



















**OPEN ACCESS**

Pre-training Vision Models for the Classification of Alerts from Wide-field Time-domain Surveys

Nabeel Rehemtulla^{1,2,3} , Adam A. Miller^{1,2,3} , Mike Walmsley⁴ , Ved G. Shah^{1,2,3} , Theophile Jegou du Laz^{5,6} ,
 Michael W. Coughlin^{6,7} , Argyro Sasli^{6,7} , Joshua Bloom^{8,9} , Christoffer Fremling^{5,10} , Matthew J. Graham⁵ ,
 Steven L. Groom¹¹ , David Hale¹⁰ , Ashish A. Mahabal^{5,12} , Daniel A. Perley¹³ , Josiah Purdum¹⁰ , Ben Rusholme¹¹ ,
 Jesper Sollerman¹⁴ , and Mansi M. Kasliwal⁵ 

¹ Department of Physics and Astronomy, Northwestern University, 2145 Sheridan Road, Evanston, IL 60208, USA; nabeelr@u.northwestern.edu

² Center for Interdisciplinary Exploration and Research in Astrophysics (CIERA), 1800 Sherman Ave., Evanston, IL 60201, USA

³ NSF—Simons AI Institute for the Sky (SkAI), 172 E. Chestnut St., Chicago, IL 60611, USA

⁴ Dunlap Institute for Astronomy & Astrophysics, University of Toronto, Toronto, ON M5S 3H4, Canada

⁵ Division of Physics, Mathematics, and Astronomy, California Institute of Technology, Pasadena, CA 91125, USA

⁶ NSF Institute on Accelerated AI Algorithms for Data-Driven Discovery (A3D3), USA

⁷ School of Physics and Astronomy, University of Minnesota, Minneapolis, MN 55455, USA

⁸ Physics Division, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA

⁹ Department of Astrophysics, University of California, Berkeley, CA 94720-3411, USA

¹⁰ Caltech Optical Observatories, California Institute of Technology, Pasadena, CA 91125, USA

¹¹ IPAC, California Institute of Technology, 1200 E. California Blvd, Pasadena, CA 91125, USA

¹² Center for Data Driven Discovery, California Institute of Technology, Pasadena, CA 91125, USA

¹³ Astrophysics Research Institute, Liverpool John Moores University, 146 Brownlow Hill, Liverpool L3 5RF, UK

¹⁴ Department of Astronomy, The Oskar Klein Center, Stockholm University, AlbaNova, SE-10691 Stockholm, Sweden

Received 2025 December 14; revised 2026 March 8; accepted 2026 March 11; published 2026 March 27

Abstract


Modern wide-field time-domain surveys facilitate the study of transient, variable and moving phenomena by conducting image differencing and relaying alerts to their communities. Machine learning tools have been used on data from these surveys and their precursors for more than a decade, and convolutional neural networks (CNNs), which make predictions directly from input images, saw particularly broad adoption through the 2010s. Since then, continually rapid advances in computer vision have transformed the standard practices around using such models. It is now commonplace to use standardized architectures pre-trained on large corpora of everyday images (e.g., ImageNet). In contrast, time-domain astronomy studies still typically design custom CNN architectures and train them from scratch. Here, we explore the effects of adopting various pre-training regimens and standardized model architectures on the performance of alert classification. We find that the resulting models match or outperform a custom, specialized CNN like what is typically used for filtering alerts. Moreover, our results show that pre-training on galaxy images from Galaxy Zoo tends to yield better performance than pre-training on ImageNet or training from scratch. We observe that the design of standardized architectures are much better optimized than the custom CNN baseline, requiring significantly less time and memory for inference despite having more trainable parameters. On the eve of the Legacy Survey of Space and Time and other image-differencing surveys, these findings advocate for a paradigm shift in the creation of vision models for alerts, demonstrating that greater performance and efficiency, in time and in data, can be achieved by adopting the latest practices from the computer vision field.

Unified Astronomy Thesaurus concepts: [Convolutional neural networks \(1938\)](#); [Time domain astronomy \(2109\)](#); [Sky surveys \(1464\)](#)

1. Introduction

Large, modern, wide-field time-domain surveys like the Zwicky Transient Facility (ZTF; Bellm et al. 2019a, 2019b;

Graham et al. 2019; Masci et al. 2019; Dekany et al. 2020), the Asteroid Terrestrial-impact Last-Alert System (Tonry 2011; Tonry et al. 2018; Smith et al. 2020), the Panoramic Survey Telescope and Rapid Response System (Pan-STARRS; Kaiser et al. 2002) and the Vera C. Rubin Observatory’s Legacy Survey of Space and Time (LSST; Ivezić et al. 2019) employ image differencing to identify new, moving, and varying sources in real-time. Image differencing involves subtracting a

 Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](#). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

reference image from a new science image and searching for statistically significant source detections in the resulting difference image. These detections create alerts which are promptly relayed to the community via an alert stream (e.g., Patterson et al. 2019) through alert brokers (e.g., Duev et al. 2019; Förster et al. 2021; Matheson et al. 2021; Möller et al. 2021; Williams et al. 2024; Jegou du Laz et al. 2025), and they typically include image cutouts of the new, reference, and difference images centered on the source location. Alert streams are fertile ground for studying an array of astrophysical phenomena, including variable stars (VarStars), galactic and extragalactic transients, active galactic nuclei (AGN), and asteroids. Consequently, significant filtering of an alert stream is required to isolate events relevant to specific scientific cases. See Rehemtulla et al. (2025a) for an overview of the current state of these workflows and a projection for how they will evolve in the Rubin era.

There exists a rich legacy of machine learning (ML) classifiers enabling time-domain astronomy by assisting with filtering alert streams. Early work adopted models such as support vector machines and tree-based classifiers (e.g., Bailey et al. 2007; Bloom et al. 2012; Brink et al. 2013) which produced classifications using descriptive statistical features extracted from the data. Later, the extraordinarily rapid development of computer vision techniques spurred the adoption of convolutional neural networks (CNNs; Fukushima & Miyake 1982; LeCun et al. 2015) which eliminate the need for manually designing statistical features to extract from the data by processing image cutouts directly and often yielding fruitful models (e.g., Cabrera-Vives et al. 2017; Gieseke et al. 2017; Sedaghat & Mahabal 2018; Duev et al. 2019, 2021; Carrasco-Davis et al. 2021; Duev & van der Walt 2021; Rehemtulla et al. 2024; Sheng et al. 2024; Weston et al. 2024).

Continued rapid progress in computer vision, supported by large, labeled, diverse and publicly available datasets provided the scale and breadth necessary for training modern vision models. MNIST (Deng 2012) and CIFAR (Krizhevsky & Hinton 2009) standardized evaluation, while the introduction of large-scale datasets like ImageNet (Russakovsky et al. 2014), COCO (Lin et al. 2014) and Places (Zhou et al. 2017) provided data diversity through immense scale. The standard practices around creating computer vision models have evolved with time, in part to take advantage of the availability of these datasets. It is now commonplace in computer vision studies to adopt pre-trained models with standardized, or “off-the-shelf,” architectures for practically any task, often irrespective of whether or not the pre-training dataset is closely related to the task at hand. Pre-training is the process of training a model on a large, generic dataset to learn general feature extraction capabilities before conducting a task-specific training routine (“fine-tuning”). Pre-training confers benefits including faster convergence, stronger generalization, and

lower data requirements (Girshick et al. 2013; He et al. 2021). Alongside this, off-the-shelf architectures offer well-tested baselines with proven design choices, allowing researchers to avoid time-consuming low-level model design.

Some time-domain astronomy studies creating vision models follow these practices and have deployed their products into production workflows (Weston et al. 2024; Junell et al. 2025) but most still train custom CNNs from scratch. Those that do follow modern practices, e.g., AppleCiDER (Junell et al. 2025), also tend to include analyses demonstrating the superior performance and/or computational efficiency offered by pre-trained off-the-shelf vision models. Stoppa et al. (2025) take an entirely different approach and explore real-bogus classification of alert cutouts with a multi-modal large language model (LLM). The integration of an LLM allows for the model to provide textual explanations of its classifications, although the considerable computational cost of these models is not conducive to real-time deployment. Studies in other areas of astronomy have more broadly adopted pre-training and off-the-shelf architectures for their vision models (e.g., Wu & Boada 2019; Walmsley et al. 2022; Sadr et al. 2023; Mohale & Lochner 2024). Researchers creating models to operate on time-series data have also taken a special interest in pre-training, in part to overcome small training sets typically with large class imbalances (Fiscale et al. 2021; Zhang et al. 2024; Gupta et al. 2025a, 2025b). These works have demonstrated that pre-training and off-the-shelf architectures can unlock models that yield better performance with less, or even zero (Li et al. 2025), task-specific training data.

