Deep Committee kNN for Anomaly Detection

Hadar Sharvit
Department of Computer Science
Hebrew University
Jerusalem
hadar.sharvit1@mail.huji.ac.il

Abstract

Over the last years, techniques for anomaly detection that utilize advanced and complex mechanisms was shown to gradually improve classification, detection and segmentation benchmarks. Simultaneously, incorporating classical machine learning approaches such as nearest neighbours also seemed to have provided similar if not superior results throughout. In this work, we investigate whether these results could be further extended by using the principle of committee classification, that is based on decision making provided from various activation layers of a pre-trained ImageNet network. In out results, we show that such mechanisms has the potential to enhance the model's capabilities while also remaining rather simple, compared to other approaches. We support our results with a publicly documented repository[1]

1 Introduction

Anomaly detection is learning task in which we aim to label a data point as either anomalous or regular (non-anomalous). Under the scope of machine learning research, it is common to consider three anomaly detection frameworks. The first framework is supervised anomaly detection, in which our data is labeled as either regular of anomalous. Such framework, which can be thought of as a classification task, is usually not as common, mostly due to the fact that in most cases we are not given with a set of anomalous examples, as they differ from one another depending on the setup of our problem, and we would not want to restrict ourselves to only a group of anomalies. The second framework is unsupervised anomaly detection, in which our training data consists of both regular and anomalous samples, yet without the corresponding labels. In this framework it is usually assumed that there is imbalance between those samples, or more specifically, that there are more regular samples than anomalous ones. The final framework is known as semi-supervised anomaly detection, in which our training data consists of only regular data, without anomalies - this framework is a common one, as it resembles real-life scenarios in which we only have regular samples at our disposal. In this paper we will deal with the problem of semi-supervised anomaly detection, mostly under the umbrella of unimodal classification task (also known as one-vs-other). Using this approach means we use data-sets from the world of classification, and treat one class as regular, while the others are considered anomalous. This formalism does perform worse compared to other-vs-one, but regardless represents the case of anomalies with more resemblance to real life, as in most cases we are dealing with variety of anomalies, but not so many varieties of non-anomalous instances.

In our work, we aim to combine the already verified method of extracting deep activations followed by a k-nearest-neighbours, as was demonstrated for example in (Bergman et al. [2020]), with the basic concept of committee based classification, where the committee in hand takes into account not only the last activation, but rather considers many activation that are not as deep. More specifically, using

[1] please refer to our Deep-Committee-kNN repository

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

a pre-trained ImageNet based neural network, we will extract more than one embedding for every sample in our train and test data. Then, we will measure the distance of the test data embedding from the train embeddings and use a committee to generate a final classification, using a pre-determined distance threshold for every different embedding.

2 Previous Work

Prior to deep learning methods, a paradigm that was commonly used was to estimate the density function $p(x)$ of the normal data. By doing so, one could take a test sample y and classify it as anomalous if $p(y)$ is smaller then some threshold. In some cases, few assumption were made on the normal data, as for example that it could be represented as a mixture of Gaussian distributions, hence one could compute the multivariate mean and covariance and use the Mahalanobis distance as anomaly score (Roth [2006]). Furthermore, non-parametric methods including kNN (Eskin et al. [2002]) and Kernel Density Estimation (Latecki et al. [2007]) were widely used (and some still are), as those usually mean fast or even zero training time, with the exclusion of some pre-processing that is done on the data. In the realm of parametric based approaches, Isolation-Forest (Liu et al. [2008]) made great impact by utilizing the concept of random forests, in which the main concept was to choose a random axis and random threshold for every tree, split the samples based on those choices and increase an anomaly score if some sample has turned isolated by this splitting. Another paradigm expanded the concept of poor Out-of-Distribution generalization - in this approach, a task with easily generated labels was defined, and a model would be fitted on this data that contained no anomalies. During test time, if the calculated loss was lower then some threshold, the sample was considered normal. This concept stems from the idea that as our model is fitted to regular data only, the loss score would be very high to anomalous (OOD) samples. Some of the more common approaches that used this paradigm are PCA (Abdi and Williams [2010]), in which an attempt of disentanglement is made to find the most important features that a normal sample is made up of, and One-Class SVM (Schölkopf et al. [1999]) which tries to fit a large percentage of the normal data into a sphere with minimal volume. As disentanglement is a very difficult task on its own, and as the minimal volume sphere relies deeply on the feature space, those methods usually did not achieve great results on their own, and some pre-processing was required. For example, in the case of One-Class SVM, the best feature space representation was learned in an iterative process, and only afterwards an attempt to calculate a minimum volume sphere was conducted.

