Research Article

PaRCE: Probabilistic and Reconstruction-based Competency Estimation for CNN-based Image Classification

Sara Pohland[1], **Claire Tomlin**[1]

1. University of California, Berkeley, United States

Convolutional neural networks (CNNs) are extremely popular and effective for image classification tasks but tend to be overly confident in their predictions. Various works have sought to quantify uncertainty associated with these models, detect out-of-distribution (OOD) inputs, or identify anomalous regions in an image, but limited work has sought to develop a holistic approach that can accurately estimate perception model confidence across various sources of uncertainty. We develop a probabilistic and reconstruction-based competency estimation (PaRCE) method and compare it to existing approaches for uncertainty quantification and OOD detection. We find that our method can best distinguish between correctly classified, misclassified, and OOD samples with anomalous regions, as well as between samples with visual image modifications resulting in high, medium, and low prediction accuracy. We describe how to extend our approach for anomaly localization tasks and demonstrate the ability of our approach to distinguish between regions in an image that are familiar to the perception model from those that are unfamiliar. We find that our method generates interpretable scores that most reliably capture a holistic notion of perception model confidence.

Corresponding authors: Sara Pohland, spohland@berkeley.edu; Claire Tomlin, tomlin@berkeley.edu

1. Introduction

Convolutional neural networks (CNNs) are very useful in image classification tasks but can fail dramatically with little explanation or warning[1][2][3][4]. To safely deploy CNN-based classification models in real-world systems, it is crucial to accurately assess the confidence level of their predictions. Extensive research has aimed to quantify uncertainty in CNN predictions[5], but these methods are often overly confident for out-of-distribution (OOD) inputs that differ significantly from those seen during training[3][6]. This has led to work focused on detecting inputs outside of the model's training distribution[7] and identifying anomalous regions in an image[8][9][10]. While these methods help identify unfamiliar data, they rely on arbitrary thresholds to make binary decisions about whether an input is anomalous and lack nuanced probabilistic insights to capture the full scope of predictive uncertainty.

doi.org/10.32388/8SPGR0

We propose a probabilistic and reconstruction-based competency estimation (PaRCE) method. Our method integrates multiple aspects of predictive uncertainty into a single, comprehensive score that reflects the probability that a model's prediction is accurate for a given image. The probabilistic nature of this score increases its interpretability and enhances its utility in autonomous decision-making systems that benefit from accurate risk assessment. Additionally, we extend our method to estimate a model's competency at the regional level within an image, providing valuable information for human users and decision-making systems. To evaluate PaRCE, we introduce several new datasets and evaluation schemes, comparing the performance of our method to existing uncertainty quantification, OOD detection, and anomaly localization methods. Our results demonstrate that PaRCE provides more reliable and interpretable competency estimates across various image types, enhancing model trustworthiness and practical applicability.

2. Background & Related Work

CNNs for image classification usually output softmax scores, which can be interpreted as the probability that an image belongs to each of the training classes. The maximum softmax probability can serve as a measure of model confidence, but these probabilities tend to be very close to one[2] and are particularly unreliable for OOD data[3]. These scores can be better-calibrated through temperature scaling[2] and other methods, but these calibrated scores still tend to be overly confident. This has motivated many other approaches to quantify model uncertainty (Sec. 2.1). These methods focus on data/aleatoric uncertainty, arising from complexities of the data (i.e., noise, class overlap, etc.), and model/epistemic uncertainty, reflecting the ability of the CNN to capture the true underlying model of the data[11]. These methods do not tend to capture distributional uncertainty resulting from mismatched training and test distributions[12]. This has led to methods that seek to detect inputs that are OOD (Sec. 2.2). There is also related work that aims to identify and localize anomalous regions in input images that differ from training images (Sec. 2.3).

2.1. Uncertainty Quantification (UQ)

Extensive research has focused on understanding and quantifying uncertainty in a neural network's prediction. The modeling of these uncertainties can be divided into methods based on (1) Bayesian neural networks (BNNs), (2) deterministic networks, and (3) ensembles of networks[5]. A BNN is a stochastic model whose output is a probability distribution over its predictions[13][14]. Approaches that employ BNNs extract uncertainty as a statistical measure over the outputs of the model. Methods that use a single deterministic network often rely on Monte Carlo (MC) dropout as approximate Bayesian inference[15]. Ensemble methods combine the predictions of multiple deterministic networks to form a probability density function[16]. While these methods address the typical overconfidence in neural network predictions with better-calibrated confidence scores, they generally focus on predictions for in-distribution inputs, which come from the same distribution as the training data. These approaches are generally insufficient to appropriately assign confidence scores for OOD inputs that differ significantly from those seen during training[3][6].

2.2. Out-of-Distribution (OOD) Detection

Many recent approaches have focused on quantifying distributional uncertainty and determining if an input falls outside of the input-data distribution. These approaches are generally either (1) classification-based, (2) density-based,

(3) distance-based, or (4) reconstruction-based[7]. Classification-based methods seek to revise the overly confident softmax scores at the output of neural networks to detect OOD samples more robustly[17][18][19][20]. Density-based methods model the training distribution with some probabilistic model and flag test data in low-density regions as OOD[21][22][23][24]. Distance-based methods use distance metrics in the feature space with the assumption that OOD samples should lie relatively far from the centroids or prototypes of the training classes[25][26][27]. Finally, reconstruction-based methods rely on the reconstruction loss of autoencoders (AEs) or generative adversarial networks (GANs), assuming OOD samples will be reconstructed poorly[28][29][30]. While all these methods have successfully identified OOD samples, they do not fully capture the predictive uncertainty associated with CNNs.

2.3. Anomaly Detection & Localization

A field of study closely related to OOD detection is anomaly detection and localization, wherein the goal is to segment the particular pixels containing anomalies[8][9][31][10]. Within the area of anomaly localization, most approaches are (1) reconstruction-based, (2) synthesizing-based, or (3) embedding-based[32]. Reconstruction-based methods use pixel-wise reconstruction errors to detect anomalies under the assumption that networks trained with only “normal” data will not accurately reconstruct anomalous regions[33][34][35]. Synthesizing-based methods estimate the decision boundary between normal and anomalous data by training on synthetic anomalies generated from anomaly-free images[36][37]. Embedding-based methods embed image features into a compressed space and assume that anomalous features are far from the normal clusters in the embedding space[38][39][40][41][42]. Similar to the OOD detection methods, these methods can successfully identify anomalies in an image but cannot fully capture uncertainty.

2.4. Probabilistic Competency Estimation

UQ techniques focus on quantifying data/aleatoric uncertainty and model/epistemic uncertainty, but generally fail to capture uncertainty arising from distributional shift. OOD and anomaly detection techniques better address this type of uncertainty but generally rely on thresholds to generate a binary decision, rather than capturing a holistic measure of uncertainty. We are interested in *perception model competency*–a generalized form of predictive uncertainty for CNNs that combines these various aspects of model uncertainty into a single probabilistic score[43].

3. Definition of PaRCE Score

We define a probabilistic score that captures model, data, and distributional uncertainty arising in a CNN-based perception model (Sec. 3.1), using a reconstruction-based approach to estimate distributional uncertainty (Sec. 3.2). Our method is designed to directly reflect the prediction accuracy of the perception model (Sec. 3.3) and can be extended to generate regional competency images (Sec. 3.4).

3.1. Estimating Model Competency

Let f be the true underlying model of the system from which our images are drawn and $\hat{f}$ be the predicted model (referred to as the perception model or classifier). For an input image, $\mathbf{X}$, the perception model aims to estimate the true class of the image, $f(\mathbf{X})$, from the set of all classes, $\mathcal{C}$. The competency of the model for this image is given by

$$\rho(\mathbf{X}) := P(\{\hat{f}(\mathbf{X}) = f(\mathbf{X})\}|\mathbf{X}). \tag{1}$$

To simplify our notation, let $\hat{c}$ be the class predicted by the perception model (i.e., $\hat{f}(\mathbf{X}) = \hat{c}$) such that

$$\rho(\mathbf{X}) = P(\{f(\mathbf{X}) = \hat{c}\}|\mathbf{X}). \tag{2}$$

Often, the perception model uses the softmax function to obtain an estimate of the probability $P(\{f(\mathbf{X}) = c\}|\mathbf{X})$ for each class $c \in \mathcal{C}$. However, the perception model cannot truly estimate this probability because it is limited by the data contained in the training sample. It instead estimates the probability $P(\{f(\mathbf{X}) = c\}|\mathbf{X}, D)$, where D is the event that the input image is in-distribution (ID) (i.e., drawn from the same distribution as the training samples). Let us then write the following lower bound on competency:

$$\begin{aligned} \rho(\mathbf{X}) &\geq P(D \cap \{f(\mathbf{X}) = \hat{c})\}|\mathbf{X}) && (3) \\ &= P(\{f(\mathbf{X}) = \hat{c}\}|\mathbf{X}, D)P(D|\mathbf{X}). && (4) \end{aligned}$$

We can assume that the perception model provides an estimate of the class probability, but we need a method to estimate the in-distribution probability.

3.2. Estimating In-Distribution (ID) Probability

To estimate the ID probability, we design an autoencoder (AE) to reconstruct the same images used to train the classifier. The AE's encoder mirrors the classifier's architecture, while its decoder is the inverse of the classifier. This architecture establishes a direct connection between the classifier's feature vector and the AE's latent representation, making reconstruction loss a proxy for the classifier's familiarity with an input image and the ability of its learned features to capture the input. When the classifier's features effectively represent an image, the reconstruction loss remains low, indicating higher prediction confidence. Conversely, a high reconstruction loss suggests a discrepancy between the input image and ID data, implying a decrease in confidence.

The probability an image, $\mathbf{X}$, is drawn from the same distribution as those in the training sample is given by

$$P(D|\mathbf{X}) = \sum_{c \in \mathcal{C}} P(D|\{f(\mathbf{X}) = c\})P(\{f(\mathbf{X}) = c\}|\mathbf{X}). \tag{5}$$

Assume now that the classifier provides an estimate of $P(\{f(\mathbf{X}) = c\}|\mathbf{X})$. We can estimate $P(D|\{f(\mathbf{X}) = c\})$ as the probability that the reconstruction loss, $\ell(\mathbf{X})$, for image $\mathbf{X}$ falls within the lowest $N\%$ of losses from the training images. Using a holdout set, $\mathcal{X}$, randomly selected from the same distribution as the training set but not used to train the classifier, we estimate the distribution of reconstruction loss for each class $c \in \mathcal{C}$ as a Gaussian with mean μ_c and standard deviation σ_c. Because $\mathcal{L}_c$ follows a Gaussian distribution, N corresponds to a z-score, z. We estimate the class in-distribution probability, $P(D|\{f(\mathbf{X}) = c\})$, as

$$\hat{P}(D|\{f(\mathbf{X}) = c\}) = 1 - \phi\left(\frac{\ell(\mathbf{X}) - 2\mu_c}{\sigma_c} - z\right), \tag{6}$$

where ϕ denotes the cumulative distribution function (CDF) of the standard normal distribution. (See Eqs. (9) to (13) for the derivation of this estimate.) Let $\hat{p}_c$ be the probabilistic output of the perception model corresponding to class $c \in \mathcal{C}$. We now have the following estimate of competency:

$$\hat{\rho}(\mathbf{X}) := \hat{p}_{\hat{c}} \sum_{c \in \mathcal{C}} \hat{p}_c \left(1 - \phi\left(\frac{\ell(\mathbf{X}) - 2\mu_c}{\sigma_c} - z\right)\right). \tag{7}$$

3.3. Calibrating Competency Score

In Eq. (7), $\hat{p}_c$ is provided by the trained perception model, while μ_c and σ_c are determined by the holdout set drawn from the training distribution. We select the z-score, z, such that the average competency score reflects the prediction accuracy for the ID holdout set, $\mathcal{X}$. To calibrate our competency estimator, we want to find z such that

$$\frac{1}{|\mathcal{X}|}\sum_{\mathbf{X}\in\mathcal{X}}\hat{\rho}(\mathbf{X}) = \frac{1}{|\mathcal{X}|}\sum_{\mathbf{X}\in\mathcal{X}}1_{\{\hat{f}(\mathbf{X})=f(\mathbf{X})\}}(\mathbf{X}). \tag{8}$$

Rather than computing the exact value of z that satisfies this equation, we select a z-score to the nearest five hundredth that results in the average competency score (left side of Eq. (8)) closest to the prediction accuracy (right side of Eq. (8)). (See Fig. 5 for a plot of the calibration curves.)

3.4. Generating Competency Images

Suppose that, in addition to estimating the overall competency score for the image as a whole, we wish to estimate the competency of a model for particular regions in the image. Now, instead of letting $\mathbf{X}$ be the entire input image, $\mathbf{X}$ is a segmented region of the image. In our work, images are segmented using the graph-based image segmentation algorithm developed by Felzenszwalb[44]. We follow roughly the same procedures to estimate the probability that each region in the input image came from the same distribution as the training samples. Now, rather than designing an AE to reconstruct the input image, we design an image inpainting model to reconstruct a missing segment of the image and measure the average reconstruction loss over the pixels corresponding to that image segment. We calibrate these scores again to reflect the prediction accuracy of the perception model. (See Fig. 7 for the calibration curves.) We can then use these scores to generate regional competency images. (See the last column of Figs. 17 to 19.)

