilities at the output of the layer, ensuring optimal classification on seen classes. On the other, L_{CCG} shapes the class-conditional distributions at the layer input, forcing them to be Gaussian. Finally, because the output class-posterior probability distributions are compatible with any exponential family distribution for the class-conditionals, the addition of this regularization does not hinder classification performance. Overall, the resulting classifier is *consistent with novelty detection,* which can be equally implemented by thresholding class-conditional probabilities or Bregman divergences, with little loss of classification performance on seen classes.



(a) ND-inconsistent CNN (b) ND-consistent CNN

Figure 1. 1(a): A CNN trained for classification with the cross-entropy loss L_{CE} is inconsistent with novelty detection (ND). Because the class-conditional distributions learned by the CNN are unidentifiable, multiple sets of distributions (visualized using contour plots) are compatible with the CNN parameters. 1(b): Regularization with the proposed CCG loss L_{CCG} makes the distributions identifiable, in fact Gaussian, without sacrificing classification performance.

The paper makes four contributions to the study of novelty detection. The first is a theoretical analysis of the softmax classifier, showing that although it learns exponential family class-conditional distributions, these are not identifiable. The second is the derivation of *identifiability conditions*, that guarantee Gaussian distributions and associated Mahalanobis distances. The third is the CCG regularization loss that encourages these conditions to hold, producing classifiers that are consistent with novelty detection. Finally, evaluations on various fine-grained visual classification datasets demonstrate that our proposed method significantly advances the state-of-the-art for novelty detection.

2. Related Works

Novelty Detection: Novelty detection has long been investigated in the machine learning and signal processing literature [32, 33, 41]. While many strategies have been proposed, two are of particular relevance to this work: probabilistic and distance-based novelty detection. Probabilistic methods are based on estimating class-conditional probabilities. However, most density estimation techniques used in prior works, such as Gaussian mixtures [58] or kernel density estimation [24], do not scale well to high-dimensional data, e.g. high-resolution images. Distance-based approaches rely on distance metrics in feature space to compute distances (or similarity measures) between examples and known classes. As we will show in the next section, these two philosophies can be unified in the framework of CNN learning, but both the class-conditional distri-

butions learned by CNNs and the distances that define the geometry of CNN feature space are unidentifiable.

CNN-based methods: In recent years, many works have used deep neural networks for visual novelty detection [20, 28, 12, 30, 21, 39]. For example, [30] proposed to use the Kernel Null Foley-Sammon transform (KNFST) [9] to learn a mapping from CNN feature space to a kernel feature space, where novelty detection is performed by thresholding distances between test examples and seen classes. This approach and its variants [9, 8, 30] suffer from the limitation that the CNNs are not specifically trained to enable optimal novelty detection. Several works instead propose to train the CNN with an auxiliary dataset of examples from domains different from that of the seen classes [21, 13, 39]. These strategies are successful for out-of-distribution detection, where examples in the auxiliary dataset effectively mimic unseen test examples, teaching the CNN to discriminate between them and known classes examples. However, they lose effectiveness when known and novel examples are from fine-grained classes within the same category.

Generative Models: Deep generative models, such as variational autoencoders and generative adversarial networks, have also been proposed for novelty detection. For example, [46, 57, 2, 38] use the image reconstruction error or the latent vector reconstruction error produced by these models as novelty score. Alternatively, [40, 1] propose to employ generative models for modeling the probability distribution of known classes. However, methods based on image generation are usually only successful in simple scenarios, with low resolution (e.g., 28×28 or 32×32) images and small numbers of classes [11].

Related Topics: Out-of-distribution (OOD) detection can be viewed as a special case of novelty detection where novel examples are from another problem domains or datasets. However, as discussed above, most approaches to OOD detection are not suitable for the more challenging task of novelty detection within fine-grained classes. Multi-class novelty detection also has close relationships with research problems such as one-class classification (OCC) [36, 38] and open-set recognition (OSR) [45, 6, 31]. OSR aims to simultaneously identify unknown examples and classify examples from known classes. OCC approaches can be used for multi-class novelty detection by treating the multiple training classes as one super class. This, however, fails to exploit the richness of label information in the training data and tends to underperform in multi-class novelty detection. In addition to the aforementioned topics, uncertainty estimation and probabilistic neural networks [14, 18, 15, 26] aim to address the overconfidence issue of deep CNNs and may also be promising in novelty detection. The method of deep ensemble [26], which achieves state-of-the-art performance in uncertainty calibration [35], is evaluated for novelty detection in this work.

3. Novelty Detection Consistent Classifier

In this section, our approach for novelty detection is presented in details. We start by briefly reviewing the training of CNN classifiers and discussing the difficulties of identifying either class-conditional distributions or distance metrics learned by CNNs.

3.1. Learning CNN classifiers

Consider a classification problem with observations and labels drawn from random variables $\mathbf{X} \in \mathcal{X}$ and $Y \in \mathcal{Y} = \{1, \cdots, C\}$. A CNN performs classification in three stages. The first is a feature extractor or embedding $\mathbf{v} : \mathcal{X} \to \mathcal{V} \subset \mathbb{R}^d$ which maps an image $\mathbf{x} \in \mathcal{X}$ into a d-dimensional feature space $\mathcal{V}$. This is typically achieved through a sequence of layers combining convolutional and non-linear transformations. The second is a softmax regression

$$P_{Y|\mathbf{X}}(y|\mathbf{v}(\mathbf{x})) = \frac{e^{\langle \mathbf{w}_y, \mathbf{v}(\mathbf{x})\rangle + b_y}}{\sum_{k=1}^{C} e^{\langle \mathbf{w}_k, \mathbf{v}(\mathbf{x})\rangle + b_k}} \tag{1}$$

where $\mathbf{w}_y/b_y$ is the classification weight/bias for class y and $\langle \cdot, \cdot \rangle$ denotes the dot product . Finally, classification predictions are made by the Bayes decision rule

$$y^* = \arg\max_{y \in \mathcal{Y}} P_{Y|\mathbf{X}}(y|\mathbf{v}(\mathbf{x})). \tag{2}$$

