

Anomaly Detection by Clustering DINO Embeddings using a Dirichlet Process Mixture

Nico Schulthess^(✉) and Ender Konukoglu^(iD)

Computer Vision Lab, ETH Zurich, Zurich, Switzerland
`nico.schulthess@vision.ee.ethz.ch`

Abstract. In this work, we leverage informative embeddings from foundational models for unsupervised anomaly detection in medical imaging. For small datasets, a memory-bank of normative features can directly be used for anomaly detection which has been demonstrated recently. However, this is unsuitable for large medical datasets as the computational burden increases substantially. Therefore, we propose to model the distribution of normative DINOv2 embeddings with a Dirichlet Process Mixture model (DPMM), a non-parametric mixture model that automatically adjusts the number of mixture components to the data at hand. Rather than using a memory bank, we use the similarity between the component centers and the embeddings as anomaly score function to create a coarse anomaly segmentation mask. Our experiments show that through DPMM embeddings of DINOv2, despite being trained on natural images, achieve very competitive anomaly detection performance on medical imaging benchmarks and can do this while at least halving the computation time at inference. Our analysis further indicates that normalized DINOv2 embeddings are generally more aligned with anatomical structures than unnormalized features, even in the presence of anomalies, making them great representations for anomaly detection. The code is available at <https://github.com/NicoSchulthess/anomalydino-dpmm>.

Keywords: Unsupervised Anomaly Detection · Dirichlet Process Mixture Model · Foundation Model.

1 Introduction

Anomaly detection focuses on identifying samples that deviate from the norm, such as the detection of lesions in medical imaging or defects in industrial inspection. Here, the norm is defined by normal example samples, for instance by images of healthy volunteers in medical contexts or intact objects in industrial settings. Of particular interest is the unsupervised setting, where no anomalous samples are used during training. This avoids introducing unwanted biases towards certain types of anomalies, does not introduce any potential class imbalances, and alleviates the need for annotating anomalous samples for each modality and anatomy.

One line of work to tackle unsupervised anomaly detection (UAD) are reconstruction-based approaches, where a generative model is trained to reconstruct

normal images. The underlying hypothesis is that since the model is trained to reconstruct normal images well, it will do a poor job in anomalous regions and the reconstruction error will highlight such regions. Popular choices for the reconstruction model are adaptations of autoencoders [6,24], variational autoencoders [34,8,7], masked autoencoders [27], generative adversarial networks [29], and denoising diffusion probabilistic models [31]. Instead of reconstructing images, a feature reconstruction objective can be used in a knowledge distillation setup where a student network is trained on normal data to predict similar features to a pretrained teacher network [23]. Then, the embeddings are assumed to differ between the student and teacher networks for anomalous data while they are similar for normal data.

Another line of work aims to model the distribution of the features extracted from normal samples. One choice is to use normalizing flow models to map the normal patch feature distribution to a tractable distribution [14]. However, normalizing flow models can assign high probabilities to anomalous samples and should thus be used with caution for anomaly detection [20,25]. SPADE [9] constructs a memory bank of normal embeddings and uses the average distance to k nearest neighbors as an anomaly score for a given sample. PaDiM [11] creates a probabilistic representation of the features by modeling feature distribution for each pixel by a multivariate Gaussian. PatchCore [28] and ProtoAD [17] employ a memory bank as in SPADE, however, both methods compress the memory bank to reduce its size using a greedy coresets selection and a nonparametric hierarchical clustering, respectively. AnomalyDINO [10] stores patch features from DINOv2 [26] using only a few normal images to construct a memory bank.

The comparison with large memory banks usually leads to accurate anomaly detections, however at the cost of substantially long runtimes and high memory utilization. Having small yet expressive memory banks is crucial for anomaly detection on a large scale as it would be required for medical imaging.

