HARVEST Inference: Characterizing Digital Agriculture Workloads across Compute Continuum

Tian Chen
chen.9891@osu.edu
The Ohio State University
Columbus, Ohio, USA

Quentin Anthony
anthony.301@osu.edu
The Ohio State University
Columbus, Ohio, USA

Dhabaleswar K. Panda
panda.2@osu.edu
The Ohio State University
Columbus, Ohio, USA

Abstract

The advancement of AI has revolutionized digital agriculture, enabling processing of diverse data types from sensor tabular data to drone imagery. This supports applications like nitrogen fertilizer management and residue cover estimation.

Meanwhile, the complexity of AI deployment creates an urgent need for democratized AI as cyberinfrastructure (CI). HARVEST-2.0 provides a framework for end-to-end farm localized model training and deployment services. It supports multi-scenario deployment online, offline, and real-time across the compute continuum, making digital agriculture by and for those who work the land.

Flexible deployment enables diverse applications but complicates model selection due to the accuracy latency trade off. In this work, we evaluate the HARVEST inference pipeline across models, datasets, and platforms. We demonstrate how elaborate selected hyperparameters can improve throughput under latency constraints, and highlight the benefits of GPU-accelerated data preprocessing. Together, Our findings offer performance insights to guide application-specific tuning.

CCS Concepts

• **Applied computing** → **Agriculture**; • **Computing methodologies** → *Artificial intelligence*; • **Computer systems organization** → *Architectures.*

Keywords

Inference Characterization, Digital Agriculture, Compute Continuum

ACM Reference Format:
Tian Chen, Quentin Anthony, and Dhabaleswar K. Panda. 2025. HARVEST Inference: Characterizing Digital Agriculture Workloads across Compute Continuum. In *54th International Conference on Parallel Processing Companion (ICPP Companion '25), September 08–11, 2025, San Diego, CA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3750720.3758082

1 Introduction

In recent years, Artificial Intelligence (AI) has achieved significant success across various domains. With the emergence of Convolutional Neural Networks (CNNs) and Transformer-based models, AI has expanded its capabilities from single image or sensor inputs to multi-modal inputs, including tabular data, image-text pairs, process-based data, text data, and audio. AI-accelerated digital agriculture has now sparked the fourth agricultural revolution.

However, while digital agriculture represents a technological advancement, its implementation has encountered resistance and exacerbated existing inequalities [12]. Valuable agricultural data are frequently controlled by private companies, restricting access for farmers. Producers face pressure to purchase costly technologies to maintain competitive yields, with smallholders especially at risk of being left behind due to limited access to knowledge and localized AI models. Regional variations in land data further limit the effectiveness of digital tools without model adaptation capabilities.

To address these challenges, the Intelligent Cyberinfrastructure with Computational Learning in the Environment (ICICLE) [4] project aims to transform today's AI landscape from a narrow set of privileged disciplines to democratized cyberinfrastructure. ICICLE democratizes AI through workforce development, auditable workflows, and co-designed infrastructure, ensuring data integrity, privacy, and explainability—making digital agriculture by and for those who work the land. This enables a range of agricultural applications including nitrogen fertilizer management, residue cover estimation, soil aggregate size estimation, and pest detection [13].

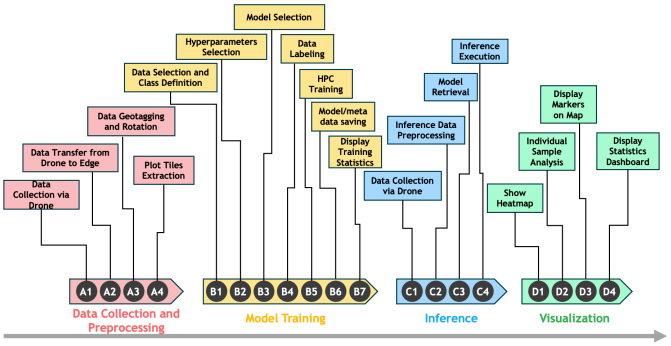


Figure 1: High-level overview of the HARVEST-2.0 framework.

HARVEST-2.0 [6], as shown in Fig. 1, serves as the framework developed under the ICICLE project to drive these applications. It provides farmers with an end-to-end AI training and deployment platform, enabling landholders to easily train localized AI models with their own data. The framework encompasses data preprocessing, distributed model training, cloud or edge-based inference, and result visualization components, covering the entire AI

ICPP Companion '25, San Diego, CA, USA

ACM ISBN 979-8-4007-2109-0/25/09
https://doi.org/10.1145/3750720.3758082

model lifecycle while enabling agile deployment with fast training times. Developed with privacy and data ownership priorities, combined with semi-supervised learning techniques, this framework mitigates the time and expert effort required for labeling challenges [29].

HARVEST supports deployment across the compute continuum, allowing end users to tailor inference to specific application scenarios. A single training process enables deployment on both edge and cloud systems—inference can run in the cloud with high throughput after unified preprocessing, or be performed on edge devices in the field for low-latency results supporting real-time decisions.

However, this flexibility introduces the challenge of selecting the appropriate model for each deployment context, requiring careful balance of accuracy-latency trade-offs. To address this challenge, this work thoroughly evaluates the HARVEST inference pipeline across multiple models and datasets on both edge and cloud platforms, with the objective of better understanding the pipeline and providing end users with guidance for application-specific tuning.

