Deep Transfer Learning for Multiple Class Novelty Detection

Pramuditha Perera and Vishal M. Patel
Department of Electrical and Computer Engineering
Johns Hopkins University, Baltimore, MD 21218, USA
pperera3@jhu.edu, vpatel36@rutgers.edu [*]

Abstract

We propose a transfer learning-based solution for the problem of multiple class novelty detection. In particular, we propose an end-to-end deep-learning based approach in which we investigate how the knowledge contained in an external, out-of-distribution dataset can be used to improve the performance of a deep network for visual novelty detection. Our solution differs from the standard deep classification networks on two accounts. First, we use a novel loss function, membership loss, in addition to the classical cross-entropy loss for training networks. Secondly, we use the knowledge from the external dataset more effectively to learn globally negative filters, filters that respond to generic objects outside the known class set. We show that thresholding the maximal activation of the proposed network can be used to identify novel objects effectively. Extensive experiments on four publicly available novelty detection datasets show that the proposed method achieves significant improvements over the state-of-the-art methods.

1. Introduction

In recent years, intelligent systems powered by artificial intelligence and computer vision that perform visual recognition have gained much attention [8],[12],[1],[25]. These systems observe instances and labels of known object classes during training and learn association patterns that can be used during inference. A practical visual recognition system should first determine whether an observed instance is from a known class. If it is from a known class, then the identity of the instance is queried through classification.

The former process is commonly known as novelty detection (or novel class detection) [14] in the literature. Given a set of image instances from known classes, the goal of novelty detection is to determine whether an observed image during inference belongs to one of the known classes. Novelty detection is generally a more challenging task than out-of-distribution detection [29], [9] since novel object samples are expected to be from a similar distribution to that of known samples.

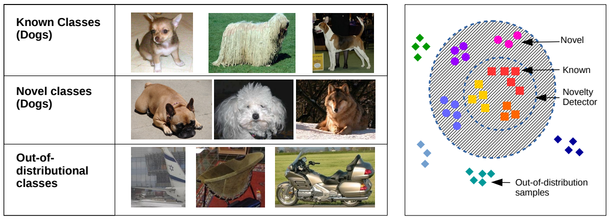


Figure 1. Novelty detection in dog-breed classification. **Left:** Sample images. **Right:** Feature representation. Both known (first row) and novel (second row) images are images of dogs. Given known images, the goal of novelty detection is to reject novel images. In order to do so, the knowledge of out-of-distribution images (final row), in this case non-dog images, are used to learn a suitable representation.

In practice, the knowledge of unknown classes is not entirely absent. Given a set of known classes from a certain problem domain, generally unknown class data from the same problem domain is unavailable. However, in some cases it is possible to obtain data outside the known class from different problem domains, which we refer to as *out-of-distribution* samples. For example, for a dog-breed recognition application, ImageNet dataset [21] that contains images of objects may be considered as *out-of-distribution* data as shown as in Figure 1. However, since the *out-of-distribution* data are from a different problem domain, they do not approximate the distribution of the *novel* samples well.

Nevertheless, since the deep-models produce generalizable features, the knowledge of *out-of-distribution* samples can be transferred to the original problem to aid novelty detection. When the problem considered is a c class problem, and when the *out-of-distribution* data of $\mathcal{C}$ classes are available, the following three strategies are used to transfer knowledge for novelty detection in the literature:

[*] This work was supported by the NSF grant 1801435.

1. **Fine-tuning**: Network is first pre-trained on the *out-of-distribution* data and later fine-tuned on the training data of the given domain. Novelty is queried by thresholding the final activation score [2].
2. **Feature Extraction**: Conventional novelty detection techniques [4],[13],[24] are used based on the fine-tuned features.
3. **Fine-tune** $(c+\mathcal{C})$: Network is first pre-trained on the *out-of-distribution* data. Both the training data and the *out-of-distribution* data are used to perform fine-tuning in $(c+\mathcal{C})$ classes together. Novelty is determined in the same way as in approach 1.

We note that in all these baselines, the *out-of-distribution* data is employed in the training process. In fact, any novelty detection method operating on the pre-trained/finetuned deep features are implicitly making use of the *out-of-distribution* data. In this work, we introduce a new framework to perform novelty detection based on transfer learning. First, we show that using cross-entropy loss alone for training is not optimal for the novelty detection task. Secondly, we empirically show that the *out-of-distribution* data can be used more effectively in training to produce better novelty detection performance with respect to considered baseline solutions. Specifically, we make following primary contributions in this paper.

1. We propose an end-to-end novelty detection framework based on deep learning. To the best of our knowledge, this is one of the first end-to-end deep learning solutions targeting visual novelty detection.
2. We introduce a new loss function, *membership loss* which has a similar functionality to that of the cross-entropy loss but encourages an embedding that produces high activation for known object classes consistently.
3. We propose to take advantage of large-scale external datasets to learn the *globally negative filters* to reduce high activations caused by the novel images.
4. We show empirically, that the proposed method outperforms the baseline novelty detection methods across four publicly available datasets.

