ARISTOTLE UNIVERSITY OF THESSALONIKI
FACULTY OF SCIENCES
SCHOOL OF INFORMATICS

POSTGRADUATE STUDIES PROGRAM ON INFORMATICS AND COMMUNICATIONS
SPECIALIZATION ON DIGITAL MEDIA AND COMPUTATIONAL INTELLIGENCE

Content-based Image Retrieval

Presenter: Efstathios Chatzikyriakidis

M.Sc. in Informatics and Communications
Specialization in Computational Intelligence and Digital Media
School of Informatics, Faculty of Sciences
Aristotle University of Thessaloniki, Hellas

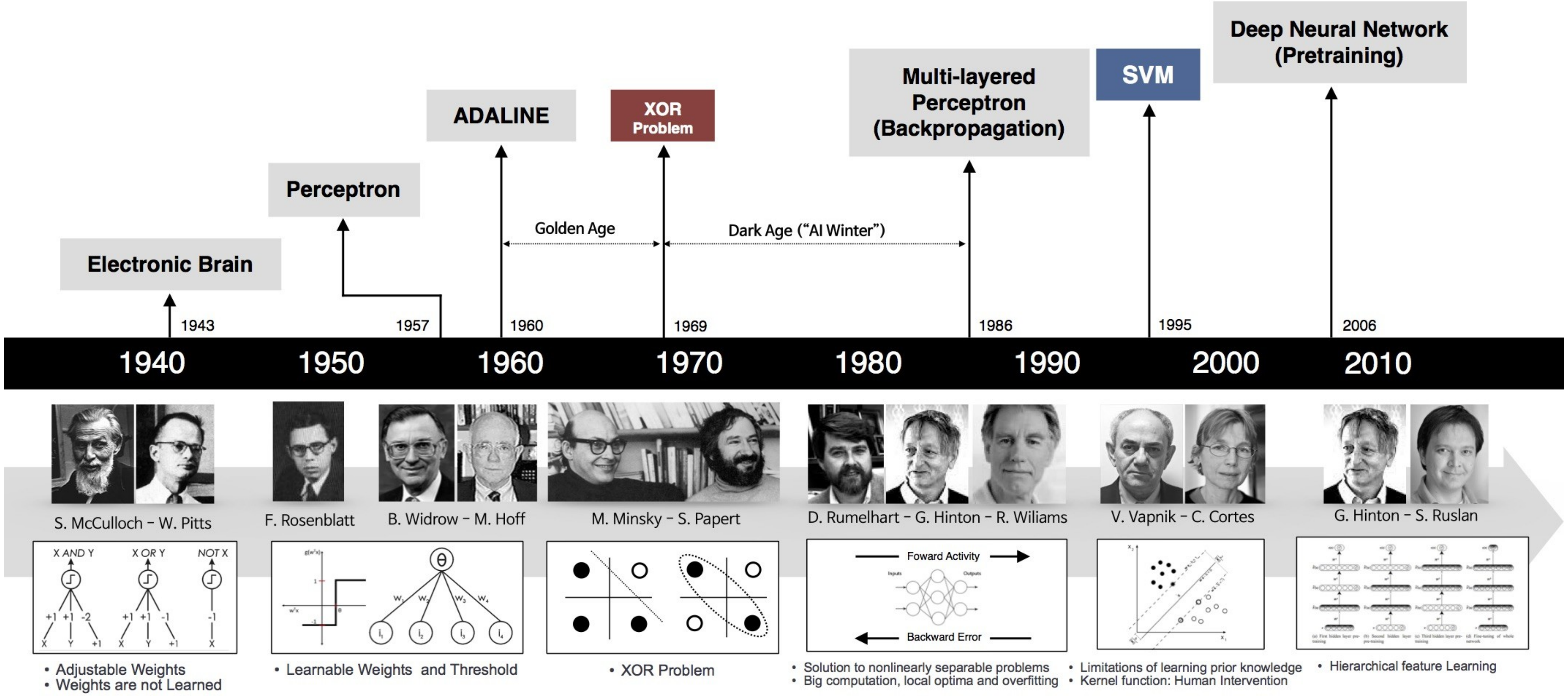
B.Sc. in Informatics and Communications
Specialization in Software Engineering
Department of Informatics and Communications, Faculty of Applied Technology
Technological Educational Institute of Central Macedonia, Hellas

Email : stathis.chatzikyriakidis@gmail.com **Website :** http://www.efxa.org/

Introduction

Important history of “Artificial Intelligence”

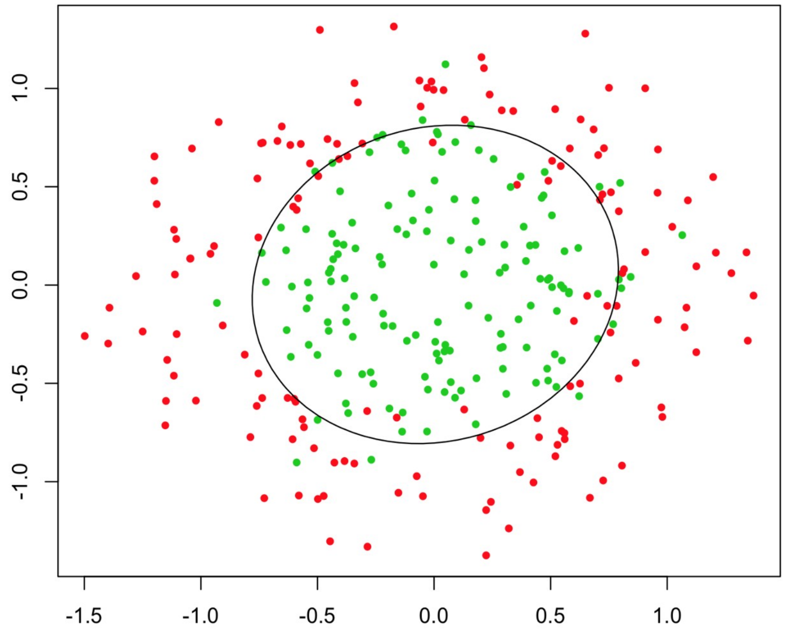
We had a long journey… and we are still at the birth of it...

