ISSN 1213-6972 (print)
ISSN 1213-6964 (online)
Journal of WSCG
http://www.wscg.eu
Vol.33, No-1-2, 2025

Detecting Out-Of-Distribution Labels in Image Datasets With Pre-trained Networks

Susanne Wulz
Graz University of Technology
Institute of Visual Computing
Austria, Graz

Ulrich Krispel
Fraunhofer Austria Research GmbH
Graz University of Technology
Institute of Visual Computing
Austria, Graz
ulrich.krispel@fraunhofer.at

ABSTRACT

Ensuring the correctness of annotations in training datasets is one way to increase the trustworthiness and reliability of Machine Learning. This study aims to detect semantic shifts in datasets using Feature-Based Out-Of-Distribution and outlier detection methods, assuming Out-Of-Distribution samples are far from In-Distribution data. The experiments began with distance-based methods, such as k-Nearest Neighbours and Mahalanobis, followed by feature pyramids and dimensionality reduction techniques to address high-dimensional challenges. The results showed that the k-Nearest Neighbours detector performed robustly, achieving 100% AUROC when using ResNet50 on the Caltech-101 dataset, while the Mahalanobis detector showed unstable results with scores close to 50%. Moreover, selecting the right backbone model and feature levels, particularly low-level features from ResNet50, improved performance achieving AUROC score of 96% on the DelftBikes dataset for both k-Nearest Neighbours and Local Outlier Factor. The study highlights that k-Nearest Neighbours, Local Outlier Factor, alongside feature pyramids and dimensionality reduction constitute an effective setup for Out-of-Distribution detection, but optimal performance depends on tailored configurations across varying data conditions.

Keywords

Machine Learning · Neural Networks · Convolutional Neural Networks · Feature-Based Out-Of-Distribution Detection · Distance-Based Methods · Outlier Detection · Local Outlier Factor ·

1 INTRODUCTION

In recent years, Neural Networks (NN) have been incorporated into numerous fields, such as autonomous driving [7], medical diagnosis [22], and smart buildings [21], driving significant improvements in daily life. In applications where even small errors can have catastrophic consequences, such as self-driving cars or medical decision-making, the reliability of Machine Learning (ML) models becomes crucial. One of the key factors affecting model performance is the quality of the training datasets used. Models trained on poor-quality, noisy, or incorrectly annotated data can make unreliable predictions, undermining their utility and trustworthiness. One key aspect that determines a dataset's quality is the correctness of annotations of its samples. The reliance on large datasets, often collected from diverse sources, introduces the risk of annotation errors and distributional shifts, which can significantly degrade the performance of ML models [25]. These issues can go undetected unless robust mechanisms are in place to identify such anomalies. Finding and, as a next step, improving the quality of training datasets through automated Out-of-Distribution (OOD) detection methods represents a vital step toward enhancing model reliability. This study explores the potential of feature-based OOD detection techniques for identifying annotation errors in image datasets. It leverages distance-based methods, including K-Nearest Neighbors (k-NN) and Mahalanobis distance, as well as density-based approaches like the Local Outlier Factor (LOF). Furthermore, feature pyramids, which aggregate multilevel representations, and dimensionality reduction techniques, such as UMAP, are explored to improve detection accuracy in high-dimensional feature spaces.

2 RELATED WORK

This section covers key concepts in OOD detection, including feature extraction networks, distance-based methods like Mahalanobis, k-NN, and LOF, and the role of dimensionality reduction and feature pyramids in improving accuracy. Unlike classification-based methods, the feature-based OOD detection approach uses feature embeddings from pre-trained neural networks, making it a post-hoc method that reduces computational cost [25]. The method can be parametric, assuming a Gaussian distribution in feature space, as in Mahalanobis distance-based approaches [18], or non-parametric, making no distributional assumptions

for greater flexibility [23]. In the parametric case, Mahalanobis distance defines confidence scores based on proximity to class distributions, while in the non-parametric case, k-NN distance thresholds are used to detect OOD samples [16].

The **Local Outlier Factor (LOF)** is an unsupervised machine learning algorithm that detects outliers based on local density differences, assigning outlier scores by comparing the local reachability density of a sample to that of its neighbors [1]. While effective for anomaly detection, LOF, like k-NN, suffers from the curse of dimensionality, which limits its performance in high-dimensional feature spaces.

High-dimensional feature spaces can hinder distance-based OOD detection methods due to the curse of dimensionality [15], and **dimensionality reduction techniques** like PCA, LDA, and UMAP are used to address this issue by removing redundant features and improving detection accuracy [24]. In this work, UMAP is employed to project high-dimensional feature embeddings into a lower-dimensional space while preserving their structure, with key parameters such as the number of neighbors and minimum distance affecting its performance [19].