In this work, we explore a wide breadth of design choices faced when creating vision models for alert classification. Our experiments measure classification performance on a transient candidate vetting task with two pre-training regimens (Section 2.2) compared against training from scratch, two standardized model architectures with orthogonal philosophies (Section 2.1), and two systematically different fine-tuning datasets (Section 2.3). These experiments are also repeated across an order of magnitude difference in training set size to reveal trends as a function of the quantity of labeled data available for training. Although custom CNNs trained from scratch have served time-domain astronomers well for years, the growth of our data quantity and variety in the Rubin-era will challenge this paradigm because it is expected to far outpace the growth of data labeling. Such an investigation is particularly timely as there are numerous image-differencing facilities which have recently—or will soon—become operational: LSST, the BlackGEM Telescope Array (Groot et al. 2024), the Nancy Grace Roman Space Telescope (Akeson et al. 2019), the La Silla Schmidt Southern Survey (LS4; Miller et al. 2025), the Argus Optical Array (Law et al. 2022), and others. We conclude with a review of our findings which serve as a set of recommendations for most efficiently creating

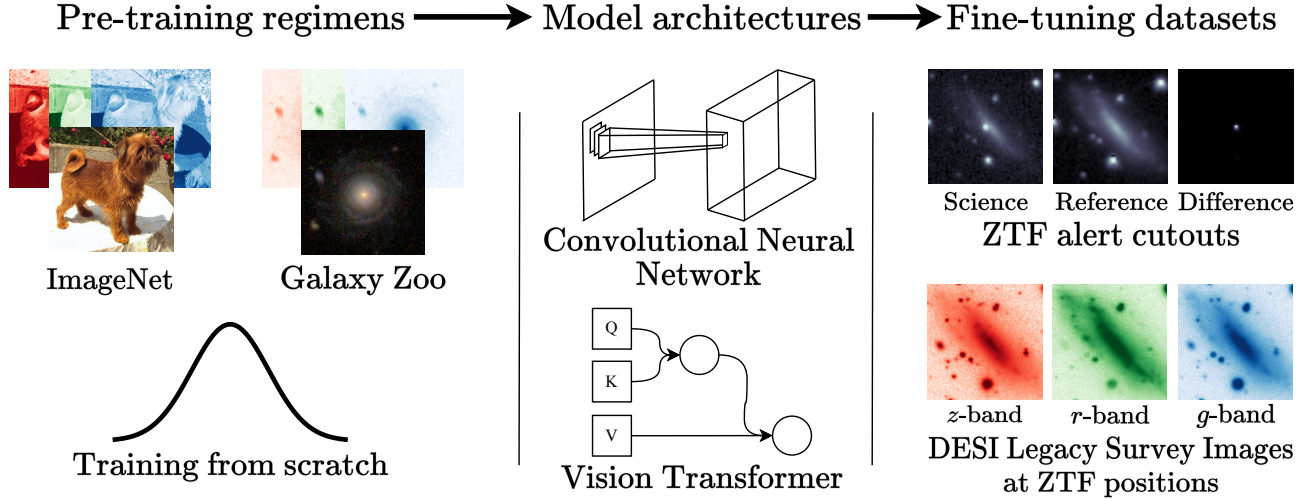


Figure 1. Schematic of our experimental design. We evaluate model performance for all permutations of the depicted pre-training regimens, model architectures, fine-tuning datasets, as well as four settings of fine-tuning dataset size.

vision models to be deployed into the alert streams of these current and upcoming image-differencing surveys.

2. Experimental Design

As shown in Figure 1, we explore vision model design by evaluating models using various off-the-shelf architectures (Section 2.1), pre-training regimens (Section 2.2) and fine-tuning datasets on a transient candidate vetting task (Section 2.3). We train models with modern optimization techniques and hyperparameter sweeps (Section 2.4) and present results as a function of training set size.

2.1. Choice of off-the-shelf Architectures

We select the ConvNeXt (Liu et al. 2022) and MaxViT (Tu et al. 2022) architectures to evaluate against a small, custom, specialized CNN like those often used in the time-domain astronomy literature. This choice reflects the aim to compare two dominant classes of modern but fundamentally different vision models: fully convolutional models (i.e., those without attention mechanisms) and vision transformers (ViT; Dosovitskiy et al. 2020), i.e., those predominantly driven by attention mechanisms.

ConvNeXt (Liu et al. 2022) represents a modern reinterpretation of CNNs by revisiting the ResNet architecture (He et al. 2016) and borrowing insights from recent transformer architectures (Vaswani et al. 2017) as well as other highly efficient CNNs (Chollet 2016; Howard et al. 2017). In general, CNNs apply small, sliding filters across an image to detect local patterns, textures, or small-scale structures. Numerous studies have shown that CNNs can very effectively capture real astronomical features like galaxy morphologies (e.g., Bhambra et al. 2022; Alfonzo et al. 2024). The ConvNeXt architecture preserves the strong inductive bias toward

translational invariance and feature locality inherent to CNNs while incorporating architectural refinements such as large kernel sizes and improved normalization layers. (Liu et al. 2022) present detailed ablation studies and interpretation to why certain choices improve performance. Overall, these updates allow ConvNeXt models to achieve performance competitive with ViTs on large-scale image classification benchmarks, while retaining the stability and computational efficiency characteristic of CNNs (Liu et al. 2022).

MaxViT (Tu et al. 2022) represents a refinement of the original ViT design. ViTs use attention mechanisms to learn relationships across an entire image and to prioritize regions of particular interest. While the CNN’s convolution operations only consider a region the size of a single kernel at a time, the ViT’s self-attention mechanism allows for each input element to “attend” to each other element. In practice, these input elements are tokens which originate from patches of pixels and are then fed to the attention heads. This balances the substantial computational cost of the attention layers while maintaining its flexibility and generality. Stemming from the immense representation capabilities of ViTs, they have become established as a highly performant tool across a breadth of domains (Khan et al. 2022). The MaxViT architecture introduces a “multi-axis” attention mechanism which alternates between local and global self-attention, allowing the model to capture relationships across large and small scales without a steep increase in computational cost.

Together, ConvNeXt and MaxViT represent complementary points along the inductive bias spectrum, enabling an assessment of whether convolutions or attention mechanisms yield superior performance and efficiency.

Our model architecture choices are affirmed by the large architecture benchmarking experiment presented by Walmsley et al. (2024). In their study, performance on a galaxy

morphology task is measured for various modern architectures, model sizes, and training dataset sizes. They find that ConvNeXt and MaxViT perform the best overall, and that the larger, higher parameter count versions of the ConvNeXt and MaxViT models tend to only outperform their smaller analogs when accompanied by significantly larger quantities of training data. Given the limited size ($N \approx 25,000$ total sources) of our training set (Section 2.3), we select the smallest available versions of the ConvNeXt and MaxViT architectures: ConveNeXt-pico (8.5M trainable parameters) and MaxViT-tiny (28.5M trainable parameters).¹⁵

When training any model, its head is replaced by a randomly initialized multi-layer perceptron (MLP) with three layers whose hyperparameters are optimized with the procedure described in Section 2.4. During training, all weights and biases in the vision backbone and the MLP head are unfrozen and allowed to be updated.

2.2. Choice of Pre-training Regimens

We select two pre-training regimens to compare against training from scratch: pre-training on ImageNet (Russakovsky et al. 2014) and pre-training on Galaxy Zoo (Lintott et al. 2008, 2011). Comparing results across these configurations allows us to assess the effect of pre-training relative to training from scratch and the importance of pre-training on real astronomical observations. In other terms, we hope to assess whether pre-training on real astronomical observations (Galaxy Zoo) yields better results than naïve pre-training on entirely out-of-domain non-astronomy data (ImageNet) and if pre-training on out-of-domain data provides anything useful relative to training from scratch. These goals are deliberately oriented towards providing time domain computer vision practitioners with useful guidance when creating models for use with alert cutouts.

ImageNet (Russakovsky et al. 2014) is a large, labeled dataset containing images of everyday objects, natural scenes, animals, etc. while excluding astronomical observations. In this study we adopt the ImageNet-1k subset which contains ~ 1.28 million images across 1000 classes. We select it here because it is a literature-standard choice for benchmarking vision models. We obtain the ConvNeXt and MaxViT models pre-trained on ImageNet-1k from the PyTorch Image Models library (timm; Wightman 2019).

