CASC Newsletter | Vol 12 November 2022

In This Issue:

From the Director

Contact: Jeff Hittinger

"If you want to go quickly, go alone. If you want to go far, go together." – *African Proverb*

Lawrence Livermore National Laboratory is primarily thought of as a nuclear weapons laboratory. In truth, the mission space for LLNL is much broader, addressing national security writ large and encompassing basic scientific research in support of this national security mission. While the roots of CASC are deeply entwined with the Lab's stockpile stewardship mission, in more than a quarter century, CASC research has branched out across many of the other programs within the Lab. At the same time, basic research in CASC has grown thanks to active participation in the LDRD Program and the DOE Advanced Scientific Computing Research Program. CASC's reach extends beyond the Lab as well, with many CASC researchers actively participating and serving in leadership roles in the external computing community. As such, CASC serves as an envoy to academia and industry ensuring that the interests of LLNL programs are well represented and helping to establish LLNL's external representation as a leader in HPC.

In this edition of the CASC Newsletter, we highlight several examples of the broader influence that CASC has beyond (yet often still to the benefit of) the core nuclear weapons mission. The MFEM high-order finite element library, originally developed in collaboration with Weapons and Complex Integration scientists, has seen dramatic uptake in the simulation community, and we provide a recap of community workshops sponsored by the MFEM team. The RAJA loop abstraction capability, which arose from a collaboration between CASC and Computing's Applications, Simulations, and Quality Division, has proven to be an effective approach to performance portability, and CASC members of the RAJA team have been actively working with compiler vendors to ensure

that the C++ language features on which RAJA depends are fully supported with high performance implementations. In a related area, programming languages, such as C and C++, and widely used programming abstractions, like MPI and OpenMP, are improved and extended by standards committees, and several CASC researchers represent Lab interests on the standards committees of greatest importance to HPC. Finally, we feature data science applications within the broader national security community to which CASC researchers are making novel contributions. We hope that you enjoy these vignettes as an informative glimpse into the many ways that CASC is making a beneficial impact across the Lab and within the external community.

Collaborations | A New Tradition: MFEM Community Workshop

Contact: Aaron Fisher and Tzanio Kolev

This October, a team of CASC researchers hosted the second annual workshop for the MFEM user and developer community. The event is designed to promote collaboration, describe the MFEM library's latest features, expand application engagements, and solicit feedback to guide future development. More than 150 researchers from dozens of organizations and countries attended each of the one-day virtual workshops.

MFEM is a C++ software library that provides advanced finite element discretization methods to many HPC applications across the DOE, academia, and industry. The large-scale scientific simulations powered by MFEM include magnetic and inertial confinement fusion, compressible and incompressible flows, additive manufacturing, topology and shape optimization, nuclear reactor modeling, structural mechanics, and more. The project's discretization methods enable HPC systems to run these simulations more efficiently. The DOE's Exascale Computing Project (ECP) and Office of Science SciDAC Program both include MFEM in their strategic software portfolios.

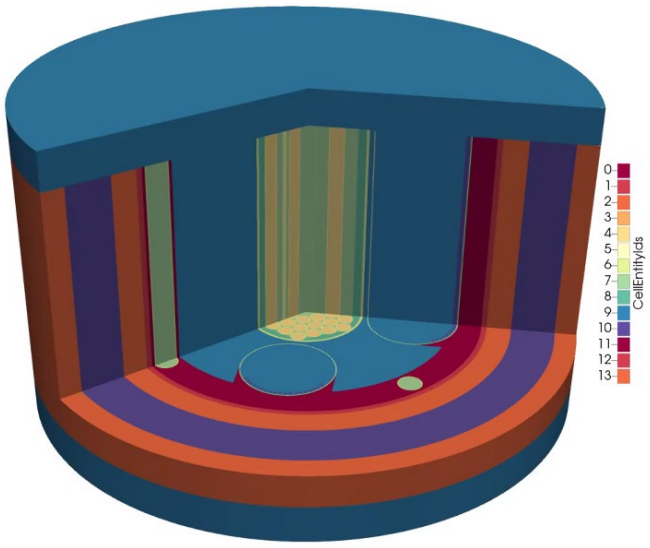

Figure 1. *North Carolina State University researchers are using MFEM in neutron transport modeling. The 2D MARVEL reactor model runs on Oak Ridge National Laboratory's Summit supercomputer, and a 3D model, like the one shown here, is in development. Image courtesy of William Dawn.*