4. Analysis of PaRCE Score

We first compare our overall PaRCE score (Sec. 3.1) to existing methods for uncertainty quantification (Sec. 2.1) and OOD detection (Sec. 2.2) in Sec. 4.1. We then compare our regional PaRCE image (Sec. 3.4) to existing methods for anomaly detection and localization (Sec. 2.3) in Sec. 4.2.

4.1. Overall Competency Scores

We compare our overall model competency estimation method against various existing methods for quantifying uncertainty and detecting OOD inputs. In particular, we consider the Maximum Softmax Probability (MSP) baseline, the calibrated MSP with Temperature Scaling[2], Monte Carlo (MC) Dropout[15], Ensembling[16], the Energy Score[17], ODIN[18], OpenMax[19], DICE[20], KL-Matching[25], the Mahalanobis Distance[26], and k-Nearest Neighbors (k-NN)[27]. These methods serve as a representative selection of the many existing approaches and are often used as baselines in other works. Note that we focus on the most popular methods that are easily implemented with the help of the PyTorch-OOD library[45].

4.1.1. Evaluation 1: Manually Identified OOD Samples

Datasets:

We conduct analysis across three unique datasets that contain manually identified OOD samples. The first dataset was obtained from a lunar environment, in which the training set contains images from an uninhabited moon and the test set contains images from a habited moon. The second dataset contains speed limit signs in Germany[46]. While the training dataset is composed of common speed limit signs (30 km/hr and higher), the test set also contains an uncommon speed limit (20 km/hr). The third dataset contains images from regions in a park, in which the training data only contains images from forested and grassy regions of the park, while the test set additionally includes images from around the park's pavilion. See Appendix B for additional details and example images.

Metrics:

We evaluate scoring methods based on their computation time and their ability to distinguish between correctly classified, misclassified, and OOD samples. To quantify the ability to distinguish between sets of samples, we consider the distance between score distributions using the Kolmogorov–Smirnov (KS) test, the extent of overlap between distributions as measured by the area under the receiver operating characteristic (AUROC), and the detection error determined by the false positive rate (FPR) at a 95% true positive rate (TPR), where a true positive indicates the correct identification of a misclassified or OOD sample. See Appendix D.1 for additional details on these metrics.

Method	**ComputationTime (sec) ↓**	**Correct vs. Incorrect**			**Correct vs. OOD**			**Incorrect vs. OOD**		
		Dist. ↑	**AUROC ↑**	**FPR ↓**	**Dist. ↑**	**AUROC ↑**	**FPR ↓**	**Dist. ↑**	**AUROC ↑**	**FPR ↓**
Softmax	**0.0001**	0.72	0.91	1.00	0.45	0.76	1.00	0.13	**0.76**	**1.00**
Temperature[2]	0.0002	0.72	0.91	1.00	0.45	0.76	1.00	0.13	0.75	1.00
MC Dropout[15]	0.2700	0.62	0.86	0.47	0.46	0.80	0.72	0.05	0.79	0.73
Ensemble[16]	0.0668	**0.80**	**0.93**	**0.28**	0.49	0.79	1.00	0.01	0.78	1.00
Energy[17]	0.0002	0.71	0.90	1.00	0.45	0.75	1.00	0.19	0.75	1.00
ODIN[18]	0.1056	0.02	0.40	1.00	0.03	0.48	1.00	0.19	0.49	1.00
OpenMax[19]	0.0017	0.31	0.62	0.87	0.11	0.50	0.87	0.06	0.49	0.87
DICE[20]	0.0129	0.61	0.87	0.50	0.31	0.72	0.81	0.11	0.71	0.81
KL-Matching[25]	0.0004	0.65	0.80	0.83	0.44	0.74	0.88	0.22	0.74	0.88
Mahalanobis[26]	0.1092	0.53	0.80	0.58	0.45	0.81	0.58	0.20	0.80	0.58
k-NN[27]	0.0190	0.59	0.83	0.41	0.61	0.87	0.37	0.29	0.87	0.38
PaRCE (Ours)	0.0270	0.64	0.87	0.59	**0.89**	**0.99**	**0.08**	**0.71**	**0.99**	**0.09**

Table 1. Comparison of measures of overall model competency for the lunar dataset with manually identified OOD samples.

Method	ComputationTime (sec) ↓	Correct vs. Incorrect			Correct vs. OOD			Incorrect vs. OOD		
		Dist. ↑	**AUROC ↑**	**FPR ↓**	**Dist. ↑**	**AUROC ↑**	**FPR ↓**	**Dist. ↑**	**AUROC ↑**	**FPR ↓**
Softmax	**0.0001**	**0.86**	**0.97**	**0.10**	**0.78**	**0.94**	0.18	0.06	**0.93**	0.20
Temperature[2]	0.0002	**0.86**	**0.97**	**0.10**	**0.78**	**0.94**	0.18	0.06	**0.93**	0.20
MC Dropout[15]	6.7872	0.83	0.94	0.17	**0.78**	0.93	**0.17**	0.15	**0.93**	0.20
Ensemble[16]	1.5363	0.77	0.90	0.19	0.68	0.90	0.28	0.17	0.89	0.31
Energy[17]	0.0002	**0.86**	**0.97**	**0.10**	**0.78**	**0.94**	0.18	0.03	**0.93**	0.20
ODIN[18]	0.8650	0.41	0.70	1.00	0.20	0.55	1.00	0.00	0.53	1.00
OpenMax[19]	0.0012	0.53	0.77	0.83	0.14	0.48	0.84	0.03	0.45	0.84
DICE[20]	0.3228	0.74	0.92	0.22	0.77	0.92	0.21	0.16	0.91	0.21
KL-Matching[25]	0.0004	**0.86**	0.96	**0.10**	0.76	**0.94**	0.18	0.00	**0.93**	0.20
Mahalanobis[26]	0.8937	0.34	0.66	0.72	0.50	0.78	0.67	**0.38**	0.79	0.66
k-NN[27]	0.3330	0.45	0.74	0.60	0.55	0.83	0.45	0.28	0.84	0.44
PaRCE (Ours)	0.0928	0.85	**0.97**	0.11	**0.78**	**0.94**	0.18	0.07	**0.93**	**0.19**

Table 2. Comparison of measures of overall model competency for the speed dataset with manually identified OOD samples.

Method	ComputationTime (sec) ↓	Correct vs. Incorrect			Correct vs. OOD			Incorrect vs. OOD		
		Dist. ↑	AUROC ↑	FPR ↓	Dist. ↑	AUROC ↑	FPR ↓	Dist. ↑	AUROC ↑	FPR ↓
Softmax	0.0001	0.35	0.67	1.00	0.33	0.69	1.00	0.07	0.69	1.00
Temperature[2]	0.0002	**0.69**	**0.87**	**0.31**	0.33	0.56	1.00	0.08	0.55	1.00
MC Dropout[15]	0.2030	0.01	0.41	0.99	0.38	0.62	0.97	0.47	0.67	0.96
Ensemble[16]	0.0471	0.44	0.64	1.00	0.61	0.85	1.00	0.36	0.85	1.00
Energy[17]	0.0002	0.35	0.65	1.00	0.31	0.67	1.00	0.04	0.67	1.00
ODIN[18]	0.0621	0.48	0.72	1.00	0.00	0.46	1.00	0.02	0.46	1.00
OpenMax[19]	0.0018	0.17	0.44	1.00	0.54	0.75	0.44	0.80	0.76	0.44
DICE[20]	0.0109	0.26	0.49	0.80	0.07	0.34	0.90	0.03	0.34	0.90
KL-Matching[25]	0.0004	0.24	0.32	1.00	0.42	0.72	0.57	0.72	0.73	0.57
Mahalanobis[26]	0.0727	0.49	0.71	0.75	0.84	0.96	0.27	0.88	**0.96**	0.26
k-NN[27]	0.0141	0.31	0.54	1.00	0.89	0.97	0.10	0.79	0.98	0.09
PaRCE (Ours)	0.0234	0.34	0.58	0.73	**0.94**	**0.98**	**0.03**	**0.91**	**0.99**	**0.04**

Table 3. Comparison of measures of overall model competency for the park dataset with manually identified OOD samples.

Results: The results are summarized in Tables 1 to 3. We display the distribution of competency scores for correctly classified, misclassified, and OOD samples in Figs. 8 to 10.

The MSP baseline is the simplest and fastest method, but it tends to assign scores of one or close to one regardless of whether an image is correctly classified, misclassified, or OOD, implying nearly complete confidence across sample types. Temperature Scaling is another fast and relatively simple method that seeks to calibrate the softmax scores, but the distributions of scores for this method tend to look quite similar to those of the original softmax method, resulting in an insignificant improvement overall. These methods are sufficient to distinguish correctly classified samples from misclassified and OOD samples in the speed dataset, but they do not perform as well for other evaluations.

MC Dropout and Ensembling are related methods commonly used for uncertainty quantification. The performance of these methods is inconsistent across datasets. Both methods generally assign lower scores to misclassified and OOD samples, as compared to correctly classified ones, but this is not always the case. These methods are generally unable to distinguish between misclassified and OOD samples and often assign high scores to OOD images. Note that Dropout is by far the slowest method we consider, and Ensembling is also on the slower side. Furthermore, Ensembling requires significant computational effort during training.

The classification-based OOD detection techniques–Energy, ODIN, OpenMax, and DICE–have varying performance across datasets, but they generally never outperform other methods we consider. The Energy Score tends to be very similar to the Softmax and Temperature Scaling scores. ODIN assigns a maximum score to nearly all points across all datasets. OpenMax and DICE show variations in score distributions for correctly classified, misclassified, and OOD sets but produce scores with a lot of distribution overlap and little difference in average scores.

The distance-based OOD detection techniques–KL-Matching, Mahalanobis, and k-NN–tend to assign lower scores to misclassified and OOD images than to correctly classified images, but the extent of this difference varies across each method and dataset. KL-matching does a good job distinguishing correctly classified images from misclassified and OOD ones, but these three methods generally do not outperform other methods we consider.

Our method, PaRCE, performs the most reliably, outperforming other methods across the majority of evaluations. It is generally not the best at distinguishing between correctly classified and misclassified samples but tends to best distinguish correctly classified and misclassified samples from OOD samples. Note that while all methods, other than the MSP, generate scores with somewhat arbitrary values, our method is probabilistic, generating intuitive scores between zero and one. Our method tends to assign scores very close to one to correctly classified samples, lower scores to misclassified samples, and even lower scores to OOD samples.

4.1.2. Evaluation 2: Generated Data with New Properties

Datasets:

In addition to the spatial anomalies considered in the previous evaluation, we are interested in non-regional anomalies that reduce model confidence and prediction accuracy. For each dataset discussed in 4.1.1, we generate new data from the ID samples by adjusting saturation, contrast, brightness, pixelation, or noise levels. We consider 21 values for each of these image properties, each of which results in either high, medium, or low prediction accuracy. See Appendix B for additional details and examples.

Metrics:

We evaluate methods based on their ability to distinguish between image modifications resulting in high, medium, and low accuracy. To evaluate the ability to distinguish between sets of samples using each scoring method, we consider the same metrics discussed in 4.1.1.

Method	High vs. Medium			High vs. Low			Medium vs. Low		
	Dist. ↑	AUROC ↑	FPR ↓	Dist. ↑	AUROC ↑	FPR ↓	Dist. ↑	AUROC ↑	FPR ↓
Softmax	0.16	0.59	1.00	0.08	0.57	1.00	0.00	0.51	1.00
Temperature[2]	0.24	0.63	**0.75**	0.24	0.62	0.85	0.07	0.56	0.90
MC Dropout[15]	0.23	0.64	1.00	0.30	0.63	1.00	0.08	0.57	1.00
Ensemble[16]	0.31	0.69	1.00	0.36	0.68	1.00	0.12	0.60	1.00
Energy[17]	0.15	0.59	1.00	0.08	0.57	1.00	0.00	0.52	1.00
ODIN[18]	0.00	0.46	1.00	0.00	0.43	1.00	0.00	0.43	1.00
OpenMax[19]	0.12	0.55	0.92	0.26	0.58	0.92	0.15	0.56	0.93
DICE[20]	0.08	0.53	0.98	0.02	0.46	1.00	0.00	0.41	1.00
KL-Matching[25]	0.19	0.59	0.96	0.46	0.65	0.92	0.34	0.64	0.93
Mahalanobis[26]	0.13	0.56	0.96	0.46	0.65	0.94	0.36	0.66	0.92
k-NN[27]	0.11	0.50	1.00	0.45	0.50	1.00	0.37	0.50	1.00
PaRCE (Ours)	**0.47**	**0.80**	0.78	**0.73**	**0.83**	**0.74**	**0.47**	**0.73**	**0.82**

Table 4. Comparison of measures of overall model competency for the lunar dataset with modified image properties.

