STScI | SPACE TELESCOPE SCIENCE INSTITUTE

WFC3 Instrument Science Report 2024-14

Using machine learning for object classification and filtering

V. Bajaj, F. Dauphin

December 20, 2024

Abstract

We present a process for designing and training a convolutional neural network to classify detections from star finding algorithms in the exposure (FLT/FLC) frame for WFC3/UVIS images. This network can be used to filter out common spurious detections (cosmic rays or diffraction spikes), which are often present in imaging data where the point spread function is under or critically sampled (FWHM = 2 pix). The neural network achieved high accuracy for correctly identifying stars (96%). Falsely detected sources in catalogs (a common side effect of DAOFind-like algorithms) can cause incorrect matches and resulting astrometric transforms. Eliminating false detections can provide significant improvements to image alignment workflows and clean photometric catalogs. We also present caveats of this method, as well as general considerations for building neural networks for use in astronomical data analyses.

1. Introduction

A very common task in data analysis of astronomical images is the detection of point spread functions (PSFs) in images, which usually are created by stellar objects. It is often desirable to make detections in the exposure frame, rather than coadded (drizzled) frames due to prior knowledge of the PSF for tasks like PSF fitting photometry.

However, in the case of the undersampled PSF (when the full width at half maximum or FWHM is less than 2 pixels) popular algorithms similar to DAOFIND (Stetson 1987), which detect local maxima based on FWHM and brightness inputs, often return many false

positives. This occurs because other sharp features such as diffraction spikes and cosmic rays (in the case of space based observatories) also appear to be "point-like" (look like stars) to these kind of algorithms as much of the flux is contained within just a few pixels. These resulting false positives contaminate the catalog of detected sources. While some of the morphological information calculated by the detection algorithms is useful for filtering out these spurious sources, this often requires tailoring the cuts made for each dataset, limiting efficacy. Furthermore, other source detection algorithms such as local maximum detectors (e.g. the `find_peaks()` function in PhotUtils Bradley et al. 2017), while being more broadly applicable than DAOFind style algorithms, do not consider morphological properties of sources, just their brightness. Catalogs of local maxima thus include sources both astrophysical sources (e.g. galaxies, HII regions, AGNs, gas emission and stars) as well as optical/detector artifacts (e.g. diffraction spikes, cosmic rays, scattered light, optical ghosts, bad pixels and other PSF features). As a result, filtering sources returned from these types of detection algorithms requires even more post processing to generate clean catalogs. Clean catalogs are especially necessary when aligning imaging data via matching of point source positions to other images or catalogs via TweakReg (Fruchter and et al. 2010), as the false detections can be used to generate spurious matches between the catalogs, leading to incorrect astrometric transforms (and thus poor quality coadded images). The problem is magnified for deep imaging of sparse fields, where there may not be many actual point sources, but many cosmic rays which can overwhelm the stars when matching sources. Additionally, when combining photometric catalogs from many images, cross matching sources can often become muddled with cosmic rays or diffraction spikes, yielding poor final measurements.

In recent years, advances in computer vision have streamlined the process of creating software pipelines for detection and classification of objects in images. In many cases, these are processes are powered by neural networks, which are machine learning models that "learn" (empirically fit) extremely complex functions to data. The models are "trained" on large volumes of sample data, with the end goal of being able to recognize, classify, or predict properties of new data. Whilst many of the well known applications use these networks to detect every day objects or features (e.g. cars, faces, roads, buildings, etc.) the same tools can be employed to find objects in astronomical images. Example applications to HST/WFC3 data include the detection of optical ghosts in images (Dauphin et al. 2022) and guide star failure identification (Jones and Dauphin 2024). In this paper, we present a workflow to create a neural network that can be used to classify the detections from peak finding algorithms into 3 different classes: stars, cosmic rays, and PSF features (e.g. diffraction spikes, halos etc).

2. Data

The datasets used in training and validation of the neural network were selected based on three criteria. First, the images must contain a sufficient number of stars to yield sufficient data for the training sample, while not being overly crowded. Second, the stars must cover a wide range of fluxes, so the network could be trained to recognize both faint and bright stars, as well as artifacts around the bright stars. Third, the exposures had to be long enough to contain sufficient number of cosmic rays, so the network could learn how cosmic rays appear.

Observations of the outskirts of the Milky Way globular cluster Ω Centauri satisfied these criteria and were chosen for the study. While the core of the cluster is crowded even for HST, the chosen field is approximately 11 arcminutes from the center, where the stellar density is much lower. The specific images are listed below in Table 1.

Image	Filter	Exptime [s]	Date-Obs	Roll Angle
ictj49naq_flc.fits	F336W	1230.0	2016-07-05	259.910706
ictj50b6q_flc.fits	F336W	1230.0	2016-07-01	259.907806
ictj51nlq_flc.fits	F336W	1230.0	2016-07-05	259.913391
ictj52ktq_flc.fits	F336W	1230.0	2016-07-03	259.910492

Table 1: The metadata for each of the images used in the training and validation sets of the data. While the data were observed in different visits and on different days, the data cover the same field and are executed at nearly identical roll angles.

3. Methods

Generating Labels

The fully calibrated (FLC) exposures listed in Table 1 were downloaded from the Milkulski Archive for Space Telescopes (MAST) and processed with `HST1Pass` (Anderson 2022) to generate initial source catalogs. The HST1Pass catalogs were then used to align the images via TweakReg, and the images for each filter were combined using AstroDrizzle (Fruchter and et al. 2010).

Next, a peak finding algorithm similar the PhotUtils `find_peaks()` function was run on each image to detect local maxima. Each maximum was only recorded if it was at least 5 pixels away from any brighter pixel, and had at least a minimum flux of 100 electrons (to ensure a minimum number of pixels were above the background) within the 2x2 pixel box surrounding the center of the source. All the maxima in the image that met this criteria were stored in single catalog for that image. These individual image catalogs were then combined into a merged catalog, with any sources within 40mas of each other (when projected onto the sky) considered to be the same source, and thus having their position measurements averaged.

Subsequently, the drizzled image data underwent normalization. The ZScale[1] limits (z_1, z_2) were first calculated and the data was clamped within the range of $(z_1, 100 * z_2)$. The data was then logarithmically scaled and further normalized to span the interval [0,1]. The (RA, Dec) coordinates from the merged catalog were projected onto (x, y) positions within the drizzled image using the World Coordinate System (WCS) associated with the image data. A 11x11 pixel cutout centered at each (x, y) position was extracted from the normalized drizzled image. Only cutouts corresponding to regions of the drizzled image covered by at least two of the input exposures were retained for subsequent analysis to ensure cosmic rays were not present in the drizzled image cutouts for label generation (but would be present in the FLT image cutout).

