SANDIA REPORT
SAND2022-13025
Printed Click to enter a date

Sandia National Laboratories

Automatic Detection of Defects in High-Reliability Components

Kevin M Potter, Soroush Famili, Anthony P Garland, Jessica E Jones, Aniket Pant

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico
87185 and Livermore,
California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: https://classic.ntis.gov/help/order-methods/

ABSTRACT

Disastrous consequences can result from defects in manufactured parts—particularly the high consequence parts developed at Sandia. Identifying flaws in as-built parts can be done with non-destructive means, such as X-ray Computed Tomography (CT). However, due to artifacts and complex imagery, the task of analyzing the CT images falls to humans. Human analysis is inherently unreproducible, unscalable, and can easily miss subtle flaws. We hypothesized that deep learning methods could improve defect identification, increase the number of parts that can effectively be analyzed, and do it in a reproducible manner. We pursued two methods: 1) generating a defect-free version of a scan and looking for differences (PandaNet), and 2) using pre-trained models to develop a statistical model of normality (Feature-based Anomaly Detection System: FADS). Both PandaNet and FADS provide good results, are scalable, and can identify anomalies in imagery. In particular, FADS enables zero-shot (training-free) identification of defects for minimal computational cost and expert time. It significantly outperforms prior approaches in computational cost while achieving comparable results. FADS' core concept has also shown utility beyond anomaly detection by providing feature extraction for downstream tasks.

ACKNOWLEDGEMENTS

This work was supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

First, a thank you to all the interns who contributed to this project (Abigail Pribisova, Mike Adams, Aniket Pant, Soroush Famili, JayCe Leonard).

A thank you to Scott Roberts for helping to socialize our work.

We give special thanks to Chris Turner, Ariana Beste, John Korbin, and David Peterson for CT data and labels.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

This page left blank

EXECUTIVE SUMMARY

Problem

Disastrous consequences can result from defects in manufactured parts—particularly the high consequence parts developed at Sandia. For limited use components, such as body armor or explosives, the ability to predict failure of as-built parts would save lives, drastically improve efficiency, and increase confidence in performance. Currently, many single use components are validated with lot testing, which is wasteful. Furthermore, lot testing gives statistical evidence about how parts of a particular manufacturing run can be expected to perform, but it does not take into account attributes (e.g., cracks or shape) of an individual item to predict its performance.

Background

Analyzing images, such as non-destructive X-ray CTs, for flaws provides a valuable alternative to lot testing. However, no generally accepted method exists for identifying defects and anomalies in scans. For X-ray CT, the primary difficulty arises from the image artifacts created during the reconstruction process. Because of these artifacts, simple heuristic algorithms (such as thresholding to determine material type) often fail to produce useful results for CT scans. These artifacts also complicate downstream tasks, such as anomaly detection and component identification. As a result, human subject matter experts (SMEs) must identify and categorize outlying scans as either anomalous or normal. With some applications needing consistent assessments over multiple years or decades, this human dependency makes the process costly and unreproducible, thereby limiting potential applications.

Deep learning models have achieved human level (or superior) performance on diverse image datasets and tasks [1]–[5]. This includes generating high-resolution complex images[6]–[8]. Deep learning models can also classify complicated images[6]. Both of these capabilities have been exploited for anomaly detection[9]–[13].

We have shown that a generative approach can create realistic nominal images for a given input in a computationally efficient manner for a variety of image modalities, in both 2 and 3-D[14]. We implemented a generative network called PandaNet which can localize potential anomalies across a variety of datasets, both public and internal to Sandia.

Additionally, we explored classification approaches and devised a novel method based on the intermediate features created by pre-trained convolutional neural networks (CNNs). We call this method Feature-based Anomaly Detection System (FADS)[15]. Its major advantage is that it requires zero training, unlike most other classification approaches. This makes it practical both for low-compute settings and for users without prior deep learning knowledge.

Our contributions

- A novel CNN architecture (PandaNet) that outputs a normal version of a query image that supports 2 and 3-D data with arbitrary input channels
- FADS, a novel method using pre-trained networks to identify anomalous images
    - Does not require expensive updating of model weights
    - Preparation requires only a single pass of the normal images through a pre-trained model
    - Can be applied to any input format for which a pre-trained model is available

- A novel method of increasing CNNs' accuracy by eliminating the effect of object orientation in images
- A GUI tool which allows users to apply FADS to their own datasets is being developed

Conclusion

This LDRD sought to demonstrate a means of automatically detecting defects through imagery of various types. In this report and the prior works cited, we detailed two methods for detecting anomalies in a fast, scalable, and reproducible manner: PandaNet and FADS.

PandaNet[14] improved upon the performance of AnoGAN[12]. While we believe FADS shows more promise for defect and anomaly detection, PandaNet also improved the quality of image reconstructions and showed that a novel second reconstruction loss was beneficial.

FADS[15] brings several major advances to anomaly detection. First, it employs pre-trained models and requires zero additional training (i.e., no fine-tuning). This reduces both the computational cost and the expertise necessary to develop a top-quality model for detecting anomalies in images. These attributes allow anyone to use FADS for anomaly detection, particularly since we have developed a GUI for the algorithm. Second, we can cluster similar images (or CT slices) together using the algorithm's output for each image; this does not require prior knowledge about the images or their similarities. Third, our experiments show FADS can perform well with little normal data.

Several Sandia projects have incorporated FADS since it was developed less than a year and half ago. FADS produces features for one of the Voronoi applications, and other possible uses are being explored. The JARVIS project has employed FADS to identify anomalous activity in video and is experimenting with clustering activities in videos using FADS. FADS also enabled a clustering approach for CT scan slices and provided the initial testing ground for a novel rotation-invariant approach to neural networks.