Method	High vs. Medium			High vs. Low			Medium vs. Low		
	Dist. ↑	AUROC ↑	FPR ↓	Dist. ↑	AUROC ↑	FPR ↓	Dist. ↑	AUROC ↑	FPR ↓
Softmax	0.34	0.71	0.91	0.78	0.80	0.84	0.49	0.76	0.83
Temperature[2]	0.34	0.71	0.91	0.78	0.80	0.84	0.49	0.76	0.83
MC Dropout[15]	0.34	0.70	0.90	0.72	0.77	0.83	0.38	0.70	0.82
Ensemble[16]	0.34	0.70	0.83	0.54	0.73	0.74	0.20	0.63	0.77
Energy[17]	0.34	0.71	0.91	0.78	0.80	0.84	0.49	0.76	0.83
ODIN[18]	0.02	0.49	1.00	0.27	0.55	1.00	0.28	0.58	1.00
OpenMax[19]	0.29	0.66	0.89	0.74	0.75	0.85	0.46	0.72	0.85
DICE[20]	0.32	0.70	0.91	0.79	0.80	0.84	**0.54**	**0.77**	0.83
KL-Matching[25]	0.34	0.71	0.88	0.80	0.81	0.81	0.53	**0.77**	0.82
Mahalanobis[26]	0.00	0.41	0.98	0.00	0.31	0.99	0.00	0.30	0.99
k-NN[27]	0.00	0.34	0.99	0.00	0.23	1.00	0.00	0.24	0.99
PaRCE (Ours)	**0.49**	**0.81**	**0.79**	**0.83**	**0.87**	**0.68**	0.36	**0.77**	**0.72**

Table 5. Comparison of measures of overall model competency for the speed dataset with modified image properties.

Method	High vs. Medium			High vs. Low			Medium vs. Low		
	Dist. ↑	AUROC ↑	FPR ↓	Dist. ↑	AUROC ↑	FPR ↓	Dist. ↑	AUROC ↑	FPR ↓
Softmax	0.11	0.56	1.00	0.18	0.57	1.00	0.12	0.52	1.00
Temperature[2]	0.38	0.52	1.00	0.55	0.54	1.00	0.25	0.53	0.94
MC Dropout[15]	0.14	0.58	0.95	0.44	0.61	0.94	0.31	0.55	0.94
Ensemble[16]	0.30	0.66	1.00	0.76	0.70	1.00	0.50	0.58	1.00
Energy[17]	0.11	0.56	1.00	0.18	0.57	1.00	0.12	0.51	1.00
ODIN[18]	0.01	0.50	1.00	0.12	0.51	1.00	0.12	0.51	1.00
OpenMax[19]	0.11	0.56	0.96	0.13	0.55	0.96	0.03	0.49	0.98
DICE[20]	0.05	0.51	0.96	0.14	0.48	1.00	0.12	0.45	1.00
KL-Matching[25]	0.25	0.58	1.00	0.68	0.62	1.00	0.64	**0.60**	0.94
Mahalanobis[26]	0.09	0.52	0.98	0.70	0.56	0.99	0.65	0.57	0.99
k-NN[27]	0.08	0.50	1.00	0.71	0.49	1.00	**0.67**	0.48	0.99
PaRCE (Ours)	**0.59**	**0.85**	**0.81**	**0.88**	**0.87**	**0.77**	0.37	0.57	**0.93**

Table 6. Comparison of measures of overall model competency for the park dataset with modified image properties.

Results: The results are summarized in Tabs. 4-6. We also display the distribution of competency scores for high, medium, and low accuracy modifications in Figs. 11-13.

The MSP and Temperature Scaling methods perform similarly to each other across all datasets. These methods are not insightful for the lunar or park datasets, assigning a near-maximum score to the majority of samples, regardless of model accuracy. For the speed dataset, these methods more clearly assign different distributions of scores to high, medium, and low accuracy image modifications, and their performance is more similar to other existing methods.

MC Dropout and Ensembling also perform similarly to one another. Their average scores tend to be similar across accuracy levels, but there is generally more variation for medium and low accuracy samples. However, the performance of these methods varies significantly across datasets.

For the classification-based OOD detection methods, there tends to be little variation in score distributions across high, medium, and low accuracy image modifications. They all generate more reasonable score distributions for the speed dataset, but they generally do not outperform previously mentioned methods, even for this dataset.

Looking at distance-based OOD detection methods, KL-matching generally assigns lower scores to low accuracy modifications than to high and medium accuracy ones, indicating some utility of this method. k-NN assigns lower scores on average for the low accuracy images in the lunar and park datasets but higher scores for the speed dataset. The Mahalanobis method assigns similar scores for high, medium, and low accuracy images across all datasets.

Our method performs the best across nearly all metrics for the modified datasets–best distinguishing between high, medium, and low accuracy image modifications. Again, other than the MSP, our method is the only probabilistic one, making it the most intuitive. It tends to assign scores close to one to images with high accuracy modifications, scores close to zero for low accuracy modifications, and scores in between zero and one for medium accuracy.

4.2. Regional Competency Images

We compare our regional competency images to various existing methods for anomaly detection and localization. In particular, we compare our approach to GANomaly[35], DRAEM[36], FastFlow[42], PaDiM[38], PatchCore[39], Reverse Distillation[40], and Student-Teacher Feature Pyramid Matching[41]. These are several of the state-of-the-art anomaly detection algorithms that generate anomaly maps similar to our regional competency maps. All methods were implemented with the help of the Anomalib library[47].

4.2.1. Evaluation: Anomalous Regions of Images

Datasets:

We conduct analysis using the three datasets discussed in 4.1.1. These datasets allow us to assess the ability of methods to identify unfamiliar objects, detect regions associated with unseen classes, and recognize unexplored areas in an environment. See Appendix B for details.

Metrics:

We evaluate methods based on their computation time and their ability to distinguish between familiar regions (in both ID and OOD images) and unfamiliar regions (in OOD images). Familiar regions are all of the pixels that occupy image structures that exist in the training set, and unfamiliar pixels are those that occupy structures that were not present during training. To evaluate the ability to distinguish between sets of samples using each mapping method, we consider the same metrics discussed in Sec. 4.1.1.

Method	Computation Time (sec) ↓	ID All vs. OOD Unfamiliar			OOD Familiar vs. OOD Unfamiliar		
		Dist. ↑	AUROC ↑	FPR ↓	Dist. ↑	AUROC ↑	FPR ↓
GANomaly[35]	0.074	0.11	0.58	0.91	0.11	0.56	0.92
DRAEM[36]	0.197	0.36	0.68	0.97	0.46	0.69	0.97
FastFlow[42]	0.036	0.84	0.97	0.15	0.74	0.96	0.17
PaDiM[38]	0.026	0.89	**0.98**	0.07	0.80	**0.98**	0.09
PatchCore[39]	0.640	**0.94**	0.59	0.83	0.78	0.57	0.86
Reverse Distillation[40]	0.083	0.85	0.95	0.10	0.80	0.95	0.11
Student-Teacher[41]	**0.014**	0.79	0.95	0.33	0.75	0.95	0.32
PaRCE (Ours)	0.122	0.89	**0.98**	**0.06**	**0.87**	**0.98**	**0.07**

Table 7. Comparison of measures of regional model competency for the lunar dataset with anomalous regions.

Method	ComputationTime (sec) ↓	ID All vs. OOD Unfamiliar			OOD Familiar vs. OOD Unfamiliar		
		Dist. ↑	AUROC ↑	FPR ↓	Dist. ↑	AUROC ↑	FPR ↓
GANomaly[35]	0.075	0.15	0.55	0.80	0.08	0.55	0.80
DRAEM[36]	0.118	0.22	0.62	0.85	0.19	0.62	0.85
FastFlow[42]	0.035	0.43	0.77	0.78	0.34	0.77	0.78
PaDiM[38]	0.025	0.58	0.86	**0.42**	0.43	0.86	**0.44**
PatchCore[39]	0.870	0.45	0.50	1.00	0.33	0.50	1.00
Reverse Distillation[40]	0.068	0.22	0.62	0.79	0.16	0.62	0.79
Student-Teacher[41]	**0.014**	0.32	0.69	0.72	0.18	0.69	0.72
PaRCE (Ours)	0.145	**0.75**	**0.89**	0.54	**0.52**	**0.89**	0.54

Table 8. Comparison of measures of regional model competency for the speed dataset with anomalous regions.

Method	ComputationTime (sec) ↓	ID All vs. OOD Unfamiliar			OOD Familiar vs. OOD Unfamiliar		
		Dist. ↑	AUROC ↑	FPR ↓	Dist. ↑	AUROC ↑	FPR ↓
GANomaly[35]	0.036	0.06	0.53	0.93	0.07	0.53	0.93
DRAEM[36]	0.120	0.09	0.51	0.95	0.17	0.51	0.95
FastFlow[42]	0.034	0.52	0.83	0.72	0.28	0.83	0.72
PaDiM[38]	0.025	0.54	0.84	0.73	0.28	0.84	0.73
PatchCore[39]	0.937	**0.90**	0.51	0.98	**0.36**	0.51	0.98
Reverse Distillation[40]	0.070	0.45	0.77	0.72	0.17	0.76	0.72
Student-Teacher[41]	**0.014**	0.67	**0.92**	**0.39**	0.34	**0.92**	**0.39**
PaRCE (Ours)	0.296	0.45	0.74	0.94	0.16	0.74	0.94

Table 9. Comparison of measures of regional model competency for the park dataset with anomalous regions.

Results:

The results are summarized in Tabs. 7-9. We display the distribution of competency scores for pixels in ID images, familiar pixels in OOD images, and unfamiliar pixels in OOD images in Figs. 14-16. Examples of the regional competency images are shown in Figs. 17-19.

GANomaly, DRAEM, and Reverse Distillation generally perform worse than competing methods and never outperform other methods because they tend to assign similar scores to both familiar and unfamiliar pixels.

FastFlow and PaDiM tend to perform comparably to one another, generating similar score distributions and competency images. They generally perform better than the first three mentioned methods. Quantitatively, PaDiM tends to perform a bit better that FastFlow. It is one of the methods with the highest AUROC scores for the lunar dataset and achieves the lowest FPRs for the speed limit signs dataset.

PatchCore is by far the slowest method, requiring around one second to obtain an output for a single image. It tends to obtain high KS distances, achieving the highest distances for the park dataset and the highest distance between ID and unfamiliar OOD distributions for the lunar dataset. However, it also tends to obtain very high (and often the highest) FPRs due to the large range of scores for familiar pixels.

Student-Teacher Feature Pyramid Matching is consistently the fastest method, maintaining the same low average computation time across all datasets. It does not perform particularly well for the lunar or speed datasets but performs better across a number of metrics for the park dataset, achieving the highest AUROC scores and lowest FPRs.

Our method, PaRCE, scores best across the majority of metrics for the lunar and speed datasets. For the lunar dataset, it achieves the highest or second highest KS distances, highest AUROC scores, and lowest FPRs. For the speed dataset, it achieves the highest KS distances, highest AUROCs, and second lowest FPRs. Along with DRAEM, it is one of only two methods with a consistent score range. Unlike other methods, it outputs intuitive probability scores that tend to be around one for familiar pixels and below one for unfamiliar pixels. Despite its success on the lunar and speed datasets, like most other methods, it performs quite poorly for the park dataset. This reduction in performance may be due to ambiguity in manually identifying familiar and unfamiliar regions in OOD images for this dataset or limitations of the simple Felzenszwalb algorithm[44] in segmenting these more complex and detailed images.

5. Conclusions

In this work, we present a probabilistic and reconstruction-based competency estimation (PaRCE) method that captures predictive uncertainty in CNN-based perception models by estimating prediction accuracy. Through comparisons with various UQ and OOD detection techniques, we demonstrate that PaRCE effectively distinguishes between correctly classified, misclassified, and OOD samples. It also shows robust performance in identifying visual image modifications that correspond to high, medium, and low prediction accuracy. Additionally, we extend PaRCE to generate regional competency maps, which reflect model confidence for specific regions within an image. Our comparisons with existing anomaly localization techniques indicate that PaRCE often outperforms other methods in distinguishing familiar from unfamiliar pixels within an image.

Despite these strengths, we identified a dataset for which PaRCE, alongside other anomaly localization methods, performs suboptimally. This limitation could stem from ambiguity in model familiarity with this dataset or challenges in segmenting images from complex, heterogeneous environments. Future work could explore domain-specific segmentation methods to enhance performance in complex scenarios. To further improve performance, one could consider the use of different reconstruction loss functions or ensembles of autoencoders. Another promising direction would involve integrating competency scores into decision-making frameworks that benefit from risk assessment, particularly for autonomous systems or those that rely on human experts sparingly. Finally, while we evaluated competency estimation methods using autonomous navigation datasets, future work should evaluate methods in other domains, such as medical imaging and industrial inspection, where the need for robust and trustworthy models is equally critical.

