Efficient Computer Vision: Pushing the Frontier of Edge Computing

Michael Krah, Zach Gentile, and Alex Lavaee

April 6, 2025

Abstract

The Low Power Computer Vision Challenge (LPCVC) emphasizes the importance of optimizing models for efficient deployment on edge devices. Low-power, efficient computer vision is critical for applications requiring real-time processing with limited energy and hardware resources, such as mobile and IoT devices. Efficient models enable widespread deployment, reduce energy consumption, and enhance privacy by processing data locally rather than relying on cloud-based solutions. Inspired by LPCVC, our project investigates advanced optimization methods tailored for edge devices. We explore architectural improvements including quantization and token-efficient approaches, alongside training enhancements such as 8-bit General Matrix Multiplication (GEMM) and Multi-head Latent Attention (MLA). Our solution aims to be lightweight, efficient, and optimized for direct execution on edge devices with limited hardware.

Introduction

Computer vision at the edge represents a critical advancement in artificial intelligence, enabling real-time image analysis on resource-constrained devices without reliance on cloud connectivity. The 2025 IEEE Low-Power Computer Vision Challenge (LPCVC) highlights the importance of optimizing computer vision models for deployment on low-power, edge-based hardware, focusing specifically on image classification across varying lighting conditions and artistic styles. Ideally, smaller models could be run locally on portable devices to assist in real-time classification.

While our project does not directly participate in LPCVC, it is deeply inspired by the competition's goals and methodologies. Our work seeks to explore and extend the competition's emphasis on efficient, robust, and lightweight models tailored for edge deployment. Specifically, we aim to address similar challenges outlined by LPCVC by investigating innovative neural network architectures and optimization strategies, including quantization,

token-efficient approaches, and specialized training methods such as 8-bit General Matrix Multiplication (GEMM) and Multi-head Latent Attention (MLA).

Our objective is to advance state-of-the-art practices in efficient computer vision, balancing the trade-off between classification accuracy, inference speed, and robustness to diverse visual conditions. By drawing inspiration from LPCVC's structured approach to benchmarking, we aim to develop solutions beneficial for researchers and developers working within the constraints of limited computational resources, with potential applications spanning augmented reality, autonomous systems, and smart devices. Additionally, we prioritize creating comprehensive documentation to ensure our methods are accessible, well-explained, and easy to replicate. This documentation is intended to guide future users, researchers, and practitioners interested in learning, implementing, and extending the optimization strategies and architectures explored in this project.

Related Work

Efficient computer vision for edge devices has seen significant advancements across several key areas. We organize our literature review around three main themes particularly relevant to the LPCVC challenge:

Model Compression and Quantization

Neural network compression has become crucial for deploying deep learning models on resource-constrained devices. Work by Han et al. [5] introduced deep compression, combining pruning, quantization, and Huffman coding to significantly reduce model size while maintaining accuracy. Building on this foundation, Jacob et al. [10] developed quantization-aware training techniques that allow for 8-bit integer operations without significant accuracy degradation. More recently, Nagel et al. [12] demonstrated data-free quantization methods that enable post-training conversion to lower precision formats, making deployment more accessible.

Architectures for Edge Deployment

Specialized neural network architectures designed explicitly for edge devices have shown remarkable efficiency. MobileNetV2 [14] and MobileNetV3 [9] introduced inverted residuals and linear bottlenecks that dramatically reduced computational costs. EfficientNet [15] proposed a principled scaling method that optimizes depth, width, and resolution dimensions simultaneously. Most relevant to our work, Lin et al. [11] developed MCUNet for ultra-small microcontrollers, demonstrating techniques to adapt neural networks to extremely limited hardware.

1D Image Tokenization

Recent advancements have explored alternative image representation methods, notably the work by Yu et al. [17] which introduced TiTok, a transformer-based 1-Dimensional tokenizer. TiTok significantly reduces image representations to compact 1D latent sequences, achieving efficient tokenization using as few as 32 tokens for a 256×256×3 image. This approach effectively handles redundancies in images, significantly improving computational efficiency in image reconstruction and generation tasks. Despite its compact form, TiTok outperforms state-of-the-art models on benchmarks like ImageNet, providing faster generation speeds and competitive generative quality.

