Out-of-Distribution Image Detection in AI-Based Pest Management App

Aloysius Lim [*1] **Austin Nguyen** [*1] **Eric Helmold** [*1] **Erin Tomlinson** [*1] **Molly Liu** [*1] **Weiwei Pan** [1]
Jerome White [2] **Soma Dhavala** [2]

Abstract

Our goal is to identify and implement one or more effective solutions to the problem of out-of-distribution (OOD) image detection in Wadhwani AI's CottonAce pest management app, allowing it to reject errant images with minimal processing overhead. Our contributions to this work are an exploration of supervised, unsupervised, and self-supervised approaches for OOD detection. We explore convolutional autoencoders (CAEs) paired with latent dimensional analysis as our primary technique for OOD detection. We also find great potential in contrastive learning approaches to circumvent the need for costly annotation and handcrafted feature approaches which offer high interpretability and low computational costs.

1. Introduction

Wadhwani AI is an independent, nonprofit institute developing AI-based solutions for underserved communities in developing countries. One important area of focus is pest management for cotton farmers. Cotton is the most important fiber and a cash crop for India, providing about 6 million farmers with a direct livelihood and 40-50 million people work in the cotton trade. Small-holder farmers, contributing 75% of aggregate production, struggle with uncertainty in crop yields and income. Cotton is exceptionally vulnerable to pest attacks, with bollworms responsible for an estimated 70% of all pest damage (White et al., 2022b; Dalmia et al., 2020).

Wadhwani AI has developed a mobile phone application called CottonAce that helps cotton farmers manage bollworm infestations in their fields. Bollworms are a pernicious pest, requiring consistent monitoring and expert decision-making to properly address. CottonAce app uses an AI algorithm to identify and count bollworms in photos, and then makes customized treatment recommendations to the farmers based on what it has found. This process is illustrated in Figure 1.

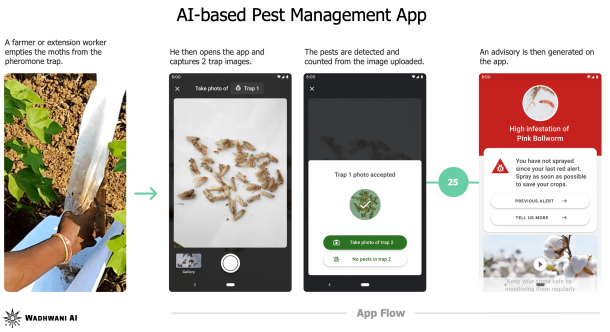

Figure 1. Example user-flow for the CottonAce pest management app. Graphic created by app development team at Wadhwani AI.

A key challenge is gracefully handling photos that are outside of the expected domain. The app instructs users to submit photos of bollworms on top of a blank, white sheet of paper that fills the entire frame (White et al., 2022b). Images that follow these guidelines are referred to as in-distribution (ID), while those that do not are considered out-of-distribution (OOD). Examples of OOD images include both photos of bollworms taken under a relaxed interpretation of the guidelines (e.g., on soiled paper that does not fill the frame), and photos that contain no bollworms at all (e.g., test images taken by new users experimenting with the app's functionality). In either case, the AI algorithm is liable to make a mistake, and the app should abstain from making a recommendation.

The app's ability to detect OOD images and abstain from making a recommendation is critical. From a user-design perspective, farmers can simply be prompted to retake and resubmit the photo. From a broader perspective, however, such actions have the potential to build farmers' trust and dramatically reduce false infestation alerts, which have been linked to several negative outcomes, such as excessive pesticide use, emotional distress, and early app abandonment (White et al., 2022b).

[*]Equal contribution [1]Harvard University, Institute for Applied Computational Science (IACS) [2]Wadhwani AI. Correspondence to: Aloysius Lim <aloysius_lim@g.harvard.edu>, Austin Nguyen <austinnguyen@g.harvard.edu>, Erin Tomlinson <erintomlinson@g.harvard.edu>, Eric Helmold <ehelmold@g.harvard.edu>, Molly Liu <chuyueliu@g.harvard.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

Our project goal is to identify effective solutions for OOD detection that could enable the CottonAce app to reject errant images with minimal processing overhead. An ideal solution to this problem would maximize OOD detection accuracy while remaining deployable in a resource constrained environment (i.e., without internet connectivity and with limited computing resources such as on a mobile device in a rural area). Solutions that violate these constraints are still considered, but will be given lower priority.