[1]For explanation of the ZScale algorithm see: https://js9.si.edu/js9/plugins/help/scalecontrols.html

The cutouts were then processed through a UMAP dimensionality reduction transform (McInnes, Healy, and Melville 2020). This transform projects high dimensionality data (each pixel of the 11x11 cutout being its own dimension) into a much lower dimensional space, while preserving global structure (maintaining distances between groups of samples in feature space). This allows similar looking objects to be grouped together, while separating from other types of objects in the low dimensional space. The parameters for the UMAP fit were varied until populations of points clearly visually separated into different clusters. A clear separation of clusters was obtained with the following parameters: `n_neighbors=35, min_dist=.01, metric='euclidean', n_components=2`. The choice of values for `n_neighbors` and `min_dist` were selected to balance how tightly clustered the points were and how separated the clusters were. The final embedding is shown in Figure 1. The choice of 2 components was to simplify visualization and cluster selection while allowing sufficient breadth of output parameter space. Several samples were drawn from each of the clusters of points to determine which morphological type the cluster belonged to. Each cluster was labeled with a different numerical ID, and points in each cluster were then assigned that label in the merged catalog.

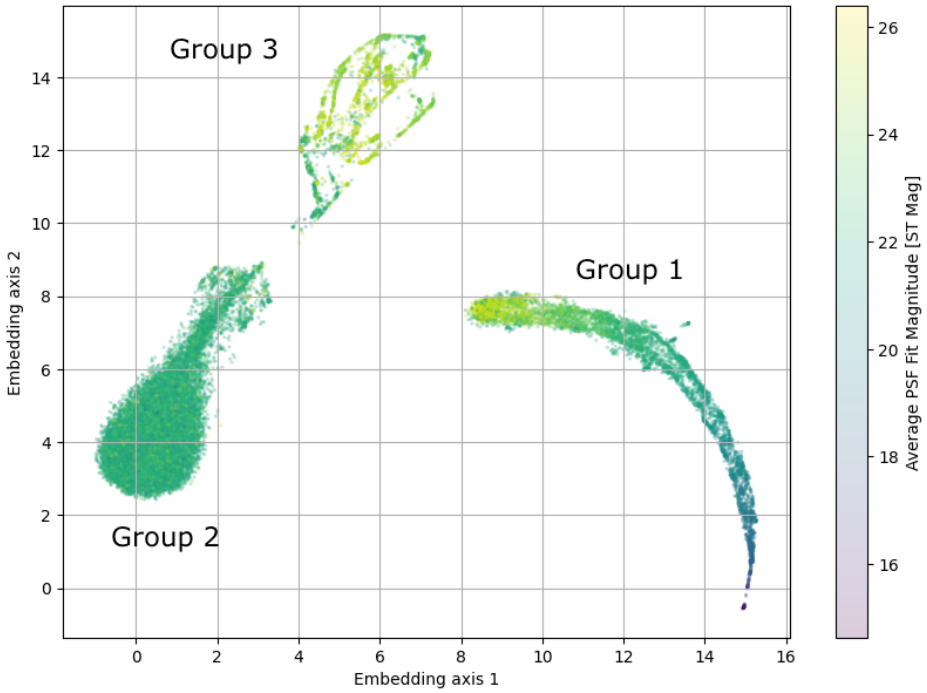

Figure 1: Visualization of cutout embeddings in the two dimensional space generated by UMAP. Each point represents an individual cutout, with spatial proximity indicating similarity in the feature space. The points are colored by the mean HST1Pass PSF fit magnitude of the source in the FLC images in which it appears. The embedding notably reveals the separation of the images into three distinct groups, suggesting the presence of different morphological classes within the data. Note that the axes are arbitrary and do not correspond to any directly interpretable features; they represent abstract dimensions created by the dimensionality reduction process.

As the groups of points formed three distinct populations, we separated the cutouts into three clusters with the following constraints:

- Group 1: $x > 7.5, y < 10$
- Group 2: $x < 5, y < 10$
- Group 3: $y > 10$

Where (x, y) are the coordinates for each cutout on the UMAP embedding axes for the feature space shown in Figure 1. To ensure these clusters actually correspond to different types of objects, random samples from each group were selected. A subset of these samples are shown in Figure 2. Visual inspection of these samples revealed that the three classes of objects corresponded to stars, cosmic rays detected in the FLC frame, and diffraction spikes/other sharp portions of the PSF halo for groups 1, 2 and 3, respectively. For clarity we shall subsequently refer to each class as stars, cosmic rays (CRs) or spikes.

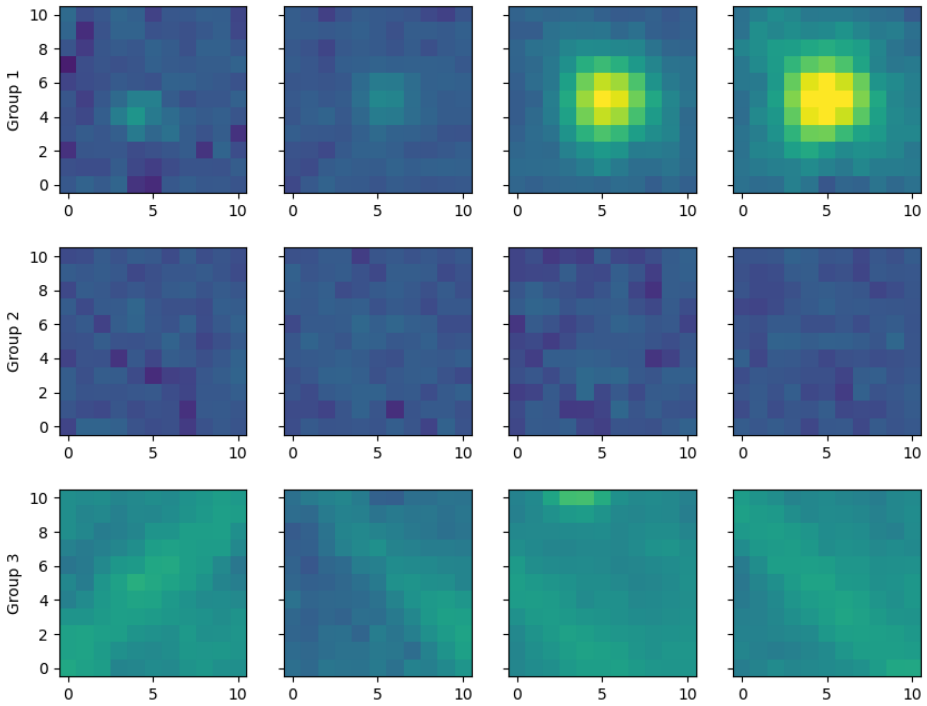

Figure 2: Cutouts of random samples taken from each group from the UMAP embedding. The top row corresponds to group one, which appears to contain stars. The middle row corresponds to group two, which appears to contain no objects in the drizzled image (which likely means these samples were cosmic rays detected in the FLC frame). The bottom row corresponds to group three, which appears to contain diffraction spikes or other parts of the PSF halo.

