Object Recognition using Convolutional Neural Networks

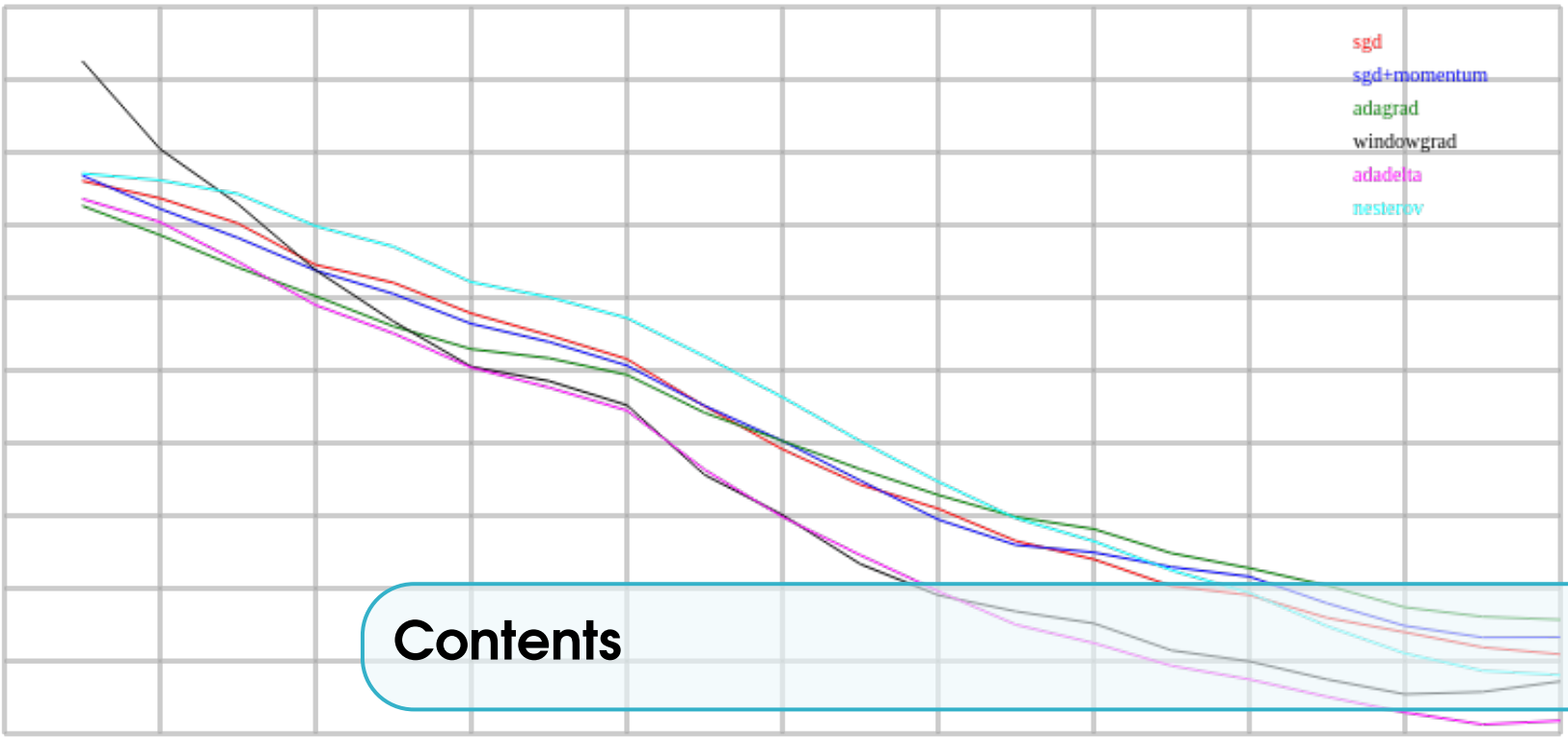
MEET PRAGNESH SHAH | YASH BHALGAT | SHARAD MIRANI