Convolutional Neural Networks (CNNs) are widely used in feature-based OOD detection to extract feature embeddings that help distinguish OOD from in-distribution (ID) samples, with layers capturing essential patterns from low-level edges to high-level object components [27]. Common feature extraction networks include ResNet, DenseNet, EfficientNet, VisFormer, and Vision Transformer, each offering unique advantages, such as ResNet's residual learning for deep networks [8], DenseNet's improved feature propagation [11], and Vision Transformer's use of image patch embeddings [4].

Pre-trained models, particularly those trained on ImageNet [3], are biased towards recognizing textures rather than shapes, with CNNs struggling when texture and shape cues conflict [6]. Research by Hermann et al. [10] and Islam et al. [12] found that shape information is primarily extracted in later layers of CNNs, and aggregating features from multiple stages, such as through feature pyramids, can mitigate the loss of shape information and improve performance.

Feature pyramids have gained popularity in computer vision tasks like object detection and image recognition due to their ability to aggregate multi-scale features, with earlier layers capturing high-resolution features and later layers encoding more context at lower resolutions [2, 17, 26]. However, excessive feature aggregation can lead to redundancy, and skipping connections have been shown to improve performance, especially when removing high-level features to mitigate the ImageNet bias [6, 11].

This study uses the **PyTorch-OOD** library for detecting OOD samples, which follows a three-stage approach: training a Deep Neural Network (DNN), designing an OOD detector around it, and evaluating the detector [14]. The library includes state-of-the-art detectors like k-NN, based on scikit-learn's implementation [20], the Mahalanobis distance for multivariate Gaussian estimation. Due to its absence in the PyTorch-OOD library, we utilized the implementation of the LOF algorithm from scikit-learn.

3 METHOD

Three workflows were employed to detect ID and OOD samples using distance-based methods. Each workflow involves extracting feature embeddings from images with a pre-trained model, followed by the application of distance-based techniques to identify OOD samples. The steps for each workflow were as follows: first, we imported the pre-trained models using the timm library and removed the classification layer to extract feature embeddings, applying model-specific transformations to the imported images. Next, we imported the dataset; for detectors like k-NN and Mahalanobis, we created both a training and a test set, while for LOF, only a test set was needed as no training was required. Finally, detection was performed, followed by evaluation using metrics such as the area under the receiver operating characteristic curve (AUROC), the area under the precision-recall curve (AUPR), and the false positive rate at 95% true positive rate (FPR95TPR) scores. AUROC was chosen as the comparison metric as it reflects the combined classification behavior. It can be sensitive to class imbalances, therefore, the minority class (out of distribution) was chosen as the positive label. AUPR provides a more informative comparison of model performance in the presence of class imbalance, as it focuses on the trade-off between precision and recall. Since the base rate of the positive class heavily influences the AUPR score, it is important to specify which class is considered positive. In this context, AUPRIN refers to treating in-distribution samples as the positive class, while AUPROUT refers to treating out-of-distribution samples as positive [9]. The average precision (AP) is computed as the weighted average of precision values across different thresholds, where weights are the increases in recall. The FPR95TPR score examines the number of false positives a model generates when the true positive rate is at 95%.

In the **Mahalanobis and k-NN-based workflow**, we used various pre-trained models, including ResNet50, DenseNet, EfficientNetV2, and Vision Transformer, to extract feature embeddings. For this experiment, we selected the "faces" class as the ID class and the "bonsai" class as the OOD class from the Caltech-101 dataset. To ensure the validity of the experiment, we verified that these classes were not included in the ImageNet-1K dataset, as all the pre-trained models were trained on ImageNet-1K. Models are trained on examples of the ID class, and tested against a test set consisting of examples from the ID class and the OOD class. Additionally, we conducted experiments where we polluted the training set with a number of OOD samples, specifically 0%, 1%, 5%, and 10% - to assess their impact on the detector's performance.

In both the **k-NN (supervised) workflow** and the **LOF (unsupervised) workflow**, we incorporated feature pyramids and applied dimensionality reduction using UMAP, to assess whether these approaches could enhance performance. Furthermore, the outliers for multiple dataset classes were analyzed within these two workflows.
To create the feature pyramid, we extended the framework with a custom class that leverages layers from conv1 to layer of the ResNet50, capturing features at varying levels of abstraction. These lower-level feature maps are resized to match the spatial resolution of higher-level maps, and then concatenated along the channel dimension. The concatenated multi-scale feature maps are subsequently flattened into a single vector for further processing.

3.1 Datasets

In this study two datasets were used for the experiments: Caltech-101 and DelftBikes. The following section provides an overview of each dataset.
The Caltech-101 [5] dataset is a well-known benchmark dataset for object detection and consists of 101 categories with 40 to 800 images per class, most around 50. The size of these images is approximately 300 x 200 pixels and vary in background, orientation, and scale, which makes it difficult for model evaluation. Notable classes include faces, saxophones, llamas.
The DelftBikes dataset [13], consisting of around 8,000 bicycle images with 22 annotated parts per image, was used in our experiments. Each part is labeled by condition (intact, absent, occluded, or damaged), enabling precise segmentation. A key challenge was defining inliers and outliers when using the LOF detector, as bounding boxes often included unintended parts or excluded relevant details. Non-overlapping components, such as the saddle or steering, were preferred for LOF-based detection, as they allowed more accurate inlier-outlier classification.

