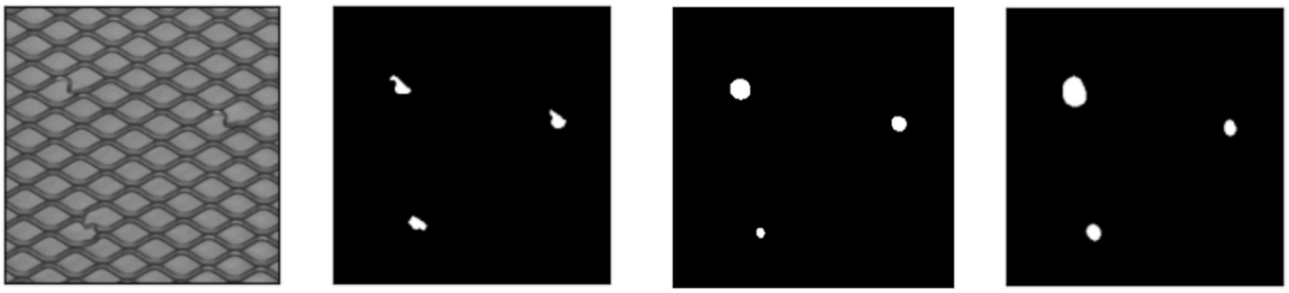
Image Anomaly Detection and Localization

Shruthan Radhakrishna (sr73@illinois.edu)

Anomaly (or outlier) detection is the task of identifying data objects that differ in their expected behavior. In the context of computer vision, anomaly detection is generally performed at two levels - (i) identifying if a given image is anomalous relative to other normal images (anomaly detection) and (ii) identifying the pixel/patch(s) in the anomalous image (anomaly localization).

In this project, two approaches from the literature - Sub-Image Anomaly Detection with Deep Pyramid Correspondences (SPADE) and a Patch Distribution Modeling Framework for Anomaly Detection and Localization (PaDiM) that address the anomaly detection and localization tasks have been implemented. Both approaches use a pre-trained WideResNet model to extract either image or pixel/patch level features from the image. SPADE then adopts a proximity-based approach to anomaly detection using these features, while PaDiM uses a distribution-based approach to anomaly detection.

Experimental results obtained after implementing the two methods show that PaDiM outperforms SPADE marginally for the anomaly localization task and outperforms SPADE significantly for the anomaly detection task. It is further concluded in the context of the current problem that while distribution-based approaches marginally outperform proximity-based approaches, pixel/patch-level features show significant improvements over image-level features.



A sample output: the leftmost image is the original image of a grid. The second image represents the true positions of the anomalies. The third and fourth images represent the predicted locations of the anomalies by the SPADE and PaDiM methods

Image Anomaly Detection and Localization

Shruthan Radhakrishna (sr73@illinois.edu)

1. Introduction

Anomaly (or outlier) detection is the task of identifying data objects that differ in their expected behavior. In practice, it is generally considered that most data objects follow the expected behavior and hence anomaly detection becomes the task of identifying the rare data objects that deviate from the behavior of most other data objects. For example, in manufacturing, anomalous objects are objects that vary from the specification and are rare occurrences. While humans are generally able to detect anomalous objects, the rarity of occurrence makes it a cumbersome task to be done manually.

In the context of computer vision, anomaly detection is generally performed at two levels - (i) at the level of the image where each image is given an anomaly score to identify if the image is anomalous compared to other images and (ii) at the level of pixels/patches in an image, where each pixel/patch of pixels in the image is given an anomaly score to identify which part of the image makes it anomalous w.r.t other images (this task is referred to as localization).

While anomaly detection can be viewed as a binary classification task (between two classes - anomalous and non-anomalous), fully-supervised techniques become ineffective as its often infeasible to include all possible anomalies in the training data. Hence a common approach to identifying anomalies involves training in a one-class setting i.e., a model is trained on a dataset with only normal (i.e., non-anomalous) images, and at test time, the images that differ from the normal-training set are classified as anomalous. This method is effective when a sufficient amount of normal examples are available for training.