Diverging from classical approaches, the very early ideas that used a deep neural network usually revolved around the concept of taking a classic approach and improving it with a neural net. For example, using the reconstruction loss of a deep auto-encoder (An and Cho [2015]) was similar in concept the PCA low dimension representation, and DeepSVDD (Ruff et al. [2018]) was similar to one class SVM with the improvement of using a neural network to find the best set of features representation. Another early idea was AnoGAN (Schlegl et al. [2017]) that used a discriminator of a GAN to detect anomaly. Some newer approaches that were based on self-supervised learning tried to learn a meaningful feature representation from normal training data. For example, RotNet (Gidaris et al. [2018]) learned image features by training ConvNets to recognize rotations that were applied to input images. Then, at inference time, if the network can predict the correct rotation, it is assumed that the sample is normal. The concept of RotNet was later expanded with CLI (Tack et al. [2020]), that added a contrastive learning term to the objective.

Additionally, with relation to this article, there are also deep k-nearest-neighbours methods (Papernot and McDaniel [2018]), which are a simple yet powerful extension to classical kNN. deep-kNN methods uses deep activation of normal training images as data-set that are provided from some state-of-the-art pre-trained model. The distance from the k-nearest-neighbors is then used as an anomaly score (Bergman et al. [2020]). Other newer approaches that uses kNN performs fine-tuning on the pre-trained features of the normal dataset using the setup of one class classification (Reiss et al. [2021]).

3 Deep Committee kNN Anomaly Detection

In this section, we will describe our general pipeline for anomaly detection, based on deep features and committee k-nearest-neighbours. We will start with describing a pre-processing scheme, and

continue with specifically showing every step of the algorithm. Our pipeline is also schematically displayed in figure 1

3.1 Pre-processing multiple activations

Let us denote the dataset $\{x_i\}_{i=1}^N$ where $\forall i$, x_i is a "normal" image sample (i.e. not anomalous) as X, and a pre-trained ImageNet based neural net as N_θ. let us also denote the layers of N_θ as $\{\ell_i\}_{i=1}^M$, where ℓ_1 is the input layer and ℓ_M is the output layer. Finally, if a sample $x_j \in X$ is propagated through N_θ, the activation map that is associated with the k'th layer is $\ell_k(\ell_{k-1}(...(\ell_1(x_j))...) := a_k^j$.

Now, given X and N_θ as described above, as well as a pre-determined set of layers $\{\ell_i\}_{i=1}^K$, we construct a new dataset that is comprised of all the activations $\{a_1^j, a_2^j, ..., a_K^j\}_{j=1}^N$, i.e - we propagate every sample x_j from our data through N_θ and save the activations that corresponds to the K layers that we previously chose. We denote this new dataset as $X_a = \bigcup_{i=1}^K X_i$, where $X_i = \{a_i^j\}_{j=1}^N$ are all the activations given from the layer ℓ_i. Moving forward, we will only work with X_a, rather than the original X.

3.2 Deep Committee kNN Pipeline

Once the pre-processing stage is completed, and given a new sample $\hat{x}$, we will classify it as "regular" or "anomalous" based on the following pipeline that we call **D**eep-**C**ommittee-**k**-**N**earest-**N**eighbours (DCkNN):