After three years, the prospects for automatic defect detection are strong. We have developed new approaches (and accompanying software). Sandia researchers have already embraced these approaches and we believe they will empower a large number of ND applications in years to come.

ACRONYMS AND TERMS

Acronym/Term	Definition
AE	Autoencoder
AUROC	Area Under the Receiver Operator Curve
CAMI	Credible, Automated Meshing of Images
CNN	Convolutional Neural Network
COTS	Commercial Off The Shelf
CT	Computed Tomography
DL	Deep Learning
FADS	Feature-based Anomaly Detection System
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
GUI	Graphical User Interface
LDRD	Laboratory Directed Research and Development
Localization	Identifying regions of an input that contributed to a model's answer
ML	Machine Learning
ND	Nuclear Deterrence
NN	Neural Network
ROC	Receiver Operator Curve
r-vector	Normalized measure of anomalousness as compared to the normal dataset
SME	Subject Matter Expert
VAE	Variational Autoencoder

1. INTRODUCTION

1.1. Background

Non-destructive imaging techniques, such as X-ray CT, provide a valuable tool for ensuring quality in manufacturing. However, no generally accepted method exists for identifying defects and anomalies from scans. For X-ray CT, the primary difficulty arises from the image artifacts created during the reconstruction process. In X-ray CT, multiple X-rays images are taken from many angles around the object of interest. The X-ray CT reconstruction process produces a 3-D representation of the object from these images. During this, dense regions (from high Z material, e.g. most metals) can cause shadow-like artifacts that extend across the reconstruction[16]. Because of these artifacts, simple heuristic algorithms (such as thresholding to determine material type) often fail to produce useful results for CT scans. These artifacts also complicate downstream tasks, such as segmentation and anomaly detection. As a result, human subject matter experts (SMEs) must identify and categorize outliers as anomaly or nominal. With some applications needing consistent assessments over multiple years or decades, this human dependency makes the process costly, unreproducible, and limits potential applications.

Deep learning models have achieved human level (or superior) performance on diverse image datasets and tasks [1]–[5]. This includes generating high-resolution complex images from constrained domains[6]–[8]. Deep learning models can also classify complicated images[6]. Both of these capabilities have been exploited for anomaly detection[9]–[13].

We have shown that a generative approach can create realistic nominal images for a given input in a computationally efficient manner for a variety of image modalities, in both 2 and 3-D[14]. We implemented a generative network called PandaNet which can localize potential anomalies across a variety of datasets, both public and internal to Sandia.

Additionally, we explored classification approaches and devised a novel method based on the intermediate features created by pre-trained convolutional neural networks (CNN). We call this method Feature-based Anomaly Detection System (FADS)[15]. While it no longer matches state-of-the-art accuracy for the anomaly detection benchmark dataset, MVTec Anomaly Detection dataset (MVTecAD)[17], it retains a major advantage over newer methods in that it requires zero training. This makes it practical both for low compute settings and for users without prior deep learning knowledge.

1.2. Our contributions

- A novel CNN architecture (PandaNet) that outputs a close nominal version of a query image (Subsection 2.1.1)
    - Supports 2-D and 3-D data with arbitrary input channels
    - Demonstrated a new cyclic loss improvement applicable to any autoencoder
- A novel method using pre-trained networks to identify anomalous images directly (Subsection 2.1.2)
    - Does not require updating model weights
    - Preparation instead consists of a single forward pass of the nominal images through a pre-trained CNN

    - Can be applied to any input form that has a pre-trained model available
- A novel method of making a CNN layer rotation invariant (Subsection 3.4)
- A GUI tool which allows users to apply FADS to their own datasets is being developed (Subsection 3.1)
    - No computer vision/deep learning experience required
    - Support for images and video
    - Local and server supported workflows
- Makes use of available GPUs to improve speed but they are not required

2. PRIOR PUBLICATIONS AND RELATED WORK

2.1. Prior publications

Here we list work and give a brief summary of work detailed in prior publications.

2.1.1. Generative (PandaNet)

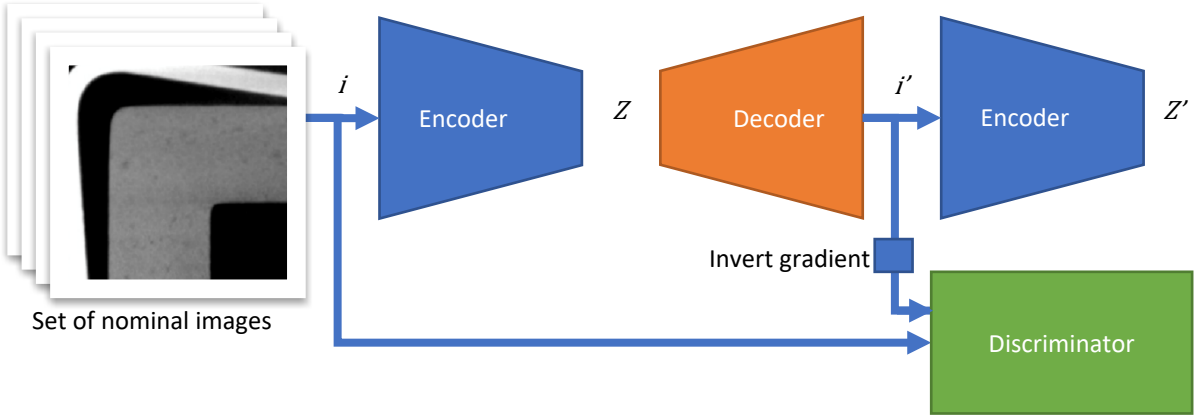

Figure 1 PandaNet architecture[14]. During training, the encoder learns to take a nominal image *i* to an output latent space *Z* while a decoder learns to reconstruct the image output as *i'*. The discriminator is tasked with identifying whether *i* or *i'* is the real image. Since the generator and discriminator have opposite goals, gradients from the discriminator to the generator are inverted. Additional mean absolute losses are calculated between *i* and *i'* as well as between *Z* and *Z'*.