4 EVALUATION

We conduct three experiments on two datasets (Caltech-101 and DelftBikes) to evaluate the workflows: **Experiment 1** (Mahalanobis and k-NN-based workflow, supervised) solely uses distance-based methods, such as the k-NN and Mahalanobis detector to evaluate a classfier to distinguish between ID and OOD for classes of the Caltech-101 dataset. We initially trained the detectors using a training set consisting solely of ID data. This was followed by examinations where we progressively increased the proportion of OOD samples in the training set.
In **Experiment 2** (k-NN workflow, supervised), feature pyramids and dimensionality reduction techniques were introduced and combined with the k-NN detector to classify samples in the DelftBikes data set. The class used was saddles, where the ID class is based on intact images, and the OOD class consists of images of the absent class. For the setup, we used the ResNet50 as a backbone, and the k-NN detector, 150 ID images to fit the detector, 75 images per class for testing. The training and test size remained the same for all subsequent experiments.
Subsequently, the setup for **Experiment 3** (LOF workflow, unsupervised) included the LOF detector to detect outliers in several classes of the DelftBikes dataset (steer, back wheel, back light, and saddle class), which are known to contain label errors. Unlike the other detectors, this detector did not require training and performed the outlier detection directly on the test data set. For evaluation, the images had to be manually labeled as outliers or inliers for the test set. A challenge was deciding which images to label as outliers, as some bounding boxes included too many or too few parts. Additionally, distinct appearances in parts, especially for the saddle and back light classes, contributed to further labeling difficulties.
The summary of the experiment evaluations, i.e. the best results, is presented in Table 1.

4.1 Experiment 1 (Mahalanobis and k-NN workflows, supervised)

As seen in Table 2, the k-NN detector's performance is consistently high with average precision scores close to or at 1.00. In contrast, the Mahalanobis detector performs moderately, with average precision scores ranging from 0.39 to 0.77. It shows variability across models. Moreover, it can be observed that a small amount of OOD data in the training set is beneficial for the Mahalanobis detector whereas it is not affecting the k-NN detector's performance. Conversely, introducing too much OOD data can overwhelm the Mahalanobis detector but only impacts the k-NN detector's performance negatively for some models.
In Figure 1 the precision-recall curve can be observed for both detectors using ResNet50 as a backbone. The average precision (AP) score indicates optimal performance for the k-NN detector while illustrating mediocre performance for the Mahalanobis detector.
From these results, it can be derived that the k-NN detector is highly effective for OOD detection across different models and configurations whereas Mahalanobis performs moderately and inconsistently. This is due to the fact that the ID and OOD samples in this experiment ensured minimal overlap in the feature space which demonstrates that k-NN can perform very well when the features are distinct enough.
The results of the Mahalanobis detector indicate that the ID features might not conform to a Gaussian distribution, which lead to the detector not being able to estimate the boundaries of the features. Additionally, high-dimensional data has a negative impact on the detector's performance, as the estimation of the covariance matrix is inaccurate. Therefore, for the subsequent experiments using the k-NN detector, i.e. a non-parametric approach, is preferable for its robustness and reliability when it comes to OOD detection.

Table 1: Overview of experiments, detectors, and the best AUROC score for different configurations using ResNet50 as a backbone.

Exp.	Description	Detector	Dims.	Layers	AUROC
Exp 1	Bonsai and Face (Caltech-101) with 0% OOD in training data	k-NN	-	-	100
Exp 1	Bonsai and Face (Caltech-101) with 1% OOD in training data	k-NN	-	-	100
Exp 1	Bonsai and Face (Caltech-101) with 5% OOD in training data	k-NN	-	-	100
Exp 1	Bonsai and Face (Caltech-101) with 10% OOD in training data	k-NN	-	-	100
Exp 2	Baseline, Saddle (DelftBikes)	k-NN	-	-	85.14
Exp 2	Saddle (DelftBikes)	k-NN	30	3	96.30
Exp 3	Steer (DelftBikes)	LOF	10	5	96.40
Exp 3	Back Wheel (DelftBikes)	LOF	10	1	96.15
Exp 3	Back Light (DelftBikes)	LOF	10	4	54.94
Exp 3	Saddle (DelftBikes)	LOF	20	1	67.02

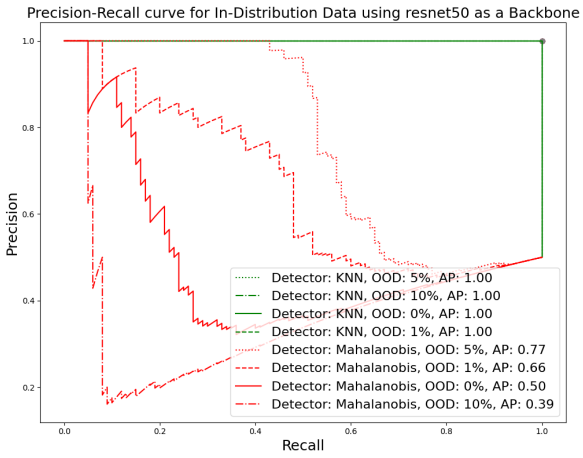

Figure 1: (Experiment 1) Precision-recall curves for ID data using the ResNet50 as a backbone, Caltech-101 as the dataset, the k-NN and Mahalanobis detector with various pollutions of OOD samples in the training set.