Galaxy Zoo (Lintott et al. 2008, 2011) is a citizen science project where volunteers classify galaxies in real astronomical images based on their morphologies. The Galaxy Zoo project began with volunteers annotating images from the Sloan Digital Sky Survey (York et al. 2000) and has since expanded to images from the Hubble Space Telescope, the Dark Energy Spectroscopic Instrument’s (DESI’s) Legacy Survey (DESI

Collaboration et al. 2016; Dey et al. 2019), the UKIRT Infrared Deep Sky Survey (UKIDSS; Lawrence et al. 2007), and more. Over nearly two decades, >100 thousand Galaxy Zoo volunteers have contributed >100 million total annotations. In this study we use ZOOBOT models pre-trained by Walmsley et al. (2024) on $\sim 842,000$ annotated Galaxy Zoo images from five observatories. The ZOOBOT models learn galaxy morphology by being trained to predict the volunteers’ answers to questions posed by Galaxy Zoo (see Walmsley et al. 2024 for details). We obtain the models pre-trained by Walmsley et al. (2024) from the Hugging Face platform.¹⁶

We compare these two pre-training regimens against training from scratch, which involves drawing values for the model’s trainable parameters from distributions carefully designed to stably and quickly being the training process. In all cases, we adopt the default initialization schemes used in PyTorch (Paszke et al. 2019), which make use of findings from studies exploring initialization strategies (e.g., Glorot & Bengio 2010; He et al. 2015). Models without pre-training are obtained from timm.

2.3. Choice of Training Set and Task

We adopt the BTSBOT (Rehemtulla et al. 2024) training set and transient candidate vetting task because they provide high quality labels on a very large training set with representation across a wide range of source types. BTSBOT originates from the ZTF Bright Transient Survey (BTS; Fremling et al. 2020; Perley et al. 2020; Rehemtulla et al. 2024), an effort aiming to spectroscopically classify all bright ($m_{\text{peak}} \leq 18.5$ mag in g_{ZTF} or r_{ZTF}) extragalactic transients in the ZTF public alert stream. BTS members conduct daily visual inspection of bright transient candidates and send promising events to the SED Machine (Blagorodnova et al. 2018; Rigault et al. 2019; Kim et al. 2022) and other spectrographs for classification. Over its ~ 7 yr lifetime, BTS has produced an immense dataset with high quality binary labels. Still, there is inevitably non-zero label noise. In Section 5, we do identify two mislabeled sources, but we expect that the level of label noise is not significant enough to have a substantial detrimental effect on the performance of models we train here.

Rehemtulla et al. (2024) compiled sources selected and rejected from the BTS sample to train BTSBOT to automate bright transient candidate vetting. This task involves considering a wide breadth of sources including VarStars, AGN, asteroids, galactic and extragalactic transients, and even alerts produced by artifacts of the image differencing process (“bogus” alerts). The comprehensive nature of this training set and task make it an appropriate choice for representatively probing the effect of pre-training on alert classification models. Moreover, the BTSBOT training set is significantly larger than

¹⁵ Number of trainable parameters are counted only from the model’s vision backbone and exclude any trainable parameters in the model’s MLP head.

¹⁶ <https://huggingface.co/mwalmsley>

that of other comparable, published models, allowing us to experiment with pre-training and model architectures at greater scales. We rerun the training set compilation and pre-processing described by Rehemtulla et al. (2024) which makes use of ZTF’s first-party alert broker (Kowalski; Duev et al. 2019) and marshal (SkyPortal/Fritz; van der Walt et al. 2019; Coughlin et al. 2023) to update the `BTSbot` training set originally queried in 2023 August. The updated training set contains 769,056 alerts across 25,609 sources: $\sim 30\%$ larger than the original.

The ZTF alert cutouts in this dataset utilize a significantly different arrangement than those in the pre-training datasets. Both ImageNet and Galaxy Zoo have three-channel red-green-blue images while the three channels of the ZTF alert cutouts represent the science, reference and difference images (see right side of Figure 1). The red-green-blue images naturally stack into a color composite image where the relative pixel values of a given region across channels conveys additional astrophysical information, e.g., bluer parts of a galaxy tend to be populated with young, massive, hot stars and thus are associated with recent star formation. Alert cutouts produced by image differencing do not share this property; instead, the difference image (third channel) is a linear combination of the science and reference images (first and second channels). Instead of color information, this schema offers information on the quality of the subtraction and the morphology of the source’s PSF. Thus, the cross-channel correlations between these two image schema are dramatically different. This introduces a distribution shift between the pre-training and the fine-tuning data. In order to probe the consequence of this difference, we introduce an additional fine-tuning dataset consisting of DESI Legacy Survey data release 10 (DR10) *grz* color images at the same field-of-view ($63'' \times 63''$) and pixel-scale ($1'' \text{ pixel}^{-1}$) as the ZTF alert cutouts.¹⁷ Thus, this Legacy Survey dataset contains the same environmental contextual information critical for transient candidate vetting while minimizing the distribution shift between pre-training and fine-tuning. Legacy Survey also provides greater depth and introduces color information, both of which are absent in the ZTF alert cutouts. A significant subset of alerts ($\sim 22\%$) are omitted from this fine-tuning dataset because there is no Legacy Survey DR10 coverage at their position. The omitted alerts predominantly occur at low galactic latitudes, so certain phenomena, like cataclysmic variables, are disproportionately removed relative to others. Thus, an absolute comparison of model performance metrics between the two fine-tuning datasets is flawed but comparing the relative performance differences between pre-training regimens can aid with better understanding the distribution shift between the ZTF alert cutouts and the pre-training datasets.

¹⁷ This is not the native pixel-scale of Legacy Survey images, however, we bin the images to this pixel-scale to match that of the ZTF alert cutouts.

We also assess the data efficiency of each model configuration by repeating experiments for four settings of training set size. We create smaller versions of the train split by randomly selecting 5, 10, and 50% of alerts to preserve while removing all others. The models trained on the smaller datasets still use the full validation and test splits to ensure performance metrics are representative of the model’s true performance.

Both the ConvNeXt and MaxViT models we select take in images much larger than that of the image cutouts from either fine-tuning dataset. To handle this, we up-sample our input images to the model’s expected resolution while maintaining smooth source profiles with bilinear pixel interpolation.

2.4. Training and Hyperparameter Optimization

Models are trained with an updated version of the `BTSbot` training code¹⁸ rewritten to use the PyTorch API (Paszke et al. 2019). Most aspects of training are kept consistent with that of `BTSbot`, e.g., the binary cross entropy loss function, data augmentation techniques and class weighting (Rehemtulla et al. 2024). Some select settings are updated to more modern practices; we adopt the AdamW optimizer (Loshchilov & Hutter 2017) and a learning rate scheduler which contains a linear warm-up followed by a cosine decay (Loshchilov & Hutter 2016). The best performing optimizer and model hyperparameters can vary for each selection of pre-training regimen, architecture, and fine-tuning dataset, so we execute a separate Bayesian hyperparameter sweep for each configuration to ensure they are fairly assessed. Table 1 shows the hyperparameters ranges over which we search. Sweeps are conducted using the Weights and Biases platform (Biewald 2020), and we find that < 60 trials per sweep is typically sufficient to converge to a near-optimal set of hyperparameters.¹⁹ Once a sweep is completed, we use the hyperparameters of the best performing trial to run five final trials with the optimized hyperparameters but with varied random seeds. The performance metrics we report are the median and standard deviation for these final trials computed on the test split.

We primarily focus on models trained only with images to distill the effect of the updated vision model design, but we also explore multi-modal models to serve as a direct replacement to the production `BTSbot` model. Models are made multi-modal by introducing a metadata branch whose embedding is concatenated with that of the image branch before being passed to an MLP head. The metadata features are taken exactly from that of the production `BTSbot` model.

Lastly, we also re-train the uni-modal and multi-modal versions of the custom CNN used by `BTSbot` to serve as a

¹⁸ <https://github.com/nabeelre/BTSbot>

¹⁹ We do identify a small number of cases where this number of trials is insufficient and the measured performance slightly diverts from trends observed in related sweeps (see Sections 3, 4).

