3.1. Semi-supervised Anomaly Detection

DN2 takes a set of input images $X_{train} = x_1, x_2..x_N$. In the semi-supervised setting we assume that all input images are normal. DN2 uses a pre-trained feature extractor F to extract features from the entire training set:

$$f_i = F(x_i) \tag{1}$$

In this paper, we use a ResNet feature extractor that was pretrained on the Imagenet dataset. At first sight it might appear that this supervision is a strong requirement, however such feature extractors are widely available. We will later show experimentally that the normal or anomalous images do not need to be particularly closely related to Imagenet.

The training set is now summarized as a set of embeddings $F_{train} = f_1, f_2..f_N$. After the initial stage, the embeddings can be stored, amortizing the inference of the training set.

To infer if a new sample y is anomalous, we first extract its feature embedding: $f_y = F(y)$. We then compute its kNN distance and use it as the anomaly score:

$$d(y) = \frac{1}{k} \sum_{f \in N_k(f_y)} \|f - f_y\|^2 \tag{2}$$

$N_k(f_y)$ denotes the k nearest embeddings to f_y in the training set F_{train}. We elected to use the euclidean distance, which often achieves strong results on features extracted by deep networks, but other distance measures can be used in a similar way. By verifying if the distance $d(y)$ is larger than a threshold, we determine if an image y is normal or anomalous.

3.2. Unsupervised Anomaly Detection

In the fully-unsupervised case, we can no longer assume that all input images are normal, instead, we assume that only a small proportion of input images are anomalous. To deal with this more difficult setting (and inline with previous works on unsupervised anomaly detection), we propose to first conduct a cleaning stage on the input images. After the feature extraction stage, we compute the kNN distance between each input image and the rest of the input images. Assuming that anomalous images lie in low density regions, we remove a fraction of the images with the largest kNN distances. This fraction should be chosen such that it is larger than the estimated proportion of anomalous input images. It will be later shown in our experiments that DN2 requires very few training images. We can therefore be very aggressive in the percentage of removed image, and keep only the images most likely to be normal (in practice we remove 50% of training images). After removal of the suspected anomalous input images, the images are now assumed to have a very high-proportion of normal images. We can therefore proceed exactly as in the semi-supervised case.

3.3. Group Image Anomaly Detection

Group anomaly detection tackles the setting where the input sample consists of a set of images. The particular combination is important, but not the order. It is possible that each image in the set will individually be normal but the set as a whole will be anomalous. As an example, let us assume normal sets consisting of M images, a randomly sampled image from each class. If we trained a point (per-image) anomaly detector, it will be able to detect anomalous sets containing pointwise anomalous images e.g. images taken from classes not seen in training. An anomalous set containing multiple images from one seen class, and no images from another will however be classified as normal as all images are individually normal. Previously, several deep autoencoder methods were proposed (e.g. DOro et al. (2019)) to tackle group anomaly detection in images. Such methods suffer from multiple drawbacks: i) high sample complexity ii) sensitivity to reconstruction metric iii) potential lack of sensitivity to the groups. We propose an effective kNN based approach. The proposed method embeds the set by orderless-pooling (we chose averaging) over all the features of the images in the set:

1. Feature extraction from all images in the group g, $f_g^i = F(x_g^i)$
2. Orderless pooling of features across the group: $f_g = \frac{\sum_i f_g^i}{number\ of\ images}$

Having extracted the group feature described above we proceed to detect anomalies using DN2.

4. Experiments

In this section, we present extensive experiments showing that the simple kNN approach described above achieves better than state-of-the-art performance. The conclusions generalize across tasks and datasets. We extend this method to be more robust to noise, making it applicable to the unsupervised setting. We further extend this method to be effective for group anomaly detection.

4.1. Unimodal Anomaly Detection

The most common setting for evaluating anomaly detection methods is unimodal. In this setting, a classification dataset is adapted by designating one class as normal, while the other classes as anomalies. The normal training set is used to train the method, all the test data are used to evaluate the inference performance of the method. In line with previous works, we report the ROC area under the curve (ROCAUC).