CNNs are usually trained under the principle of maximum log-likelihood, i.e.,

$$\text{maximize} \sum_{(\mathbf{x},y)\in\mathcal{D}^{\text{train}}} \log P_{Y|\mathbf{X}}(y|\mathbf{v}(\mathbf{x})) \tag{3}$$

where $\mathcal{D}^{\text{train}}$ denotes the training set. This is typically done via stochastic optimization. Given a batch of training examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$, the CNN parameters are optimized by minimizing the cross-entropy loss:

$$L_{\text{CE}} = -\frac{1}{m}\sum_{i=1}^{m} \log \frac{e^{\langle \mathbf{w}_{y_i}, \mathbf{v}(\mathbf{x}_i)\rangle + b_{y_i}}}{\sum_{k=1}^{C} e^{\langle \mathbf{w}_k, \mathbf{v}(\mathbf{x}_i)\rangle + b_k}}. \tag{4}$$

Multi-class novelty detection [9, 39] addresses the detection of examples from new classes, on which the network has not been trained. For example, when a dog classifier trained on C breeds is faced with an example of a dog of an unseen breed. A simple solution is to threshold some confidence score derived from the class-posterior distribution of (1), e.g. its maximum value [20]. A value below the threshold signals that the CNN has little confidence on the image class, suggesting that the image is likely to be from an unseen class and should be rejected. However, the estimation of class-posterior probabilities via (1) is unreliable in the open-world case, where test examples can be from unknown novel classes. In fact, its underlying assumption that

$\sum_{k\in\mathcal{Y}} P_{Y|\mathbf{X}}(k|\mathbf{v}(\mathbf{x})) = 1$ does not hold anymore. Hence, this approach tends to be suboptimal for novelty detection.

This problem can be avoided by performing the novelty detection before the softmax layer, i.e. by acting directly on the output of the feature extractor $\mathbf{v}(\cdot)$. Two alternatives are possible. The first is to threshold the class-conditional probability distributions $P_{\mathbf{X}|Y}(\mathbf{x}|y)$ or $P_{\mathbf{X}|Y}(\mathbf{v}(\mathbf{x})|y)$ [41]. While these model the generative distribution of examples from the known classes, they are valid models to measure the probability of $\mathbf{x}$ under class y, even when $\mathbf{x}$ is from an unseen class. The second is to measure distances between $\mathbf{x}$ and some representative of the distributions of the known classes in the feature space $\mathcal{V}$, e.g. the class mean. The intuition is that, in $\mathcal{V}$, examples from class y cluster around the class mean. Novelty detection should thus be possible by either thresholding probability distributions or distances in $\mathcal{V}$. The main difficulty is that both the distributions learned by the CNN and the distances that define the geometry of $\mathcal{V}$ are usually unknown. In fact, as we show next, they are not even identifiable from the learned CNN.

3.2. Unidentifiability of Class-conditional Distributions

Using Bayes' rule, the class-posterior distribution can be written as

$$P_{Y|\mathbf{X}}(y|\mathbf{v}(\mathbf{x})) = \frac{P_{\mathbf{X}|Y}(\mathbf{v}(\mathbf{x})|y)P_Y(y)}{\sum_{k=1}^{C} P_{\mathbf{X}|Y}(\mathbf{v}(\mathbf{x})|k)P_Y(k)}. \quad (5)$$

It follows from (1) and (5) that the class-posterior distributions learned by a CNN are compatible with any set of class-conditional distributions of the form

$$P_{\mathbf{X}|Y}(\mathbf{v}(\mathbf{x})|y)P_Y(y) \propto_{\mathbf{x}} e^{\langle \mathbf{w}_y, \mathbf{v}(\mathbf{x})\rangle + b_y} \quad (6)$$

where $\propto_{\mathbf{x}}$ denotes a proportional relationship whose proportionality constant is determined by $\mathbf{x}$. This holds when

$$P_{\mathbf{X}|Y}(\mathbf{v}(\mathbf{x})|y) = q(\mathbf{x})e^{\langle \mathbf{w}_y, \mathbf{v}(\mathbf{x})\rangle - \psi(\mathbf{w}_y)} \quad (7)$$

$$P_Y(y) = \frac{e^{\psi(\mathbf{w}_y)+b_y}}{\sum_{k=1}^{C} e^{\psi(\mathbf{w}_k)+b_k}}, \quad (8)$$

where $q(\cdot)$ is a non-negative function and $\psi(\mathbf{w}_y)$ is a constant such that (7) integrates to 1. In this case, $P_{\mathbf{X}|Y}(\mathbf{v}(\mathbf{x})|y)$ is an exponential family distribution of canonical parameter $\mathbf{w}_y$, sufficient statistic $\mathbf{v}(\mathbf{x})$, and cumulant function $\psi(\cdot)$ [5]. However, the learned CNN only provides us with $\mathbf{v}(\mathbf{x})$ and $\mathbf{w}_y$. We cannot determine $P_{\mathbf{X}|Y}(\mathbf{v}(\mathbf{x})|y)$ for any $\mathbf{x}$ without knowledge of $\psi(\mathbf{w}_y)$ or $q(\mathbf{x})$. In other words, there are multiple exponential family distributions compatible with $\mathbf{v}(\mathbf{x})$ and $\mathbf{w}_y$ learned by the CNN. A toy example is provided in Supplementary Material to illustrate this. In conclusion, the class-conditional distributions $P_{\mathbf{X}|Y}(\mathbf{v}(\mathbf{x})|y)$ are *not identifiable* from the leaned CNN, as shown in Figure 1(a).