In this work, we aim to effectively compress the memory bank of AnomalyDINO. We propose to model the patch features using a Dirichlet process mixture model (DPMM) [1,13] with Gaussian component distributions, which is a Gaussian mixture model without a fixed number of components that instead determines the necessary number of components based on the data, allowing for a more flexible, data-driven approach. We fit the DPMM using a batched expectation maximization algorithm [19]. We use the component means as prototypes for the normal distribution and define the anomaly score as the similarity to these prototypes. For an overview of our approach refer to Fig. 1.

Our framework achieves competitive performance on the BMAD benchmark [3] while being highly efficient to compute thanks to the small number of prototypes in use to model the normative distribution. Further, our analyses indicate that features from DINOv2 trained on natural images are useful for anomaly detection in medical imaging, especially after normalization.

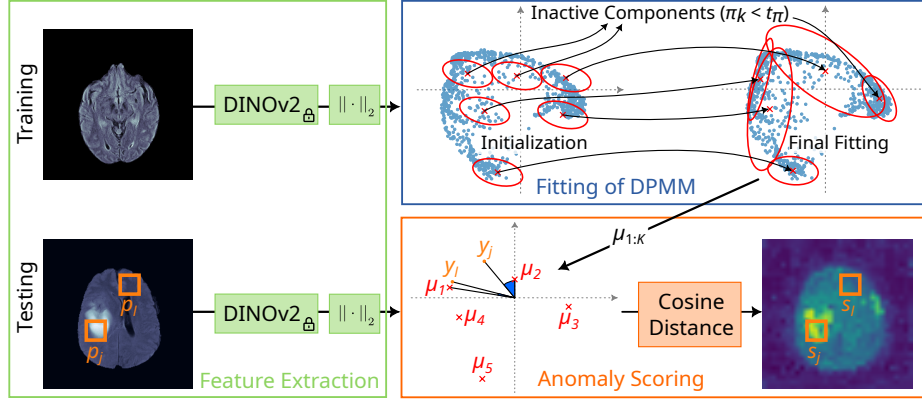


Fig. 1: In the training phase, we fit a Dirichlet process mixture model (DPMM) with Gaussian components (indicated in red) to the DINOv2 patch embeddings of the normal samples. After fitting the DPMM to the normal embeddings, we use the cosine distance between each patch feature and the closest component mean to obtain a patch-level anomaly score, which we interpolate for a pixel-wise anomaly map.

2 Methods

2.1 Fitting the Dirichlet Process Mixture Model (DPMM)

Following [19], we fit the DPMM with Gaussian components using a batched expectation maximization (EM) approach. For this, we consider a DPMM truncated to K components. For sufficiently large K , the truncated DPMM is more flexible than a standard Gaussian mixture model as components can vanish in the DPMM if they are not required. We define the probability density of a sample y under the DPMM with parameters Φ as

$$p(y|\Phi) = \sum_{k=1}^K \pi_k \mathcal{N}(y|\theta_k), \quad (1)$$

where $\theta_k = (\mu_k, \Sigma_k)$ and the component weights $\{\pi_k\}_{k=1}^K$ are defined as

$$\pi_k = \begin{cases} v_1, & \text{if } k = 1 \\ v_k \prod_{j=1}^{k-1} (1 - v_j), & \text{otherwise,} \end{cases} \quad (2)$$

with $v_k \sim \text{Beta}(1, \alpha)$ for $k \in \{1, \dots, K-1\}$, $v_K = 1$, and α being the concentration parameter of the Dirichlet process. Given a batch of embeddings $y_{1:N}$, we want to estimate the parameters $\Phi = (v_{1:K-1}, \theta_{1:K})$ and α by maximizing

$$Q_{\Phi}(\Phi, \Phi^{(t-1)}) = \mathbb{E}_{x_{1:N}|y_{1:N}, \theta_{1:K}^{(t-1)}, \alpha^{(t-1)}} [\log p(x_{1:N}, y_{1:N}, v_{1:K}|\theta_{1:K}, \alpha)] \quad (3)$$

and

$$Q_\alpha(\alpha, \alpha^{(t-1)}) = \mathbb{E}_{x_{1:N} | y_{1:N}, \theta_{1:K}^{(t-1)}, \alpha^{(t-1)}} \left[\log \int p(x_{1:N}, y_{1:N}, v_{1:K} | \theta_{1:K}, \alpha) dv_{1:K} \right] \quad (4)$$

with respect to Φ and α , respectively, where $x_n \in \{1, \dots, K\}$ is the random variable for the component assignment of the embedding y_n .