2. Related Works

OOD detection methods can be broadly organized into two main categories: supervised methods and unsupervised methods. The CottonAce team at Wadhwani AI has already explored and implemented two supervised methods, one of which is currently deployed on their mobile app (Dalmia et al., 2020). In addition to the deep neural network that it uses to count bollworms, CottonAce has a second network that it uses to classify each image as ID or OOD. This second network was trained using images that human annotators labeled as ID or OOD. This approach is not ideal for two reasons: a) image annotations are costly to obtain and b) it is challenging for humans to identify whether an image should be labeled as ID or OOD due to the significant amount of natural variety in the images submitted by users. The CottonAce team have also shown that they can use low confidence scores output by their bollworm-counting network as a basis for rejecting OOD images (White et al., 2022b). However, Wadhwani AI remains unsatisfied with these solutions and have asked our project team to explore additional techniques for this task.

Our exploration of solutions began with unsupervised learning. Our assessment of unsupervised techniques is based largely on a pair of comprehensive survey papers on the topic (Yang et al., 2021; Ruff et al., 2021). Unsupervised techniques for OOD detection fall into one of the following three categories: 1) reconstruction-based methods which assume that OOD samples are incompressible and therefore cannot be reconstructed from lower-dimensional projections, 2) dimensionality reduction methods followed by clustering analysis, and 3) one-class classification methods which learn a discriminative boundary around all ID samples (Zong et al., 2018). For high-dimensional image data, all three methods involve some form of continuous dimensionality reduction; common techniques in the literature include robust principle component analysis (PCA), kernel PCA, autoencoders, and many modern approaches attempt to learn lower-dimensional representations for the data through adversarial training with variational autoencoders and generative adversarial networks. Once a lower-dimensional representation for the data has been learned, cluster analysis can be performed in the latent space to derive additional insights about the composition of the dataset. Furthermore, if the clusters correspond to human-identifiable image prototypes, then nearest-cluster proximity can serve as a more nuanced and human-interpretable basis for OOD rejection. One-class classification methods designed to operate on the latent-space representation of images often use one-class support vector machines (SVM) or feed-forward neural networks (FFNN) to learn a discriminative boundary around the ID samples.

3. Background

Our analysis is based on images from the Wadhwani AI Pest Management Open Data repository (White et al., 2022a). This repository provides a dataset consisting of 9,727 images captured by farmers using the CottonAce app during a period of its initial deployment. The images are annotated with bounding boxes around each bollworm, which support the primary pest-counting task. To support the task of OOD detection, our team sorted the images into three categories based on how closely they adhered to the app's image quality guidelines. Example images from each category are shown in Figure 2. In addition to the aforementioned ID and OOD categories, we made the decision to introduce a third category for edge case (EC) images that fell somewhere in-between. These images lie near the difficult-to-define boundary between ID and OOD images and, as such, present a unique challenge that any OOD detection algorithm will ultimately have to resolve.

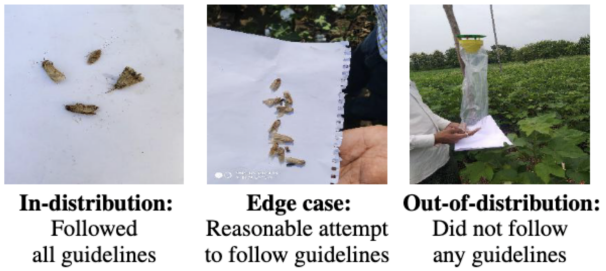

Figure 2. Our group organized the data into three categories based on how well the images aligned with the quality guidelines provided by the CottonAce mobile app (White et al., 2022b).