3.3. Unidentifiability of Bregman Divergence

The cumulant function $\psi(\cdot)$ of an exponential family distribution $P_{\mathbf{X}|Y}(\mathbf{v}(\mathbf{x})|y)$ possesses several important properties [34, 4, 5]. First, it is a convex function. Second, its first and second order derivatives satisfy $\nabla\psi(\mathbf{w}_y) = \boldsymbol{\mu}_y$ and $\nabla^2\psi(\mathbf{w}_y) = \boldsymbol{\Sigma}_y$, where $\boldsymbol{\mu}_y = \mathbb{E}_{\mathbf{X}|Y}[\mathbf{v}(\mathbf{x})|y]$ and $\boldsymbol{\Sigma}_y = \mathbb{E}_{\mathbf{X}|Y}[(\mathbf{v}(\mathbf{x}) - \boldsymbol{\mu}_y)(\mathbf{v}(\mathbf{x}) - \boldsymbol{\mu}_y)^\top|y]$ are the mean and covariance of $\mathbf{v}(\mathbf{x})$ under class y. Third, it has a conjugate function defined as

$$\phi(\boldsymbol{\mu}_y) = \sup_{\mathbf{w}} \{\langle \mathbf{w}, \boldsymbol{\mu}_y\rangle - \psi(\mathbf{w})\}, \quad (9)$$

and it is the canonical parameter $\mathbf{w}_y$ associated with ψ and $\boldsymbol{\mu}_y$ that achieves the supremum, i.e.

$$\phi(\boldsymbol{\mu}_y) = \langle \mathbf{w}_y, \boldsymbol{\mu}_y\rangle - \psi(\mathbf{w}_y). \quad (10)$$

From this, it follows that

$$\begin{aligned} & \langle \mathbf{w}_y, \mathbf{v}(\mathbf{x})\rangle - \psi(\mathbf{w}_y) \\ = \; & \langle \mathbf{w}_y, \boldsymbol{\mu}_y\rangle - \psi(\mathbf{w}_y) + \langle \mathbf{w}_y, \mathbf{v}(\mathbf{x}) - \boldsymbol{\mu}_y\rangle \\ = \; & \phi(\boldsymbol{\mu}_y) + \langle \mathbf{w}_y, \mathbf{v}(\mathbf{x}) - \boldsymbol{\mu}_y\rangle \\ = \; & \phi(\boldsymbol{\mu}_y) + \langle \nabla\phi(\boldsymbol{\mu}_y), \mathbf{v}(\mathbf{x}) - \boldsymbol{\mu}_y\rangle \\ = \; & -d_\phi(\mathbf{v}(\mathbf{x}), \boldsymbol{\mu}_y) + \phi(\mathbf{v}(\mathbf{x})) \end{aligned} \quad (11)$$

where

$$d_\phi(\mathbf{a}, \mathbf{b}) = \phi(\mathbf{a}) - \phi(\mathbf{b}) - \langle \nabla\phi(\mathbf{b}), \mathbf{a} - \mathbf{b}\rangle \quad (12)$$

is the Bregman divergence [10] between $\mathbf{a}$ and $\mathbf{b}$ associated with ϕ. Using (11), (7) can be rewritten as

$$P_{\mathbf{X}|Y}(\mathbf{v}(\mathbf{x})|y) = q(\mathbf{x})e^{\phi(\mathbf{v}(\mathbf{x})) - d_\phi(\mathbf{v}(\mathbf{x}), \boldsymbol{\mu}_y)} \quad (13)$$

$$\propto_{\mathbf{x}} e^{-d_\phi(\mathbf{v}(\mathbf{x}), \boldsymbol{\mu}_y)}. \quad (14)$$

Hence, learning a CNN under the cross-entropy loss endows $\mathcal{V}$ with a geometry defined by the Bregman divergence $d_\phi(\mathbf{v}(\mathbf{x}), \boldsymbol{\mu}_y)$. In fact, it can be shown that the correspondence between $P_{\mathbf{X}|Y}(\mathbf{v}(\mathbf{x})|y)$ and $d_\phi(\mathbf{v}(\mathbf{x}), \boldsymbol{\mu}_y)$ is bijective, i.e. there is a unique Bregman divergence for every exponential family distribution [4]. Since, as discussed in the last subsection, multiple exponential family distributions are compatible with the learned CNN, there are multiple Bregman divergences corresponding to them. Hence, like the class-conditional distributions, the distances defining the geometry of $\mathcal{V}$ are *not identifiable*, either.

3.4. Identifiability Regularization Constraints

In this work, we propose to add regularization constraints to CNN training so as to enable the identification of both the class-conditional probability distributions and the distance functions that define the geometry of $\mathcal{V}$. We note that these constraints do not affect the optimality of the

classifier, whose posterior distribution remains of the form of (1) and whose class-conditionals remain of the form of (6). The only difference is that we eliminate the extra degrees of freedom that make the learned CNN compatible with multiple distributions in the exponential family. This is accomplished by enforcing one particular distribution.