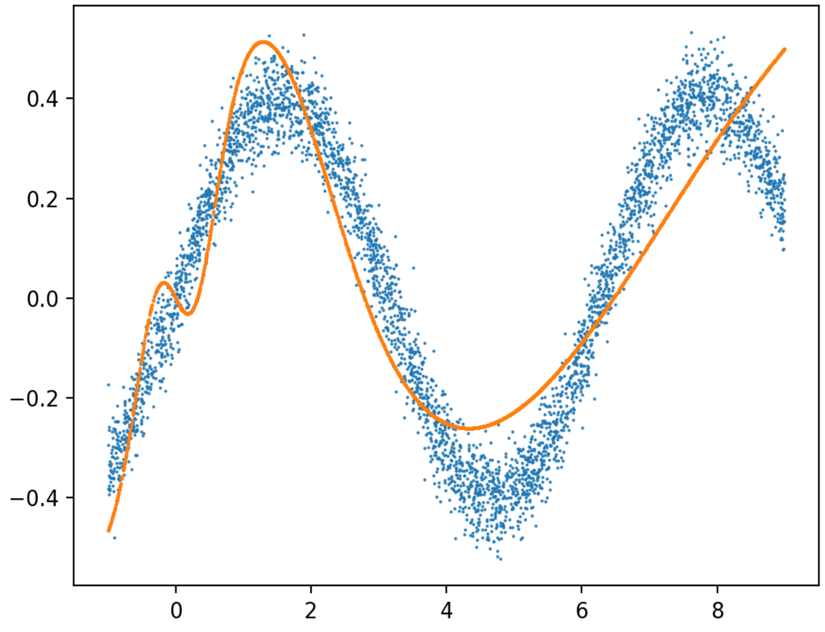


Introduction

Advantages of Artificial Neural Networks

- Satisfactory separation of non-linear separable input data
- Satisfactory function approximation using only input data

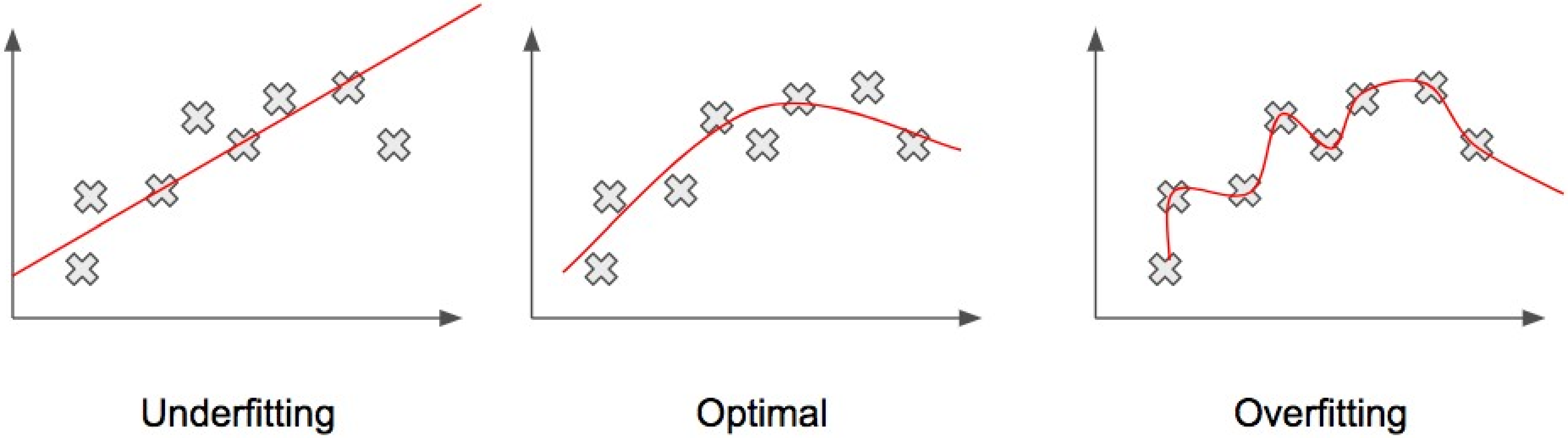




Introduction

Advantages of Artificial Neural Networks

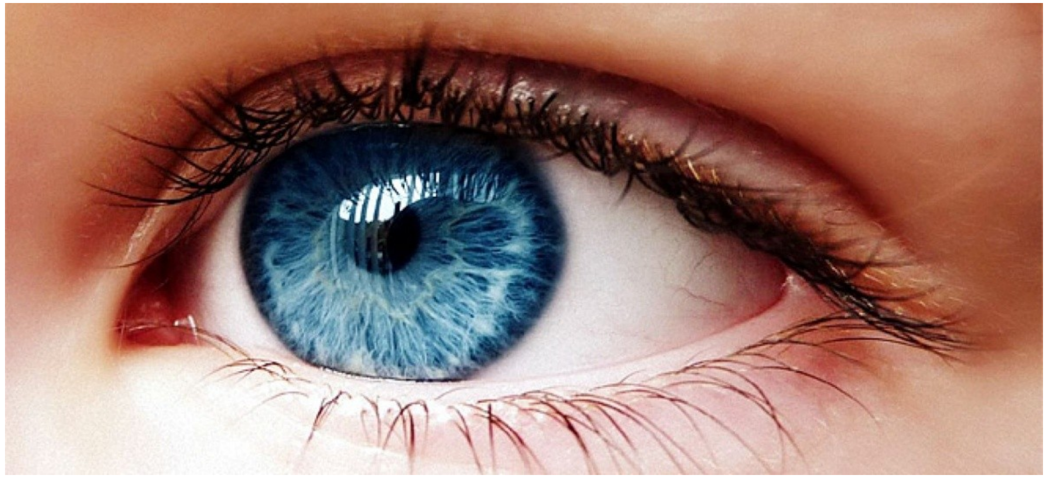
- Good generalization that captures the general manifold of data