2 Inference Deployment Scenarios

2.1 Digital Agriculture Applications

Machine learning (ML) has proven effective for extracting patterns and conducting analysis from multimodal data, becoming widely adopted in agriculture to improve farming profitability and sustainability.

For commonly used image data, Convolutional Neural Networks (CNNs), such as ResNet [15], are widely applicable to various downstream tasks, including applications already supported by the HARVEST framework: residue cover on soil surface estimation, corn borer detection, and leaf aphid detection.

Vision Transformers (ViTs) [11] have emerged as successors to CNNs, bringing the powerful Transformer architecture—which achieved remarkable success in Natural Language Processing—into image recognition. ViTs offer enhanced semantic understanding capabilities for agricultural applications.

These vision-based models are often deployed with Unmanned Aerial Systems (UAS) or ground vehicles to analyze pixel-level input from onboard cameras. Notable examples include the Mineral Rover [1], which traversed agricultural fields to collect high-quality images of individual plants. Combined with complementary datasets, this platform provided valuable insights into plant growth patterns and environmental interactions

Beyond vision-based approaches, with the rise of ChatGPT [7], text-based large language models (LLM) have gained widespread recognition. These models have significantly lowered barriers to accessing domain-specific agricultural knowledge, promoting knowledge dissemination and improving accessibility for farmers and researchers. Large Language Models (LLMs) offer new paradigms for text-based data analysis in agriculture. LLMs trained specifically on agricultural knowledge, such as AgroLLM [25], combined with Retrieval-Augmented Generation (RAG), can provide more specialized and contextually relevant answers than general-purpose LLMs.

Vision-Language Models (VLMs) such as Molmo [9] enable interactive analysis of plant diseases directly from combined image and text inputs, greatly extending ML model capabilities and making expert knowledge more accessible to practitioners. Beyond this, the exploration of additional modalities continues to grow. Gadiraju et al.[14] demonstrated joint prediction using spatial, spectral, and temporal data for crop type identification, and growing research on data-driven agricultural systems leveraging diverse sensor inputs underscores the potential of multimodal approaches [26].

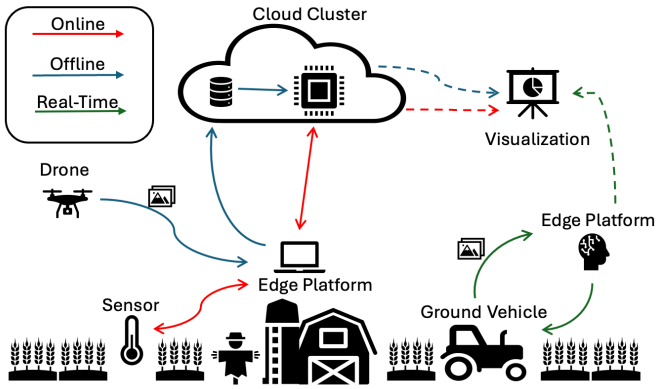


Figure 2: Overview of Dataflow Across Deployment Scenarios

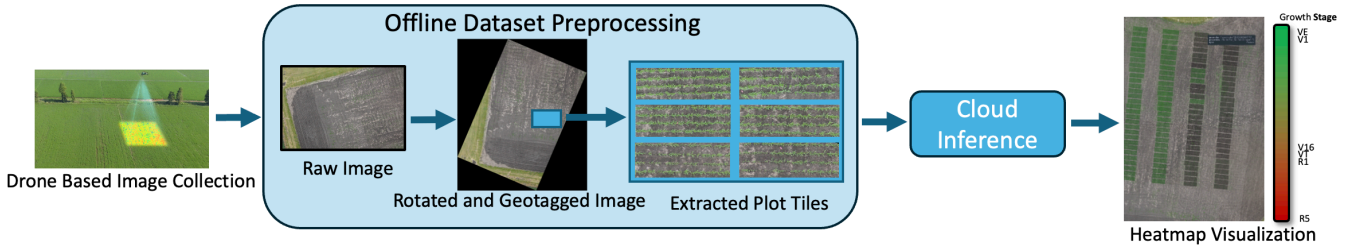
2.2 Deployment Scenarios

The diverse digital agriculture applications mentioned above pose significant challenges to the design and support of inference pipelines. Tuning the HARVEST inference process for different scenarios is the central focus of this characterization study. Based on the requirements for data processing, the nature of the compute platform, and how results are utilized, we broadly categorize deployment scenarios into the following three:

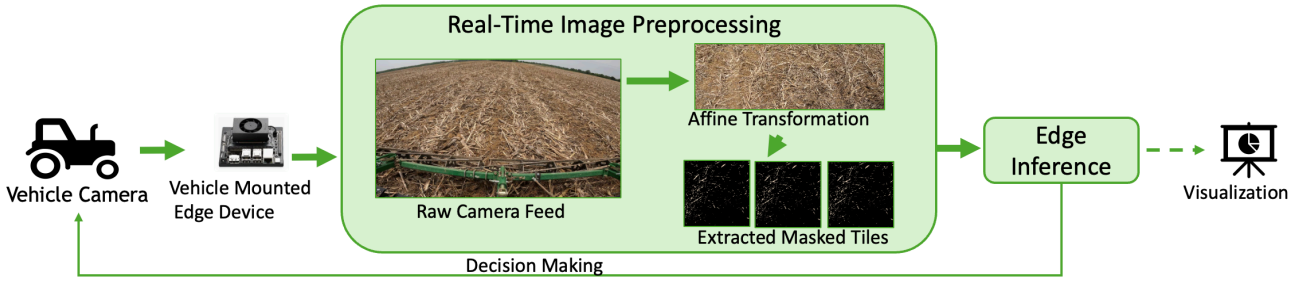
2.2.1 Online Inference. Online inference provides streaming inference services for target applications. Data is processed and returned in real time upon being uploaded to the compute platform (either edge or cloud), enabling inference on demand. Under online inference, real-time latency is traded off for high throughput. This setup presents challenges for data transmission, especially when transmitting large image data to the cloud. It would be beneficial to leverage advanced wireless capabilities to enable real-time data collection and transmission.