While, in principle, any member of the exponential family could be used, a natural choice is to require the distributions to be Gaussians with different means, i.e.

$$P_{\mathbf{X}|Y}(\mathbf{v}(\mathbf{x})|y) = \mathcal{G}(\mathbf{v}(\mathbf{x}); \boldsymbol{\mu}_y, \boldsymbol{\Sigma}) \tag{15}$$

where

$$\mathcal{G}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}. \tag{16}$$

Two immediate consequences are that, from (15) and (14), the associated Bregman divergence is the Mahalanobis distance

$$\begin{aligned} d_\phi(\mathbf{v}(\mathbf{x}), \boldsymbol{\mu}_y) &= \frac{1}{2}(\mathbf{v}(\mathbf{x}) - \boldsymbol{\mu}_y)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{v}(\mathbf{x}) - \boldsymbol{\mu}_y) \\ &= \frac{1}{2}\|\mathbf{v}(\mathbf{x}) - \boldsymbol{\mu}_y\|^2_{\boldsymbol{\Sigma}} \end{aligned} \tag{17}$$

and that

$$P_{\mathbf{X}|Y}(\mathbf{v}(\mathbf{x})|y) = K e^{-d_\phi(\mathbf{v}(\mathbf{x}), \boldsymbol{\mu}_y)} \tag{18}$$

where K is a constant. This enables novelty detection by thresholding Mahalanobis distances, which are intuitive and easy to compute. In fact, this simplicity has led many previous works to use Mahalanobis distances in $\mathcal{V}$ for tasks such as image retrieval [22, 29], out-of-distribution detection [27], person re-identification [42], etc.

The difficulty, ignored by most of these works, is that the Mahalanobis distance only reflects the geometry of $\mathcal{V}$ when (15) holds. This must be *enforced* during CNN training, as a regularization constraint. However, this constraint is not trivial to implement. A possibility would be to add one regularizer across classes, forcing the sample covariance of the feature vectors to be the desired $\boldsymbol{\Sigma}$. This has three problems. First, it is difficult to estimate a d-by-d covariance $\boldsymbol{\Sigma}$ when d is large (e.g. d = 4096 for our experiments in Section 4). Second, even if this were possible, it is not clear what the target covariance $\boldsymbol{\Sigma}$ should be. Third, and most important, forcing a distribution to have a certain covariance $\boldsymbol{\Sigma}$ is insufficient to guarantee that the distribution is Gaussian. In summary, this regularization would 1) require the specification of the target covariance $\boldsymbol{\Sigma}$ and 2) would not guarantee the desired Gaussianity. Both of these are undesirable properties. The following lemma provides a more effective and efficient path towards the regularization.

Lemma 1. *Consider an exponential family distribution $P_{\mathbf{X}|Y}(\mathbf{v}(\mathbf{x})|y)$ of sufficient statistic $\mathbf{v}(\mathbf{x})$ and canonical parameter $\mathbf{w}_y$. Then (15) holds if and only if*

$$\boldsymbol{\mu}_y = \boldsymbol{\Sigma}\mathbf{w}_y \tag{19}$$

where $\boldsymbol{\mu}_y$ and $\boldsymbol{\Sigma}$ are the mean and covariance of $\mathbf{v}(\mathbf{x})$ under class y.

3.5. The Class-Conditional Gaussianity Loss

The lemma shows that there is a simple way to guarantee Gaussian class-conditionals. It suffices to enforce the constraint of (19) during CNN training. Even this, however, is not trivial to implement. One possibility is to estimate $\boldsymbol{\mu}_y$ and $\boldsymbol{\Sigma}$ by the sample mean and sample covariance of the training examples and then minimize the norm of difference between the two sides of (19). This, however, is not well suited for the mini-batch style of optimization commonly used for CNN training.

In this work, we propose a better alternative. This consists of parameterizing $\boldsymbol{\Sigma}$ by learnable parameters $\boldsymbol{\theta}$ and then forcing $\mathbf{v}(\mathbf{x})$ under class y to have mean $\boldsymbol{\mu}_y(\boldsymbol{\theta}) = \boldsymbol{\Sigma}(\boldsymbol{\theta})\mathbf{w}_y$ and covariance $\boldsymbol{\Sigma}(\boldsymbol{\theta})$. We tackle this from two aspects.

First, $\{\boldsymbol{\mu}_k(\boldsymbol{\theta})\}_{k=1}^C$ and $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ should fit the data distribution in $\mathcal{V}$. This can be done efficiently by minimizing the negative log-likelihood (NLL) of the Gaussian models $\{\mathcal{G}(\mathbf{v}(\mathbf{x}); \boldsymbol{\mu}_k(\boldsymbol{\theta}), \boldsymbol{\Sigma}(\boldsymbol{\theta}))\}_{k=1}^C$ with respect to the training data. From (16), up to constants independent on $\{\boldsymbol{\mu}_k(\boldsymbol{\theta})\}_{k=1}^C$ and $\boldsymbol{\Sigma}(\boldsymbol{\theta})$, the NLL for mini-batch training examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$ is given by

$$L_{\text{NLL}} = \frac{\log|\boldsymbol{\Sigma}(\boldsymbol{\theta})|}{2} + \frac{1}{2m}\sum_{i=1}^m \|\mathbf{v}(\mathbf{x}_i) - \boldsymbol{\mu}_{y_i}(\boldsymbol{\theta})\|^2_{\Sigma(\boldsymbol{\theta})}. \tag{20}$$

Since this NLL minimization is just in order to update our estimation of $\{\boldsymbol{\mu}_k\}_{k=1}^C$ and $\boldsymbol{\Sigma}$, we only optimize the parameters of $\{\boldsymbol{\mu}_k(\boldsymbol{\theta})\}_{k=1}^C$ and $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ for minimizing L_{NLL}. In other words, we detach $\{\mathbf{v}(\mathbf{x}_i)\}_{i=1}^m$ in (20) from the computational graph for backpropagation.