2. Related Work

Object classification schemes are often equipped with a suitable mechanism to detect novel objects. For example, Eigenfaces [28] was accompanied by a reconstruction error-based novel object detection method. In sparse representation-based classification (SRC) algorithm [30], Sparsity Concentration Index (SCI) was proposed for the same purpose. In contrast, there is no formal novelty detection mechanism proposed for deep-learning based classification. In its absence, thresholding the highest class activation score of the deep model has been used as a baseline in the literature [2]. As an alternative, several recent works have proposed novelty detection schemes based on deep features [2],[24]. In the same spirit, it is also a possibility to use classical novelty detection tools such as Kernel PCA [10], Kernel null space-based novelty detection (KNFST) [4] and its variants [3],[13] on deep features. KNFST operating on deep-features produces the current state of the art performance in visual novelty detection [13]. However, advantages of deep-learning are not properly exploited in all of these approaches due to the absence of an end-to-end learning framework.

On the other hand, novelty detection problem has a close resemblance to both anomaly detection [17],[19],[20], [18], [5],[16], and open-set recognition problems [22],[2],[7],[15]. Therefore, it is possible to solve anomaly detection using tools proposed in these alternative domains. In anomaly detection, given a single *normal* class, the objective is to detect out-of-class instances. One-class SVM [23] and SVDD [27] are two of the most widely used tools in anomaly detection. Novelty detection can be viewed as an anomaly detection problem if all known classes are considered as a single augmented class. On the other hand, objective in open-set recognition (OSR) is similar to that of novelty detection. But in addition, OSR requires correct classification of samples detected as known samples. Therefore, it is also possible to use open-set recognition tools to perform novelty detection. However, we note that due to subtle differences in objectives, OSR algorithms are not optimal for novelty detection.

In the proposed framework, maximal activation of the final layer of a deep network is considered as a statistic to perform novelty detection. We design the network and choose loss functions appropriately so that this statistic is low for novel objects compared to the known object classes.

3. Background

In this section, we briefly review how deep networks produce activations in response to input stimuli. Based on this foundation, we introduce the notion of *positive filters* and *negative filters*. Consider a c class fully-supervised object classification problem with a training image set $\mathbf{x} = x_1, x_2, \ldots, x_n$ and the corresponding labels $\mathbf{y} = y_1, y_2, \ldots, y_n$ where $y_i \in \{1, 2, \ldots c\}$. Deep convolutional neural networks (CNNs) seek to learn a hierarchical, convolutional filter bank with filters that respond to visual stimuli of different levels. In c class classification, the top most convolutional filter activation $\mathbf{g}$ is subjected to a non-linear transformation to generate the final activation vector $\mathbf{f} \in \mathbb{R}^c$ (for example, $\mathbf{g}$ is the conv5-3 layer in VGG16 and conv5c in Resnet50. $\mathbf{f}$ is the fc8 and fc1000 layers in the respective networks). In a supervised setting, network parameters are learned such that $\arg\max \mathbf{f} = y_i$ for $\forall i \in \{1, 2, \ldots, n\}$. This is conventionally done by optimizing the network parameters based on the cross-entropy loss.

If there exist k filters in the top most convolution filter

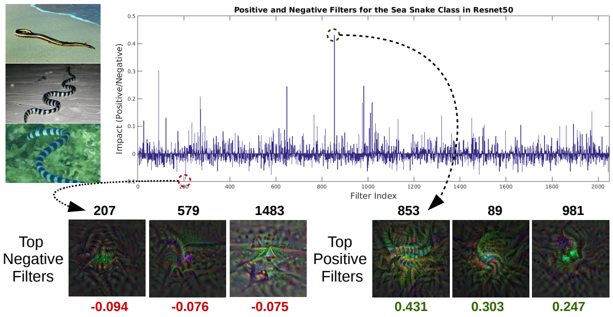


Figure 2. Positive and negative filters of the *sand snake* class in the Resnet50 trained on ILSVRC12 dataset. Top: weights of the fully connected layer corresponding to the *sand snake* class. We call filters associated with positive weights as positive filters of *sand snake* class. All other filters are named as negative filters. Bottom: Visualization of top negative and positive filters. These patterns are likely to produce high activation in these filters. We note top positive filters are activated by snake-like structures.

bank, its output $\mathbf{g}$ is a set of k number of activation maps. The final activation vector of the network $\mathbf{f}$ is a function of $\mathbf{g}$. For a given class i, there exists some k_i filters in the filter bank ($1 \leq k_i \leq k$) that generally generates positive activation values. These activations provide supporting (positive) evidence that an observed image is from class i. Conversely, all remaining filters provide evidence against this hypothesis. Activation score of each class in $\mathbf{f}$ is determined by taking into account the evidence for and against each class. For the remainder of the paper, we call filters that provide evidence for and against a particular class as *positive filters* and *negative filters* of the class, respectively.

This concept can be easily explained by taking the Resnet architecture [8] as an example. In Resnet, final convolution output $\mathbf{g}$ is subjected to global average pooling followed by a fully connected layer. Therefore, the i^{th} component of the final activation vector $\mathbf{f}$ can be written as $\mathbf{f}_i = W_i \times GAP(g)$, where GAP is global average pooling operation (mean of filter map) and W is the weight matrix of the fully connected layer. Here, activation of the i^{th} class is a weighted summation of mean feature maps found in $\mathbf{g}$. From the above definition, filters associated with positive weights for a given class in W can be identified as *positive filters* for that particular class. Conversely, filters associated with the negative weights become *negative filters* of the class.