1. calculate the corresponding activations $\{\hat{a}_1, \hat{a}_2, ..., \hat{a}_K\}$.
2. for every $i \in [K]$:
   (a) Given a predefined distance metric d_i, find the k-nearest neighbours of $\hat{a}_i$ from all other activations of the same layer ℓ_i in the training set X_i , that is, find $\{a_i^{\pi_1}, ..., a_i^{\pi_k}\} \subseteq X_i$ such that $d_i(a_i^{\pi_1}, \hat{a}_i) \leq ... \leq d_i(a_i^{\pi_k}, \hat{a}_i)$, where $\pi_1, ..., \pi_k$ is some permutation on the elements of X_i.
   (b) if the mean distance of $\hat{a_i}$'s neighbours is larger than some predefined threshold t_i, we determine that the i'th classification of $\hat{x}$ is "anomalous". More precisely, denoting the classification as c_i, we have
   $$c_i = \begin{cases} 1, & \frac{1}{k}\sum_{j=1}^{k} d_i(a_i^{\pi_j}, \hat{a}_i) \geq t_i \\ 0, & \text{otherwise} \end{cases}$$
   where 1 is considered "anomalous".
3. given the set of K classifications $\{c_1, c_2, ..., c_K\}$, a majority vote is performed, i.e the final decision c is "anomalous" if and only if more than half of $\{c_1, c_2, ..., c_K\}$ represent the "anomalous" classification. Formally we will write this as
   $$c = \arg\max_{y \in \{0,1\}} \sum_{i=1}^{K} \mathbb{1}\{c_i = y\}$$
   where $\mathbb{1}$ is the indicator function

In terms of the metric d_i that was used, we separate into two distinct cases: for the case of the deepest layer ℓ_K, as the activation was a one dimensional vector, we simply used l_2 norm. For the other layers ($\ell_{k<K}$), the distance was performed on three-dimensional tensors, and l_2 performed poorly. Instead, we choose to use a variation of the Earth Movers Distance that it implemented using the cumulative density function, which was proven as equivalent (Hou et al. [2016]). We further discuss this metric in later sections.

Briefly summarizing the above - in the pre-processing stage we simply generate a set of activations for every "normal" train sample, and in the pipeline that follows we take all the activations of a test sample and generate a set of predictions for every such activation using kNN. We finally take the majority vote of every such prediction to return the final result.

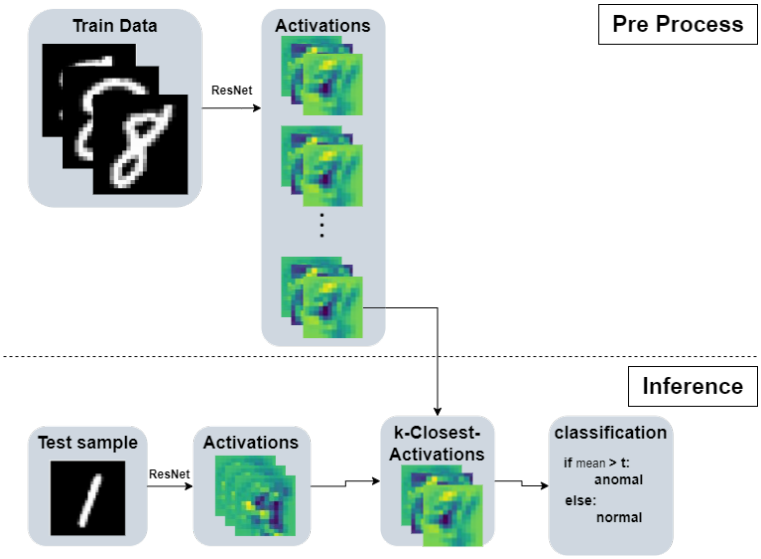

Figure 1: An illustration of the pipeline: In the pre-process step, we save a set of activations for every sample in out dataset. Later, when inferring a new sample - we find the k closest activations to every single one of the activations. Then, we generate K classification and take the mean to generate the last conclusion.

4 Experiments & Discussion

In the following section we display the experiments that were conducted to evaluate the performance of our model. In every such experiments, we will first describe the results in high level. Next, we will compare anomaly detection benchmark scores and discuss the results in more details.