Second, the embedding $\mathbf{v}(\cdot)$ should adapt so that $\mathbb{E}_{\mathbf{X}|Y}[\mathbf{v}(\mathbf{x})|y] = \boldsymbol{\Sigma}(\boldsymbol{\theta})\mathbf{w}_y$. We encourage this by minimizing the Mahalanobis distances from $\{\mathbf{v}(\mathbf{x}_i)\}_{i=1}^m$ to the corresponding means $\{\boldsymbol{\mu}_{y_i}(\boldsymbol{\theta}) = \boldsymbol{\Sigma}(\boldsymbol{\theta})\mathbf{w}_{y_i}\}_{i=1}^m$, i.e.,

$$L_{\text{MD}} = \frac{1}{2m}\sum_{i=1}^m \|\mathbf{v}(\mathbf{x}_i) - \boldsymbol{\mu}_{y_i}(\boldsymbol{\theta})\|^2_{\boldsymbol{\Sigma}(\boldsymbol{\theta})}. \tag{21}$$

Noting that it is easy for the minimization of Mahalanobis distances to get away with simply increasing the magnitude of $\boldsymbol{\Sigma}(\boldsymbol{\theta})$, we only optimize the parameters of $\mathbf{v}(\cdot)$ and $\{\boldsymbol{\mu}_k(\boldsymbol{\theta})\}_{k=1}^C$ for minimizing L_{MD}.

By combining (20) and (21), we have the identifiability regularization loss

$$L_{\text{CCG}} = \gamma L_{\text{NLL}} + L_{\text{MD}} \tag{22}$$

where $\gamma > 0$ is a multiplier that balances the contributions of the two terms. This proposed regularization has two advantages. First, there is no need to specify the target covariance $\mathbf{\Sigma}$, which is simply learned as a byproduct of the optimization. Second, the minimization of (22) also encourages the distribution of $\mathbf{v}(\mathbf{x})$ under class y to have mean $\mathbf{\Sigma}(\boldsymbol{\theta})\mathbf{w}_y$ and covariance $\mathbf{\Sigma}(\boldsymbol{\theta})$, it follows from Lemma 1 that it forces the class-conditional distributions $P_{\mathbf{X}|Y}(\mathbf{v}(\mathbf{x})|y)$ to be Gaussian. For this reason, we refer to (22) as the *Class-Conditional Gaussianity* (CCG) loss.

The cross-entropy loss of (4) and the CCG loss can be naturally combined into an overall objective

$$\mathcal{L} = L_{\text{CE}} + \lambda L_{\text{CCG}} \tag{23}$$

where $\lambda > 0$. As shown in Figure 1(b), this can be seen as a loss function that operates on the two sides of softmax layer. On one hand, L_{CE} shapes the class-posterior probabilities at the output of the layer, ensuring optimal classification on seen classes. On the other hand, L_{CCG} shapes the distributions at the layer input, forcing them to be Gaussian. Both losses constrain the classifier parameters $\{\mathbf{w}_k\}_{k=1}^{C}$ that connect the input to the output. This enforces the condition of (19), which guarantees consistency between the input and output distributions, removing the ambiguity at the input, where the class-conditional distributions are forced to be Gaussian. Finally, because the output class-posterior distributions are compatible with any exponential family distribution for the class-conditionals, the addition of this regularization does not hinder classification performance. Furthermore, the fact that the seen classes have identifiable class-conditional distributions simplifies novelty detection, since there is no need to explicitly learn the distance metric that defines the geometry of $\mathcal{V}$. This distances simply "fall out" of the optimization of (23), enabling improved novelty detection performance. These observations are validated by our empirical evaluation in Section 4.

3.6. Summary

The proposed Novelty Detection Consistent Classifiers (NDCC) is implemented as follows. First, a CNN is trained with known classes examples and the joint loss $\mathcal{L}$ of (23). This produces a pair of parameters $(\mathbf{w}_y, \mathbf{\Sigma}(\boldsymbol{\theta}))$ per class y, and encourages the class-conditional distributions to be Gaussians of mean $\widehat{\boldsymbol{\mu}}_y = \mathbf{\Sigma}(\boldsymbol{\theta})\mathbf{w}_y$ and covariance $\widehat{\mathbf{\Sigma}} = \mathbf{\Sigma}(\boldsymbol{\theta})$. Given a test example $\mathbf{x}$, its novelty score is computed as the smallest Bregman divergence of (17), between $\mathbf{x}$ and the known classes

$$\text{Novelty}(\mathbf{x}) = \min_{y \in \mathcal{Y}} \|\mathbf{v}(\mathbf{x}) - \widehat{\boldsymbol{\mu}}_y\|^2_{\widehat{\mathbf{\Sigma}}}. \tag{24}$$

	Dogs	FounderType	CUB-200	Caltech
fine-grained	✓	✓	✓	✗
# classes	120	200	200	256
# images	20580	1352600	6033	30607

Table 1. Statistics of the datasets used for evaluation.

A novelty detection decision is finally made by thresholding $\text{Novelty}(\mathbf{x})$. For practical applications, the threshold can be chosen by different strategies. A simple one is to choose a percentile of the distribution of novelty scores for examples from known classes (for instance, we can choose the 90th percentile if the acceptable false negative rate is 10%). In literature, the novelty score is usually used for performance evaluations of novelty detection methods.

4. Experiments

Datasets. NDCC was evaluated on three fine-grained datasets, Stanford Dogs [23], FounderType-200 [30] and CUB-200-2010 [53]. To show that NDCC is not limited to the fine-grained setting, we also conducted evaluations on the coarse-grained Caltech-256 [17] dataset. Some statistics and sample images from these datasets are given in Table 1 and Figure 2.

Test Protocol. For fair comparison, we followed the protocol (seen/novel and train/test splits, etc) used in the literature [30, 39]. All methods are evaluated with two backbone CNNs, AlexNet [25] and VGG-16 [49]. ImageNet pretrained models are used for initialization of NDCC. Novelty detection performance is evaluated with the area under the receiver operating characteristic curve (AUROC). This is a measure of average performance across all thresholds of (24), which captures the ability of a statistic (e.g., novelty score) to distinguish two groups (e.g., novel and seen classes) and is widely used in the novelty detection literature [9, 8, 30, 40, 38, 1, 30, 7].