For example consider the *Sand Snake* class appearing in the ILSVRC12 dataset [21]. Shown in Figure 2 (top) are the weights associated with the *Sand Snake* class in the final fully connected layer of the Resnet50 network trained on the ILSVRC12 dataset. We recognize filters associated with positive and negative weights as positive and negative filters, respectively for the given class. In Figure 2 (bottom) we visualize per-unit visualization of top positive and top negative filters for the considered class using the DeepVis toolbox [31] (these are the images that are most likely to activate the corresponding filters). By observation, we notice that the top *positive filters* are activated when the network observes structures similar to snakes. On the other hand, the top *negative filters* are unrelated to the appearance of *sand snakes*.

4. Deep Novelty Detection

Based on the above background, we propose to learn the distributions of known object classes using a CNN framework with the objective of performing joint classification and novelty detection. In our formulation, assuming each known class has a unique single label, we force the final activation vector $\mathbf{f}$ to model the probability distribution vector of known classes. Formally, for a given data-label pair (x_i, y_i), we expect $\mathbf{f}_i = 1$ and $\mathbf{f}_j = 0$, $\forall j \neq i$. Once such a representation is learned, $\arg\max \mathbf{f}$ returns the most-likely class of an observed sample. Then, $\max \mathbf{f}$ yields the likeliness of the sample belonging to the most likely class. Similar to binary classification, identity I of a test instance can be queried using hard thresholding. In order to learn a representation suitable for the stated objective, we use conventional classification networks as the foundation of our work and propose the following two alternations.

1. Membership loss. Assuming each known class has a unique single label, if the observed image is from a known class, only a single positive activation should appear in $\mathbf{f}$. We observe that when cross-entropy loss is used, this is not the case. To alleviate this, we introduce a new loss called *membership loss* in addition to the cross-entropy loss.

2. Globally negative filters. In a classification setting, a negative filter of a certain class is also a positive filter of another class. In other words, there exist no explicit negative filters. In our formulation, we propose to generate *globally negative filters* (filters that generate negative evidence for all known classes) to reduce the possibility of a novel sample registering high activation scores.

4.1. Limitations of Cross-Entropy Loss

When a classification network is trained, each element f_i of the activation vector $\mathbf{f}$ is first normalized using the softmax function to arrive at a normalized activation vector $\tilde{\mathbf{f}}$ as in, $\tilde{f}_j = e^{f_j} / \sum_{l=1}^{c} e^{f_l}$. When it is assumed that all image classes appearing during inference are known ahead of time, j^{th} element of vector $\tilde{\mathbf{f}}$ is interpreted as the likelihood of the input image x_i belonging to the j^{th} class. Neural network-based classification systems are learned by minimizing the cross-entropy loss which is the negative log likelihood of the correct class $\tilde{\mathbf{f}}$. However, since this is a relative measure, the learned representation deviates from our objective due to the following reasons.

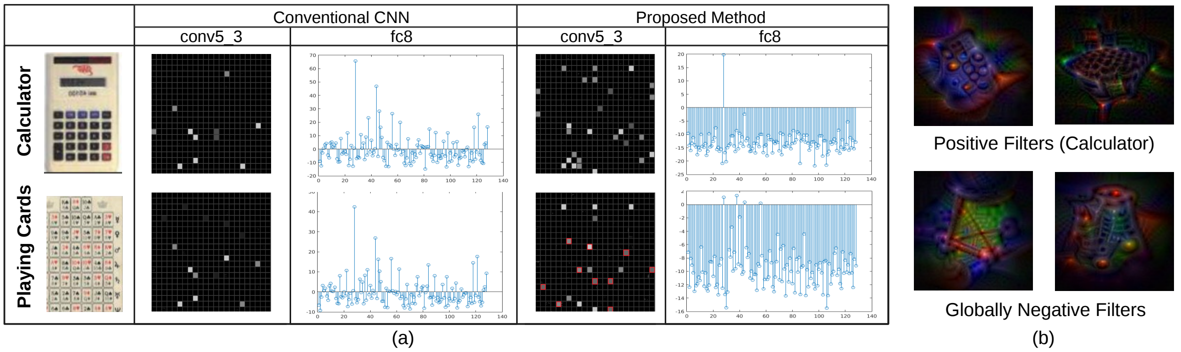


Figure 3. (a) Activations of known (Calculator) and unknown samples (Playing Cards) in a VGG16 model. In conventional CNN, both known and unknown samples activates similar conv5-3 filters and results in a similar fc8 activation map. Novelty detection of the novel sample fails due to high activation scores present in fc8 layer. In the proposed method, Calculator object activates filters related to Calculators whereas top activated filters in Playing Cards is unrelated to known classes (globally negative). Since all activations in fc8 are very small for the Playing Cards object, it can be detected as a novel sample by thresholding. (b) Top positive filters and top globally negative filters of the calculator class.