Some other approaches like SPADE [1] and PaDiM [2] involve no/minimal training (but instead use pre-trained models like ResNet trained on the ImageNet dataset) and perform well on standard anomaly detection datasets. The aim of this project is to reproduce the results of these two papers using a common framework.

2. Background

Deriving from [3], anomaly detection approaches in a one-class setting may broadly be classified based on their approach as proximity-based or distribution based.

2.1 Proximity-based

These approaches assume that an image is an outlier if the proximity (in the feature space) of the image to its neighbors significantly deviates from he proximity of most other images to their neighbors in the same dataset of images. Design considerations in this approach involve feature representation of the images and the distance function to determine proximity,

2.2 Distribution-based

These approaches assume that normal (non-anomalous) images are generated by a statistical model $\mathscr{M}$ and images not following $\mathscr{M}$ are considered as outliers. Distribution-based approaches may further be divided into two classes:

(i) The first try to model the probability density function (PDF) of the distribution of normal images directly. A test image is then evaluated using the PDF and those images with low probability density values are considered to be anomalous.
(ii) The second class try to model the normal images using a reconstruction-function $\mathscr{F}$. Given a test image, if the test image cannot faithfully be reconstructed using $\mathscr{F}$, then the test image is considered to be anomalous. The architecture of $\mathscr{F}$ commonly involves autoencoders [4-6], variational autoencoders [7-8], or generative adversarial networks [9-11]. Common challenges in reconstruction-based methods as described in [1] include sensitivity to particular loss functions to evaluate the quality of reconstruction and identification of the most appropriate $\mathscr{F}$.

3. Approach

In this project, SPADE [1] and PaDiM [2] are implemented using a common framework. This framework involves two stages (i) feature extraction and (ii) anomaly detection and localization. The feature extraction step is common to both SPADE and PaDiM. Using the extracted features, SPADE uses a proximity-based approach to identify anomalies, while PaDiM uses a distribution based-approach to identify anomalies. These steps are further elucidated in the following sub-sections.

3.1 Feature Extraction

For feature-extraction, a Wide-ResNet50-2 model (from [12]) pre-trained on the ImageNet dataset (from [13]) is used. As implemented in the paper, the features from the ResNet model at the end of the first block (56×56), second block (28×28) and third block (14×14) are used. For notation, let F denote the feature extractor. Then for a image x_i, denote the extracted features as $f_i = F(x_i)$. While SPADE concatenates the features from the various blocks, PaDiM associates each patch to a spatially corresponding activation vector in the CNN activation maps. Since the activation maps have a lower resolution than the input image, the input image is divided into a $W \times H$ grid where $W \times H$ represents the resolution of the largest activation map. Then for a patch at location (i, j) in the grid, the activation vectors from different layers are concatenated to get the embedding vector x_{ij}. The embedding vector x_{ij} hence encapsulates both fine-grained and global contexts.

3.2 Anomaly Detection

This stage involves identifying if a given image is anomalous or not.

3.2.1 SPADE

Given a test image y, the k nearest neighbors $N_k(f_y)$ to the given test image from the training set are identified. Euclidean distance is used to measure the distance between the extracted features of the images. The test image y is considered to be anomalous if the average distance between y and its k nearest neighbors exceeds a threshold τ. Hence,

$$\text{Classify } y \text{ as anomalous if } d(y) = \frac{1}{K} \sum_{x \in N_K(f_y)} \|f_y - f_x\|^2 > \tau.$$

3.2.2 PaDiM

Unlike SPADE that compares the features of the test image to its nearest neighbors, PaDiM tries to compute model each position in the $W \times H$ grid as a multivariate Gaussian. The training images are used to compute the parameters of the Gaussian. For each position (i, j) a set of patch-embedding vectors $X_{ij} = \{x_{ij}^k, k = 1...N\}$ are computed from the N non-anomalous training images. Then a