4.1 Unimodal Anomaly detection (One-Vs-Other)

In this setting, we use a classification dataset with N classes and consider one of the classes as "regular" while the other $N-1$ classes are "anomalous". Under the scope of classification data-sets, this setup may be the most related to anomaly detection, as more often than not we are dealing with various types of anomalies, yet only one or so regular instances. We perform this experiment N times such that at the end of our trials, every class was considered "regular" exactly one time. For example, considering the MNIST digits dataset, in which $N = 10$ and the class instances are handwritten digits of all numbers $\{0, 1, 2, ..., 9\}$ - Our first iteration would be where class "0" is regular and the other classes $\{1, 2, ..., 9\}$ are anomalous. We will train our algorithm on class "0" and test our algorithm on all the test data (containing both "regular" and "anomalous" never-seen-before samples). We remind our reader that the MNIST dataset contains of $60k$ images of hand-written digits with size $28 \times 28 \times 1$ (grey scale). Each class $\{0, 1, ..., 9\}$ contains 5000 training samples and 1000 test samples.

In terms of high level results, we have that (1)- a conclusion that is based on many activation layers of a pre-trained network performs better than only considering the last activation, as commonly used, and (2)- using a shallow network and small training dataset still performed better then some other known anomaly detection methods, yet does not outperform some state-of-the-art methods.

Diving deeper into our results, we report the area under the Receiver operating characteristic curve, or ROCAUC for short in 1. The experiments were conducted with respect to a classification based method that only considers last activation (see LLO, as in Last Layer Only, in Table1), and in that aspect it was shows how using a committee could be beneficial to us, as the results were slightly improved, stepping from around 90% to 93% on average. It is also important to state that even though our method could not currently compete with state-of-the-art methods such as Multi-Headed RotNet (Hendrycks et al. [2018]) or DN2 (Bergman et al. [2020]) that provide around 95%ROCAUC, it is still able to outperform kNN that is based on LLO, and it does so while using a very shallow deep-net and minimum training samples, as described in detail in 4.4. Also note that the error values associated

Table 1: Anomaly Detection on MNIST digits (ROCAUC%)

class #	OC-SVM	Deep SVDD	AnoGAN	LLO	Ours
0	98.6±0.0	98.0±0.7	96.6±1.3	95.4	96.3
1	99.5±0.0	99.7±0.1	99.2±0.6	98.3	99.0
2	82.5±0.1	91.7±0.8	85.0±2.9	83.1	88.8
3	88.1±0.0	91.9±1.7	88.7±2.1	87.9	88.4
4	94.9±0.0	94.9±0.8	89.4±1.3	92.3	92.8
5	77.1±0.0	88.5±0.9	88.3±2.9	91.0	93.7
6	96.5±0.0	98.3±0.5	94.7±2.7	89.8	93.0
7	93.7±0.0	94.6±0.9	93.5±1.8	89.4	91.0
8	88.9±0.0	93.9±1.6	84.9±2.1	85.8	91.2
9	93.1±0.0	96.5±0.3	92.4±1.1	85.9	93.1
Avg.	91.3±0.0	94.8± 0.3	91.27±4.1	89.9	92.7

with the results from the literature were computed using a common error aggregation method, where for $f(\{x_i\}) = \frac{1}{n}\sum_{i=1}^{n} x_i$ the error $\Delta f = \sqrt{\sum_i (\frac{\partial f}{\partial x_i}\Delta x_i)^2} = \frac{1}{n}\sqrt{\sum_i \Delta x_i^2}$

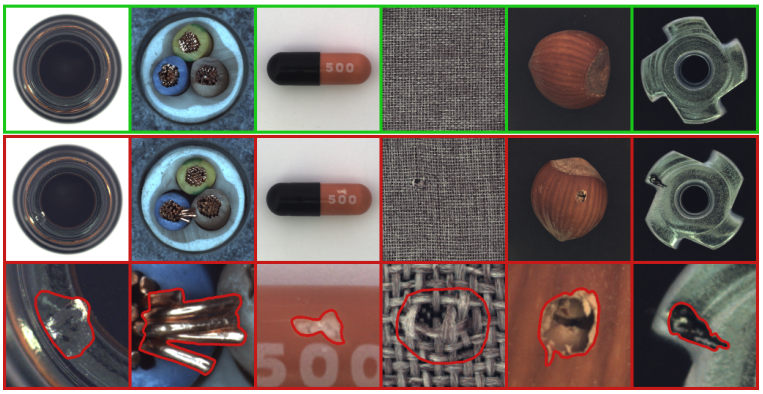

Figure 2: Various samples from the MVTec dataset - the first row (green) is a representation of "normal" samples, the second row is a representation of corresponding "anomalous" samples and the third row is the segmentation of the abnormality. From left to right: Bottle, Cable, Capsule, Carpet, Hazelnut, Screw.