As open-source software, MFEM is readily available to researchers outside of LLNL, and most of the workshops' talks were given by external collaborators and users. For example, William Dawn from North Carolina State University described his work with unstructured neutron transport. His team models micro-reactors, a new class of compact reactor with relatively small electrical output. As part of the ECP, Dawn's team is modeling the MARVEL reactor, which is planned for construction at Idaho National Laboratory. MFEM satisfies their need for a finite element framework with GPU support and rapid prototyping. With MFEM, the team discretizes a neutron transport equation with six independent variables in space, direction, and energy (see Figure 1). Dawn noted, "MFEM has been a game-changer for our project. Being able to start the day with a new idea for a solver and finish with a new implementation that can run on the Summit supercomputer is huge for us."

Mathias Davids from Harvard Medical School demonstrated MFEM's use in a medical setting. Magnetic resonance imaging (MRI) generates anatomical and physiological images via non-ionizing electromagnetic waves. Davids explained how MRI machines use magnetized gradient coils to perform image encoding in MRI. Harvard researchers are investigating MRI-induced peripheral nerve stimulation (PNS), which can cause pain or muscle contractions during the scan and limit the achievable spatial and temporal image resolution (see Figure 2). Davids noted, "MFEM plays a central role in our efforts to develop these PNS models." The high-spatial-resolution models contain electromagnetic pathways for 2,000 nerve segments in the human body. MFEM provides the solvers to render electric field patterns on hexahedral meshes made up of 80 million mesh elements.

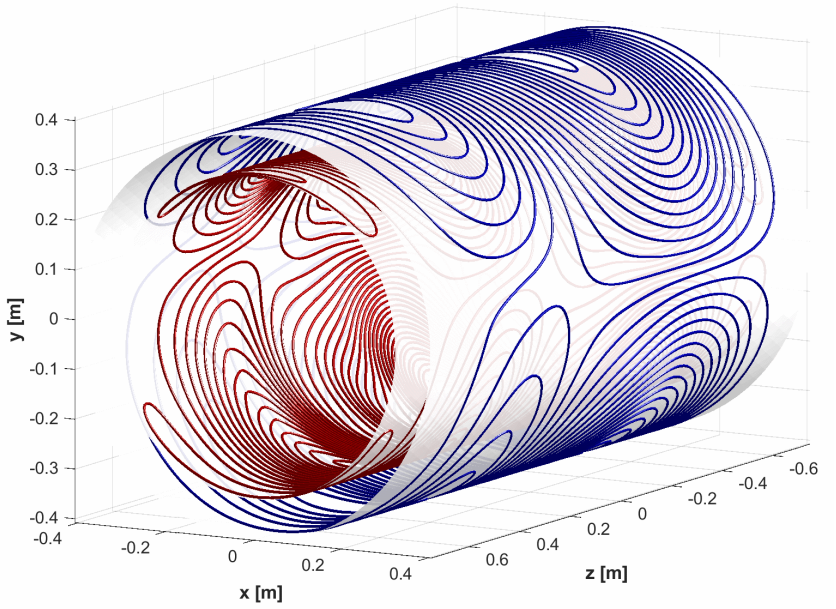

Figure 2. *Harvard Medical School researchers are using MFEM in predictive models that investigate peripheral nerve stimulation (PNS). The results can help optimize gradient coils (shown in red and blue) in MRI machines to reduce PNS effects. Image courtesy of Mathias Davids.*

The workshops also showcased unique applications developed at LLNL, where many code teams rely on MFEM's finite element discretization, mesh partitioning, and other features. For instance, Jorge-Luis Barrera from the Computational Engineering Division discussed the Livermore Design Optimization (LiDO) code, which solves optimization problems for a wide range of Lab-relevant

engineering applications. Leveraging MFEM and the LLNL-developed engineering simulation code Serac, LiDO delivers a powerful suite of design tools that run on HPC systems. Barrera highlighted several design examples that benefit from LiDO's integration with MFEM, including multi-material geometries, octet truss lattices, and a concrete dam under stress (see Figure 3). LiDO's graph architecture that seamlessly integrates MFEM features ensures robust topology optimization, as well as shape optimization using nodal coordinates and level set fields as optimization variables.