CS663 DIGITAL IMAGE PROCESSING

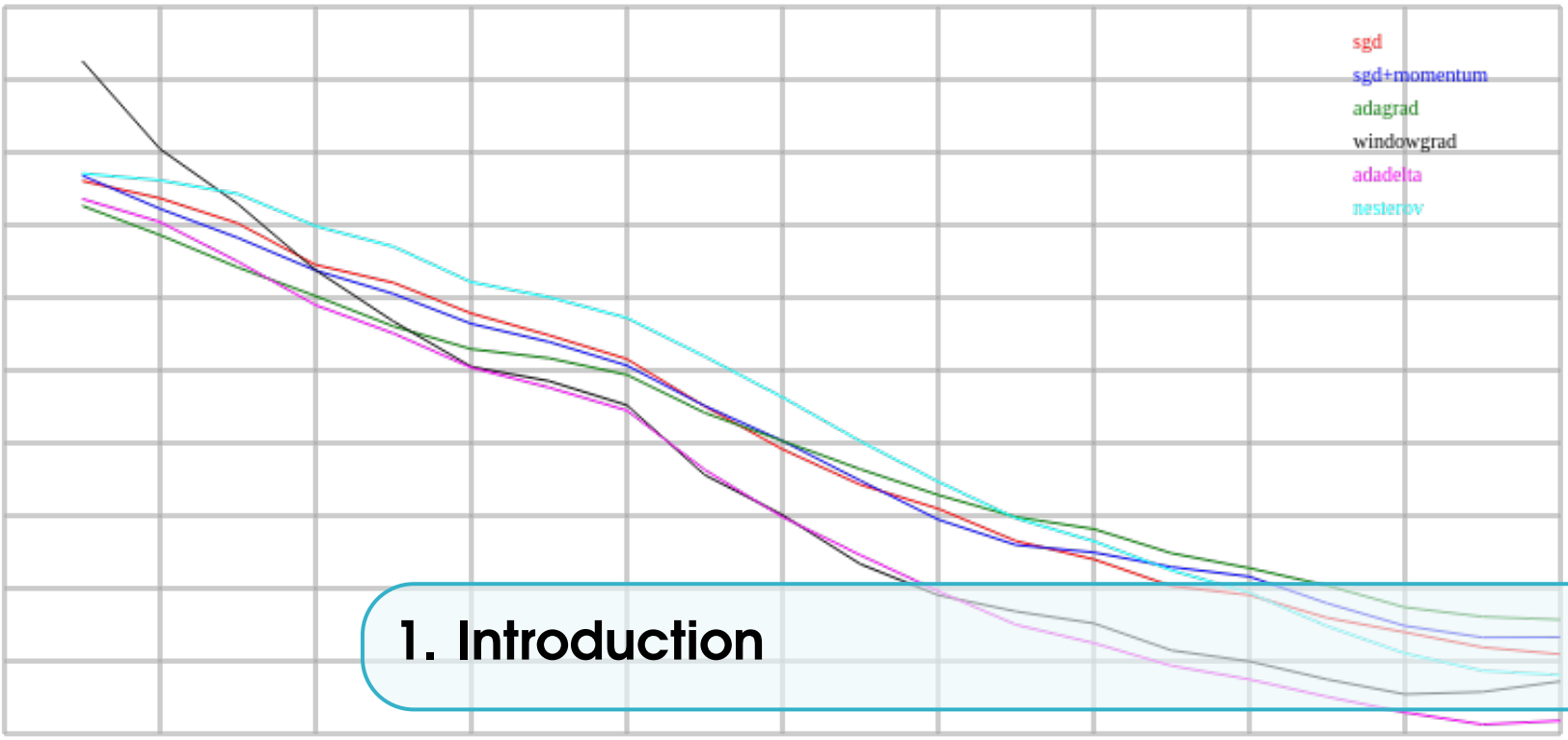
INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY

https://github.com/meetshah1995/ConvolutionalNeuralNetwork

This project was done under the supervision of Prof. Suyash Awate, and Prof. Ajit Rajwade, instructors for the course along with help of Standford OpenWare Course Neural Networks for Visual Recognition



Contents



1. Introduction

1.1 Prologue

In machine learning, a convolutional neural network (CNN, or ConvNet) is a type of feed-forward artificial neural network where the individual neurons are tiled in such a way that they respond to overlapping regions in the visual field. Convolutional networks were inspired by biological processes and are variations of multilayer perceptrons which are designed to use minimal amounts of preprocessing.They are widely used models for image and video recognition.