Appendix

A. Data & Code Sources

The lunar, speed, and pavilion datasets we use for evaluation, as well as the trained lunar, speed, and pavilion models, are provided through Zenodo. Instructions for reproducing our results are provided in the README of our code repository.

B. Datasets for Evaluation

We conduct analysis across three unique datasets that contain manually identified OOD samples. The first dataset was obtained from a lunar environment, in which the training data contains images from an uninhabited moon and the test data contains images from a habited moon. While the training images only contain the lunar surface, the sky, and shadows, the test images additionally contain astronauts and human-made structures. This dataset enables us to assess the ability of overall competency approaches to detect images with unfamiliar objects and regional competency methods to identify these objects. Column 1 of Fig. 1 displays example in-distribution images from this dataset, and column 2 shows example OOD images.

The second dataset contains speed limit signs in Germany[46]. While the training dataset is composed of common speed limit signs (30 km/hr and higher), the test dataset set also contains signs with an uncommon speed limit (20 km/hr). This dataset assesses the ability of overall competency methods to identify images associated with unfamiliar class labels and the ability of regional competency estimators to detect regions associated with previously unseen classes. Column 1 of Fig. 2 displays example in-distribution images from this dataset, and column 2 shows example OOD images.

The third dataset contains images from regions in a park. While the training dataset only contains images from forested and grassy regions of the park, the test dataset additionally includes images from around the park's pavilion. This dataset allows for the assessment of the ability of overall competency methods to identify images from unexplored regions in an environment and for regional approaches to further identify these regions within a given image. Column 1 of Fig. 3 displays example in-distribution images from this dataset, and column 2 shows example OOD images.

While many existing works in OOD detection and anomaly localization focus on spatial anomalies–where specific regions in the image are anomalous–we are also interested in the ability of existing methods to identify non-regional anomalies that reduce model confidence. For each of the three mentioned datasets, we generate new data from the in-distribution samples. Column 1 of Figs. 1-3 displays examples of unmodified images from each of the original datasets, and columns 3-7 of these figures show examples of generated images. These images have been modified by adjusting saturation (column 3), contrast (column 4), brightness (column 5), pixelation (column 6), or noise levels (column 7). These particular image modifications were chosen because they have a significant impact on classification accuracy. We consider 21 different values for each of these five image properties. Each image modification results in either high (0.9 to 1.0), medium (0.5 to 0.9), or low (0.0 to 0.5) prediction accuracy. We are interested in whether overall competency estimation approaches can distinguish between visual image modifications that result in high, medium, and low prediction accuracy.

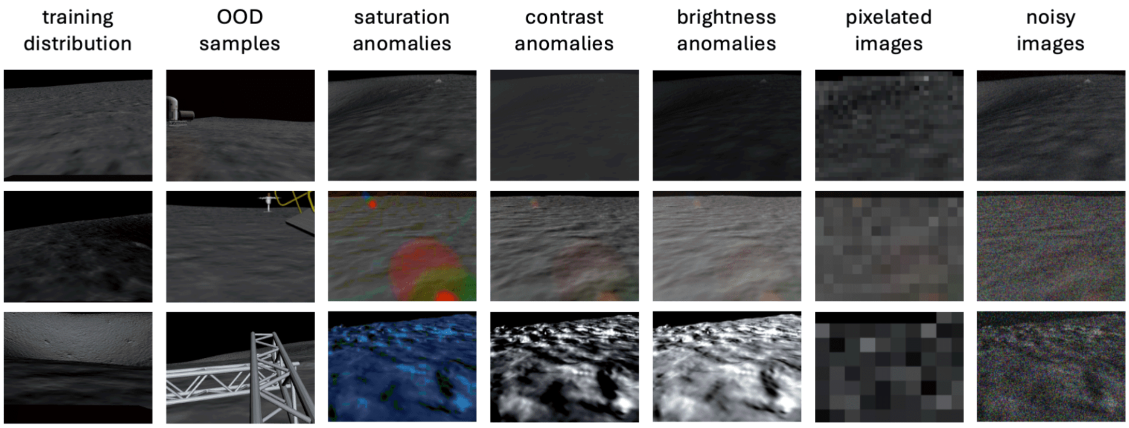


Figure 1. Example images from the *lunar* dataset. Column 1: test images drawn from the same distribution as the training images (uninhabited moon). Column 2: test images with unfamiliar obstacles drawn from a new distribution (habited moon). Column 3: test images with varied saturation levels generated from in-distribution images. Column 4: test images with varied contrast levels generated from in-distribution images. Column 5: test images with varied brightness levels generated from in-distribution images. Column 6: in-distribution images that have been pixelated. Column 7: in-distribution images with additive noise.

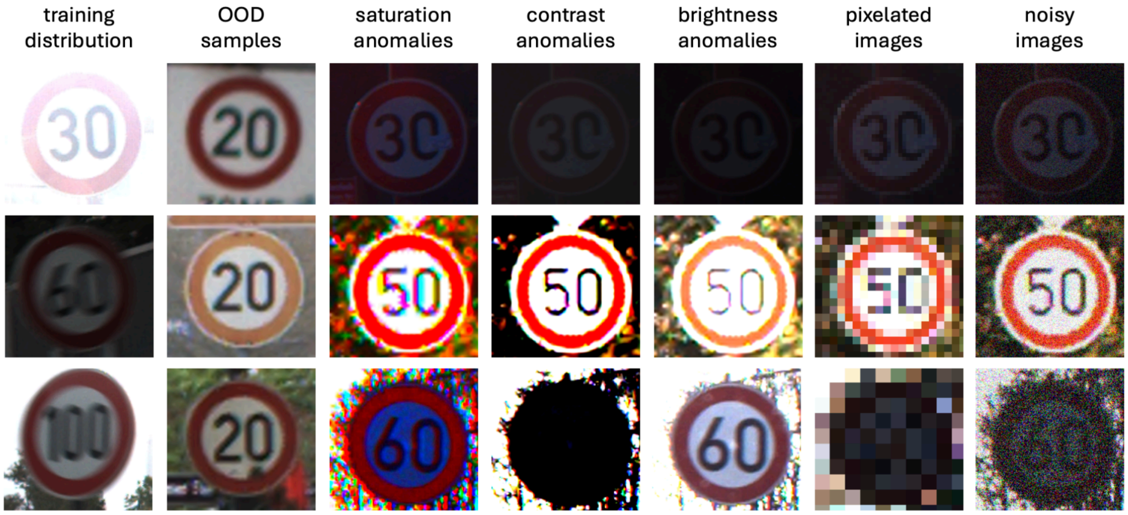


Figure 2. Example images from the *speed limit signs* dataset. Column 1: test images drawn from the same distribution as the training images (common speed limit signs). Column 2: test images with unfamiliar class labels drawn from a new distribution (uncommon speed limit signs). Column 3: test images with varied saturation levels generated from in-distribution images. Column 4: test images with varied contrast levels generated from in-distribution images. Column 5: test images with varied brightness levels generated from in-distribution images. Column 6: in-distribution images that have been pixelated. Column 7: in-distribution images with additive noise.

doi.org/10.32388/8SPGR0

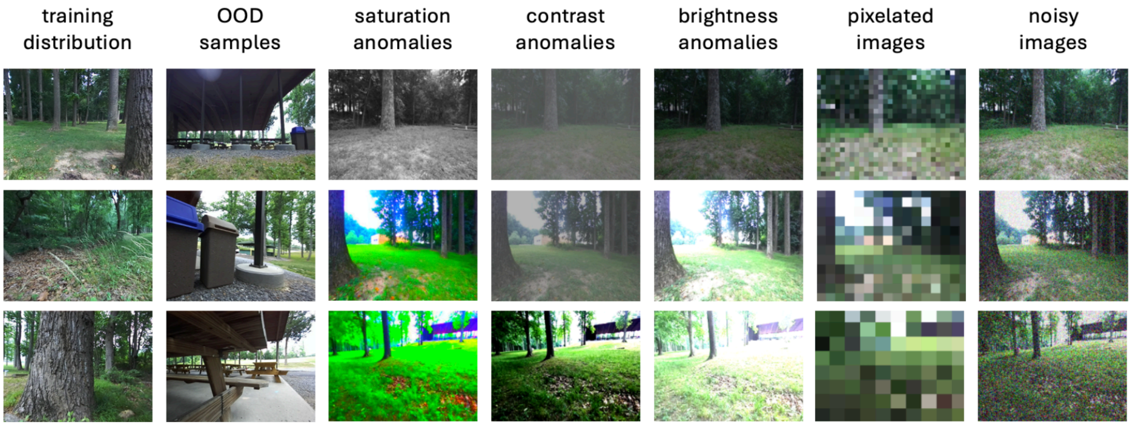


Figure 3. Example images from the *outdoor park* dataset. Column 1: test images drawn from the same distribution as the training images (forested/grassy regions). Column 2: test images associated with unexplored regions drawn from a new distribution (regions around pavilion). Column 3: test images with varied saturation levels generated from in-distribution images. Column 4: test images with varied contrast levels generated from in-distribution images. Column 5: test images with varied brightness levels generated from in-distribution images. Column 6: in-distribution images that have been pixelated. Column 7: in-distribution images with additive noise.

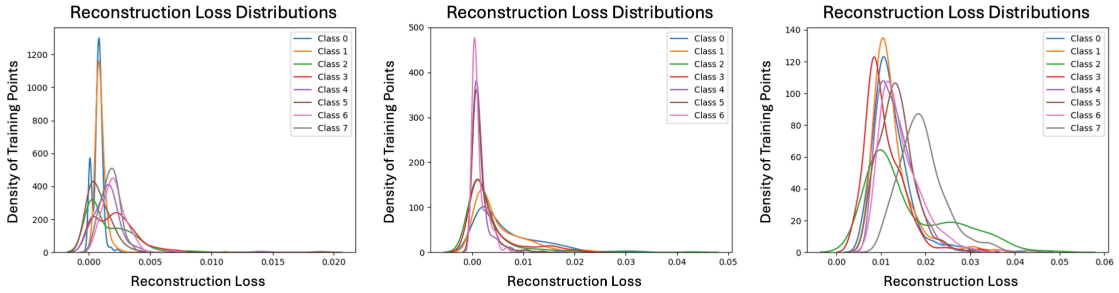
C. Additional Details on PaRCE Score

To estimate the in-distribution probability of our overall PaRCE score, we design an autoencoder to reconstruct the input image. A holdout set of images, drawn from the same distribution as the training samples but not seen by the perception model, is used to estimate the distribution of the reconstruction loss for each class. The distributions for this set across each of the datasets described in Appendix B are displayed in Fig. 4. Similarly, to estimate the in-distribution probability of our regional PaRCE score, we design an autoencoder to inpaint missing segments of an image. A holdout set of segments is used to estimate the distributions of reconstruction loss, which are displayed in Fig. 6 for each of the datasets.

The probability that an image, $\mathbf{X}$, is drawn from the same distribution as those in the training sample is given by Eq. (5). We estimate $P(D|\{f(\mathbf{X})=c\})$ as the probability that the reconstruction loss, $\ell(\mathbf{X})$, corresponding to image $\mathbf{X}$ aligns with the bottom $N\%$ of losses for the training distribution. For each class, we assume the reconstruction loss, $\mathcal{L}_c$, follows a Gaussian distribution with mean μ_c and standard deviation σ_c. Because $\mathcal{L}_c$ is a Gaussian random variable, N corresponds to a z-score, z. We then estimate the class in-distribution probability, $P(D|\{f(\mathbf{X})=c\})$, in the following way:

$$\begin{aligned}\hat{P}(D|\{f(\mathbf{X})=c\}) &= P(\{\mathcal{L}_c > \ell(\mathbf{X}) - (\mu_c + z\sigma_c)\}) \qquad (9)\\ &= 1 - P(\{\mathcal{L}_c \leq \ell(\mathbf{X}) - (\mu_c + z\sigma_c)\}) \qquad (10)\\ &= 1 - F_{\mathcal{L}_c \sim \mathcal{N}(\mu_c,\sigma_c)}(\ell(\mathbf{X}) - \mu_c - z\sigma_c) \qquad (11)\\ &= 1 - F_{Z\sim\mathcal{N}(0,1)}\left(\frac{\ell(\mathbf{X}) - 2\mu_c - z\sigma_c}{\sigma_c}\right) \qquad (12)\\ &=: 1 - \phi\left(\frac{\ell(\mathbf{X}) - 2\mu_c}{\sigma_c} - z\right). \qquad (13)\end{aligned}$$

To calibrate our competency estimator, we select the z-score parameter such that the average competency score (left side of Eq. (8)) reflects the prediction accuracy (right side of Eq. (8)) for the in-distribution holdout set. Fig. 5 displays a plot of the prediction accuracy and average overall competency score for the holdout set across various z-score values for each of the three datasets. Fig. 7 displays the prediction accuracy and average regional competency score. The z-score is chosen such that the absolute difference between the prediction accuracy and average competency score is minimized.