	Class	Embedding region	Training Samples	Validation Samples
Group 1	Star	x >7.5, y <10	5526	3258
Group 2	Cosmic Ray	x <5, y <10	5526	8347
Group 3	Spike/Halo	y >10	3711	928

Table 2: Summary of object classes and training/validation set contents.

Generating Datasets for Neural Network

With the labels for supervised learning generated, the datasets for training and validating the neural network were created. Since the desired frame for detection and classification was the exposure (FLC) frame, the positions in the merged catalog (and their corresponding labels) were cross-matched to the individual FLC catalogs. For further discussion of why the FLC cutouts were used, see Section 5. The ZScale, log scale, and normalization procedures applied to the drizzled frame were also applied to each exposure's data. Cutouts of size 11x11 pixels were extracted for all matched sources within an exposure. Note that a cutout was only extracted if the position in the merged catalog had a corresponding source in the image catalog within 0.5 pixels. This constraint is necessary as a given source in the merged catalog may correspond to a cosmic ray, which would likely appear in only one of the exposures. If all positions in the merged catalog were simply projected back to the exposure frame, many resulting cutouts would contain only empty sky (a cosmic ray in one exposure does not appear in the others). This procedure was repeated for all input exposures.

The dataset, comprised of 27376 samples, was then split 80/20 for training and validation, respectively. To balance the training set any label's dataset that was 50% larger than the smallest label's had excess samples above the extra 50% placed in the validation set. This rough balancing of the features improved training performance. The resulted in a final training set with 5526, 5526, and 3711 samples and a validation set with 3258, 8347, and 928 samples for stars, CRs and diffraction spikes/PSF halo objects, respectively.

Constructing Neural Network

Given our goal of classifying sources detected in images based on their shape, we selected a convolutional neural network (CNN) for this task (LeCun, Bengio, and G. Hinton 2015). Convolutional layers in neural networks are particularly well-suited for identifying features, such as shapes and edges, in images before passing outputs to fully connected layers. We therefore developed a compact model consisting of four layers in total (excluding the input and output layers), with the first two layers being convolutional and the last two fully connected (dense) layers. A 3x3 kernel with a stride of 1 and padding of 1 was used for both convolutional layers, to ensure resolution is preserved. After each convolutional layer, batch normalization (BN) and a rectified linear unit (ReLU) activation function were applied. Unlike typical CNNs max pooling (see Max Pooling in Section 5) was not applied to the outputs of the convolutional layers, as the cutout size was already small, and the features of interest were sharp and only a pixel or two across. Following each of the fully connected layers a ReLU activation function was applied. The model architecture is shown in Figure 3.

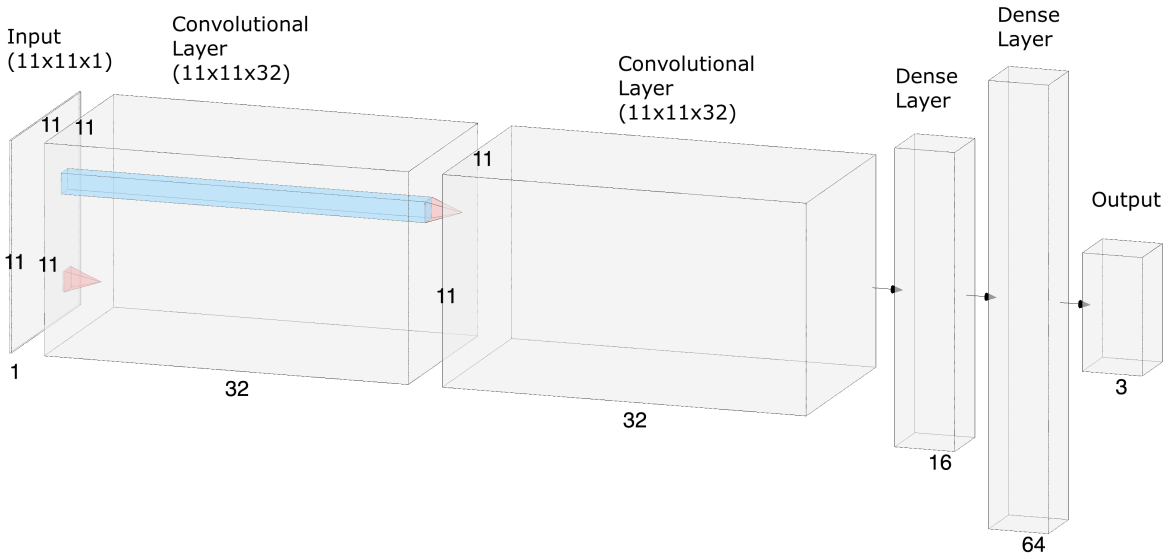

Figure 3: Diagram of the convolutional neural network model used in this analysis. The two convolutional layers both use 32 filters. After the convolutional layers, batch normalization and ReLU activation functions are used (not shown). After the dense layers, a ReLU activation function was used (not shown). The convolutional layers are used to identify spatial features. After the second convolutional block, the data is "flattened" from 11x11x32 to 1x3872 and passed to the first dense layer containing 16 neurons. The blue and red shapes are visual aides to show how the convolutions feed into the next layer. Diagram created via NN-SVG (LeNail 2019)

Model Training

Training was performed over 10 epochs with a batch size of 128. The cross-entropy loss function was minimized using the Adam optimization algorithm (Kingma and Ba 2017). Following each epoch the model parameters, along with the corresponding training and validation loss values, were recorded, as well as the overall model validation accuracy. To mitigate the effects of overfitting the parameters corresponding to the epoch with the lowest validation loss were selected for the final model. The model was trained multiple times until a high accuracy model was obtained.

4. Results

The best model performed quite well at identifying all three morphological types. Both training and validation loss improved steadily over the first seven epochs of training. Validation loss showed little improvement after the seventh epoch and increased in the final epoch (indicating overfitting as training loss decreased), despite a small increase in accuracy over the last few epochs (see Figure 4). As the last epoch had a lower accuracy, and a validation loss that diverged from the training loss, the parameters from the penultimate training epoch

were selected for the final model.