4. Methods

We explored seven different OOD detection methodologies, each described in separate sections below. The first three methodologies were directly supervised using the labels described in Section 3. The other four were trained primarily in an unsupervised manner (i.e., with no access to labels), although some make indirect use of the labels after training to boost performance. All methodologies share a common goal: to learn a function that assigns each image a numerical score that will serve as a basis for separating ID and

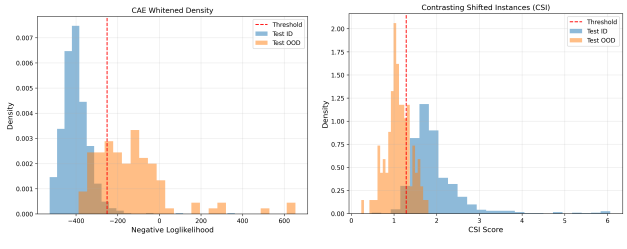

Figure 3. Example histograms of OOD detection scores using methodologies described in Sections 4.4.3 and 4.5, respectively.

OOD images. Once learned, we can use these functions to visualize the distributions of ID and OOD image scores by plotting their histograms, two examples of these plots can be seen in Figure 3. The best technique would be one that results in minimal overlap (ideally zero) and maximal separation between these two distributions, although we have found that this is very difficult to achieve in practice.

4.1. Handcrafted Features

We carried out feature engineering to produce features relevant for OOD detection. For example, ID images tend to be images taken on clean white sheets of paper in which the paper takes up the majority of the image with minimal obstructions in the periphery, whereas OOD images tend to be images taken of the ground, green plants, people, or tools used in agricultural contexts. Inspired by older computer graphic and image processing techniques, we can produce handcrafted (HC) features based on the RGB color channels, HSL (hue, saturation, light), entropy, and contrast of an image. In this model approach, we treat OOD detection as a classical supervised learning problem in which we use HC features to produce a classifier that can separate OOD images from ID images. We explored logistic regression and XGBoost as downstream classifiers on tabular data. While XGBoost is larger in size than logistic regression, the size of the model is still small enough to be loaded onto a mobile device. We also explored using synthetic minority over-sampling technique (SMOTE) to handle class imbalance (Bowyer et al., 2011). A benefit of this approach is that it can be done on images with arbitrary dimensionality, circumventing the need for pre-processing.

4.2. Structural Similarity Index Measure

Structural similarity index measure (SSIM) is a metric that quantifies the similarity between two images (Bakurov et al., 2022). SSIM contains structural information in the image that incorporates critical perceptual phenomena such as luminance masking and contrast masking elements. In this modeling approach, we use a representative subset of the ID images (in particular, 10 cluster centroid images produced by a k-means clustering algorithm run on the ID image dataset) and compare all images against this representative ID image subset. Each image is given an overall SSIM score relative to the representative subset, which is computed as the max over all pairwise SSIM scores of an input image to each of the ID images in the representative set. Similar to the HC features approach, we treat this as a classical supervised machine learning problem. Images with a higher SSIM score to the representative ID subset are considered more similar and therefore more likely to be ID than OOD. A decision boundary is created by fitting a class-balanced logistic regression model to the aggregate SSIM scores of each image in the training dataset.

4.3. Convolutional Neural Network

Convolutional neural networks (CNNs) are a powerful class of deep learning models that are particularly well suited for computer visions tasks and image processing. CNNs carry out a series of convolutions, activation, pooling layers that produce higher-order and lower-order representations of images throughout its architecture. The final output of a CNN can then be fed into a fully connected layer for classification. By framing this as a supervised learning problem, we feed images into a CNN to produce predictions as to whether they belong to the ID or OOD class.

Our CNN model was comprised of 6 encoder blocks, followed by two fully connected dense layers with a final output layer using sigmoid activation to produce a binary OOD classification. The structure of our encoder blocks (2 convolution layers each followed by a batch norm and activation layers) are structured to have the same architecture as those used in the convolutional autoencoder discussed in the next section. In training the CNN, we utilize a class-weighted binary-cross entropy loss function to account for the class imbalance of our data set in order to maintain high predictive accuracy for both ID and OOD input images.

4.4. Convolutional Autoencoder

Convolutional autoencoders (CAEs) are unsupervised neural network models that learn to extract salient features from images by repeatedly trying to reproduce them through an information bottleneck, this process is illustrated in Figure 4. These features dramatically reduce the dimensionality of the input data, making it easier for algorithms to recognize and exploit the differences between ID and OOD images. This automated feature extraction process is also a powerful form of data compression, which is critical for operating seamlessly on mobile devices.