4.2 Experiment 2 (k-NN workflow, supervised)

This experiment contains an analysis of various methods of feature extraction and dimensionality reduction. Our results are compared to a baseline with an AUROC of 85.14, AUPR-IN of 81.31, AUPR-OUT of 88.52, and FPR95TPR of 37.33. For the baseline, we used the ResNet50 as a backbone, and the k-NN detector without feature pyramids or dimensionality reduction. For the remaining experiments, both feature pyramids and dimensionality reduction were applied. Figure 2 shows the results for the ResNet50 backbone trained exclusively on ImageNet-1K. The analysis focuses on AUROC as the main evaluation metric, with layers indicating the number of layers in the feature pyramid. The configuration using 3 layers and 30 dimensions provides the best performance in AUROC. This suggests that earlier layers, capturing low-level features like edges, are sufficient for this dataset. Additionally, using 30 dimensions preserves both global and local structures of high-dimensional data. A dimensionality trade-off is observed, where smaller dimensions capture essential features, while higher dimensions can introduce noise. Increasing the number of layers negatively affects performance, likely due to ImageNet bias. In Figure 3, we compare the detector's performance with and without applying dimensionality reduction when

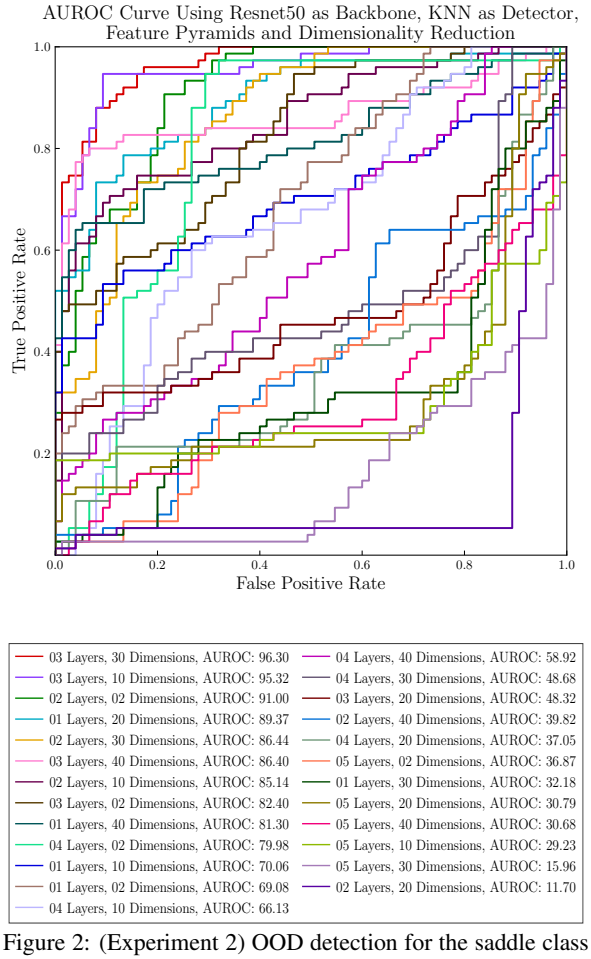

Figure 2: (Experiment 2) OOD detection for the saddle class of DelftBikes, showing ROC curves for ResNet50 pre-trained on ImageNet-1K using different configurations concerning the the number of used layers and dimensional feature reduction using UMAP. A wide variation in performance can be observed.

applying the detector on the features obtained from the first 3 layers. AUROC performance is observed to increase by a large margin from 47.91% without reduction to 95.32% with reduction to 10 dimensions and 96.30% with reduction to 30 dimensions, respectively.

4.3 Experiment 3 (LOF workflow, unsupervised)

The detector was tested on classes with varying configurations and appearances, revealing that parts with diverse angles and appearances were harder to detect accurately. For example, the steer class, which had

Table 2: Results of models and detectors at various OOD pollution rates in the training set (Experiment 1).