For Gaussian components, we can obtain the update formulas

$$\theta_k^{(t)} = \left(\mu_k^{(t)}, \Sigma_k^{(t)} \right) = \left(\frac{\bar{m}_k^{(t)}}{\sum_{j=1}^K \bar{p}_j^{(t)}}, \frac{\bar{c}_k^{(t)}}{\sum_{j=1}^K \bar{p}_j^{(t)}} - \mu_k^{(t)} \mu_k^{(t)\top} \right) \quad (5)$$

$$v_k^{(t)} = \frac{\bar{p}_k^{(t)}}{\bar{p}_k^{(t)} + \alpha^{(t-1)} - 1 + \sum_{j=k+1}^K \bar{p}_j^{(t)}}, \quad (6)$$

where \bar{p}_k , \bar{m}_k , and \bar{c}_k are moving averages of the sufficient statistics defined by

$$\bar{p}_k^{(t)} = (1 - \gamma_t) \bar{p}_k^{(t-1)} + \gamma_t \mathbb{E}_y \left[\Pr(x = k | y, \Phi^{(t-1)}) \right] \quad (7)$$

$$\bar{m}_k^{(t)} = (1 - \gamma_t) \bar{m}_k^{(t-1)} + \gamma_t \mathbb{E}_y \left[y \Pr(x = k | y, \Phi^{(t-1)}) \right] \quad (8)$$

$$\bar{c}_k^{(t)} = (1 - \gamma_t) \bar{c}_k^{(t-1)} + \gamma_t \mathbb{E}_y \left[yy^\top \Pr(x = k | y, \Phi^{(t-1)}) \right]. \quad (9)$$

Finally, the update formula for the concentration parameter α is

$$\alpha^{(t)} = \frac{N - 1}{\sum_{k=1}^{N-1} \Psi \left(\alpha^{(t-1)} + 1 + \bar{C}_k^{(t)} + \bar{C}_{>k}^{(t)} \right) - \Psi \left(\alpha^{(t-1)} + \bar{C}_{>k}^{(t)} \right)}, \quad (10)$$

with $\bar{C}_k^{(t)} = B \cdot \bar{p}_k^{(t)}$, $\bar{C}_{>k}^{(t)} = B \sum_{j=k+1}^K \bar{p}_j^{(t)}$, and B being the batch size. Further details on the derivation of the formulas above can be found in [19].

2.2 Anomaly Scoring

Given the DPMM fitted to the normal distribution of patch embeddings, we can evaluate the alignment of a test sample with the modeled distribution using several metrics. Specifically, we consider the likelihood of the most probable component as well as the cosine similarity and Euclidean distance to the nearest component center. As shown in Fig. 2, certain components tend to model specific anatomical structures. Among the three metrics, cosine similarity provides the most consistent assignment of anatomical structures across different samples, particularly for anomalous samples. Using the full likelihood of the model in Eq. (1) is not an option, as in our experiments the full likelihood is usually dominated by the likelihood of only one component. Based on this observation, we 1) choose to normalize the embeddings y for training and testing and 2) use cosine similarity as our anomaly score function.