2.2.2 Offline Inference. Unlike online inference, offline inference is performed after a batch of data has been collected. It is suitable for non-time-sensitive tasks, such as field-by-field processing. With full data availability, this approach allows for more extensive preprocessing before AI inference—ideal for applications requiring image stitching or orthomosaic generation from UAS or satellite imagery. As shown in Fig. 3a, the workflow at the Northwest Agricultural Research Station [28] demonstrates this approach: drone images are first stitched using OpenDroneMap [3], followed by tiling and offline processing via the HARVEST inference pipeline, ultimately generating fine-grained heatmaps and other visual outputs.

2.2.3 Real-Time Inference. For on-the-fly decision-making—such as device actions triggered by inference results, similar to autonomous driving in agricultural contexts—real-time inference on edge devices offers a rapid-response solution. This mode prioritizes low



(a) Drone Based Offline Inference Example



(b) Ground Vehicle Based Real-time Inference Example

Figure 3: Illustrations of Different HARVEST Inference Scenarios

latency. From raw image preprocessing to ML model output, the entire pipeline must operate within strict time constraints. Given the limited compute, memory, and energy resources on edge platforms, optimizing the end-to-end pipeline under such constraints becomes a key focus of this characterization study.

3 The HARVEST Inference Pipeline

The HARVEST inference pipeline follows a modular design that decouples the frontend—which handles diverse task requests—from the backend, which executes model inference. The frontend is responsible for transmitting or locally reading input data and generating requests to the backend.

The backend hosts model instances, each dedicated to a specific inference task. For example, a trained ViT model for residue cover estimation at the Northwest Agricultural Research Station.

Each model family is paired with its own preprocessing method, and in some cases, the dataset itself may require task-specific preprocessing. These preprocessing routines are also encapsulated as separate backend engine instances. A single request may trigger multiple backend calls to support different downstream tasks, which can reuse shared preprocessing steps when applicable.

Backend request orchestration is currently provided by the NVIDIA Triton Server [21]. Each model instance can be powered by popular deep learning frameworks like PyTorch [23], TensorFlow [5], or inference-oriented engines such as vLLM [16] and TensorRT [19]. This allows the backend to support a majority of model formats, including framework-neutral options like ONNX [22], maximizing compatibility.

Preprocessing is handled via Torchvision [17], OpenCV [2], GPU-accelerated frameworks such as NVIDIA DALI [20], or custom Python scripts. This backend architecture is also prepared for future scale-out through different parallelism strategies.

3.1 Inference Workload

As previously described, the latency of an inference request consists of three components: dataset-specific preprocessing, model-specific preprocessing, and model inference time.

For a given hardware platform, compute capability is typically expressed as FLOPS (Floating-point Operations Per Second), which varies by numerical precision and hardware support. Lower-precision formats like INT8 or FP16 offer faster inference but may reduce accuracy. BF16 or FP16, as used in our experiments, provides a common balance between speed and accuracy.

Similarly, for a given ML model, the required FLOPs and per-image memory footprint can be estimated layer-wise. Computational cost varies by layer type. For instance, attention layers in Transformer models generally have much higher computational intensity than CNN layers with comparable parameter counts. Additionally, attention layers scale quadratically with respect to input sequence length, making them less suitable for large image inputs. Recent work seeks to address this limitation through state-based architectures such as RWKV [24]. By summing the total FLOPs required by the model and dividing by the available hardware FLOPS, we can estimate the theoretical upper bound on throughput or the minimum achievable latency for a given batch size.

3.2 Preprocessing Requirements

Compared to the relatively stable model inference latency, preprocessing introduces greater variance due to diverse input formats and operations.

Models require preprocessing consistent with their training-time distribution; otherwise, input mismatch may lead to unexpected outputs. For vision models, such preprocessing often includes image decoding, resizing, cropping, and pixel-wise normalization. The computational cost scales with image size, and if handled solely by the CPU, can bottleneck the pipeline. GPU-accelerated preprocessing can potentially help mitigate this bottleneck.

Certain data sources also require task-specific preprocessing. For example, UAS images may need offline stitching, while raw camera streams may require perspective transformation. These dependencies collectively determine whether a given application is suitable for real-time inference.

3.3 Challenges in Application Specific Optimization

The dynamic nature of the inference pipeline presents significant challenges for tuning in digital agriculture applications.

Effective optimization begins with clearly defining the target deployment scenario and platform. For edge deployments, model size must be considered early in training, as memory and compute constraints limit both model selection and concurrency.

When considering end-to-end serving—from raw data input all the way to result output—throughput depends not solely on compute intensity, but also on CPU, I/O, and network subsystems. At larger scales, distributed deployment introduces added complexity, making system characterization and tuning even more challenging.