Firstly, even a low activation of the ground truth class could yield a low cross-entropy provided that the activations of all other (non-matching) classes are very low. As a result, lower score values may not get heavily penalized during training. Therefore, a model trained using the cross-entropy loss may end up producing low activation scores for known classes during inference. In closed set classification, this behavior will not cause complications as long as the correct class records the highest score. However, in threshold-based novelty detection, this poses a problem as having low scores for the positive class will result in false negatives. Secondly, the cross-entropy loss does not necessarily drive activations of unrelated classes below zero. As a result, inaccurate cross-class relationships are encouraged during training.

In order to illustrate this point, we trained a VGG16 [26] based CNN classification network using the first 128 classes of the Caltech256 dataset. For the considered example, the Calculator class (indexed at 27) is a known class and the Playing Cards class (indexed at 163) is a novel class. Shown in Figure 3 are the activations of conv5-3 and fc8 layers of the network for two inputs of the two classes. As can be seen from this figure, when the network observes a calculator object (known object), it correctly associates the highest score in **f** to the correct class (class 27). However, there is also a significant miss-association between the calculator class and coin (class 43), keyboard (class 45), dice (class 55) and joystick classes (class 120).

4.2. Membership Loss

In our approach, we first independently translate each activation score value f_i into the range $0 - 1$ using the sigmoid(σ) function. We interpret each transformed activation score as the probability of the input image belonging to each individual class. If the ground truth label of a given observation x is y, we aim at learning a function that produces absolute probabilities for the membership of each class as follows

$$\mathbb{P}(y = i) = \sigma(f(x)_i) \;\; \forall i \in \{1, 2, \dots c\}. \tag{1}$$

Ideally, the learned transformation will produce $\sigma(f(x)_i) = 1$ for $i = y$ and $\sigma(f(x)_i) = 0$, otherwise. We denote the risk of associating a higher score with a wrong class ($\sigma(f(x)_i) = 1$ for $i \neq y$) as R_{W1} and risk of associating a low score with the correct class ($\sigma(f(x)_i) = 0$ for $i = y$) as R_{C0}. We define the *membership loss* L_M as the risk of classification as

$$L_M(x, y) = R_{C0}(x, y) + R_{W1}(x, y), \tag{2}$$

where λ is a positive scalar. With our formulation, we define $R_{W1}(x, y) = [1 - \mathbb{P}(y = 1)]^2 = [1 - \sigma(f(x)_y)]^2$. Here, the quadratic term is introduced to impose a heavy penalty on very high deviations. Similarly, $R_{C0}(x, y)$ becomes,

$$\begin{aligned} R_{C0}(x, y) &= \frac{1}{c-1} \sum_{i=1, i\neq y}^{c} [\mathbb{P}(i = 1)]^2 \\ &= \frac{1}{c-1} \sum_{i=1, i\neq y}^{c} [\sigma(f(x)_i)]^2. \end{aligned}$$

By substitution, we get

$$L_M(x, y) = \lambda[1 - \sigma(f(x)_y)]^2 + \frac{1}{c-1} \sum_{i=1, i\neq y}^{c} [\sigma(f(x)_i)]^2. \tag{3}$$

Here, the parameter λ controls relative weight given to each risk source. In our experiments, we set $\lambda = 5$. Taking the partial derivative of the membership loss yields the following back-propagation formula

$$\frac{\partial L_M(x, y)}{\partial f(x)_i} = \begin{cases} -2\lambda[1 - \sigma(f(x)_i)] \times \sigma(f(x)_i)' & \text{for } i = y \\ \frac{2}{c-1}\sigma(f(x)_i) \times \sigma(f(x)_i)' & \text{for } i \neq y, \end{cases}$$

where, $\sigma(f(x)_i)' = \sigma(f(x)_i)(1 - \sigma(f(x)_i))$.

The proposed *membership loss* does not operate on the closed-set assumption. It takes individual score values into account in an absolute sense. Therefore, when the membership loss is used, known samples that produce small activations will be penalized regardless of the score values of the other classes. When the membership loss is used together with the cross-entropy loss, the network learns a representation that produces relatively higher activation scores for the correct class. For example, consider the $fc8$ activation map of the proposed method for the Calculator object input shown in Figure 3. There, we observe that the correct class (indexed at 27) produces a large positive score whereas all other classes produce negative scores.

4.3. Globally Negative Filters

When a conventional classification network is used, novel images are often able to produce very high activation scores there by leading to false positive detections. Such an example is shown in Figure 3(bottom) where a Playing Cards instance has produced a very high activation score in the index corresponding to the Calculator class (indexed at 27). Final activation score of a class is generated based on the responses of the positive and negative filters as discussed in Section 3. Once the network is trained, given an input of a particular known class, the input stimulates some *positive filters* and *negative filters* associated with the class. If the model is well trained, the response of the positive filters exceeds the response of the negative filters to produce a high positive activation score.

Given this background, it is interesting to investigate how a novel sample is able to produce a high activation score. Let us revisit activations of Playing Cards image (novel image) shown in Figure 3 (bottom). In this example, Playing Cards image has stimulated some positive filters of the Calculator class despite the differences in content. At the same time, by chance, it has not produced sufficient stimulation in negative filters of the Calculator class, thereby producing a large positive activation in $\mathbf{f}$. This can be clearly observed in Figure 3 where both the Calculator and the Playing Cards images have activated similar filters in the conv5-3 layer.