During fitting the DPMM, some components can vanish as their weight π_k shrinks towards zero. To suppress all components with vanishing weights, we

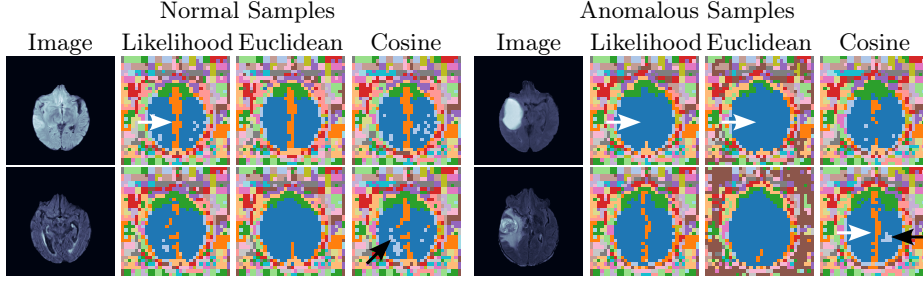


Fig. 2: For two normal and two anomalous images, we visualized the index of the closest component for each patch in terms of component likelihood ($\arg\max_k \mathcal{N}(y_n | \mu_k, \Sigma_k)$), Euclidean distance ($\arg\min_k \|y_n - \mu_k\|_2$), and cosine similarity ($\arg\max_k y_n^\top \mu_k / (\|y_n\|_2 \cdot \|\mu_k\|_2)$). To highlight the consistency over multiple samples, we selected samples showing similar anatomical structures for this visualization, with and without lesions. Different brain structures of the normal images are seemingly represented by different components: tissues at the centerline by the orange component (white arrow), ventricles by the bright blue component (black arrow). For the anomalous images, we can observe a similar assignment when using cosine similarity. However, the component likelihood and the Euclidean distance fail to assign these structures to the correct component in several anomalous cases.

apply a threshold t_π to the component weights first, and the anomaly score function becomes

$$s(y_n) = \max_{k: \pi_k > t_\pi} \frac{y_n^\top \mu_k}{\|y_n\|_2 \|\mu_k\|_2}. \quad (11)$$

Since DINOv2 only creates embeddings of image patches, this yields a coarse, patch-wise anomaly score map. To obtain pixel-wise anomaly scores, we interpolate the patch-wise anomaly scores to full resolution.

To create an anomaly segmentation mask, we threshold the pixel-level anomaly score map. To stay in line with the unsupervised setting, we select the threshold such that a predefined false positive rate on the healthy validation set is achieved as in [8].

3 Experiments and Results

3.1 Experimental Setup

Model Architecture As backbone, we use the small architecture of DINOv2 with the official pretrained weights¹, leading to a 384-dimensional embedding space. We truncate the DPMM to 500 components and use diagonal covariance matrices for the Gaussian components. We set the discount factor in the moving

¹ <https://github.com/facebookresearch/dinov2> under the Apache 2.0 License.

average γ_t to 0.2 to accumulate the statistics over many batches and the component weight threshold t_π to $1e-6$. We train the DPMM with a batch size of 12 images, which are resized to a resolution of 448×448 , resulting in batches of 12 288 embedding vectors. We trained the DPMM for 40 epochs and selected the model achieving the highest log-likelihood on the normal validation samples.

Datasets We conducted our experiments using the anomaly segmentation datasets from the BMAD benchmark [3], which include BraTS2021 [2] (brain MRI), BTCV [21] and LiTs [4] (liver CT), and RESC [16] (retinal OCT). BraTS2021 comprises 8179 normal and 3119 anomalous images, the liver CT datasets have 2468 normal and 733 anomalous images, and RESC contains 5408 normal and 809 anomalous images.

Baselines We compare our work with multiple unsupervised anomaly detectors, namely DRAEM [33], UTRAD [5], RD4AD [12], STFPM [30], PaDiM [11], PatchCore [28], CFA [22], CFLOW [14] and AnomalyDINO [10]. For PatchCore, we applied the proposed reduction to 10% of the memory bank as well as a much stronger reduction to 1024 and to 150 prototypes, which we refer to as few-shot setting. For AnomalyDINO, we tested the full-shot and one-shot setting, with the one-shot setting corresponding to 1024 prototypes.

