USF CIS 4930/ CIS 6930 (Spring 2024)

Hardware Accelerators for Machine Learning

Final Report - 4/21/2024

Team A CPU

Herbert J Keels - Team Leader

Quang Huynh

Jasper Franken

Geoffroy Leyniers

Team B GPU

Matthew Gerken - Team Leader

Silvana Chain

Carlos Otero

Paula Briceño

Bibliographic Study

A survey on Image Data Augmentation for Deep Learning

Data augmentation is crucial for flag identification because flags are cloth, malleable, and rarely appear straight-on. Training data must include variations in lighting, angle, folds, and minor occlusions; otherwise, models overfit. Since collecting exhaustive datasets is impractical, data augmentation techniques are applied. These include:

- Geometric transformations (rotation, flipping, scaling)
- Color space augmentations
- Kernel filters
- Mixing images or random erasing
- Feature space augmentation
- Adversarial training
- Generative methods (GANs)
- Neural style transfer and meta-learning

Reference: Shorten, C. *A survey on Image Data Augmentation for Deep Learning*. Journal of Big Data, 2019. https://journalofbigdata.springeropen.com/counter/pdf/10.1186/s40537-019-0197-0.pdf

Flag Detection using a CNN

CNNs have been applied to flag detection under challenging conditions such as occlusion and varying scales. Techniques like 1x1 convolutional layers improve computational efficiency while maintaining accuracy. Multi-scale matching strategies help classify flags based on image size.

Reference: Gu, M., Hao, K., & Qu, Z. *Flag Detection with Convolutional Network*. 2018. https://dl.acm.org/doi/abs/10.1145/3297156.3297159

FlagDetSeg: Multi-Nation Flag Detection and Segmentation in the Wild

To handle random movements, occlusion, and lack of datasets, authors augmented existing flag images with transformations and synthetic environments. Models such as Mask-RCNN, YOLACT++, and PointRend were retrained. PointRend achieved 76.3% AP on synthetic test sets, while Mask-RCNN performed best (87.92% AP) on real+synthetic images.

Reference: Wu, S.-F., et al. *FlagDetSeg: Multi-Nation Flag Detection and Segmentation in the Wild*, 2021. https://www.albany.edu/faculty/mchang2/files/2021_11_AVSS_FlagDetSeg.pdf

Connecting national flags – a deep learning approach

This article presents a method for quantifying similarities between national flags using multi-scale optical features. Six deep learning models (YOLO V4/V5, SSD, RetinaNet, Fast R-CNN, FCOS, CornerNet) were evaluated. Anchor-free models, particularly CornerNet and RetinaNet, performed best. The study highlights unique challenges in flag identification and offers insights for future solutions.

Reference: Kalampokas, T., Mentizis, D., Vrochidou, E., et al. *Connecting national flags – a deep learning approach.* Multimed Tools Appl 82, 39435–39457 (2023). https://doi.org/10.1007/s11042-023-15056-y

Countries flags detection based on local context network and color features

This article presents a method for quantifying similarities between national flags using multi-scale optical features. Six deep learning models (YOLO V4/V5, SSD, RetinaNet, Fast R-CNN, FCOS, CornerNet) were evaluated. Anchor-free models, particularly CornerNet and RetinaNet, performed best. The study highlights unique challenges in flag identification and offers insights for future solutions.

Reference: Said, Y., Barr, M. *Countries flags detection based on local context network and color features.* Multimed Tools Appl 80, 14753–14765 (2021). https://doi.org/10.1007/s11042-021-10509-8

Deep Learning Techniques for Banner Image Classification.

This study addresses automatic recognition of banners for crowd monitoring. Pre-trained CNNs are trained on the DIAT Banner Dataset. Data augmentation via overlapping cropping increased dataset size and improved model accuracy. ResNet50 achieved the highest accuracy of 99.4%. Cropping techniques are applicable to flag datasets to enrich image variety and improve classification performance.

Reference: Pal, Chandrodoy, Sudhir Deshmukh, Sunita Dhavale, and Suresh Kumar. *Deep Learning Techniques for Banner Image Classification.* IETE Journal of Research, 2022, 15 pages. https://doi.org/10.1080/03772063.2022.2125450

Image processing and machine learning-based bone fracture detection and classification using X-ray images

Though focused on medical imaging, the study demonstrates preprocessing (edge detection, Hough lines, Harris corner detection) and machine learning classification, achieving 88.67% accuracy. Similar preprocessing could enhance flag detection by identifying edges and small details, aiding classification accuracy.

