Instrument Science Report WFC3 2022-03

WFC3/UVIS Figure-8 Ghost Classification using Convolutional Neural Networks

F. Dauphin, M. Montes, N. Easmin, V. Bajaj, P. R. McCullough

March 25, 2022

Abstract

"Figure-8 ghosts" are a type of HST/WFC3 anomaly caused by light reflecting off a part of the WFC3/UVIS detector. Since anomalies affect both science and calibration data, it is vital that we flag them to properly support the WFC3 user community. Traditional methods of anomaly detection will become impractical as the volume of data sets increases in future missions such as the Roman Space Telescope. WFC3 data provides the opportunity to test anomaly detection with machine learning in current generation astronomical images. We trained five convolutional neural networks (CNNs) of varying architectures to classify figure-8 ghosts in WFC3/UVIS images. This report outlines the data selection, preparation, and augmentation to construct our training, validation, and test sets. Our models' accuracies range from 62% to 83%, showing that CNNs can be a powerful tool to detect other anomalies in WFC3 or other images. In addition to applying machine learning to a specific astronomical task and discussing future work, we consider astronomy's role in machine learning, i.e., innovating computer vision with large images, as a potential challenge for the future.

1. Introduction

The Wide Field Camera 3 (WFC3) instrument on board the Hubble Space Telescope (HST) has fascinated astronomers and the public for over a decade with its phenomenal ability to capture the universe in high resolution. Although these images are stunning, they are sometimes affected by visual artifacts. These anomalies can reduce their aesthetic quality and sometimes their potential for scientific research (Gosmeyer et al. 2017). It is important that these anomalies are identified to a high accuracy in order to better exploit WFC3 images.

With the expansion of accessible resources for artificial intelligence, the sciences have begun to utilize machine learning (ML) as a means of advancing science at an unprecedented rate. Enabled by powerful ML libraries in Python released over the past decade, the astronomy community has seen an explosion of ML-related papers, such as predicting galaxy morphology, predicting spectra of galaxies, and classifying exoplanets (Dieleman et al. 2015; Wu & Peek 2020; Valizadegan et al. 2021). The impact of ML on astronomy and astrophysics will be significant; we look forward especially to its potential for enabling scientific discovery.

Aligned with the rise of ML is astronomy's entrance into the era of big data. Due to the volume of data anticipated from future observatories, such as the Vera Rubin Observatory and Roman Space Telescope, traditional analysis techniques may become unfeasible (Eifler et al. 2020). Anomaly detection in images, a common analysis task, is traditionally performed by a person or team of people manually sorting through images, which could be time consuming depending on data volume, rate, and variety. However, ML excels in anomaly detection (Pang et al. 2021). Therefore, it is essential that we take advantage of ML as a solution for anomaly detection to enable high-efficiency astrophysics research. To prepare for this data-driven paradigm shift, we make use of WFC3 images, an ideal testing ground for ML-based anomaly detection in astronomical images due to the extensive volume and variety of data. We previously built a model to classify WFC3/IR blobs and plan to expand on that work to other anomalies (Dauphin et al. 2021). By developing models for a well-maintained instrument such as WFC3, we can automate time-consuming processes in addition to building human expertise and software tools that directly support the astronomy community.

The report is organized as follows: in Section 2, we define a figure-8 ghost and the proposed methods for identifying it. In Section 3, we describe our data selection, processing, and augmentation pipeline to prepare our training, validation, and test sets. In Section 4, we describe our models' various training procedures. In Section 5, we review and compare efficacies of our models on validation and test sets. In Section 6, we discuss topics of interest

with regards to this work and the astronomy community's role in machine learning. In Section 7, we present the conclusions. In addition, the Appendix contains our models' architectures, hyperparameters, and supplementary figures.

2. Identifying Figure-8 Ghosts

2.1. The Figure-8 Ghost Anomaly

Figure-8 ghosts are the most common anomaly for the WFC3/UVIS detector, with 5924 observations containing the anomaly as of September 2021. The ghost image looks like the numeral 8 rotated 45 degrees counterclockwise. It appears when some small fraction of the light from an object incident upon quadrant D (see Figure 2.1 of Sahu et al. (2021)) is reflected back towards the CCD's window. Then, some of that light is reflected again off the window back towards the CCD, especially in quadrant A along the diagonal.

Due to the extra path length, the ghosts images are defocused, and because the window glass has both front and back surfaces, the ghosts come in pairs. Because of the tilted optics, the ghosts look like elongated ellipses, not circles, and are distributed on the diagonal from quadrant A to quadrant D (Gosmeyer et al. 2017). Figure 1 illustrates the appearances and the locations of figure-8 ghosts. McCullough (2011) provided a detailed description of the ghost's geometric model and required optics. Typically, they are contained within a 300 pixel by 300 pixel to 500 pixel to 500 pixel box. The intensity of a figure-8 ghost is correlated with the intensity of the parent object it is reflecting; heavily saturated sources produce bright figure-8 ghosts, and low intensity sources (above some threshold) produce fainter figure-8 ghosts. They can also appear partially off the frame, with a portion of the ghost landing off the edge of the detector (see top left of Figure 1). The number of figure-8 ghosts in an observation is determined by the position and brightness of sources in quadrant D. Observations with the anomaly typically contain one to four ghosts. Figure 2 illustrates some examples of observations with figure-8 ghosts.

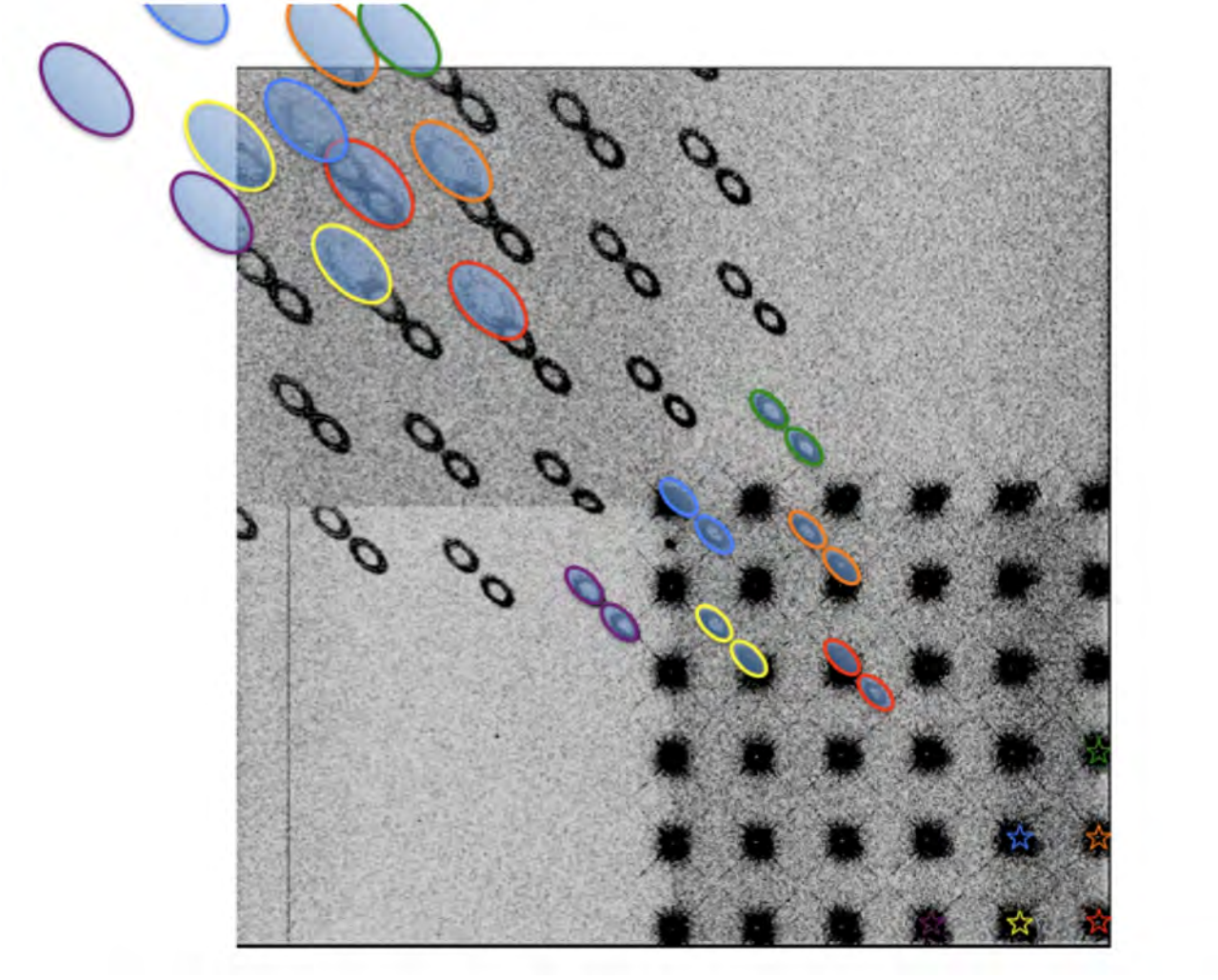
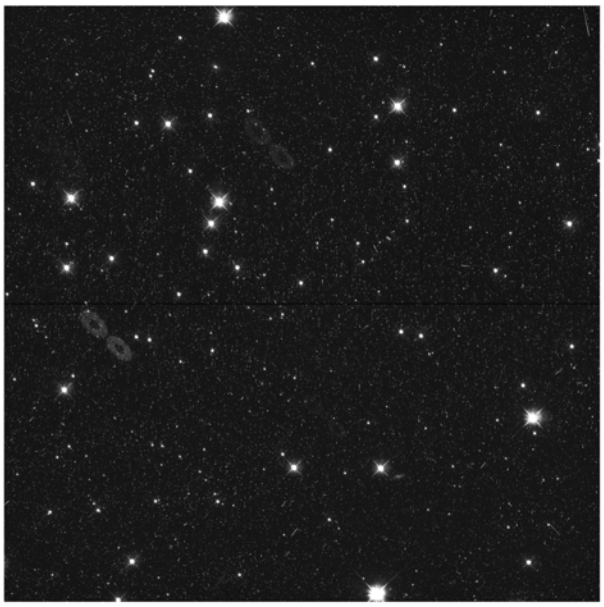
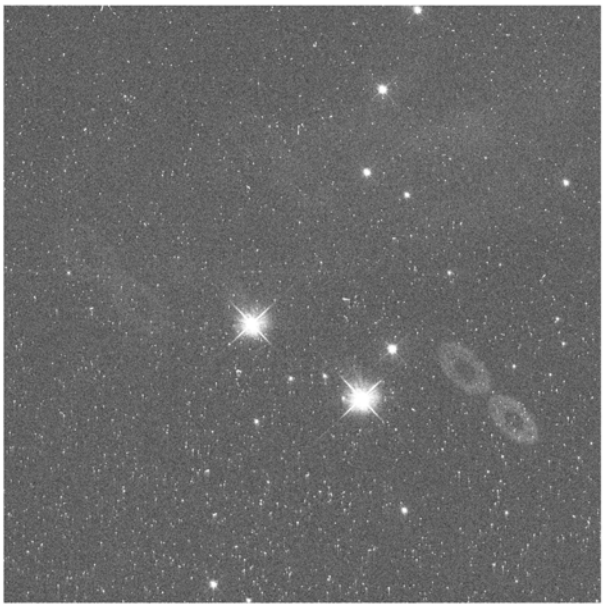


Fig. 1.—*Schematic diagram of ghosts from Figure 4 of McCullough (2011). Starting from the top left quadrant and ordering clockwise, we have the A, B, D, and C UVIS quadrants. This figure is a co-added image of 37 individual frames obtained during ground testing in TV3. Each star in the lower right quadrant produces four elliptically-shaped ghosts to the upper left. To aid in identification of the four ghosts associated with each star, we have color-coded six stars and their associated ghosts. Note that of the 36 over-exposed stars in the lower right quadrant, only the one nearest the lower right corner (red) produces four ghosts that land entirely on the CCD. Some ghosts manifest only as a singular elliptical ring (top left), and we are concerned with identifying those anomalies as well.*



(a) *Unspecified target, with two faint ghosts in quadrants A and C, taken in F606W on 04-20-2019 (Program 15212, obsID idq2ghc2q).*



(b) *Cassiopeia A, with one intermediate ghost and two extremely faint ghost (middle-left), taken in F606W and UVIS1-2K2A-SUB on 09-27-2012 (Program 12577, obsID ibqo01cuq).*

Fig. 2.—*Examples of observations with figure-8 ghosts. The ghosts are roughly the same size in pixel space, though appear larger in b) due to the magnification of the image for this figure.*

These ghosts can even appear in subarray observations where the parent source does not land in the subarray. For example, a UVIS1-2K2A-SUB observation, which only reads out UVIS quadrant A, can still contain a figure-8 ghost if a source projects onto the area of the focal plane covered by quadrant D. This observation will contain the anomaly without the reflecting source. An example is depicted in Figure 2b.

2.2. Flagging Methods

Currently, figure-8 ghosts are identified by eye by WFC3 Quicklook team members (Gosmeyer et al. 2017). Using the Quicklook database, they sort through new observations every day and flag them with the appropriate anomalies if they appear. Although human classification is well refined, there are some flaws and shortcomings without automation. First, fatigue from flagging is possible. After looking through a few hundred images in a week, a human's sensitivity may change, resulting in inconsistent classification accuracy over

time. Automation maintains consistency when flagging vast quantities of images. Second, faint and off-frame figure-8 ghosts are harder to identify. Third, random human error is rare, but possible. A figure-8 ghost can be incorrectly flagged as a different anomaly or not be flagged at all. Lastly, and most importantly for larger data rate missions, this method scales poorly. While the data rate from WFC3 is small enough that human flagging is a feasible approach, if either the number of daily images or the image size increased by a few orders of magnitudes, this method would become if not impossible. We then must look to machine learning as a solution for the described issues.