Generative approaches learn to produce a defect-free image which is similar to an input image. By comparing this generated nominal image to the original input image, we can highlight potential anomalies by simply taking the difference between the two. Schlegl et al. [12] first demonstrated this using a GAN. Donahue et al. [18] applied this approach to Sandia datasets. Building on this work, we created a faster approach to generating closest nominal images. The architecture for this new approach is shown in Figure 1. See Potter et al. [14] for complete details.

2.1.2. Classification (FADS)

Ruff et al.[19, p.] demonstrate deep one-class classification, a classification-based approach to identifying anomalies. They train a network to place nominal images into a small region of feature

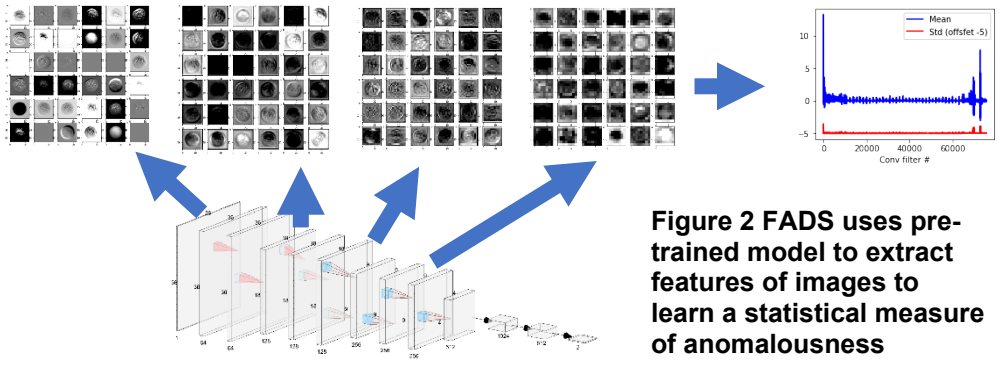

Figure 2 FADS uses pre-trained model to extract features of images to learn a statistical measure of anomalousness

space. While this works somewhat, the network is prone to mode collapse: i.e., predicting the same output regardless of input.

We adapt this concept to use pre-trained convolutional neural networks and the entirety of the convolutional feature space for anomaly detection[15]. By passing an image through this network and recording the activations, we obtain a rich set of features without costly training (see Figure 2). We calculate the mean and standard deviation of a set of nominal image activations to establish a baseline. For a new image, we can produce a normalized vector (the r-vector) which represents how many standard deviations the image's activations vary from the nominal set's mean. Large deviations are indicative of abnormal behavior. See Garland et al[15] for complete details

2.2. Related work

Anomaly detection has widespread applications in such diverse areas as credit card fraud detection[20], medical diagnosis [11], [12], and manufacturing. For a more in-depth review, please refer to [21]–[23].

Our generative methods (see Section 2.1.1) rely on an adversarial approach called a Generative Adversarial Network (GAN)[24], as described in the original AnoGAN[12] and f-AnoGAN[11] papers. AnoGAN uses a generative model to identify anomalous sections of a 2-D medical scan in an unsupervised manner. The technique requires iterative backpropagation during query time which leads to slow performance. F-AnoGAN adds an encoder which learns a direct mapping from input image to latent space. This considerably shortens inference time by eliminating the backpropagation requirement.

When limited training data is available, deep learning practitioners commonly compensate by utilizing a neural network that has been trained on some auxiliary task(s) for which large datasets are available [25]. The pre-trained model might undergo a final fine-tuning[26], [27] (a short training process updating all or a portion of network weights) or be used as-is[28]. Such transfer learning has succeeded in the natural language processing[29]–[31] and image domains [32]–[34].

As described in Section 2.1.2, the deep one class classification method trains a CNN to output nominal images to a small portion of the feature output hyperspace centered around some point C[19]. A query image is passed through the CNN, and the output vector's distance from C is used to determine whether the image is considered anomalous. Unfortunately, the network may map all output vectors near C regardless of the input image. Various regularization techniques have been proposed to solve this problem[19], [35], [36].

3. METHODS

Please see Potter et al.[18] for details on PandaNet methods and Garland et al.[15] for details on FADS. FADS pre-trained models were obtained from PyTorch's[37] model hub unless otherwise noted.

3.1. FADS GUI

To improve the usability of FADS, we are developing a web application which will enable anyone, regardless of their level of deep learning (DL) expertise, to apply the FADS algorithm to their own data. To this audience, the intricacies of the algorithm itself are unimportant and should be abstracted away. They just want to run the model and be able to easily interpret the results. The graphical user interface allows us to show the user which pixels in an image caused the FADS algorithm to categorize the image as anomalous. We identify these localized anomalies by using guided backpropagation to minimize the r-vector.