To this end, we make the following proposal. We wish to learn a set of filters that are stimulated generally by natural images and produce evidence against all known classes. In other words, these filters are *negative filters* with respect to all known classes - hence we call them *globally negative filters*. If any of such filters are stimulated during inference, it would prove greater evidence that the observed image is novel. However, this proposal will succeed only if the *globally negative filters* are stimulated by arbitrary images outside the known class set.

In order to learn the *globally negative filters*, we propose a joint-learning network structure. In addition to the known object dataset, we use the *out-of-distribution* data samples in training. For the remainder of the paper we refer the *out-of-distribution* dataset as the *reference dataset*. We learn features that can perform classification in both the known dataset and the reference dataset. If the *reference dataset* has $\mathcal{C}$ classes, once trained, the filter bank will contain positive filters of all $c + \mathcal{C}$ classes. Filters associated with the reference dataset will likely act as *negative filters* for all classes in the known dataset, thereby be globally negative. In this framework, the *globally negative filters* are likely to respond to arbitrary natural images provided that the reference dataset is a large-scale diverse dataset.

In Figure 3, we show the impact of using the *globally negative filters*. Visualization of top activated filters for the Calculator class are shown at the top in Figure 3(b). As can be seen from this figure, these filters are positively corelated with the Calculator class. With the new formulation, we observe that playing cards object activates some extra filters which are not in common with the calculator class (highlighted in red). At the bottom of Figure 3(b) we visualize filters with the highest activation for the Playing Cards object. By inspection, these two visualizations look arbitrary and do not have an obvious association with any of the Caltech256 classes. We interpret these filters as instances of the *globally negative filters*. Due to the availability of more negative evidence, the overall activation value of the playing cards object has been drastically reduced.

4.4. Training Procedure

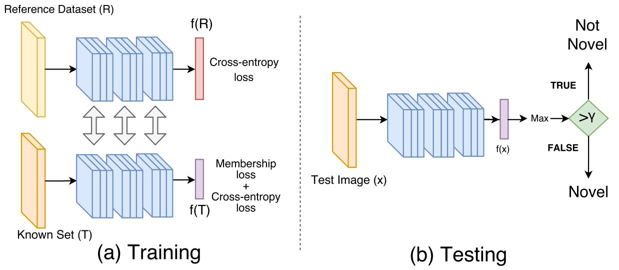
We propose a network architecture and a training mechanism to ensure that the network learns the *globally negative filters*. For this purpose, we use an external multi-class labeled dataset which we refer to as the *reference dataset*.

We first select a CNN backbone of choice (this could be a simple network such as Alexnet [12] or a very deep/complex structure such as DenseNet [11]). Two parallel CNN networks of the selected backbone are used for training as shown in Figure 4(a). The only difference between the two parallel networks is the final fully-connected layer where the number of outputs is equal to the number of classes present in either dataset. For the purpose of our discussion, we refer the sub-network up to the penultimate layer of the CNN as the feature extraction sub-network.

Initially, weights of the two feature extraction subnetworks are initialized with identical weights and they are kept identical during training. Weights are not shared between the final layer of two parallel networks. During training, two mini batches from two datasets (reference dataset (R) and known classes (T)) are considered and they are fed into the two branches independently. We calculate the cross-entropy loss (L_{ce}) with respect to the samples of the reference dataset and both the membership loss (L_m) and the cross-entropy loss with respect to the samples of known classes. The cumulative loss of the network then becomes a linear combination of the two losses as follows,

$$CumulativeLoss = L_{ce}(R) + \alpha_1 \, L_{ce}(T) + \alpha_2 \, L_m(T). \tag{4}$$

In our experiments, we keep $\alpha_1, \alpha_2 = 1$. The cumulative loss is back-propagated to learn the weights of the two CNN branches. Reducing membership loss and cross-entropy loss with respect to the known-class dataset increases the potential of performing novelty detection in addition to classification as discussed in the preceding sub-section. On the other hand, having good performance (low cross-entropy loss) in the reference dataset suggests the existence of filters that are responsive to generic objects provided that the reference dataset is sufficiently diverse. When classes appearing in the reference dataset do not intersect with known classes, these filters serve as the *globally negative filters*.



(a) Training (b) Testing

Figure 4. Proposed architecture for novelty detection. We use an external multi-class dataset (reference dataset (R)) in addition to the known object dataset (T). Two parallel CNN networks with identical structure and weights are used to extract features from both datasets. We train separate classifier networks operating on the same feature to perform classification in either dataset. During inference, novelty detection is performed by thresholding the maximal activation of the bottom branch of the network.

4.5. Testing (Novelty Detection)

During inference, we propose to use the setup shown in Figure 4(b) where we only consider the bottom CNN branch of the training network. Given a test image x, we perform a forward pass using the learned CNN network to obtain the final feature $\mathbf{f}(x)$. The largest element of $\sigma(\mathbf{f}(x))$, $\max \sigma(\mathbf{f}(x))$ is thresholded using a predetermined threshold γ to arrive at the identity of the test image. If the yielded score is below the threshold γ, we identify the test sample to be novel. In a practical system, threshold γ is chosen considering the percentile of the matched score distribution (for example threshold can be chosen to be 95th percentile if the accepted false negative rate is 5%) . In addition to the novelty detection procedure, the same network structure can be used to perform classification as well. Here, $\arg\max \sigma(\mathbf{f}(x))$ yields the identity of the predicted class for the test sample x. We note that this step is identical to the classification procedure used in the standard CNN-based classification.