An analytical solution could be implemented. Because the mechanism creating figure-8 ghosts is understood and simple, i.e., optical reflections from the silicon surface of the CCD and the tilted glass window covering the CCD, the ghosts could be predicted *a priori*. For any given HST pointing and orientation, stars sufficiently bright[1] and in a specific region of the UVIS focal plane will produce figure-8 ghosts detectable with a predictable surface brightness, shape, location and orientation. Figure 1 shows a digital overlay intended to be used with Aladin within APT to assist observers in this prediction, in a manner equivalent to a transparent plastic overlay used with a hardcopy photographic print.

One could use a similar approach to digitally model the detailed appearance of specific figure-8 ghosts in any given UVIS image. Such modeling would be a step beyond the simple overlay in that the digital model could precisely superpose each figure-8 ghost with appropriate location, size, and brightness. We would start with a specific region of the UVIS focal plane known from geometry to produce figure-8 ghosts, i.e., most of quadrant D and some of the silicon outside the edges of quadrant D's active pixels. After converting that region to sky coordinates, a database query would extract all stars within the region that are brighter than a given magnitude in an appropriate bandpass. The magnitude limit could be set empirically from inspection of a few figure-8 ghosts and their associated stars from a few selected WFC3 images. Then for each star that met the simplified selection criteria to be pulled from the database, a model figure-8 would be predicted and assessed if it would be detectable in the given image. To verify in the actual image if the predicted figure-8 ghost was in fact detected, a matched-filter approach could be used in which we form the digital 2-D cross correlation between the actual image and the modeled figure-8 ghost. If there was a statistically significant peak in the cross correlation where expected, then a figure-8 ghost, where one was anticipated, would be detected.

[1]Table 2 of McCullough (2011) lists V magnitude limits for stars that would produce marginally detectable figure-8 ghosts. Stars fainter than V=17 mag are not expected to produce detectable figure-8 ghosts.

There are several complications to this "simple" analytic solution:

- We will need to tabulate or formulate the effects of the filters. Choosing an appropriate bandpass for the stellar catalog will be difficult since the filters in WFC3 do not have a 1-to-1 correspondence with cataloged filters.
- The model for the surface brightness of the figure-8 ghost in the CCD image is wavelength dependent: it depends on the filter transmission curves, the transmission of the glass window, the reflection coefficients for the various surfaces, and the CCD's quantum efficiency. These parameters are not known to a sufficient accuracy and may be challenging to calculate.
- Crowded star fields and extended nebulosity increase the complexity of generalized analytical predictions for figure-8 ghosts.
- Computing the cross correlation near the edges of the CCDs where the imagery is incomplete, i.e., where a figure-8 ghost is cut off, may be difficult.

Generally speaking, the "simple" analytic approach is complicated to develop. Some of these problems can be resolved using standard image processing techniques, such as zero-padding or image re-normalization. Also, machine learning mitigates these challenges because the model empirically learns everything necessary to classify the ghosts purely through a data-driven approach, rather than relying on precise optics. Therefore, the ML-based approach may be easier to build, evaluate, manage, and expand upon than the analytical solution.

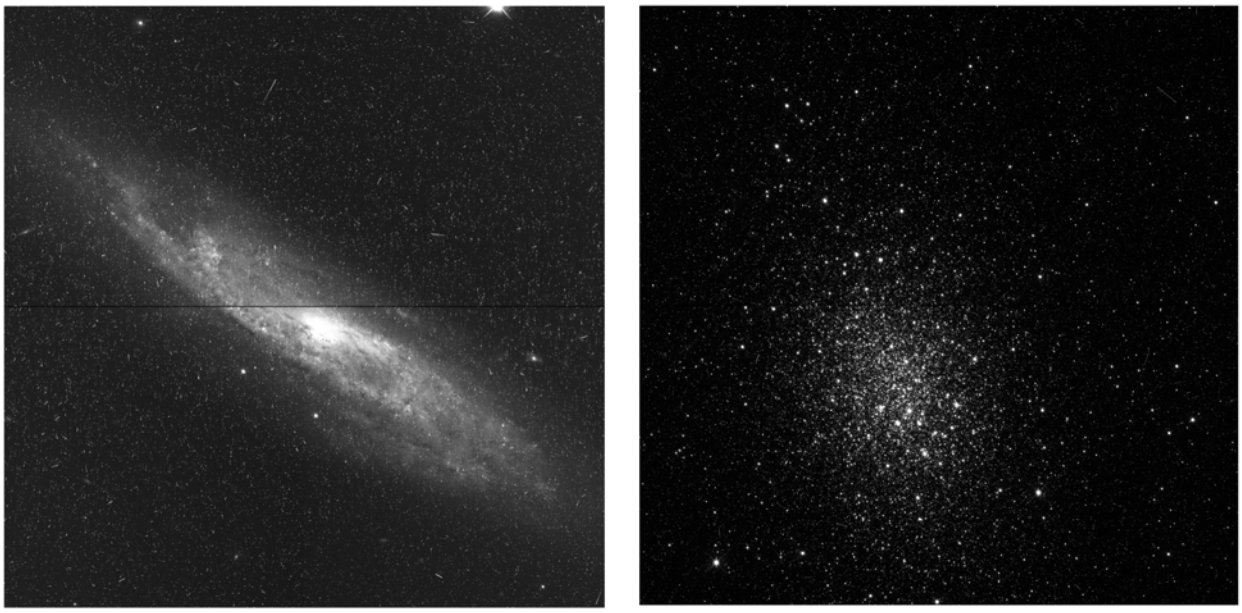
3. Data

3.1. Data Selection

To build our data set for training and testing of the models, we require both null observations (images not containing figure-8 ghosts) and anomalous observations (images containing figure-8 ghosts). The Quicklook team had already inspected all WFC3 images, identified those with figure-8 ghosts, and stored meta-data with appropriate labels.

For our null observations, we chose UVIS general observer (GO) images from 01-01-2018 to 01-01-2020 with no anomalies as verified by the Quicklook team for a total of 11714 calibrated (flt) images. GO observations also had a diverse image distribution and included images of point sources and extended sources with a variety of spatial distributions and

intensities. We chose to exclude calibration proposal images because they tend to be homogeneous, i.e., hundreds of observations of essentially the same object. We did not want the data to be skewed towards any distinct types of null images, allowing our models to focus on the anomaly instead. Figure 3 shows some examples of our null observations.



(a) *NGC2770, a resolved nearby spiral galaxy, taken in F814W on 01-23-2018 (Program 15166, obsID idi105hyq).*

(b) *NGC1978, a globular cluster in the Large Magellanic Cloud, taken in F814W on 09-14-2019 (Program 15630, obsID idxz14juq).*

Fig. 3.—*Examples of null observations, or observations without figure-8 ghosts.*

For our anomalous observations, we chose all images flagged with figure-8 ghosts from our database, which contained 5924 observations dating back to 2009. These images included figure-8 ghost observations with other anomalies so the presence of other anomalies did not affect our data selection. There were likely a few false positives, or images incorrectly flagged as having a figure-8 ghost, but we were confident in the purity of anomalous observations.

3.2. Data Processing

Since our observations varied in size (from 512x512 pixels in subarrays to 4096x4096 pixels in full frames) and pixel value distributions, we processed our data (see Figure 4) to a simpler and more uniform state by performing the following steps:

1. If the image was a full frame (used both chips), we removed the six bad rows from the center of the image (Mack et al. 2016). Since figure-8 ghosts are mostly in full frames, we wanted to remove those bad rows from our data to appear more similar to subarray observations.

2. We set all the pixel values that were less than 1 to 1. UVIS pixel units are in electrons so negative pixel values, which are a result of over-subtraction in calibration, are not physical. In addition, we chose 1 as a cutoff to ensure $\log_{10}$ scaling in the next step.

3. We scaled the data logarithmically, resulting in a minimum pixel value of 0.

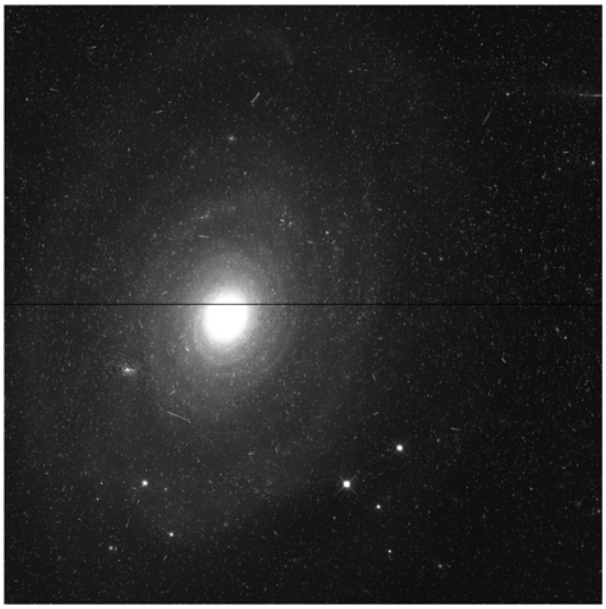
4. We clipped pixels that were above the 99.9 percentile to a maximum value.

5. We normalized the pixels to a standard normal distribution $N(\mu = 0, \sigma = 1)$, which is a common practice in machine learning, using:

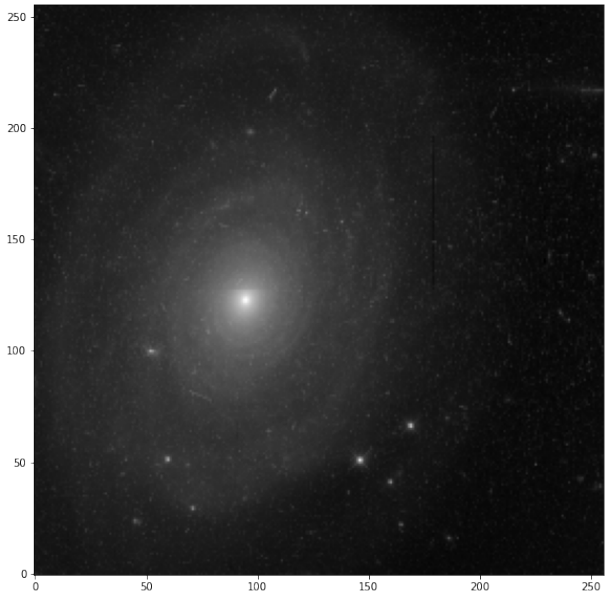
$$z = \frac{x - \mu}{\sigma}$$

   where z is the scaled pixel, x is the original pixel, μ is the image mean, and σ is the image standard deviation.

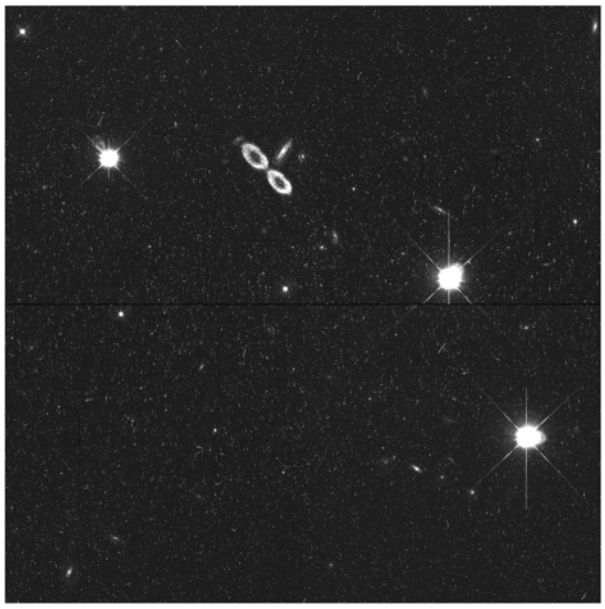
6. We rebinned the image, whether 512x512 or 4096x4096, to 256x256 pixels. By rebinning, we ensured all the prominent objects, such as the figure-8 ghosts, still had structure in the image. With smaller images, we can train significantly faster than using the original larger images. In addition, we can build deeper models without having the already long training time large images require. Furthermore, most state-of-the-art computer vision algorithms, such as VGG-19 and ResNet-18, were built using 224x224 images, which are a similar size (Simonyan & Zisserman 2014; He 2016).