For this project, we developed a customized CAE model and trained it to compress ID images. Deep CAE models can be difficult to train due to vanishing gradients, and are particularly susceptible to getting stuck in sub-optimal regions of the loss function (Goodfellow et al., 2016). To

mitigate these issues, we implemented a greedy, layer-wise training procedure (Bengio et al., 2006). With this procedure, sometimes referred to as "unsupervised pretraining", we add pairs of compression and decompression layers to the network one at a time, training it from the outside-in. The results are shown in Figure 5.

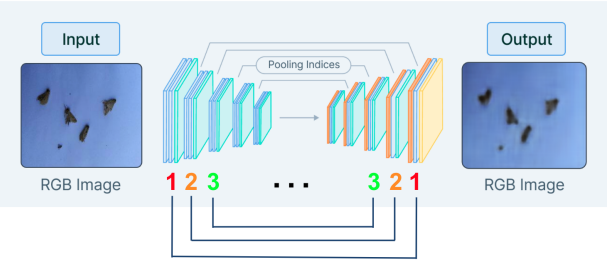

Figure 4. Our CAE model is trained to minimize reconstruction loss. Graphic adapted from (Bandyopadhyay, Hmrishav, 2022). Layer numbers illustrate greedy, layer-wise training procedure.

4.4.1. Mean-Squared Error

When we pass an image through our trained CAE model, we get back a reconstructed version that has been compressed by the first half of the network (the encoder), and then decompressed by the second half (the decoder). The CAE learns how to do this by finding a compression scheme (i.e., latent-space representation) that minimizes the mean-squared error (MSE) between input images and their reconstructions. This compression scheme will be specialized for the types of images in its training set and, in general, will perform very poorly on images that are highly dissimilar. By training our CAE on ID images alone, it will learn to reconstruct them with high fidelity, achieving low MSE scores on average. High MSE scores, on the other hand, will be much more likely to come from OOD images, and can be used as a basis for rejection. We generated MSE scores for all of the ID and OOD images in our training set and fed them into a logistic regression model with balanced class weights to find an effective threshold for OOD detection.

4.4.2. Relative Mahalanobis Distance

Continuing with our CAE approach, we explored using Relative Mahalanobis Distance (RMD) to detect OOD images (Ren et al., 2021). Using the latent representation from our CAE, we apply a Gaussian mixture model only on the ID data, deriving mean vectors and covariance matrices for k-clusters, and one baseline cluster. We then calculate the ordinary Mahalanobis Distance (MD) for both ID, and OOD images to these ID k-clusters and to the one baseline cluster. We calculate RMD by taking the difference between these two MD scores. The motivation is that OOD images should be geometrically far away from the ID k-cluster centroids. With RMD calculated for each image, we fit a linear classifier to obtain a threshold to separate OOD from ID.

4.4.3. Whitened Density Estimation

To help with our goal of OOD detection, we explored separating the features of ID and OOD images by applying statistical whitening (Su et al., 2021). Applying the whitening on the latent representation from our CAE, we then fit a Gaussian mixture model on the OOD data and derive the mean vectors and covariances for OOD. We then normalize both ID and OOD features using mean vectors and covariances for OOD features. The motivation of this technique is to normalize the OOD features into white noise, while pushing ID features away from white noise to achieve the goal of separating the two.

4.5. Contrasting Shifted Instances

The recently introduced method of Contrasting Shifted Instances (CSI) uses self-supervised contrastive learning techniques to find latent-space representations for images that are well-suited for OOD detection (Tack et al., 2020). In contrastive learning, an encoder network is tasked with finding a latent-space representation that maps similar images close to one another, and dissimilar images far away. The notion of "similarity" is task-specific and must be chosen carefully. Image augmentations are commonly used to generate "similar" pairs for this purpose, but the authors of CSI make a crucial distinction between distribution-maintaining augmentations and distribution-shifting augmentations.