Reference:

Sahin, Muhammet Emin. *Image Processing and Machine Learning-based Bone Fracture Detection and Classification Using X-ray Images.* International Journal of Imaging Systems and Technology 33, no. 3 (2023): 853–865. https://doi.org/10.1002/ima.22849

Computer-aided identification of flags using color features

Avishikta L. and Parekh R. use color features and statistical feature vectors to classify flags with a K-NN classifier. Experiments with 4x3 partitioning and Euclidean distance achieved 99% accuracy on synthetic datasets, though real-world images showed reduced performance.

Reference: Lodh, Avishikta, and Ranjan Parekh. *Computer-aided identification of flags using color features.* International Journal of Computer Applications 975: 8887, 2016. https://www.ijcaonline.org/archives/volume149/number11/lodh-2016-ijca-911587.pdf

Detailed Description of Methods

Network Architecture

For this project, both CPU and GPU implementations were carried out in Google Colab. The goal was to implement a Convolutional Neural Network (CNN) for accurate classification of country flags.

The neural network architecture is as follows:

1. **Convolutional Layer 1:** Input 3 channels (RGB), output 32 feature maps.
2. **Pooling Layer 1:** Window size 2x2, stride 2 for dimensionality reduction.
3. **Convolutional Layer 2:** Input 32 channels, output 64 feature maps.
4. **Fully Connected Layer 1:** Input 64 channels × 56×56 dimensions, output 120 neurons.
5. **Fully Connected Layer 2:** Input 120 neurons, output 84 neurons.
6. **Output Layer:** Fully connected layer producing 24 features corresponding to the 24 possible classifications.

Convolutional layers extract spatial features from the flag images, pooling layers reduce dimensionality, and fully connected layers perform classification. Rectified Linear Unit (ReLU) activation is used in convolutional layers to enable the network to model complex patterns effectively.

Dataset

Source: Kaggle: Countries Flags Images

Description: The dataset contains 30+ images per country, providing sufficient variety for reliable classification. Its moderate size allows for efficient training and testing in Colab.

Training/Augmentation

To improve generalization, the following preprocessing and augmentation techniques were applied:

- **AutoAugment Policy:** IMAGENET policy for diverse transformations.
- **Resize:** All images resized to 224×224.
- **Normalization:** Standard mean and standard deviation for RGB channels.

The augmented dataset was split into **80% training and 20% testing**, ensuring the model is trained on varied examples while retaining sufficient test images for evaluation.

Testing

Performance metrics used for evaluation included:

- **Accuracy (%)** – Correct classification rate on test images.
- **Power (W)** – Measured on GPU using nvidia-smi.
- **Runtime (s)** – Time to run inference over multiple passes.
- **Efficiency (GFLOPs/W)** – GPU-specific metric combining computation and power usage.
- **Throughput (GFLOPs)** – Total number of operations per second.

A single test image was repeatedly passed through the network (up to 10,000 times) to benchmark CPU and GPU performance. GPU power and efficiency were measured, while CPU energy estimation was not feasible in Colab.

Experimental Results

The dataset was divided into an **80/20 split**: 80% of the images were used for training, and 20% for testing. The model was trained for 10 epochs, during which **accuracy and loss stabilized**, as shown in Figures 1 & 2. This occurred consistently on both CPU and GPU, confirming that the training process is independent of hardware during forward/backward passes.