Model & Detector		0% OOD				1% OOD				5% OOD				10% OOD			
Model	**Detector**	AUROC	AUPR -IN	AUPR -OUT	FPR95 TPR	AUROC	AUPR -IN	AUPR -OUT	FPR95 TPR	AUROC	AUPR -IN	AUPR -OUT	FPR95 TPR	AUROC	AUPR -IN	AUPR -OUT	FPR95 TPR
resnet50	KNN	100.00	100.00	100.00	0.00	100.00	100.00	100.00	0.00	100.00	100.00	100.00	0.00	100.00	100.00	100.00	0.00
resnet50	Mahalanobis	50.00	56.00	69.00	100.00	61.00	69.43	72.87	100.00	69.50	77.80	78.83	100.00	37.50	36.76	62.82	100.00
resnet18	KNN	100.00	100.00	100.00	0.00	100.00	100.00	100.00	0.00	100.00	100.00	100.00	0.00	100.00	100.00	100.00	0.00
resnet18	Mahalanobis	52.50	58.53	72.50	100.00	57.50	70.00	75.89	100.00	59.00	67.37	72.16	100.00	60.50	69.67	75.18	100.00
vit_base_patch14_dinov2.lvd142m	KNN	95.50	95.87	100.00	9.00	95.50	95.87	100.00	9.00	94.00	94.64	100.00	12.00	93.50	94.25	100.00	13.00
vit_base_patch14_dinov2.lvd142m	Mahalanobis	64.00	77.44	78.30	100.00	63.50	73.61	76.91	100.00	51.50	57.25	73.91	100.00	62.00	79.33	78.20	100.00
densenet201	KNN	100.00	100.00	100.00	0.00	100.00	100.00	100.00	0.00	100.00	100.00	100.00	0.00	100.00	100.00	100.00	0.00
densenet201	Mahalanobis	43.50	37.20	68.66	100.00	51.00	55.92	74.02	100.00	52.50	60.83	74.43	100.00	50.50	54.03	74.13	100.00
efficientnetv2_rw_s.ra2_in1k	KNN	98.84	98.32	99.26	2.00	98.82	98.28	99.25	2.00	98.14	97.30	98.75	5.00	96.71	94.14	97.96	6.00
efficientnetv2_rw_s.ra2_in1k	Mahalanobis	57.00	67.17	75.10	100.00	55.50	68.22	75.52	100.00	53.00	63.21	74.81	100.00	57.00	68.46	75.51	100.00
visformer_small.in1k	KNN	100.00	100.00	100.00	0.00	100.00	100.00	100.00	0.00	99.42	99.15	99.63	1.00	99.41	99.14	99.61	1.00
visformer_small.in1k	Mahalanobis	50.50	53.94	72.64	100.00	61.50	70.73	75.56	100.00	52.50	58.57	72.01	100.00	44.50	39.54	69.20	100.00

many outliers such as pedals, showed strong performance with higher nearest neighbors, achieving an AUROC score of 96.40%. The back wheel class, which had more consistent angles, also performed well with a high AUROC score of 96.15%, and outliers like mislabeled front wheels were successfully detected. Conversely, the saddle and back light classes performed poorly due to diverse appearances, lighting conditions, and varying part types. The back light class achieved an AUROC score of 54.94%, while the saddle class reached 67.02%. The results indicated that increasing the number of neighbors helped performance for more complex configurations, while simpler configurations worked better for the saddle class. In general, the model struggled to distinguish between the saddle and back light classes, with most scores falling below 50%. These findings suggest that for certain classes, like saddles and back lights, the model's ability to differentiate between outliers is limited.

5 CONCLUSION

This study evaluated feature-based Out-of-Distribution (OOD) detection methods, including k-NN, Mahalanobis, and Local Outlier Factor (LOF), using pre-trained model embeddings. The results showed that k-NN performed robustly, particularly when combined with feature pyramids and dimensionality reduction techniques; it requires a training set of clean labels. In the case where no ground truth is known, the LOF method is applicable, but it struggled with classes exhibiting high visual variability. The

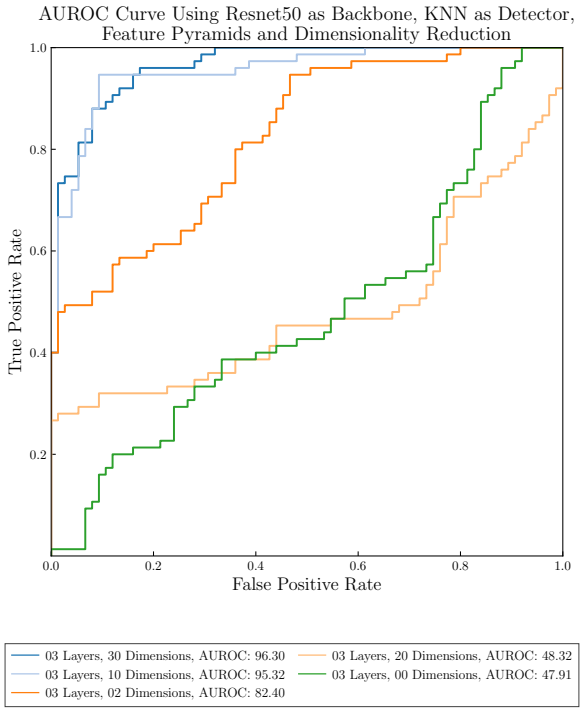

Figure 3: (Experiment 2) Evaluating the impact of dimensionality reduction for OOD detection: using a feature pyramid of the activations from the first 3 convolutional layer, we observe a large increase in OOD performance when using UMAP to reduce the features to 30 or 10 dimensions.