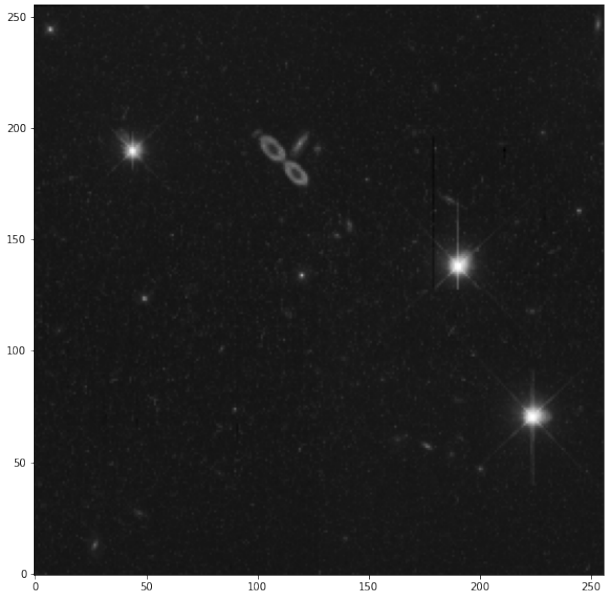
(a) *N0691 taken in F814W on 01-10-2018 (Program 15145, obsID idgg07r1q).*



(b) *idgg07r1q after completing the data processing pipeline.*



(c) *SDSSJ0742+3341 taken in F606W on 10-23-2019 (Program 15923, obsID ie5h04kaq).*



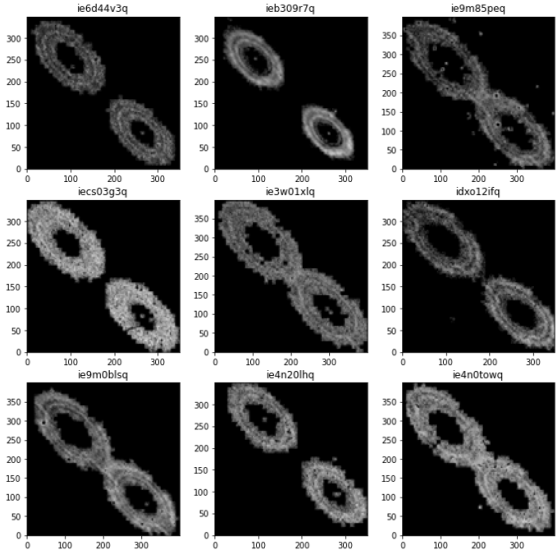
(d) *ie5h04kaq after completing the data processing pipeline.*

Fig. 4.—*Samples before (left) and after (right) completing the data processing pipeline. Null and anomalous samples are on the top and bottom rows, respectively. These examples decrease resolution from 4096x4096 (≈16M) pixels to 256x256 (≈65K) pixels, which is a factor of 1/256.*

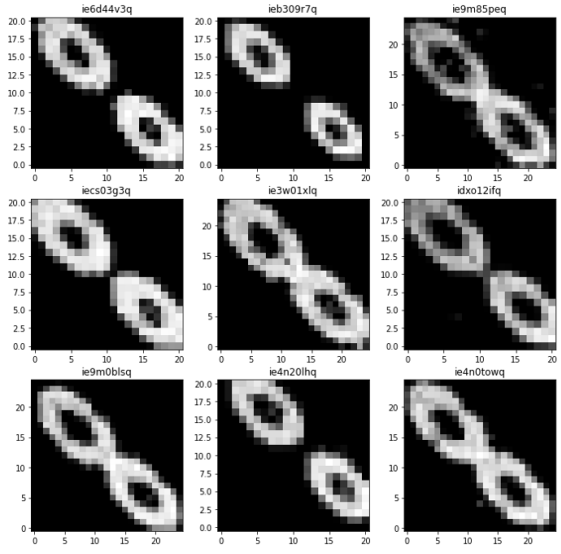
3.3. Synthetic Data

As mentioned in Section 2.1, figure-8 ghosts occur when the detector reflects light from a bright source in UVIS quadrant D. Since our figure-8 ghost observations were skewed toward full frame observations, a majority of them had bright objects in them. We did not want our model to solely depend on bright objects in quadrant D to predict if a ghost was present: we wanted it to identify the figure-8 ghosts themselves, *regardless* of if the image had a parent source or not. We created synthetic data to break that dependency. We extracted 25 diverse figure-8 ghosts from UVIS GO images to superimpose onto null images. These new synthetic images did not necessarily have bright objects in them so the model should learn the figure-8 ghost itself and not the characteristics for one to be likely to appear. By creating synthetic images, we augment the actual observations in our data set. We extracted the figure-8 ghosts as follows (see Figure 5 for examples):

1. Enclose a figure-8 ghost by a 300x300 to a 500x500 pixel box.
2. Min and max clip pixels within the box to 0, i.e., pixels outside a predetermined range are set to 0. This left a high signal of figure-8 ghost pixels and removed the background, astronomical objects contained in the box, cosmic rays, etc. In addition, set the remaining background pixels within the ghost pixel range to 0.
3. Perform steps 2-5 from Section 3.2 in the same order.
4. Decrease resolution by 1/16 (the same resolution figure-8 ghosts in a 4096x4096 image would be if down sampled to a 256x256 image), making a figure-8 ghost contained to 20x20 to 30x30 box.
5. Min-max scale the figure-8 ghosts to have a minimum pixel value of 0 and a maximum pixel value of 1.



(a) *Some figure-8 ghosts after completing step 2 of the extraction pipeline.*



(b) *The figure-8 ghosts from a) after fully completing the extraction pipeline.*

Fig. 5.—*Samples before (left) and after (right) completing the extraction pipeline. The HST observation IDs from which the ghosts were extracted are the subplot titles. These ghosts were decreased in resolution in order to appropriately fit our 256x256 images. Although they lost some of the internal ring structure, the overall shapes were conserved.*

We created our synthetic images as follows (see Figure 6 for examples):

1. Randomly choose a number between 1-3 of figure-8 ghosts to superimpose.
2. Scale the cutout by the image mean and multiply it by a random intensity from 1-5.
3. Superimpose the cutout onto the processed null image at a random location. They were not superimposed according to an optical model, so in particular, the size of a synthetic figure-8 was not correlated with its position in an image, as they would be for authentic full frame images.

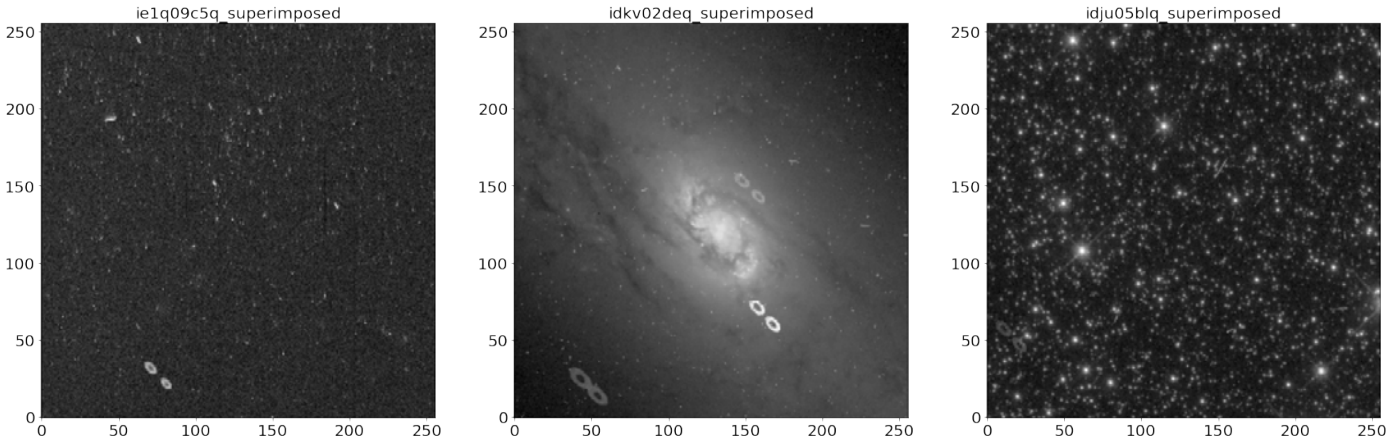


Fig. 6.—*Three examples of synthetic figure-8 ghost observations. The HST observation IDs of the original observations are the subplot titles. From left to right, the first observation has one ghost, the second has three, and the third has one (faint, at bottom left).*

3.4. Training, Validation, and Test Sets

We created two collections of training, validation, and test sets: one using synthetic data and one using real data. Validation sets are used to ensure the models generalize after training, and test sets are used to determine the models overall performance. Our goal was to see if synthetic data can provide adequate training data sets when substituted for real data, and if not, then to train directly on real data.

Synthetic Data: For our synthetic sets, the 11714 null observations were randomly split into groups of 5000, 5000, 857, and 857, with each observation being uniquely used once. To construct the synthetic training set, we used one group of 5000 as *null classifications* (0) and superimposed figure-8 ghosts on the other group of 5000 as *anomaly classifications* (1) for a total of 10000 training samples. Similarly, for the synthetic validation set we used one group of 857 as null classifications and superimposed figure-8 ghosts on the other group of 857 as anomaly classifications for a total of 1714 validation samples. The 5924 real figure-8 ghost samples were the test set for the model that was trained and validated on synthetic data. We tested on real figure-8 ghosts to determine if a model trained on synthetic data can generalize to real data.

Real Data: To construct the real training set, we used a group of 4000 unique null observations as null classifications and the first 4000 observations with figure-8 ghosts as anomaly classifications for a total of 8000 training samples. Similarly for the real validation

set, we used a group of 1000 unique null observations as null classifications and the next 1000 observations with figure-8 ghosts as anomaly classifications for a total of 2000 validation samples. The test set for the model that was trained and validated on real data consisted of the remaining 924 real observations with figure-8 ghosts.

Any additional changes to these sets for training/validating/testing the different models are explicitly stated in their respective architecture subsections in Section 4.

4. Model Training Procedures

We trained five different convolutional neural networks (CNNs) to classify our images. CNNs are a subset of machine learning models that excel at computer vision, well-suited for this work. Figure 7 shows LeNet, a popular CNN architecture, to illustrate the workflow of these models (LeCun et al. 1998). Each models' hyperparameters (controllable parameters specific to the model for training) and architectures are listed in Appendix A. All models were trained in shuffled batches using cross entropy loss and the Adam optimizer using 8 CPUs (Kingma & Ba 2014). These networks were built using the Python library PyTorch (Paszke et al. 2019). The model trained on purely synthetic data with no modifications will be referred to as the "synthetic model" and the other four models will be referred to as models A-D. Our code is located at `https://github.com/spacetelescope/deepwfc3`.

The following subsections provide details about further data manipulation for each model. Dauphin et al. (2021) includes a glossary of ML-related vocabulary that may be useful for those who are unfamiliar with such terms. The models are detailed in the remainder of this section. Table 1 summarizes the data sets used to train, validate, and test our five CNNs.

4.1. Synthetic Model Training

First, to demonstrate that a CNN can classify figure-8 ghosts on an image, we trained a model on the synthetic data as described in Section 3.4. The synthetic model used the LeNet architecture (2 convolutional layers and 3 fully connected layers), but with different hyperparameters (see Table A1). The synthetic model trained for 10 epochs, or trained over the entire training set 10 times. It also trained with a batch size of 128, or trained using 128 samples per training iteration.

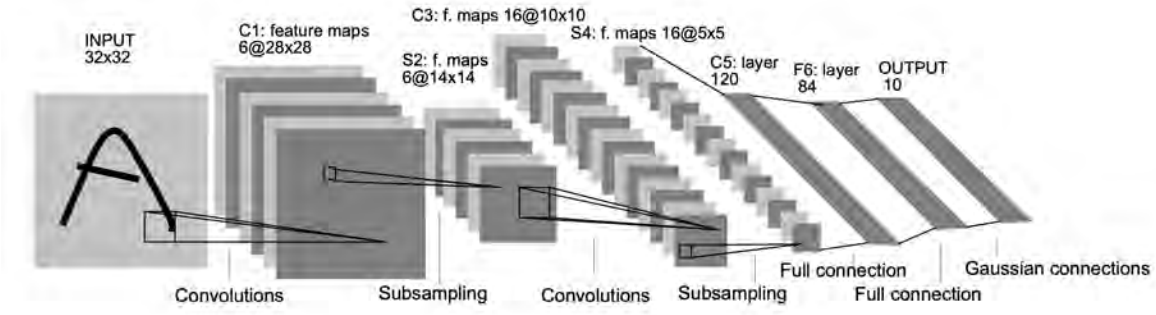


Fig. 7.—*LeNet Convolutional Neural Network Architecture (LeCun et al. 1998). CNNs use a series of convolutional filters to extract useful features from images. The workflow is to convolve the inputs into feature maps and subsample the feature maps by max pooling. After the last subsampling, the features feed through fully connected layers of a neural network. At the last layer, the maximum output from a neuron determines the classification of the original input. As batches of data are fed into the CNN, the cross entropy loss is computed based on the discrepancy of the model predictions and ground truths, and the CNN parameters are updated so as to minimize the loss.*

4.2. Model A Training

Model A used transfer learning to classify figure-8 ghost images. Transfer learning is a method of machine learning in which a pretrained model is applied to solve a problem outside of its original intent (Zhuang et al. 2020). In theory, a deep CNN trained on millions of diverse images will develop filters useful for a wide variety of computer vision tasks. Instead of training a model from scratch, one could "freeze" the convolutional layers of the pretrained model and retrain the last fully connected layer, saving compute time. By freezing all the convolutional layers and retraining the fully connected layers, the pretrained convolutional layers acts as a feature extractor and the retrained fully connected layers acts as a small neural network. In addition, transfer learning is an alternative solution for relatively small data sets. Since we only analyzed 5924 true figure-8 ghost images, this method was well-suited to this data set's size.

Model A used the GoogLeNet architecture as a feature extractor (Szegedy et al. 2015). See Table 1 and Figure 3 in Szegedy et al. (2015) for full details on the architecture. We chose GoogLeNet because it was the smallest model (7M parameters) with relatively good results compared to other pretrained models. The small size positioned us for faster training and less computing. In addition, we modified the original architecture by appending three fully connected layers (see Table A2).

