Image AI Projects

Project 1) Studying a new design primitive: how to represent images in main memory?

Image Calculator (IC) is an image storage format generator for image-AI applications, such as image recognition. Its goal is to generate the optimal storage format for image-AI applications, such that the inference time of the AI model and storage of the image dataset is dramatically reduced.

IC is based on a designed design space. The design space is composed of four design primitives, where each design primitive has its own domain of values. Every combination of the assigned design-primitive values corresponds to a specific storage format [1].

In this project, we will investigate a new design-primitive. The design primitive will decide how to represent the images in main memory. The current version of the IC uses a frequency-coefficient-based representation of images, which provides fast image-access and model-execution, but results in loss of accuracy. We will investigate whether we can represent images as regular red-green-blue (RGB) images reconstructed from the IC format. We will encapsulate this decision as a new primitive, where the primitive will define how to represent the image in main memory. It will include two values in its domain: frequency-coefficients (the current version) and RGB. We will examine whether using RGB instead of frequency-coefficients will allow us recover from the loss of accuracy, at the expense of slower image access and model-execution time.

Step 1: Understanding GPU-implementation of frequency transformation.
Step 2: Understanding GPU-implementation of inverse-frequency transformation.
Step 3: Making the GPU-implementation of inverse-frequency transformation efficient.

Step 4: Integrating the inverse-frequency transformation into Image Calculator's main training pipeline.
Step 5: Run experiments over a series of configurations and compare the results with the existing results.

[1] Utku Sirin, Stratos Idreos. The Image Calculator: 10x Faster Image-AI Inference by Replacing JPEG with Self-designing Storage Format. SIGMOD 2024.

Midway checking: Step 3 is completed.

Project 2) Studying an existing design primitive: achieving a better accuracy-inference-time and accuracy-storage tradeoff by using block size.

Image Calculator (IC) is an image-storage format generator for image-AI applications, such as image recognition. Its goal is to generate the optimal storage format for image-AI applications, such that the inference time of the AI model and storage of the image dataset is dramatically reduced.

IC is based on a designed design space. The design space is composed of four design primitives, where each design primitive has its own domain of values. Every combination of the assigned design-primitive values corresponds to a specific storage format. IC uses four design primitives: Subsampling Strategy, Block Size, DCT Coefficients, Quantization Factor (see Table 1 of [1]).

In this project, we will study Block Size primitive and its domain. The block-size primitive defines how we decompose the images into spatial blocks. For example, a block size of 8x8 decomposes a 256x256 image into 32x32 spatial (horizontal and vertical) blocks, where each block is of 8x8 size.

Currently, the domain of the block-size primitive contains six values: 8x8, 16x16, 32x32, 64x64, 128x128, 256x256. Different block sizes allow different tradeoffs in terms of how much inference-time/storage we can save versus how much accuracy we can sacrifice. In this project, we will investigate whether any block size offers a better tradeoff than others in terms of accuracy-inference-time and accuracy-storage tradeoffs.

Step 1: Understand the main training loop of the IC at the high-level.
Step 2: Understand how IC uses block size in its training pipeline.
Step 3: Run IC over a specific configuration and reproduce an existing result.
Step 4: Run IC for a set of configurations, where all the configurations use the same block size, such as 8x8. Examine how the accuracy-inference -time and accuracy-storage tradeoff change across different configurations with the same block size.
Step 5: Repeat step 4 for a different block size and compare it with the results of step 4.
Step 6: Repeat step 4 for all the six block sizes the IC uses and compare all the results.
Step 7: Repeat step 4-6 for two more datasets.

[1] Utku Sirin, Stratos Idreos. The Image Calculator: 10x Faster Image-AI Inference by Replacing JPEG with Self-designing Storage Format. SIGMOD 2024.

Midway checking: Step 4 is completed for a single block size.

Project 3) Improving the domain of an existing design primitive: achieving higher accuracy by compressing data less aggressively.

Image Calculator (IC) is an image-storage format generator for image-AI applications, such as image recognition. Its goal is to generate the optimal storage format for image-AI applications, such that the inference time of the AI model and storage of the image dataset is dramatically reduced.

IC is based on a designed design space. The design space is composed of four design primitives, where each design primitive has its own domain of values. Every combination of the assigned design-primitive values corresponds to a specific storage format. IC uses four design primitives: Subsampling Strategy, Block Size, DCT Coefficients, Quantization Factor (see Table 1 of [1]).

In this project, we will study the DCT Coefficients primitive. DCT stands for Discrete Cosine Transformation, which is a version of Fourier transformation that allows transforming images into the frequency domain rather than the usual, human-recognizable spatial domain in red-green-blue.

IC uses DCT to represent images in the frequency domain, so that it can compress the images as much as possible and save storage and inference time. DCT Coefficients primitive allows identifying the most important frequency coefficients and eliminating the rest of them from the data. This way, the amount of data to represent the images and the time to perform inference is dramatically reduced.

We will study the domain of the primitive DCT Coefficients. Currently, IC aggressively compresses the images by only keeping the most important frequency coefficients. While this allows large storage-savings, it results in accuracy drops. We will expand the domain, such that the data is less aggressively compressed and the accuracy is less aggressively sacrificed.

Step 1: Understand the main training loop of the IC at the high-level.
Step 2: Understand how IC uses DCT Coefficients in its training pipeline.
Step 3: Run IC over a specific configuration and reproduce an existing result.
Step 4: Implement the new domain values for the DCT coefficients.
Step 5: Repeat step 3 with one of the new domain values and compare the accuracy-inference-time and accuracy-storage tradeoffs.
Step 6: Repeat step 5 for a series of configurations using the old domain and new domain and compare the accuracy-inference-time and accuracy-storage tradeoffs.
Step 7: Repeat step 6 for two more datasets.

[1] Utku Sirin, Stratos Idreos. The Image Calculator: 10x Faster Image-AI Inference by Replacing JPEG with Self-designing Storage Format. SIGMOD 2024.

Midway checking: Step 4 is completed for a specific configuration.