Table 1: Evaluated Cloud and Edge Platforms

Platform	OSC Pitzer Cluster (V100)	MRI Cluster (A100)	NVIDIA Jetson Orin Nano Super
CPU	40 cores	128 cores	6 cores
GPU	NVIDIA V100 16GB×2	NVIDIA A100 40GB×2	Ampere architecture based 1024 CUDA cores 32 tensor cores
Memory	384GB	256GB	8GB
Scenario	Online, Offline	Online, Offline	Real-Time
Theory TFLOPS	112 @FP16	312 @BF16	17 @FP16
Practical TFLOPS	92.6	236.3	11.4 @BF16

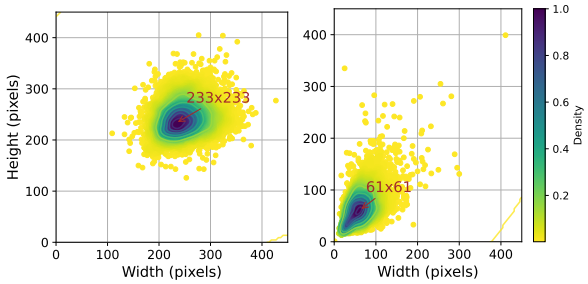
Note: V100 and A100 experiments used only one of the two available GPUs. Jetson platforms feature CPU and GPU sharing 8GB unified memory and operate in 25W power mode.

4 Experiments

This study was conducted using two cloud platforms and one edge device, with hardware specifications shown in Tab. 1. The Ohio Supercomputer Center's Pitzer Cluster provides scalable and sufficient compute resources for the HARVEST framework, with tests conducted on its V100 nodes using a single GPU setup. MRI is an in-house cluster at The Ohio State University that serves as another target deployment platform for the HARVEST framework, with evaluations conducted on its A100 nodes, also using a single GPU setup. Both clusters are capable of supporting online or offline inference services. The NVIDIA Jetson Nano represents an edge device that will be installed on ground vehicles connected directly to GoPro cameras to enable real-time inference in future deployments.

Since theoretical FLOPS numbers provided by manufacturers tend to be overly optimistic, we benchmarked practical FLOPS performance over GEMM operations on all three platforms. BF16 was used on the A100, while FP16 was used on the other two platforms due to limited support. As shown in the Table 1, the FLOPS efficiency achieved on each platform ranges from 75.74% to 82.68%.

4.0.1 Datasets. In the experiments, we used five different agricultural datasets to represent a variety of downstream tasks. The image size distribution is shown in Fig. 4, where the X-axis represents total pixels and the Y-axis represents the density of that image size, with the most common image size in each dataset labeled on top. Among these, CRSA consists of raw camera input feeds that require dataset-specific preprocessing. Our datasets span a wide range of image sizes—some have highly uniform dimensions while others vary significantly—providing coverage for common use cases in agricultural applications.



(a) Weed Detection in Soybean **(b) Sugar Cane-Spittle Bug**

Figure 4: Image Size Distribution Across Different Datasets

4.0.2 Models. The models used in the experiments, along with their specifications, are shown in Tab. 3. In this experiment, we selected four widely used deep learning models for computer vision, varying in size and covering both transformer-based and CNN-based architectures, with each model requiring a specific input size. The models are provided in the platform-neutral ONNX format and internally converted to the inference-oriented TensorRT format.

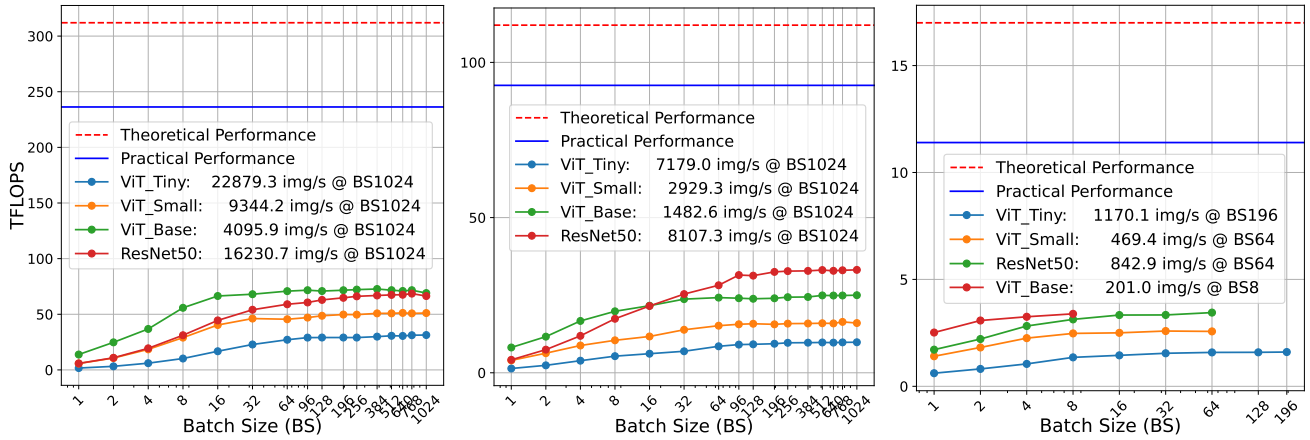
The ViT family adopts a transformer-based architecture, and comparing ViT Small with the CNN-based ResNet50 model, we observe that despite having a smaller parameter count, ViT exhibits higher computational demand. For transformer-based models, the majority of computation is consumed by MLP layers, accounting for 81.73% in ViT Tiny, while attention layers account for 18.23%. In contrast, convolution operations account for 99.5% of ResNet50's overall computational intensity.