Evaluation Metrics We compare our method with state-of-the-art anomaly detectors using the area under the receiver operator characteristic curve (AUROC) and the area under the precision recall curve (AUPR) at pixel-level. We also report the per-sample runtime and maximum memory utilization for the anomaly detection stage, measured with a batch size of 1 and one separate thread for data loading using an NVIDIA A6000 GPU with 48 GB of VRAM. Further, we compare the anomaly segmentation performance with AnomalyDINO using the Dice score at 1%, 5%, and 10% false positive rate on the normal validation images.

To test the statistical significance of our results over PatchCore with 150 prototypes, we run a paired permutation test on the image-wise AUROC and AUPR scores with 10 000 permutations for the BraTS2021 and the RESC datasets. Using image-wise scores is necessary as the anomaly score maps for PatchCore and our method are constructed using Euclidean and cosine distance to the nearest prototypes, respectively, and thus cannot be compared directly.

3.2 Results

We provide a quantitative comparison of our method with state-of-the-art approaches in Tab. 1. Over all datasets, only AnomalyDINO in the full-shot setting manages to outperform our approach consistently. However, AnomalyDINO comes with enormous memory requirements and with long runtimes. For BraTS and BTCV+LiTs, the AUROC for our approach is only 2.13% and 1.83% lower

Table 1: AUROC and AUPR scores for pixel-level anomaly detection as well as per-sample runtime (without data loading or metric computation) and memory utilization of the anomaly detection on the test set. The methods with the suffix "R50" use WideResNet50 [32] as the backbone instead of ResNet18 [15]. PatchCore-R50 in the standard setting is not available for the BraTS2021 dataset as the coreset subsampling could not fit into the GPU memory. We averaged AUROC and AUPR scores for AnomalyDINO in the one-shot setting over 25 runs.

Method	BraTS2021				BTCV+LiTs				RESC			
	AUROC [%]	AUPR [%]	Runtime [ms]	Memory [GB]	AUROC [%]	AUPR [%]	Runtime [ms]	Memory [GB]	AUROC [%]	AUPR [%]	Runtime [ms]	Memory [GB]
DRAEM	92.70	38.66	149	0.5	93.82	16.01	125	0.5	86.78	31.86	130	0.5
UTRAD	92.81	24.63	43	0.9	84.28	1.25	45	0.9	55.52	6.00	42	0.9
RD4AD	97.81	52.11	139	0.4	94.35	3.48	118	0.4	96.92	67.56	117	0.4
STFPM	96.64	45.45	122	0.0	94.70	3.20	106	0.0	93.85	45.07	108	0.0
STFPM-R50	97.32	52.87	133	0.2	96.14	5.10	118	0.2	94.78	52.66	111	0.2
PaDiM	85.77	8.21	145	0.2	89.77	1.66	127	0.2	90.00	31.07	124	0.2
PaDiM-R50	83.36	6.89	158	3.7	90.48	1.79	131	3.7	92.12	35.31	129	3.7
PatchCore	98.34	67.37	202	3.4	96.67	5.66	132	0.7	97.08	70.47	150	2.0
PatchCore-R50	N/A				96.63	5.99	160	1.9	97.55	72.62	218	5.0
CFA	97.27	54.15	134	0.3	97.29	7.65	125	0.3	91.65	26.52	116	0.3
CFLOW	95.18	32.63	255	1.0	97.13	11.63	223	0.9	94.79	53.21	219	0.9
AnomalyDINO	97.71	58.69	422	33.1	97.24	17.91	107	6.9	94.04	50.45	984	38.1
PatchCore-R50 ¹	95.38	55.07	149	0.2	45.70	0.32	130	0.1	93.28	61.88	132	0.1
PatchCore-R50 ²	95.02	42.68	158	0.2	16.31	0.23	139	0.1	78.70	25.98	136	0.1
▲	96.57	44.74			98.17	14.30			92.24	42.63		
AnomalyDINO ^{1*}	94.61	31.21	24	0.2	94.60	5.98	26	0.2	89.52	36.98	49	0.4
▼	84.97	8.72			89.45	2.93			85.10	26.51		
Ours	96.21 [†]	43.43 [†]	39	1.6	95.46	9.02	35	1.5	90.20 [†]	41.66 [†]	58	2.0