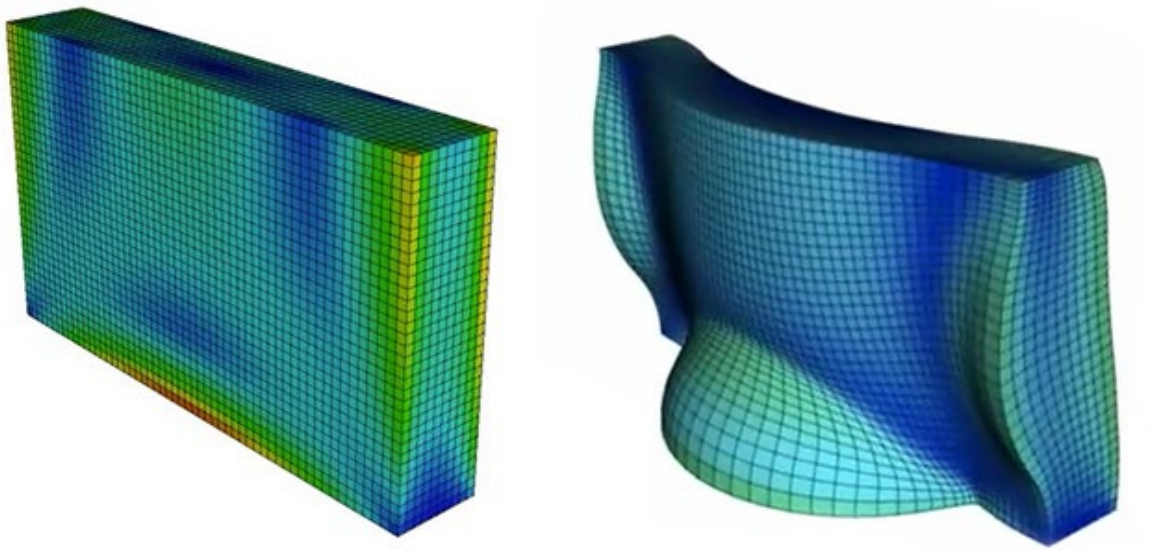

Figure 3. *MFEM enables the LiDO code to perform gradient-based systematic design optimization, as in this example of stress minimization on a concrete dam. Updating the mesh coordinates generates optimal designs that experience large shape changes compared to their initial configuration. Image courtesy of Jorge-Luis Barrera.*

Additionally, the workshops give the MFEM team a chance to explain key features to newer or inexperienced users, as well describe enhancements planned for upcoming releases. The 2022 workshop coincided with the version 4.5 release, which includes container and cloud support, improved GPU algorithms, submesh extraction, better meshing and solvers, and a fractional PDE example. The team is already preparing MFEM for LLNL's El Capitan and other heterogeneous exascale supercomputers. Beyond the roadmap and technical talks, a Q&A session and an active Slack channel offer participants a forum for discussing implementation issues, making suggestions, and learning more from the team in real time. Noting that half of the latest workshop's participants were outside the U.S., CASC group leader Aaron Fisher stated, "MFEM is a global community, and mathematics is a language that crosses borders easily."

Learn more about the workshops on the MFEM website, where speakers' slides and videos are posted. Winners of the workshops' visualization contest are featured in an online gallery.