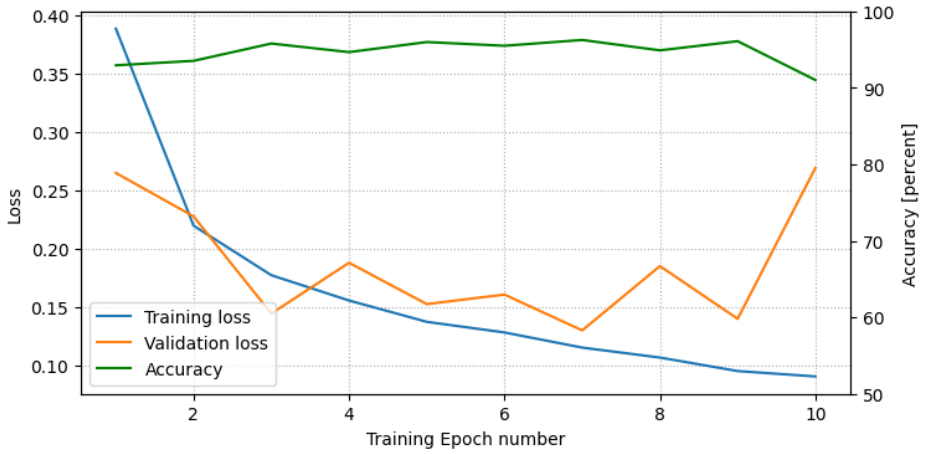

Figure 4: Training and validation loss, and accuracy, over 10 epochs of training. The left axis represents the loss values for both the training (blue) and validation (orange) datasets, whereas the right axis corresponds to the accuracy (green line).

To quantify the performance in each of the different classes a confusion matrix was generated to show the accuracy for identifying a given class. The confusion matrix was generated by comparing the the model's predicted class for each cutout in the validation set, and comparing the prediction to the true label. The model has high true positive rates across all three classes. In the cases where true stars are misclassified (approximately 6%) they are mostly mislabeled as cosmic rays. More notably, less than 1% of the time are cosmic rays misclassified as stars, and less than 1% of diffraction spikes are misclassified as stars. The low false positive rate and high true positive rate for stars renders this model especially good at making sure objects flagged as stars are truly stars and not other objects. The full confusion matrix is shown in Figure 5.

External Test Field

To further test model performance cutouts of sources in FLC images of a separate region of Ω Cen were classified by the model and predictions were inspected visually. Visual inspection indicates excellent performance, with false positive detection of stars very infrequent as shown in Figures 6 and 7 (note that the drizzled image is shown in Figure 7 for clarity, so the cosmic rays are removed, though the objects were classified in the FLC images). While not every single star is detected and a small percentage of the real stars are flagged as CRs or spikes, the model ensures that the objects classified as stars are actually stars over 96% of the time.

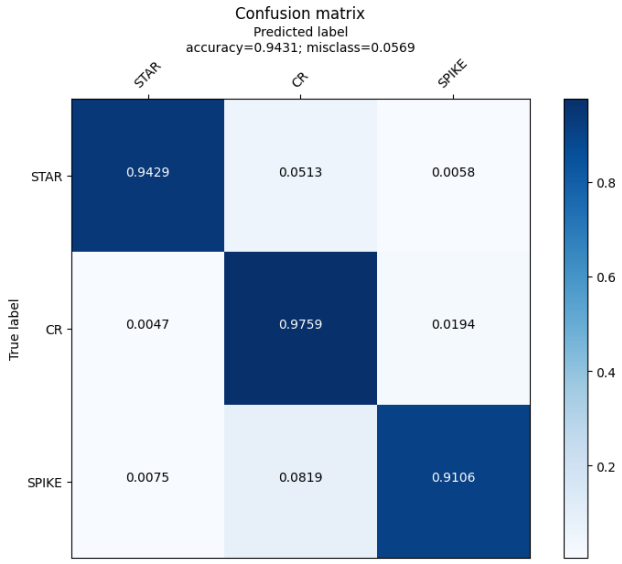

Figure 5: Confusion matrix illustrating the performance of the neural network model in classifying sources into the three source types. The matrix shows the normalized percentage of correct and incorrect classifications, with the diagonal elements representing the accuracy for each class. The color intensity corresponds to the proportion of predictions, where darker shades indicate higher accuracy.

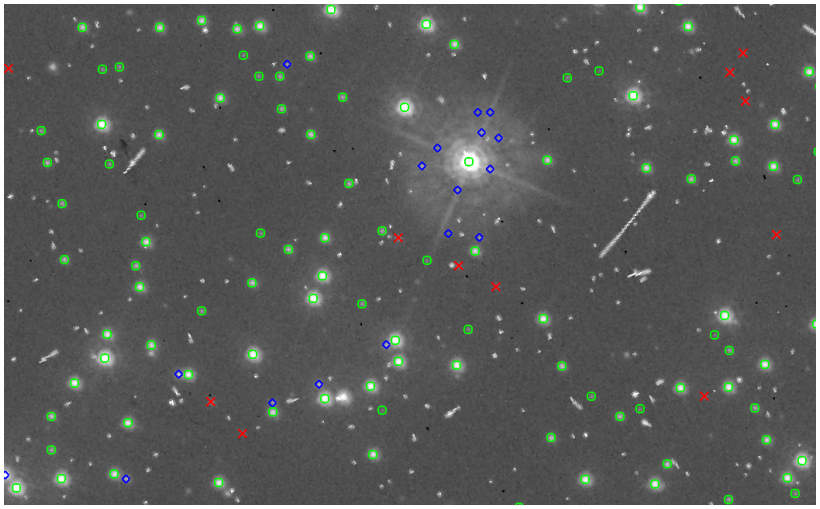

Figure 6: Visualization of a region the FLC image of the external testing field used for neural network evaluation. The green circles represent detections classified as stars, the red X's denote detections identified as cosmic rays, and the blue diamonds correspond to diffraction spikes or PSF halo features. While some stars are marked as CRs/Spikes (false negatives), very rarely is the reverse true. Note that not all cosmic rays were detected by the initial detection pass, and are thus not classified.