Figure 3 illustrates the layout of the user-interface. The user indicates in windows (a) and (b) which data to use for training and testing: either selected local files or files on a given path (which may be

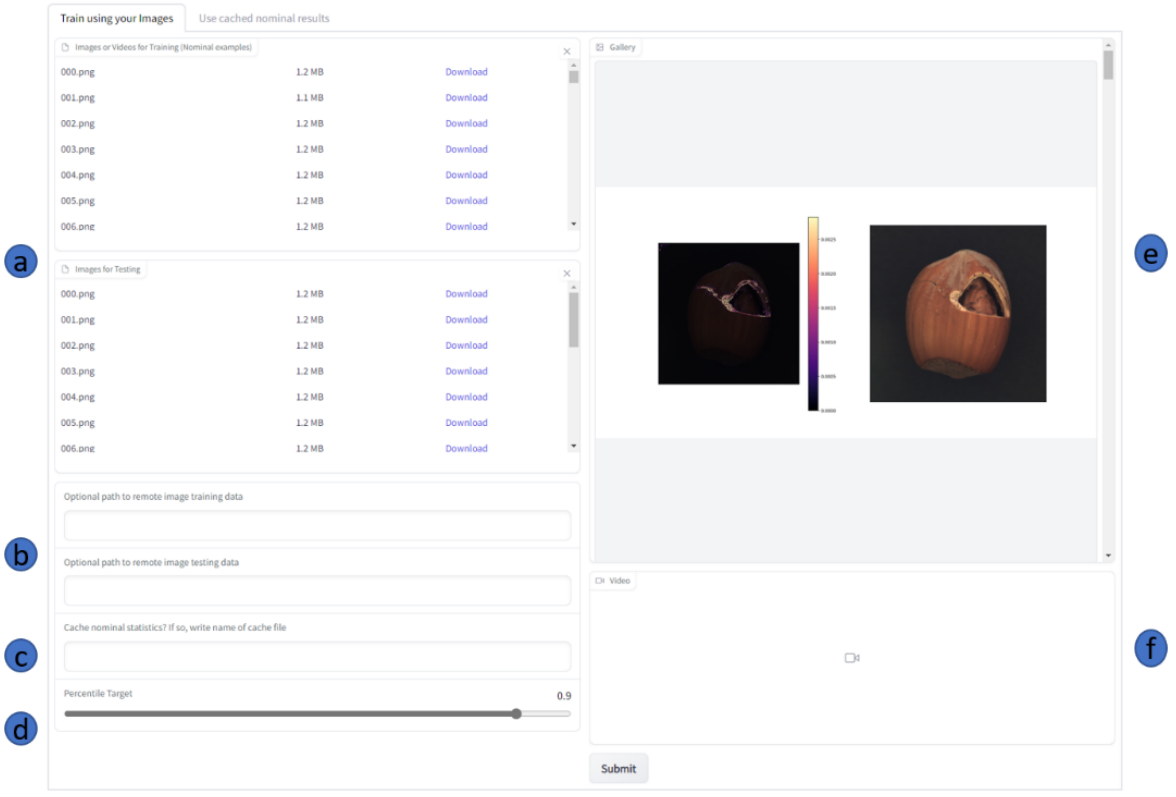

Figure 3 shows the "Training using your Images" tab of the user-interface. a) Where the user can drag and drop or go into their directories to find training and testing data. b) Where the user can instead insert a local or remote directory path to pull their data. c) Where the user can input a filename to cache the nominal statistics. d) Percentile target selection for localization. e) Where the test image localizations and frame localizations are displayed. f) Where the playable video of the localized frames would be displayed if initial data was video.

on a remote server). If the user desires to save the mean and standard deviation of the nominal dataset's activations in order to skip training in the future, they may give a filename at (c). This will cache the nominal statistics to a Python pickle file. The user can change the percentile[1] target visualization hyperparameter at (d). The user would then press the "submit" button and the FADS algorithm will run. At (e), the localization of anomalousness alongside their respective test image inputs are displayed. If the input was a video, (e) will show the localizations for each individual frame of the testing frames of the video. If the input was a video, (f) is where the playable video of the anomaly localized frames is displayed and can be played right from the user interface.

We chose Gradio[38] as the web application to host our user interface. It is an open-source Python library specifically for machine learning models that can run on the local machine or a remote server.

Figure 4 shows the FADS GUI workflow. The user inputs their training and testing data and chooses a percentile target which is used to determine anomalousness. For images, the training and testing data get copied from their original location and then put together into new training and testing folders. Because the training or testing data could come from multiple sources, we create new folders to aggregate the training and testing data. For video, the process is the same except that we

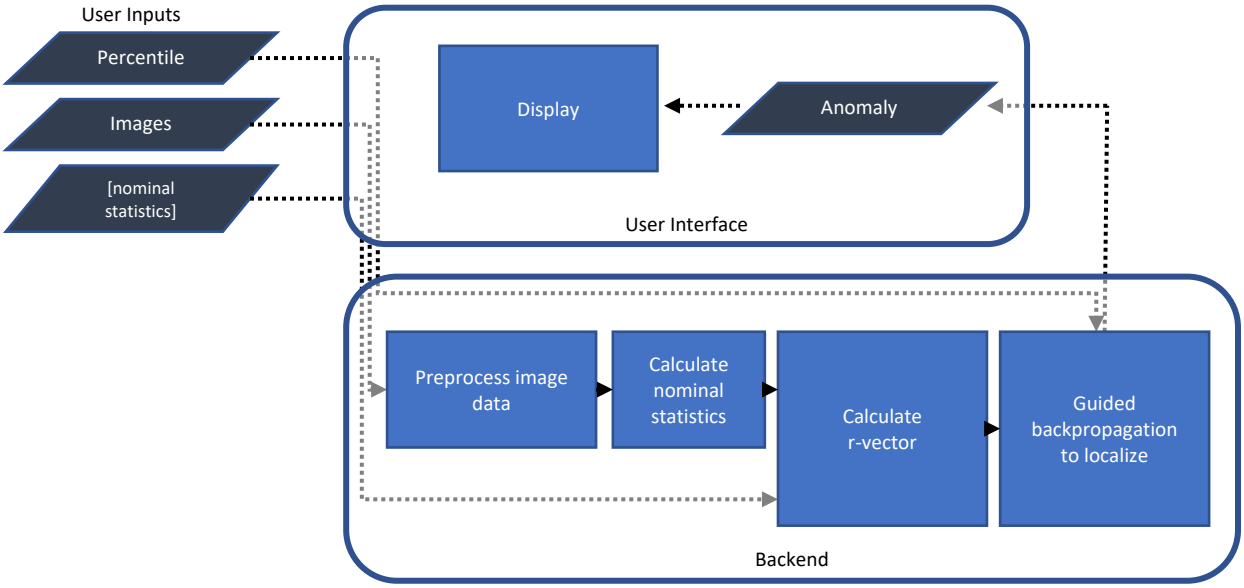

Figure 4 Program structure of FADS GUI

take a certain percentage of the initial frames as training data and the rest of the frames as testing data. The rest of the process is identical. Next, the data is resized to a size specified in a configure file and is then ready for FADS to calculate the r-vector.