Lab Impact | Compiler Co-Design with the RAJA Team

Contact: Rich Hornung and David Beckingsale

Programming abstractions like RAJA, which support fine-grained, on-node parallelism in HPC applications, are typically designed to enable execution on a variety of hardware architectures with single-source application code. This means that the abstractions insulate application source code from implementation details associated with different parallel programming models, such as OpenMP and CUDA. However, modern C++ abstractions can hinder application performance by introducing software complexities that can make it difficult for compilers to optimize. To minimize such penalties, the RAJA team has worked with compiler vendors for the past decade to improve RAJA, specifically, and C++ compilers, benefitting the broader community of C++ HPC application developers.

LLNL has a long history of developing compiler benchmarking suites. The most prominent example is the Livermore Fortran Kernels (LFK) benchmark developed by Frank McMahon in the 1980s to assess Fortran compiler vectorization. Since then, versions of this benchmark have been developed in other programming languages, such as C and C++. The C++ version, called LCALS (Livermore Compiler Analysis Loop Suite), was developed around 2010 to assess compiler support for modern C++ language features. LCALS developers worked with compiler vendors, such as Intel and GNU, over multiple major compiler releases to improve compiler optimization support for C++ template and lambda expression features, on which RAJA would eventually be based.

Today, the RAJA Performance Suite, developed during the LLNL Sierra platform procurement, is a key tool for the Lab's interactions with compiler vendors and GPU vendors, such as NVIDIA and AMD, for current LLNL supercomputers. The RAJA Performance Suite supports a wide range of compiler assessment studies and contains over 70 numerical kernels representing important algorithm patterns in HPC applications [1]. Each kernel appears in both "native" and RAJA variants for multiple programming models, including CUDA, HIP, and OpenMP. The kernels in the Suite serve as reproducers of performance issues observed in applications and verification tests for new compiler features needed by HPC application developers.

RAJA developers use the Suite to assess new compiler releases to verify that reported issues are resolved and to understand the impact of new issues on applications. Likewise, vendors exercise the Suite as part of their testing of pre-release compilers. The RAJA team employs an iterative, co-design approach when working with vendors to help ensure that the needs of RAJA application users are met (see Figure 4). Such close collaborations are fundamental engagements in Centers of Excellence established for the DOE CORAL and CORAL-2 and other platform procurements. The CORAL procurement produced the Sierra system, currently LLNL's largest supercomputer, and CORAL-2 will result in the deployment of the El Capitan system at

LLNL in 2023. El Capitan will be the first exascale system dedicated to national security applications.

RAJA is primarily supported by and focused on the LLNL Advanced Simulation and Computing Program. RAJA is also supported along with the Kokkos effort at Sandia National Laboratories in a joint DOE ECP software technology project. The RAJA team collaborates closely with the Kokkos team to drive vendor support for common DOE application needs. As the RAJA Performance Suite grows to include more examples of HPC algorithm patterns, its compiler support becomes more robust, which enables RAJA to be a production-capable tool for a wider class of application codes.