5. Experimental Setup and Results

In this section, we present experimental results for the novelty detection task. We first describe the baseline methods used for comparison. Then, we introduce the four datasets used for evaluation. Finally, we discuss the obtained results followed by the analysis of the proposed method.

5.1. Baseline Methods

We evaluate the proposed method on four novelty detection databases and we compare its performance with the standard novelty detection schemes. We use the following baseline comparisons based on the AlexNet [12] and the VGG16 [26] features fine-tuned on the given dataset.

1. Finetune [26]: $fc8$ feature scores of the trained deep model are thresholded to detect novel samples.

2. One-class SVM [23]: A one-class SVM classifier is trained for all known classes. The maximum SVM score is considered during the inference.

3. KNFST [4], [13]: Deep features are normalized and histogram intersection kernel method is used to generate inner products between the samples.

4. Local KNFST [3]: Deep features with histogram intersection kernel is considered with 600 local regions.

5. OpenMax [2]: Activations of penultimate layer of a deep model are used to construct a single channel class-wise mean activation vectors (MAV) and the corresponding Weibull distributions.

6. K-extremes [24]: Mean activations of the VGG16 $fc7$ features are considered for each class and top 0.1 activation indexes are binarized to arrive at the Extreme Value Signatures.

7. Finetune$(c{+}\mathcal{C})$: A $(c{+}\mathcal{C})$ class CNN is trained by treating classes of the reference dataset as the additional class.

In addition, we evaluate the performance based on the pre-trained deep features (trained on the ILSVRC12 database) for KNFST and local KNFST methods. Whenever pre-trained features are use they are denoted by the suffix *pre*.

5.2. Datasets

We use four publicly available multi-class datasets to evaluate the novelty detection performance of the proposed method.

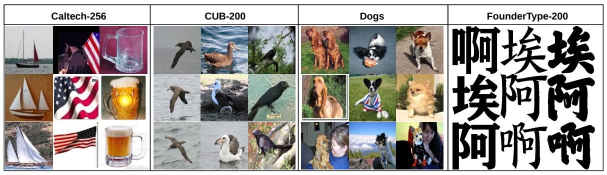


Figure 5. Sample images from the evaluation datasets. Each column contains images taken from a single class of each dataset.

Caltech256 Dataset. The Caltech256 dataset is a fully annotated dataset which consists of 30607 images from 256 object classes. Following the protocol presented in [13], we first sorted the class names alphabetically and picked the first 128 classes as the known classes and considered the images from the remaining 128 classes as the novel images.

Caltech-UCSD Birds 200 (CUB 200) Dataset. The CUB-200 dataset includes 6033 images belonging to 200 distinct bird categories. Ground truth labels for each image are provided. In our experiment, we sorted names of the bird categories alphabetically and used the first 100 classes as the known classes. The remaining classes were used to represent novel images.

Stanford Dogs Dataset. This dataset is a subset of the ImageNet dataset and was originally intended for fine-grain classification. There are 20580 images belonging to 120 different dog breeds in this dataset. We considered the first 60 classes as the known classes and treated the remaining classes as the novel classes during performance evaluation.

FounderType-200 Dataset. This dataset is a collection of Chinese character images in different font types. The dataset is organized based on the font-type. In total there are 200 different font-types with 6763 images from each class in this dataset. Following the same convention as before, we picked the first 100 classes to represent the enrolled classes. The remaining 100 classes were used to simulate the novel images.

In all datasets, following the protocol in [13], images of the enrolled classes were randomly split into two even sets to form training and testing datasets of the enrolled classes. Images of the novel classes were used only during testing. When finetuning/extracting features from the caltech256 dataset following [6], we used the pretrained model trained on the Places365 dataset [32]. For all other tasks, we used the pretrained model trained on the ILSVRC12 dataset. Accordingly, the validation sets of Places365 was used as the reference dataset for Caltech256. For all other tasks the validation set of ILSVRC12 was considered.

5.3. Results

We evaluated all methods based on the VGG16 and the AlexNet features. We used the training codes provided by the authors when evaluating the KNFST [4] and the local KNFST [3] methods. Performance of each method is evaluated using the area under the receiver operating characteristics (AUC) curve. Obtained AUC values for each method are tabulated in Table 1 for all datasets[1].

Table 1. Novelty detection results (AUC of the ROC curve) on the evaluation datasets. The best performing method for each dataset is shown in bold. Second best method is shown in italics.