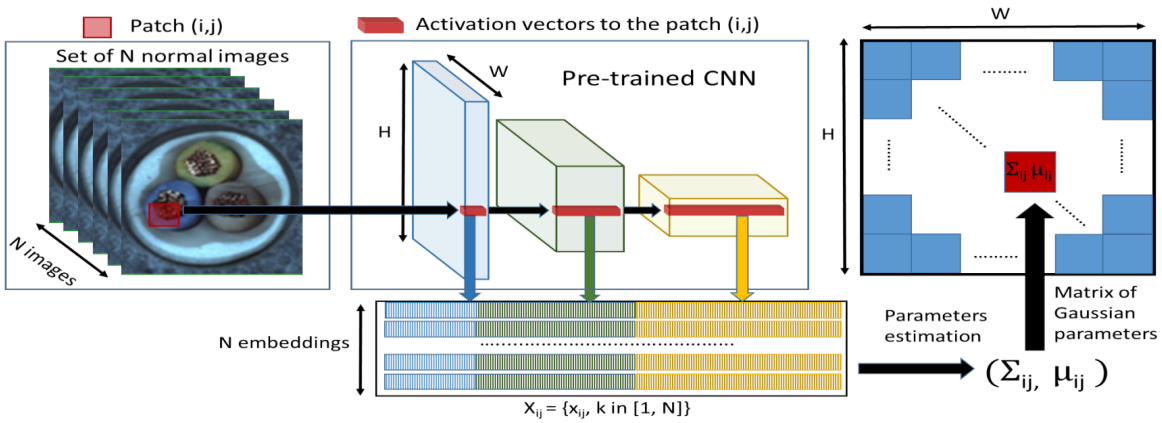


Fig 1. Learning Gaussian parameters for each position (i, j) of the largest feature map. Image from [2]

Gaussian $\mathcal{N}(\mu_{ij}, \Sigma_{ij})$ is obtained where μ_{ij} is the sample mean and sample covariance Σ_{ij} is computed as:

$$\Sigma_{ij} = \frac{1}{N-1}\sum_{k=1}^{N}(x_{ij}^k - \mu_{ij})(x_{ij}^k - \mu_{ij})^T + \epsilon I$$

To identify if a given test image is an anomaly, the Mahalanobis distance measure $M(y_{ij})$ is used to give an anomaly score to each patch at location (i, j).

$$M(y_{ij}) = \sqrt{(y_{ij} - \mu_{ij})^T \Sigma_{ij}^{-1}((y_{ij} - \mu_{ij})}$$

The maximum value in the anomaly map M is considered to be the anomaly score of the entire image, and the image is classified as anomalous if the anomaly score exceeds a threshold τ

3.3 Anomaly Localization

This stage involves identifying the location in the image where the anomaly occurs.

3.3.1 SPADE

For localizing the position of anomaly, each position in the test image y is represented by concatenating features from the output of the first, second and third blocks of the ResNet model as described in section 3.1. Next, for every pixel location p in the image, a gallery of the features of the same pixel location of the K nearest neighbors is constructed as $G = \{F(x_1, p), \ldots, F(x_K, p)\}$. Then the anomaly score at pixel p is the average distance between the features of y at location p, $F(y, p)$, and its κ ($\leq K$) nearest features from gallery G:

$$d(y, p) = \frac{1}{\kappa}\sum_{x \in N_\kappa(F(y,p))} \|F(x, p) - F(y, p)\|^2$$

Then, a pixel is determined to be anomalous if $d(y, p) > \theta$, for a threshold θ.

3.3.2 PaDiM

Since the anomaly map M from section 3.2.2 already scores for each location in the $W \times H$ grid, these scores are used to identify the location of anomalies in the image (high scores in this map indicate the anomalous areas).

4. Experimental Details

4.1 Dataset

The models have been implemented on the MVTec AD dataset from [14]. This dataset is useful to test anomaly localization algorithms for industrial quality control and in a one-class learning setting. It contains images from fifteen classes, five consist of textures (carpet, grid, leather, tile, wood) and the other ten consist of objects (bottle, cable, capsule, hazelnut, metal nut, pill, screw, toothbrush, transistor, and zipper). For each class, the training set consists of normal images, while the test set consists of normal images along with images containing different anomalies.