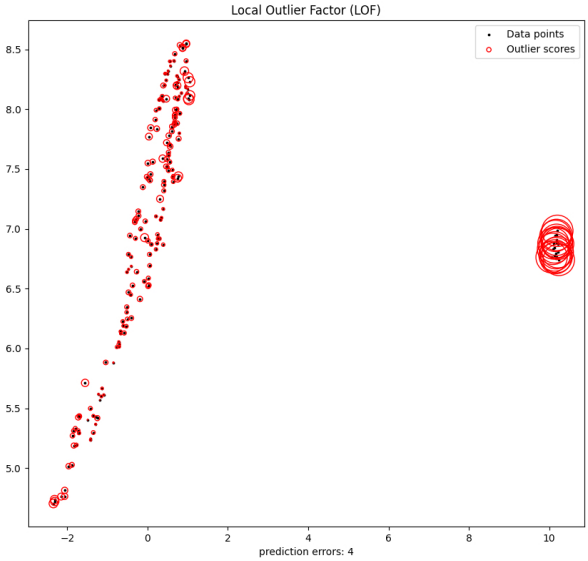

Figure 4: (Experiment 3) A visualization of outliers detected using the LOF algorithm on features obtained via UMAP for dimensionality reduction. A high diameter indicates outliers and the prediction error denotes the number of samples that were wrongly predicted by the detector.

Mahalanobis detector was highly sensitive to class variations, often resulting in performance comparable to random guessing.

Dimensionality reduction, particularly UMAP, significantly improved OOD detection by enhancing feature separation. However, these approaches also introduced trade-offs, including the potential loss of critical information in visually complex classes (for dimensionality reduction) or increased computational complexity (for feature pyramids). Future research could explore advanced techniques to address these limitations, such as ensemble models, feature selection methods, and more sophisticated explainability tools.

These findings contribute to improving OOD detection and refining Machine Learning (ML) models for real-world applications. By understanding the strengths and weaknesses of different approaches, researchers can develop more effective solutions for reliable model deployment.

REFERENCES

[1] Markus M. Breunig, Hans-Peter Kriegel, et al. “LOF: identifying density-based local outliers”. In: *SIGMOD Rec.* (2000).

[2] Niv Cohen and Yedid Hoshen. *Sub-Image Anomaly Detection with Deep Pyramid Correspondences*. 2021. arXiv: 2005.02357 [cs.CV].

[3] Jia Deng, Wei Dong, et al. “ImageNet: A large-scale hierarchical image database”. In: *CVPR’09*. 2009.

[4] Alexey Dosovitskiy, Lucas Beyer, et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *ICLR’20*. 2020.

[5] L. Fei-Fei, R. Fergus, et al. “One-Shot Learning of Object Categories”. In: *CVPR’06*. New York, NY, USA: IEEE, 2006.

[6] Robert Geirhos, Patrick Rubisch, et al. “ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness”. In: *ICLR’19*. 2019.

[7] Sorin Grigorescu, Bogdan Trasnea, et al. “A survey of deep learning techniques for autonomous driving”. In: *Journal of Field Robotics* (2019).

[8] Kaiming He, Xiangyu Zhang, et al. “Deep Residual Learning for Image Recognition”. In: *CVPR’16*. 2016.

[9] Dan Hendrycks and Kevin Gimpel. “A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks”. In: *ICLR’17* (2017).

[10] Katherine L. Hermann, Ting Chen, et al. “The origins and prevalence of texture bias in convolutional neural networks”. In: *NIPS’20*. Vancouver, BC, Canada: Curran Associates Inc., 2020.

[11] G. Huang, Z. Liu, et al. “Densely Connected Convolutional Networks”. In: *CVPR’17*. Los Alamitos, CA, USA: IEEE Computer Society, 2017.

[12] Md Amirul Islam, Matthew Kowal, et al. “Shape or texture: Understanding discriminative features in CNNs”. In: *ICLR’21*. 2021.

[13] Osman Semih Kayhan, Bart Vredebregt, et al. “Hallucination in Object Detection - A Study in Visual Part Verification”. In: *ICIP’21*. IEEE. 2021.

[14] Konstantin Kirchheim, Marco Filax, et al. “PyTorch-OOD: A Library for Out-of-Distribution Detection Based on PyTorch”. In: *CVPR’22*. 2022.

[15] Nikolaos Kouiroukidis and Georgios Evangelidis. “The Effects of Dimensionality Curse in High Dimensional kNN Search”. In: *2011 15th Panhellenic Conference on Informatics*. 2011.

[16] Kimin Lee, Kibok Lee, et al. “A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks”. In: *NIPS’18*. Curran Associates, Inc., 2018.