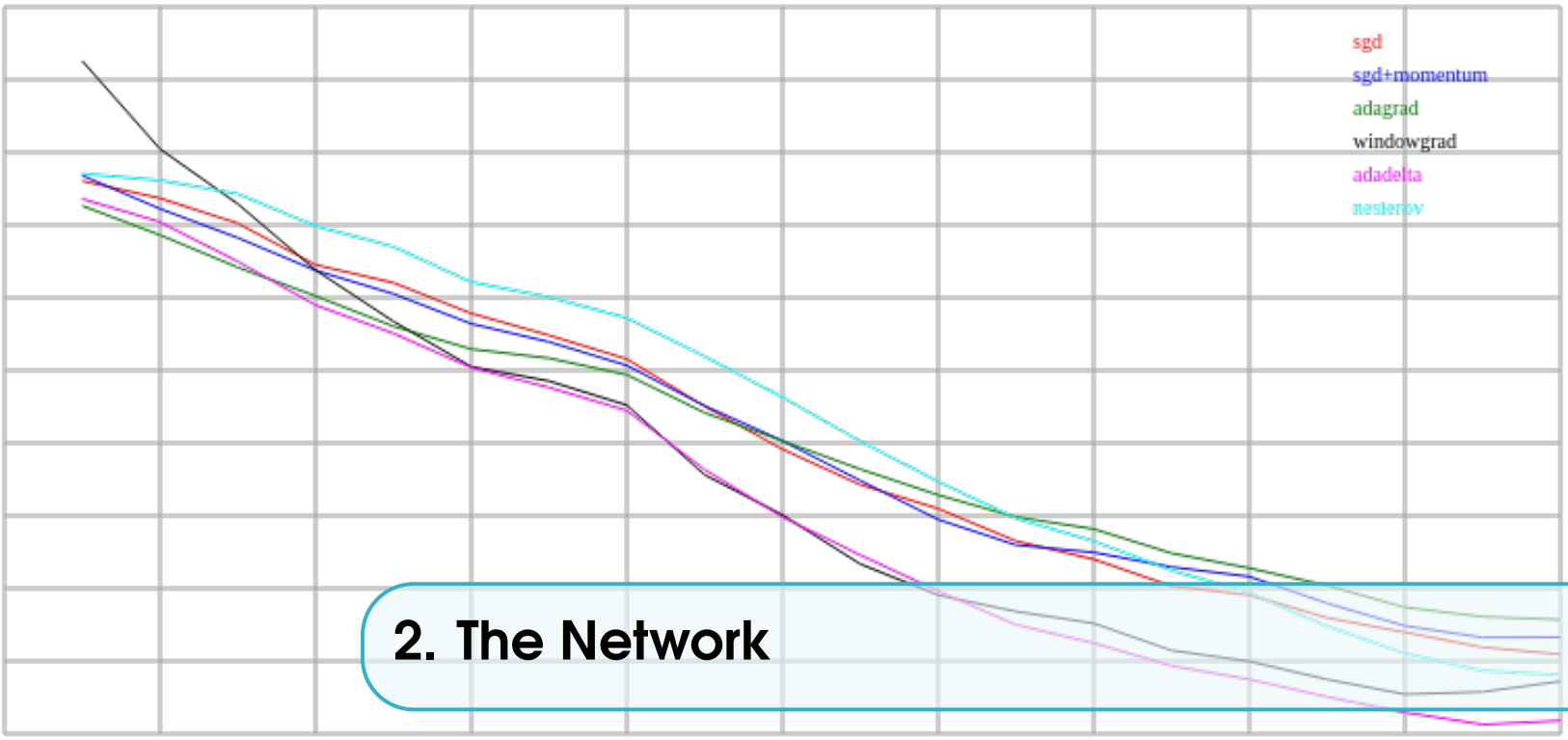
1.2 Abstract

The Convolutional Neural Network mainly consists of a layered network of neurons and uses a cost function which is minimized using back propagation algorithm.

The Convolutional Neural Network that we have implemented consists of the following components :

1.3 References

- Rich feature hierarchies for accurate object detection and semantic segmentation, *R. Girshick et al*.
- Very Deep Convolutional Networks For Large-Scale Image Recognition ,K. *Simonyan, A. Zisserman*



2. The Network

2.1 Data PreProcessing

To reduce the dimensionality of the features before passing the images to the network, the dataset is processed using ZCA-whitening. For image data, high frequency data will typically reside in the space spanned by the lower Eigenvalues. Hence ZCA is a way to strengthen these, leading to more visible and enhanced edges.