4.2 "Real-life" Anomaly Detection

To account for real-life data, we have also tested our model on the MVTec dataset (Bergmann et al. [2019]), which contains various classes of images, as can be seen in 2. Every class in the MVTec dataset consists of a small set of training samples ("normal" samples), and various kinds of "anomalous" samples. More specifically, every class has 242 train samples and a different number of test samples, ranging from few dozens to few hundreds. The samples themselves have different dimensions, where some are large, at $1024 \times 1024 \times 3$ and some are smaller with $700 \times 700 \times 3$. In our experiments we merged the anomalous samples of every class to one test set that contained various kinds of anomalies. After finishing with this short pre-process, we have continued as was previously described in one-vs-other.

In terms of high level results, we have that (1)- yet again we see that using multiple layers of activations performed better than only considering one of the last activations layers. Furthermore, the improvement was more significant compared to the improvement that was shown in the one-vs-other experiment. (2)- we also show that our method performed better than some of the known anomaly detection algorithms that were tested on MVTec, though did not outperform state-of-the-art methods like SPADE.

Table 2: Anomaly Detection on MVTec (ROCAUC%)

class	AnoGAN	SPADE	LLO	Ours
Carpet	54	98.6	78.7	80.5
Grid	58	99.0	62.3	59.6
Leather	64	99.5	88.9	90.0
Tile	50	89.8	88.0	90.0
Wood	62	95.8	92.0	91.8
Bottle	86	98.1	92.0	93.1
Cable	78	93.2	77.9	81.2
Capsule	84	98.6	79.1	81.2
Hazelnut	87	98.9	81.5	83.2
Metal nut	76	96.9	76.8	78.9
Screw	80	99.5	76.3	78.8
Toothbrush	90	98.9	85.0	93.3
Transistor	80	81.0	65.9	66.2
Zipper	78	98.8	87.8	91.7
Avg.	74	96.2	75.4	82.8

Taking a closer look at the results of our experiment, we can see an improvement of around 7% in terms of the ROCAUC%, which can be accounted to the fact that when considering many activation maps we in fact take into account a set of features with increasing details. Having combined shallow layers that are associated with broad features like edges with deeper layers that identify specific features like shapes and curves, we can identify with increasing success the cases of malfunction in real life data. Having said that, we can also see a decrease in ROCAUC% compared to the first experiment, though this was to be expected as the MNIST dataset is mostly considered a very "simple" dataset, which can be identified rather easily. In that aspect it is also interesting to examine the classes for which the difference between LLO and our method was not large, or even those for which LLO performed better - In those cases it was shown that the shallow layers performed poorly, mostly due to the fact that a reasonable threshold for anomaly classification could not have been identified, as the distances of the anomalous sample from the regular train data were very close to the distances of a regular sample from the train data. The above claim can be further discussed when looking at the scatter plot of mean kNN distances of "regular" sample vs "anomalous" samples in the MNIST dataset (though the same results were visible also for MVTec), as can be seen in figure 3.