Current Work and Preliminary Results

Our current model leverages a token efficient vision transformer, TiTok [17], as a base model for image classification. We have recreated the paper's pretraining and training pipeline to reproduce established results and finetune the model for classification.

- **Dataset Implementation:** ImageNet was loaded to the SCC and then converted to the WebDataSet format. Images were then tokenized using MaskGIT to generate "proxy codes" to be later used in training[1], following the training strategy implemented by [17]. This gave us access to more than 1 million images on the SCC for training.
- **Pretraining:** TiTok split training for the encoder and decoder into two distinct phases. Encoder training uses the "proxy code" tokens generated by MaskGIT as a desired output, ensuring stability of the model.
- **Linear Probe:** To verify TiTok's results and for a classification pipeline, we implemented linear probing. We followed the approach of [7], modifying their given code to use the Webdataset format and to be compatible with the TiTok model. To obtain preliminary results, we have added a linear layer to TiTok's encoder and frozen all layers except for the head.

Proposed Work

Our solution inspired by 2025 LPCVC Track 1 challenge builds on state-of-the-art techniques in efficient deep learning and leverages insights from recent low-power computer vision studies [2]. Our approach combines algorithmic innovations with hardware-specific optimizations to achieve superior accuracy and speed on edge devices.

Model Architecture and Optimization

We begin with a lightweight backbone based on TiTok [17], enhanced with attention mechanisms inspired by transformer architectures [4, 16] and efficient mobile designs [13]. Our approach includes:

- **Token-Efficient Vision Transformers:** We incorporate token-efficient strategies to further reduce computational overhead. Inspired by [17], our design represents images using only 32 tokens, drastically reducing the input dimensionality.
- **Mixed-Precision Training:** We implement dynamic quantization that maintains full precision for critical parameters while quantizing less sensitive components. This approach is guided by prior work on efficient integer-arithmetic-only inference [10].
- **8-bit Matrix Multiplication:** Our model utilizes 8-bit integer matrix multiplication (Int8 MatMul) with calibrated scaling factors to minimize quantization error, reducing memory and computational complexity by approximately 75% compared to FP32 operations [10].

Hardware-Specific Optimizations

To fully leverage hardware capabilities, we propose several hardware-specific enhancements:

- **Custom CUDA Kernels for GPU Optimization:** We will develop custom CUDA kernels tailored for NVIDIA GPUs, leveraging parallel processing capabilities to optimize convolutional operations. This approach aligns with recent advancements in GPU-based hardware acceleration and software optimization, maximizing computational efficiency and performance. [3].
- **GEMM and Multi-Head Latent Attention (MLA):** In addition to custom kernels, we integrate high-performance General Matrix Multiplication (GEMM) routines alongside novel Multi-head Latent Attention (MLA) mechanisms, as introduced in [3]. GEMM optimizations streamline batched computations, while MLA enhances the attention mechanism by leveraging latent representations, thus reducing computational overhead and improving feature extraction.

Robustness to Image Variability and Underrepresented Classes

Given the variability that is inherent to photographing and classifying images, we aim to test the robustness of our model in predicting underrepresented classes or images that may be taken in variable conditions:

- **Data Augmentation Strategy:** A comprehensive augmentation pipeline will simulate diverse lighting conditions and artistic styles during training, enhancing the

model's generalization.

- **Domain Adaptation Techniques:** Adversarial domain adaptation components are incorporated to minimize feature distribution shifts between different lighting and style domains.

To score our final model's performance, we hope to adapt the LPCVC's formula for calculating score [2]. This will allow of to balance accuracy and speed when determining the performance of the model. We also plan to use an F1-score to determine the model's overrall performance, especially in regards to underrrepresented classes.

$$\text{Score} = \frac{\text{Accuracy}}{\max(\text{ExecutionTime}/2, 1\text{ms})}. \tag{1}$$

Datasets

Our project leverages the original ImageNet dataset for training, with comprehensive evaluation conducted using the ImageNet, ImageNet-C, and ImageNet-P datasets. These datasets collectively enable robust assessment of both in-distribution and out-of-distribution performance.