Given an Eigendecomposition of a covariance matrix

$$\bar{X}\bar{X}^T = LDL^T$$

where $D = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ is the diagonal matrix of Eigenvalues, ordinary whitening resorts to transforming the data into a space where the covariance matrix is diagonal:

$$\sqrt{D^{-1}}L^{-1}\bar{X}\bar{X}^TL^{-T}\sqrt{D^{-1}} = \sqrt{D^{-1}}L^{-1}LDL^TL^{-T}\sqrt{D^{-1}} \qquad = \mathbf{I}$$

That means we can diagonalize the covariance by transforming the data according to the below equation. ZCA does something different than PCA –it adds a small epsilon to the Eigenvalues and transforms the data back.

$$\tilde{X} = L\sqrt{(D+\varepsilon)^{-1}}L^{-1}X.$$

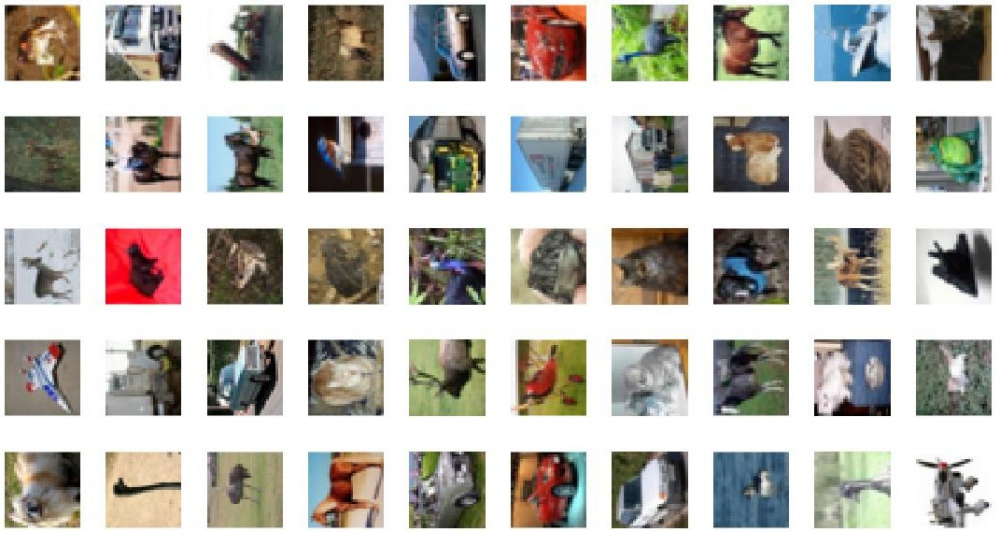


Figure 2.1: 50 randomly unprocessed images from the CIFAR Database

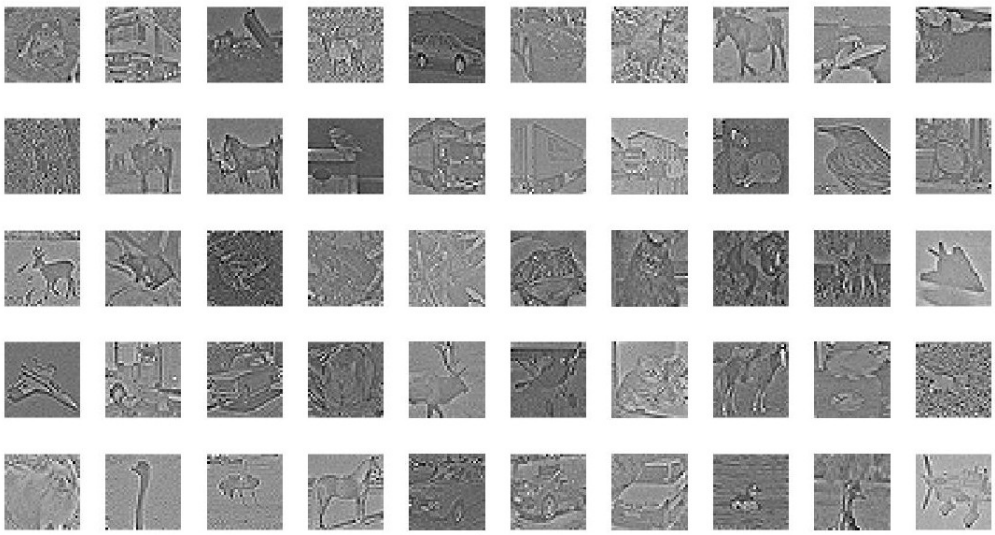


Figure 2.2: 50 ZCA whitened images from the CIFAR Database

2.2 Layers

The Convolutional Neural Network consists of the described units:

2.2.1 Convolution Layer

Convolution of every image with 20 (numFilters) weight matrices, extracting 20 features per image. During training, the features are updated as the weight matrices are updated using back-propogation.