Model	**Training Samples** [null, synthetic, real]	**Validation Samples** [null, synthetic, real]	**Testing Samples** [null, synthetic, real]
Synthetic	[5000, 5000, 0]	[857, 857, 0]	[0, 0, 5924]
A	[4000, 0, 4000]	[1000, 0, 1000]	[0, 0, 924]
B	[5000, 2500, 2500]	[857, 428, 429]	[0, 0, 2995]
C (pretrain)	[4000, 4000, 0]	[1000, 1000, 0]	N/A
C (transfer)	[4000, 0, 4000]	[1000, 0, 1000]	[0, 0, 924]
D (pretrain)	[5000, 5000, 0]	[857, 857, 0]	N/A
D (transfer)	[5000, 5000, 4000]	[857, 857, 0]	[0, 0, 1924]

Table 1: *Training, validation and test sets summary. Each column lists the null, synthetic, and real samples used for training, validation, and testing. Each row lists the number of various samples used for each model. Model C and D are split into pretrain and transfer for readability. Since pretrained Model C and D were not tested, those values are left as not applicable (N/A). The test sets only contained real figure-8 ghost observations since those were the images we were interested in classifying.*

Model A was pretrained on ImageNet, a data set containing over a million images of a thousand different classifications (Russakovsky et al. 2015). The model was retrained using the real training, validation, and test sets described in Section 3.4. The data sets were further processed to match ImageNet statistics as follows:

1. Min-max scale so that the pixel values range from 0 to 1.
2. Concatenate three copies together to make a "RGB" image.
3. Center crop/trim to 224x224 pixels. Some images with figure-8 ghosts cut off at the edge may be cropped off. We were confident that most ghost samples were not near the edge so any effect this may have on training is negligible.
4. Normalize the RGB channels to the distribution the model was pretrained with:
   $N(\mu = [0.485, 0.456, 0.406], \sigma = [0.229, 0.224, 0.225])$

After those transformations were performed, our images were in "ImageNet format", and suitable for GoogLeNet training. Model A trained for 10 epochs with a batch size of 128.

4.3. Model B Training

Model B used a seven-layer network architecture, with three convolutional layers and four fully connected layers (see Table A3). The training procedure for this model used a particular approach. We mixed both synthetic and real data in order to force the model to recognize the figure-8 ghosts in synthetic images while still being able to generalize to real data. To do so, this model used the synthetic data sets described in Section 3.4, but replaced half (2500 for training, 429 for validation) of the synthetic figure-8 ghost images with the first 2929 real figure-8 ghost observations. The test set was the remaining real figure-8 ghost observations (2995) that were not used in training or validation. The model trained for 10 epochs and the batch size chosen was 32.

4.4. Model C Training

Model C used a six-layer network, consisting of three convolutional layers followed by three fully connected layers (see Table A4). This model used a combination of conventional CNN training as well as transfer learning to classify figure-8 images. The model was first trained on the synthetic data until high accuracy was obtained on the synthetic validation set. Then the parameters for the convolutional layers were frozen, and the model was retrained on a set of data containing real figure-8 ghosts. This should allow the network to optimize the fully connected layers to detect the larger variety of figure-8s that appear in the real data, while benefiting from the focused tuning of the convolutional layers from the synthetic data. Aside from the freezing of the convolutional layer parameters, the network architecture was not changed between the two training runs.

The training set for the synthetic data consisted of 4000 images with the figure-8 superimposed, and 4000 null images. The validation set consisted of 1000 superimposed images and 1000 null images. When retraining the fully connected layers, the training set consisted of the first 4000 images of the data set that contained real figure-8s, and 4000 null images, for a total of 8000 images. The validation set for retraining consisted of the next 1000 real figure-8 images and 1000 null observations, for a total of 2000 images. The test set was the remaining 924 figure-8 ghost images. Each training run was allowed 5 epochs, with batches of size 64.

4.5. Model D Training

Model D used a seven-layer network with three convolution layers and four fully connected layers (see Table A5). Similar to Model C, Model D initially trained and validated on synthetic data sets as described in Section 3.4, and transfer learned on a combination of the synthetic set and 4000 real figure-8 ghosts. The validation set consisted of 857 images as null classifications and 857 images as anomaly classifications for a total of 1714 retraining samples. Similar to other models, Model D was tested using the remainder of the real figure-8 ghost samples (1924). The model trained for 5 epochs each training run with a batch size of 64.

5. Results

Table 2 summarizes the results of the synthetic model and Models A-D. We present the number of trainable parameters and approximate training time. We include true negative (TN: null samples correctly predicted as null), false positive (FP: null samples incorrectly predicted as figure-8), false negative (FN: figure-8 samples incorrectly predicted as null), and true positive (TP: figure-8 samples correctly predicted as figure-8) rates on the models' respective validation sets. In addition, we present the test accuracies, the percentage of authentic figure-8 images each model detected in their respective test sets.

Model	**Number of trainable parameters**	**Training time**	**TN**	**FP**	**FN**	**TP**	**Test Accuracy**
Synthetic	4.2M	1.5H	0.98	0.02	0.01	0.99	54%
A	2.1M	1.5H	0.93	0.07	0.13	0.87	83%
B	1.07M	1H	0.99	0.01	0.12	0.88	78%
C	160K, 130K(for transfer)	1H, 15m	0.94	0.06	0.25	0.75	79%
D	2.12M, 2.10M(for transfer)	1H, 15m	0.65	0.35	0.09	0.91	62%

Table 2: *Results from training five different CNNs. The TN, FP, FN, and TP rates were determined from each model's validation set, respectively. Note the validation and test sets used for each model may differ so refer to the subsections of Section 4 for further clarification. Model A has a total of 7M parameters, but since the convolutional layers were frozen and a few fully connected layers were added, we only trained using 2.1M parameters.*

In addition, we use saliency maps as a method of interpreting our models (Simonyan et al. 2013). A saliency map calculates the gradients of the output neurons with respect to the input pixels, in contrast to backpropagation which calculates the gradients of the output neurons with respect to the model's parameters. Ideally, the map highlights pixels useful for

making a prediction. Saliency maps produced by our models on several samples provided intuition for how our models were working. All saliency map figures after Figure 8 are in Appendix B.

5.1. Synthetic Model Results

The synthetic model had an overall accuracy of 98% on its validation set, indicating it generalized for synthetic images. The true negative and true positive rates were 0.98 and 0.99, respectively. The false positive and false negative rates were 0.02 and 0.01, respectively. The false positives were caused by long cosmic rays and comet observations. Since those objects were also bright and slanted with respect to the origin, images with those features were mistaken for a figure-8 ghost. The false negatives were caused by faint or off frame figure-8 ghosts, similar to the WFC3/IR Blob Classifier (Dauphin et al. 2021). Since these synthetic figure-8 ghosts were not as "visible" to the other samples, this result aligns with previous work. In addition, some false negatives were figure-8 ghosts superimposed onto physical objects, such as planets or galaxies. The figure-8 ghosts were blending in with the objects, essentially making them disappear. However, we were able to detect them by eye.

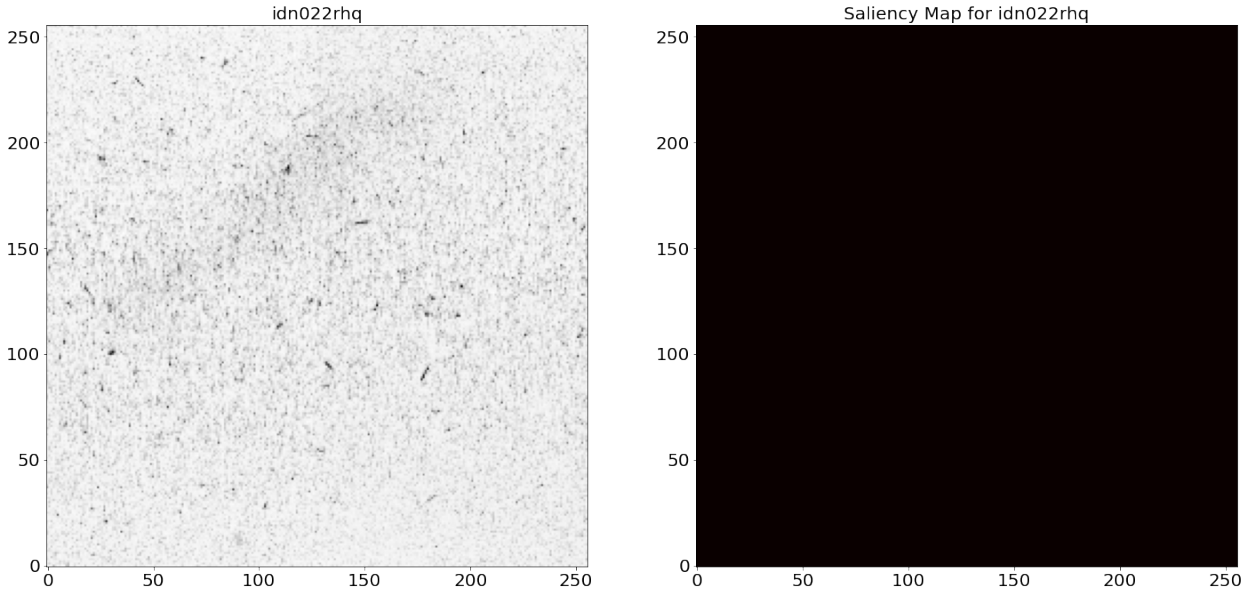
Although this model performed tremendously well on synthetic data, performance on the test set, which were real figure-8 ghost images, drastically decreased. The accuracy for the test set was only 54%, meaning the synthetic model did not generalize to real data. For the samples that the synthetic model correctly classified (TP), the figure-8 ghosts were bright and isolated, which matches expectations because those samples would be the easiest to classify. However, for the samples the synthetic model incorrectly classified (FN), the figure-8 ghosts were faint or blended in with the background of astronomical objects (galaxies, nebulae, star clusters, etc.), as it did with the synthetic data set. This result implies that the distribution of intensity of synthetic figure-8 ghosts was skewed toward bright intensities, while the true distribution was likely skewed toward fainter intensities. In addition, the GO images we chose to superimpose were not the true distribution of images figure-8 ghosts that were found in the database. Perhaps a more carefully selected training set upon which to superimpose a fainter distribution of figure-8 ghosts would better represent real observational data.

The saliency maps on synthetic data were nearly perfect. Most null saliency maps were blank, indicating the synthetic model did not focus on any pixels for classification, and the synthetic figure-8 ghost's saliency maps almost always highlighted the anomalies. Figure 8 shows two examples from its validation set. Similarly, authentic figure-8 ghosts were almost always highlighted regardless of being classified or not, which was an odd property of the

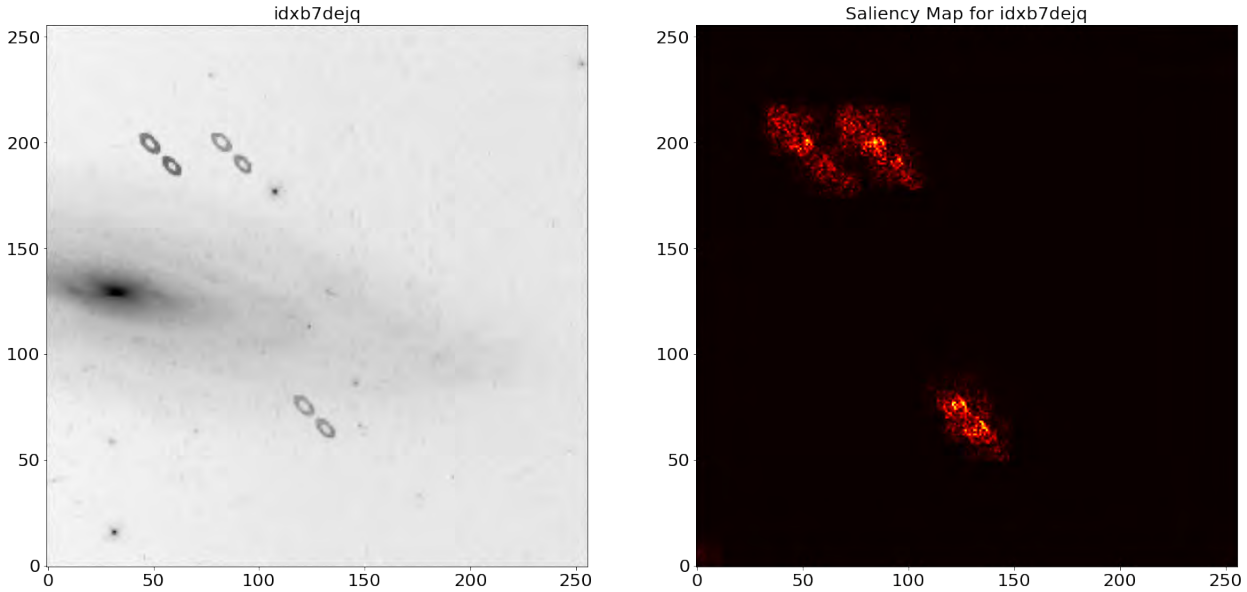
synthetic model. Despite correctly predicting only 54% of the anomalies in our test set, the saliency maps highlighted a large majority of them. The synthetic model notices the ghosts, but something in the training was preventing correct predictions. Figure B1 presents some examples from its test set. We are unsure how to interpret this curious behavior: that is, we are unable to explain the disconnect between the saliency map's apparent success and the synthetic model's failure to predict real figure-8 ghosts.