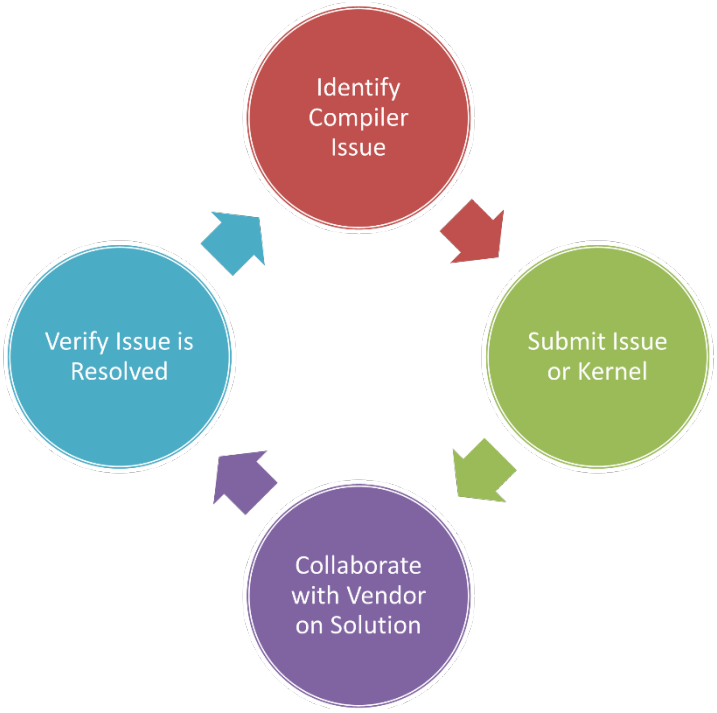

Figure 4. *The RAJA team's collaboration cycle with compiler vendors.*

[1] D. Beckingsale, J. Burmark, R. Hornung, H. Jones, W. Killian, A. Kunen, O. Pearce, P. Robinson, T. Scogland. "RAJA: Portable Performance for Large-Scale Scientific Applications." *IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC)*, 2019.

Advancing the Discipline | Improving HPC Standards with Committee Service

Contact: Tom Scogland, Ignacio Laguna, and Kathryn Mohror

An often overlooked but nevertheless important service that CASC researchers provide to the broader HPC community is their contribution to various HPC standards committees. There is tremendous value in standardization because it provides portability and performance with reduced code development, porting, and maintenance. Serving on these HPC standards committees not only ensures the Lab's issues and needs are part of the discussion, but also allows LLNL lessons learned to be shared with the rest of the HPC community.

CASC researcher Tom Scogland serves as the LLNL representative to the International Standards Organization (ISO) Standardization Subcommittee (SC) 22 programming languages, specifically on Working Group (WG) 21 C++ and 14 C. WG14 guides the standardization and growth of C, the

lingua franca of interfaces and common base language for systems software and kernel implementation. Given the stability of C, the focus from a DOE perspective usually is on ensuring that interfaces for atomics and parallelism are compatible with our needs and, especially, compatible and interoperable with C++, which is heavily used for scientific computing at LLNL. WG21, the C++ committee, is much larger and faster moving; there are in-flight proposals for multidimensional arrays, linear algebra, strongly-typed units with performant conversions, and a panoply of others. A proposal of note is for the representation of parallel execution backends, including for offload to GPUs, which could greatly improve our access to vendor support for models like RAJA and Kokkos.

CASC researcher Ignacio Laguna serves as the LLNL representative in the standardization forum for the Message Passing Interface (MPI), which is the standard abstraction for message-passing communication in HPC systems. The MPI standard defines the syntax and semantics of library routines for a wide range of users who write programs in C, C++, and Fortran using a message-passing paradigm for distributed parallel computing. As part of the development of MPI, several working groups have been established to discuss the standardization of various aspects of MPI, including collective communication, fault tolerance, accelerators, hardware topologies, sessions, tools, and others. Ignacio's role in the MPI Forum includes the standardization of a fault tolerance interface for MPI. As of the recent version (v4.0), MPI has almost no capabilities to survive failures. HPC systems, particularly large-scale systems, can suffer from process and node failures that can negatively impact scientific simulations. Ignacio is proposing a new interface for MPI, called Reinit, that will allow applications to recover from failures using checkpoint/restart without restarting the job, thus making possible rapid recovery with few modifications to the MPI application.

Tom and Ignacio both serve on the OpenMP interface committee. The OpenMP API was originally developed as an abstraction for multithreaded, on-node concurrency and today defines a simple and flexible interface for developing parallel applications on HPC platforms. Tom serves as the chair of the Accelerator Subcommittee, which is responsible for features used to target GPUs and other accelerators, and has driven features for offload, memory management, and foreign runtime interoperability since 2013. The focus of his work is to ensure that OpenMP can serve as a cross-language, parallel interface covering C, C++, and Fortran, which will allow legacy applications to incrementally incorporate the ability to offload without having to change their base language. Ignacio serves on the Tools Subcommittee, which develops OMPT, the interface for profiling, as well as OMPD, the interface for enhanced debugger information for OpenMP. This committee focuses on making it possible for debuggers and profilers to present OpenMP information rather than the low-level implementation details used to make the OpenMP runtime itself, so users can reason about their program the way they wrote it.