5. Discussion

UMAP vs. Neural Network Classification

The rationale for training a neural network despite the apparent success of the UMAP transform in separating cutouts into different classes, may initially seem unclear. In theory,

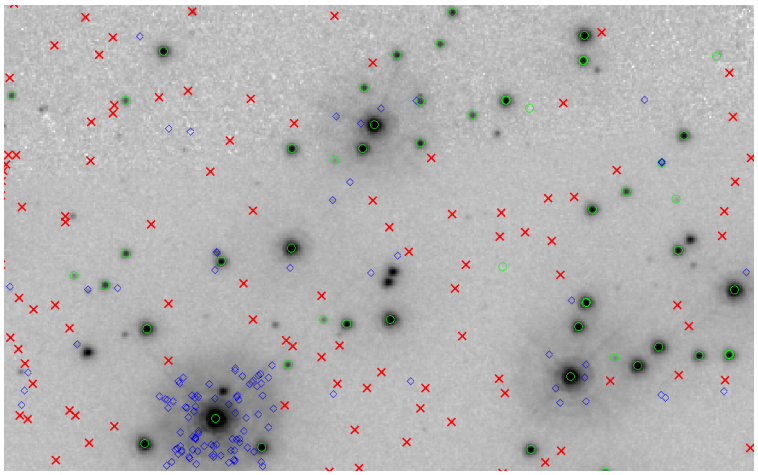

Figure 7: Visualization of a separate region of the drizzled image of the external testing field used for neural network evaluation. The meaning of the shapes/colors of markers is the same as Figure 6. Note that the drizzled frame is shown here with detections from all the input images projected onto the field. As the image is the combined drizzled image, the cosmic rays have been rejected, though each classification was performed in the FLC frame, where the cosmic rays are present. These classifications highlight the model's ability to distinguish between different types of stars and artifacts, even in the lower SNR region near the top of the image.

using the fit UMAP transform and labeling points by their embedded position in the low dimensionality space, as done for label generation, seems to have already solved the problem. In this section we discuss the motivations behind training the neural network rather than relying solely on the UMAP transform/classification.

The UMAP classifier was applied to cutouts from the drizzled image for two primary reasons. First, the drizzled image, being a coadded composite image, exhibits a higher signal-to-noise ratio, which enhances the clarity of object detection and visualization. This is particularly advantageous for faint stars with extended features that approach the noise floor in the FLC image. Second, cosmic rays are rejected in regions of the drizzled image that are covered by two or more of the input FLC images. Consequently, positions corresponding to cosmic rays in an FLC frame appear blank (only contain sky background) in the drizzled image, facilitating the UMAP algorithm's ability to distinguish these artifacts from stars or diffraction spikes. However, because cosmic rays manifest differently in the FLC frame, the UMAP classifier trained on drizzled image cutouts often misclassifies objects, as demonstrated in Figure 8. Therefore, the UMAP classifier is effective only when applied to the drizzled image and within regions covered by multiple input images for cosmic ray rejection.

Moreover, point spread functions (PSFs) and noise characteristics in drizzled images can vary substantially due to several factors, including the number of input images, their roll angles, exposure times, subpixel dither positions, and more. These variations lead to differing embedding patterns for cutouts derived from different drizzled images, often without clear separation of object classes– rendering the existing UMAP classification scheme ineffective.

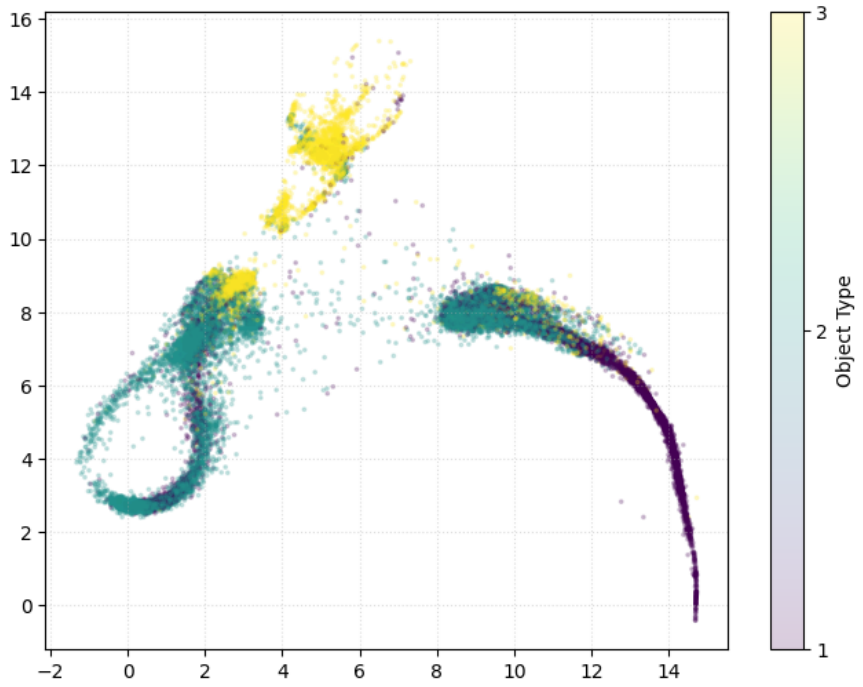

Figure 8: DRC-trained UMAP embedding of object cutouts taken in the FLC frame, colored by their object type as classified with the DRC frame cutout. The axes are arbitrary, as they are the same as Figure 1. The object types 1, 2, and 3 correspond to stars, cosmic rays and diffraction spikes, respectively. The mixing of colors in each of the three distinct clusters indicates the existing UMAP transform cannot separate the object types in the FLC frame (or each of the clusters would be a single unique color).

Additionally, generating a clean drizzled frame necessitates at least two exposures in the same filter of a given field to ensure cosmic ray rejection and requires that these exposures be well-aligned astrometrically. Even under these conditions, if certain regions of the drizzled image are covered by only one of the input exposures, the UMAP classifier's performance would degrade similarly to that as shown in Figure 8

In contrast, the changes to the PSF for a given filter in the FLC frame are much more subtle from image to image, typically involving minor variations in PSF concentration. Moreover, populations of cosmic rays and diffraction spikes exhibit consistent characteristics across different exposures. **As a result, a neural network trained on the FLC frame generalizes more effectively to other datasets**, making it a more robust solution for classification tasks.

In summary, the DRC cutout UMAP classification procedure was useful to generate labels that could be mapped back to the FLC frame, but could not perform well in the FLC frame, thus limiting generalization. The neural network was able to perform accurately in both the FLC and DRC frame for this data, as well as generalize to other datasets.

Network Size

The structure of a neural network is often a key factor in it's efficacy at a given task. In many cases such as the foundational AlexNet (Krizhevsky, Sutskever, and G. E. Hinton 2012) and its successor VGGNet (Simonyan and Zisserman 2015), deep (many-layered) neural networks containing several convolutional blocks are used for general image recognition and perform quite well. Often in astronomy and other research fields, these large networks are used as the base architecture for a model, with the last few layers being retrained to learn a specific classification, in a process known as transfer learning (Pan and Yang 2010). This approach is often successful in many astronomical use cases (see George, Shen, and Huerta 2018 and Ackermann et al. 2018), as many of the filters in the initial convolutional blocks are responsible for tasks such as edge detection, which is a common facet of nearly all computer vision algorithms. As such, we also attempted to use transfer learning with VGG16 (a VGGNet architectures with 16 hidden layers), however the model failed to achieve above 35% accuracy when tested on this dataset.