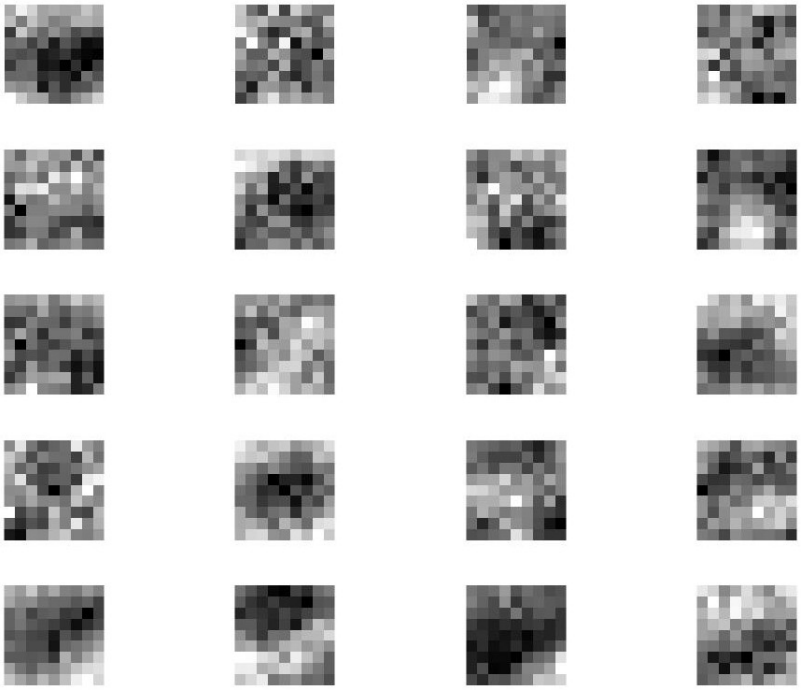


Figure 2.3: Weights of the CNN for convolution layer after learning from MNIST train images

2.2.2 Sub-Sampling Layer

Convolved images obtained in convolutional layer are subsampled by a factor of 2 to reduce the computation time.

2.2.3 Pooling Layer

This is a form of non-linear down-sampling and is done over 2x2 patched to reduce the features dimensionality.