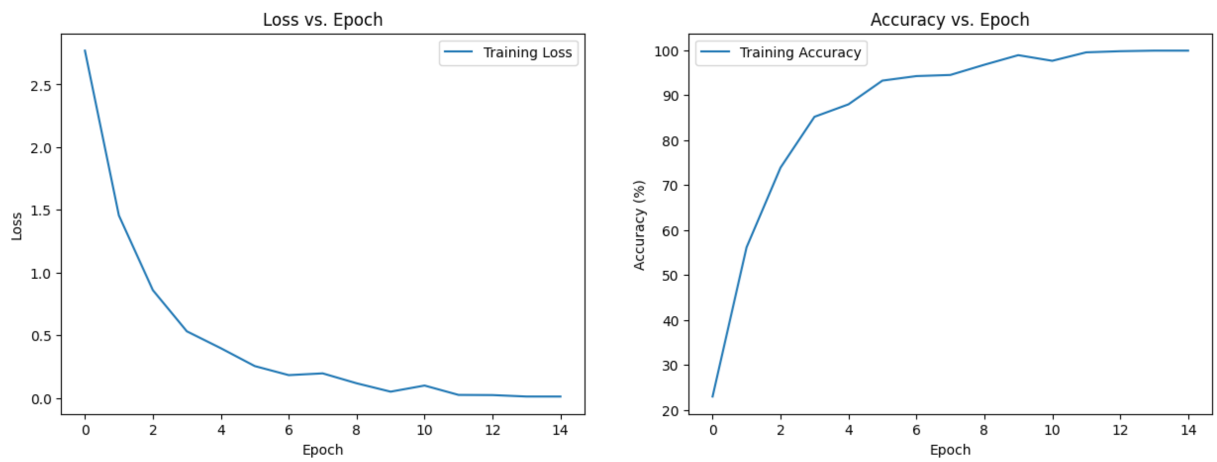

Figure 1: CPU Training Loss & Accuracy change through each epoch

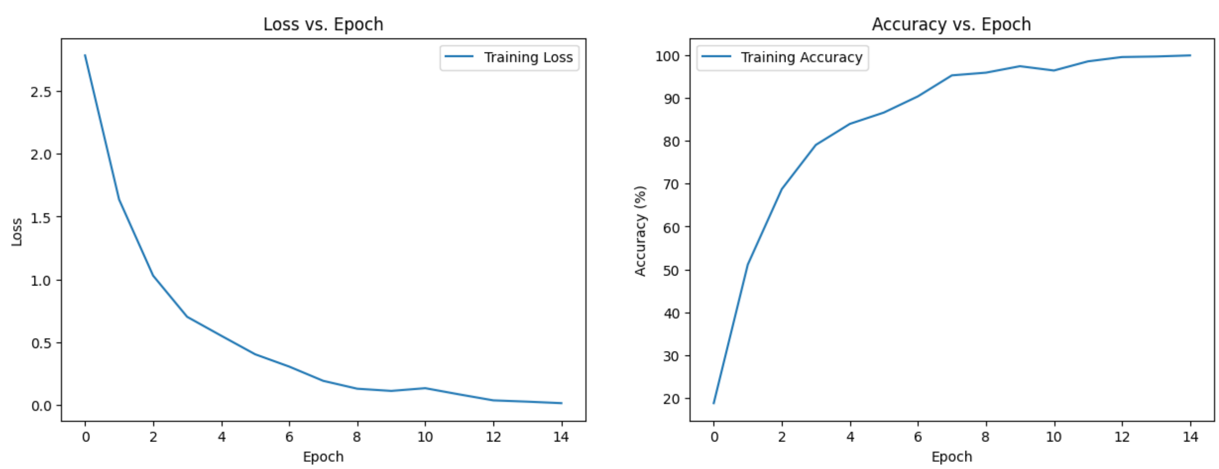

Figure 2: GPU Training Loss & Accuracy change through each epoch

CPU Results (Intel Xeon)

To measure the remaining (power, runtime, efficiency, and throughput) benchmarking metrics, we will get a single image and run it through the model 1,000 times and measure the time and throughput for both hardware platforms. Power and efficiency are only measured on the GPU because Colab does not have a reliable way to measure energy on their CPUs.

Test #	Power	Runtime (s)	Efficiency	Accuracy (%)	Throughput (GFLOPs)
Test 1	n/a	390.6	n/a	81.5	7.65
Test 2	n/a	391.1	n/a	83.5	7.64
Test 3	n/a	395.0	n/a	82	7.55
Test 4	n/a	419.5	n/a	78.5	7.12

Table 1: Table from running 10000 images through the model with the Intel Xeon CPU in colab.

Observations:

- CPU runtime is significantly longer than GPU runtime.
- Throughput is lower on CPU, but accuracy remains consistent with GPU results.
- Variability in accuracy is due to random sampling in the test set.