4.2 Implementation details

- Following the implementation details in SPADE, the images were resized to 256 x 256 and then center cropped to 224 x 224. Similarly, cv2.INTERAREA was used for resizing in the case of SPADE while bicubic interpolation was used for PaDiM (as suggested in the paper).
- A Gaussian filter with parameter σ = 4 on the anomaly maps as suggested in both the papers.
- For SPADE, the original paper uses $K = 50$ and $\kappa = 1$. In my analysis, I have used $K = 10$ and 20, and $\kappa = 1$. Implementing $K = 50$ on Google Colab was prohibitive.
- For both SPADE and PaDiM, ROC-AUC percentages have been reported instead of selecting thresholds τ and θ. This is consistent with how results are reported in the papers.

5. Results

The ROC-AUC percentages for image localization for the 15 classes have been presented in the Table 1.

Class	SPADE Paper	PaDiM Paper	SPADE K=10	SPADE K=20	PaDiM
Carpet	97.5	99.1	99.0	99.0	99.0
Grid	93.7	97.3	98.6	98.6	97.1
Leather	97.6	99.2	99.3	99.3	99.1
Tile	87.4	94.1	93.3	93.6	94.5
Wood	88.5	94.9	95.5	95.6	94.4
Bottle	98.4	98.3	97.2	97.5	98.2
Cable	97.2	96.7	93.2	93.4	96.5
Capsule	99.0	98.5	98.5	98.6	98.6
Hazlenut	99.1	98.2	98.6	98.7	98.1
Metal nut	98.1	97.2	97.2	97.3	97.5
Pill	96.5	95.7	95.3	95.4	95.6
Screw	98.9	98.5	99.2	99.3	98.4
Toothbrush	97.9	98.8	98.8	98.9	98.7
Transistor	94.1	97.5	87.9	88.5	97.6
Zipper	96.5	98.5	98.8	98.8	98.4
Average	**96.0**	**97.5**	**96.7**	**96.8**	**97.5**

Table 1: ROC-AUC% for anomaly localization

Additionally, the ROC-AUC percentages for anomaly detection (i.e. if a given image is anomalous) are shown in Table 2. Please note that the original papers do not report these values for each class separately.

Class	SPADE K=10	SPADE K=20	ViT SPADE	PaDiM
Carpet	92.7	92.7	92.3	99.8
Grid	43.6	39.1	48.7	97.3
Leather	85.6	94.3	99.6	100.0
Tile	96.1	96.0	98.2	98.4
Wood	95.8	96.1	87.5	99.3
Bottle	97.1	96.6	99.0	99.9
Cable	84.2	83.5	77.0	89.7
Capsule	87.7	86.0	78.9	90.5
Hazlenut	85.6	86.1	90.5	94.1
Metal nut	69.4	66.2	81.8	100.0
Pill	79.8	79.2	75.9	92.1
Screw	63.1	57.9	57.5	85.9
Toothbrush	86.7	86.9	88.3	97.5
Transistor	89.9	88.7	68.0	97.9
Zipper	96.5	96.0	78.5	91.6
Average	84.25	83.02	81.44	95.6

Table 2. ROC-AUC% for anomaly detection. This table includes a column - ViT SPADE. This represents an attempt to replace the WideResNet model in SPADE to a pre-trained ViT model. Due to paucity of time, I was unable to delve deeper into utilizing a ViT for detection and localization.

6. Discussion

I. *Overview of the results:*
   - *A.* The results obtained by me are largely consistent with the results presented in the PaDiM paper. The results slightly differ in the case of SPADE (possibly due to the difference in choice of K).
   - *B.* PaDiM tends to perform better than SPADE, in both my implementations and as reported in the original papers, for both the tasks of anomaly detection and localization. The difference in performance is stark in the case of anomaly detection.
   - *C.* The performance of SPADE tends to decrease as the number of nearest neighbors, K, increases. This is observed in my experiments with K = 10 and 20, and is further confirmed by the fact that the paper using K=50 reports a lower average ROC-AUC% for anomaly localization than what I have obtained.

