Summary of Computer Vision Laboratory Project

Chair of Digital Signal Processing and Circuit Technology

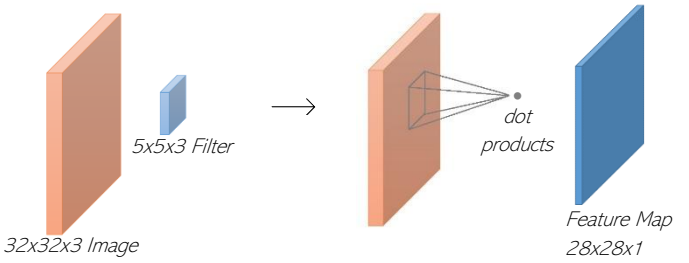
Mehmet Can Odabas
667030

TECHNISCHE UNIVERSITÄT CHEMNITZ

Convolutional Neural Networks for Visual Recognition

The overall goal of this system is to achieve accurate visual recognition through the construction of a robust Convolutional Neural Network (CNN), utilizing the **CIFAR-10** dataset as a benchmark for evaluation and optimization. The core of the system is convolutional layers which can be configure various ways and responsible for **automatically** learning - extracting features from the input images. These convolutional layers are often followed by activation functions, pooling layers, regularization and optimization steps. Implementation was done with **Python3**, **PyTorch** and **NumPy**, **Matplotlib** etc. libraries were used.

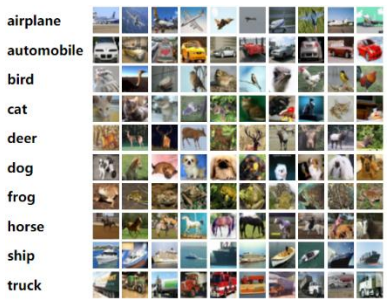
The convolution operation is central to the success of CNNs in visual recognition tasks. Its ability to capture spatial hierarchies, reduce parameters, detect translation-invariant features, and enable efficient computation makes it superior to traditional fully connected networks for handling image data and other spatially structured inputs.



The convolution operation slides the filter over the input image, computing the dot product at each position. This results in a 2D matrix (feature map) representing the filter's response to the input data, presence of features detected by the filters.

Data Preparation

CIFAR-10 dataset is used
It consists of 60.000 images in 10 classes
50,000 training images and 10,000 test images



Model Architecture

[conv-relu-pool] -> [conv-relu-pool] -> [affine] -> [softmax]

[conv-relu-pool]: Convolutional layer followed by ReLU activation and max pooling

[affine]: Fully connected layer

[softmax]: Softmax layer applied to the output of the fully connected layer for classification

Different configurations of layers can be used for more efficient accuracy.

Model Training

Loss Computation
Calculate the difference between predicted outputs and actual targets using a loss function

Regularization
L2 Weight regularization or Dropout

Backpropagation
Compute gradients which are calculated using the chain rule to propagate the error backwards through the network layers

Optimization
Adam, Adagrad…

Evaluation of model

Assess model performance on validation (or test) set and make adjustments if necessary