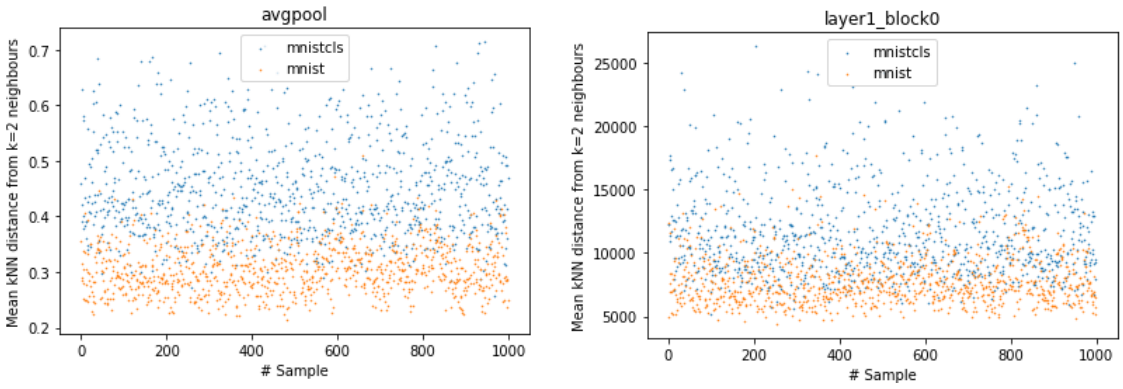

Figure 3: Two plots of the mean distance of every sample in the "normal" test set (orange) and "anomal" test set (blue) from its k nearest neighbours in the train set. Both plots represent the mean distance for when both the "normal" train data and the "normal" test data contains only samples from class#3, yet the "anomal" test data contains all samples that are *not* class#3. Left: deep activation extracted right after the last avgpool layer, Right: shallow activation extracted after the first block of the first layer. We can see that for the case of deep activations, the data is more separable and it is easier to define a threshold that minimizes false positives and false negatives.

4.3 The role of committee

Extensive work has been made on committee methods, with conclusions that are usually separated based on the correlation between the members of the committee - For an uncorrelated ensemble, we know that the probability to classify correctly reaches 1 exponentially fast. Equivalently, the probability that the committee is wrong decays with rate of $O(e^{-T})$, where T is the size of the committee. A direct calculation of the expectation and variance shows that while the expectation stays constant, the variance decays with rate of $O(1/T)$. On the other hand, a correlated committee's variance decays only up to some value that depends on the pairwise correlation between the members.

Our experiments strengthen those claims, as generally speaking the results improved when a committee was used. Having said that, more research could be made in the future as to examine how correlated out members actually are. For example - we know for a fact that using a different metric for the kNN distance decreases correlation, but the datasets that were used are highly correlated in the sense that there is a known transformation from one to another (though this transformation may only be one-directional), as the datasets are activations of the same neural network.

4.4 Implementation

Throughout all of our experiments, we used the same DCkNN framework and the same preprocessing - A sample image was first reshaped to 256×256 and then center cropped to 224×224. Furthermore, every image was normalized with respect to mean $[0.485, 0.456, 0.406]$ and standard-deviation $[0.229, 0.224, 0.225]$. In the specific case of a grey-scale image, we duplicated the grey-scale values to three distinct channels. to account for the requirement of three channels input in the ResNet architecture (this was relevant for the MNIST dataset). Next, the image was propagated through a pretrained ResNet18 which has four layer with two blocks in each and one last average pooling layer. In our pipeline we saved the activation after each block to memory, where each such activation is a tensor of different dimensions (c, h, w). Furthermore, we saved the activation that is the output of the last average pooling layer, which is a vector shaped $(512,)$.[2]

When calculating kNN for every activation, we used a different metric based on the depth of the activation itself. For activations that are not the last one, i.e activations that are not the one that occurs after the average pooling, we used a version of earth movers distance in which we calculate the cumulative summation in every dimension, and then take the ℓ_2 norm on the results. For the last activation we skipped the cumulative summation and only used ℓ_2. We have also used $k = 2$ throughout our experiments, as this was shown to provide better results, and was also suggested in (Bergman et al. [2020]). Finally, as we have performed our kNN processing on tensors that could not fit into our RAM, a "split-kNN" approach was used - First we splitted our train and test data into batches, and then calculated the pair-wise distance between every pair of batches. Once this step was finished, we extracted the k nearest neighbours from the train set that correspond to every test sample. Lastly, we extracted yet again the k nearest neighbours out of the batches of neighbours.

In terms of the datasets that was used, in most cases the training set was truncated in order to be able to save all activations to memory and avoid memory allocations problems. More specifically, for the part of One-Vs-Other - every train set contained 1000 samples, yet the test set included all the data that is provided. For the MVTec dataset, we have (as described) merged the various anomalous instances of every class into one single anomalous test data. After doing so, we have performed the same pipeline as in the unimodal test case for every class.