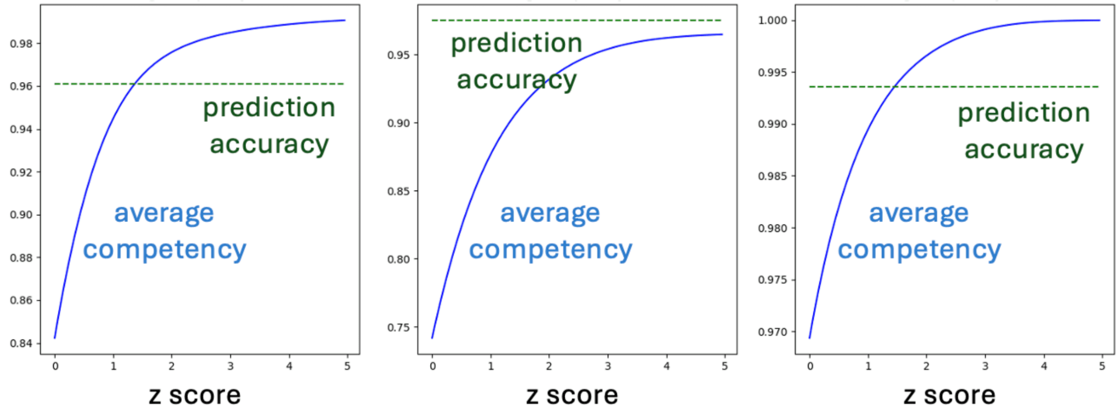


(a) Lunar dataset

(b) Speed dataset

(c) Park dataset

Figure 4. True distribution of reconstruction losses obtained by the trained image reconstruction model for images in the in-distribution holdout set of the (a) lunar, (b) speed limit signs, and (c) outdoor park datasets across class labels.



(a) Lunar dataset

(b) Speed dataset

(c) Park dataset

Figure 5. Plot of image prediction accuracy and average overall competency score for the in-distribution holdout set (defined in Eq. (8))versus the z-score parameter (from Eq. (7)) for the (a) lunar, (b) speed limit signs, and (c) outdoor park datasets.

D. Additional Analysis of Competency Scores

D.1. Evaluation Metrics

We evaluate both overall and regional competency estimation methods based on their computation time and their ability to distinguish between various types of data. For the overall competency estimation approaches, we assess their ability to distinguish between correctly classified, misclassified, and OOD samples. In addition, we evaluate their ability to distinguish between visual image modifications resulting in high, medium, and low prediction accuracy. (Refer to

Appendix B for additional details on these data types.) Comparing regional competency estimation approaches, we consider the ability to distinguish between familiar pixels (from both in-distribution and OOD images) and unfamiliar pixels (from OOD samples). Familiar regions are all of the pixels that occupy image structures that exist in the training set, and unfamiliar pixels are those that occupy structures that were not present during training. To quantify the ability of each scoring method to distinguish between sets of samples, we utilize three metrics: the Kolmogorov–Smirnov (KS) test, the area under the receiver operating characteristic curve (AUROC), and the false positive rate (FPR) at a 95% true positive rate (TPR). Each metric provides unique insights into method performance in distinguishing between sample types.

The two-sample KS test is traditionally used to assess whether two data samples come from the same distribution by posing the null hypothesis that both samples come from populations with the same distribution. It may also be used to test the extent to which two underlying probability distributions differ by measuring the maximum distance between the cumulative distribution functions (CDFs) of two score distributions. This distance is one way to quantify the separability of score distributions associated with distinct sample types, where higher KS values indicate greater dissimilarity between distributions, thus signifying better performance in distinguishing these sample types.

We also use the AUROC to quantify the separability of score distributions. A receiver operating characteristic (ROC) curve is a graph that shows the performance of a model by plotting the TPR against the FPR at different threshold values. The AUROC is a widely used metric that evaluates the performance of a binary classifier across different decision thresholds. Here, we use the AUROC to measure the degree of overlap between score distributions for different sample types. An AUROC of 1.0 signifies perfect separability, while an AUROC of 0.5 suggests no discriminative power between the classes.

Finally, the FPR at a fixed TPR of 95% assesses the rate at which a model incorrectly labels a sample as a positive point (i.e., OOD, low-accuracy, or unfamiliar) when it is, in fact, a negative point (i.e., in-distribution, high-accuracy, or familiar). By fixing the TPR at 95%, we ensure the competency estimator maintains high sensitivity to anomalies or misclassifications. A lower FPR at this threshold implies that the model is better at avoiding false alarms while still accurately identifying true OOD, low-accuracy, and unfamiliar samples. This metric is valuable because false positives can reduce trustworthiness.

D.2. Overall Competency Scores

In this section, we provide the score distributions for correctly classified, misclassified, and OOD samples across various methods for assessing overall model competency. Fig. 8 displays box plots of the distributions for the lunar dataset, Fig. 9 displays plots for the speed limit signs dataset, and Fig. 10 displays plots for the outdoor park dataset. We also provide the score distributions for samples with high, medium, and low accuracy modifications across the same methods for assessing overall model competency. Fig. 11 displays box plots of the distributions for the lunar dataset, Fig. 12 displays plots for the speed limit signs dataset, and Fig. 13 displays plots for the outdoor park dataset.

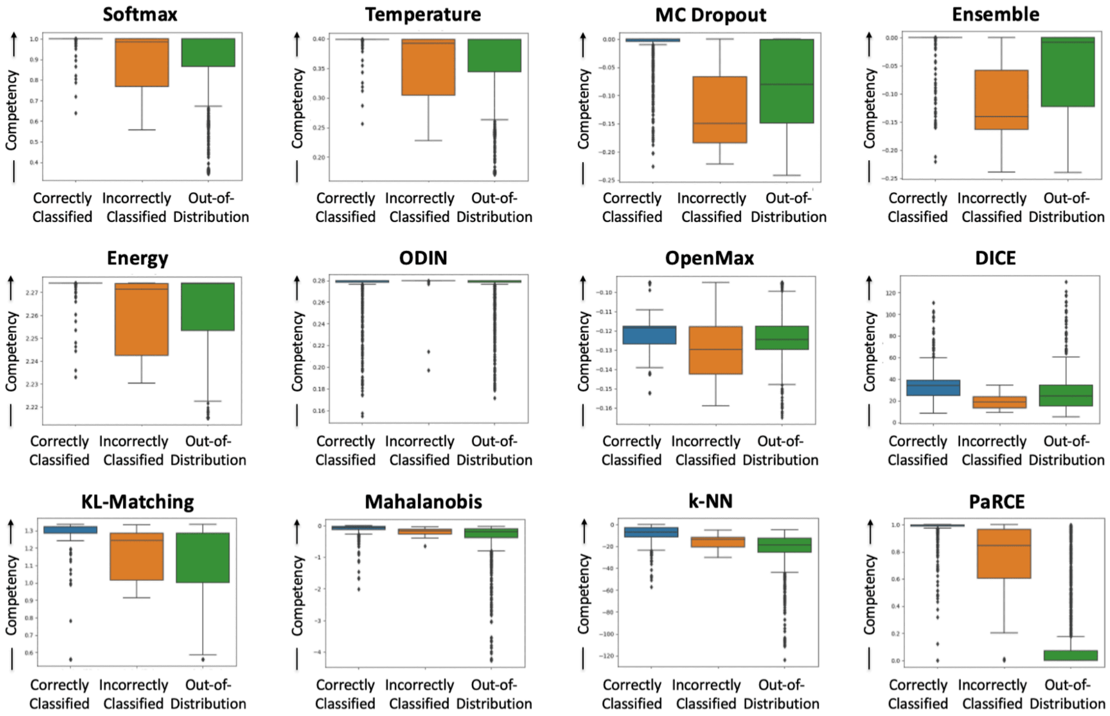


Figure 8. Box plots capturing the score distributions for correctly classified, incorrectly classified, and OOD samples in the lunar dataset, where scores are generated by the Maximum Softmax Probability (MSP) baseline, the calibrated MSP with Temperature Scaling[2], Monte Carlo (MC) Dropout[15], Ensembling[16], the Energy Score[17], ODIN[18], OpenMax[19], DICE[20], KL-Matching[25], the Mahalanobis Distance[26], k-Nearest Neighbors (k-NN)[27], and our PaRCE method. We expect correctly classified images to be assigned high competency scores, while misclassified and OOD samples are expected to have lower associated scores.

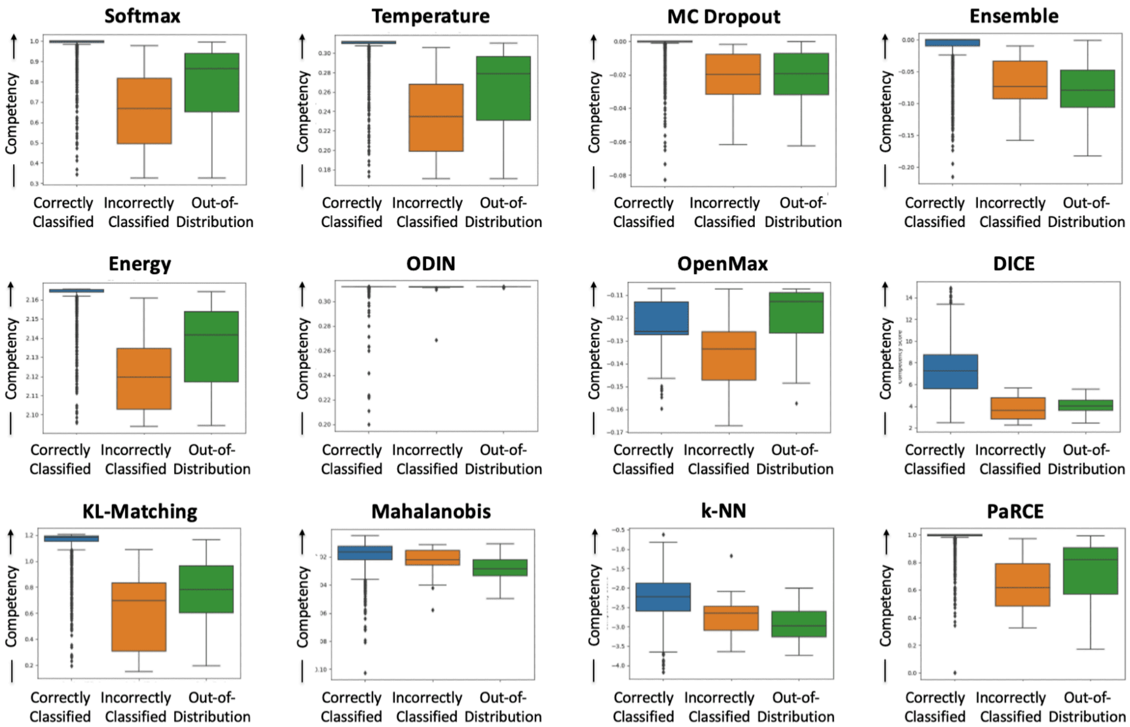


Figure 9. Box plots capturing the score distributions for correctly classified, incorrectly classified, and OOD samples in the speed limit signs dataset, where scores are generated by the Maximum Softmax Probability (MSP) baseline, the calibrated MSP with Temperature Scaling[2], Monte Carlo (MC) Dropout[15], Ensembling[16], the Energy Score[17], ODIN[18], OpenMax[19], DICE[20], KL-Matching[25], the Mahalanobis Distance[26], k-Nearest Neighbors (k-NN)[27], and our PaRCE method. We expect correctly classified images to be assigned high competency scores, while misclassified and OOD samples are expected to have lower associated scores.