Another criticism of the synthetic model is that the figure-8 ghosts we extracted were not diverse enough and the model is overfitting on them. However if that was the case, then the synthetic model wouldn't be able to predict any figure-8 ghosts if it just memorized the select 25 extracted ghosts, and it does perform better than only recognizing a small subset of figure-8 ghosts. We look towards data debugging as first steps in the future in resolving this model's generalization.

Overall, the synthetic model learned the way we expected, with the saliency maps suggesting that it can locate figure-8 ghosts in nearly all cases. However, it was not good at its intended task, which was predicting whether a particular image contained one or more real figure-8 ghosts. This work demonstrates that training using synthetic data is challenging, but provides worthy insights.



(a) *Observation and saliency map for a TN sample from the validation set.*



(b) *Observation and saliency map for a TP sample from the validation set.*

Fig. 8.—*Observations and saliency maps produced by the synthetic model for true negative (TN) and true positive (TP) samples from the validation set. The figure-8 ghosts in b) were clearly highlighted in the maps, indicating those pixels were useful for classification.*

5.2. Model A Results

Model A was trained on the real data as described in Section 3.4. The TN, FP, FN, and TP rates on its validation set were 0.93, 0.07, 0.13, and 0.87, respectively. The overall accuracy on its test set was 83%, which is much better than the synthetic model's accuracy (54%).

Although its accuracy was better, Model A produced poor saliency maps, indicating it is using extremely complex features to predict figure-8 ghosts. The saliency maps for all of our samples (positive and negative) had a random scatter of pixels centered toward the middle, and were not focused on localized structures. Figure B2 displays a few examples. Since the convolutional layers were trained on ImageNet, saliency maps may be more appropriate for tasks suited to that training set, such as classifying humans, animals, cars, etc. ImageNet did not contain any astronomy related photos, making it difficult for Model A to combine filters to create comprehensible features in the fully connected layers. The random scatter could also be an artifact of backpropagation from the fully connected layers through so many convolutional blocks. Since there were so many pretrained convolutional blocks, any intermediate features could ripple through the saliency map, creating the random scatter of activated pixels.

Model A was more influenced by background pixels than the figure-8 ghost pixels. Although we did not understand how Model A predicts, it did predict well, outperforming the synthetic model. Trading off model comprehension for performance is a case-by-case decision, but in this case, for Model A, we find it hard to trust the results because the saliency maps did not focus on the expected anomaly pixels.

5.3. Model B Results

As previously mentioned in Section 4.3, we trained using the mixed data set to determine if Model B generalized better while picking up on the synthetic nature of the superimposed figure-8 ghosts. The accuracy achieved on its validation set was 93%, with true negative and positive rates of 0.99 and 0.88, respectively. Conversely, the false positive and negative rates were 0.01 and 0.12. On its test set, the accuracy achieved was 78%.

Figures B3 and B4 show some examples of images with their corresponding saliency maps. Other extended and diffuse structures, like the halos of the stars, were also being picked up by Model B. In spite of this, the model did a reasonable job at highlighting the pixels corresponding to figure-8 ghosts and correctly classifying the images.

False negatives seem to be due to: 1) a very faint figure-8 ghost with respect to back-

ground or other diffuse sources of the image, which were difficult even to detect by eye, 2) figure-8 ghosts that were not completely within the image, and 3) in a handful of cases the original images were mislabeled by humans as negatives. These three causes for false negatives were aligned with expectations since those samples were harder to predict or did not have the correct label. In the cases where the figure-8 ghost images were not correctly classified by the model, the saliency maps still highlight the pixels associated with the ghosts, similar to the synthetic model's performance (Section 5.1).

Overall, Model B was able to achieve a decent accuracy by making use of both synthetic data and real data effectively. One thing that could help to improve figure-8 ghost classification in Model B is increasing the synthetic sample with fainter figure-8 ghosts so the model could learn to identify them and generalize better to real data, as previously mentioned in Section 5.1.

5.4. Model C Results

While Model C was able to achieve high accuracy on the synthetic data (98%), the performance when retrained and tested on real figure-8 images was significantly worse, with an overall accuracy of 79% on the test set. The true positive rate on the validation set was slightly worse, at approximately 0.75, though the true negative rate remained relatively high at 0.94. In most cases, the saliency maps showed the retrained network highlighting the pixels where the ghost appeared, even in the false negative images. This phenomena indicated that the network still identified figure-8 ghosts, but is unable to reliably predict if the detected feature is truly a ghost or not (see Figure B5a). This was likely due to the network also detecting extended features such as PSF halos as relevant objects (see Figure B5b). A more balanced training set, containing more of these types of objects in images *without* figure-8 ghosts, could help the network better discriminate between the anomaly and other extended features, yielding fewer false classifications.

5.5. Model D Results

Model D's initial training on the synthetic data resulted in an accuracy of 98%. However after transfer learning, the accuracy dropped to 62% on its test set. The TN, FP, FN, and TP rates on its validation set were 0.65, 0.35, 0.09, and 0.91, respectively. Due to the lower accuracy on the test set, it was difficult to determine any patterns from false positive or false negative samples. The saliency maps produced by Model D poorly highlighted figure-8 ghosts compared to some other models (see Figure B6). Even though Model D performed more

poorly than models A-C, it still outperformed the synthetic model, indicating the importance of real data when training. Overall, Model D performed satisfactorily in evaluation metrics and under-performed in producing interpretable saliency maps.

6. Discussion

Machine learning is a rapidly evolving field with new discoveries and insights taking place daily. It takes time and practice to become proficient. This work is a step in the right direction for utilizing machine learning in astronomical data, and we hope others will follow. The skills we have gained will be improved and applied to solve more difficult problems within STScI and astronomy as a whole. We emphasize the best way to learn is a simple, but true, cliché saying: practice makes perfect. We urge those interested in getting started to take that first step, whether it be asking a colleague how to learn, surfing the internet for the best resources, or practicing through small projects of interest.

6.1. Investigating Autoencoders as Anomaly Detectors

An autoencoder is an unsupervised machine learning technique in which the model reproduces the input as the output while compressing the input into a low dimensional representation (Bank et al. 2020). An autoencoder is comprised of an encoder and decoder. The encoder compresses the input to a "latent space", where the dimensionality of the latent space is far less than the input space. The decoder reconstructs the input from the latent space. Figure 9 illustrates a simple example of an autoencoder. If the model is well trained, the input and reconstruction should be nearly identical, and the latent space should be a sufficient low dimensional representation of the input space.

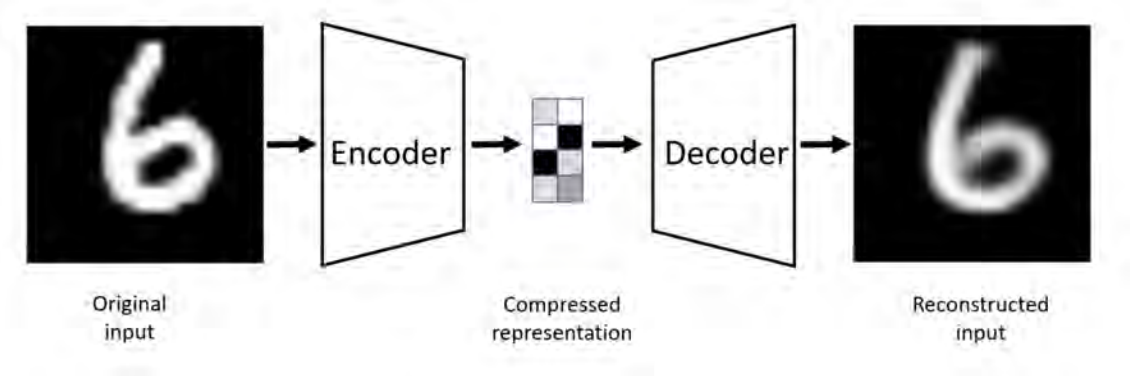


Fig. 9.—*Autoencoder example (Bank et al. 2020). The input image is compressed into a lower dimensional space by the encoder. The lower dimensional representation decompresses to the original input space by the decoder.*

Autoencoders can be used to detect anomalies via reconstruction loss. Suppose you trained an autoencoder to learn the representation of images of dogs. If an image of a bus was evaluated by the autoencoder, ideally the reconstruction would be "dog-like" because those were the features on which the model was trained. The loss between the input and reconstruction should be high and the anomaly can be automatically detected using a predetermined threshold.

As a preliminary investigation, we trained an autoencoder with a 256-dimensional latent space on the null UVIS GO images from our training set. Since figure-8 ghosts were not in these images, the model should have poor reconstructions on the anomalies. We trained until our validation set had a coefficient of determination, R^2, of 0.85 indicating a high correlation between the inputs and reconstructions. When evaluating the test set, the autoencoder was able to reconstruct the figure-8 ghost, without having any prior knowledge of the anomaly (see Figure 10). Since our data set had diverse astronomical features and the figure-8 ghost is a relatively simple anomaly, we deduced that our autoencoder's learned filters were sufficient to reconstruct features outside of its original domain. Because the autoencoder was able to reconstruct the ghosts, taking the loss between the input and the reconstruction would not be a reliable way to detect anomalies. From this result, we suggest using autoencoders for anomalies when the anomaly *is the whole image*, not when it *is a fraction of the image*. We are interested in further investigation and research about anomaly detection in astronomical images using autoencoders.

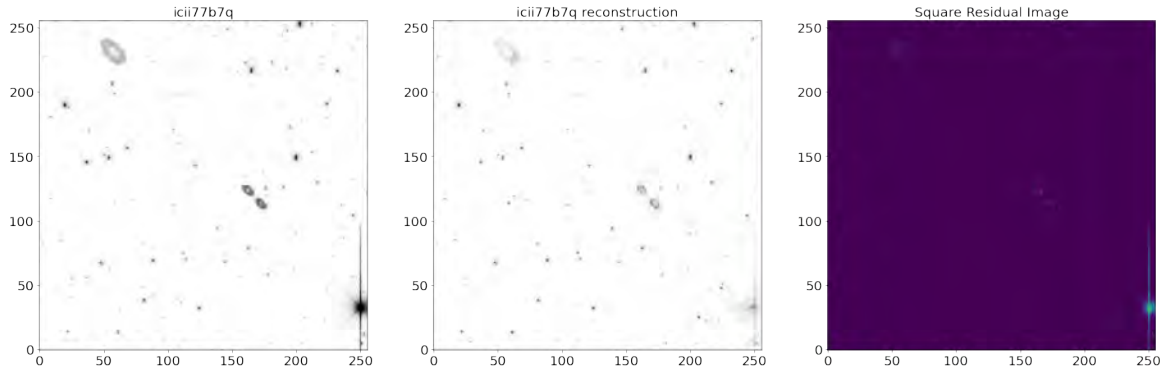


Fig. 10.—*Autoencoder anomaly reconstruction. From left to right is the original image, reconstructed image, and square residual of the images. The autoencoder reconstructs the figure-8 ghost pretty well. Although some loss between the original image and reconstructed image is due to reconstructing the ghost, most is due to reconstructing the parent source.*

6.2. Looking Inward

Solving "data problems": The power of machine learning comes from the data; clean, abundant data produce the best generalized models, regardless of which algorithm is used. Astronomical data is notoriously complicated so we did as much as we could to prepare them for our CNNs. We chose to train only using UVIS GO images from a two-year period to save time on modeling with CPUs, but perhaps our models would have increased performance using more images from the entire WFC3 database. In addition, we could have augmented our data, but we were hesitant about reusing too many images. In machine learning research, random cropping is a popular form of augmentation, but we were worried some of our figure-8 ghosts would be cropped out of the images and our model would train on a tainted data set. A deeper look into more complex augmentation techniques may make our sparse data set more useful.

Future trials of interest: Machine learning is a domain in which a problem has an infinite amount of viable solutions and it is difficult enough to just find one. There's always more we can do to improve results, but here are several suggestions we are interested in following up on for future work:

- While we selected 256x256 as the image training resolution to mitigate information loss, lower resolution images, such as 64x64 or 128x128, would increase training speed. However, we don't know the magnitude of impact that different amounts of information loss might have on model performance.

- Deeper and wider networks generally increase performance, but the time required to train these larger networks on CPUs versus the possible marginal increase in performance was not sufficient for this work. Training on GPUs and high performance computing clusters would allow this.

- We could further investigate optimization, such as tuning hyperparameters (batch size, epochs, etc.), rebalancing classes, and using focal loss as a loss function (Lin et al. 2017).

- Training on only full frames (where most figure-8 ghosts appear) would make the data simpler and could be a necessary intermediate step before generalizing to all UVIS GO images.

- To our knowledge, the UVIS detector's properties have not changed enough to have a major effect on GO images over its lifetime. This reason is why we were not concerned with using two years of GO images to train as our null classifications. In addition, since we superimposed figure-8 ghosts, we were not concerned about the models having a bias towards subarray observations to classify null observations. However, it may be worthwhile to determine whether our metrics are stable as a function of observation date or original image size.

- Instead of log scaling, we could use linear scale and a similar clipping approach as in the WFC3/IR Blob Classifier (Dauphin et al. 2021). Since the figure-8 ghosts have high intensity relative to the background, most of those pixels will likely be max clipped and appear brighter post processing. The higher contrast between the figure-8 ghost and the background, in combination with the loss of internal structure, would make the anomaly "simpler" and possibly easier for a CNN to identify.