4.1 Engine Performance

As illustrated in Fig. 5, a substantial gap exists between the Model FLOPs Utilization (MFU) and the practical upper bound during

Table 2: Agriculture Datasets Used in The Evaluation

Dataset	Classes	Samples	Image Size	Use Case
Plant Village [8]	39	43430	256x256	Plant disease classification
Weed Detection in Soybean [10]	4	10635	Show in Fig. 4a	Weed detection in soybeans
Sugar Cane-Spittle Bug [27]	2	10100	Show in Fig. 4b	Pest bugs detection
Fruits-360 [18]	81	40998	100x100	Fruits classification
Corn Growth Stage [28]	23	52198	224x224	Corn Growth Stage Classification, UAS Based
CRSA	-	992	3840x2160	Crop Residue Soil Aggregate, Ground Vehicle based



(a) A100 **(b) V100** **(c) Jetson**

Figure 5: Scaling Behavior Of Compute Intensity With Varying Batch Sizes On Different Hardware Platforms. The Dashed Lines Represent The Theoretical FLOPS, While The Solid Lines Indicate The Actual FLOPS Achieved By Each Model.

Table 3: Model Evaluated and Computational Intensity

Model		ViT Tiny	ViT Small	ViT Base	ResNet50
Parameter		5.39M	21.40M	85.80M	25.56M
Archetecture		Transformer Based			CNN Based
GFLOPs/Image		1.37	5.47	16.86	4.09
Input Size		32×32	32×32	224×224	224×224
Throughput	A100	172,508	43,214	14,013	57,775
UpperBound	V100	67,602	16,935	5,491	22,641
images/sec	Jetson	8,322	2,085	676	2,787

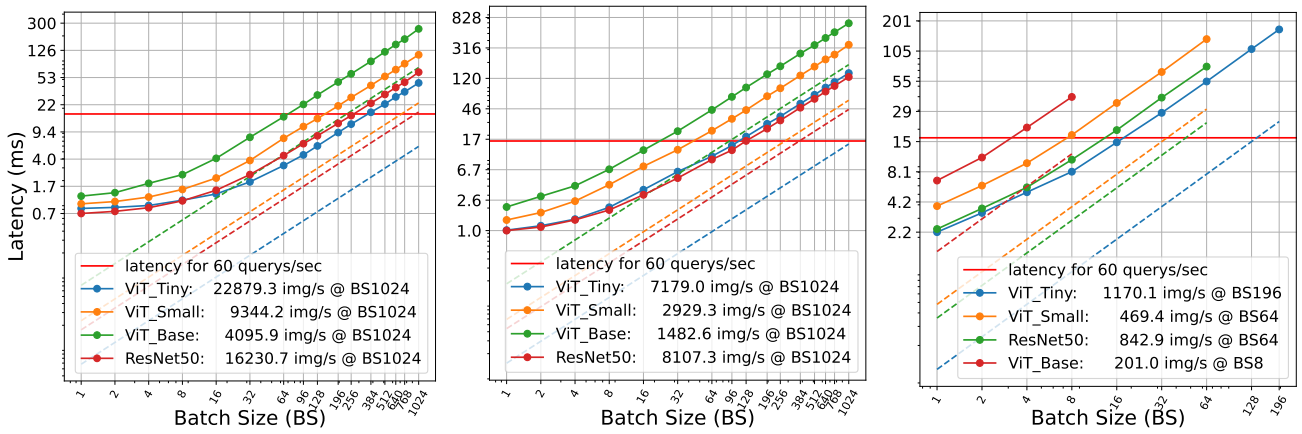
real-world inference. This gap can be narrowed through two primary mechanisms: increasing batch size, which enhances computational intensity, and deploying larger models, which similarly improves MFU.

Comparing architectures reveals interesting performance characteristics. While ViT-Small exhibits higher computational demand than ResNet50 (5.47 vs. 4.09 GFLOPs/image), ResNet achieves superior MFU. This indicates that CNN-based architectures like ResNet may be better optimized for the tested platform.

However, increasing batch size demonstrates diminishing returns: MFU improves gradually before eventually plateauing or triggering out-of-memory (OOM) conditions, particularly on resource-constrained devices such as the Jetson platform.

Latency represents another critical dimension of throughput, measuring the time required to process a batch request—essential for real-time applications. Under ideal conditions, latency scales linearly with batch size (dashed line in Fig. 6). However, low MFU at small batch sizes creates an initial nonlinear region (preceding the solid line), indicating computational underutilization.

As demonstrated in Fig. 6, the red line demarcates the 16.7ms threshold necessary to sustain 60 queries per second. The intersection with near-saturated performance defines an optimal operating region. On A100 hardware, this requires batch sizes exceeding 16; on V100, batch size 8 suffices. Jetson platforms offer considerably narrower operating margins—particularly for ViT-Tiny,



(a) A100 **(b) V100** **(c) Jetson**

Figure 6: Request Latency Vs. Batch Size Across Hardware Platforms For Multiple Models. The Dashed Lines Represent The Theoretical Latency, While The Solid Lines Indicate The Actual Latency Achieved By Each Model.

where MFU deteriorates below batch size 8, complicating the latency-utilization trade-off.