Method	Caltech-256		CUB-200		Dogs		FounderType	
	VGG16	AlexNet	VGG16	AlexNet	VGG16	AlexNet	VGG16	AlexNet
Finetune[26], [12]	0.827	0.785	0.931	0.909	0.766	0.702	0.841	0.650
One-class SVM[23]	0.576	0.561	0.554	0.532	0.542	0.520	0.627	0.612
KNFST pre[4]	0.727	0.672	0.842	0.710	0.649	0.619	0.590	0.655
KNFST[4], [13]	0.743	0.688	0.891	0.748	0.633	0.602	*0.870*	0.678
Local KNFST pre[3]	0.657	0.600	0.780	0.717	0.652	0.589	0.549	0.523
Local KNFST[3]	0.712	0.628	0.820	0.690	0.626	0.600	0.673	0.633
K-extremes[24]	0.546	0.521	0.520	0.514	0.610	0.592	0.557	0.512
OpenMax[2]	0.831	0.787	*0.935*	*0.915*	0.776	*0.711*	0.852	0.667
Finetune$(c + \mathcal{C})$	*0.848*	*0.788*	0.921	0.899	*0.780*	0.692	0.754	*0.723*
Deep Novelty (ours)	**0.869**	**0.807**	**0.958**	**0.947**	**0.825**	**0.748**	**0.893**	**0.741**

When baseline methods are considered, a variance in performance can be observed across datasets. In general, K-extremes has reported below-par performances compared to the other methods. When the number of enrolled classes are very high, the mean activation signature of a class looses its uniqueness. This is why K-extremes method fails when very large number of classes are enrolled as suggested in [24]. In the Caltech-256 and CUB-200 datasets, thresholding deep activations and OpenMax has yielded better results among the baseline methods. In Caltech256, this has improved marginally when the reference dataset (ILSVRC12) is incorporated. This method has performed reasonably well in the FounderType-200 dataset but it's performance in the Standford Dogs dataset is not convincing. In general, KNFST has out-performed local KNFST except for in the Standford Dogs dataset. KNFST (and local KNFST) operating on the finetuned deep features have performed better in general compared to the pre-trained deep features. This trend has changed only in the Standford Dogs dataset. Here we note that none of the baseline methods have yielded consistent performance across datasets.

In comparison, the proposed method is able to produce the best performance across all datasets. When AlexNet is used as the back-bone network, there is an improvement of about 3.0% over the baselines in the CUB-200 and Standford Dogs datasets. In the other two datasets this margin is 2.0%. In the Caltech256, CUB-200 and FounderType-200 datasets, the improvements in AUC are in excess of 2.0% for the VGG16 model. In the Standford Dogs dataset, the proposed method is able to introduce a significant advancement of more than 7.0% in AUC compared with the baseline methods. In general, we note that in datasets where the baseline performance is already very good, as in the CUB-200 and FounderType 200 datasets, the improvement of the proposed method is relatively small. On the other hand, when the baseline performance is poor, the proposed method is able to generate a significant improvement in the performance.

5.4. Ablation Study

In this subsection, we investigate the impact of each individual component of the proposed framework. For the purpose of the ablation study, we use the validation dataset of the ILSVRC12 dataset as the reference dataset. It should be noted that figures reported in this subsection are different from Table 1 due to this reason. Starting from the traditional CNN architecture, we added one component of the

[1] Source code of the proposed method is made available at https://github.com/PramuPerera/TransferLearningNovelty

proposed framework at a time and evaluated the novelty detection performance on the Caltech-256 dataset as a case study. Testing protocol presented in the preceding subsection was followed in all cases. Considered cases are as follows.

a) Single CNN with the cross-entropy loss (AUC 0.854). This is the CNN baseline where a CNN is trained using the enrolled classes conventionally.

b) Single CNN with the cross-entropy loss+membership loss (AUC 0.865). The network architecture is the same as in case (a). In addition to the cross-entropy loss, the membership loss is calculated with respect to the enrolled dataset.

c) Two Parallel CNNs with cross-entropy loss (AUC 0.864). The network structure proposed in Figure 4(a) is used. In contrast, only the cross-entropy loss is used in the bottom sub-network.

d) Proposed method (AUC 0.906). Proposed structure Figure 4(a) is used for training.

In the proposed method, we introduced membership loss and a parallel network structure as contributions. From the case study conducted, it appears that the novelty detection performance improves compared to the baseline even when one of the contributions are used. Moreover, we observe that the two contributions compliment each other and generate even better results when combined together.

5.5. Impact of the Reference Dataset

In the proposed method, we assumed the availability of a reference dataset with large number of classes. In this subsection, we investigate the impact of the reference dataset by varying the reference dataset of choice. In particular, we use the ILSVRC12, Caltech-256 and Standford Dogs datasets as the reference datasets to perform novelty detection using the proposed method in the CUB-200 dataset. Results obtained are tabulated in Table 2. Here we have included the performance of the best baseline method for the CUB-200 dataset (Finetune) from Table 1 as a baseline.

Compared to ILSVRC12, when Caltech-256 is used as the reference dataset, AUC drops by 0.005%. This further drops by 0.008% when the Standford Dogs dataset is used. The ILSVRC12 dataset contains 1000 image classes and has significant variance in images within each class. Caltech-256 is a similar multi-class dataset but with fewer classes. Both of these datasets contain natural images. However since ILSVRC12 has more classes and more intra-class variance, we expect it to generate *globally negative filters* better. Therefore, the performance drop of Caltech-256 compared to ILSVRC12 is expected. On the other hand, the Standford Dogs dataset only contains images of dogs. Therefore, filters learned using this dataset may not be generic to get stimulated by the arbitrary inputs. Therefore, the drop in the performance is justified. In conclusion, we note that the proposed method is able to out-perform baseline novelty detection methods even when the reference dataset is varied. However, better results are obtained when a larger dataset with high degree of intra-class variation is used as the reference dataset.