II. Analysis of the results for anomaly localization

A. We see that both SPADE and PaDiM perform well for localizing the position of the anomaly in a given image. Both the methods get representations at the pixel/patch level and then use either a nearest neighbor approach or model a Gaussian distribution for each pixel/patch location. Given their similar approaches to animal localization, it can be inferred that utilizing a distribution-based approach only yields marginal improvements in performance compared to a proximity-based approach.

III. Analysis of the results for anomaly detection

A. In this case, we see that PaDiM outperforms SPADE. This is likely due to the fact that SPADE uses an image-level representation for each image, while PaDiM uses the same patch-level features extracted for anomaly localization. The use of image-level representations may result in a loss of information, particularly since anomalies are typically confined to small portions of the image, as evident from the anomaly maps in the appendix. Therefore, only a small fraction of the image, and subsequently the image representation, is relevant for anomaly detection. On the other hand, when considering patches within the image, the patches located at anomalous positions in anomalous images are expected to exhibit significant dissimilarities compared to patches at the same positions in normal images. Consequently, pixel/patch level features are more likely to preserve relevant information regarding anomalous locations than image-level features.

IV. On inference time and memory consumption

A. Neither approach involves any training since pre-trained models are used for feature extraction. Overall running time for PaDiM took around 52 minutes on the MVTec AD dataset while SPADE taking around 118 minutes on an A100 GPU on Google Colab. It is important to note that in each run, SPADE involved two sets of feature extractions - one for anomaly detection and another for anomaly localization. Hence, each image is passed through the pre-trained model twice. PaDiM on the other hand uses the same set of features for both localization and anomaly detection and hence only has to pass the images though the pre-trained model once.

   It may be possible to implement SPADE feature-extraction in pass through the dataset, but that wasn't explored in this project. Considering this, its fair to conclude that there is no significant difference in the inference times between the two methods. The PaDiM paper however mentions a 7 times reduction in inference time.

B. On a (relatively) small dataset like MVTec AD, it was observed that PaDiM had higher memory-consumption than SPADE. However, since optimizing memory consumption was not considered while designing my experiments, I refrain from presenting an objective judgment on the memory consumption of either model.

   It is worth noting that on larger datasets, it is likely that SPADE would have a higher memory consumption than PaDiM due to the nature of the k-nearest neighbor algorithm. In general, the memory complexity of PaDiM is independent of the size of the dataset or the resolution of the images since the only memory consumption is for the pre-trained model and the parameters of the Gaussian model of each patch (number of patches is determined by the size of the largest CNN feature map in the pre-trained model. SPADE needs to compute the K nearest neighbors of the test image, hence increasing the memory complexity,

7. Conclusion and Future Work

I. *Conclusion from the experiments performed:*
   A. Distribution-based outlier detection methods perform marginally better than proximty-based methods in the context of the current problem.
   B. Pixel/Path level features are more effective in detecting anomalies.
   C. PaDiM performs better than SPADE and is likely to be more memory-efficient with large datasets.

II. *An immediate extension to the current project:*
   The current work uses a pre-trained WideResNet model for feature extraction. An attempt to use pre-trained vision transformers has also been made, however it has only been used for anomaly detection. An immediate next step would involve using a modern architecture like it to extract pixel-level features and use the proximity or distribution-based methods on those features for anomaly detection and localization.

III. *A comment on the state-of-the-art:*
   Most recent papers involve using vision transformer-based methods. For example, recent work like [15] explore combining two approaches described in section 2.2 i.e. they involve simultaneously learning a reconstruction function as well as a probability density function. This is done in the context of a vision transformer where the encoder outputs are used to learn the probability density function while the decoder is expected to reconstruct the image fed into the encoder. Experimenting with such a method would require training or fine-tuning a vision-transformer which is an expensive process.