Table 1
Hyperparameter Optimization

| Hyperparameter Name | Sweep Search Range |
|----------------------------|----------------------------------------------------|
| Optimizer | |
| Learning rate (α) | $\{1, 5\} \times 10^{-6}, \{1, 5\} \times 10^{-7}$ |
| Warm-up Epochs | $\{2, 4, 6\}$ epochs |
| Batch size | $\{16, 64, 256\}$ |
| Standardized MLP head | |
| Dense layer 1 | $\{128, 256, 512\}$ units |
| Dropout layer | 0.3–0.9 |
| Dense layer 2 | $\{8, 16, 32\}$ units |

comparison baseline. Many other models performing alert classification use similar architectures (e.g., Andreoni et al. 2017; Gieseke et al. 2017; Duev et al. 2019, 2021; Carrasco-Davis et al. 2021; Duev & van der Walt 2021; Makhoulf et al. 2022; Acero-Cuellar et al. 2023; Sheng et al. 2024; Weston et al. 2024; Liu et al. 2025; Pranshu et al. 2025), so this baseline represents the most common approach in the literature. Similarly to the off-the-shelf architectures, we execute hyperparameter sweeps using the search ranges described in Table 1 and in Rehemtulla et al. (2024).

3. Results from Single-modality Models

We present results of models benchmarked on bright transient candidate vetting. Model performance is evaluated using the area under the receiver operating characteristic curve (ROC AUC), a general-purpose indicator of classification performance that accounts for the mild class imbalance present in the training set. Higher values of ROC AUC indicate better classification performance, reaching unity for perfect classification.

Figure 2 reports ROC AUC as a function of the number of images trained on. For the ZTF alerts dataset, we find that the pre-trained ConvNeXt models yield performance consistent within uncertainties with the baseline custom CNN across all training set sizes. The ConvNeXt models trained from scratch, however, slightly underperform their pre-trained analogs and the baseline CNN in all tests: they yield ROC AUC score up to ~ 0.02 lower than the baseline. The MaxViT models trained from scratch closely trace the performance of the baseline, and the models pre-trained on ImageNet show a slight boost in performance relative to those without pre-training. The Galaxy Zoo MaxViT models yield the best performance across all training set sizes, reaching a maximum of ~ 0.025 higher ROC AUC than the baseline. Their improvement relative to ImageNet pre-training (a median of ~ 0.01 in ROC AUC) suggests that some aspect of the galaxy images promoted transfer to the

ZTF alert cutouts more readily than the seemingly unrelated content in the ImageNet images.

We also experiment with fine-tuning the off-the-shelf architectures on Legacy Survey cutouts to test the differential effect of pre-training on their red-green-blue (*zrg* bands) layout versus the science-reference-difference layout of the ZTF alert cutouts. The right panels of Figure 2 shows that the ConvNeXt models fine-tuned on Legacy Survey images unambiguously illustrate a strong advantage toward the pre-trained models relative to training from scratch. In these experiments, the relative performance gain from pre-training ($\gtrsim 0.04$) is much larger than in the experiments fine-tuning on ZTF alert cutouts ($\lesssim 0.01$). We posit that this is a product of the reduced data shift between the fine-tuning and pre-training datasets. Similarly to their analogs fine-tuned on ZTF images, the MaxViT models fine-tuned on Legacy Survey images show that pre-training on Galaxy Zoo yields a clear advantage relative to pre-training on ImageNet and training from scratch. As in the ZTF alert cutouts test, the MaxViT model pre-trained on Galaxy Zoo yields the best overall performance. We note that the performance of the Legacy Survey models is systematically worse than that of the models operating on ZTF alert cutouts. We do not draw any conclusions from this result as it does not represent a level comparison (see Section 2.3).

4. Results from Multi-modal Models

We additionally present multi-modal models benchmarked on bright transient candidate vetting to assess if an alternative vision model can improve the in-production performance of `BTSbot`. The primary task of `BTSbot` in production is to identify and allocate spectroscopic follow-up resources to classify new bright transients and infant supernovae. Rehemtulla et al. (2024, 2025b) measured model performance on this task with the completeness and purity of triggers, reflecting the importance of recovering all transients of interest without triggering follow-up for other sources. Maximizing general classification performance was the goal of Section 3 which led us to focus on ROC AUC, but here our focus is on in-production performance, so we adopt the F1 score of triggers sent. An F1 score is the harmonic mean of the purity and completeness, so it summarizes the performance of both priorities in a single metric. Table 6 of Rehemtulla et al. (2024) demonstrated that generic metrics like ROC AUC are not always directly correlated with in-production metrics, so by adopting this custom metric, we can most realistically assess the performance of a model deployed in production for automatically allocating follow-up resources. Rehemtulla et al. (2024) introduces the concept of a *policy*, a set of criteria which are used to determine

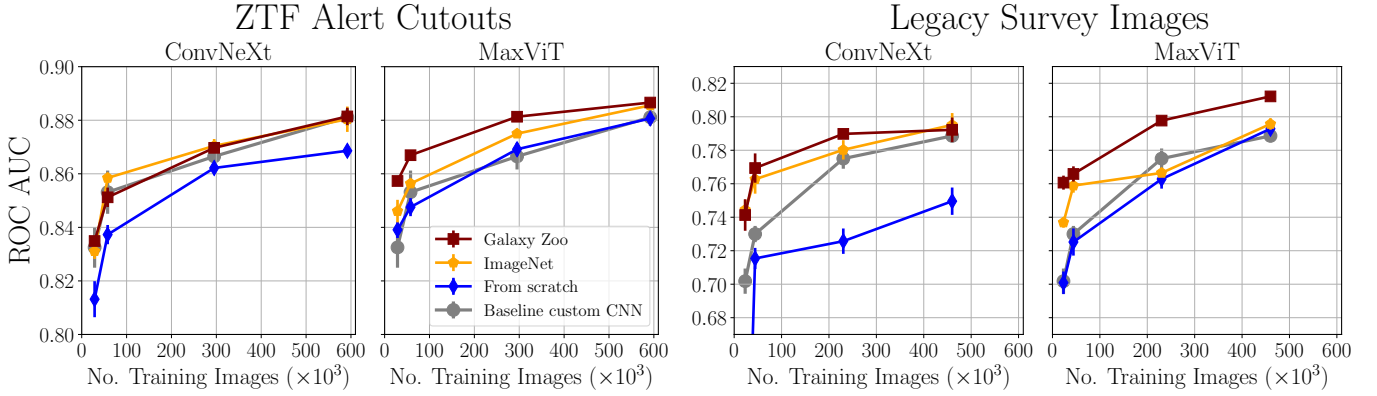


Figure 2. Area under the receiver operating characteristic curve (ROC AUC) for experiments varying the model’s pre-training regimen, architecture, and fine-tuning dataset. Comparisons are made against a custom architecture CNN (gray) alike those often adopted in the astronomy literature. Left: Models fine-tuned on ZTF alert cutouts. Across both the ConvNeXt and MaxViT architectures, pre-training on Galaxy Zoo (maroon) or ImageNet (orange) yields better performance than training from scratch (blue). The Galaxy Zoo MaxViT outperforms the baseline and the ImageNet MaxViT at all fine-tuning dataset sizes. Right: Models fine-tuned on Legacy Survey images. Pre-trained models usually significantly outperform their counterparts without pre-training. Again, the Galaxy Zoo MaxViT model surpasses the performance of all others.

when a request for follow-up is sent. Here, we use the F1 score of the `bts_p1` policy which requires a source to have at least two alerts with `BTSbot` score ≥ 0.5 and at least one alert with `magpsf` ≤ 19 mag.²⁰

We also do not present multi-modal results using the Legacy Survey images because they do not represent a realistic production scenario: the mismatched sky coverage of ZTF and Legacy Survey would prevent classifications for many alerts (see Section 2.3), resulting in very poor completeness. Image cutouts from Pan-STARRS could be substituted in place of Legacy Survey to better match the sky coverage of ZTF, however, we do also observe substantially worse maximum performance from Legacy Survey cutouts than from ZTF alert cutouts so we do not explore this option. We do note, however, that this result can vary based on the task chosen to benchmark on.

Figure 3 reports the F1 score for triggering follow-up as a function of the number of alerts trained on. We find that, for both off-the-shelf architectures, the baseline yields better performance at small training set sizes ($\lesssim 100,000$ alerts). For larger training sets sizes, the off-the-shelf architectures mostly match or surpass the performance of the baseline multi-modal CNN. The baseline ceases to improve from $\sim 300,000$ to $\sim 600,000$ total training alerts. We hypothesize this is the case because the small image branch does not have a sufficient learning capacity to take advantage of the increased training set size. This interpretation is consistent with the steadily

²⁰ We note that slightly ($\sim 1\%$) higher F1 scores can be achieved by increasing the score threshold from 0.5 to ~ 0.8 . As such, a policy using a higher score threshold is deployed in one of the real `BTSbot` production workflows. This gain is small, however, so we adopt the original `bts_p1` policy here to maintain a direct comparison with the results presented in Rehemtulla et al. (2024) without detracting from the relevance to production workflows.