- We could explore the impact of altering the order of the superimposing routine to see if there is a non-negligible difference in superimposing figure-8 ghosts directly onto the preprocessed images versus superimposing them post-processing.

- Object detection, such as using the You Only Look Once (YOLO) algorithm, is also of immediate interest for future work, as this method predicts where the objects of interest are in the image by using a bounding box (Redmon 2016). This step is natural after achieving high performance in image classification.

- We are interested in using other ML algorithms outside of CNNs, such as unsupervised clustering techniques or other supervised classifiers. Uniform Manifold Approximation and Projection (UMAP) attempts to preserve a data set's local and global structure in a lower dimensional space (McInnes et al. 2018). The images in this lower dimensional

space can be used as features. Support Vector Machines (SVMs) is a classifier which finds a hyperplane to maximize the distance of data with different classes (Cortes & Vapnik 1995). We believe these two algorithms in unison could lead to promising results.

6.3. Looking Outward

One of astronomy's unique challenges in computer vision, especially within the next decade, is dealing with large images. The most common computer vision algorithms in the literature are built using relatively small images, usually at the size of 224x224 (50K) pixels (Simonyan & Zisserman 2014; Szegedy et al. 2015; He 2016). In addition, the algorithms are classifying/detecting major objects in the image, i.e., objects that contain a noticeable portion of pixels. Astronomical images are comparably massive, and only getting larger. WFC3/UVIS full frames are 4096x4096 (16M) pixels, the images from the Wide-Field Instrument aboard the Roman Space Telescope will be 300M pixels, and the images from the Vera Rubin Observatory will be 3.2B pixels. Astronomical objects in these images will be minuscule, taking up only the tiniest fraction of pixels compared to the full image. Modern models using small images with big objects is quite the opposite of what astronomy will need: models using big images with small objects. Standard deep CNN architectures using dozens of convolutional blocks and fully connected layers could easily be hundreds of billions of trainable parameters if the full image is used as a singular input, tremendously scaling up the number of operations performed. This model size is thousands of times larger than VGG (138M parameters), one of the most popular CNN architectures, and is comparable to GPT-3, one of the largest machine learning models ever built (Simonyan & Zisserman 2014; Brown et al. 2020). The velocity and volume of data from future missions makes this approach impractical at this moment in time because it uses an exhaustive amount of computation for machine learning, even for high-performance clusters. There are two immediate solutions: downsampling and cropping.

Downsampling has the benefit of using the whole image for modeling. However, the arguably larger cost is lower resolution and information loss. In addition, smaller or fainter objects may be completely lost due to the degree of downsampling. Some examples include anomalies like cross talk blending into the background or a point source only containing a handful pixels after downsampling. We chose downsampling in this work specifically because the figure-8 ghosts were still relatively large and bright after it, and the high-resolution details did not matter for our purpose. However, in other work, downsampling might be detrimental to results (Hausen & Robertson 2020).

Cropping a part of an image is an alternative solution. Although this method preserves resolution and information, a cost is labeling the cutouts, which can be a major setback in supervised machine learning. By systematically cropping out rectangles of the whole image, e.g., dividing a 4096x4096 image to 256 256x256 cutouts, there is no way of telling what the cutout is. A human can manually label the respective cutouts, but this method does not scale well and is extremely time consuming. A machine learning model could manually label new data, but items could be mislabeled if an object is cropped. Also, the model would need to have an exceptional training set prior to labeling new data, which is often difficult.

The authors support machine learning as a solution for these essential tasks in astronomy due to the ever-growing depth of the field and countless successes in astrophysics research, but applying ML to astronomy has some obstacles in the near future. We believe one of the astronomy community's main points of focus for the next decade should be tackling these types of data management and workflow problems to ensure the advancement of research in the era of big data.

7. Conclusions

Anomaly detection in astronomical images is a necessary task for supporting the WFC3 user community. Machine learning provides a practical solution for anomaly detection when traditional methods cannot be performed due to image size, data volume, etc. We present five different convolutional neural networks (CNNs) to identify figure-8 ghosts in WFC3/UVIS images.

- The synthetic model (2 convolutional layers, 3 fully connected layers; 54% accuracy) learned the anomaly in an interpretable way, but failed to perform well on real figure-8 ghosts. Although the synthetic model did not provide reliable results for observational data sets, we encourage to model using synthetic data because it builds strong intuition for solving a similar problem with real data.
- Model A (GoogLeNet convolutional layers, 3 fully connected layers; 83% accuracy) performed well on real figure-8 ghosts, but used the entire set of image features to do so.
- Model B (3 convolutional layers, 4 fully connected layers; 78% accuracy) utilized both synthetic and real data to learn interpretable features.
- Model C (3 convolutional layers, 3 fully connected layers; 79% accuracy) trained well on synthetic data and was able to transfer that knowledge accordingly to real data.

- Model D (3 convolutional layers, 4 fully connected layers; 62% accuracy) performed weaker than other models, but still better than the synthetic model's baseline.
- Saliency maps are a helpful tool for understanding predictions in computer vision, but should not be the deciding factor for model deployment since each model is handled case-by-case.

We advise to always use real data for training models if available. However, we find significant benefit in incorporating synthetic data, either through pretraining or data set mixing when real data is sparse. In addition, we considered several additional steps to improve upon the quality of this work. Finally, the authors urge the astronomy community to prioritize and strategize on high-resolution (>16M pixels) computer vision to fully prepare for future missions.

Acknowledgements

We thank Annalisa Calamida, Joel Green, Mariarosa Marinelli, and John Wu for their exceptional revisions of this report. We also thank Lou Strolger for help with setting up our server to complete our data processing at a rapid pace. In addition, we thank Michelle Ntampaka for her insightful conversations and suggestions in completing this work.

References

Bank, D., Koenigstein, N. and Giryes, R., 2020. Autoencoders. arXiv preprint arXiv:2003.05991.

Brown, T.B., Mann, B., Ryder, N., et al., 2020. Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

Cortes, C. and Vapnik, V., 1995. Support-vector networks. Machine learning, 20(3), pp.273-297.

Dauphin, F., Medina, J.V., and McCullough, P.R., 2021. WFC3 IR Blob Classification with Machine Learning. WFC3-ISR 2021-08.

Dieleman, S., Willett, K.W. and Dambre, J., 2015. Rotation-invariant convolutional neural networks for galaxy morphology prediction. Monthly notices of the royal astronomical society, 450(2), pp.1441-1459.

Eifler, T., Simet, M., Krause, E., et al., 2020. Cosmology with the Wide-Field Infrared Survey Telescope–Synergies with the Rubin Observatory Legacy Survey of Space and Time. arXiv preprint arXiv:2004.04702.

Gosmeyer, C.M., The Quicklook Team, 2017. WFC3 Anomalies Flagged by the Quicklook Team. WFC3-ISR 2017-22.

Hausen, R. and Robertson, B.E., 2020. Morpheus: a deep learning framework for the pixel-level analysis of astronomical image data. The Astrophysical Journal Supplement Series, 248(1), p.20.

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

Ioffe, S. and Szegedy, C., 2015, June. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456). PMLR.

Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), pp.2278-2324.

Lin, T.Y., Goyal, P., Girshick, R., et al., 2017. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision (pp. 2980-2988).

Mack, J., Dahlen, T., Sabbi E., Bowers, A.S., 2016. UVIS 2.0: Chip-Dependent Flats. WFC3-ISR 2016-04.

McCullough, P., 2011. Geometric model of UVIS window ghosts in WFC3. WFC3-ISR 2011-16.

McInnes, L., Healy, J. and Melville, J., 2018. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426.

Pang, G., Shen, C., Cao, L. and Hengel, A.V.D., 2021. Deep learning for anomaly detection: A review. ACM Computing Surveys (CSUR), 54(2), pp.1-38.

Paszke, A., Gross, S., Massa, F., et al., 2019. Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32, pp.8026-8037.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

Russakovsky, O., Deng, J., Su, H., et al., 2015. Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3), pp.211-252.

Sahu, K., et al., 2021. Wide Field Camera 3 Data Handbook, Version 5.0. WFC3 DHB.

Simonyan, K., Vedaldi, A. and Zisserman, A., 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034.

Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Szegedy, C., Liu, W., Jia, Y., et al., 2015. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

Valizadegan, H., Martinho, M., Wilkens, L.S., et al., 2021. ExoMiner: A Highly Accurate and Explainable Deep Learning Classifier to Mine Exoplanets. arXiv preprint arXiv:2111.10009.

Wu, J. F., and Peek, J.E.G., 2020. Predicting galaxy spectra from images with hybrid convolutional neural networks. arXiv preprint arXiv:2009.12318.

Zhuang, F., Qi, Z., Duan, K., et al., 2020. A comprehensive survey on transfer learning. Proceedings of the IEEE, 109(1), pp.43-76.

Appendix

A. Model Architectures and Hyperparameters for 1x256x256 Input

Table A1: *Synthetic Model Architecture (2 convolutional layers, 3 fully connected layers).*

Type	**Kernel Size**	**Stride**	**Zero Padding**	**Output Size**
convolution + ReLU	5x5	1	2	16x256x256
max pool	4x4	4	0	16x64x64
convolution + ReLU	5x5	1	2	32x64x64
max pool	4x4	4	0	32x16x16
flatten				1x1x8192
dropout (0.4)				1x1x8192
fully connected + ReLU	1x1	1	0	1x1x512
dropout (0.2)				1x1x512
fully connected + ReLU	1x1	1	0	1x1x64
dropout (0.1)				1x1x64
fully connected	1x1	1	0	1x1x2

Table A2: *Model A Architecture (GoogLeNet convolutional layers, 3 fully connected layers). See Table 1 and Figure 3 in the Szegedy et al. (2015) for full details on GoogLeNet Convolutions.*

Type	**Kernel Size**	**Stride**	**Zero Padding**	**Output Size**
GoogLeNet Convolutions				1x1x1024
dropout (0.5)				1x1x1024
fully connected + ReLU	1x1	1	0	1x1x1024
dropout (0.5)				1x1x1024
fully connected + ReLU	1x1	1	0	1x1x1024
dropout (0.2)				1x1x1024
fully connected	1x1	1	0	1x1x2

Table A3: *Model B Architecture (3 convolutional layers, 4 fully connected layers).*

Type	**Kernel Size**	**Stride**	**Zero Padding**	**Output Size**
convolution + ReLU	3x3	1	1	16x256x256
max pool	2x2	2	0	16x128x128
convolution + ReLU	3x3	1	1	32x128x128
max pool	2x2	2	0	32x64x64
convolution + ReLU	3x3	1	1	64x64x64
max pool	2x2	2	0	64x32x32
flatten				1x1x65536
fully connected + ReLU	1x1	1	0	1x1x16
dropout (0.2)				1x1x16
fully connected + ReLU	1x1	1	0	1x1x32
dropout (0.2)				1x1x32
fully connected + ReLU	1x1	1	0	1x1x32
dropout (0.2)				1x1x32
fully connected	1x1	1	0	1x1x2

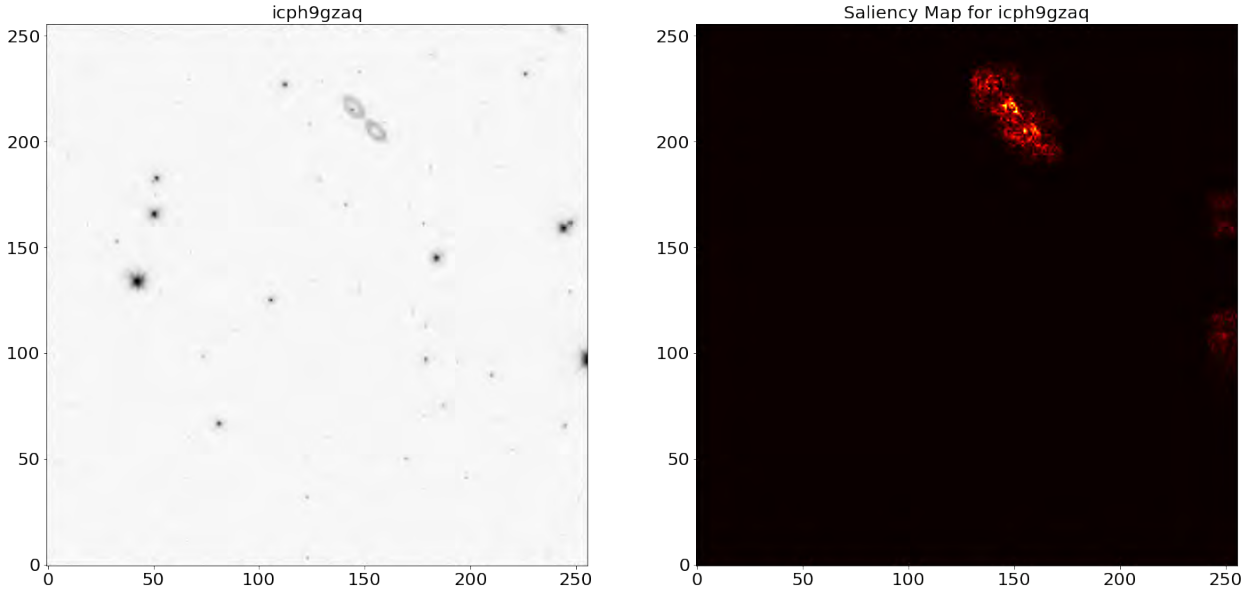
Table A4: *Model C Architecture (3 convolutional layers, 3 fully connected layers). It utilizes batch normalization, which stabilizes the network by recentering and rescaling (Ioffe & Szegedy 2015).*