CASC researcher Kathryn Mohror serves on the PMIx interface committee. PMIx is an API that facilitates the interaction of tools and middleware with system software, which is critical for ensuring portability of tools and middleware across HPC platforms. In recent years, the PMIx community recognized the need for a formal body to transform the API into a standardized specification document so that it could be more readily adopted by HPC vendors and be more easily included in HPC system procurement requirements—ultimately providing broader community benefit. Kathryn has served as a co-chair of the PMIx Administrative Steering Committee (ASC) since its inception in 2019. Because she was strongly motivated by the need for a portable system software interface for her other work, e.g., the Scalable Checkpoint Restart Library, she was heavily involved in spinning up the ASC. The PMIx ASC is responsible for describing the procedures for adopting changes to the PMIx interface, which provides stability and usability of the interface across platforms and implementations of PMIx. The PMIx ASC currently includes 14 member organizations from government, industry, and academic institutions in the U.S. and Europe. Although it is a new interface, PMIx is gaining traction and has support in ECP and in European HPC efforts.

Machine Learning & Applications | What Is AI/ML Doing for National Security?

Contact: Wesam Sakla

Artificial intelligence (AI) and machine learning (ML) have become familiar terms in the global technology space. New AI-based startups are born daily, with goals of developing and harnessing computer vision and natural language processing (NLP) algorithms to add value in the automotive (self-driving cars), medical, manufacturing, automation, robotics, and social media sectors, to name a few. The Lab has made a concerted effort as part of its Science & Technology vision to invest heavily in AI/ML and data science across all mission spaces. As one example, researchers and data scientists from Computing are designing and prototyping AI/ML algorithms based on deep neural networks for national security. This work helps ensure that the U.S. maintains a competitive advantage and technological superiority for intelligence, surveillance, and reconnaissance (ISR) for our military as well as provides indications and warnings to aid in automated threat detection.

Remote sensing is the process of detecting and monitoring the physical characteristics of an area by measuring its reflected and emitted radiation from a distance, typically from a satellite or aircraft, using sensors with different imaging modalities. These sensors collect imagery and video in different regions of the electromagnetic spectrum, providing complementary information about the nature of objects and materials within a scene. Deep learning–based ML algorithms can accurately and quickly perform common computer vision tasks, including detection, classification, and identification of

objects in remotely sensed imagery; these are crucial for national security ISR applications. Challenges in this setting include processing input imagery with millions of pixels, a low number of spatially resolved pixels of objects, dramatic changes in viewing geometry and scene illumination, occlusion, varying backgrounds, and limited labeled data. Automating such tasks using AI/ML allows for a human-in-the-loop workflow, whereby the ML algorithms can quickly triage massive volumes of remotely sensed data at scale. Imagery and video deemed interesting facilitates further inspection by analysts, freeing up their time for downstream tasks that require higher-level analysis and reasoning.

Object detection is a common computer vision task that requires the simultaneous classification and localization of every object present in an image. Object localization refers to identifying the location of one or more objects in an image and drawing bounding boxes around their extent. Conventional methods for training object detectors are fully supervised, requiring dense annotations of bounding boxes around all objects of interest for every image in the training set. In practical scenarios, such dense