Parametrization of $\mathbf{\Sigma}$. A generic covariance matrix $\mathbf{\Sigma} \in \mathbb{S}^d_{++}$ has $\frac{d(d+1)}{2}$ degrees of freedom. For large d, this is a very large number. For example, using a 4096-dimentional feature space $\mathcal{V}$ results in a number of degrees of freedom larger than the number of parameters of a 152-layer ResNet [19]. To overcome this difficulty, we restrict $\mathbf{\Sigma}$ to be diagonal and consider two parametrization strategies. From simple to complex, they are

1. $\mathbf{\Sigma} = \text{diag}(\sigma^2, \cdots, \sigma^2)$,
2. $\mathbf{\Sigma} = \text{diag}((\sigma^{(1)})^2, \cdots, (\sigma^{(d)})^2)$,

where $\sigma^{(j)} = \sigma + \delta^{(j)}$ and σ, $\{\delta^{(j)}\}_{j=1}^{d}$ are learnable parameters. Under these two strategies, the resulting Gaussians are respectively spherical and elliptical. For both of them, $\mathbf{\Sigma}$ is initialized as an identity matrix. Under strategy 1, it is true that L_{MD} of (21) bears certain resemblance

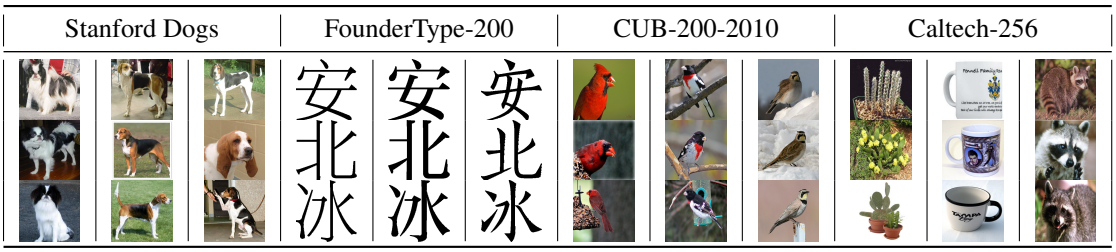


Figure 2. Sample images from the datasets used for evaluation. Images in each column are from the same class.

Method	Stanford Dogs		FounderType-200		CUB-200-2010		Caltech-256	
	AlexNet	VGG-16	AlexNet	VGG-16	AlexNet	VGG-16	AlexNet	VGG-16
Finetune [39]	0.702	0.766	0.650	0.841	0.638	0.684	0.785	0.827
OCSVM [47]	0.520	0.542	0.612	0.627	0.548	0.569	0.561	0.576
KNFST [9]	0.602	0.633	0.678	0.870	0.624	0.647	0.688	0.743
KNFST pre [9]	0.619	0.649	0.655	0.590	0.567	0.602	0.672	0.727
Local KNFST [8]	0.600	0.626	0.633	0.683	0.609	0.625	0.628	0.712
Local KNFST pre [8]	0.589	0.652	0.523	0.549	0.573	0.619	0.600	0.657
OpenMax [6]	0.711	0.776	0.667	0.852	0.664	0.708	0.787	0.831
MND [7]	0.762	0.904	-	-	-	-	0.751	0.882
TLN [39]	0.748	0.825	0.741	0.893	0.673	0.738	0.807	0.869
Deep Ensemble [26]	0.666	0.790	0.830	0.866	0.677	0.749	0.795	0.848
NDCC(σ)	0.814	0.913	0.922	0.952	0.702	0.752	0.791	0.886
NDCC($\sigma^{(j)}$)	**0.823**	**0.923**	**0.940**	**0.964**	**0.709**	**0.775**	**0.813**	**0.895**

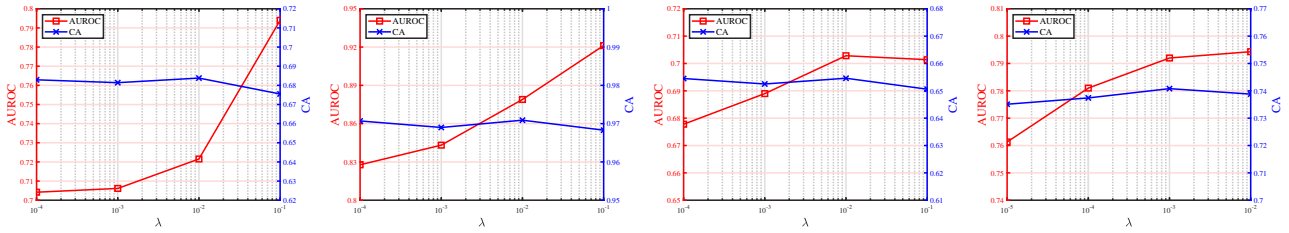
Table 2. Multi-class novelty detection performance (AUROC) of different methods. The best results are highlighted in bold, and the second best underlined. The suffix "pre" in a method name indicates that CNNs pre-trained on ILSVRC12 [44] are used for feature extraction.

to the center loss [54, 55], but the latter focus on enhancing the discriminative power of CNNs for face recognition. The NDCCs implemented with the two strategies are denoted as "NDCC(σ)" and "NDCC($\sigma^{(j)}$)", respectively.

Implementation Details. We adopted PyTorch [37] to train NDCCs by stochastic gradient descent (SGD) with momentum of 0.9. Weight decay of 0.0005 was applied to the parameters of CNN embedding $\mathbf{v}(\cdot)$. The SGD batch size was set to be 256 for all datasets. In practice, we found that the training of NDCCs can be significantly sped up if the embedding is L_2-normalized, i.e., $\|\mathbf{v}(\mathbf{x})\| = r, \forall \mathbf{x}$. This can be easily implemented with a L_2-normalization layer and a predefined multiplier $r > 0$. To minimize the discrepancies between training and test distributions, we disabled all dropout [50] layers in NDCCs. For all the datasets, we set the multiplier γ of (22) as $\gamma = \frac{1}{4096}$ and determined the multiplier λ of (23) by hold-out validation on the training set. More implementation details (hyperparameters such as learning rate, input size, etc) are included in Supplementary Material.