¹ few-shot with 1024 prototypes ² few-shot with 150 prototypes

* max (▲), average, and min (▼) value over all seeds provided [†] statistically significant improvement over PatchCore-R50² (permutation test with $p < 1\%$)

than for the best-performing method, respectively. With the exception of the one-shot AnomalyDINO on the RESC dataset, our method runs more than 1.8 times faster than all better-performing methods. This trade-off is also visualized in Fig. 3. Our method is one of the furthest towards low runtimes and high AUROC and AUPR scores, indicating a better trade-off than state-of-the-art methods.

Our approach ends up with roughly 120 to 150 prototypes, while the remaining components vanish. Comparing with similarly sized coresets for PatchCore or the one-shot setting for AnomalyDINO, our approach outperforms both in almost all metrics, while still having a substantially lower runtime compared to

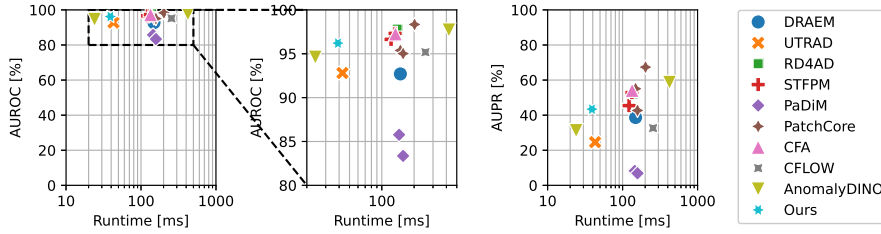


Fig. 3: Visualization of the trade-off between runtime and anomaly segmentation performance on the BraTS2021 dataset. Methods with high anomaly segmentation performance and low runtime are at the top left in each plot.

Table 2: Dice scores for anomaly segmentation for three different thresholds selected to achieve a fixed false positive rate (FPR) on the normal validation set. Results for the one-shot setting are averaged over 25 samples.

	FPR	BraTS2021			BTCV+LiTs			RESC		
		10%	5%	1%	10%	5%	1%	10%	5%	1%
AnomalyDINO (full-shot)		29.26	38.16	54.24	7.19	10.48	18.69	39.28	45.82	52.30
AnomalyDINO (one-shot)		31.89	37.45	32.98	6.17	8.20	10.63	36.48	40.34	33.88
Ours		31.64	41.45	45.87	6.19	8.52	14.76	38.00	43.74	43.49

PatchCore. The implementation of AnomalyDINO is more optimized for efficiency than ours, for instance, they use Faiss [18] for the nearest neighbor search and thus reach faster runtimes. Using a similar optimization, we expect our method to reach even lower runtimes.

We report Dice scores for our approach and both settings of AnomalyDINO in Tab. 2. Not surprisingly, AnomalyDINO in the full-shot setting yields the highest scores for most of the thresholds. However, our approach outperforms the one-shot AnomalyDINO in all but one case, usually by quite some margin. Despite using only a few prototypes, our method demonstrates its effectiveness in reducing the gap to methods with many prototypes.

Table 3: Ablation results for the anomaly score function and for the normalization of the DINOv2 features.