Type	**Kernel Size**	**Stride**	**Zero Padding**	**Output Size**
convolution + ReLU	3x3	1	1	32x256x256
max pool	2x2	2	0	32x128x128
convolution + ReLU	5x5	1	2	32x128x128
max pool	2x2	2	0	32x64x64
batch normalization				32x64x64
convolution + ReLU	3x3	1	1	8x64x64
max pool	2x2	2	0	8x32x32
batch normalization				8x32x32
flatten				1x1x8192
dropout (0.2)				1x1x8192
fully connected + ReLU	1x1	1	0	1x1x16
fully connected + ReLU	1x1	1	0	1x1x8
fully connected	1x1	1	0	1x1x2

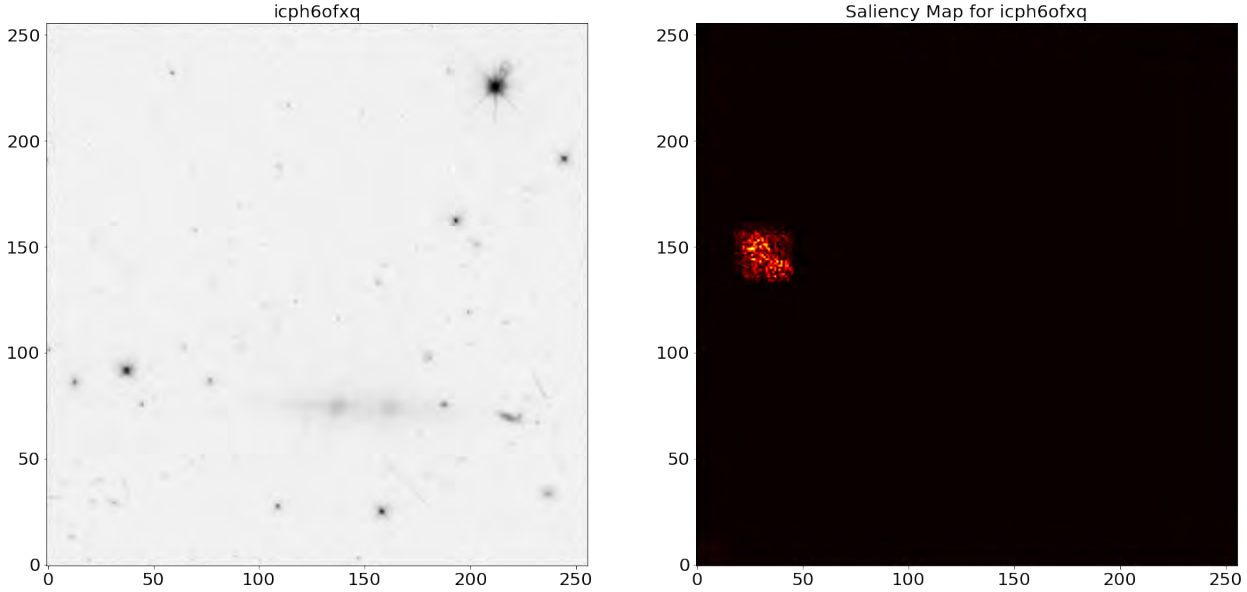
Table A5: *Model D Architecture (3 convolutional layers, 4 fully connected layers).*

Type	**Kernel Size**	**Stride**	**Zero Padding**	**Output Size**
convolution + ReLU	3x3	1	1	16x256x256
max pool	2x2	2	0	16x128x128
convolution + ReLU	3x3	1	1	32x128x128
max pool	2x2	2	0	32x64x64
convolution + ReLU	3x3	1	1	64x64x64
max pool	2x2	2	0	64x32x32
batch normalization				64x32x32
dropout (0.25)				64x32x32
flatten				1x1x65536
fully connected + ReLU	1x1	1	0	1x1x32
dropout (0.5)				1x1x32
fully connected + ReLU	1x1	1	0	1x1x64
fully connected + ReLU	1x1	1	0	1x1x16
fully connected	1x1	1	0	1x1x2

B. Saliency Maps Produced by the Models

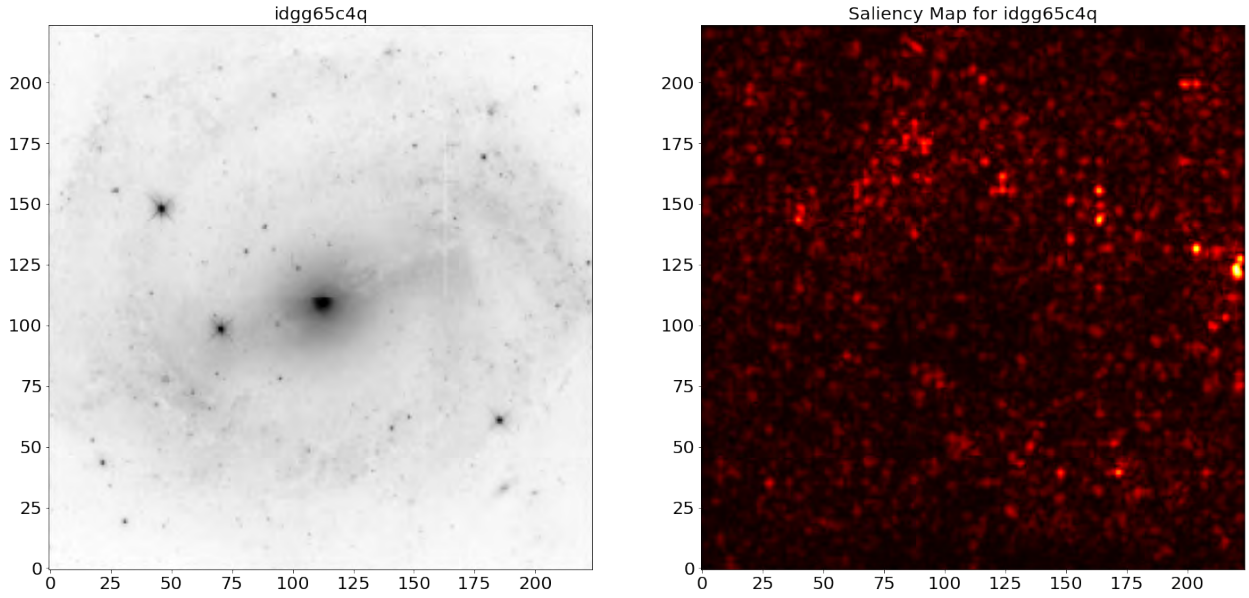


(a) *Observation and saliency map for a TP sample from the test set.*

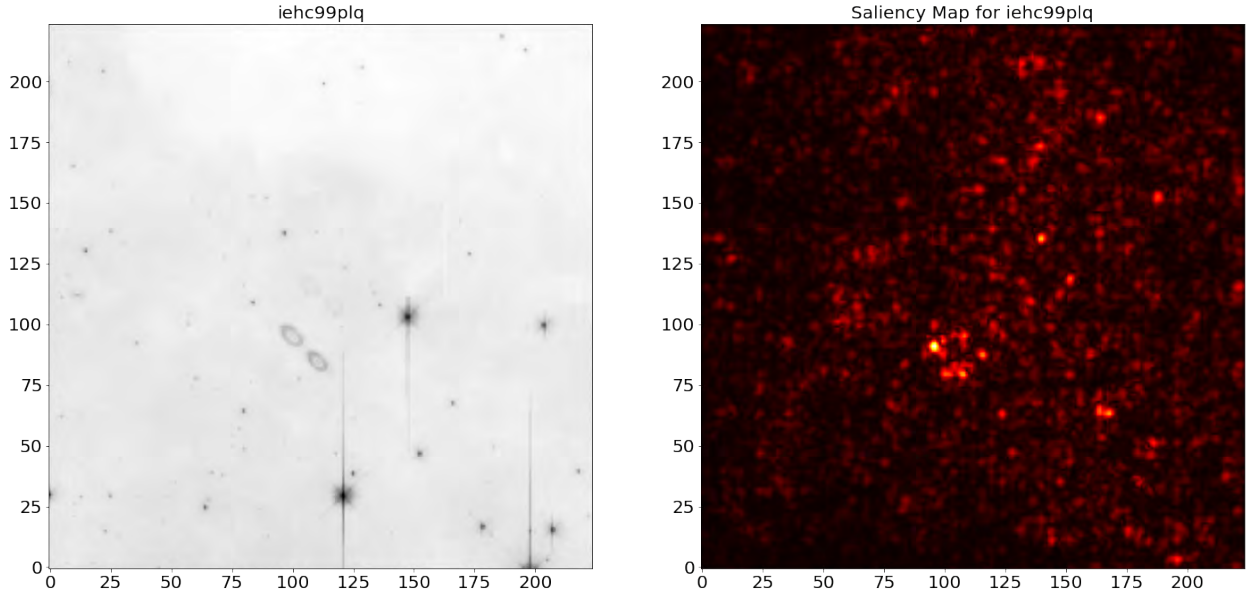


(b) *Observation and saliency map for a FN sample from the test set.*

Fig. B1.—*Observations and saliency maps produced by the synthetic model for true positive (TP) and false negative (FN) samples from its test set. The figure-8 ghosts are highlighted in each map, including b) in which the synthetic model predicted a null classification.*

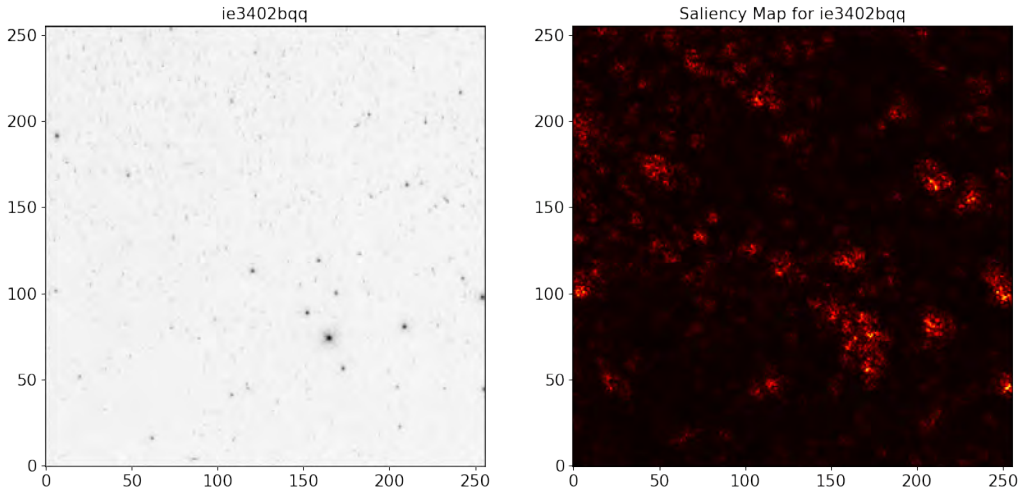


(a) *Observation and saliency map for a TN sample.*

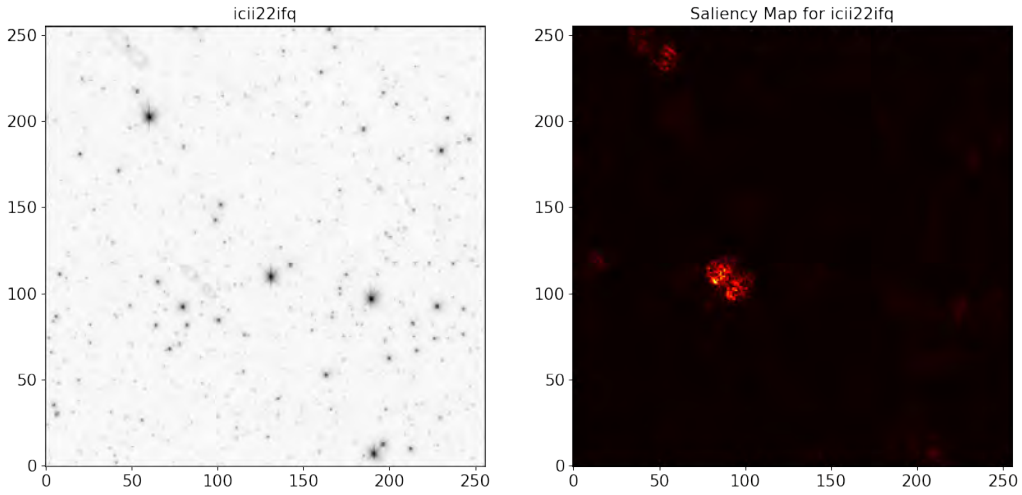


(b) *Observation and saliency map for a TP sample.*

Fig. B2.—*Observations and saliency maps produced by Model A for true negative (TN) and true positive (TP) samples. The maps have scattered highlighted pixels indicating Model A relies on the background to classify the images.*