Distribution-maintaining augmentations are transformations that we could apply to ID images that would leave them within the ID category (e.g., noise, blur, etc.). Distribution-shifting augmentations, on the other hand, are severe enough to flip ID images into the OOD category (e.g., color inversion, block permutation, etc.). Considered augmentations are shown in Figure 6; which of these will be most effective in each category will be dataset dependent. For example, the CSI authors use rotation as a distribution-shifting augmentation for ImageNet because nearly all of its images are right-side up. Our dataset consists of pictures of bollworms on white sheets of paper and, as such, we find that it is better to treat rotations as distribution-maintaining. Using the official PyTorch implementation, we trained a CSI model on our ID images using block permutation as our distribution-shifting augmentation, and fit a logistic regression model with balanced class weights to the resulting CSI scores to find an effective threshold for OOD detection.

5. Experiments

We implemented all seven techniques described in Section 4. Our hypothesis is that the supervised learning approaches will be fast, simple to train, and relatively light in size, mak-

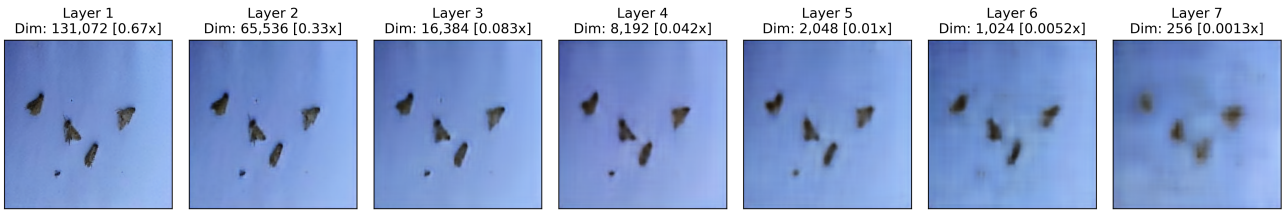

Figure 5. Results from greedy, layer-wise training of our CAE on an example ID image from the validation set. Above each image, we provide the (flattened) latent-space dimension and the corresponding image compression factor. The network has 8 layers, but layer 0 is omitted from the graphic because its latent-space representation is larger than the original image.

Figure 6. Distribution-shifting image augmentations explored by the CSI authors (Tack et al., 2020). We used block permutation (f) in all of our experiments.

ing them easy to load onto mobile devices. Our intuition is that unsupervised learning approaches such as using a CAE can produce latent representations that are at least somewhat semantically meaningful. Our expectation is that contrastive learning approaches have great potential in producing a learned representation that can optimally separate OOD from ID examples. We selected handcrafted features as a baseline against which we explored more complicated approaches such as CNN, CAE, and CSI. We hypothesize that more complex models have the potential to generalize better to unseen data than handcrafted features.

As discussed in Section 3, we first organized images into ID, EC, and OOD folders by hand using visual inspection. The EC category is somewhat artificial, however, because during deployment each image much be classified as either ID or OOD. To explore whether it is better to consider EC images as ID or OOD, we produced two different datasets, each containing all of the available images: 1) a `bollworms` dataset in which EC is classified as ID and 2) a `bollworms-clean` dataset in which EC is classified as OOD. We performed a 90%/10% train/test split on both of these datasets, and the breakdown of ID and OOD images in each can be found in Table 1.

We evaluated all our models based on four quantitative metrics that are common for evaluating classifiers for OOD detection. We included the area under the receiver operating characteristic curve (AUROC) to illustrate the trade-off between false positive rate and true positive rate, Macro-F1 to illustrate the trade-off between precision and recall, and the accuracy by class. To be cognizant of the constraint that each model needs to be loaded onto a mobile platform, we have created an extra category called "Model Size" to represent our view of the storage requirement for each model. For example, model size would be considered large in the case of the CAE (MSE) model in which both encoder and decoder are required at inference time. Model size would be considered medium when only the encoder is required, as in the case of the CAE (RMD) model. Model size is considered small for simpler models in the case of using the model with handcrafted features.

6. Results

Our primary results, derived from models that were trained and evaluated using the `bollworms` dataset where we classified EC as ID, are presented in Table 2. In the supervised setting (rows 1–3), handcrafted features can achieve comparable performance to the CNN with dramatically reduced model size, making it highly preferable for deployment to mobile platforms. That said, CNN performed the best on AUROC and Accuracy for OOD (Recall) and satisfies the criteria to be considered strong across all metrics (as we seen by the green label for the row), with the exception of model size. The second best model uses handcrafted features, which performs comparably to CNN on AUROC (0.97) and performs the best in terms of Macro-F1 (0.89) and Accuracy (0.98) for the ID class, but underperforms on Accuracy for OOD class (0.74) compared to CNN.