Anomaly Score Function	Feature Normalization	BraTS2021		BTCV+LiTs		RESC	
		AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
Likelihood	False	92.85	23.56	94.62	5.90	85.92	29.04
Euclidean	False	93.52	29.04	93.99	7.54	86.26	35.47
Cosine	False	95.06	38.41	95.30	7.96	89.78	42.52
Likelihood	True	95.49	33.44	95.66	9.92	89.27	33.80
Euclidean	True	96.17	42.68	95.44	9.22	90.15	41.00
Cosine	True	96.21	43.43	95.46	9.02	90.20	41.66

3.3 Ablation Studies

We analyze the choice of cosine distance over Euclidean distance and the log-likelihood for our anomaly metric as well as the choice of normalizing the DINOv2 features. The resulting AUROC and AUPR scores are in Tab. 3. Generally, the results using normalized features are higher than for unnormalized features. The detection with cosine distance leads to the highest scores for most of the experiments, further supporting the findings of our preliminary analysis in Sec. 2.2.

4 Conclusion

In this work, we proposed an unsupervised anomaly detection framework that leverages the expressive power of DINOv2 embeddings and models their distribution using a Dirichlet Process Mixture model (DPMM), which automatically adjusts the number of components to the data at hand. Experimental results on the BMAD benchmark [3] demonstrate that our method achieves competitive performance while significantly reducing the computational cost. Currently, the feature extraction with DINOv2 is the main bottleneck, so future research could explore this setup with distilled models to further improve efficiency.

Acknowledgments. This project was funded by the Swiss National Science Foundation (SNF) (205320_200877).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Antoniak, C.E.: Mixtures of dirichlet processes with applications to bayesian non-parametric problems. *The Annals of Statistics* **2**(6), 1152–1174 (1974)
2. Baid, U., Ghodasara, S., Mohan, S., Bilello, M., Calabrese, E., et al.: The RSNA-ASNR-MICCAI BraTS 2021 benchmark on brain tumor segmentation and radiogenomic classification. *arXiv preprint arXiv: 2107.02314* (2021)
3. Bao, J., Sun, H., Deng, H., He, Y., Zhang, Z., Li, X.: BMAD: Benchmarks for medical anomaly detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. pp. 4042–4053 (2024)
4. Bilic, P., Christ, P., Li, H.B., Vorontsov, E., Ben-Cohen, A., et al.: The liver tumor segmentation benchmark (LiTS). *Medical Image Analysis* **84**, 102680 (2023)
5. Chen, L., You, Z., Zhang, N., Xi, J., Le, X.: UTRAD: Anomaly detection and localization with u-transformer. *Neural Networks* **147**, 53–62 (2022)
6. Chen, X., Konukoglu, E.: Unsupervised detection of lesions in brain MRI using constrained adversarial auto-encoders. *arXiv preprint arXiv:1806.04972* (2018)
7. Chen, X., Pawlowski, N., Glocker, B., Konukoglu, E.: Normative ascent with local gaussians for unsupervised lesion detection. *Medical Image Analysis* **74**, 102208 (2021)
8. Chen, X., You, S., Tezcan, K.C., Konukoglu, E.: Unsupervised lesion detection via image restoration with a normative prior. *Medical Image Analysis* **64**, 101713 (2020)