Introduction

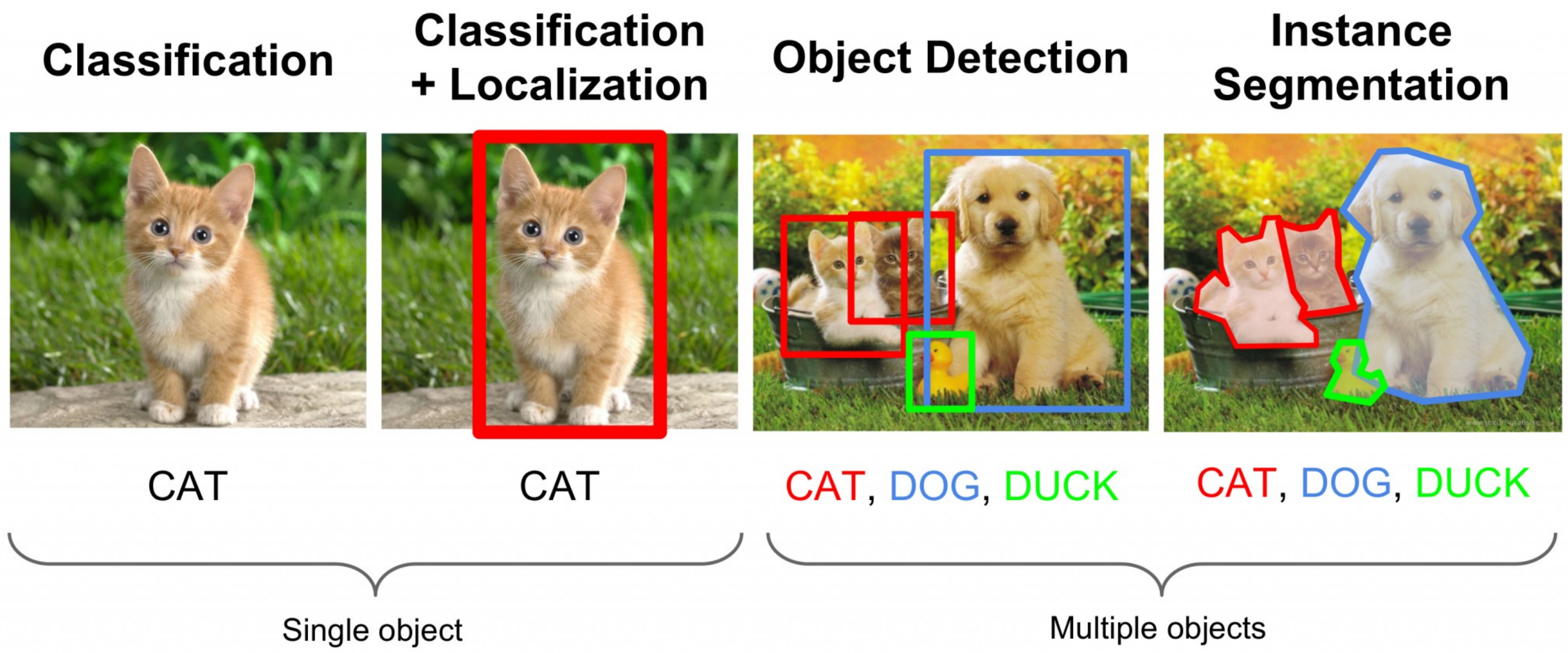
Computer Vision and its future goal

- The strongest sense of most animal species
- Simulate how brains see and understand the world through vision sense



Computer Vision

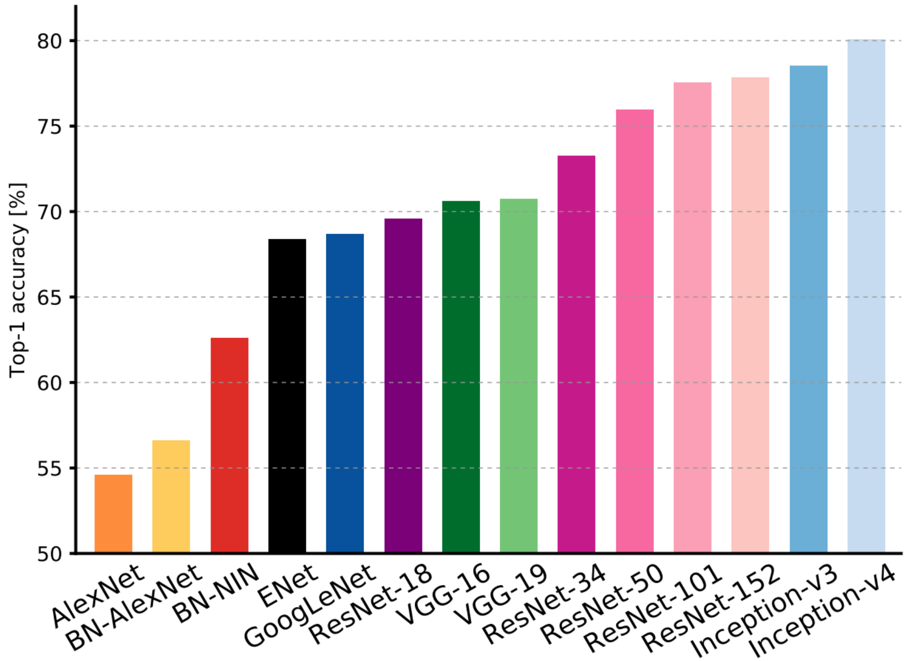
Well-known tasks of high-level image understanding

