



Figure 5. Number of training images vs. ROCAUC (left) CIFAR10 - Strong performance is achieved by DN2 even from 10 images, whereas Geometric deteriorates critically. (center) FashionMNIST - similarly strong performance by DN2. (right) Impurity ratio vs ROCAUC on CIFAR10. The training set cleaning procedure, significantly improves performance.

the performance degradation as percentage of impurities.

4.5. Group Anomaly Detection

To compare to existing baselines, we first tested our method on the task in [DOro et al. \(2019\)](#). The data consists of normal sets containing 10 – 50 MNIST images of the same digit, and anomalous sets containing 10 – 50 images of different digits. By simply computing the trace-diagonal of the covariance matrix of the per-image ResNet features in each set of images, we achieved 0.92 ROCAUC vs. 0.81 in the previous paper (without using the training set at all).

As a harder task for group anomaly detection in unordered image sets, we designate the normal class as sets consisting of exactly one image from each of the M CIFAR10 classes (specifically the classes with ID $0..M - 1$) while each anomalous set consisted of M images selected randomly among the same classes (some classes had more than one image and some had zero). As a simple baseline, we report the average ROCAUC (Fig. 4.2) for anomaly detection using DN2 on the concatenated features of each individual image in the set. As expected, this baseline works well for small values of M where we have enough examples of all possible permutations of the class ordering, but as M grows larger ($M > 3$), its performance decreases, as the number permutations grows exponentially. We compare this method, with 1000 image sets for training, to nearest neighbours of the orderless max-pooled and average-pooled features, and see that mean-pooling significantly outperforms the baseline for large values of M . While we may improve the performance of the concatenated features by augmenting the dataset with all possible orderings of the training sets, it will grow exponentially for a non-trivial number of M making it an ineffective approach.

4.6. Implementation

In all instances of DN2, we first resize the input image to 256×256 , we take the center crop of size 224×224 , and

Table 6. Accuracy on CIFAR10 using K-means approximations and full kNN (ROCAUC %)

C=1	C=3	C=5	C=10	kNN
91.94	92.00	91.87	91.64	92.52

using an Imagenet pre-trained ResNet (101 layers unless otherwise specified) extract the features just after the global pooling layer. This feature is the image embedding.

5. Analysis

In this section, we perform an analysis of DN2, both by comparing kNN to other classification methods, as well as comparing the features extracted by the pretrained networks vs. features learned by self-supervised methods.

5.1. kNN vs. one-class classification

In our experiments, we found that kNN achieved very strong performance for anomaly detection tasks. Let us try to gain a better understanding of the reasons for the strong performance. In Fig. 6 we can observe t-SNE plots of the test set features of CIFAR10. The normal class is colored in yellow while the anomalous data is marked in blue. It is clear that the pre-trained features embed images from the same class into a fairly compact region. We therefore expect the density of normal training images to be much higher around normal test images than around anomalous test images. This is responsible for the success of kNN methods.

kNN has linear complexity in the number of training data samples. Methods such as One-Class SVM or SVDD attempt to learn a single hypersphere, and use the distance to the center of the hypersphere as a measure of anomaly. In this case the inference runtime is constant in the size of the training set, rather than linear as in the kNN case. The drawback is the typical lower performance. Another popular way ([Fukunaga & Narendra, 1975](#)) of decreasing the inference