4.2 Preprocessing Performance

As shown in Fig. 7, we evaluate multiple preprocessing frameworks with distinct performance characteristics. PyTorch serves as the CPU-based baseline, exhibiting varying performance across datasets—likely attributable to differences in image encoding formats (e.g., TIFF vs. JPEG). OpenCV, employed specifically for the CRSA dataset with heavy CPU-bound operations, demonstrates poor performance in real-time scenarios and is therefore excluded from further evaluation. GPU-accelerated optimization for CPU-bound frameworks remains planned as future work.

DALI provides GPU-accelerated preprocessing capabilities. The numerical indicators 224, 96, and 32 represent output resolutions (e.g., 3×224×224), where larger dimensions require more intensive transformations. Since image loading and decoding costs remain constant, smaller output images (e.g., DALI 32) achieve faster preprocessing speeds. As transformation complexity dominates at higher resolutions (DALI 96, 224), performance differences across datasets converge. On the A100, Fruits-360 presents an anomalous outlier pattern that remains under investigation.

4.3 End-to-End Performance

End-to-end performance provides a comprehensive evaluation of overall system capability, as illustrated in Fig. 8.

On the A100 platform, larger models such as ViT-Base and ViT-Small benefit from effective preprocessing-inference latency overlap, achieving performance approaching the model engine's theoretical upper bound. Conversely, smaller models remain preprocessing-bottlenecked, particularly on platforms with limited preprocessing capabilities like the V100.

The resource-constrained Jetson platform exhibits inverted performance dynamics. Combined memory consumption from preprocessing and inference constrains the model engine's available batch size, which subsequently restricts concurrent request handling and further degrades performance. ViT-Base, possessing the highest memory requirements, demonstrates the most severe performance degradation, while remaining models exhibit comparable performance reductions.

5 Conclusion

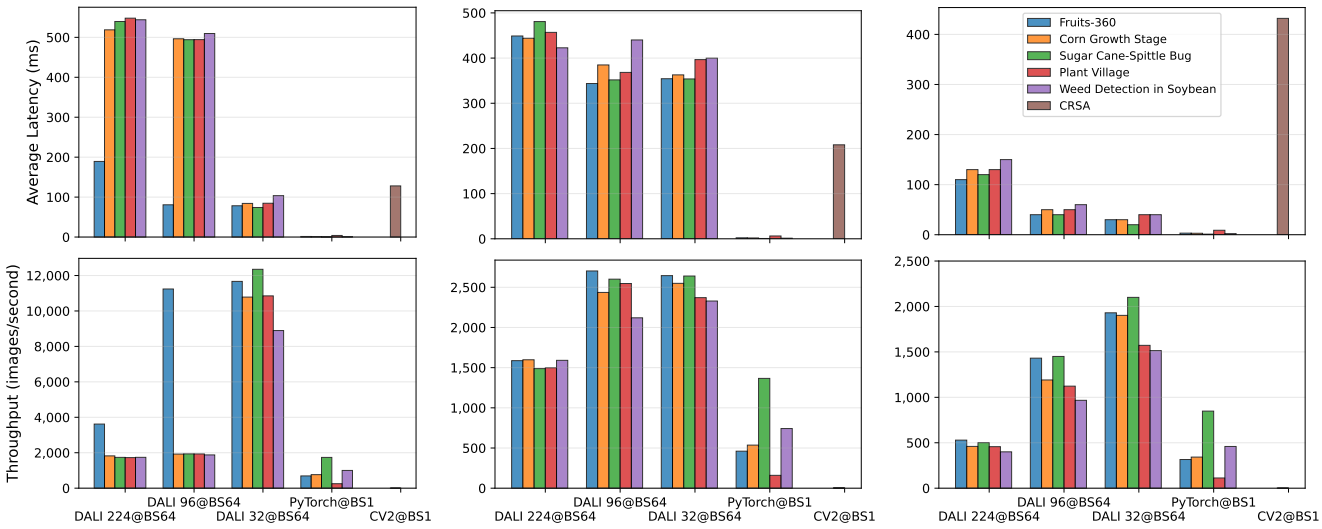
The proliferation of AI has empowered diverse digital agriculture applications, enhancing agricultural productivity in unprecedented ways. However, traditional AI training paradigms often overlook the highly variable deployment scenarios typical in agriculture, where models must be fine-tuned on farm-specific datasets and deployed across heterogeneous platforms.

This paper evaluates representative CNN and Transformer-based models across various sizes and datasets, deployed on both cloud and edge platforms. Through comprehensive performance characterization, we provide multi-level guidance—from model selection to end-to-end pipeline optimization—tailored to agricultural AI requirements.

Our analysis reveals a fundamental trade-off between throughput and batch size, forming a performance roofline constrained by either compute saturation or memory exhaustion. For smaller models, moderate batch sizes often suffice to utilize most platform capability and meet inference requirements. Beyond this threshold, increasing batch size yields diminishing returns, making multi-instance strategies more effective for improving responsiveness.