Appendix

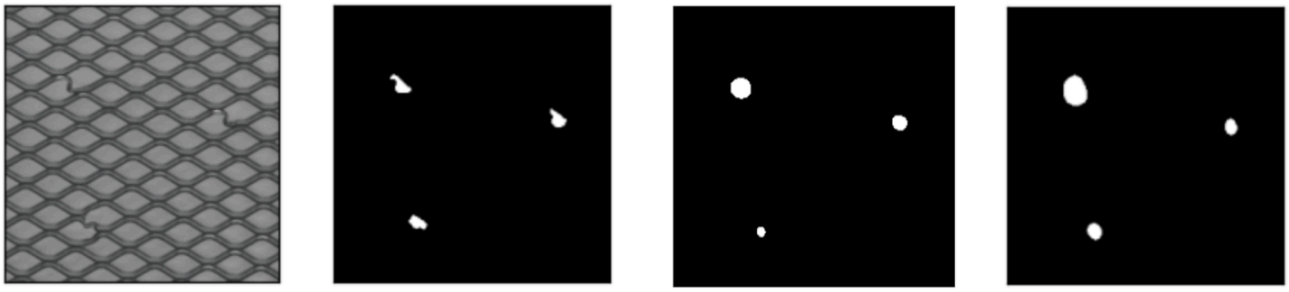
Some outputs from the experiments carried out are shown below. Each image shows the actual image, the ground truth mask (location of anomaly in white), the mask predicted by SPADE (K=20) and by PaDiM.

I. Image of a transistor. In this case SPADE produces spurious anomalies.

II. Image of a toothbrush In this case SPADE misses a part of the anomaly

Actual Image | Ground Truth Mask | SPADE Prediction | PaDiM Prediction

III. Image of a grid. In this case SPADE both predictions are reasonable accurate.



Actual Image | Ground Truth Mask | SPADE Prediction | PaDiM Prediction

References

[1] Niv Cohen and Yedid Hoshen. 2020. Sub-image anomaly detection with deep pyramid correspondences.

[2] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. 2020. Padim: a patch distribution modeling framework for anomaly detection and localization.

[3] Han, Jiawei, Jian Pei, and Hanghang Tong. *Data mining: concepts and techniques*. Morgan kaufmann, 2022.

[4] P. Bergmann, S. Lowe, M. Fauser, D. Sattlegger, and C. Steger, "Improving unsupervised defect segmentation by applying structural similarity to autoencoders," in VISIGRAPP, 2019.

[5] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. van den Hengel, "Memorizing normality to detect anomaly: Memoryaugmented deep autoencoder for unsupervised anomaly detection," in ICCV, 2019.

[6] C. Huang, F. Ye, J. Cao, M. Li, Y. Zhang, and C. Lu, "Attribute restoration framework for anomaly detection," in arXiv, 1911.10676, 2019

[7] S. Venkataramanan, K.-C. Peng, R. V. Singh, and A. Mahalanobis, "Attention guided anomaly localization in images," in arXiv, 1911.08616, 2019.

[8] K. Sato, K. Hama, T. Matsubara, and K. Uehara, "Predictable uncertainty-aware unsupervised deep anomaly segmentation," in IJCNN, 2019.

[9] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, “Adversarially learned one-class classifier for novelty detection,” in CVPR, 2018.

[10] S. Pidhorskyi, R. Almohsen, D. A. Adjeroh, and G. Doretto, “Generative probabilistic novelty detection with adversarial autoencoders,” in NIPS, 2018.

[11] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, “Ganomaly: Semisupervised anomaly detection via adversarial training,” ACCV, 2018.

[12] Zagoruyko, Sergey, and Nikos Komodakis. "Wide residual networks." *arXiv preprint arXiv:1605.07146* (2016).

[13] Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009.

[14] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. 2019. “MVTec AD — A comprehensive real-world dataset for unsupervised anomaly detection”. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 9584–9592.

[15] Mishra, Pankaj, et al. "VT-ADL: A vision transformer network for image anomaly detection and localization." 2021 IEEE 30th International Symposium on Industrial Electronics (ISIE). IEEE, 2021.