Figure 2.4: Original Image

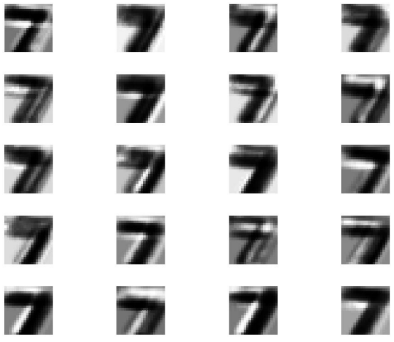
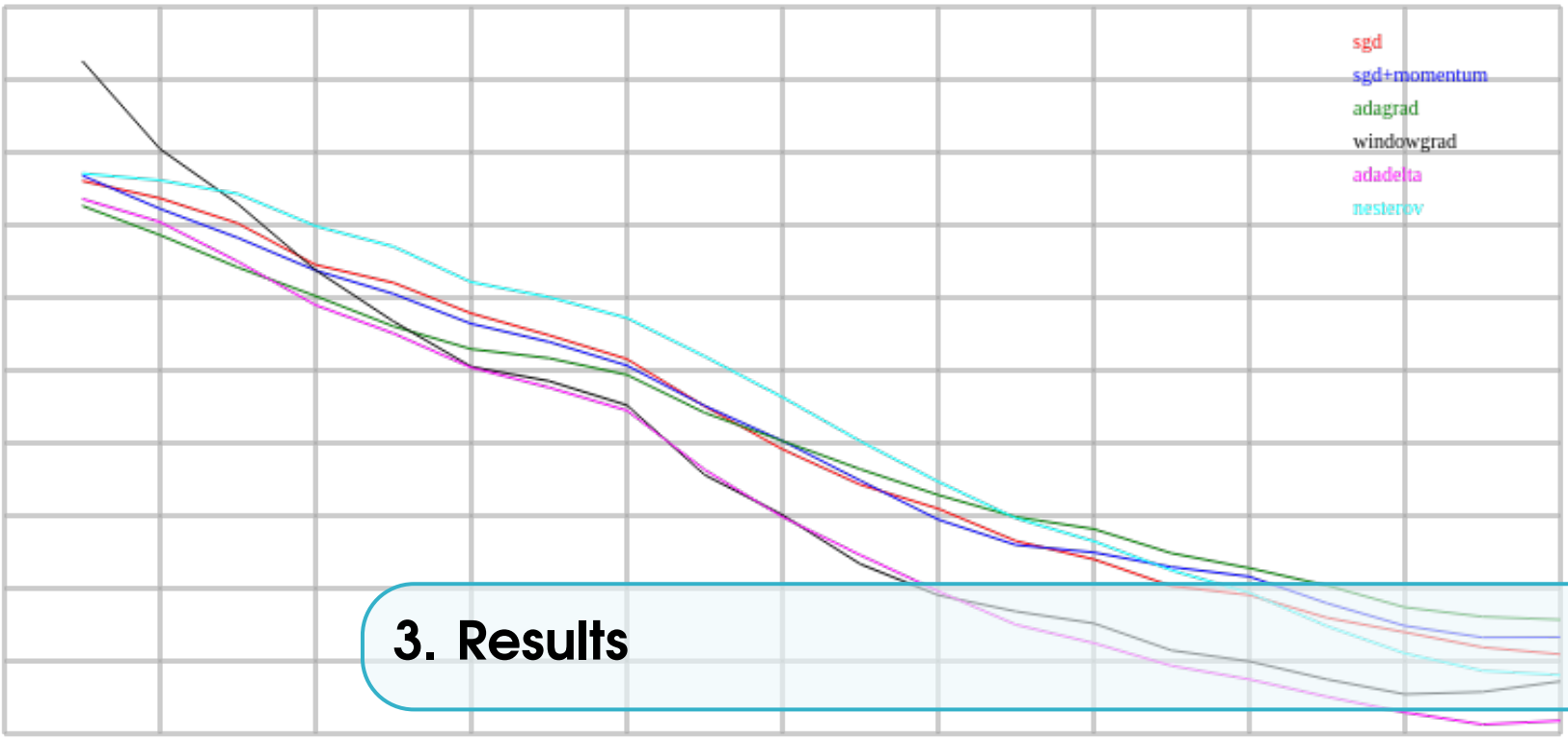


Figure 2.5: Activation of Image across 20 feature extraction filters

2.2.4 Loss Layer

1: **procedure** STOCHASTIC GRADIENT DESCENT
2: Choose an initial vector of parameters w and learning rate η.
3: Repeat until an approximate minimum is obtained using random shuffles:
4: **for** i = 1, 2, ... n **do**
5: $w := w - \eta \nabla Q_i(w)$.
6: **end for**
7: **end procedure**



3. Results

3.1 MNIST Data

The MNIST database of the 10 handwritten digits, available from this page, has a training set of 60,000 examples, and a test set of 10,000 examples. We tried to train the network for different number of trainImages everytime - 60000, 20000, 10000, 5000.

3.1.1 Results

Accuracy v/s Training Data	
No of Training images	Accuracy
60000	97.04%
20000	91.73%
10000	88.64%
5000	76.55%

3.2 CIFAR-10 Dataset

The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

For this dataset, we tried converting the rgb-image to single channels - Hue of HSV space, Y of YCbCr space and simple grayscale image. Then we applied ZCA-whitening on the images which enhanced the features significantly, also helping in dimensionality reduction. The zca step gave us the best acuracy for Y channel of 58%.

3.2.1 Classes

1. airplane
2. automobile
3. bird
4. cat
5. deer
6. dog
7. frog
8. horse
9. ship
10. truck

3.2.2 Results

Results after ZCA-whitening	
Channel used	Accuracy
Y of YCbCr	58.0%
Hue of HSV	45.37%
grayscale	38.31%

3.3 STL-10 Data

The STL-10 dataset, inspired by the CIFAR-10 dataset, is an image recognition dataset for developing unsupervised feature learning, deep learning, self-taught learning algorithms. The higher resolution of this dataset (96x96) made it a challenging benchmark for developing the network We used very less number of images in this testing (2000 train and 3200 test). The testing was done in a similar way to the CIFAR dataset.

3.3.1 Classes

1. airplane
2. car
3. bird
4. cat
5. deer
6. dog
7. monkey
8. horse
9. ship
10. truck

3.3.2 Results

Results after ZCA-whitening for stl10	
Channel used	Accuracy
Y of YCbCr	41.44%
Hue of HSV	33.98%
grayscale	30.24%