This work is very early and will likely change as feedback is gathered from use.

3.2. JARVIS

The Justified Anomaly Recognition in Video Surveillance (JARVIS) project, sponsored by NA-241, develops image analysis tools for International Atomic Energy Agency (IAEA) safeguards inspectors. JARVIS particularly emphasizes unsupervised and computationally efficient approaches,

[1] Percentile of the r-vector values to use as the threshold for determining anomalouslness. Percentile takes the location at x% of the overall sorted list of values (e.g., 100% would be max, 50% median, 0% min).

given limits on the time and resources available to inspectors. The JARVIS team applied the FADS[15] algorithm initially developed for 2-dimensional data to individual frames from IAEA surveillance cameras. The JARVIS team also extended FADS to be able to use CNNs pre-trained on video datasets.

3.3. Clustering

The FADS[15] r-vector output has utility beyond anomaly detection. Initial work suggests that the features can be directly incorporated into other ML methods as a quick, computationally efficient feature extractor. For example, we clustered the r-vectors from each slice of a CT scan via k-means[39]. This provided insight into the structure of the component (along an axis) without requiring any prior knowledge about said structure.

3.4. Rotation invariance

Some of the categories in the MVTecAD[17] dataset (most notably the screw category) contain images taken from multiple angles. Comparing the features output for the screw category, we found that the separation between the nominal and anomaly class means was smaller than the standard deviation for both classes and smaller than the minimum distance for any image from the class mean. Additionally, an appropriate hyperplane could perfectly separate the two classes. These two facts, plus our knowledge of the data, suggested that rotation was actually one of the largest factors to feature variance.

The networks used for FADS[15] are all CNNs. Specifically, FADS considers the activations of the convolutional layers. Each convolutional layer has a set of weights (typically in a 3×3 grid). That set of weights is multiplied by each block of the input image's channels or the previous layer's feature maps, and the sum for each location becomes the input for the next layer to use. Depending on the weights, the activation may or may not respond differently to horizontally and vertically aligned inputs (see Figure 5a versus Figure 5b). This difference in response means the CNN will behave differently depending on the input orientation. This can cause FADS to label rotated image as anomalous incorrectly or to extend the statistics for normality, reducing the algorithm's sensitivity.

We devised an approach to address this issue. We pass the image through the CNN twice: once using the default weights and once with weights transposed (or, equivalently, first rotating the input

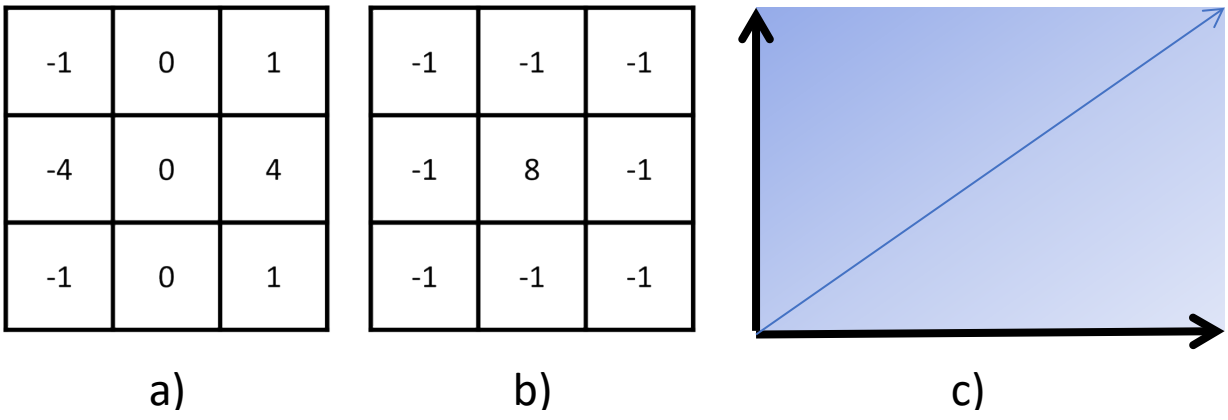

Figure 5 Convolutional filter examples a) An example of a convolution that will respond only to changes in the x direction. After transposition, it will respond only to y direction. b) An example of convolution that will be the same after transposition. c) By taking the vector norm of the "x" and "y" aligned activations, we get a measure that is independent of the rotation of the input.

image and then the output feature maps). This produces a vector for each point in the feature map. If we calculate the vector norm for each of these vectors, we get a rotation-independent activation or at least an approximation to one.

NOTE: For easier implementation, we transpose the image instead of the weights and then transpose the resultant feature map (to bring the two maps back into alignment). Mathematically, the two approaches are equivalent.

We tested multiple ways to integrate this approach into the FADS algorithm:

1. Rotation-invariant output by itself (i.e., replacing the r-vector)
2. Normal output and rotation-invariant output combined by aggregation
3. Normal output and rotation-invariant output combined by concatenation

We discuss the results of these approaches in subsection 4.3.