Comparison to the State-of-the-art. The NDCC variants are compared to several baseline methods including One-Class SVM (OCSVM) [47, 48], KNFST [9], Local KNFST [8], Transfer Learning Novelty (TLN) [39], Mixing Novelty Detection (MND) [7] and Deep Ensemble [26]. Among these, TLN and MND have state-of-the-art performance for multi-class novelty detection. A simple baseline "Finetune" is also used in the comparisons. This is to finetune the CNN with a cross-entropy loss and use the negative of the maximum activation in the last fully-connected layer as novelty score. For deep ensemble, the ensemble size is set as 5 and the maximum class-posterior probability (averaged over ensemble members' predictions) is used for novelty score, i.e., $\text{Novelty}(\mathbf{x}) = -\max_{y \in \mathcal{Y}} P_{Y|\mathbf{X}}(y|\mathbf{v}(\mathbf{x}))$.

The evaluation results are summarized in Table 2. Results of all baseline methods except deep ensemble on Standford Dogs, FounderType-200, and Caltech-256 are quoted from [39, 7]. All other results are produced by our experiments. A clarification on this is provided in Supplementary Material. The table shows that NDCC beats the state-of-the-art on all the four datasets. In fact, all variants of NDCC achieve state-of-the-art results for most networks and datasets. The only exception is NDCC(σ) which underperforms the state-of-the-art for the combination of AlexNet and Caltech-256. Among NDCCs, best performance is usually achieved with strategy 2, i.e. NDCC($\sigma^{(j)}$). For AlexNet/VGG-16, NDCC($\sigma^{(j)}$) outperforms the current state-of-the-art by a margin of 6.1%/1.9% on Stan-



(a) Stanford Dogs (b) FounderType-200 (c) CUB-200-2010 (d) Caltech-256

Figure 3. AUROC and closed-world classification accuracy (CA) versus λ.

Model	Stanford Dogs		FounderType-200		CUB-200-2010		Caltech-256	
	AlexNet	VGG-16	AlexNet	VGG-16	AlexNet	VGG-16	AlexNet	VGG-16
Finetune+$\mathcal{G}_1$	0.632	0.849	0.759	0.811	0.547	0.577	0.557	0.612
Finetune+$\mathcal{G}_2$	0.591	0.841	0.732	0.823	0.542	0.570	0.571	0.626

Table 3. Multi-class novelty detection performance (AUROC).

ford Dogs, 11.0%/7.1% on FounderType-200, 3.2%/2.6% on CUB-200-2010, and 0.6%/1.3% on Caltech-256.

Comparing datasets, the gains of NDCC are larger for Stanford Dogs, FounderType-200, and CUB-200-2010 than for Caltech-256. This can be explained by the fact that the formers are fine-grained datasets, while the latter is not. As shown in Figure 2, differences between fine-grained classes are subtler, requiring more sophisticated novelty detection algorithms. It is, in fact, worth noting that all NDCC variants significantly outperform TLN on all three fine-grained datasets, despite the fact that TLN uses extra auxiliary data (e.g. ILSVRC12 dataset) for training. This is unsurprising, since images from a completely different problem domain offer limited guidance on how to reject images from an unseen class within the same category of seen classes. Overall, while the methods in the literature have noticeably weaker performance for fine- than coarse-grained data, this is much less the case for NDCC.

Ablation Study. To further demonstrate the necessity of the proposed regularization, we evaluated the performance of simply modeling the class-conditional distributions in $\mathcal{V}$ of the "Finetune" method. Specifically, we modeled $P_{\mathbf{X}|Y}(\mathbf{v}(\mathbf{x})|y)$ using two Gaussian models:

1. $\mathcal{G}_1(\mathbf{v}(\mathbf{x}); \boldsymbol{\mu}_y, \text{diag}(\sigma^2, \cdots, \sigma^2))$,
2. $\mathcal{G}_2(\mathbf{v}(\mathbf{x}); \boldsymbol{\mu}_y, \text{diag}((\sigma^{(1)})^2, \cdots, (\sigma^{(d)})^2))$.

The parameters $\{\boldsymbol{\mu}_k\}_{k=1}^C$, σ, and $\{\sigma^{(j)}\}_{j=1}^d$ were learned by maximum likelihood estimation on the training set and the corresponding Bregman divergences were used to obtain the novelty score of (24). The resulting novelty detection performance is shown in Table 3. Comparing with the NDCC results of Table 2, shows that the proposed regularization is critical for the strong NDCC performance. Another observation is that the model with more covariance freedom (Finetune+$\mathcal{G}_2$) fails to guarantee better performance. This might be because there are not enough training examples to constrain the covariance estimation.

Closed-world Classification. To investigate the impact of the CCG regularization on the ability of CNNs to classify known classes, we evaluated the closed-world classification accuracy and the novelty detection performance of NDCC(σ) with an AlexNet backbone, as a function of λ in (23). The results are presented in Figure 3. While the novelty detection performance improves dramatically as λ increases, the classification accuracy on known classes remains nearly constant. This is consistent with our analysis in Section 3.5.

In addition, some qualitative results are presented in the Supplemental Material to visualize the efficacy of NDCC.

5. Conclusion

We considered the problem of novelty detection in fine-grained visual classification. We first showed that unidentifiability of both class-conditional distributions and distance metrics is a significant hurdle to learning CNNs jointly optimal for classification and novelty detection. To address this problem, we proposed a new regularization, the CCG loss, that enforces Gaussianity of class-conditional distributions. This was shown to enable state-of-the-art novelty detection results on both small- and large-scale fine-grained visual classification datasets.