Dataset Composition

The ImageNet dataset comprises approximately 1.28 million training images across 1,000 diverse object classes, ranging from animals and plants to household and industrial items.

Corruptions and Perturbations (ImageNet-C and ImageNet-P)

For evaluating model robustness to common corruptions and perturbations, we employ ImageNet-C and ImageNet-P, two benchmarks introduced by Hendrycks and Dietterich [8].

ImageNet-C contains validation images corrupted by 15 diverse types of visual corruptions across four categories:

- **Noise:** Gaussian, shot, impulse noise
- **Blur:** defocus, frosted glass, motion, zoom blur
- **Weather:** snow, frost, fog, brightness variations
- **Digital:** contrast changes, elastic transformations, pixelation, JPEG compression artifacts

Each corruption type is presented at five severity levels, enabling precise measurement of robustness across varying degrees of visual distortions.

ImageNet-P evaluates perturbation robustness through sequences of images subjected to subtle transformations. This benchmark captures the stability of model predictions under minute perturbations, including:

- Gaussian and shot noise perturbations
- Motion and zoom blur perturbations
- Weather-based perturbations (e.g., snow, brightness changes)
- Spatial perturbations (translations, rotations, viewpoint tilts, scaling)

Each perturbation sequence consists of multiple frames with incremental transformations, measuring how consistently the model maintains correct classifications.

Dataset Format and Preprocessing

All images from these datasets are standardized to RGB format at a resolution of 224×224 pixels. The input tensors are structured with dimensions (batch, 3, 224, 224) and pixel values normalized to the [0, 1] range as floating-point values. Our preprocessing pipeline includes:

- Data normalization using ImageNet's mean and standard deviation values
- Data augmentation through random cropping, horizontal flipping, and color jittering
- Additional augmentation techniques to simulate corruption and perturbation scenarios

Dataset Split and Evaluation

For evaluation and testing we will be using ImageNet datasets. The base Titok model was trained on ImageNet and we plan to use the same dataset to refine and evaluate the model. Evaluation will be done on Imagenet-C/P as well as the original ImageNet.

The variety of classes in ImageNet and the modifications provided in ImageNet-C and Imagenet-P provide a wide range of test data to train that can be used to evaluate the adaptability of the model.

Evaluation

Our evaluation methodology follows the official metrics for the 2025 LPCVC Track 1 challenge while incorporating additional analyses to guide our development process.

Primary Metrics

In accordance with the goals initially established by the challenge, our primary evaluation metrics are:

- **Execution Time:** Our solution will seek to minimize CPU and GPU inference time. To have been considered valid for the LPCVC competition, we would have to execute within 10ms per image on the Snapdragon 8 Elite QRD platform .
- **Classification Accuracy:** The percentage of correctly classified images across all 1000 classes in ImageNet.

We plan to use classification results from the TiTok model [17] as a baseline and work to reproduce and improve this score.

Development Evaluation Strategy

During the development process, we will implement a more comprehensive evaluation approach:

- **Validation Set Evaluation:** We will measure performance on the validation representing corrupted and perturbed versions of ImageNet.
- **Confusion Matrix Analysis:** We will analyze the confusion matrix to identify frequently misclassified class pairs and target these specifically in our model refinement.
- **Performance Profiling:** We will profile memory usage, CPU/GPU/DSP utilization, and power consumption to guide our optimization efforts.

Baseline Comparisons

We will establish the following baselines for comparison:

- **Standard MobileNetV2:** We will evaluate a standard MobileNetV2 model to understand the improvements offered by our specialized approach.
- **TiTok-Tokenizer:** As we build upon this architecture, we will track improvements over the original implementation.