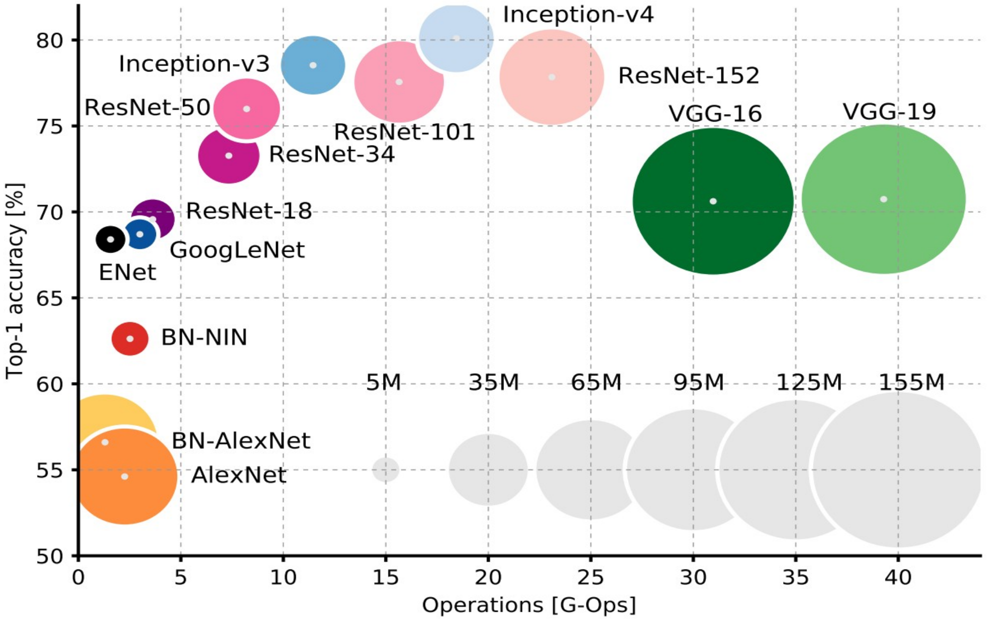


Computer Vision

State-of-the-art Image Classification

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)





Fashion MNIST

Dataset

Fashion MNIST Dataset

Characteristics

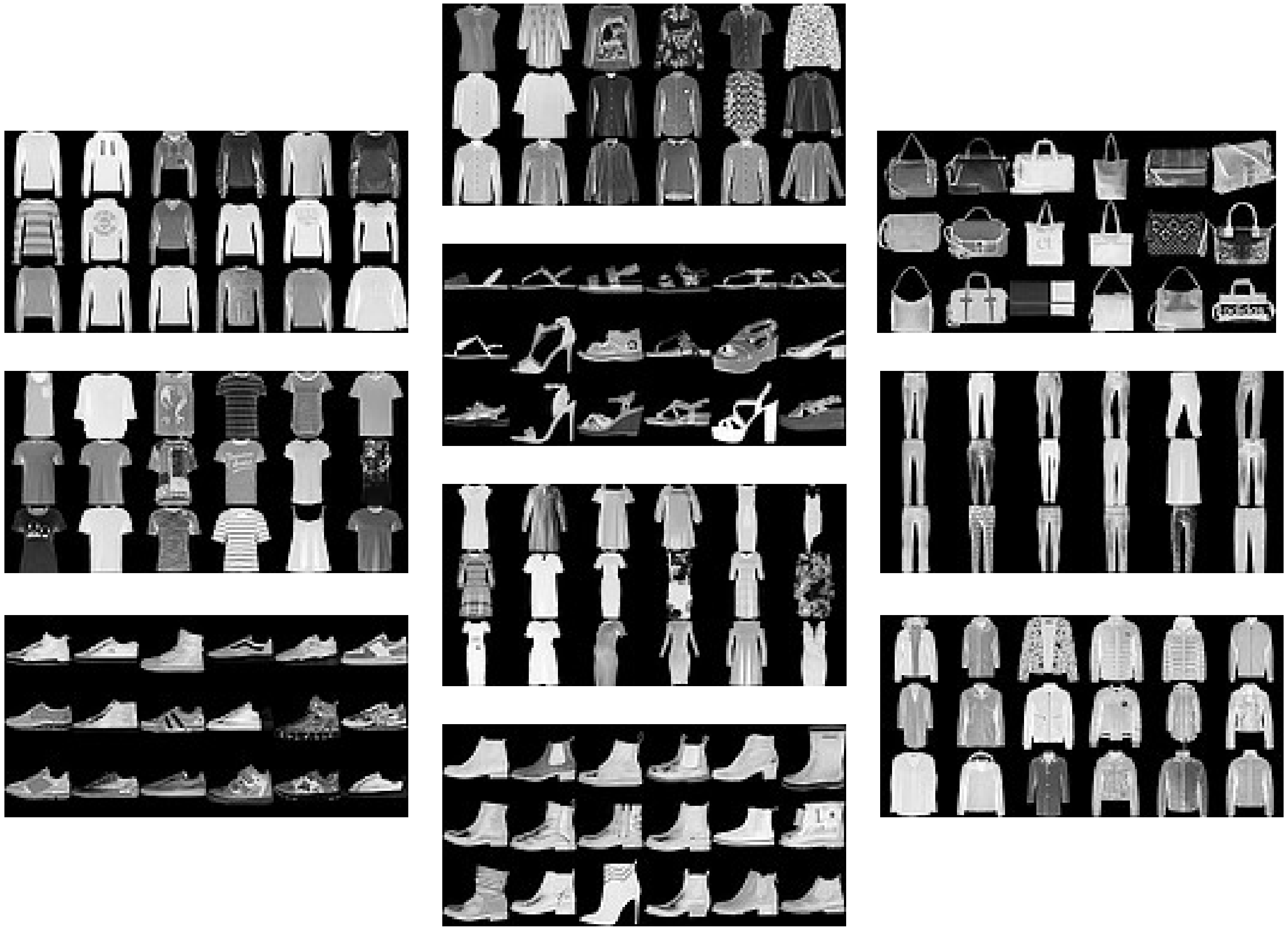
- 70000 total grayscale images
- 28x28 pixels image resolution
- 10 total class labels
- 60000 training images (6000 images per class label)
- 10000 testing images (1000 images per class label)