9. Cohen, N., Hoshen, Y.: Sub-image anomaly detection with deep pyramid correspondences. arXiv preprint arXiv: 2005.02357 (2021)
10. Damm, S., Laszkiewicz, M., Lederer, J., Fischer, A.: AnomalyDINO: Boosting Patch-based Few-shot Anomaly Detection with DINOv2. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (2025)
11. Defard, T., Setkov, A., Loesch, A., Audigier, R.: PaDiM: A patch distribution modeling framework for anomaly detection and localization. In: Pattern Recognition. ICPR International Workshops and Challenges. pp. 475–489 (2021)
12. Deng, H., Li, X.: Anomaly detection via reverse distillation from one-class embedding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9737–9746 (2022)
13. Ferguson, T.S.: Bayesian density estimation by mixtures of normal distributions. In: Recent Advances in Statistics, pp. 287–302 (1983)
14. Gudovskiy, D., Ishizaka, S., Kozuka, K.: CFLOW-AD: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 98–107 (2022)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
16. Hu, J., Chen, Y., Yi, Z.: Automated segmentation of macular edema in OCT using deep neural networks. Medical Image Analysis **55**, 216–227 (2019)
17. Huang, C., Kang, Z., Wu, H.: A Prototype-Based Neural Network for Image Anomaly Detection and Localization. Neural Processing Letters **56**(3), 169 (2024)
18. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. IEEE Transactions on Big Data **7**(3), 535–547 (2019)
19. Kimura, T., Tokuda, T., Nakada, Y., Nokajima, T., Matsumoto, T., Doucet, A.: Expectation-maximization algorithms for inference in Dirichlet processes mixture. Pattern Analysis and Applications **16**(1), 55–67 (2013)
20. Kirichenko, P., Izmailov, P., Wilson, A.G.: Why normalizing flows fail to detect out-of-distribution data. In: Advances in Neural Information Processing Systems. vol. 33, pp. 20578–20589 (2020)
21. Landman, B., Xu, Z., Igelsias, J., Styner, M., Langerak, T., Klein, A.: Miccai multi-atlas labeling beyond the cranial vault—workshop and challenge. In: Proc. MICCAI Multi-Atlas Labeling beyond Cranial Vault—Workshop Challenge. vol. 5, p. 12 (2015)
22. Lee, S., Lee, S., Song, B.C.: CFA: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. IEEE Access **10**, 78446–78454 (2022)
23. Liu, M., Jiao, Y., Lu, J., Chen, H.: Anomaly detection for medical images using teacher–student model with skip connections and multiscale anomaly consistency. IEEE Transactions on Instrumentation and Measurement **73**, 1–15 (2024)
24. Lu, S., Zhang, W., Zhao, H., Liu, H., Wang, N., Li, H.: Anomaly detection for medical images using heterogeneous auto-encoder. IEEE Transactions on Image Processing **33**, 2770–2782 (2024)
25. Nalisnick, E., Matsukawa, A., Teh, Y.W., Gorur, D., Lakshminarayanan, B.: Do deep generative models know what they don’t know? In: International Conference on Learning Representations (2019)
26. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., et al.: DINOv2: Learning robust visual features without supervision. Transactions on Machine Learning Research (2024)

27. Rashmi, K., Das, A., Matcha, N., Ram, K., Sivaprakasam, M.: Ano-swinMAE: Unsupervised anomaly detection in brain MRI using swin transformer based masked auto encoder. In: *Medical Imaging with Deep Learning* (2024)
28. Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., Gehler, P.: Towards total recall in industrial anomaly detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 14318–14328 (2022)
29. Schlegl, T., Seeböck, P., Waldstein, S.M., Langs, G., Schmidt-Erfurth, U.: F-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. *Medical Image Analysis* **54**, 30–44 (2019)
30. Wang, G., Han, S., Ding, E., Huang, D.: Student-teacher feature pyramid matching for anomaly detection. In: *The British Machine Vision Conference (BMVC)* (2021)
31. Wyatt, J., Leach, A., Schmon, S.M., Willcocks, C.G.: AnoDDPM: Anomaly detection with denoising diffusion probabilistic models using simplex noise. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. pp. 650–656 (2022)
32. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: *Proceedings of the British Machine Vision Conference (BMVC)*. pp. 87.1–87.12 (2016)
33. Zavrtanik, V., Kristan, M., Skočaj, D.: DRAEM - a discriminatively trained reconstruction embedding for surface anomaly detection. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. pp. 8330–8339 (2021)
34. Zimmerer, D., Isensee, F., Petersen, J., Kohl, S., Maier-Hein, K.: Unsupervised anomaly localization using variational auto-encoders. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*. pp. 289–297 (2019)