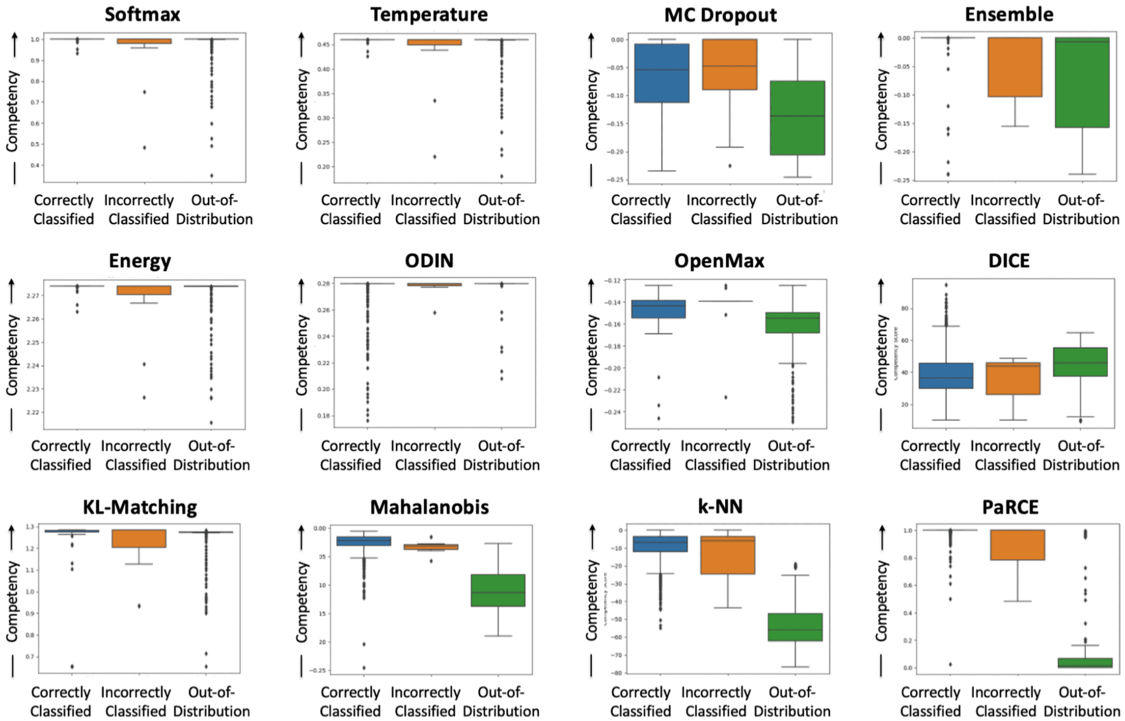


Figure 10. Box plots capturing the score distributions for correctly classified, incorrectly classified, and OOD samples in the outdoor park dataset, where scores are generated by the Maximum Softmax Probability (MSP) baseline, the calibrated MSP with Temperature Scaling[2], Monte Carlo (MC) Dropout[15], Ensembling[16], the Energy Score[17], ODIN[18], OpenMax[19], DICE[20], KL-Matching[25], the Mahalanobis Distance[26], k-Nearest Neighbors (k-NN)[27], and our PaRCE method. We expect correctly classified images to be assigned high competency scores, while misclassified and OOD samples are expected to have lower associated scores.

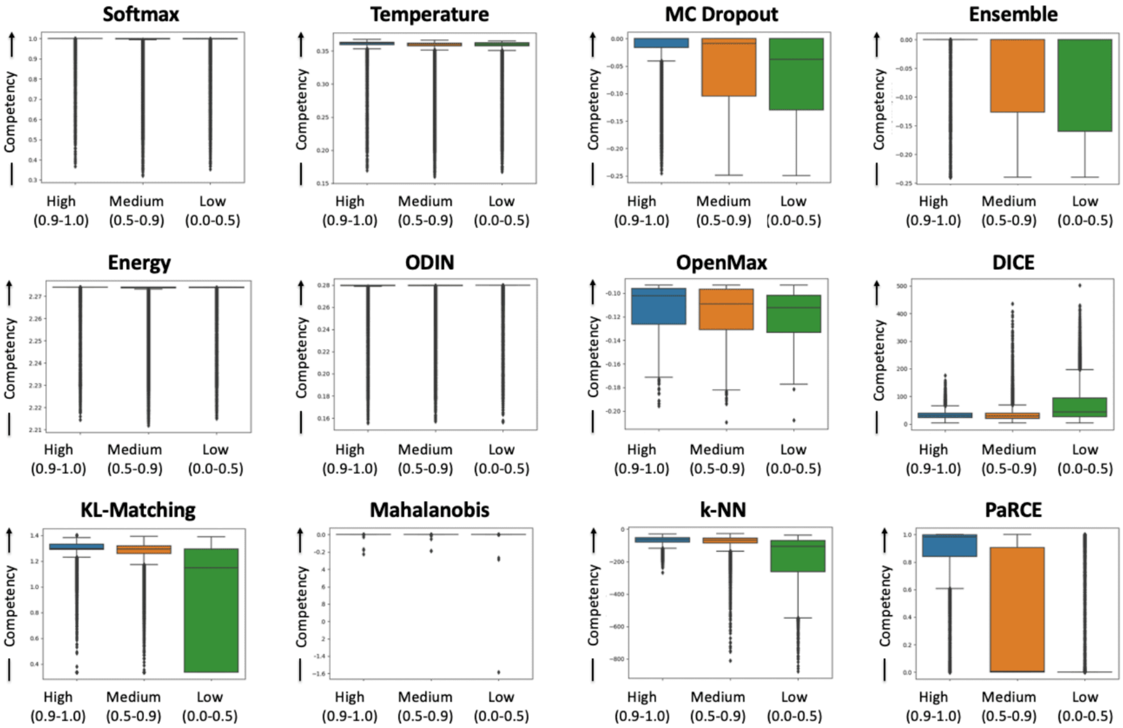


Figure 11. Box plots capturing the score distributions for images with high, medium, and low accuracy visual modifications in the lunar dataset, where scores are generated by the Maximum Softmax Probability (MSP) baseline, the calibrated MSP with Temperature Scaling[2], Monte Carlo (MC) Dropout[15], Ensembling[16], the Energy Score[17], ODIN[18], OpenMax[19], DICE[20], KL-Matching[25], the Mahalanobis Distance[26], k-Nearest Neighbors (k-NN)[27], and our PaRCE method. We expect high accuracy images to be assigned high competency scores, while medium and low accuracy samples are expected to have lower associated scores.

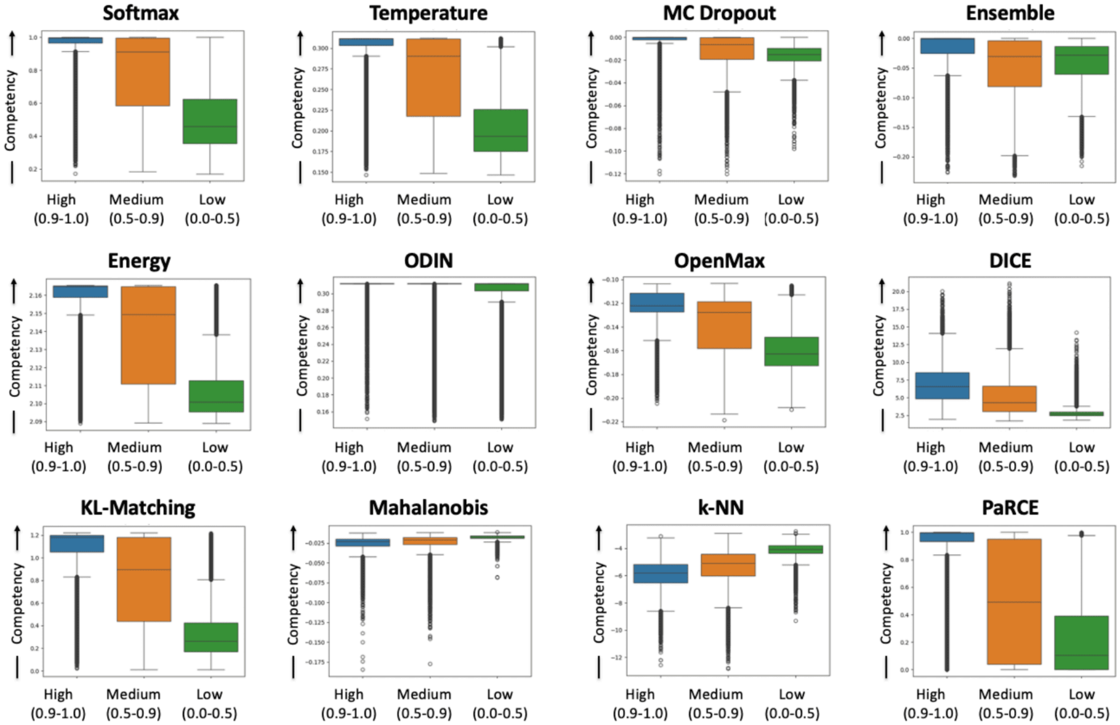


Figure 12. Box plots capturing the score distributions for images with high, medium, and low accuracy modifications in the speed limit signs dataset, where scores are generated by the Maximum Softmax Probability (MSP) baseline, the calibrated MSP with Temperature Scaling[2], Monte Carlo (MC) Dropout[15], Ensembling[16], the Energy Score[17], ODIN[18], OpenMax[19], DICE[20], KL-Matching[25], the Mahalanobis Distance[26], k-Nearest Neighbors (k-NN)[27], and our PaRCE method. We expect high accuracy images to be assigned high competency scores, while medium and low accuracy samples are expected to have lower associated scores.

doi.org/10.32388/8SPGR0

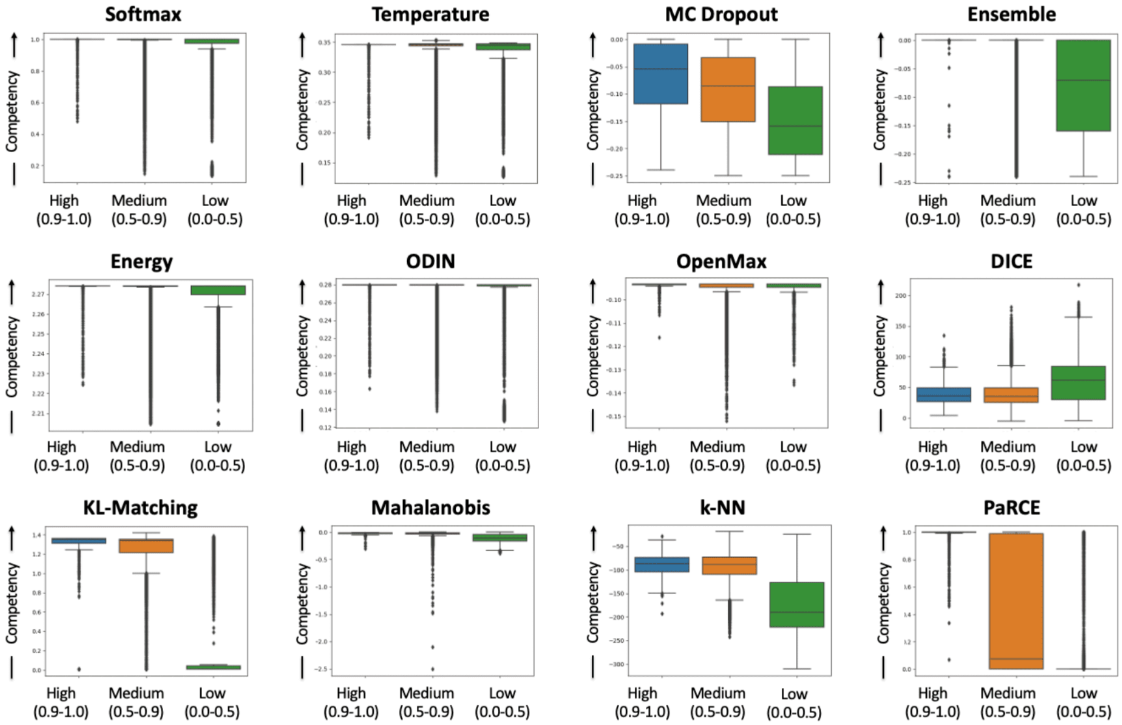


Figure 13. Box plots capturing the score distributions for images with high, medium, and low accuracy visual modifications in the *outdoor park* dataset, where scores are generated by the Maximum Softmax Probability (MSP) baseline, the calibrated MSP with Temperature Scaling[2], Monte Carlo (MC) Dropout[15], Ensembling[16], the Energy Score[17], ODIN[18], OpenMax[19], DICE[20], KL-Matching[25], the Mahalanobis Distance[26], k-Nearest Neighbors (k-NN)[27], and our PaRCE method. We expect high accuracy images to be assigned high competency scores, while medium and low accuracy samples are expected to have lower associated scores.

D.3. Regional Competency Images

In this section, we provide the score distributions for familiar regions in in-distribution (ID) images, familiar regions in OOD images, and unfamiliar regions in OOD images across various methods for assessing regional model competency. Figures 14 displays box plots of the distributions for the *lunar* dataset, 15 displays plots for the *speed limit signs* dataset, and 16displays plots for the *outdoor park* dataset. We also provide examples of the competency images generated by each of the same methods for assessing regional model competency. Figures 17 displays example images for the *lunar* dataset, 18 displays examples for the *speed limit signs* dataset, and 19 displays examples for the *outdoor park* dataset.