GPU Results (A100)

The same benchmark was run on the GPU to evaluate power efficiency and performance. Results are summarized in Table 2:

Test #	Power (W)	Runtime (s)	Efficiency (GFLOPs/W)	Accuracy (%)	Throughput (GFLOPs)
Test 1	122.73	4.10	5.96	82	731.5
Test 2	122.22	4.14	5.90	85.5	721.3
Test 3	118.45	4.35	5.79	77.5	686.7
Test 4	120.26	4.29	5.79	82	696.1

Table 2: Results from running 10000 images through the model with the A100 GPU in Colab.

Observations:

- GPU performs ~94× faster than CPU in runtime and throughput.
- Power efficiency is high due to parallelized computations on GPU cores.
- Accuracy across GPU tests is similar to CPU tests, confirming correct network behavior.
- Minor variability in accuracy reflects the random train/test splits and small sample sizes.

Final Results

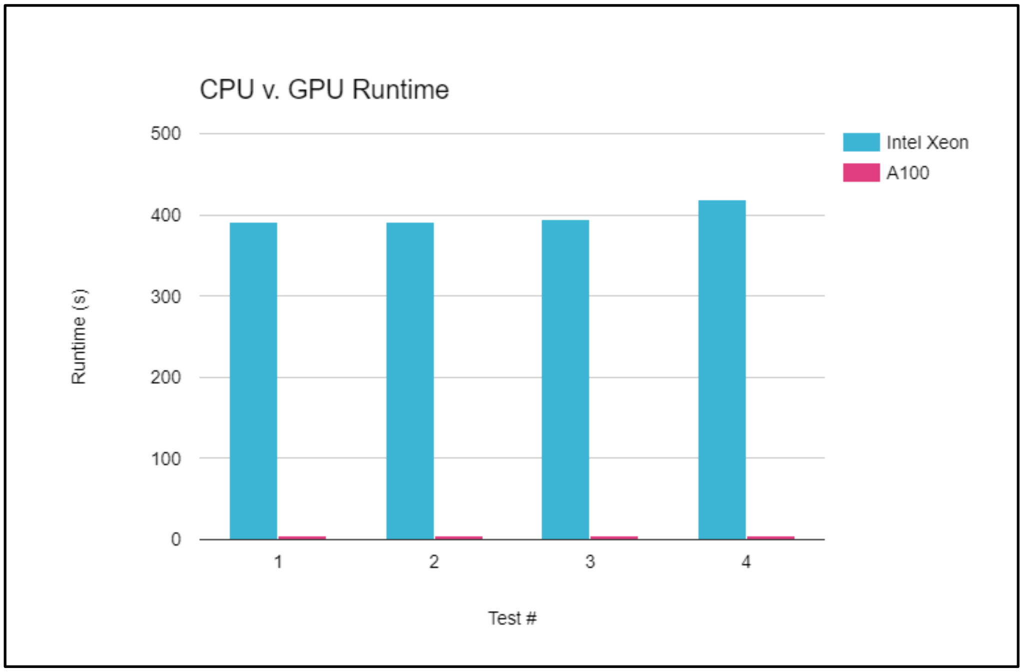

Figure 3: Results comparing GPU and CPU runtime. GPU is 94.6x faster than the CPU.