For our unsupervised approaches (rows 4–6), image features discovered by our CAE can be improved for anomaly detection task by applying a whitening transformation that increases the distance between ID and OOD samples. For

	`bollworms`		`bollworms-clean`	
Label	`train`	`test`	`train`	`test`
ID	7,779	865	5,800	645
OOD	975	108	2,954	328
Total	8,754	973	8,754	973

Table 1. Breakdown of ID and OOD images in each of the datasets that we used in our experiments.

Model	AUROC	Macro-F1	Acc. (ID)	Acc. (OOD)	Model Size
Handcrafted features	**0.97**	**0.89**	**0.98**	0.74	**Small**
SSIM (k-means)	0.74	0.61	0.73	0.75	**Small**
CNN	**0.97**	0.88	0.95	**0.87**	Medium
CAE (MSE)	0.83	0.69	0.94	0.44	Large
CAE (RMD)	0.86	0.70	0.91	0.54	Medium
CAE (whitened density)	**0.97**	0.86	**0.98**	0.70	Medium
Contrastive learning (CSI)	0.95	0.83	0.96	0.73	Large

Table 2. Comparison of model performance on `bollworms` dataset. Each model was trained on `bollworms-train` and evaluated on `bollworms-test` (shown above). In this data set, edge case (EC) images are considered in distribution (ID), with 865 examples as ID and 108 examples as OOD. Quantitative metrics are color-coded green if greater than or equal to 0.85, yellow if between 0.7 and 0.85, and red if less than 0.7.

Model	AUROC	Macro-F1	Acc. (ID)	Acc. (OOD)	Model Size
Handcrafted features	0.87	0.81	**0.89**	0.72	**Small**
SSIM (k-means)	0.75	0.74	0.80	0.70	**Small**
CNN	**0.92**	0.80	0.77	**0.89**	Medium
CAE (MSE)	0.80	0.76	0.87	0.64	Large
CAE (RMD)	0.84	0.77	0.87	0.67	Medium
CAE (whitened density)	0.90	**0.83**	**0.89**	0.77	Medium
Contrastive learning (CSI)	0.73	0.70	0.86	0.53	Large

Table 3. Comparison of model performance on `bollworms-clean` dataset. Each model was trained on `bollworms-clean-train` and evaluated on `bollworms-clean-test` (shown above). In this data set, edge case (EC) images are considered out-of-distribution (OOD), with 645 examples as ID and 328 examples as OOD. Quantitative metrics are color-coded green if greater than or equal to 0.85, yellow if between 0.7 and 0.85, and red if less than 0.7.

the non-whitening CAE approaches, CAE (MSE) and CAE (RMD) do not perform as well because CAE aims to minimize reconstruction error and does not explicitly learn a representation that maximizes the difference between ID and OOD. In contrast, CAE with whitened density outperforms CAE (MSE) and CAE (RMD) because the whitening transformations applied to the CAE embeddings explicitly further separates OOD from ID by normalizing OOD into white noise.

The contrastive learning (CSI) model achieves strong performance out-of-the-box (row 7), but does not outperform any of the previous methods based on quantitative or qualitative measures. Furthermore, the CSI model requires loading a ResNet-18 architecture on a mobile device, which is a significant constraint. We recommend the use of simple techniques such as a model with handcrafted features if model size is a severe constraint. Otherwise, using a CNN or CAE (whitened density) is recommended if that constraint can be relaxed. For better generalization, we should go with the unsupervised approach of CAE (whitened density) trained on `bollworms-clean`.

Our secondary results from Table 3 are from models trained using the `bollworms-clean` dataset where we classified EC as OOD. Overall, we observe a performance drop when we consider EC images as OOD. The close proximity of EC images to ID images makes the challenge of OOD detection more difficult. One takeaway from these results is that there will be a trade-off in performance if we make the definition of ID stricter. Our findings in Table 3 on the comparison between models are similar to our findings in Table 2 from the comparison from primary results.