Acknowledgements

This work was partially funded by NSF awards IIS-1924937 and IIS-2041009, a gift from Amazon, a gift from Qualcomm, and NVIDIA GPU donations. We also acknowledge and thank the use of the Nautilus platform for some of the experiments discussed above.

References

[1] Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Latent space autoregression for novelty detection. In *CVPR*, 2019. 3, 6

[2] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *ACCV*, 2018. 3

[3] Alexander Amini, Wilko Schwarting, Guy Rosman, Brandon Araki, Sertac Karaman, and Daniela Rus. Variational autoencoder for end-to-end control of autonomous driving with novelty detection and training de-biasing. In *IROS*, 2018. 1

[4] Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *Journal of Machine Learning Research (JMLR)*, 6(Oct):1705–1749, 2005. 2, 4

[5] Ole Barndorff-Nielsen. *Information and Exponential Families: in Statistical Theory*. John Wiley & Sons, 2014. 1, 4

[6] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *CVPR*, 2016. 1, 3, 7

[7] Supritam Bhattacharjee, Devraj Mandal, and Soma Biswas. Segregation network for multi-class novelty detection. In *WACV*, 2020. 6, 7

[8] Paul Bodesheim, Alexander Freytag, Erik Rodner, and Joachim Denzler. Local novelty detection in multi-class recognition problems. In *WACV*, 2015. 3, 6, 7

[9] Paul Bodesheim, Alexander Freytag, Erik Rodner, Michael Kemmler, and Joachim Denzler. Kernel null space methods for novelty detection. In *CVPR*, 2013. 1, 3, 6, 7

[10] Lev M Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200 – 217, 1967. 2, 4

[11] Philippe Burlina, Neil Joshi, and I-Jeng Wang. Where's wally now? deep generative and discriminative embeddings for novelty detection. In *CVPR*, 2019. 3

[12] Terrance DeVries and Graham W Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018. 1, 3

[13] Akshay Raj Dhamija, Manuel Günther, and Terrance Boult. Reducing network agnostophobia. In *NeurIPS*, 2018. 3

[14] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016. 3

[15] Yarin Gal, Jiri Hron, and Alex Kendall. Concrete dropout. In *NeurIPS*, 2017. 3

[16] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 1

[17] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007. 6

[18] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017. 3

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 6

[20] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017. 1, 3

[21] Dan Hendrycks, Mantas Mazeika, and Thomas G Dietterich. Deep anomaly detection with outlier exposure. *ICLR*, 2019. 3

[22] Prateek Jain, Brian Kulis, and Kristen Grauman. Fast image search for learned metrics. In *CVPR*, 2008. 5

[23] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, CVPR*, 2011. 6

[24] JooSeuk Kim and Clayton D Scott. Robust kernel density estimation. *Journal of Machine Learning Research (JMLR)*, 13(Sep):2529–2565, 2012. 2

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 1, 6

[26] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017. 3, 7

[27] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 2018. 1, 5

[28] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *ICLR*, 2018. 1, 3

[29] Daryl Lim and Gert Lanckriet. Efficient learning of mahalanobis metrics for ranking. In *ICML*, 2014. 5

[30] Juncheng Liu, Zhouhui Lian, Yi Wang, and Jianguo Xiao. Incremental kernel null space discriminant analysis for novelty detection. In *CVPR*, 2017. 3, 6

[31] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, 2019. 3

[32] Markos Markou and Sameer Singh. Novelty detection: a review—part 1: statistical approaches. *Signal Processing*, 83(12):2481–2497, 2003. 2

[33] Markos Markou and Sameer Singh. Novelty detection: a review—part 2:: neural network based approaches. *Signal Processing*, 83(12):2499–2521, 2003. 2

[34] John Ashworth Nelder and Robert WM Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384, 1972. 4

[35] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019. 3

[36] Poojan Oza and Vishal M Patel. Active authentication using an autoencoder regularized cnn-based one-class classifier. *CVPR*, 2019. 1, 3

[37] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming

Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 7

[38] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. Ocgan: One-class novelty detection using gans with constrained latent representations. In *CVPR*, 2019. 1, 3, 6

[39] Pramuditha Perera and Vishal M Patel. Deep transfer learning for multiple class novelty detection. In *CVPR*, 2019. 1, 3, 6, 7

[40] Stanislav Pidhorskyi, Ranya Almohsen, and Gianfranco Doretto. Generative probabilistic novelty detection with adversarial autoencoders. In *NeurIPS*, 2018. 3, 6

[41] Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014. 1, 2, 4

[42] Peter M Roth, Martin Hirzer, Martin Köstinger, Csaba Beleznai, and Horst Bischof. Mahalanobis distance learning for person re-identification. In *Person re-identification*, pages 247–267. Springer, 2014. 5

[43] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *ICML*, 2018. 1

[44] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 7

[45] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(7):1757–1772, 2012. 3

[46] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *IPMI*, 2017. 1, 3

[47] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001. 7

[48] Bernhard Schölkopf, Robert C Williamson, Alex J Smola, John Shawe-Taylor, and John C Platt. Support vector method for novelty detection. In *NeurIPS*, 2000. 7

[49] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1, 6

[50] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 15(1):1929–1958, 2014. 7

[51] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014. 1

[52] Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L Willke. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *ECCV*, 2018. 1

[53] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 6

[54] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016. 7

[55] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A comprehensive study on center loss for deep face recognition. *International Journal of Computer Vision (IJCV)*, 127(6):668–683, 2019. 7

[56] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. In *CVPR*, 2018. 1

[57] Houssam Zenati, Manon Romain, Chuan-Sheng Foo, Bruno Lecouat, and Vijay Chandrasekhar. Adversarially learned anomaly detection. In *ICDM*, 2018. 3

[58] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. *ICLR*, 2018. 2