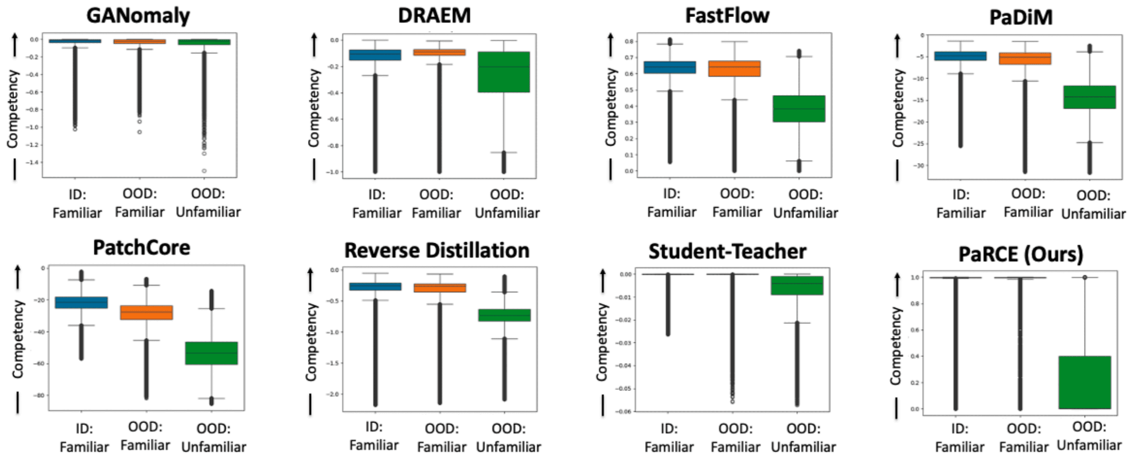


Figure 14. Box plots capturing the score distributions for all pixels of in-distribution (ID) images, familiar pixels of OOD images, and unfamiliar pixels of OOD images in the *lunar* dataset, where scores are generated by GANomaly[35], DRAEM[36], FastFlow[42], PaDiM[38], PatchCore[39], Reverse Distillation[40], Student-Teacher Feature Pyramid Matching[41], and our PaRCE method. We expect high competency scores for familiar pixels (from both ID and OOD images) and lower scores for unfamiliar pixels.

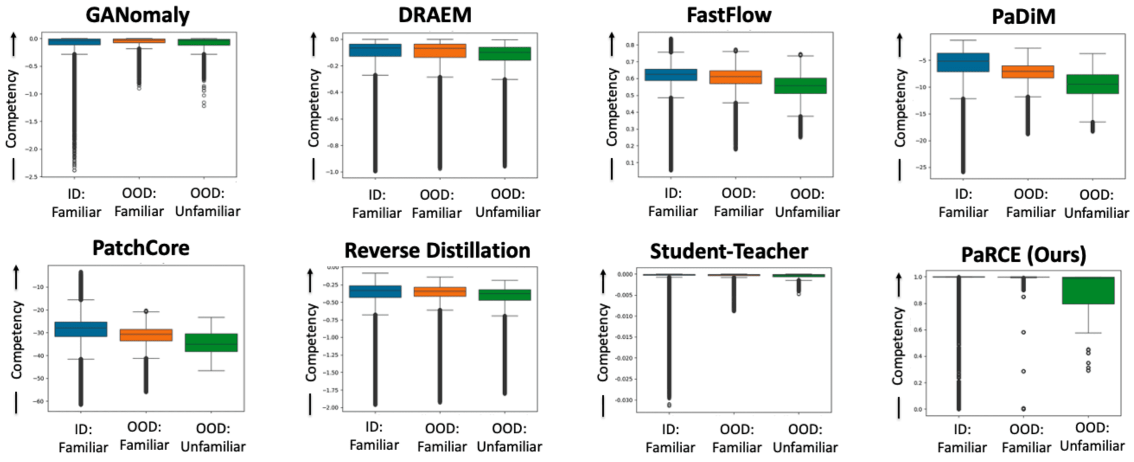


Figure 15. Box plots capturing the score distributions for all pixels of in-distribution (ID) images, familiar pixels of OOD images, and unfamiliar pixels of OOD images in the *speed limit signs* dataset, where scores are generated by GANomaly[35], DRAEM[36], FastFlow[42], PaDiM[38], PatchCore[39], Reverse Distillation[40], Student-Teacher Feature Pyramid Matching[41], and our PaRCE method. We expect high competency scores for familiar pixels (from both ID and OOD images) and lower scores for unfamiliar pixels.

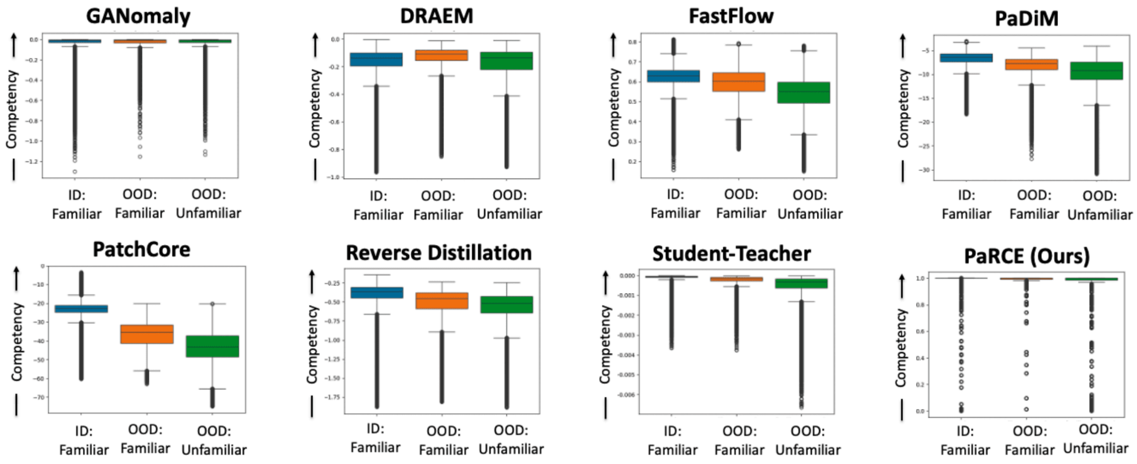


Figure 16. Box plots capturing the score distributions for all pixels of in-distribution (ID) images, familiar pixels of OOD images, and unfamiliar pixels of OOD images in the *outdoor park* dataset, where scores are generated by GANomaly[35], DRAEM[36], FastFlow[42], PaDiM[38], PatchCore[39], Reverse Distillation[40], Student-Teacher Feature Pyramid Matching[41], and our PaRCE method. We expect high competency scores for familiar pixels (from both ID and OOD images) and lower scores for unfamiliar pixels.

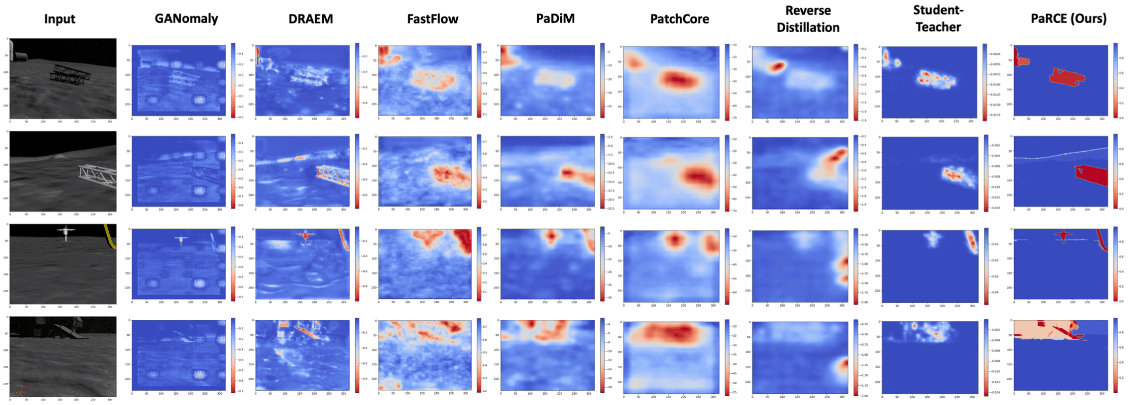


Figure 17. A comparison of the regional competency images obtained for example OOD images in the *lunar* dataset, for which the sky and lunar surface are familiar to the perception model while astronauts and human-made structures are unfamiliar. A pixel assigned a high competency score appears more blue, while a pixel assigned a low score appears more red. We expect more red regions to correspond to unfamiliar objects (i.e., an astronaut or human-made structure) while more blue regions should correspond to familiar structures.

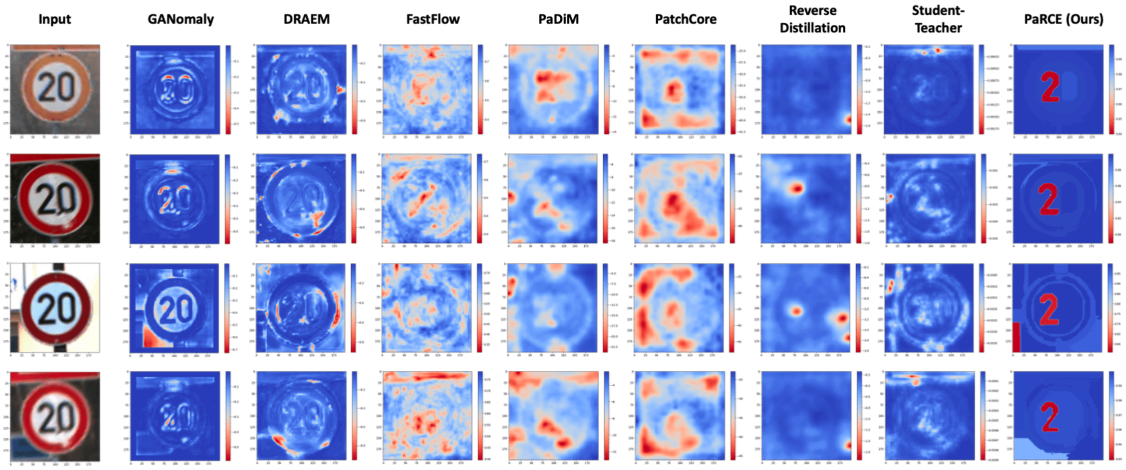


Figure 18. A comparison of the regional competency images obtained for example OOD images in the *speed limit signs* dataset, for which speeds 30-120 km/hr are familiar to the perception model while the speed 20 km/hr is unfamiliar. A pixel assigned a high competency score appears more blue, while a pixel assigned a low score appears more red. We expect more red regions to correspond to regions associated with the unseen class (i.e., the number 2) while more blue regions should correspond to other parts of the traffic sign.

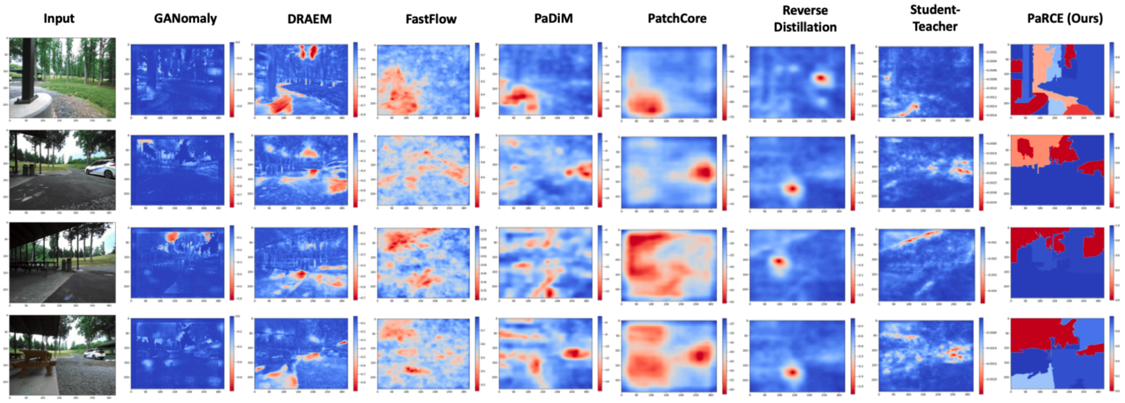


Figure 19. A comparison of the regional competency images obtained for example OOD images in the *outdoor park* dataset, for which the grassy and forested regions are familiar to the perception model while regions around the pavilion are unfamiliar. A pixel assigned a high competency score appears more blue, while a pixel assigned a low score appears more red. We expect more red regions to correspond to unexplored regions in the environment (i.e., areas around the pavilion) while more blue regions should correspond to familiar regions.

References

1. ^Saleem R, Yuan B, Kurugollu F, Anjum A, Liu L (2022). *"Explaining deep neural networks: A survey on the global interpretation methods". Neurocomputing. 513: 165–180. Available from: https://linkinghub.elsevier.com/retrieve/pii/S09252312220122*

18.