4. RESULTS

4.1. JARVIS

The JARVIS team applied FADS to a set of safeguards surveillance camera footage received from the IAEA. This dataset consists of images from a single camera collected over 22 days. Most images show a large red container on the right side. The nominal set contained clips selected at regular intervals over two consecutive days during which the red container did not move, although other activity occurred in the room. The test set contained clips from two consecutive days during which the container was moved; the movement occurred at the end of the first day and continued through the beginning of the second. JARVIS researchers selected clips from the beginning and end of these two days, ensuring the test dataset included frames with and without the red container in its usual position. The nominal dataset contained 1504 images and the test dataset contained 3000 images.

The ten frames in the test dataset which FADS scored as least anomalous relative to the nominal set all show the red container in its usual position. In all ten frames which scored as most anomalous, the red container has been moved out of view. When researchers reviewed the 30 most anomalous frames, two show the container in the usual position. The algorithm likely considered these anomalous because the overhead crane was in motion. This result underscores the importance of a large and diverse nominal set: if we want the algorithm to consider crane motion normal, we should ensure the nominal set includes several images with the crane in motion.

4.2. Clustering

We analyzed a CT image of shaped charge using the clustering approach described in Section 3.3; Figure 6 depicts the results. We ran the FADS algorithm on the 2D images representing slices along the scan's z-axis (left to right). We clustered the resulting r-vectors using k-means[39] with k=7. Colored lines in Figure 6 show the portion of the scan assigned to each cluster. For this example, the clustering corresponds to different relevant regions of the object.

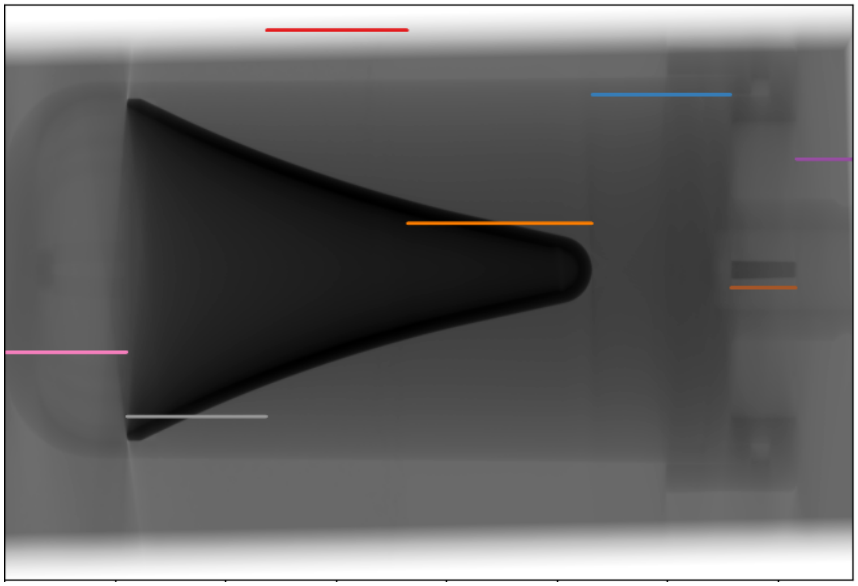

Figure 6 Clustering against each slice of a CT scan.

4.3. Rotation invariance

Datasets may contain images with different orientations. FADS[15] aims to provide results that do not depend on a uniform orientation. For example, images in the MVTecAD[17] "screw" show screws in varying orientations: i.e., some images show the tip of the screw pointing to the right, some images show it pointing to the bottom of the image, etc.

We measure FADS' performance in identifying anomalies with the Area Under Receiver Operator Curve (AUROC). The Receiver Operator Curve (ROC) plots the true positive rate versus false positive rate as a threshold is varied. In an ideal case, the AUROC would be 1 indicating a perfect ability to separate anomaly from nominal classes. A guessing version would have an AUROC of 0.5.

Using ensembles[2] of FADS r-vectors from various models, at different scales, and with and without rotation invariance, we tested samples across all MVTecAD categories. To understand potential performance improvements, we designed baseline studies using the pre-trained CNNs ResNet18 and ResNet152.

Initial results show mixed benefit from inclusion of rotation invariance (Table 1). No method shows a clear win over others for all categories.

Data trends:

- Ensembles at image scales of 256 and 512 do well
- Those including ResNet152 also show good performance
- ResNet18 has generally poor performance, but it is significantly improved by combining "vanilla" and rotation-invariant methods
- Some categories are effectively solved with an AUROC of nearly 1 (hazelnut, leather, tile, wood)
- Many categories show significant hits to performance at high (1024) and low (128) resolutions

Comparing averages over methods using either only rotation invariance, only vanilla, or only both, we see a slight improvement for using both together, but it is not a significant change. Performance in some categories significantly improve with rotation invariance (e.g., bottle, carpet, and grid), some categories see declines (e.g., capsule, pill), and rotation invariance has no effect on others. As a result, we can make no clear recommendation as to whether to apply rotation invariance to a new dataset.

[2] Ensembles are a collection of ML models used together to improve accuracy, reduce dependence on hyperparameter selection, or provide a measure of uncertainty.