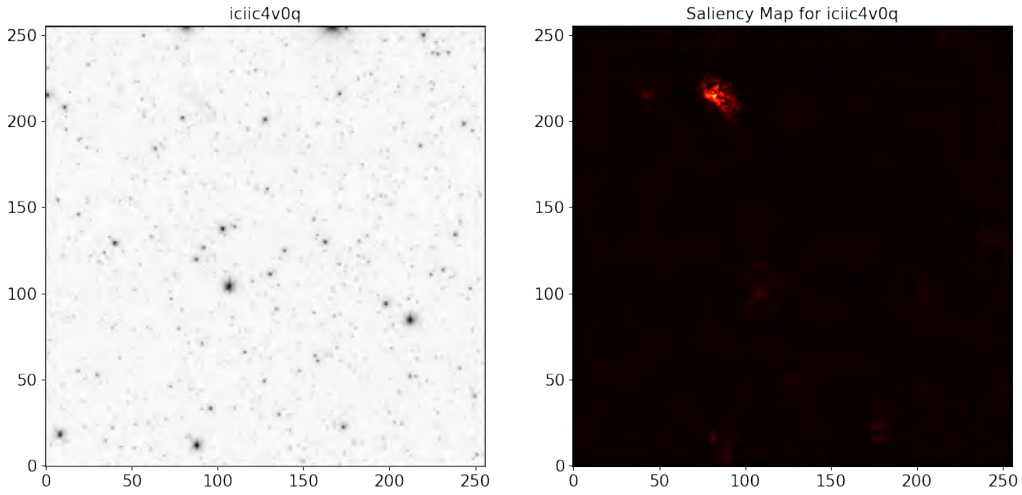


(a) *Image and saliency map for a TN sample.*

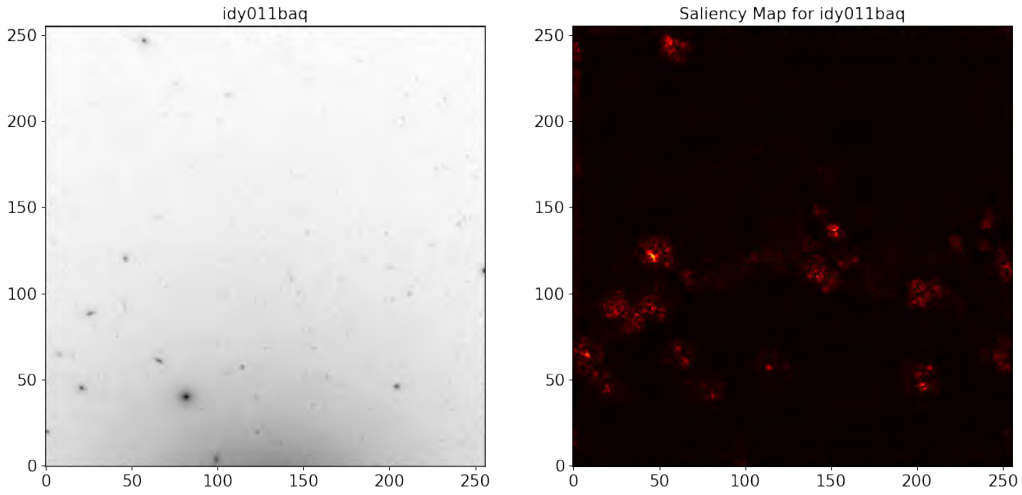


(b) *Image and saliency map for a TP sample.*

Fig. B3.—*Images and saliency maps produced by Model B for true negative (TN) and true positive (TP) samples. Model B focused on stellar objects for null observations and figure-8 ghosts for anomalous observations.*

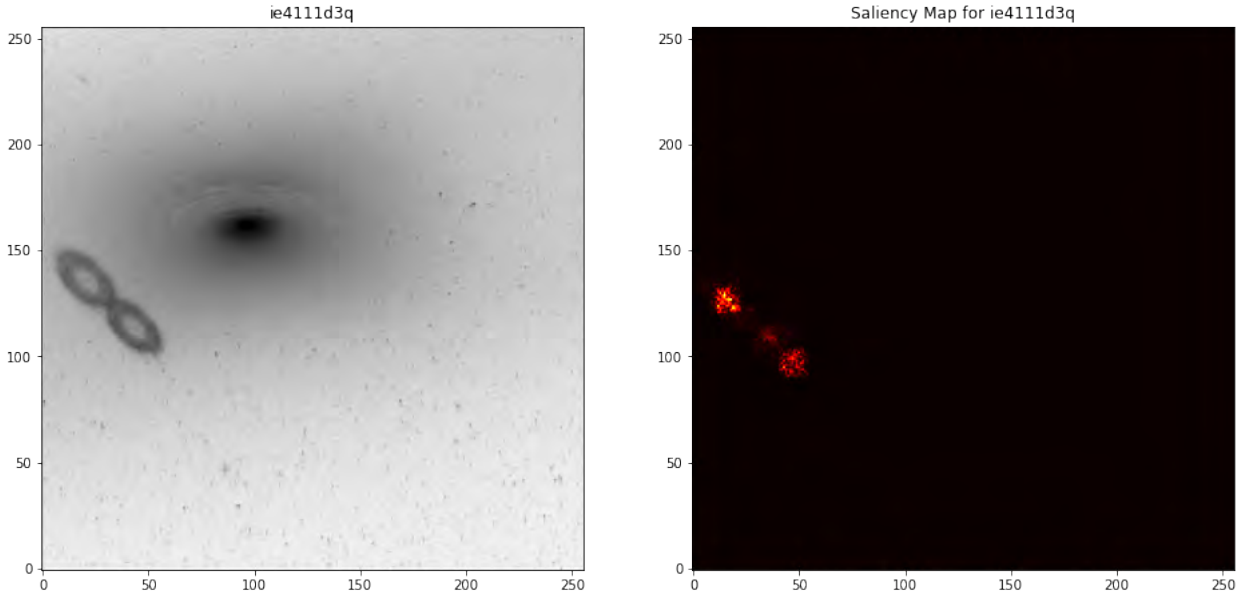


(a) *Image and saliency map for a FN sample.*

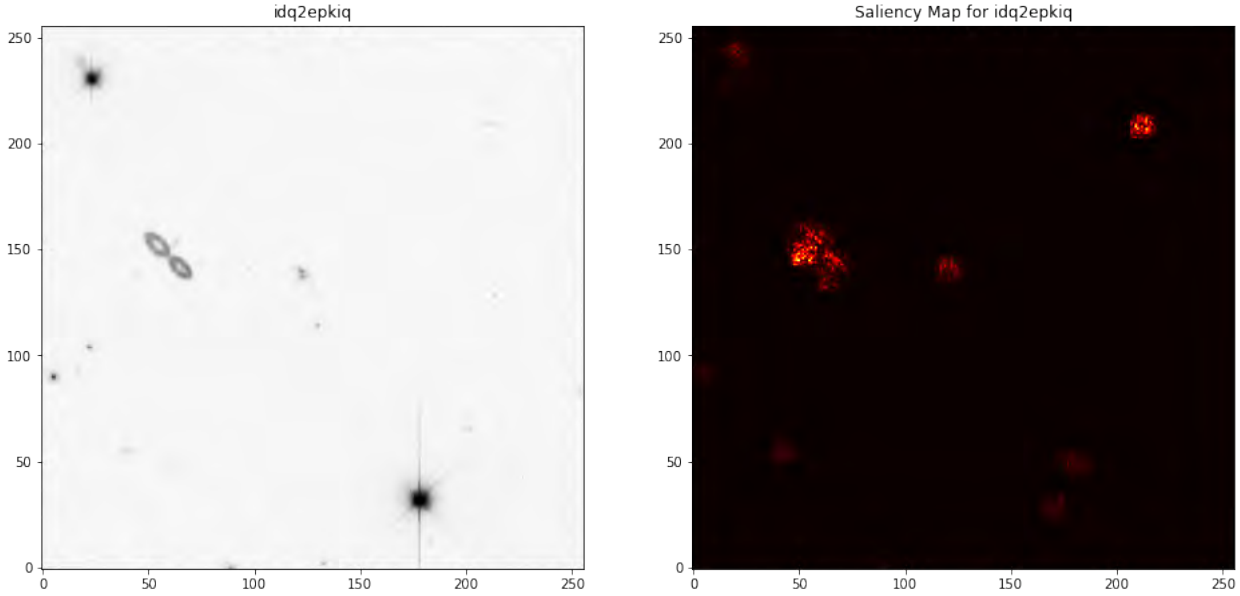


(b) *Image and saliency map for a FP sample.*

Fig. B4.—*Observations and saliency maps produced by Model B for false negative (FN) and false positive (FP) samples. Even falsely predicted by Model B as not having a figure-8 ghost, it still drew attention to the anomaly.*

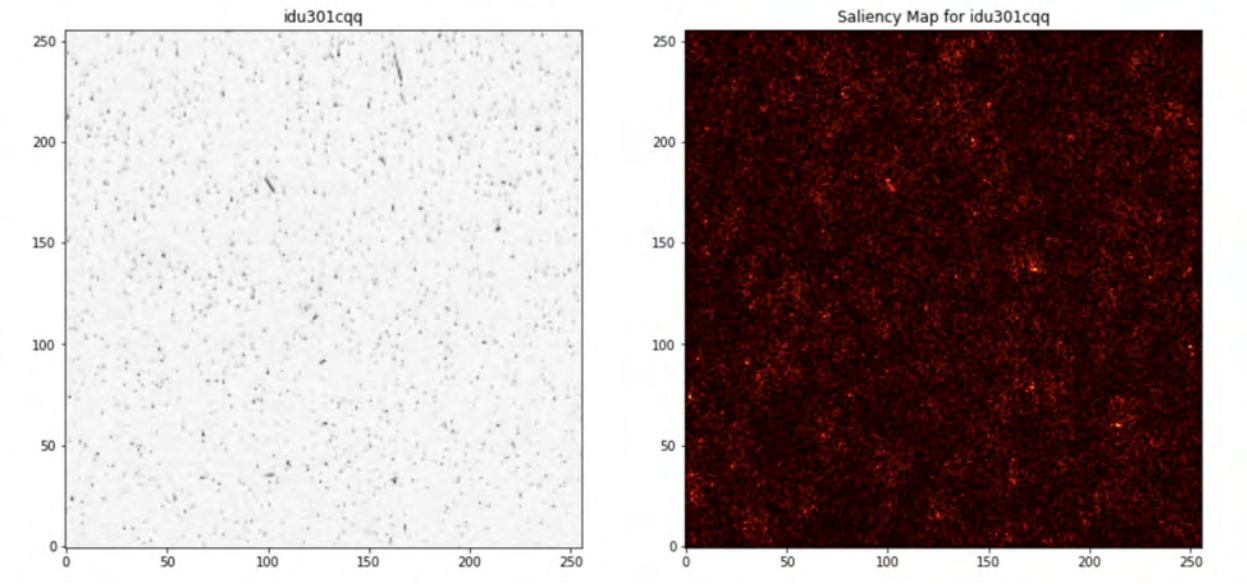


(a) *Image and saliency map for a FN sample.*

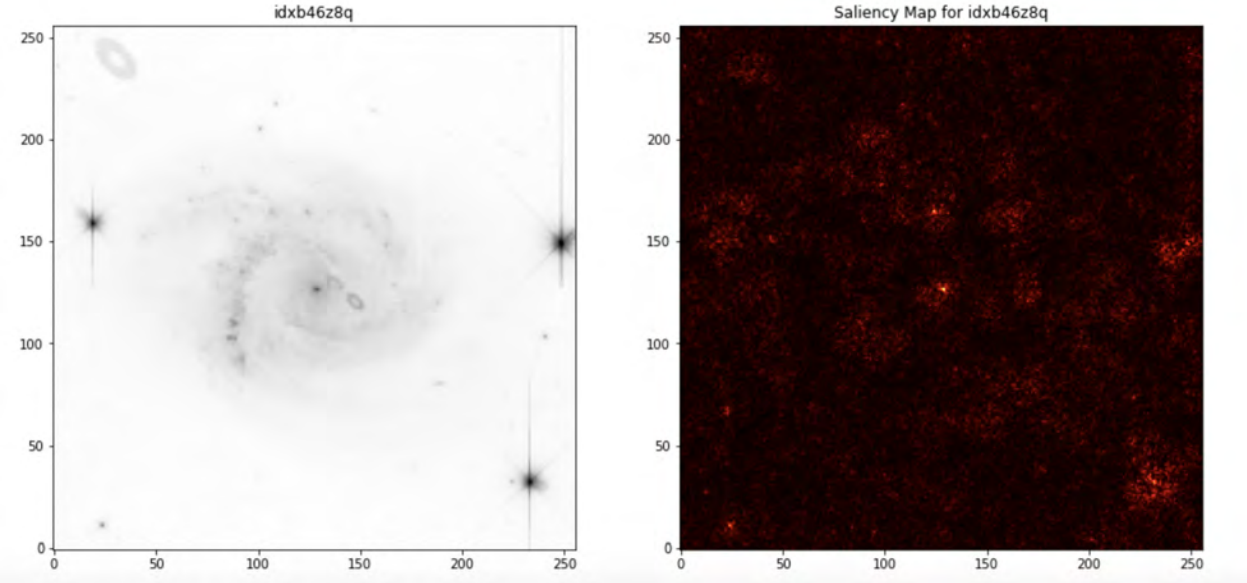


(b) *Image and saliency map for a TP sample.*

Fig. B5.—*Observations and saliency maps produced by Model C for false negative (FN) and true positive (TP) samples. Although Model C was able to highlight figure-8 ghosts of different sizes, it struggled to classify the larger figure-8 ghost.*



(a) *Observation and saliency map for a TN sample.*



(b) *Observation and saliency map for a TP sample.*

Fig. B6.—*Observations and saliency maps produced by Model D for true negative (TN) and true positive (TP) samples. Model D was not able to determine the most predictive features as well as the other models.*