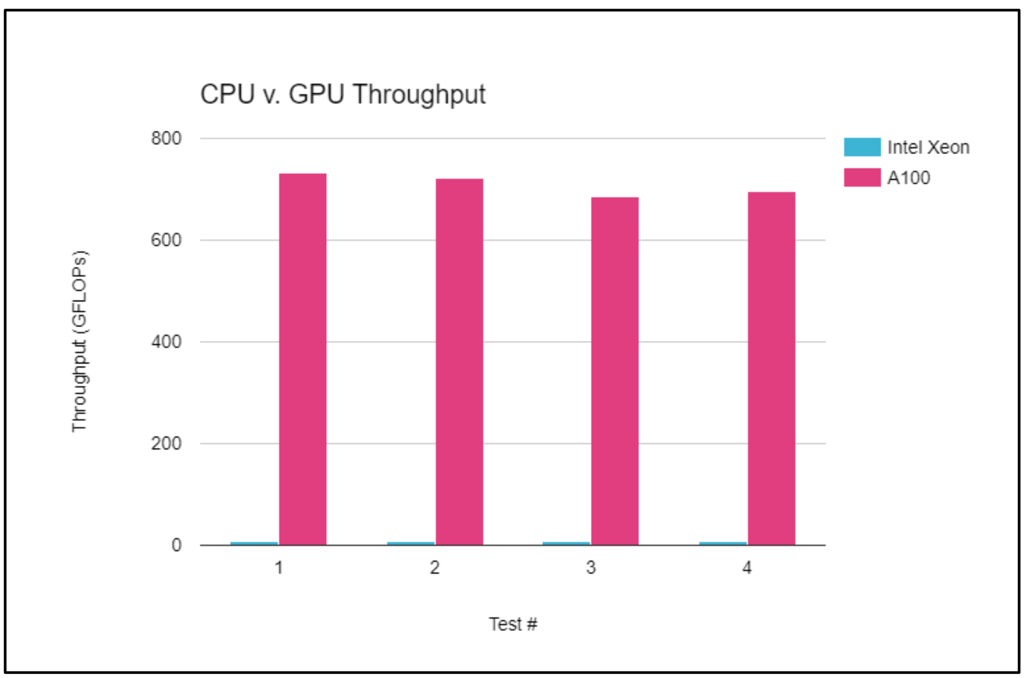

Figure 4: Results comparing GPU and CPU throughput. GPU has 94.7x higher throughput than the CPU.

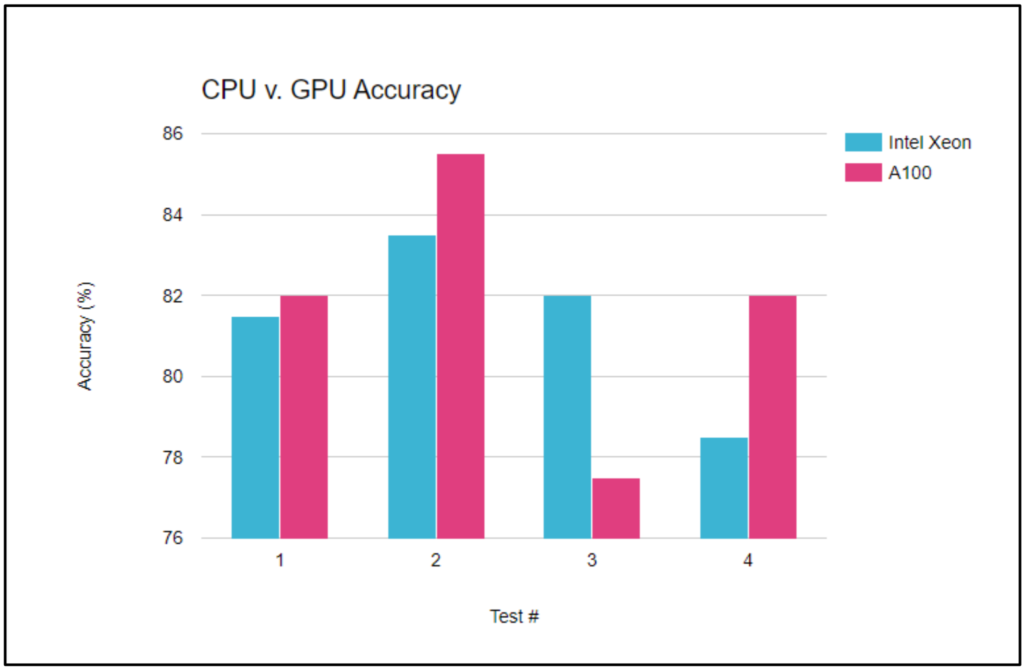

Figure 5: Results comparing GPU and CPU accuracy. The GPU had an average accuracy of 81.75 and the CPU was 81.375.

Summary

- The **GPU clearly outperforms the CPU** in runtime, throughput, and energy efficiency.
- Accuracy is largely unaffected by hardware choice, demonstrating that the network generalizes well.
- Variability in results highlights the importance of dataset size and split randomness.
- These findings validate the use of GPU acceleration for flag identification tasks in real-world applications.