Table 1 FADS ensemble AUROC results for MVTecAD categories. Ensembles were varied across scale (128, 256, 512, 1024), rotation invariance (with – rotinv, without - vanilla, both), and model (ResNet18 and ResNet152). Any variance not mentioned in the name is included (e.g., ensemble-RN152 includes all combinations of scale, rotation invariance that use ResNet152). Ensembles have individual r-vectors concatenated before checking using the maximum to determine image anomalousness.

	bottle	cable	capsule	carpet	grid	hazelnut	leather	metal nut	pill	screw	tile	toothbrush	transistor	wood	zipper	average
ensemble-everything	0.965	0.882	0.873	0.989	0.942	0.995	0.997	0.964	0.904	0.784	0.996	0.928	0.922	0.979	0.680	0.920
ensemble-RN152	0.911	0.879	0.888	0.989	0.942	0.994	0.997	0.965	0.902	0.788	0.996	0.950	0.917	0.971	0.677	0.918
ensemble-RN18	0.975	0.897	0.869	0.958	0.893	0.980	0.994	0.907	0.884	0.794	0.991	0.869	0.855	0.989	0.804	0.911
ensemble-rotinv	0.952	0.893	0.861	0.990	0.952	0.989	0.996	0.951	0.858	0.788	0.997	0.933	0.891	0.976	0.718	0.916
ensemble-vanilla	0.952	0.876	0.928	0.984	0.937	0.995	0.995	0.966	0.931	0.796	0.995	0.903	0.923	0.994	0.699	0.925
ensemble-rotinv-RN152	0.927	0.893	0.877	0.990	0.952	0.989	0.996	0.953	0.859	0.795	0.997	0.933	0.885	0.969	0.689	0.914
ensemble-vanilla-RN152	0.856	0.875	0.928	0.984	0.937	0.993	0.995	0.964	0.929	0.790	0.994	0.922	0.922	0.995	0.727	0.921
ensemble-rotinv-RN18	0.965	0.846	0.842	0.966	0.855	0.980	0.996	0.827	0.869	0.700	0.991	0.925	0.851	0.992	0.882	0.899
ensemble-vanilla-RN18	0.973	0.894	0.887	0.947	0.840	0.973	0.989	0.920	0.871	0.781	0.994	0.847	0.851	0.996	0.767	0.902
ensemble-scale-128	0.922	0.915	0.765	0.904	0.708	0.983	1.000	0.907	0.856	0.608	0.996	0.911	0.899	0.994	0.823	0.879
ensemble-scale-256	0.926	0.951	0.916	0.981	0.908	0.996	1.000	0.976	0.889	0.725	0.997	0.825	0.919	0.972	0.865	0.923
ensemble-scale-512	0.959	0.911	0.907	0.981	0.920	0.989	0.999	0.976	0.905	0.775	0.996	0.936	0.897	0.989	0.735	0.925
ensemble-scale-1024	0.948	0.831	0.866	0.918	0.957	0.961	0.981	0.870	0.852	0.811	0.980	0.953	0.855	0.994	0.633	0.894
ensemble-scale-rotinv-128	0.957	0.911	0.722	0.928	0.741	0.986	1.000	0.916	0.798	0.643	0.996	0.914	0.868	0.994	0.824	0.880
ensemble-scale-rotinv-256	0.972	0.919	0.898	0.986	0.896	0.989	1.000	0.946	0.842	0.733	0.997	0.825	0.881	0.969	0.936	0.919
ensemble-scale-rotinv-512	0.960	0.895	0.892	0.984	0.944	0.987	0.997	0.987	0.898	0.742	0.997	0.922	0.848	0.989	0.756	0.920
ensemble-scale-rotinv-1024	0.918	0.858	0.852	0.903	0.942	0.963	0.974	0.892	0.814	0.794	0.973	0.972	0.862	0.979	0.702	0.893
ensemble-scale-vanilla-128	0.860	0.887	0.780	0.882	0.683	0.985	1.000	0.903	0.869	0.523	0.992	0.900	0.911	0.996	0.799	0.865
ensemble-scale-vanilla-256	0.883	0.947	0.901	0.953	0.906	0.998	0.998	0.974	0.888	0.700	0.997	0.831	0.942	0.993	0.845	0.917
ensemble-scale-vanilla-512	0.948	0.930	0.915	0.964	0.908	0.981	1.000	0.970	0.916	0.798	0.994	0.906	0.909	0.989	0.787	0.928
ensemble-scale-vanilla-1024	0.925	0.809	0.920	0.896	0.937	0.966	0.984	0.863	0.898	0.798	0.979	0.919	0.808	0.996	0.672	0.891
Average over ensembles using only rotation invariance	**0.950**	0.888	0.849	**0.964**	**0.898**	0.983	0.994	0.925	0.848	0.742	**0.993**	**0.918**	0.869	0.981	**0.787**	0.906
Average over ensembles using only vanilla	0.914	0.888	**0.894**	0.944	0.878	0.984	**0.995**	0.937	**0.900**	0.741	0.992	0.890	**0.895**	**0.994**	0.757	0.907
Average of ensembles using both	0.944	**0.895**	0.869	0.960	0.896	**0.985**	**0.995**	**0.938**	0.885	**0.755**	**0.993**	0.910	**0.895**	0.984	0.745	**0.910**

5. PROJECT METRICS

5.1. Publications

- Potter et al. *Automatic detection of defects in high reliability as-built parts using x-ray CT,* Applications of Machine Learning 2020 [2]
- Garland et al., *Feature anomaly detection system (FADS) for intelligent manufacturing*, (in review, preprint [3])

5.2. Presentations

- Applications of Machine Learning 2020
- Seagate-Minnesota AI/ML Virtual Distinguished Speaker Series (invited)
- ASME VVUQ 2022 Symposium

5.3. Career development

- Anthony Garland converted from postdoc
- Interns supported:
  - Undergraduate
    - Abigail Pribisova
    - Aniket Pant
    - Mike Adams
    - JayCe Leonard
  - Graduate
    - Soroush Famili

5.4. Partnerships

- We shared PandaNet and FADS software with members of org 5522 in an ongoing partnership to advance anomaly detection at Sandia
- The JARVIS project is using FADS for video analysis and is exploring use of FADS to cluster activity trends automatically into a human recognizable format
- The Voronoi LDRD using FADS for feature extraction and is exploring the use of clustering for semi-supervised segmentation
- We have shared FADS GUI software with 7637 in an ongoing partnership to help them identify potential defects in custom cables