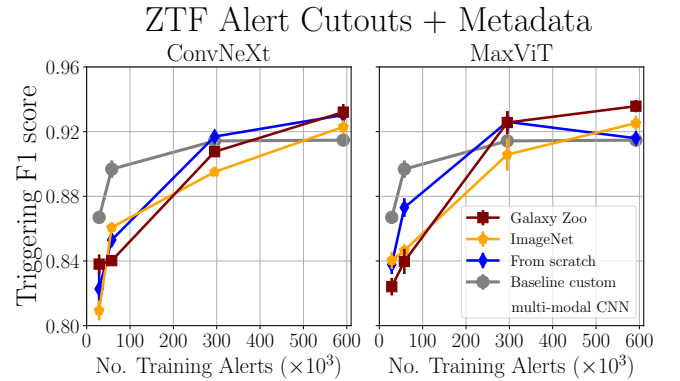


Figure 3. F1 scores representing how well multi-modal architectures with varying pre-training regimens can identify bright transient candidates for automatic spectroscopic follow-up compared against the baseline production `BTSbot` model (gray). At small training set sizes ($< 100 \times 10^3$ alerts), the baseline production model outperforms the off-the-shelf architectures across all pre-training options (Galaxy Zoo, maroon; ImageNet, orange; no pre-training, blue). As training set size increases ($> 300 \times 10^3$ alerts), off-the-shelf architectures begin to outperform the baseline custom CNN, likely a product of their greater learning capacity. Pre-training on Galaxy Zoo unlocks the greatest overall performance across both the ConvNeXt and MaxViT architectures.

positive sloped performance of the larger, off-the-shelf architectures.²¹

The performance trends across pre-training regimens for multi-modal models are generally not as clear as those of the models trained with images only (Section 3), but we do find that the relative performance difference between pre-trained models and models trained from scratch is clearly decreased.

²¹ The MaxViT model trained from scratch on $\sim 600,000$ alerts is an exception to this trend. This is likely a symptom of its hyperparameter sweep not being broad enough or run for long enough to identify sufficiently performant hyperparameters (see Section 2.4).

We attribute this to the overall reduced importance of the vision branch to this multi-modal prediction. Rehemtulla et al. (2024) show that a model performing this task with the metadata features alone yields comparable performance to the multi-modal model on similar in-production metrics. Still, we do find that the models with vision branches pre-trained on Galaxy Zoo generally outperform those pre-trained with ImageNet. This again demonstrates that pre-training on Galaxy Zoo is more conducive to later fine-tuning on ZTF alerts than pre-training on ImageNet.

5. Further Model Comparisons

Although performance metrics such as ROC AUC and our tailored in-production metrics remain essential for evaluating classification quality, they provide only a partial view of how models differ from one another. These metrics neglect the computational costs and can obscure more nuanced information on what the model has learned. In practice, two models with nearly identical ROC curves may impose vastly different computational burdens, exhibit distinct representational structures in their latent spaces, or behave differently under distribution shifts. For these reasons, we complement traditional performance metrics with analyses of computational efficiency (inference runtime and memory demands) and qualitative comparisons of the models’ learned feature spaces via uniform manifold approximation and projection (UMAP; McInnes et al. 2018) embeddings. These additional analyses offer a more holistic view of their trade-offs and practical suitability for real-time production environments of large image-differencing surveys.

5.1. Computational Cost

To describe a model’s computational cost, we evaluate the speed at which a model runs inference and the memory requirements to do so. These are critical considerations as the computer systems running survey pipelines have limited computational resources and, in certain scenarios, models can bottleneck the speed at which alerts are relayed to the community, adversely affecting time-sensitive science cases (e.g., multi-messenger astronomy). Furthermore, vision models are typically trained and designed to run on graphics processing units (GPUs) which very efficiently handle image inputs. However, most survey computer systems are not equipped with GPUs and are thus forced to run these vision models on central processing units (CPUs). We report runtime measurements as the image throughput for each model architecture running on both a GPU and a CPU. Throughput is the number of alerts in the test split divided by the time it takes to compute predictions for each of them using a batch size of 32 and 32-bit floating point precision. The GPU tests use an NVIDIA A30 GPU and the CPU tests use 12 cores of an Intel Xeon Gold 6338 CPU. The memory usage is measured as

Table 2
Computational Cost of Vision Models

| Architecture | Throughput GPU (alert s ⁻¹) | Throughput CPU (alert s ⁻¹) | Memory (MB) |
|--------------|--------------------------------------------|--------------------------------------------|----------------|
| Custom CNN | 3708.5 | 136.1 | 345.5 |
| ConvNeXt | 6622.3 | 874.6 | 62.7 |
| MaxViT | 459.0 | 43.2 | 1045.6 |

the peak GPU memory usage during the inference test. The numbers we report are the medians of three trials.

Table 2 shows the results of the computational cost analysis. The MaxViT shows a large computational burden: 20.2× and 3.2× slower on CPU throughput than the ConvNeXt and the custom CNN, respectively, while also requiring 16.7× and 3.0× more memory. This is a product of the attention mechanisms used in the model which introduce large intermediate feature maps whose memory footprint scales roughly quadratically with the spatial dimensions of the input (Dosovitskiy et al. 2020; Tu et al. 2022). These costs are intrinsic to the MaxViT design as it is predominantly driven by these costly attention layers. The ConvNeXt achieves the best throughput and memory efficiency, despite having more trainable parameters than the custom CNN. Quantitatively, the ConvNeXt is 6.4× faster on CPU throughput and uses 5.5× less memory. The cost advantage of the ConvNeXt stems from numerous factors of its design; here we highlight two. The first is that the ConvNeXt adopts “depthwise convolutions” in which each channel input into a layer is convolved independently with its own kernel, rather than all channels being jointly processed as in a traditional convolutional layer. This is followed by a “pointwise convolution” which uses a 1 × 1 kernel to restore the ability to mix information across channels which was prohibited by the depthwise convolution (Chollet 2016; Howard et al. 2017). This dramatically reduces the total number of operations computed relative to an analogous standard convolutional layer. The second advantage is that ConvNeXt also adopts a large kernel stride in early convolutional layers which significantly decreases the spatial resolution of the input early on in the model. This has the effect of reducing the size of the intermediate feature maps produced within the model, reducing computation and memory requirements in later stages of the model. The custom CNN lacks these optimizations. Although it could be redesigned to adopt some of them, this illustrates one critical advantage of using off-the-shelf architectures; the design of a model architecture involves highly technical and nuanced choices which can be easily overlooked while still having a strong influence on the model’s computational cost.

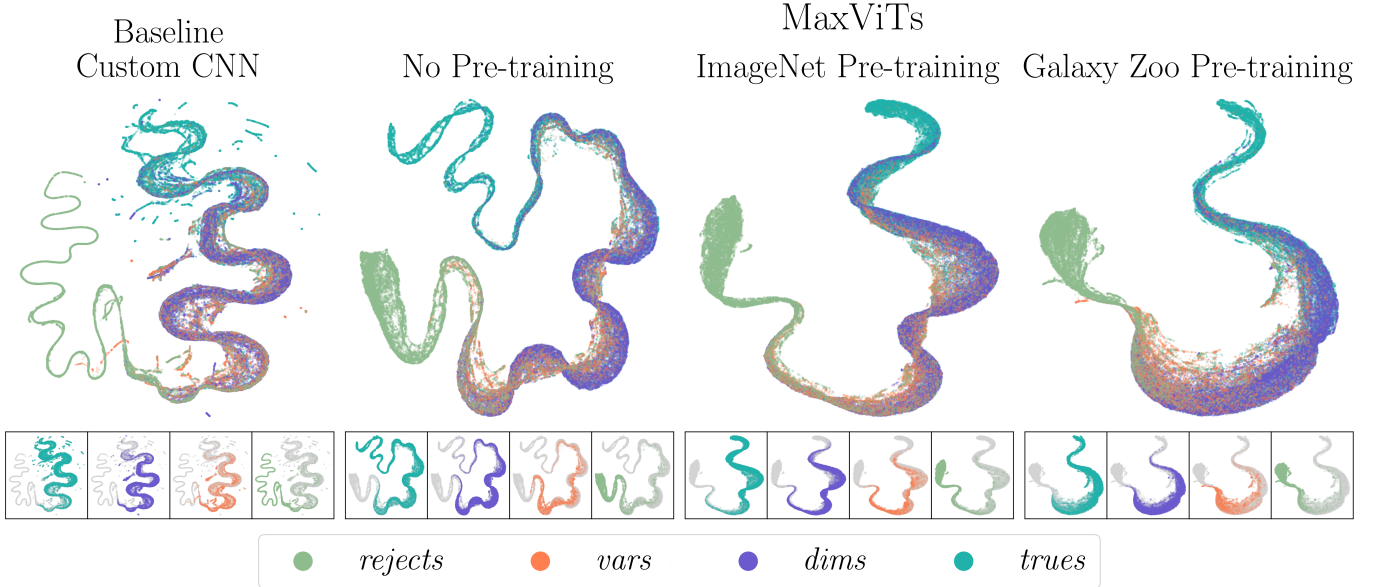


Figure 4. UMAP embeddings for the baseline custom CNN and MaxViT models with two pre-training regimens and another MaxViT trained from scratch. Points are colored by the subset of the training data from which they originate: *trues* (teal) bright extragalactic transients; *dims* (dark blue) faint sources with transient-like light curves; *vars* (orange) AGN, CVs, quasars, VarStars; *rejects* (light green) miscellaneous sources rejected as bright transient candidates. All latent spaces show the similar qualitative structure with a continuum having *trues* and *rejects* at opposite ends with *dims* and *vars* intermediate to them, but none are identical implying that the pre-training has a lasting effect on the models after fine-tuning.