2. a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p*Guo C, Pleiss G, Sun Y, Weinberger KQ. "On calibration of modern neural networks." In: Proceedings of Machine Learning Research (PMLR). 2017. p. 1321–1330. Available from: https://dl.acm.org/doi/10.5555/3305381.3305518.*
3. a, b, c, d*Ovadia Y, Fertig E, Ren J, Nado Z, Sculley D, Nowozin S, Dillon JV, Lakshminarayanan B, Snoek J. "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift." In: Advances in Neural Information Processing Systems (NeurIPS). 2019;* **32***: 14003–14014. Available from: https://dl.acm.org/doi/abs/10.5555/3454287.3455541.*
4. ^*Nguyen A, Yosinski J, Clune J. "Deep Neural Networks Are Easily Fooled: High Confidence Predictions for Unrecognizable Images." In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2015 Jun. p. 427-436. Available from: https://ieeexplore.ieee.org/document/7298640.*
5. a, b*Gawlikowski J, Tassi CRN, Ali M, Lee J, Humt M, Feng J, Kruspe A, Triebel R, Jung P, Roscher R, Shahzad M, Yang W, Bamler R, Zhu XX (2023). "A survey of uncertainty in deep neural networks". Artificial Intelligence Review.* **56***: 1513–1589. doi:10.1007/s10462-023-10562-9.*
6. a, b*Schwaiger A, Sinhamahapatra P, Gansloser J, Roscher K. Is uncertainty quantification in deep learning sufficient for out-of-distribution detection? In: Proceedings of the Workshop on Artificial Intelligence Safety (AISafety); 2020. Available from: https://publica.fraunhofer.de/handle/publica/408442.*
7. a, b*Yang J, Zhou K, Li Y, Liu Z (2021). "Generalized Out-of-Distribution Detection: A Survey". http://arxiv.org/abs/2110.11334.*
8. a, b*Tao X, Gong X, Zhang X, Yan S, Adak C (2022). "Deep Learning for Unsupervised Anomaly Localization in Industrial Images: A Survey". IEEE Transactions on Instrumentation and Measurement.* **71***: 1–21. Available from: https://ieeexplore.ieee.org/document/9849507.*
9. a, b*Tschuchnig ME, Gadermayr M. Anomaly Detection in Medical Imaging - A Mini Review. In: Data Science – Analytics and Applications. 2022. p. 33–38. Available from: https://link.springer.com/chapter/10.1007/978-3-658-36295-9_5.*
10. a, b*Anoopa S, Salim A. Survey on anomaly detection in surveillance videos. Materials Today: Proceedings. 2022;* **58(1)***: 162–167. Available from: https://linkinghub.elsevier.com/retrieve/pii/S2214785322001997.*
11. ^*Gal Y. Uncertainty in Deep Learning [Doctor of Philosophy]. Cambridge, England: University of Cambridge; 2016. Available from: https://www.cs.ox.ac.uk/people/yarin.gal/website/thesis/thesis.pdf.*
12. ^*Quiñonero-Candela J, Sugiyama M, Schwaighofer A, Lawrence N, editors. Dataset shift in machine learning. Cambridge, MA: The MIT Press; 2008. Available from: http://www.acad.bg/ebook/ml/The.MIT.Press.Dataset.Shift.in.Machine.Learning.Feb.2009.eBook-DDU.pdf.*
13. ^*Neal RM. "Bayesian learning via stochastic dynamics." In: Advances in Neural Information Processing Systems (NIPS). 1992; 5:475–482. Available from: https://papers.nips.cc/paper_files/paper/1992/hash/f29c21d4897f78948b91f03172341b7b-Abstract.html.*
14. ^*Neal RM. Bayesian Learning for Neural Networks. New York, NY: Springer; 1996. (Lecture Notes in Statistics; vol. 118). Available from: http://link.springer.com/10.1007/978-1-4612-0745-0.*
15. a, b, c, d, e, f, g, h, i, j, k, l, m, n*Gal Y, Ghahramani Z (2016). "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: Proceedings of Machine Learning Research (PMLR).* **48***: 1050-1059. Available from: https://proceedings.mlr.press/v48/gal16.html.*

16. a, b, c, d, e, f, g, h, i, j, k, l, m, nLakshminarayanan B, Pritzel A, Blundell C (2017). "Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles". In: Advances in Neural Information Processing Systems (NIPS). **30**: 6405–6416. Available from: https://papers.nips.cc/paper_files/paper/2017/hash/9ef2ed4b7fd2c810847ffa5fa85bce38-Abstract.html.

17. a, b, c, d, e, f, g, h, i, j, k, l, m, nLiu W, Wang X, Owens JD, Li Y (2020). "Energy-based out-of-distribution detection." In: Advances in Neural Information Processing Systems (NeurIPS). **33**: 21464–21475. Available from: https://papers.nips.cc/paper_files/paper/2020/hash/f5496252609c43eb8a3d147ab9b9c006-Abstract.html.

18. a, b, c, d, e, f, g, h, i, j, k, l, m, nLiang S, Li Y, Srikant R. Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks. In: The Sixth International Conference on Learning Representations (ICLR); 2018. Available from: https://openreview.net/forum?id=H1VGkIxRZ.

19. a, b, c, d, e, f, g, h, i, j, k, l, m, nBendale A, Boult TE. Towards open set deep networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016. p. 1563-1572. Available from: https://ieeexplore.ieee.org/document/7780542.

20. a, b, c, d, e, f, g, h, i, j, k, l, m, nSun Y, Li Y. "DICE: Leveraging Sparsification for Out-of-Distribution Detection." In: Computer Vision -- ECCV 2022, 2022:691-708. Available from: https://link.springer.com/chapter/10.1007/978-3-031-20053-3_40#citeas.

21. ^Zong B, Song Q, Min MR, Cheng W, Lumezanu C, Cho D, Chen H. "Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection." In: The Sixth International Conference on Learning Representations (ICLR); 2018 Feb. Available from: https://openreview.net/forum?id=BJJLHbb0-.

22. ^Rezende DJ, Mohamed S. Variational Inference with Normalizing Flows. In: Proceedings of Machine Learning Research (PMLR). 2015. p. 1530-1538. Available from: https://proceedings.mlr.press/v37/rezende15.html.

23. ^Kingma DP, Dhariwal P (2018). "Glow: Generative Flow with Invertible 1x1 Convolutions". In: Advances in Neural Information Processing Systems (NeurIPS). **31**. Available from: https://papers.nips.cc/paper_files/paper/2018/hash/d139db6a236200b21cc7f752979132d0-Abstract.html.

24. ^Ren J, Liu PJ, Fertig E, Snoek J, Poplin R, DePristo MA, Dillon JV, Lakshminarayanan B. Likelihood ratios for out-of-distribution detection. In: Advances in Neural Information Processing Systems (NeurIPS). 2019; **32**. Available from: https://papers.nips.cc/paper_files/paper/2019/hash/1e79596878b2320cac26dd792a6c51c9-Abstract.html.

25. a, b, c, d, e, f, g, h, i, j, k, l, m, nHendrycks D, Basart S, Mazeika M, Zou A, Kwon J, Mostajabi M, Steinhardt J, Song D. "Scaling Out-of-Distribution Detection for Real-World Settings." Proceedings of Machine Learning Research (PMLR). 2022; **162**: 8759-8773. Available from: https://proceedings.mlr.press/v162/hendrycks22a.html.

26. a, b, c, d, e, f, g, h, i, j, k, l, m, nLee K, Lee K, Lee H, Shin J (2018). "A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks". In: Advances in Neural Information Processing Systems (NeurIPS). **31**: 7167–7177. Available from: https://papers.nips.cc/paper_files/paper/2018/hash/abdeb6f575ac5c6676b747bca8d09cc2-Abstract.html.

27. a, b, c, d, e, f, g, h, i, j, k, l, m, nSun Y, Ming Y, Zhu X, Li Y (2022). "Out-of-Distribution Detection with Deep Nearest Neighbors". In: Proceedings of Machine Learning Research (PMLR). **162**: 20827-20840. Available from: https://proceedings.mlr.press/v162/sun22d.html.

28. ^Xia Y, Cao X, Wen F, Hua G, Sun J. "Learning Discriminative Reconstructions for Unsupervised Outlier Removal." In: 2015 IEEE International Conference on Computer Vision (ICCV); 2015. p. 1511-1519. Available from: https://ieeexplore.ieee.org/document/7410534.

29. ^Gong D, Liu L, Le V, Saha B, Mansour MR, Venkatesh S, van den Hengel A. "Memorizing Normality to Detect Anomaly: Memory-augmented Deep Autoencoder for Unsupervised Anomaly Detection". In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). 2019. p. 1705-1714. Available from: https://ieeexplore.ieee.org/document/9010977.
30. ^Sabokrou M, Khalooei M, Fathy M, Adeli E. "Adversarially Learned One-Class Classifier for Novelty Detection." In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2018. p. 3379-3388. Available from: https://ieeexplore.ieee.org/document/8578454.
31. ^Narayanan K, Latifi S (2023). "A Survey of Techniques for Brain Anomaly Detection and Segmentation Using Machine Learning". International Journal of Communications, Network and System Sciences (IJCNS). **16** (7): 151–167. doi:10.4236/ijcns.2023.167011.
32. ^Liu Z, Zhou Y, Xu Y, Wang Z. "SimpleNet: A Simple Network for Image Anomaly Detection and Localization." In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023. p. 20402-20411. Available from: https://ieeexplore.ieee.org/document/10203746/.
33. ^Bergmann P, Löwe S, Fauser M, Sattlegger D, Steger C. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. In: Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP). 2019. p. 372–380. Available from: https://www.scitepress.org/Papers/2019/73645/73645.pdf.
34. ^Zavrtanik V, Kristan M, Skočaj D (2021). "Reconstruction by inpainting for visual anomaly detection". Pattern Recognition. **112**. Available from: https://linkinghub.elsevier.com/retrieve/pii/S0031320320305094.
35. a, b, c, d, e, f, g, hAkcay S, Atapour-Abarghouei A, Breckon TP. GANomaly: Semi-supervised anomaly detection via adversarial training. In: Asian Conference on Computer Vision (ACCV). 2019. p. 622–637. Available from: https://link.springer.com/chapter/10.1007/978-3-030-20893-6_39.
36. a, b, c, d, e, f, g, hZavrtanik V, Kristan M, Skocaj D. "DRÆM – A discriminatively trained reconstruction embedding for surface anomaly detection." In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV); 2021. p. 8310-8319. Available from: https://ieeexplore.ieee.org/document/9710329/.
37. ^Li CL, Sohn K, Yoon J, Pfister T. "CutPaste: Self-supervised learning for anomaly detection and localization." In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021. p. 9659-9669. Available from: https://ieeexplore.ieee.org/document/9578875/.
38. a, b, c, d, e, f, g, hDefard T, Setkov A, Loesch A, Audigier R (2021). "PaDiM: A Patch Distribution Modeling Framework for Anomaly Detection and Localization". In: Pattern Recognition. ICPR International Workshops and Challenges, pp. 475–489. Available from: http://link.springer.com/10.1007/978-3-030-68799-1_35.
39. a, b, c, d, e, f, g, hRoth K, Pemula L, Zepeda J, Schölkopf B, Brox T, Gehler P. Towards total recall in industrial anomaly detection. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022. p. 14298-14308. Available from: https://ieeexplore.ieee.org/document/9879738.
40. a, b, c, d, e, f, g, hDeng H, Li X (2022). "Anomaly Detection via Reverse Distillation from One-Class Embedding". In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). p. 9727-9736. Available from: https://ieeexplore.ieee.org/document/9879956.

41. [a, b, c, d, e, f, g, h] *Wang G, Han S, Ding E, Huang D. "Student-Teacher Feature Pyramid Matching for Anomaly Detection". 2021 March. Available from: https://arxiv.org/abs/2103.04257.*

42. [a, b, c, d, e, f, g, h] *Yu J, Zheng Y, Wang X, Li W, Wu Y, Zhao R, Wu L. FastFlow: Unsupervised Anomaly Detection and Localization via 2D Normalizing Flows. 2021 Nov. Available from: https://arxiv.org/abs/2111.07677.*

43. [^] *Rajendran V, LeVine W. Accurate layerwise interpretable competence estimation. In: Advances in Neural Information Processing Systems (NeurIPS). 2019. p. 13981-13991. Available from: https://papers.nips.cc/paper_files/paper/2019/hash/a11da6bd58b95b334f8cd49f00918f16-Abstract.html.*

44. [a, b] *Felzenszwalb PF, Huttenlocher DP (2004). "Efficient Graph-Based Image Segmentation". International Journal of Computer Vision. 59 (2): 167--181. doi:10.1023/B:VISI.0000022288.19776.77.*

45. [^] *Kirchheim K, Filax M, Ortmeier F. "PyTorch-OOD: A Library for Out-of-Distribution Detection Based on PyTorch." In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2022. p. 4351-4360. Available from: https://ieeexplore.ieee.org/document/9857485.*

46. [a, b] *Houben S, Stallkamp J, Salmen J, Schlipsing M, Igel C. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In: The 2013 International Joint Conference on Neural Networks (IJCNN). 2013. p. 1-8. Available from: https://ieeexplore.ieee.org/document/6706807.*

47. [^] *Akcay S, Ameln D, Vaidya A, Lakshmanan B, Ahuja N, Genc U. "Anomalib: A deep learning library for anomaly detection." In: 2022 IEEE International Conference on Image Processing (ICIP); 2022 Oct; p. 1706-1710. Available from: https://ieeexplore.ieee.org/document/9897283.*

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.