5.5. Life after

A number of proposals, potential applications, and actual applications have been built across the labs:

- Working with 1819 on identifying potential defects in a COTS ND component *continuing*
- Working with 9748 to verify vaccination proof submissions *continuing*
- Working with 2323 to identify defects in connector assemblies *agreement being drafted*
- Working with 7642 to aid processing CT data for COTS components *waiting on data*

- Working with 7568 to identify anomalies and predict performance on an NG component *waiting on data*
- Both Kansas City and Sandia manufacturing have expressed interested in using FADS within their product lines *open talks*
- Proposal with 1851 to apply FADS to CT data *proposal*
- A&L proposal to identify aging in energetic materials with 7555 *proposal*
- INWAP proposal with 5571 to study the potential defect detection benefits of FADS compared to solely human examination *proposal*

6. CONCLUSION

This LDRD sought to demonstrate a means of automatically detecting defects through imagery of various types. In this report and the prior works cited, we detailed two methods for detecting anomalies in a fast, scalable, and reproducible manner: PandaNet and FADS.

PandaNet[14] improved upon the performance of AnoGAN[12]. While we believe FADS shows more promise for defect and anomaly detection, PandaNet also improved the quality of image reconstructions and showed that a novel second reconstruction loss was beneficial.

FADS[15] brings several major advances to anomaly detection. First, it employs pre-trained models and requires zero additional training (i.e., no fine-tuning). This reduces both the computational cost and the expertise necessary to develop a top-quality model for detecting anomalies in images. These attributes allow anyone to use FADS for anomaly detection, particularly since we have developed a GUI for the algorithm. Second, we can cluster similar images (or CT slices) together using the algorithm's output for each image; this does not require prior knowledge about the images or their similarities. Third, our experiments show FADS can perform well with little nominal data.

Several Sandia projects have incorporated FADS since it was developed 18 months ago. FADS produces features for one of the Voronoi applications, and other possible uses are being explored. The JARVIS project has employed FADS to identify anomalous activity in video and is experimenting with clustering activities in videos using FADS. FADS also enabled a clustering approach for CT scan slices and provided the initial testing ground for a novel rotation-invariant approach to neural networks.

After three years, the prospects for automatic defect detection are strong. We have developed new approaches (and accompanying software). Sandia researchers have already embraced these approaches and we believe they will empower a large number of ND applications in years to come.

REFERENCES

[17] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD–A comprehensive real-world dataset for unsupervised anomaly detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9592–9600.

[18] E. A. Donahue, T.-T. Quach, K. Potter, C. Martinez, M. Smith, and C. D. Turner, "Deep learning for automated defect detection in high-reliability electronic parts," in *Applications of Machine Learning*, 2019, vol. 11139, pp. 30–40.

[19] L. Ruff *et al.*, "Deep one-class classification," in *International conference on machine learning*, 2018, pp. 4393–4402.

[20] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis," in *2017 International Conference on Computing Networking and Informatics (ICCNI)*, Oct. 2017, pp. 1–9. doi: 10.1109/ICCNI.2017.8123782.

[21] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[22] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

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

[24] I. Goodfellow *et al.*, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[25] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *International conference on artificial neural networks*, 2018, pp. 270–279.

[26] Y. Guo, H. Shi, A. Kumar, K. Grauman, T. Rosing, and R. Feris, "Spottune: transfer learning through adaptive fine-tuning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4805–4814.

[27] H.-W. Ng, V. D. Nguyen, V. Vonikakis, and S. Winkler, "Deep learning for emotion recognition on small datasets using transfer learning," in *Proceedings of the 2015 ACM on international conference on multimodal interaction*, 2015, pp. 443–449.

[28] J. W. Soh, S. Cho, and N. I. Cho, "Meta-transfer learning for zero-shot super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3516–3525.

[29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[30] S. Gururangan *et al.*, "Don't stop pretraining: adapt language models to domains and tasks," *arXiv preprint arXiv:2004.10964*, 2020.

[31] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146*, 2018.

[32] S. U. H. Dar, M. Özbey, A. B. Çatlı, and T. Çukur, "A transfer-learning approach for accelerated MRI using deep neural networks," *Magnetic resonance in medicine*, vol. 84, no. 2, pp. 663–685, 2020.

[33] M. M. Ghazi, B. Yanikoglu, and E. Aptoula, "Plant identification using deep neural networks via optimization of transfer learning parameters," *Neurocomputing*, vol. 235, pp. 228–235, 2017.

[34] S. Tammina, "Transfer learning using vgg-16 with deep convolutional neural network for classifying images," *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 10, pp. 143–150, 2019.

[35] L. Ruff *et al.*, "A unifying review of deep and shallow anomaly detection," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 756–795, 2021.

[36] L. Ruff *et al.*, "Deep semi-supervised anomaly detection," *arXiv preprint arXiv:1906.02694*, 2019.

[37] A. Paszke *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
[38] A. Abid, A. Abdalla, A. Abid, D. Khan, A. Alfozan, and J. Zou, "Gradio: Hassle-Free Sharing and Testing of ML Models in the Wild," *arXiv preprint arXiv:1906.02569*, 2019.
[39] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Journal of the royal statistical society. series c (applied statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

DISTRIBUTION

Email—Internal

Name	Org.	Sandia Email Address
Chris Turner	2434	cdturn@sandia.gov
Satyanadh Gundimada	5522	sgundim@sandia.gov
Charlie Snider	5575	cjsnide@sandia.gov
Ariane Beste	7555	abeste@sandia.gov
Technical Library	1911	sanddocs@sandia.gov

This page left blank

Sandia National Laboratories

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.