Analyzing the compressibility of CNN kernels with Program Induction

Atharva Sehgal
University of Texas at Austin

Problem Definition

Motivating Questions:

1. Can we use program synthesis over a differentiable DSL to compress computer vision models?
2. Can we impose a syntactic prior on convolutional neural networks?

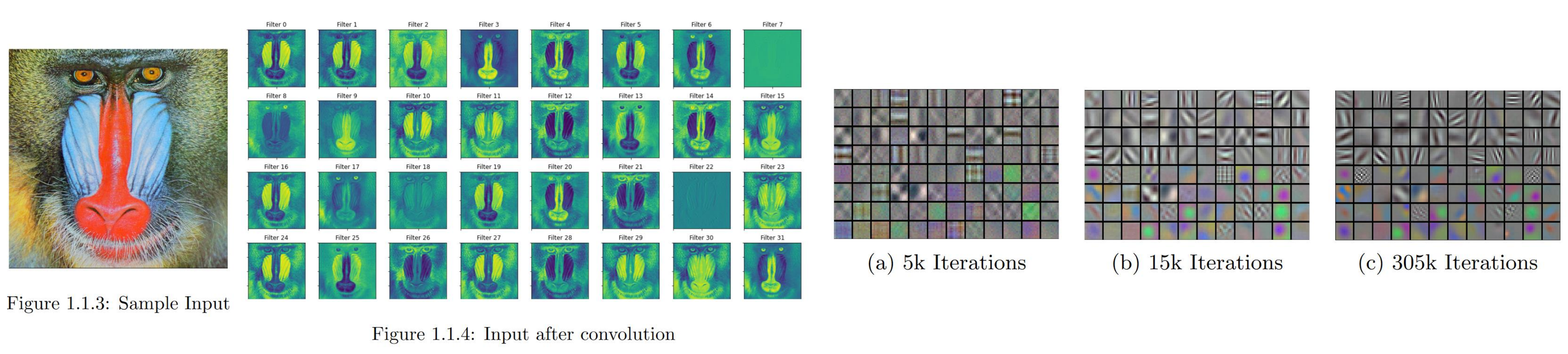


Figure 1: **Left**: Applying different learned kernels to a sample image. **Right:** Evolution of kernels with time.

Problem Formulation:

Given a neural architecture α, a DSL of convolutional kernels DSL, and a dataset of input-output examples $\mathbb{D}$, we are interested in learning an architecture $\hat{\alpha}$ such that $|\hat{\alpha}| < c|\alpha|$ for some constant $c \in (0,1)$ and:

$$E_{(x,y)\sim\mathbb{D}}[l(\alpha, \theta_1, x, y)] - E_{(x,y)\sim\mathbb{D}}[l(\hat{\alpha}, \theta_2, x, y)] < \epsilon$$

Approach

(α) ≔ VGG-16 Network:

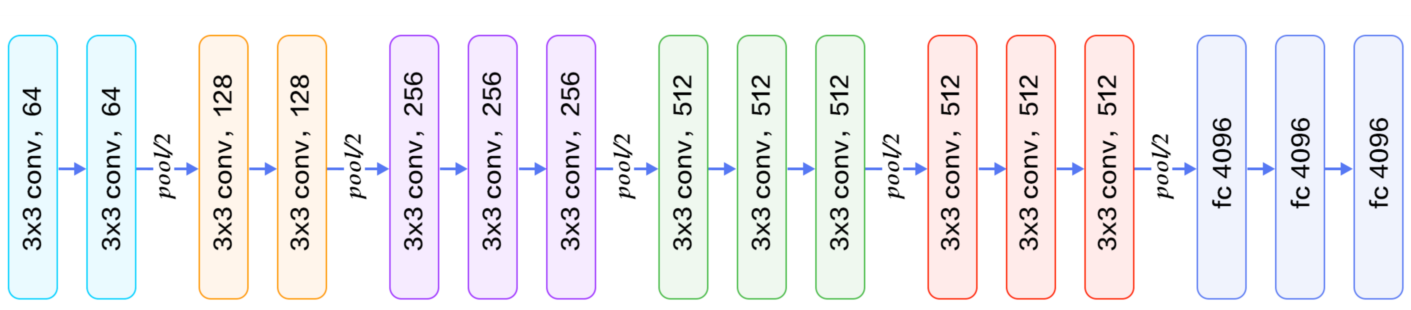


Figure 2: VGG network architecture

(DSL) ≔ Predefined/Clustered

- Predefined: Use formalisms of common CNN functions:

$$\texttt{edge-filter}(l,m,r) = \begin{bmatrix} l & m & r \\ l & m & r \\ l & m & r \end{bmatrix}$$

$$\begin{aligned} \texttt{original-kernel}() &= \ldots \\ \texttt{square-tetronimo}(x,y,\text{fill}) &= \ldots \\ \texttt{L-tetronimo}(x,y,\text{fill}) &= \ldots \\ \texttt{T-tetronimo}(x,y,\text{fill}) &= \ldots \end{aligned}$$