Fashion MNIST Dataset

Class labels

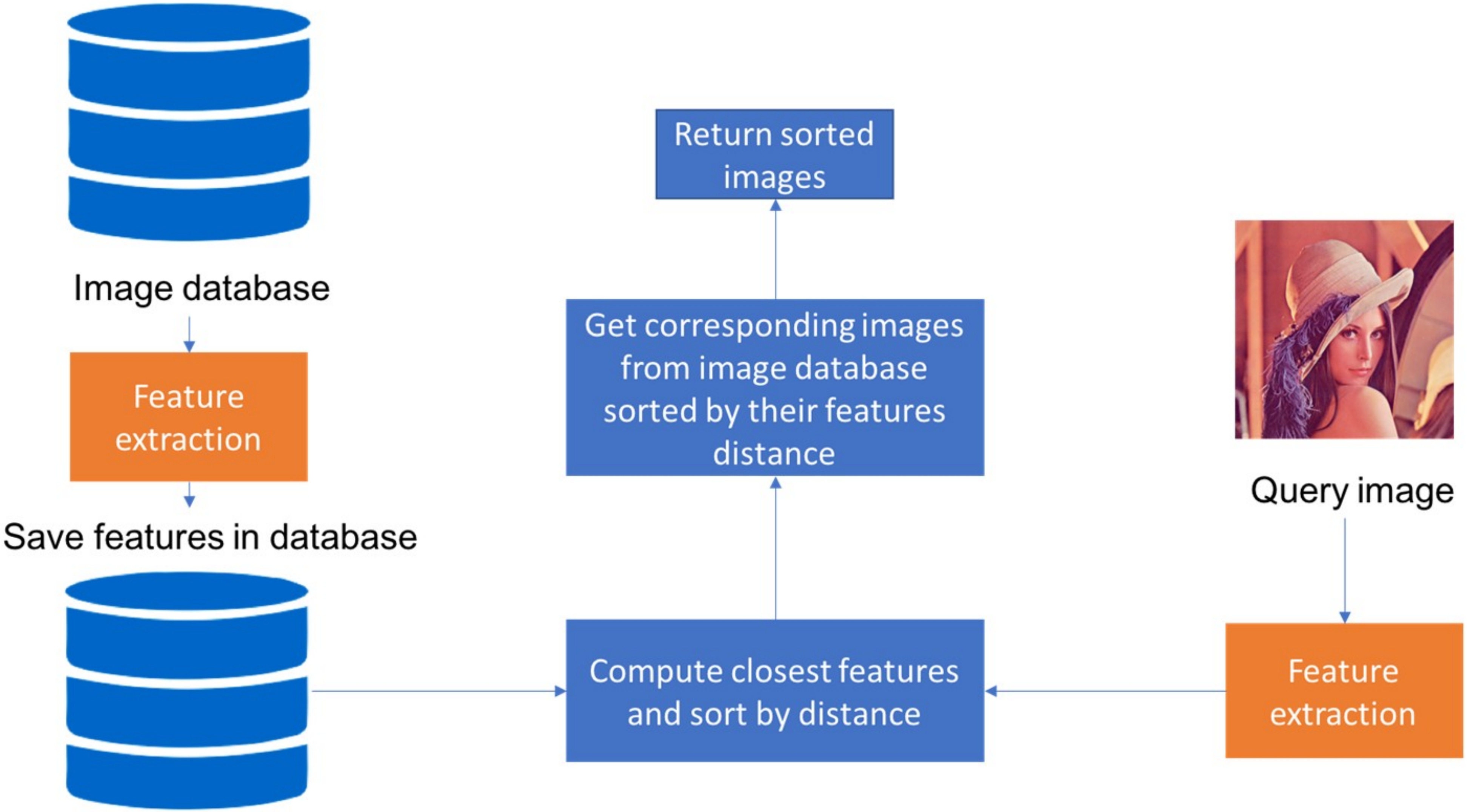
- T-shirt/top
- Trouser
- Pullover
- Dress
- Coat
- Sandal
- Shirt
- Sneaker
- Bag
- Ankle boot

Fashion MNIST Dataset
Samples



Content-based Image Retrieval (CBIR)

The general idea



A simple CBIR prototype

Using a convolutional deep neural network and the KNN algorithm

How it works:

1. Train a convolutional deep neural network with the Fashion MNIST for multi-label classification
2. Extract the feature vectors of all dataset images as represented internally in the neural network
3. Store the extracted image feature vectors in a feature database
4. Fit a KNN model using the extracted image feature vectors stored in the feature database
5. For any input query image:
    - Extract the image feature vector using the same technique used in step 2
    - Use the image feature vector with the KNN model to find the closest K neighbor images
    - Present the closest K images sorted by distance