To generate the committee decision, we first plotted a ROC curve for every member in the ensemble. By doing so, we were able to find the best threshold for which the members classifications are optimized. We have defined the optimal threshold as the one that corresponds to a point (FPR, TPR) that is closest to $(0, 1)$ in our ROC plot. In other words, the threshold that was chosen is the one that corresponds to the point (FPR, FPR) which minimizes the distance $\sqrt{FPR^2 + (TPR - 1)^2}$. Once the threshold of every member in the ensemble was defined, the final prediction was the majority vote of all the binary classifications of all members.

[2]More details can be found in the provided code

5 Conclusions & Outlook

In this work we have presented a simple pipeline for anomaly detection that uses multiple activation maps given from a pre-trained ImageNet neural network like ResNet. Given those activations, we have generalized the concept of what features are the ones that describe a "normal" sample the most. Using this information, and given a new test sample - the answer to whether it is "normal" or "anomalous" was given by looking at the distance of its features from the features of the "normal" dataset. By doing to we were able to distinguish anomalous samples, as those (in most cases) were further away in terms of their feature values from the training activations set.

We have shown that using more than one deep feature has the power of improving the results of our classifier with some significant margins in some cases, though we did not manage to improve on state-of-the-art works from recent years. More specifically, for "vanilla" dataset like MNIST digits, an improvement of around 3% in ROCAUC was manageable, while also being at around -3%ROCAUC when compared to SOTA results. For "real-life" data on the other-hand, the improvement was more significant with around $+7\%$ to the ROCAUC, but did not manage to come close to SOTA approaches.

We attributed our results to how well a committee can improve on itself given the success rate of each member in it. Subsequently, in the case where shallow activation of anomalous instances did not differ much from those of a regular instance, the results of our approach did not work as well, and generally speaking were similar to the outcome when only considering the deepest layer as feature extractor.

It is also worth taking into consideration the vast amount of memory that is needed to perform our suggested pipeline - Even though in our case we have used 9 activations given from ResNet18, considering all activations of different, deeper, networks might be implausible. Regardless, it would be interesting to see, in the future, how one can choose the right subset of activations to consider in the final committee as the ones that maximize the anomaly detection scores.

References

H. Abdi and L. J. Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

J. An and S. Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1):1–18, 2015.

L. Bergman, N. Cohen, and Y. Hoshen. Deep nearest neighbor anomaly detection. *CoRR*, abs/2002.10445, 2020. URL `https://arxiv.org/abs/2002.10445`.

P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger. Mvtec ad – a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo. *A Geometric Framework for Unsupervised Anomaly Detection*, pages 77–101. Springer US, Boston, MA, 2002. ISBN 978-1-4615-0953-0. doi: 10.1007/978-1-4615-0953-0_4. URL `https://doi.org/10.1007/978-1-4615-0953-0_4`.

S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. *CoRR*, abs/1803.07728, 2018. URL `http://arxiv.org/abs/1803.07728`.

D. Hendrycks, M. Mazeika, and T. G. Dietterich. Deep anomaly detection with outlier exposure. *CoRR*, abs/1812.04606, 2018. URL `http://arxiv.org/abs/1812.04606`.

L. Hou, C.-P. Yu, and D. Samaras. Squared earth mover's distance-based loss for training deep neural networks. *arXiv preprint arXiv:1611.05916*, 2016.

L. J. Latecki, A. Lazarevic, and D. Pokrajac. Outlier detection with kernel density functions. In P. Perner, editor, *Machine Learning and Data Mining in Pattern Recognition*, pages 61–75, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-73499-4.

F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, 2008. doi: 10.1109/ICDM.2008.17.

N. Papernot and P. McDaniel. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*, 2018.

T. Reiss, N. Cohen, L. Bergman, and Y. Hoshen. Panda: Adapting pretrained features for anomaly detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2806–2814, 2021.

V. Roth. Kernel fisher discriminants for outlier detection. *Neural Computation*, 18(4):942–960, 2006. doi: 10.1162/neco.2006.18.4.942.

L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR, 2018.

T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pages 146–157. Springer, 2017.

B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL `https://proceedings.neurips.cc/paper/1999/file/8725fb777f25776ffa9076e44fcfd776-Paper.pdf`.

J. Tack, S. Mo, J. Jeong, and J. Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11839–11852. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/8965f76632d7672e7d3cf29c87ecaa0c-Paper.pdf`.