[17] Tsung-Yi Lin, Piotr DollÃ¡r, et al. “Feature Pyramid Networks for Object Detection”. In: *CVPR’17*. 2017.

[18] P. C. Mahalanobis. “On the generalized distance in statistics”. In: *National Institute of Science of India* (1936).

[19] Leland McInnes, John Healy, et al. “UMAP: Uniform Manifold Approximation and Projection”. In: *The Journal of Open Source Software* (2018).

[20] F. Pedregosa, G. Varoquaux, et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* (2011).

[21] Diego Rodríguez-Gracia, María de las Mercedes Capobianco-Uriarte, et al. “Review of artificial intelligence techniques in green/smart buildings”. In: *Sustainable Computing: Informatics and Systems* (2023).

[22] Tanzila Saba, Ahmed Sameh Mohamed, et al. “Brain tumor detection using fusion of hand crafted and deep learning features”. In: *Cognitive Systems Research* (2020).

[23] Yiyou Sun, Yifei Ming, et al. “Out-of-Distribution Detection with Deep Nearest Neighbors”. In: *PMLR’22*. Proceedings of Machine Learning Research. 2022.

[24] S. Velliangiri, S. Alagumuthukrishnan, et al. “A Review of Dimensionality Reduction Techniques for Efficient Computation”. In: *ICRTAC’19*. 2019.

[25] Jingkang Yang, Kaiyang Zhou, et al. “Generalized Out-of-Distribution Detection: A Survey”. In: *International Journal of Computer Vision* (2024).

[26] Fisher Yu, Dequan Wang, et al. “Deep Layer Aggregation”. In: *CVPR’18*. 2018.

[27] Matthew D. Zeiler and Rob Fergus. “Visualizing and Understanding Convolutional Networks”. In: *ECCV’14*. Cham: Springer International Publishing, 2014.

ISSN 1213-6972 (print)
ISSN 1213-6964 (online)
Journal of WSCG
http://www.wscg.eu
Vol.33, No-1-2, 2025

CAD-RAG: A multi-modal retrieval augmented framework for user editable 3D CAD model generation

Ananthakrishnan A
Indian Institute of Technology, Madras
India, 600036, Chennai, Tamil Nadu
ananthu2014@gmail.com

Anush Bharathi
Madras Institute of Technology
India, 600036, Chennai, Tamil Nadu
anushbharathi2411@gmail.com

Dharanivendhan V
Indian Institute of Technology, Madras
India, 600036, Chennai, Tamil Nadu
dharanivendhanv01@gmail.com

Ramanathan Muthuganapathy
Indian Institute of Technology, Madras
India, 600036, Chennai, Tamil Nadu
emry01@gmail.com

ABSTRACT

Computer-Aided Design (CAD) has revolutionized design and manufacturing by enabling precise, complex models in collaborative environments. While similar CAD models with application-specific modifications are often required, designs are typically created from scratch due to challenges in retrieving existing models or generating editable ones. Although parametric CAD modeling has advanced through deep generative approaches treating CAD as a language task to generate user-editable designs, building truly scalable multi-modal datasets and networks tailored for 3D design tasks, particularly in engineering domains remains a significant challenge. Developing such datasets, especially those incorporating images, point clouds and user-like text and hand-drawn sketches is difficult as these modalities demand fine-grained geometric understanding and extensive human-in-the-loop evaluations. While large foundational models like CLIP have improved cross-modal retrieval, they are primarily trained on natural images and fail to capture the geometric and structural complexities inherent to CAD data.
In this paper, we propose a novel multi-modal pipeline for CAD command sequence generation using state-of-the-art Vision-Language Models (VLMs). We introduce a unique multimodal CAD dataset comprising hand-drawn sketches, CAD command sequences, images and basic text prompts. These modalities are integrated through a Multi-modal Retrieval-Augmented Generation (MM-RAG) framework to enable user-editable CAD model retrieval and generation. Our RAG-based pipeline streamlines the CAD design process by enabling iterative, user-guided model generation based on simple sketches or text queries. This approach aims to streamline CAD model design by creating an advanced, end-to-end pipeline that supports design workflows. The dataset and code will be made publicly available at: `https://github.com/ananthu2014/cadrag`.

Keywords

Computer Aided Design(CAD), 3D shape retrieval, Multi-modal dataset

1 INTRODUCTION

Computer-Aided Design (CAD) has been the torchbearer of modern design and manufacturing, transforming traditional workflows with greater precision, pace and efficiency. From small-scale 3D printed artifacts to large machinery and systems, the need for high-quality designs remains crucial. However, the development of skilled designers continues to be essential, necessitating investments in training to ensure that individuals are industry-ready to design using CAD software such as SolidWorks, Fusion 360 and others. With the advent of deep learning and advanced architectures, significant attention has been paid to data-driven approaches that help designers create better designs, with a primary focus on 3D representation learning for retrieval and generation of CAD models [1].