5.2. Qualitative Comparison of UMAP Embeddings

We explore two-dimensional UMAP embeddings from MaxViT models to investigate whether or not models with distinct pre-training regimens converge to identical final states during fine-tuning. We extract high-dimensional representations from the penultimate layer of the models and fit a UMAP transformation to reduce their dimensionality and assist with making a human-interpretable visualization of the model’s latent space. In all cases, we adopt default values for the UMAP transformation hyperparameters established in McInnes et al. (2018) and the UMAP Python package:²² $N_{\text{neighbors}} = 15$, $\text{min_dist} = 0.1$, and $N_{\text{components}} = 2$. In Figure 4, we show the UMAP embeddings for all alerts in the test split from the baseline custom CNN and three MaxViT models with different pre-training regimens or without pre-training. The points are colored by the subset of the training data from which they originate: *trues* represents spectroscopically confirmed bright ($m_{\text{peak}} \leq 18.5$ mag) extragalactic transients; *dims* represents dim ($m_{\text{peak}} > 18.5$ mag) sources with transient-like light curves; *vars* represents sources classified as AGN, cataclysmic variables (CVs), VarStars, or quasars; and *rejects* represents ZTF sources which pass the BTS alert filter but were not cataloged as bright extragalactic transients (see Rehemtulla et al. 2024 for details). We find that all four models have qualitatively similar latent spaces. They

all place points along a continuum with one end nearly entirely populated by the bright extragalactic transients and the other nearly entirely populated by the *rejects*. Intermediate to these, we observe significant overlap between the tail of the distribution of the *trues* and the *dims*. Many of the *trues* alerts in this space do belong to bright transients but occur while the transient is fainter than the BTS threshold (i.e., $m > 18.5$ mag). Overlapping the other end of the *dims* are the *vars* and some of the *rejects*. This aligns with the overlap in source populations in these subsets because all three are expected to include, e.g., CVs and AGN flares. Beyond strong high-level similarities, the UMAP visualizations of the MaxViT latent spaces do show some local substructure differences. The Galaxy Zoo pre-trained MaxViT, e.g., exhibits peninsulas protruding from the main continuum while the others do not. We caution the over-interpretation of the UMAP visualizations as they represent only a two-dimensional view of a much higher dimensional space, although we do find that these are suggestive that the pre-training has a lasting effect after the fine-tuning process.

We also explore the utility of UMAP embeddings for assisting with identifying outliers in the training set. We run an Isolation Forest (IF; Liu et al. 2008) independently on each subset (i.e., one IF each for the *trues*, *vars*, *dims* and *rejects*) of test split for each MaxViT UMAP embedding and select the 1% of alerts with the highest IF scores. We then select only the alerts which were marked as anomalous by all three MaxViTs. We visually inspect these sources and identify a number of interesting outliers. Some sources are genuinely rare and

²² <https://github.com/lmcinnes/umap>

astrophysically interesting, e.g., ZTF21abzithf which is an instance of two Type Ia supernovae (SN 2021yhz and SN 2024svu) occurring at the same location.²³ Type Ia supernova “siblings” like these can be excellent aids for improving standardization relations in their use as cosmological probes (Graham et al. 2022; Dhawan et al. 2025). Other outliers are rare but less astrophysically interesting, e.g., ZTF20aaewhin (AT 2017foz) which is a CV projected atop a background galaxy creating an image which looks much like a supernova. We also identify multiple supernovae with peak magnitudes very close to the BTS threshold ($18.5 < m_{\text{peak}}[\text{mag}] < 18.6$), e.g., ZTF20aaetirf (SN 2020mbe) with $m_{\text{peak}} = 18.517$ mag. Lastly, we identify two sources with incorrect labels in our training set. ZTF19aalbaff (AT 2019blg) is very likely a real extragalactic transient that was internally cataloged as an AGN because it is near the nucleus of its AGN host. ZTF22aafvbku is a duplicate ZTF source of ZTF22aadfrjm (SN 2022fbf) which is a Type Ia supernova (i.e., both contain mostly the same supernova light curve). The incorrect label originated from this duplicate ZTF source being rejected from BTS instead of being cataloged as a bright transient alongside its copy. Duplicates are rarely rejected instead of being cataloged, so we do not expect there is a significant population of other sources mislabeled in this way.

These findings demonstrate that pairing UMAP embeddings with IF enables the recovery of astrophysically interesting sources as well as label noise in our training set. Further use of UMAP and IF on `BTSbot` embeddings could assist with the identification of bright transients missed by the BTS team, a critical factor in their supernova rates calculations.

6. Discussion and Conclusions

Our experiments reveal several distinct advantages of using pre-trained, off-the-shelf models for alert classification. Compared to a baseline model specially designed for this transient candidate vetting task, these models consistently achieve equivalent or superior performance across a wide range of training set sizes. From another lens, the pre-trained off-the-shelf models are able to attain an equal level of performance with as many or fewer labeled examples. Off-the-shelf architectures also yield higher maximum performance than the baseline model (Sections 3, 4). We also find that pre-training always improves performance relative to training from scratch for the models using only images for prediction. Critically, pre-training on Galaxy Zoo generally facilitates better performance than pre-training on ImageNet. We argue that this finding is non-trivial as there remain key differences between the Galaxy Zoo and ZTF alert cutouts datasets. (i) ZTF alert cutouts are not red-green-blue color images like the Galaxy Zoo pre-training data; (ii) Galaxy Zoo images always

have a galaxy centered in the cutout, while ZTF cutouts often have off-center galaxies or no galaxies at all; and (iii) the Galaxy Zoo data come from observatories with different noise and detector properties than ZTF images.

The increases in performance metrics are modest in absolute terms, but they are made to models that already deliver a high level of performance. Furthermore, these modest gains have meaningful scientific impact when (i) data volumes scale significantly, as expected by LSST, and (ii) when models are used for tasks with minimal tolerance for misclassifications, such as autonomously triggering target of opportunity requests to oversubscribed facilities (Rehemtulla et al. 2025b).

Beyond performance metrics, off-the-shelf model architectures offer substantial benefits in terms of computational efficiency and ease-of-use. The process of designing, debugging and perfecting a custom model architecture can take many months, but integrating a publicly available, meticulously designed, and standard architecture into one’s code base takes just a small fraction of that time. Off-the-shelf architectures, like ConvNeXt are designed to be highly efficient (see Section 5). Nuanced choices made in the ConvNeXt design yield throughput and memory efficiency considerably better than the smaller, custom CNN which is representative of models presented in the literature. These efficiency gains are especially pertinent in the present day, as new image-differencing surveys are quickly coming online and alert rates are growing. New models must be rapidly transferred across surveys with minimal labeled data to aid with quickly beginning productive science operations.

Given these demonstrated benefits, we advocate for the broader adoption of pre-trained off-the-shelf models for classifying alerts from image-differencing surveys. The strong performance and low computational cost of the ConvNeXt models indicate they are an excellent choice for the bulk of applications. The MaxViT models do unlock higher maximum performance, so practical deployments may benefit from hybrid configurations, where attention-based models are used for high-priority or ambiguous alerts, while convolutional models process the bulk of the stream.