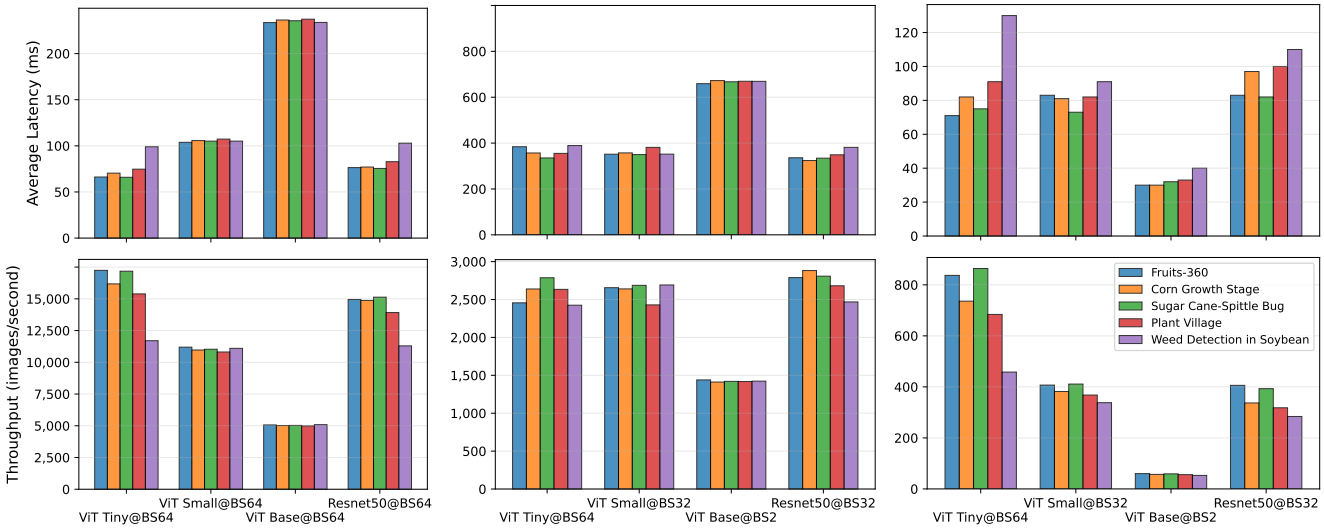
For high-resolution datasets, preprocessing operations (perspective transform, resizing, normalization) can become performance bottlenecks, particularly on resource-constrained edge platforms. GPU-accelerated preprocessing frameworks like NVIDIA DALI demonstrate significant speedups over traditional CPU-based pipelines, substantially enhancing system responsiveness.

Our characterization underscores the necessity of optimizing the specific dataset-model-platform combination for target applications, balancing latency requirements with energy efficiency and



(a) A100 (b) V100 (c) Jetson

Figure 7: Preprocessing Throughput And Latency For Different Datasets Across Platforms. Upper figure show request latency, lower show throughput across different preprocessing Methods. Batch Size varies by method.



(a) A100 (b) V100 (c) Jetson

Figure 8: End-To-End Pipeline Inference Latency And Throughput For Different Datasets Across Platforms. Upper figure show request latency, lower show throughput across different model. The largest Batch Size before Out-of-memory (OOM) was used.

memory utilization. Future work will develop comprehensive quantitative models for scalable performance prediction and provide deployment toolkits that enable practitioners to establish performance expectations before deployment, ultimately making digital agriculture more accessible and widely applicable.

Acknowledgments

This research is supported in part by NSF grants #2112606, #2311830, #2312927, #2415201, and XRAC grant #NCR-130002.

References

[1] [n. d.]. Mineral - A Google X Moonshot. https://x.company/projects/mineral/.
[2] 2025. Opencv/Opencv. OpenCV. https://github.com/opencv/opencv.
[3] 2025. OpenDroneMap/ODM. OpenDroneMap. https://github.com/OpenDroneMap/ODM.
[4] Fri, 07/09/2021 - 09:16. ICICLE: Intelligent CI with Computational Learning in the Environment. https://icicle.osu.edu/.
[5] Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. *Software available from tensorflow. org* (2016).

[6] Nawras Alnaasan, Anirudh Potlapally, Tian Chen, Matthew Lieber, Aamir Shafi, Hari Subramoni, Scott Shearer, and Dhabaleswar K Panda. [n. d.]. HARVEST-2.0: High-Performance Vision Framework for End-to-end Preprocessing, Training, Inference, and Visualization. ([n. d.]).

[7] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models Are Few-Shot Learners. http://arxiv.org/abs/2005.14165. arXiv:2005.14165 [cs] doi:10.48550/arXiv.2005.14165

[8] Antonio Bruno, Davide Moroni, Riccardo Dainelli, Leandro Rocchi, Silvia Morelli, Emilio Ferrari, Piero Toscano, and Massimo Martinelli. 2022. Improving Plant Disease Classification by Adaptive Minimal Ensembling. *Frontiers in Artificial Intelligence* 5 (Sept. 2022). doi:10.3389/frai.2022.868926 https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2022.868926/full.