In both Tables 2 and 3, we observed that supervised models demonstrate strong performance on specific types of OOD images, but we hypothesize that unsupervised/semi-supervised models (rows 4–7) may generalize better. To test this hypothesis, we evaluated models trained on `bollworms` and `bollworms-clean` on external datasets, namely the Stanford Dogs and Oxford Flowers 102 datasets, with results shown in Table 4. When evaluating on external data for models trained on the `bollworms` dataset, we observe that performance is relatively strong with accuracy greater than 0.90 for the Oxford Flowers dataset across all models, and with most greater than 0.90 for the Stanford Dogs dataset (with the exception of CAE with MSE at 0.72 and CAE with whitened density at 0.84). Notably, CSI performs equally well across both Stanford Dogs and Oxford Flowers, with 0.98 OOD accuracy for each; whereas other techniques demonstrate differential performance between the two datasets. When evaluating on external data

	Acc. (OOD): `bollworms`			Acc. (OOD): `bollworms-clean`		
Model	Bollworms	Dogs	Flowers	Bollworms	Dogs	Flowers
Handcrafted features	0.74	0.94	0.98	0.72	0.99	> 0.99
SSIM (k-means)	0.76	0.90	0.91	0.70	0.97	0.98
CNN	0.87	0.95	0.99	0.89	0.99	> 0.99
CAE (MSE)	0.44	0.72	0.91	0.64	0.96	> 0.99
CAE (RMD)	0.54	0.90	0.98	0.67	1.00	1.00
CAE (whitened density)	0.70	0.84	0.97	0.77	1.00	1.00
Contrastive learning (CSI)	0.73	0.98	0.98	0.53	> 0.99	> 0.99

Table 4. Results from experiments to assess model generalizability based on two external datasets: Stanford Dogs and Oxford Flowers 102. Each of the models trained on `bollworms` and `bollworms-clean`, respectively, were evaluated for accuracy on Wadhwani `bollworms`, Stanford dogs, and Oxford flowers images. All images considered as inputs to this table are OOD, hence the ideal performance for each model in this setting would be 1.00. Quantitative metrics are color-coded green if greater than or equal to 0.85, yellow if between 0.7 and 0.85, and red if less than 0.7.

for models trained on the `bollworms-clean` dataset, our unsupervised/semi-supervised models CAE (RMD) and CAE (whitened density) achieved 100% OOD detection accuracy, which provides evidence in support of our hypothesis that these approaches generalize better (even if marginally so). Future work can focus on validating generalization error on additional external datasets.

7. Future Work

In future directions, we want to continue training models with more data with better ID/OOD labels acquired via crowd-sourced voting procedures to resolve edge cases. In addition, it would be helpful to explore the generalization error of each method to a broader set of potential input OOD images by measuring the OOD accuracy of each model on additional external data sets beyond the Stanford Dogs and Oxford Flowers 102 datasets considered above.

Furthermore, while our team has performed an initial exploration of utilizing another density estimation method called normalizing flows, further investigation into the usefulness of this technique could be resourceful for the task at hand. Normalizing flows are a likelihood-based deep generative modeling approach that can be used to perform OOD detection (Kobyzev et al., 2021). Unlike other likelihood-based models, normalizing flows can produce an exact likelihood transformation rather than a lower bound or an approximation. We explored OpenAI's GLOW model, a simple type of generative flow using an invertible 1×1 convolution (Kingma & Dhariwal, 2018). The GLOW model aims to generate realistic high-resolution images and discovers features that can be used to manipulate attributes of the data, which is slightly different from our original goal of OOD detection. We were able to train on a toy dataset and saw some initial results. However, we discovered some key limitations of using normalizing flows for this specific task, including training time and requiring extensive computing resources. Given that the app is deployed on mobile platforms, resource-intensive models such as GLOW might not be realistic to use given the number of parameters. Initial results by the team flagged this technique as promising, but resource-intensive and non-trivial to implement.

Thirdly, in future work we hypothesize it could be potentially quite promising to apply a whitening transformation to the latent image representations obtained from a contrastive learning (CSI) model to combine the benefits of each model into one consolidated OOD detection approach. We see a strong possible potential in a synergy between these two techniques which were both top performers when utilized individually and we hypothesize that combining them together would provide an additive benefit.

