

EFFICIENT CONVOLUTION ALGORITHMS

i. Fourier Transform

- The **Fourier Transform** is a tool that breaks a waveform (a function or signal) into an alternate representation characterized by **sine and cosine** functions.
 - It converts a signal from the **time domain** to the **frequency domain**.
-

ii. Separable Kernels

- Convolution is equivalent to converting both **input** and **kernel** into the **frequency domain** using a **Fourier Transform**, performing **point-wise multiplication**, and converting back to the time domain using an **inverse Fourier Transform**.

Mathematical Representation:

- $F\{f * g\} = F\{f\} \times F\{g\}$
 - $f * g = F^{-1}\{F\{f\} \times F\{g\}\}$
 - When a **d-dimensional kernel** can be expressed as the **outer product of d vectors** (one vector per dimension), the kernel is called a **separable kernel**.
 - Separable kernels require **fewer parameters** and reduce computational cost.
-

Runtime Complexity Comparison

Kernel Type	Runtime Complexity for d-dimensional kernel with w elements wide
Traditional Kernel	$O(w^d)$
Separable Kernel	$O(w \times d)$

Random and Unsupervised Features

- To reduce the **cost of convolutional network training**, features can be used that are **not trained in a supervised way**.

iii. Random Initialization:

- Layers consisting of **convolution followed by pooling** naturally become **frequency selective** and **translation invariant** when assigned random weights.
- Randomly initialize several CNN architectures and just **train the last classification layer**.

- Once a winner is determined, train that model using a **more expensive approach** (supervised approach).
 - **Hand Designed Kernels:** Used to detect edges at a certain orientation or scale.
 - **Unsupervised Training:**
 - Unsupervised pre-training may offer a **regularization effect**.
 - It may also allow for **training of larger CNNs** because of reduced **computation cost**.
-

iv. Greedy Layer-wise Pre-training

Instead of training an entire convolutional layer at a time, we can **train a model of a small patch**:

- Train the **first layer in isolation**.
- Extract all features from the first layer **only once**.
- Once the first layer is trained, its **output is stored** and used as **input for training the next layer**.
- We can train **very large models** and incur a **high computational cost only at inference time**.