Timeline

We will structure our project with the following timeline:

Phase 1: Research and Exploration (Week 1-2, March 2-16)

- Literature review of efficient vision models and optimization techniques

- Find base architecture to iterate on
- Setup development environment

Phase 2: Development and Optimization (Weeks 3-7, March 17-April 13)

- Convert Imagenet dataset to WebDataSet format
- Run Pretokenization to generate proxy codes for TiTok model training
- Implementation of core model architecture (TiTok), run Linear Probing and achieve baseline
- Exploring model hyperparameters
- Comprehensive evaluation and performance tuning
- Explore quantization, 8-bit General Matrix Multiplication, and Multi-head Latent Attention

Phase 3: Refinement and Submission (Week 8-10, April 14-30)

- Continue refining algorithm, model, and hyperparameters
- Final model benchmarking and validation
- Documentation preparation and code cleanup
- Write-up sharing our insights and common techniques to increase inference speed
- Create demo of model running on iPhones using the MLX framework (if time) [6]
- Preparation of final project paper

Conclusion

This document outlines our approach inspired by the 2025 IEEE Low-Power Computer Vision Challenge - Track 1, focusing on efficient image classification robust to lighting variations and artistic styles. Our strategy leverages token-efficient methods alongside planned implementations of quantization, Multi-head Latent Attention, 8-bit General Matrix Multiplication. and optimized CUDA kernels to achieve both high accuracy and low inference latency, thereby enhancing the accessibility and practicality of deep learning models on edge devices.

References

[1] Huiwen Chang et al. *MaskGIT: Masked Generative Image Transformer*. 2022. arXiv: 2202.04200 [cs.CV]. URL: https://arxiv.org/abs/2202.04200.

[2] Leo Chen et al. *2023 Low-Power Computer Vision Challenge (LPCVC) Summary*. 2024. arXiv: 2403.07153 [cs.CV]. URL: https://arxiv.org/abs/2403.07153.

[3] DeepSeek-AI et al. *DeepSeek-V3 Technical Report*. 2025. arXiv: 2412.19437 [cs.CL]. URL: https://arxiv.org/abs/2412.19437.

[4] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV]. URL: https://arxiv.org/abs/2010.11929.

[5] Song Han, Huizi Mao, and William J. Dally. *Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding*. 2016. arXiv: 1510.00149 [cs.CV]. URL: https://arxiv.org/abs/1510.00149.

[6] Awni Hannun et al. *MLX: Efficient and flexible machine learning on Apple silicon*. Version 0.0. 2023. URL: https://github.com/ml-explore.

[7] Kaiming He et al. *Masked Autoencoders Are Scalable Vision Learners*. 2021. arXiv: 2111.06377 [cs.CV]. URL: https://arxiv.org/abs/2111.06377.

[8] Dan Hendrycks and Thomas Dietterich. *Benchmarking Neural Network Robustness to Common Corruptions and Perturbations*. 2019. arXiv: 1903.12261 [cs.LG]. URL: https://arxiv.org/abs/1903.12261.

[9] Andrew Howard et al. *Searching for MobileNetV3*. 2019. arXiv: 1905.02244 [cs.CV]. URL: https://arxiv.org/abs/1905.02244.

[10] Benoit Jacob et al. *Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference*. 2017. arXiv: 1712.05877 [cs.LG]. URL: https://arxiv.org/abs/1712.05877.

[11] Ji Lin et al. *MCUNet: Tiny Deep Learning on IoT Devices*. 2020. arXiv: 2007.10319 [cs.CV]. URL: https://arxiv.org/abs/2007.10319.

[12] Markus Nagel et al. *A White Paper on Neural Network Quantization*. 2021. arXiv: 2106.08295 [cs.LG]. URL: https://arxiv.org/abs/2106.08295.

[13] Junting Pan et al. *EdgeViTs: Competing Light-weight CNNs on Mobile Devices with Vision Transformers*. 2022. arXiv: 2205.03436 [cs.CV]. URL: https://arxiv.org/abs/2205.03436.

[14] Mark Sandler et al. *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. 2019. arXiv: 1801.04381 [cs.CV]. URL: https://arxiv.org/abs/1801.04381.

[15] Mingxing Tan and Quoc V. Le. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. 2020. arXiv: 1905.11946 [cs.LG]. URL: https://arxiv.org/abs/1905.11946.

[16] Hugo Touvron et al. *Training data-efficient image transformers distillation through attention*. 2021. arXiv: 2012.12877 [cs.CV]. URL: https://arxiv.org/abs/2012.12877.

[17] Qihang Yu et al. *An Image is Worth 32 Tokens for Reconstruction and Generation*. 2024. arXiv: 2406.07550 [cs.CV]. URL: https://arxiv.org/abs/2406.07550.