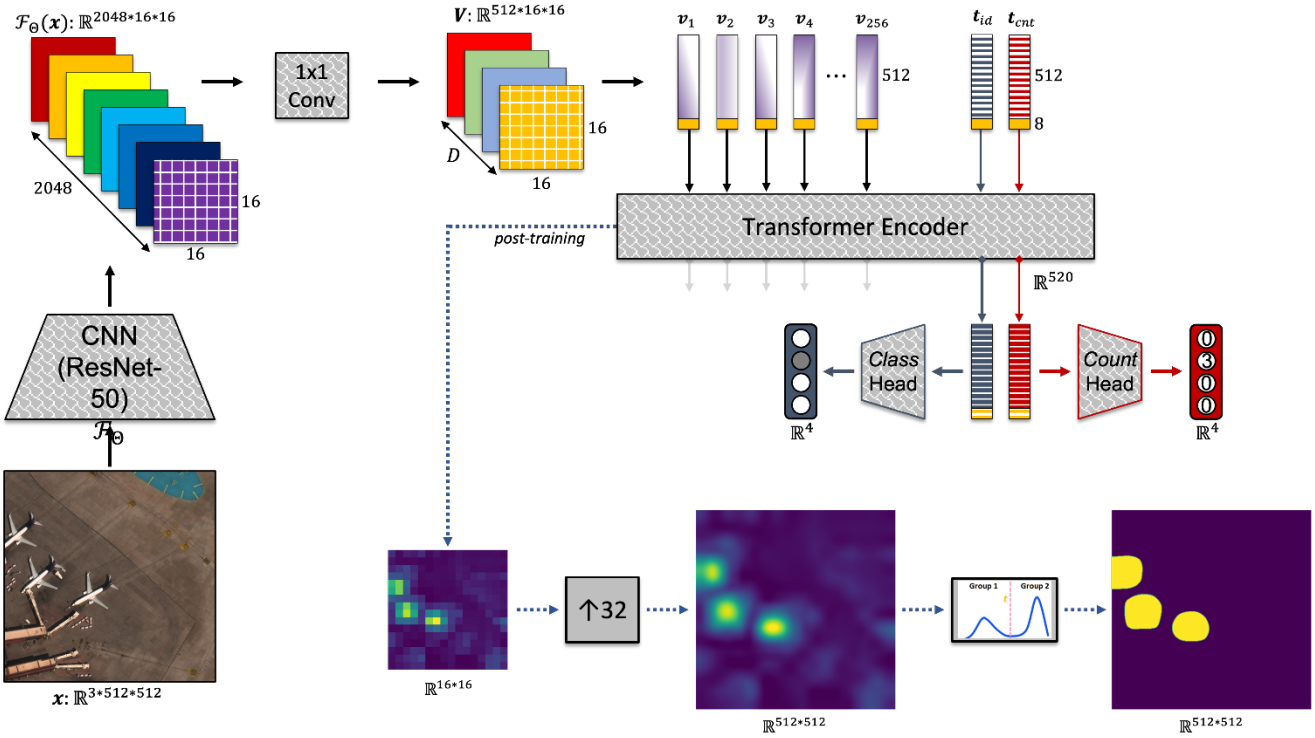

Figure 5. *A block diagram of the multi-task weakly supervised object detection architecture. During inference, an image is passed through a convolutional neural network to extract features. These features are appended with learned class ID and count tokens that are input to a transformer encoder and used to simultaneously predict the class labels and corresponding counts in the image. The learned self-attention is extracted from the transformer encoder and post-processed to provide a heatmap that localizes the predicted objects in the image.*

annotations are infeasible, either due to the lack of resources required to produce these annotations or the data having been annotated only at the image level for a different purpose when the originally collected. The goal of weakly supervised object detection is to train an ML model using only image-level labels, yet still provide localization capability with a reasonable level of accuracy.

CASC researcher Wesam Sakla recently led a team to develop and prototype a weakly supervised model for a national security customer. The model was trained using an open-source dataset of overhead imagery from airports around the world, acquired by commercial satellites. The model was supervised using class labels and corresponding counts of aircraft in the imagery. Additionally, it was trained with a multi-task objective to predict (1) the presence of four types of aircraft styles in satellite imagery: civilian passenger aircraft of three different sizes (small, medium, large) and military aircraft; and (2) their corresponding counts. To simulate a realistic scenario where the count label information was not available for all the data of interest, researchers trained the surrogate model by excluding a large quantity of the count labels and observed minor degradation in the corresponding metrics. The model, shown in Figure 5, also included a novel attention-based module known as a transformer encoder—a state-of-the-art component in language models in the NLP domain. After the model is trained, the self-attention weights inside the transformer encoder are post-processed to provide localization information about objects in the imagery during inference. This example is merely one of many that shows how data associated with problems and missions within the national security community present unique challenges that require further innovation beyond simply applying conventional AI/ML algorithms to a problem.

CASC Newsletter Sign-Up

Was this newsletter link passed along to you? Or did you happen to find it on social media? Sign up to be notified of future newsletters.

*This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL-WEB-842411. E*dited by Ming Jiang.