In the case where a model only needs to identify a small number of different object types with simple structure, and the individual input data are quite small, such large networks are not required (and in our case, fail to) to achieve excellent performance. More notably, as the number of parameters increases, model performance becomes much harder to understand and can quickly become unreliable as often seen with the so-called "hallucinations" of large language models. As AI models grow in complexity, they also grow significantly in computational requirement and energy consumption for both training and inference. Specifically in the case of our dataset, training the VGG16 transfer learning model took approximately 20 times longer to train than our small network and failed to achieve an accurate solution.

Smaller neural networks are fairly simple to design, and both training and inference are much faster. Training and validating the network architecture presented in this paper over 10 epochs with approximately 11000 samples takes only a few seconds, compared to minutes for VGG16 (with only last 3 layers being retrained) or hours for larger networks, or networks where many convolutional layers are being trained. Thus, small networks allow for much more rapid iteration and testing, and in this case adds minimal additional computational time when deploying this network in a full source detection algorithm.

Max Pooling

Many popular convolutional neural network architectures employ downsampling to reduce feature space (size of the image), and computational time. Typical algorithms for this include average pooling or, more popularly, maximum pooling. Max pooling computes the maximum value in a $N \times N$ box and discards the other values, thus downsampling (reducing the number of pixels) by a factor of N. Typically, a 2x2 max pooling is used (along with a stride of 2) after convolutional blocks. For images of every day objects, this is often helpful, as objects of interest are typically tens to hundreds of pixels across.

In the case of astronomical imaging, however, objects are often significantly smaller. Point spread functions for cameras aboard HST and JWST have FWHM values of just 1-2 pixels. Cosmic rays and other artifacts can also be similarly sized. Thus, the reduction in feature space can lead to a reduction in neural network performance. A faint star with only

a few pixels significantly above the background would likely appear very similar to a small cosmic ray after max pooling. To test the effect we retrained the model 50 times with and without max pooling after the two convolutional layers and compared accuracy. In our case, the effect of including max pooling is small, but overall detrimental, reducing overall model accuracy, shown in Figures 9 and 10. Using more convolutional blocks with max pooling would likely result in further degradation.

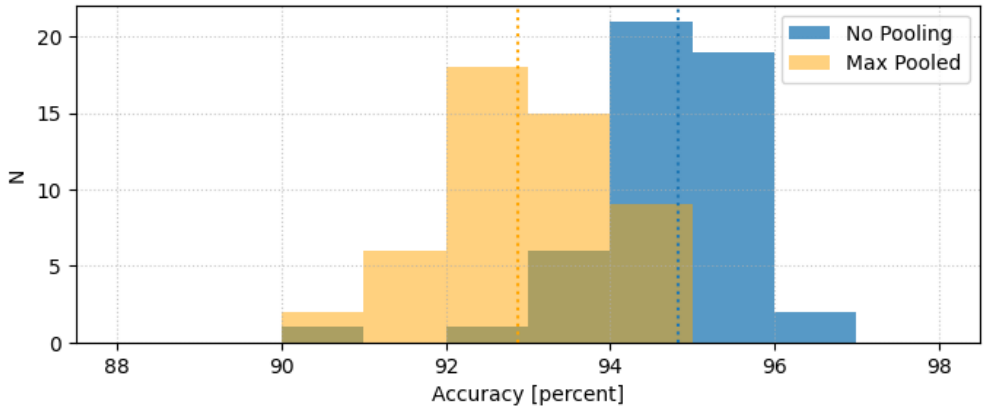

Figure 9: Overall model accuracy comparison for 50 networks with (orange) and without (blue) max pooling. The medians of each population are shown by the dashed vertical lines of matching color. The difference between the medians is approximately 2% in favor of the non pooled network.

In the case of larger networks, several max pooling layers are applied. In the case of VGG16 there are 5 max pooling blocks. As the data in this experiment are relatively small (11x11 pixels), and the structures within them generally small as well, we suspect the excessive down sampling likely removes much of the spatial information from the data before getting to the fully connected layers, leading to poor model performance.

Caveats and Future work

While the model performs quite well in the test cases presented thus far, there are some aspects to be considered when applying it to other cases. First, the model was only trained on cutouts taken from a small number of WFC3/UVIS F336W FLC images. Naturally, applying the model to images taken from other instruments, or even other types of UVIS filters may not have similarly good performance. As the model must learn what the PSF looks like in detector space, PSFs for other detectors are likely not recognizable. For example, the diffraction spikes taken in ACS/WFC images are very nearly parallel to the x and y axes of the detector, whereas for WFC3/UVIS they are rotated 45 degrees. In the case of other UVIS filters the PSF may broaden or shrink, and in the case of medium/narrow band filters, may have more obvious diffraction patterns. However, the model does seem to already perform well in other UVIS wide band filter data (as the PSF only changes subtly in the wide bands, this makes sense). It is likely that broadening the training set to encompass more filters would allow the model to learn these features, though may require an increased