To facilitate this transition, we release `BTSbot` models presented here publicly on the Hugging Face platform,²⁴ along with open-source training code on GitHub.²⁵ We intend for these models to be used as a strong starting point for fine-tuning to other alert classification tasks on ZTF alerts or those from another survey. Lastly, we are deploying the multi-modal ConvNeXt pre-trained on Galaxy Zoo into production to assume the operational roles of the current `BTSbot` model. The considerably higher throughput of the ConvNeXt vision backbone will prevent `BTSbot` inference from being a

²³ One half of the alerts marked as anomalous from this source come from SN 2024svu and the other half come from SN 2021yhz.

²⁴ <https://huggingface.co/collections/nabeelr/pre-trained-off-the-shelf-btsbots>

²⁵ <https://github.com/nabeelre/BTSbot>

bottleneck in the alert processing pipelines of the brokers it is deployed in. The throughput of the custom CNN demonstrated with 12 CPU cores (see Section 5.1) is insufficient to keep up with the LSST alert stream which is expected to average upwards of $\mathcal{O}(10^2)$ alerts per second. Furthermore, the throughput of the larger, multi-modal production `BTSbot` is yet smaller. Thus, these updates to the `BTSbot` model architecture support the model’s longevity in the increasingly demanding landscape of real-time production environments and mark a transition toward processing alerts with pre-trained off-the-shelf vision models.

Acknowledgments

We thank members of the Northwestern Image and Video Processing Lab (Patrick Koller, Santiago López-Tapia, Philipp Srivastava) and the Bio Inspired Vision Lab (Emma Alexander, Mehmet Kerem Aydin, Tianao Li) for insightful discussions. We also thank the ZTF ML technical working group for helpful comments.

N.R. is supported by a Northwestern University Presidential Fellowship Award. N.R. and A.A.M. are partially supported by DoE award #DE-SC0025599. A.A.M. is also supported by Cottrell Scholar Award #CS-CSA-2025-059 from the Research Corporation for Science Advancement. N.R. is also partially supported by NSF grant # AST-2421845.

We gratefully acknowledge the support of the NSF-Simons AI-Institute for the Sky (SkAI) via grants NSF AST-2421845 and Simons Foundation MPS-AI-00010513.

Zwicky Transient Facility access for N.R., A.A.M. and V.G. S. was supported by Northwestern University and the Center for Interdisciplinary Exploration and Research in Astrophysics (CIERA).

M.W.C. acknowledges support from the National Science Foundation with grant Nos. PHY-2409481, PHY-2308862 and PHY-2117997.

This research was supported in part through the computational resources and staff contributions provided for the Quest high performance computing facility at Northwestern University which is jointly supported by the Office of the Provost, the Office for Research, and Northwestern University Information Technology.

The Gordon and Betty Moore Foundation, through both the Data-Driven Investigator Program and a dedicated grant, provided critical funding for SkyPortal. This publication uses data generated via the Zooniverse.org platform, development of which is funded by generous support, including a Global Impact Award from Google, and by a grant from the Alfred P. Sloan Foundation.

Based on observations obtained with the Samuel Oschin Telescope 48 inch and the 60 inch Telescope at the Palomar Observatory as part of the Zwicky Transient Facility project. ZTF is supported by the National Science Foundation under

grants Nos. AST-1440341, AST-2034437, and currently Award #2407588. ZTF receives additional funding from the ZTF partnership. Current members include Caltech, USA; Caltech/IPAC, USA; University of Maryland, USA; University of California, Berkeley, USA; University of Wisconsin at Milwaukee, USA; Cornell University, USA; Drexel University, USA; University of North Carolina at Chapel Hill, USA; Institute of Science and Technology, Austria; National Central University, Taiwan, and OKC, University of Stockholm, Sweden. Operations are conducted by Caltech’s Optical Observatory (COO), Caltech/IPAC, and the University of Washington at Seattle, USA.

This research has made use of NASA’s Astrophysics Data System.

Facility: PO:1.2m.

Software: `astropy` (Astropy Collaboration et al. 2013, 2018, 2022), `matplotlib` (Hunter 2007), `numpy` (Harris et al. 2020), `pandas` (The pandas development team 2020; McKinney 2010), `penquins` (Duev et al. 2021), `scikit-learn` (Pedregosa et al. 2011), `PyTorch` (Paszke et al. 2019), `SkyPortal` (van der Walt et al. 2019; Coughlin et al. 2023), `tim` (Wightman 2019), and the Weights and Biases platform (Biewald 2020).

ORCID iDs

Nabeel Rehemtulla  <https://orcid.org/0000-0002-5683-2389>
 Adam A. Miller  <https://orcid.org/0000-0001-9515-478X>
 Mike Walmsley  <https://orcid.org/0000-0002-6408-4181>
 Ved G. Shah  <https://orcid.org/0009-0009-1590-2318>
 Theophile Jegou du Laz  <https://orcid.org/0009-0003-6181-4526>
 Michael W. Coughlin  <https://orcid.org/0000-0002-8262-2924>
 Argyro Sasli  <https://orcid.org/0000-0001-7357-0889>
 Joshua Bloom  <https://orcid.org/0000-0002-7777-216X>
 Christoffer Fremling  <https://orcid.org/0000-0002-4223-103X>
 Matthew J. Graham  <https://orcid.org/0000-0002-3168-0139>
 Steven L. Groom  <https://orcid.org/0000-0001-5668-3507>
 David Hale  <https://orcid.org/0000-0002-4662-122X>
 Ashish A. Mahabal  <https://orcid.org/0000-0003-2242-0244>
 Daniel A. Perley  <https://orcid.org/0000-0001-8472-1996>
 Josiah Purdum  <https://orcid.org/0000-0003-1227-3738>
 Ben Rusholme  <https://orcid.org/0000-0001-7648-4142>
 Jesper Sollerman  <https://orcid.org/0000-0003-1546-6615>
 Mansi M. Kasliwal  <https://orcid.org/0000-0002-5619-4938>

References

- Acero-Cuellar, T., Bianco, F., Dobler, G., et al. 2023, *AJ*, 166, 115
 Akesson, R., Armus, L., Bachelet, E., et al. 2019, arXiv:1902.05569
 Alfonzo, J. P., Iyer, K. G., Akiyama, M., et al. 2024, *ApJ*, 967, 152