Convolutional Neural Network Classifier

Summary and architecture of the neural network

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 28, 28, 64)        320
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 14, 14, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 14, 14, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 14, 14, 32)        8224
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 7, 7, 32)          0
_________________________________________________________________
dropout_2 (Dropout)          (None, 7, 7, 32)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 1568)              0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               401664
_________________________________________________________________
dropout_3 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 10)                2570
=================================================================
Total params: 412,778
Trainable params: 412,778
Non-trainable params: 0
```

Conv(64, Kernel(2, 2), Padding(Same))+Relu
MaxPooling(PoolSize(2, 2), Strides(2, 2))
Dropout(0.3)
Conv(32, Kernel(2, 2), Padding(Same))+Relu
MaxPooling(PoolSize(2, 2), Strides(2, 2))
Dropout(0.3)
FC(256)+Relu
Dropout(0.5)
FC(10)+Softmax

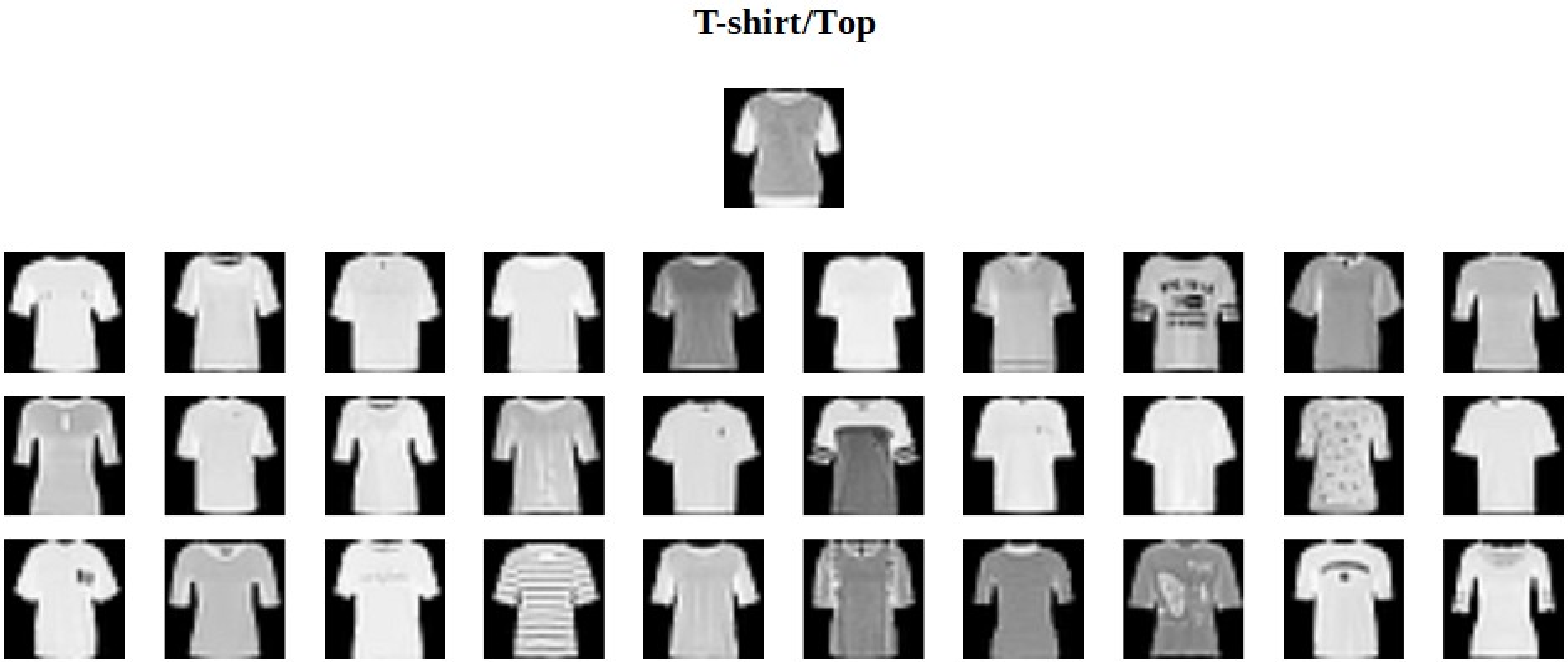
Convolutional Neural Network Classifier

Training information

Dataset	Fashion MNIST
Total Classes	10