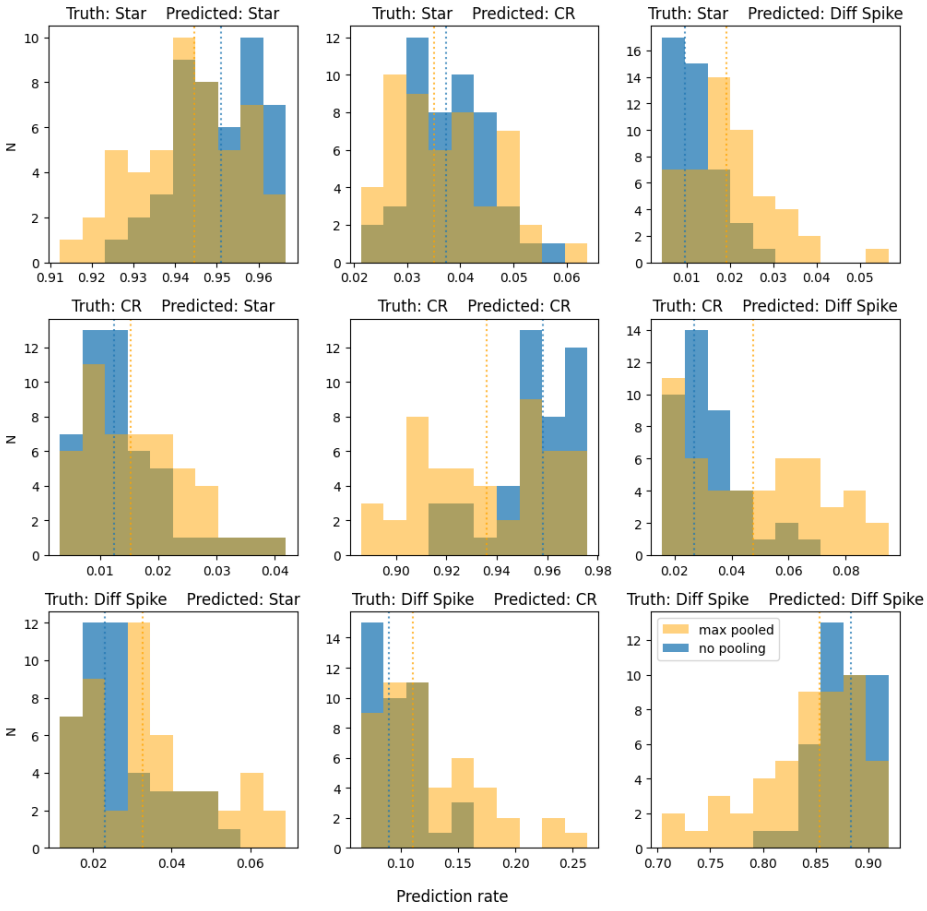

Figure 10: Accuracy comparison for 50 networks with (orange) and without (blue) max pooling separated by class. The medians of each population are shown by the dashed vertical lines of matching color. In almost all combinations, the non pooled networks performed slightly better. However, the false positive rate for star classifications increased by only about 1.5% when using max pooling.

number of parameters. In future work, we seek to compare the efficacy of training a single larger network vs. multiple separate networks (for each detector) on a training set comprised of multiple detectors and filters.

Additionally, this model was only trained on cutouts where a source detection algorithm had initially found a local maximum, rather than locations randomly sampled across an image. Thus, performing model inference on a region of the image where there is not a peak can lead to nonsensical results. To test this behavior, a small region of an image in the training set was extracted, and cutouts centered on every pixel of the region were evaluated by the model. The resulting classification map is shown in Figure 11. As there is no class for empty sky, the model outputs the classification type for empty regions of the image as cosmic rays. The model also seems to flag some parts of cosmic rays as stars (though these are not the parts detected by peak finders), and regions near large cosmic rays as PSF halo features. In future work, experimentation with adding an extra class for empty sky may be added, to allow for understandable outputs from "scanning" the model over every pixel.

However, for source detection use cases this seems unlikely, as the background pixels would not be detected via local maximum finding algorithms.

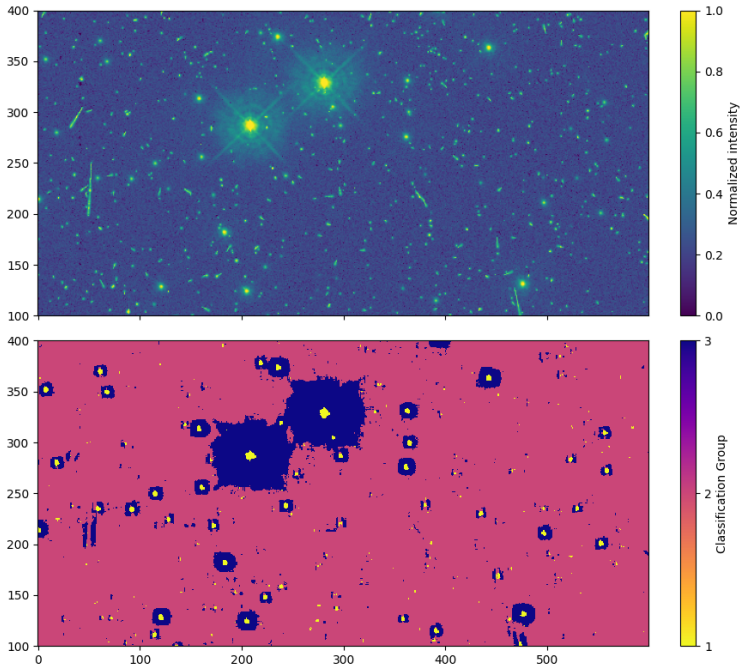

Figure 11: Result of "scanning" the model over all pixels in a small region of one of the training set images. The top panel shows the normalized image data, whereas the bottom shows the output class for a cutout centered at each pixel. Yellow indicates classification as a star, red for cosmic ray, and blue for diffraction spike/PSF halo feature. In regions of blank sky, the model appears to detect cosmic rays.

As the training data was taken from the outskirts of Ω Cen, a globular cluster, there were very few extended objects in the frame. Objects such as background galaxies, bright regions of nearby galaxies, or nebular emission can also have nonsensical outputs from the model. To demonstrate, the model was scanned over a small region of a WFC3/UVIS F555W FLC image of the LMC superbubble N44, which contains large scale structure of Hα and other gas emission. The resulting classification map is shown in Figure 12. The model appears to have generalized across filters, as it is able to correctly identify stars in the F555W filter as well. However, the emission of the hot gas is flagged as PSF halo structure, and seems to overwhelm fainter stars in the emission regions. It is likely that fainter foreground stars in front of other bright regions (such as nebulae or other galaxies, or highly crowded fields) would get flagged similarly. Future work will broaden the training dataset to contain stars superimposed onto extended structures, to allow the model to better identify these stars.