8. Broader Impact

Improving the robustness of the app through accurate out-of-distribution detection will improve the overall user experience, increase user confidence in the app, and increase mobile app user retention, thus amplifying the effectiveness of Wadhwani AI's solution to the pest management problem. Helping farmers protect their crops from pests will naturally improve their crop yields, reduce their usage of (and exposure to) toxic pesticides, and will lead to better mental health outcomes by alleviating unnecessary stressors and anxiety. Since cotton pest infestations are a common challenge that farmers around the world are facing, a further social benefit could be a scalable solution that would be able to help farmers around the world protect their crops and increase farmland productivity. Our work could also potentially serve as a useful reference for the broader machine learning community by demonstrating how a solution to the out-of-distribution detection problem has been deployed in an AI for social good setting.

One potential negative social impact from automating cotton farm pest detection that we can foresee is its threat to the

jobs of farm extension workers who are not technologically savvy, but are hired to deal with the pests. A more effective version and widespread adoption of this app could create short-term unemployment within the farming community. If our OOD detection model is not effective, it can also lead to an overuse of pesticides. Given that a false negative is more concerning than a false positive in the case of bollworms, farmers would be incentivized to use pesticides whenever possible to prevent future infestations and mitigate the risk of losing their season's crops. A high usage of pesticides can have negative effects on the local biome, especially for organisms living in the soil. It can also potentially lead to the contamination of drinking water and an increased pest resistance overtime rendering them less and less effective for future seasons. There could also potentially be the risk of this technology being maliciously exploited by hackers to manipulate pesticides usage on cotton farms.

References

Bakurov, I., Buzzelli, M., Schettini, R., Castelli, M., and Vanneschi, L. Structural similarity index (ssim) revisited: A data-driven approach. *Expert Syst. Appl.*, 189 (C), mar 2022. ISSN 0957-4174. doi: 10.1016/j.eswa.2021.116087. URL https://doi.org/10.1016/j.eswa.2021.116087.

Bandyopadhyay, Hmrishav. Autoencoders in deep learning: Tutorial & use cases, 2022. URL https://www.v7labs.com/blog/autoencoders-guide. [Online; accessed December 6, 2022].

Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19, 2006.

Bowyer, K. W., Chawla, N. V., Hall, L. O., and Kegelmeyer, W. P. SMOTE: synthetic minority over-sampling technique. *CoRR*, abs/1106.1813, 2011. URL http://arxiv.org/abs/1106.1813.

Dalmia, A., White, J., Chaurasia, A., Agarwal, V., Jain, R., Vora, D., Dhame, B., Dharmaraju, R., and Panicker, R. Pest management in cotton farms: An ai-system case study from the global south. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3119–3127, 2020.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press, 2016.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions, 2018. URL https://arxiv.org/abs/1807.03039.

Kobyzev, I., Prince, S. J., and Brubaker, M. A. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979, nov 2021. doi: 10.1109/tpami.2020.2992934. URL https://doi.org/10.1109%2Ftpami.2020.2992934.

Ren, J., Fort, S., Liu, J., Roy, A. G., Padhy, S., and Lakshminarayanan, B. A simple fix to mahalanobis distance for improving near-ood detection. *CoRR*, abs/2106.09022, 2021. URL https://arxiv.org/abs/2106.09022.

Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., Dietterich, T. G., and Müller, K.-R. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5):756–795, 2021.

Su, J., Cao, J., Liu, W., and Ou, Y. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*, 2021.

Tack, J., Mo, S., Jeong, J., and Shin, J. CSI: Novelty detection via contrastive learning on distributionally shifted instances. *Advances in neural information processing systems*, 33:11839–11852, 2020.

White, J., Madaan, P., Shenoy, N., Agnihotri, A., Sharma, M., and Doshi, J. Wadhwani AI pest management open data. https://github.com/WadhwaniAI/pest-management-opendata, 2022a.

White, J., Madaan, P., Shenoy, N., Agnihotri, A., Sharma, M., and Doshi, J. A case for rejection in low resource ML deployment. *arXiv preprint arXiv:2208.06359*, 2022b.

Yang, J., Zhou, K., Li, Y., and Liu, Z. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021.

Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., and Chen, H. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*, 2018.