Table 2. Impact of the reference dataset used. Results of the case study conducted on the CUB-200 dataset by varying the reference dataset.

	Baseline	ILSVRC12	Caltech-256	Dogs
Novelty Detection AUC	0.931	**0.958**	0.953	0.945

5.6. Impact on Classification Accuracy

When a test image is present, the proposed method produces a set of class activation scores. It is still possible to perform classification using the same system by associating the test image with the class containing the highest activation. In what follows, we consider test samples of the known classes and perform closed-set classification in the same experimental setup described in Section 5.3. In other words, we do not consider novel samples for the purpose of this study. Obtained classification accuracies for the four datasets are tabulated in Table 3. Although the proposed method is designed for the purpose of novelty detection, we note that the proposed changes have contributed towards increasing the classification accuracy of the system as well. This is because the *membership loss* explicitly enforces correct class to have a high score and all other classes to have scores closer to zero.

Table 3. Classification accuracy obtained for conventional fine-tuning and the proposed method for the four evaluation datasets.

	Caltech-256	CUB-200	Dogs	FounderType
VGG16	0.908	0.988	0.730	0.945
Proposed Method	**0.939**	**0.990**	**0.801**	**0.950**

6. Conclusion

We presented an end-to-end deep learning-based solution for image novelty detection. We build up on the conventional classification networks and introduce two novel contributions; namely, *membership loss* and a training procedure that produces *globally negative filters*. In the proposed method, novelty is quarried simply by thresholding the highest activation of the output vector. We demonstrate the effectiveness of the proposed method on four publicly available multi-class image datasets and obtain state-of-the-art results.

References

[1] M. Abavisani, H. R. Vaezi Joze, and V. Patel. Improving the performance of unimodal dynamic hand-gesture recognition with multimodal training. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[2] A. Bendale and T. E. Boult. Towards open set deep networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 1563–1572, 2016. 2, 6, 7

[3] P. Bodesheim, A. Freytag, E. Rodner, and J. Denzler. Local novelty detection in multi-class recognition problems. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 813–820, 2015. 2, 6, 7

[4] P. Bodesheim, A. Freytag, E. Rodner, M. Kemmler, and J. Denzler. Kernel null space methods for novelty detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013. 2, 6, 7

[5] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, 2009. 2

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 7

[7] A. R. Dhamija, M. Günther, and T. Boult. Reducing network agnostophobia. In *Advances in Neural Information Processing Systems 31*, pages 9157–9168. 2018. 2

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, June 2016. 1, 3

[9] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proceedings of International Conference on Learning Representations*, 2017. 1

[10] H. Hoffmann. Kernel pca for novelty detection. *Pattern Recognition*, 40(3):863 – 874, 2007. 2

[11] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 5

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105, 2012. 1, 5, 6, 7

[13] J. Liu, Z. Lian, Y. Wang, and J. Xiao. Incremental kernel null space discriminant analysis for novelty detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4123–4131, July 2017. 2, 6, 7

[14] M. Markou and S. Singh. Novelty detection: a review – part 1: statistical approaches. *Signal Processing*, 83(12):2481 – 2497, 2003. 1

[15] P. Oza and V. Patel. C2ae: Class conditioned auto-encoder for open-set recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[16] P. Oza and V. M. Patel. Active authentication using an autoencoder regularized cnn-based one-class classifier. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*. IEEE, 2019. 2

[17] P. Oza and V. M. Patel. One-class convolutional neural network. *IEEE Signal Processing Letters*, 26(2):277–281, 2019. 2

[18] P. Perera, R. Nallapati, and B. Xiang. Ocgan: One-class novelty detection using gans with constrained latent representations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[19] P. Perera and V. M. Patel. Learning Deep Features for One-Class Classification. *ArXiv e-prints*. 2

[20] P. Perera and V. M. Patel. Dual-minimax prob-ability machines for one-class mobile active authentication. In *IEEE Conference on Biometrics: Theory, Applications,and Systems (BTAS)*, September 2018. 2

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision*, 115(3):211–252, Dec. 2015. 1, 3

[22] W. J. Scheirer, A. Rocha, A. Sapkota, and T. E. Boult. Towards open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 36, July 2013. 2

[23] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13(7):1443–1471, 2001. 2, 6, 7

[24] A. Schultheiss, C. Käding, A. Freytag, and J. Denzler. Finding the unknown: Novelty detection with extreme value signatures of deep neural activations. In *Pattern Recognition - 39th German Conference, Proceedings*, pages 226–238, 2017. 2, 6, 7

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 1

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2014. 4, 6, 7

[27] D. M. J. Tax and R. P. W. Duin. Support vector data description. *Mach. Learn.*, 54(1):45–66, 2004. 2

[28] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991. 2

[29] A. Vyas, N. Jammalamadaka, X. Zhu, D. Das, B. Kaul, and T. L. Willke. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VIII*, pages 560–574, 2018. 1

[30] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(2):210–227, Feb. 2009. 2

[31] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. In *Deep Learning Workshop, International Conference on Machine Learning (ICML)*, 2015. 3

[32] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 7