- Clustered: Agglomerative clustering on image similarity

$$dist(k_1, k_2) = \frac{1}{N}\sum_{i=1}^{N} SSIM(k_1 \otimes x_i, k_2 \otimes x_i)$$

$(\hat{\alpha})$ ≔ Discovered with iterative Gumbel-Softmax refining

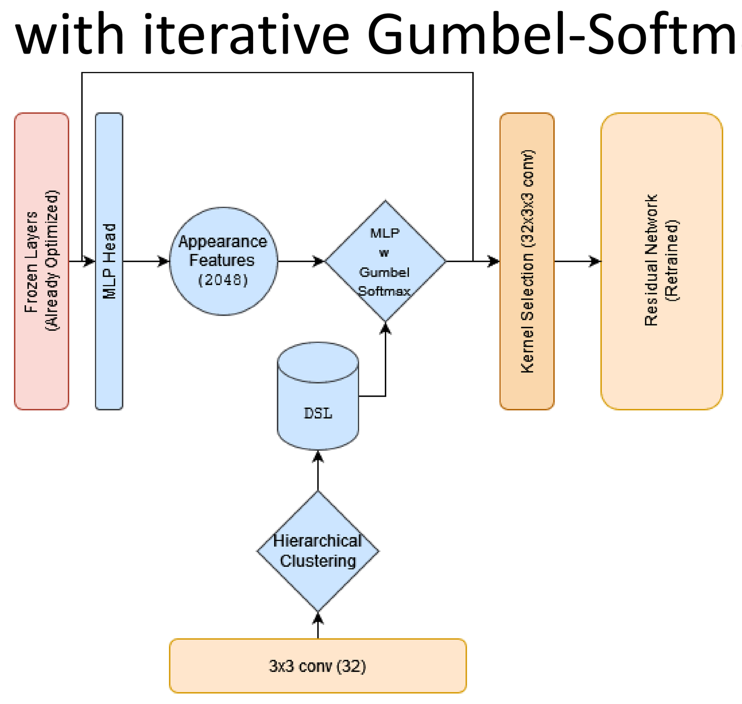
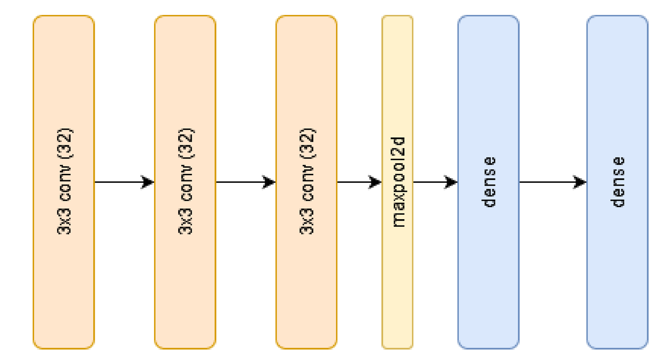


Figure 3: Full Algorithm training regime

Experiments

Constraints:

1. **"VGG-Tiny"**
   - 512K Parameters
   - 2.1 MB



1. **Fashion-MNIST**
   - Train: 48K
   - Valid: 12K
   - Test: 10K

Fixed DSL Experiments:

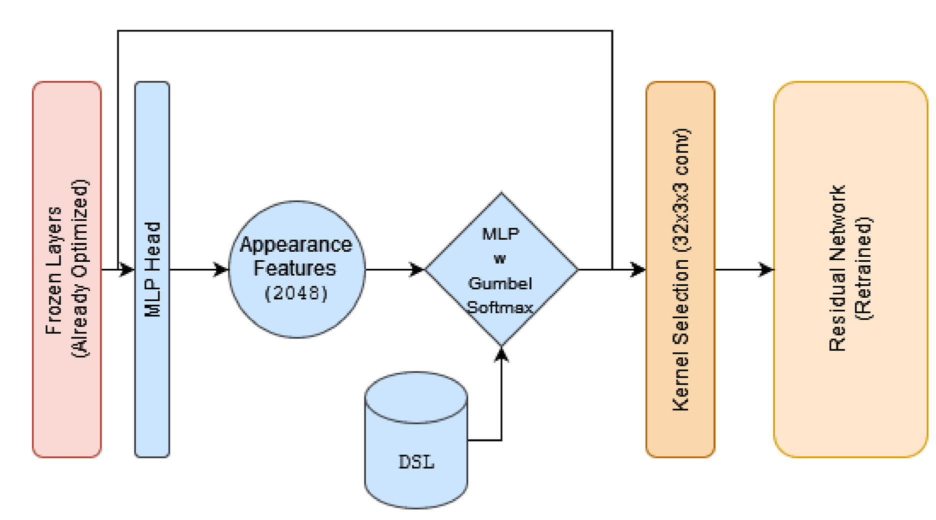


Figure 4:Training regime for fixed DSL

Learned DSL Experiments:

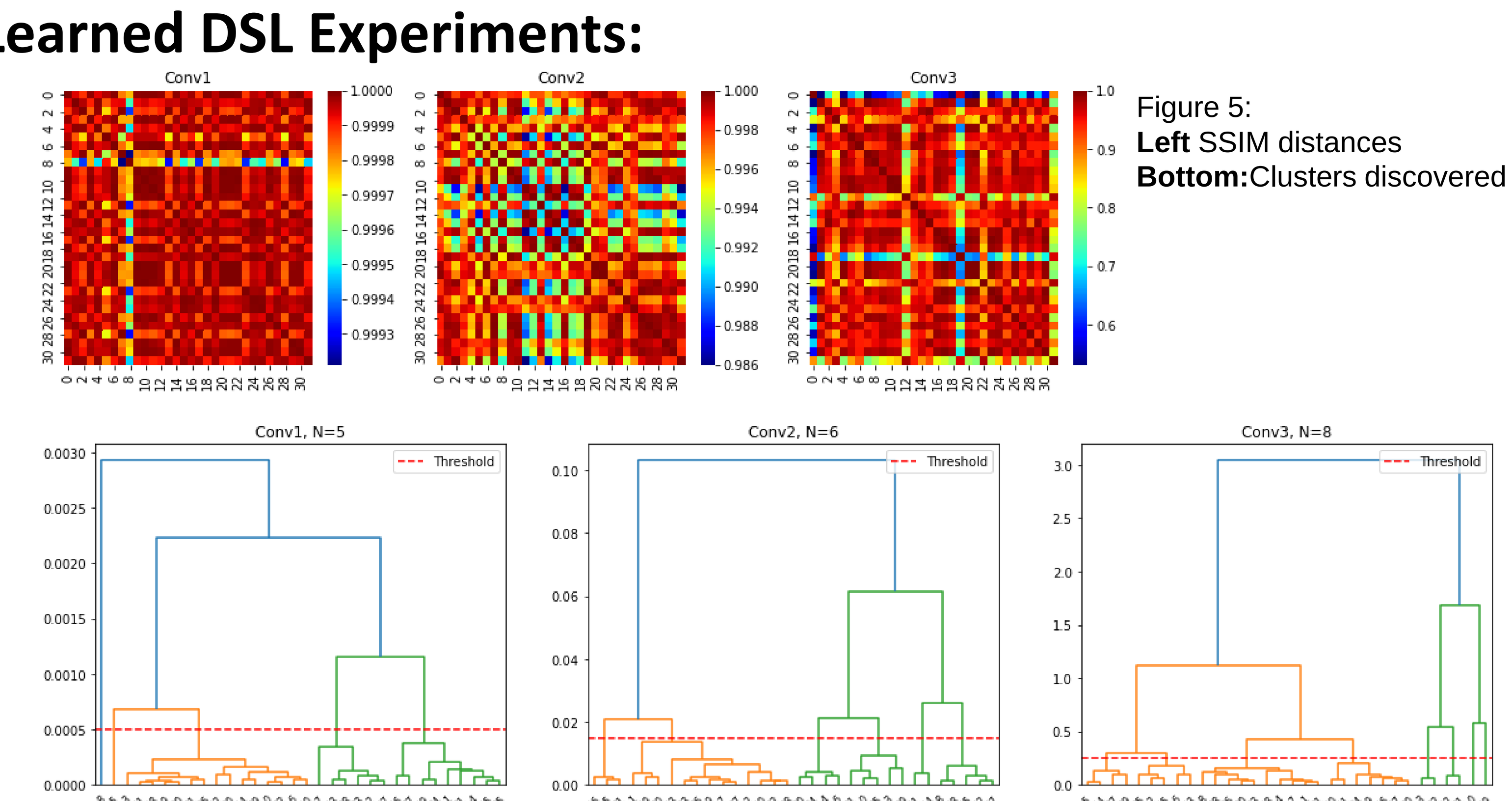


Figure 5:
Left SSIM distances
Bottom: Clusters discovered

FashionMNIST Experiments	Test Accuracy	# Parameters (CNN Layers Only)
VGG-Tiny	0.9167	18660
VGG-Tiny *Fixed DSL*	0.7219	18660
VGG-Tiny *Learned DSL*	0.9031	4077

What Next?

Drawbacks:

- Cannot compress Dense layers.
- Each layer needs to be trained iteratively. Time complexity is dependent on number of layers (regardless of layer size).

Opportunities:

- Using the same algorithm on larger datasets (Imagenet/CIFAR)
- Introducing a measure of compressibility to discover better DSLs
- Visualizing the learned programs

Contact

Atharva Sehgal
Email: atharvas@utexas.edu