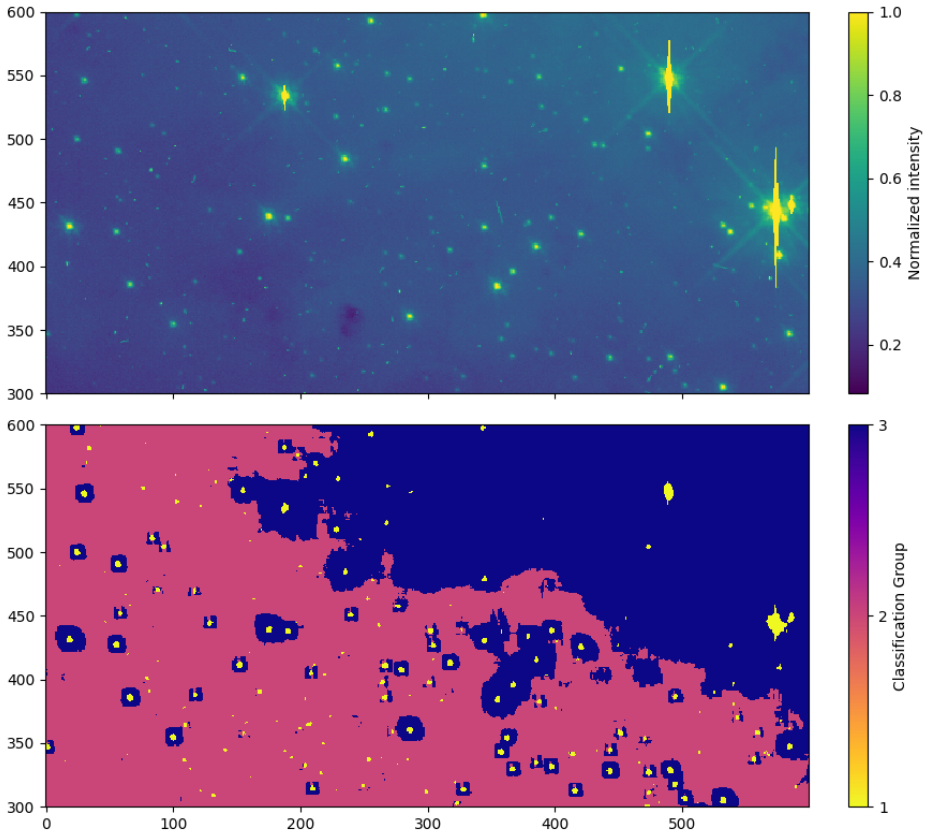

Figure 12: Result of "scanning" the model over all pixels in a small region of a F555W image of LMC N44. The top panel shows the normalized image data, whereas the bottom shows the output class for a cutout centered at each pixel. The meanings of the colors are the same as Figure 11. The extended Hα emission is flagged as a PSF halo feature, and overwhelms fainter stars surrounded by the emission.

6. Conclusions

Detection of stellar objects in images from space-based observatories is critical for many data analyses. Even when stars are not the primary objects of interest, they often serve as references for flux comparison, calibration, or image alignment. In this work, we present a workflow and results from training a convolutional neural network to classify three different types of objects commonly detected by peak-finding algorithms. This approach facilitates the removal of nonstellar objects, resulting in cleaner catalogs. Our procedure significantly reduces the time required for manual label generation by utilizing a UMAP transformation on data from a "cleaner" image frame (the drizzled image) and applying the labels to the native image frame. While supervised learning with a neural network was employed, the automated UMAP transformation was crucial for rapidly generating labels for thousands of cutouts and determining the appropriate number of classes needed for accurate classification. Leveraging known information about sample data from external sources, such as an external catalog or cleaned combined image, can lead to substantial efficiency gains in machine learning tasks.

Additionally, this document explores various architectural choices for neural networks

when the input data is relatively regular but small in terms of pixel dimensions. Although the network contains multiple hidden layers, making it a "deep" neural network, limiting the architecture to just a few layers significantly improves training time compared to using transfer learning with larger networks like VGG16. Using small neural networks to address specific tasks within larger workflows, rather than adopting an end-to-end solution, may yield better results with less time invested in model development and training.

Looking forward, we plan to enhance this model by expanding the training dataset to more broadly encompass WFC3 data. Though stellar source detection is often a critical step of WFC3/IR analysis, the lack of cosmic rays (due to CR detections and rejections in the up-the-ramp fitting) invalidates the need for a cosmic ray class. Thus the label generation procedure for WFC3/IR and other HgCdTe detector data (such as the near infrared detectors on JWST and the Nancy Grace Roman Space Telescope) would be distinct than those for CCDs.

We aim to further build upon this work by employing the model to assist in de-blending objects and developing a subsequent network for determining fluxes and positions of stellar objects. In the near future a Jupyter notebook will be provided demonstrating the process for creating a dataset, generating labels using UMAP and training a neural network for similar analyses.

Acknowledgments

We would like to thank M. Marinelli for their technical review of this document. We would also like to thank the HST Machine Learning group for their feedback and discussion during the exploration of this analysis. Additionally, we thank J. Green and S. Baggett of the WFC3 team for their editorial review of this document.

References

Ackermann, Sandro et al. (Sept. 2018). "Using transfer learning to detect galaxy mergers". In: 479.1, pp. 415–425. DOI: 10.1093/mnras/sty1398. arXiv: 1805.10289 [astro-ph.IM].

Anderson, J. (July 2022). *One-Pass HST Photometry with hst1pass*. WFC3 ISR 2022-05.

Bradley, Larry et al. (Oct. 2017). *astropy/photutils: v0.4*. DOI: 10.5281/zenodo.1039309. URL: https://doi.org/10.5281/zenodo.1039309.

Dauphin, F. et al. (Mar. 2022). *WFC3/UVIS Figure-8 Ghost Classification using Convolutional Neural Networks*. Instrument Science Report WFC3 2022-3, 41 pages.

Fruchter, A. S. and et al. (July 2010). "BetaDrizzle: A Redesign of the MultiDrizzle Package". In: *2010 Space Telescope Science Institute Calibration Workshop*, pp. 382–387.

George, Daniel, Hongyu Shen, and E. A. Huerta (May 2018). "Classification and unsupervised clustering of LIGO data with Deep Transfer Learning". In: 97.10, 101501, p. 101501. DOI: 10.1103/PhysRevD.97.101501. arXiv: 1706.07446 [gr-qc].

Jones, M. and F. Dauphin (Apr. 2024). *WFC3/UVIS Guide Star Failure Classification with Machine Learning*. Instrument Science Report WFC3 2024-03, 28 pages.

Kingma, Diederik P. and Jimmy Ba (2017). *Adam: A Method for Stochastic Optimization*. arXiv: 1412.6980 [cs.LG]. URL: https://arxiv.org/abs/1412.6980.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (May 2015). "Deep learning". In: 521.7553, pp. 436–444. DOI: 10.1038/nature14539.

LeNail, Alexander (2019). "NN-SVG: Publication-Ready Neural Network Architecture Schematics". In: *Journal of Open Source Software* 4.33, p. 747. DOI: 10.21105/joss.00747. URL: https://doi.org/10.21105/joss.00747.

McInnes, Leland, John Healy, and James Melville (2020). *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. arXiv: 1802.03426 [stat.ML]. URL: https://arxiv.org/abs/1802.03426.

Pan, Sinno Jialin and Qiang Yang (2010). "A Survey on Transfer Learning". In: *IEEE Transactions on Knowledge and Data Engineering* 22.10, pp. 1345–1359. DOI: 10.1109/TKDE.2009.191.

Simonyan, Karen and Andrew Zisserman (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv: 1409.1556 [cs.CV]. URL: https://arxiv.org/abs/1409.1556.

Stetson, Peter B. (Mar. 1987). "DAOPHOT: A Computer Program for Crowded-Field Stellar Photometry". In: 99, p. 191. DOI: 10.1086/131977.