- Andreoni, I., Jacobs, C., Hegarty, S., et al. 2017, *PASA*, **34**, e037
- Astropy Collaboration, Price-Whelan, A. M., Lim, P. L., et al. 2022, *ApJ*, **935**, 167
- Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., et al. 2018, *AJ*, **156**, 123
- Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, *A&A*, **558**, A33
- Bailey, S., Aragon, C., Romano, R., et al. 2007, *ApJ*, **665**, 1246
- Bellm, E. C., Kulkarni, S. R., Barlow, T., et al. 2019b, *PASP*, **131**, 068003
- Bellm, E. C., Kulkarni, S. R., Graham, M. J., et al. 2019a, *PASP*, **131**, 018002
- Bhambra, P., Joachimi, B., & Lahav, O. 2022, *MNRAS*, **511**, 5032
- Biewald, L. 2020, Experiment Tracking with Weights and Biases, <https://www.wandb.com/>
- Blagorodnova, N., Neill, J. D., Walters, R., et al. 2018, *PASP*, **130**, 035003
- Bloom, J. S., Richards, J. W., Nugent, P. E., et al. 2012, *PASP*, **124**, 1175
- Brink, H., Richards, J. W., Poznanski, D., et al. 2013, *MNRAS*, **435**, 1047
- Cabrera-Vives, G., Reyes, I., Förster, F., Estévez, P. A., & Maureira, J.-C. 2017, *ApJ*, **836**, 97
- Carrasco-Davis, R., Reyes, E., Valenzuela, C., et al. 2021, *AJ*, **162**, 231
- Chollet, F. 2016, arXiv:1610.02357
- Coughlin, M. W., Bloom, J. S., Nir, G., et al. 2023, *ApJS*, **267**, 31
- Dekany, R., Smith, R. M., Riddle, R., et al. 2020, *PASP*, **132**, 038001
- Deng, L. 2012, *ISPM*, **29**, 141
- DESI Collaboration, Aghamousa, A., Aguilar, J., et al. 2016, arXiv:1611.00036
- Dey, A., Schlegel, D. J., Lang, D., et al. 2019, *AJ*, **157**, 168
- Dhawan, S., Mortzell, E., Johansson, J., et al. 2025, *A&A*, **702**, A190
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al. 2020, arXiv:2010.11929
- Duev, D., Shin, K. M., & Singer, L. 2021, dmitryduev/penguins: A Python Client for dmitryduev/kowalski, v2.1.2, Zenodo, doi:10.5281/zenodo.5651471
- Duev, D. A., Bolin, B. T., Graham, M. J., et al. 2021, *AJ*, **161**, 218
- Duev, D. A., Mahabal, A., Masci, F. J., et al. 2019, *MNRAS*, **489**, 3582
- Duev, D. A., & van der Walt, S. J. 2021, arXiv:2111.12142
- Fiscale, S., Ciaramella, A., Inno, L., et al. 2021, *RNAAS*, **5**, 91
- Förster, F., Cabrera-Vives, G., Castillo-Navarrete, E., et al. 2021, *AJ*, **161**, 242
- Fremling, C., Miller, A. A., Sharma, Y., et al. 2020, *ApJ*, **895**, 32
- Fukushima, K., & Miyake, S. 1982, *PatRe*, **15**, 455
- Gieseke, F., Bloemen, S., van den Bogaard, C., et al. 2017, *MNRAS*, **472**, 3101
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. 2013, arXiv:1311.2524
- Glorot, X., & Bengio, Y. 2010, in Proc. Machine Learning Research, Proc. Thirteenth Int. Conf. Artificial Intelligence and Statistics, 9, ed. Y. W. Teh & M. Titterton (PMLR), 249, <https://proceedings.mlr.press/v9/glorot10a.html>
- Graham, M. L., Fremling, C., Perley, D. A., et al. 2022, *MNRAS*, **511**, 241
- Graham, M. J., Kulkarni, S. R., Bellm, E. C., et al. 2019, *PASP*, **131**, 078001
- Groot, P. J., Bloemen, S., Vreeswijk, P. M., et al. 2024, *PASP*, **136**, 115003
- Gupta, R., Muthukrishna, D., & Audenaert, J. 2025a, arXiv:2510.12958
- Gupta, R., Muthukrishna, D., Rehemtulla, N., & Shah, V. 2025b, *MNRAS*, **542**, L132
- Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, *Natur*, **585**, 357
- He, K., Chen, X., Xie, S., et al. 2021, arXiv:2111.06377
- He, K., Zhang, X., Ren, S., & Sun, J. 2015, arXiv:1502.01852
- He, K., Zhang, X., Ren, S., & Sun, J. 2016, in IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 1
- Howard, A. G., Zhu, M., Chen, B., et al. 2017, arXiv:1704.04861
- Hunter, J. D. 2007, *CSE*, **9**, 90
- Ivezić, Ž., Kahn, S. M., Tyson, J. A., et al. 2019, *ApJ*, **873**, 111
- Jegou du Laz, T., Coughlin, M. W., Bachant, P., et al. 2025, arXiv:2511.00164
- Junell, A., Sasli, A., Fontinele Nunes, F., et al. 2025, arXiv:2507.16088
- Kaiser, N., Aussel, H., Burke, B. E., et al. 2002, *SPIE*, **4836**, 154
- Khan, S., Naseer, M., Hayat, M., et al. 2022, *ACM Comput. Surv.*, **54**, 1
- Kim, Y. L., Rigault, M., Neill, J. D., et al. 2022, *PASP*, **134**, 024505
- Krizhevsky, A., & Hinton, G. 2009, Learning multiple layers of features from tiny images, Tech. rep
- Law, N. M., Corbett, H., Galliher, N. W., et al. 2022, *PASP*, **134**, 035003
- Lawrence, A., Warren, S. J., Almaini, O., et al. 2007, *MNRAS*, **379**, 1599
- LeCun, Y., Bengio, Y., & Hinton, G. 2015, *Natur*, **521**, 436
- Li, W., Chen, H.-Y., Lin, Q., et al. 2025, arXiv:2510.06200
- Lin, T., Maire, M., Belongie, S. J., et al. 2014, arXiv:1405.0312
- Lintott, C., Schawinski, K., Bamford, S., et al. 2011, *MNRAS*, **410**, 166
- Lintott, C. J., Schawinski, K., Slosar, A., et al. 2008, *MNRAS*, **389**, 1179
- Liu, F. T., Ting, K. M., & Zhou, Z.-H. 2008, Proc. Eighth IEEE Int. Conf. Data Mining (IEEE), 413
- Liu, Y., Fan, L., Hu, L., et al. 2025, *A&A*, **693**, A105
- Liu, Z., Mao, H., Wu, C.-Y., et al. 2022, arXiv:2201.03545
- Loshchilov, I., & Hutter, F. 2016, arXiv:1608.03983
- Loshchilov, I., & Hutter, F. 2017, arXiv:1711.05101
- Makhlof, K., Turpin, D., Corre, D., et al. 2022, *A&A*, **664**, A81
- Masci, F. J., Laher, R. R., Rusholme, B., et al. 2019, *PASP*, **131**, 018003
- Matheson, T., Stubens, C., Wolf, N., et al. 2021, *AJ*, **161**, 107
- McKinney, W. 2010, in Proc. 9th Python in Science Conf., ed. S. van der Walt & J. Millman, 56
- McInnes, L., Healy, J., & Melville, J. 2018, arXiv:1802.03426
- Miller, A. A., Abrams, N. S., Aldering, G., et al. 2025, *PASP*, **137**, 094204
- Mohale, K., & Lochner, M. 2024, *MNRAS*, **530**, 1274
- Möller, A., Peloton, J., Ishida, E. E. O., et al. 2021, *MNRAS*, **501**, 3272
- Paszke, A., Gross, S., Massa, F., et al. 2019, arXiv:1912.01703
- Patterson, M. T., Bellm, E. C., Rusholme, B., et al. 2019, *PASP*, **131**, 018001
- Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, *Journal of Machine Learning Research*, **12**, 2825
- Perley, D. A., Fremling, C., Sollerman, J., et al. 2020, *ApJ*, **904**, 35
- Pranshu, K., Misra, K., Ailawadhi, B., et al. 2025, *MNRAS*, **538**, 133
- Rehemtulla, N., Coughlin, M. W., Miller, A. A., & du Laz, T. J. 2025a, *NatAs*, **9**, 1764
- Rehemtulla, N., Jacobson-Galán, W. V., Singh, A., et al. 2025b, *ApJ*, **985**, 241
- Rehemtulla, N., Miller, A. A., Jegou du Laz, T., et al. 2024, *ApJ*, **972**, 7
- Rigault, M., Neill, J. D., Blagorodnova, N., et al. 2019, *A&A*, **627**, A115
- Russakovsky, O., Deng, J., Su, H., et al. 2014, arXiv:1409.0575
- Sadr, V., Bassett, A., & Sekyi, B. A. 2023, *RASTI*, **2**, 586
- Sadeghat, N., & Mahabal, A. 2018, *MNRAS*, **476**, 5365
- Sheng, X., Nicholl, M., Smith, K. W., et al. 2024, *MNRAS*, **531**, 2474
- Smith, K. W., Smartt, S. J., Young, D. R., et al. 2020, *PASP*, **132**, 085002
- Stoppa, F., Bulmus, T., Bloemen, S., et al. 2025, *NatAs*, **9**, 1869
- The pandas development team 2020, pandas-dev/pandas: Pandas, latest, Zenodo, doi:10.5281/zenodo.3509134
- Tonry, J. L. 2011, *PASP*, **123**, 58
- Tonry, J. L., Denneau, L., Heinze, A. N., et al. 2018, *PASP*, **130**, 064505
- Tu, Z., Talebi, H., Zhang, H., et al. 2022, arXiv:2204.01697
- van der Walt, S. J., Crellin-Quick, A., & Bloom, J. S. 2019, *JOSS*, **4**, 1247
- Vaswani, A., Shazeer, N., Parmar, N., et al. 2017, arXiv:1706.03762
- Walmsley, M., Bowles, M., Scaife, A. M. M., et al. 2024, arXiv:2404.02973
- Walmsley, M., Scaife, A. M. M., Lintott, C., et al. 2022, *MNRAS*, **513**, 1581
- Weston, J. G., Smith, K. W., Smartt, S. J., Tonry, J. L., & Stevance, H. F. 2024, *RASTI*, **3**, 385
- Wightman, R. 2019, PyTorch Image Models, v0.8.10dev0, GitHub Zenodo, doi:10.5281/zenodo.4414861
- Williams, R. D., Francis, G. P., Lawrence, A., et al. 2024, *RASTI*, **3**, 362
- Wu, J. F., & Boada, S. 2019, *MNRAS*, **484**, 4683
- York, D. G., Adelman, J., Anderson, J. E. J., et al. 2000, *AJ*, **120**, 1579
- Zhang, J., Dessart, L., Wang, X., et al. 2024, *ApJL*, **970**, L18
- Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., & Torralba, A. 2017, *Journal of Vision*, **17**, 296