Retrieval systems focus primarily on searching and locating relevant 3D shapes in large databases through semantic/similarity matching. This includes point cloud-based and image-based [2] approaches among others. Given an input and a target modality, training aim to bring similar models closer in the embedding space and dissimilar ones farther apart. 3D model understanding lies at the crux of this problem, where models are trained to extract relevant features from the data and align them within a shared embedding space [3, 4, 5].

Earlier, content-based retrieval (CBR) systems (where relevant items are searched by analyzing their intrinsic features) employed rule-based techniques such as Poisson histograms [6] and Histograms of Orientation [7].

Model	Sketch	Text	Recall				MAP			
			k=1	k=2	k=5	k=10	k=1	k=2	k=5	k=10
Ours (SBIR)	×	✓	7.94	10.42	17.37	26.05	7.94	9.18	10.93	13.12
Ours (TBIR)	✓	×	14.14	22.08	34.74	47.39	14.04	18.11	21.68	23.37
Ours (STBIR)	✓	✓	**18.11**	**24.81**	**37.47**	**48.64**	**18.11**	**21.46**	**24.95**	**26.47**

Table 1: Zero-shot (Baseline) performance for Sketch-based (SBIR), Text-based (TBIR) and Sketch-Text based (STBIR) Image retrieval on our model measured by Recall and Mean Average Precision (MAP) at top-k retrieval for 403 test samples of CAD-RAG dataset.

With the development of learning-based approaches, useful features were extracted and learned from images, videos, sketches, text, point clouds and more, further advancing retrieval systems. Among these modalities, sketches and text are the most practical for user queries, particularly in the CAD domain, though point cloud could also be considered.

Significant developments have been made in sketch-based content retrieval, particularly in Sketch-Based Image Retrieval (SBIR) methods which mostly utilize dual-encoder Siamese networks to map a given sketch to its corresponding image(s) [8]. A major challenge in adopting this approach is the scarcity of large sketch-based CAD datasets corresponding to engineering shapes. Further, until the development of large pre-trained CLIP-like models [9], using text as a query was not feasible due to the unavailability of datasets that correlate text queries with other modalities to enable cross-modal retrieval.

Over the years, considerable attention has been directed toward geometric deep learning due to advancements in architectures capable of learning such complex representations. Many studies have focused on learning 3D representations from discrete forms. ComplexGen [10] reconstructs B-Rep models from point clouds, Sketch2Mesh [11] generates meshes from sketches while SDFusion [12] performs 3D reconstruction and completion in the form of Signed Distance Functions (SDFs) from multi-modal inputs such as images and text. Although these advancements enhance user control over the generation process, the resulting parametric representations remain non-editable, which is undesirable in a design workflow.

Further advances were introduced in DeepCAD [13], enabling the sequential generation of user-editable CAD models by treating modeling as a language-based task. Models such as Point2Cyl [14], Free2CAD [15], OpenECAD [16] and Text2CAD [17] support cross-modal generation, improving user control and bringing significant attention to parametric CAD generation.

With the advent of large models like CLIP [9], their application in self-supervised learning and adaptation to zero-shot downstream tasks has gained significant attention [18]. However, these models were trained on large-scale internet datasets, which differ significantly from the CAD domain leading to reduced zero-shot performance (see Table 1), particularly in text-to-image retrieval. This highlights the need for a comprehensive text dataset tailored to CAD.

Furthermore, the results reveal that the fine-grained nuances of engineering shapes make sketches a more expressive modality, achieving better performance compared to textual queries. This reinforces the need for a quality sketch dataset as well, one that mimics actual user queries. Additionally, large foundational models such as GPT [19] have demonstrated strong generalization capabilities in zero- and few-shot tasks, especially with retrieval-augmented generation (RAG) [20], suggesting promising avenues for CAD applications.

Hence, in this paper, a multi-modal dataset integrating text, point clouds, CAD command sequences, free-hand sketches and images is introduced, created through a combination of human-in-the-loop processes and deep learning methodologies leveraging SOTA foundational models for text and generative models for sketches. Furthermore, a novel RAG-based network is proposed, enabling sequential retrieval and refined generation of user-editable CAD models from simple user prompts. The entire pipeline is designed to be compute-efficient, with training conducted on low-end GPUs such as the NVIDIA RTX 3080 Ti and 4070 Ti.

The key contributions of this paper are:

- A one-of-its-kind multi-modal dataset, incorporating free hand-drawn sketches, command sequences, images, point clouds and 3-level text prompts based on DeepCAD[13].
- A novel multi-modal RAG pipeline, which is perhaps the first work in the field that performs sequential retrieval and generation of user-editable engineering/CAD shapes.

2 RELATED WORKS

2.1 CAD as a Language Task

Wu et al., in their work DeepCAD [13] proposed a Transformer-based autoencoder for the sequential generation of CAD models, treating CAD design similarly to a language task. To achieve this, a dataset was created from the Sketch-and-Extrude subset of the ABC Dataset [21] with a domain-specific language designed