[9] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, YenSung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi, Ross Girshick, Ali Farhadi, and Aniruddha Kembhavi. 2024. Molmo and PixMo: Open Weights and Open Data for State-of-the-Art Vision-Language Models. http://arxiv.org/abs/2409.17146. arXiv:2409.17146 [cs] doi:10.48550/arXiv.2409.17146

[10] Alessandro dos Santos Ferreira, Daniel Matte Freitas, Gercina Gonçalves da Silva, Hemerson Pistori, and Marcelo Theophilo Folhes. 2017. Weed Detection in Soybean Crops Using ConvNets. *Computers and Electronics in Agriculture* 143 (Dec. 2017), 314–324. doi:10.1016/j.compag.2017.10.027 https://www.sciencedirect.com/science/article/pii/S0168169917301977.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. http://arxiv.org/abs/2010.11929. arXiv:2010.11929 [cs] doi:10.48550/arXiv.2010.11929

[12] Madeleine Fairbairn, Hilary Oliva Faxon, Maywa Montenegro De Wit, Kelly Bronson, Zenia Kish, Sarah-Louise Ruder, Jane Ezirigwe, Selamawit Abdella, Chidi Oguamanam, and Matthew A. Schnurr. 2025. Digital Agriculture Will Perpetuate Injustice Unless Led from the Grassroots. *Nature Food* 6, 4 (March 2025), 312–315. doi:10.1038/s43016-025-01137-8 https://www.nature.com/articles/s43016-025-01137-8.

[13] Logan Frank, Christopher Wiegman, Jim Davis, and Scott Shearer. 2021. Confidence-Driven Hierarchical Classification of Cultivated Plant Stresses. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, Waikoloa, HI, USA, 2502–2511. doi:10.1109/wacv48630.2021.00255 https://ieeexplore.ieee.org/document/9423244/.

[14] Krishna Karthik Gadiraju, Bharathkumar Ramachandra, Zexi Chen, and Ranga Raju Vatsavai. 2020. Multimodal Deep Learning Based Crop Classification Using Multispectral and Multitemporal Satellite Imagery. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, Virtual Event CA USA, 3234–3242. doi:10.1145/3394486.3403375 https://dl.acm.org/doi/10.1145/3394486.3403375.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. doi:10.48550/ARXIV.1512.03385

[16] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. http://arxiv.org/abs/2309.06180. arXiv:2309.06180 [cs] doi:10.48550/arXiv.2309.06180

[17] TorchVision maintainers and contributors. 2016. TorchVision: PyTorch's Computer Vision Library. https://github.com/pytorch/vision.

[18] Horea Mureşan and Mihai Oltean. 2021. Fruit Recognition from Images Using Deep Learning. http://arxiv.org/abs/1712.00580. arXiv:1712.00580 [cs] doi:10.48550/arXiv.1712.00580

[19] NVIDIA. [n. d.]. NVIDIA TensorRT. https://developer.nvidia.com/tensorrt.

[20] NVIDIA. 2024. NVIDIA Data Loading Library (DALI). https://developer.nvidia.com/dali.

[21] NVIDIA Corporation. 2025. Triton Inference Server: An Optimized Cloud and Edge Inferencing Solution. https://github.com/triton-inference-server/server.

[22] ONNX Runtime developers. [n. d.]. ONNX Runtime. https://onnxruntime.ai/. [Online; Accessed 21-January-2023].

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[24] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, Xuzheng He, Haowen Hou, Jiaju Lin, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartlomiej Koptyra, Hayden Lau, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Bolun Wang, Johan S. Wind, Stanislaw Wozniak, Ruichong Zhang, Zhenyuan Zhang, Qihang Zhao, Peng Zhou, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. 2023. RWKV: Reinventing RNNs for the Transformer Era. http://arxiv.org/abs/2305.13048. arXiv:2305.13048 [cs] doi:10.48550/arXiv.2305.13048

[25] Dinesh Jackson Samuel, Inna Skarga-Bandurova, David Sikolia, and Muhammad Awais. 2025. AgroLLM: Connecting Farmers and Agricultural Practices through Large Language Models for Enhanced Knowledge Transfer and Practical Application. http://arxiv.org/abs/2503.04788. arXiv:2503.04788 [cs] doi:10.48550/arXiv.2503.04788

[26] Muhammad Habib ur Rehman, Ibrar Yaqoob, Khaled Salah, Muhammad Imran, Prem Prakash Jayaraman, and Charith Perera. 2019. The Role of Big Data Analytics in Industrial Internet of Things. *Future Generation Computer Systems* 99 (Oct. 2019), 247–259. doi:10.1016/j.future.2019.04.020 https://www.sciencedirect.com/science/article/pii/S0167739X18313645.

[27] Carlos Vasquez. 2020. Sugar Cane - Spittle Bug. https://data.mendeley.com/datasets/pwprmck9h5/1. doi:10.17632/PWPRMCK9H5.1

[28] Lucas Waltz, Sushma Katari, Chaeun Hong, Adit Anup, Julian Colbert, Anirudh Potlapally, Taylor Dill, Canaan Porter, John Engle, Christopher Stewart, Hari Subramoni, Scott Shearer, Raghu Machiraju, Osler Ortez, Laura Lindsey, Arnab Nandi, and Sami Khanal. 2025. Cyberinfrastructure for Machine Learning Applications in Agriculture: Experiences, Analysis, and Vision. *Frontiers in Artificial Intelligence* 7 (Jan. 2025), 1496066. doi:10.3389/frai.2024.1496066 https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2024.1496066/full.

[29] Steven Euijong Whang, Yuji Roh, Hwanjun Song, and Jae-Gil Lee. 2023. Data Collection and Quality Challenges in Deep Learning: A Data-Centric AI Perspective. *The VLDB Journal* 32, 4 (July 2023), 791–813. doi:10.1007/s00778-022-00775-9 https://link.springer.com/10.1007/s00778-022-00775-9.