Total Images	70000
Training Images	55000
Validation Images	5000
Testing Images	10000
Images Resolution	28x28
Images Normalization	MinMax
Learning Rate	1e-3
Training Algorithm	Backpropagation
Optimization Method	Adam
Loss Function	Cross-entropy
Batch Size	64
Training Epochs	10

Example Queries

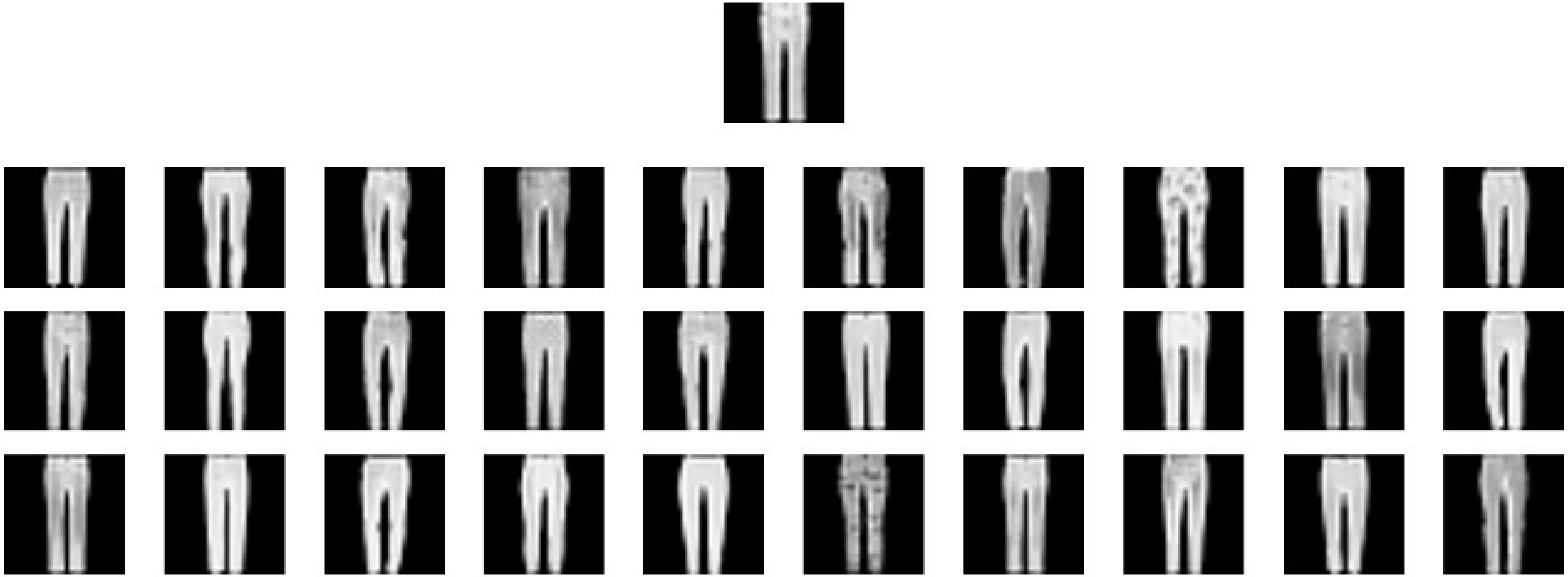
Example 1 (closest K=30 neighbors)



Example Queries

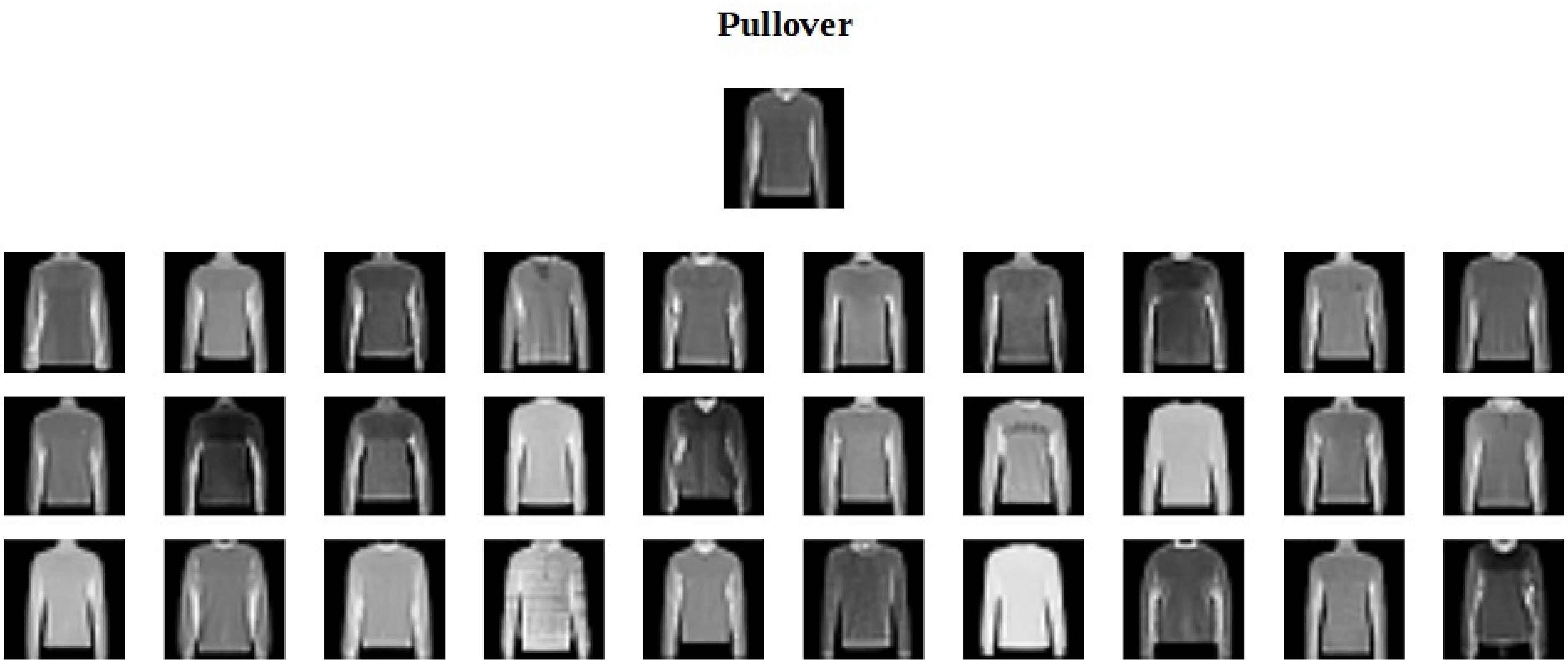
Example 2 (closest K=30 neighbors)

Trouser



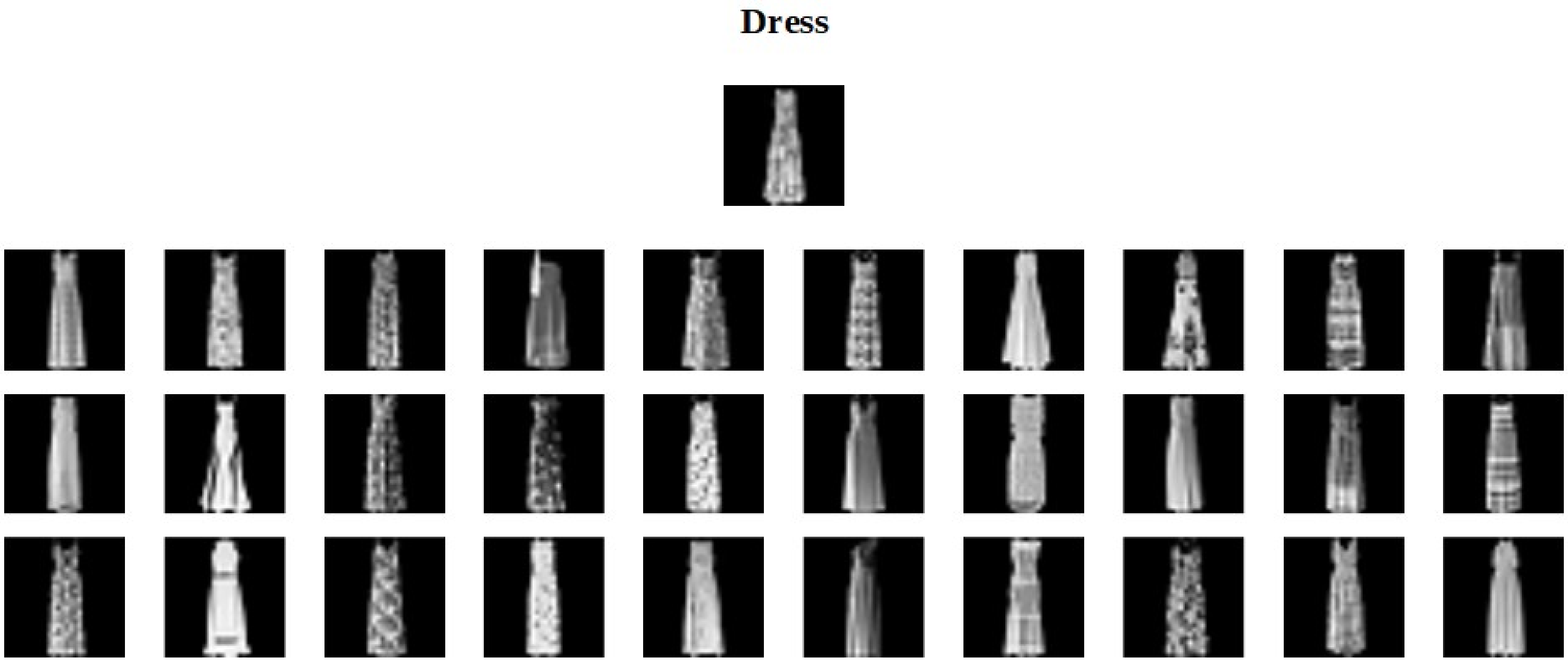
Example Queries

Example 3 (closest K=30 neighbors)



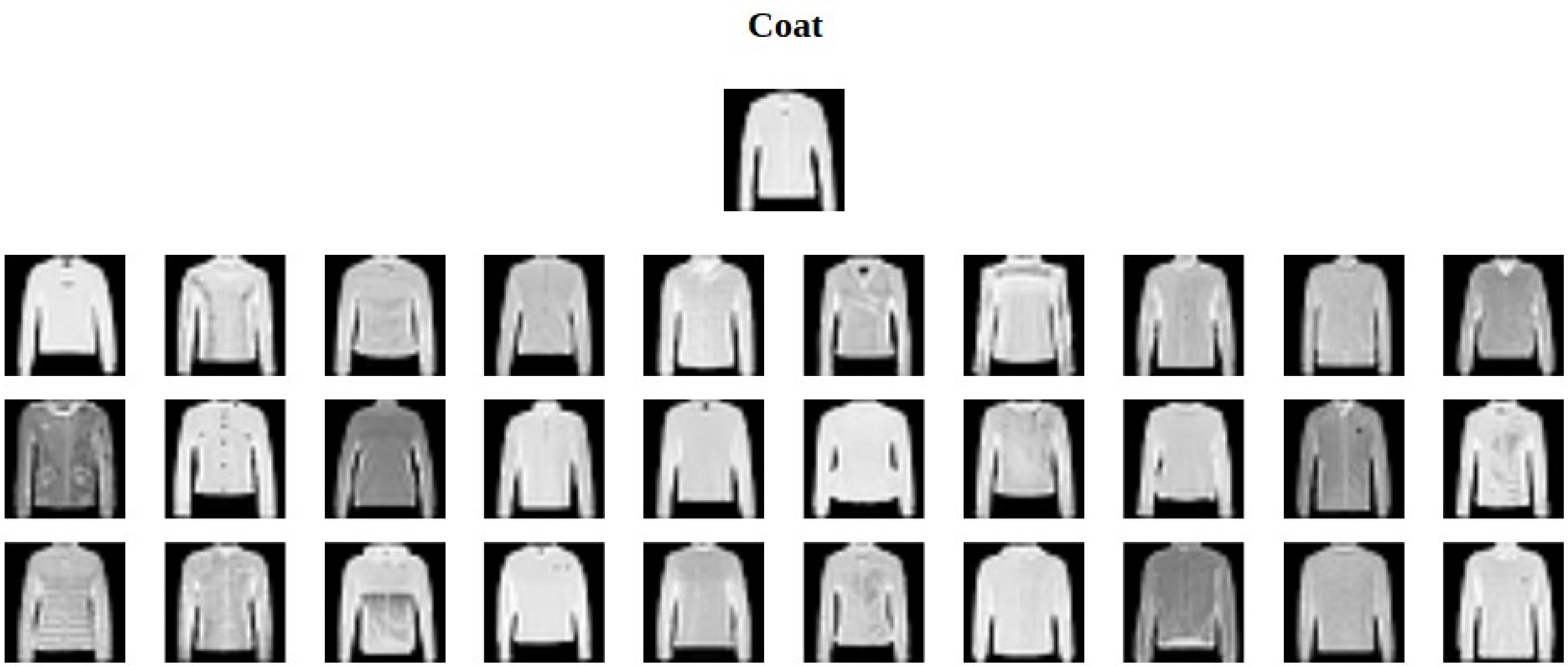
Example Queries

Example 4 (closest K=30 neighbors)



Example Queries

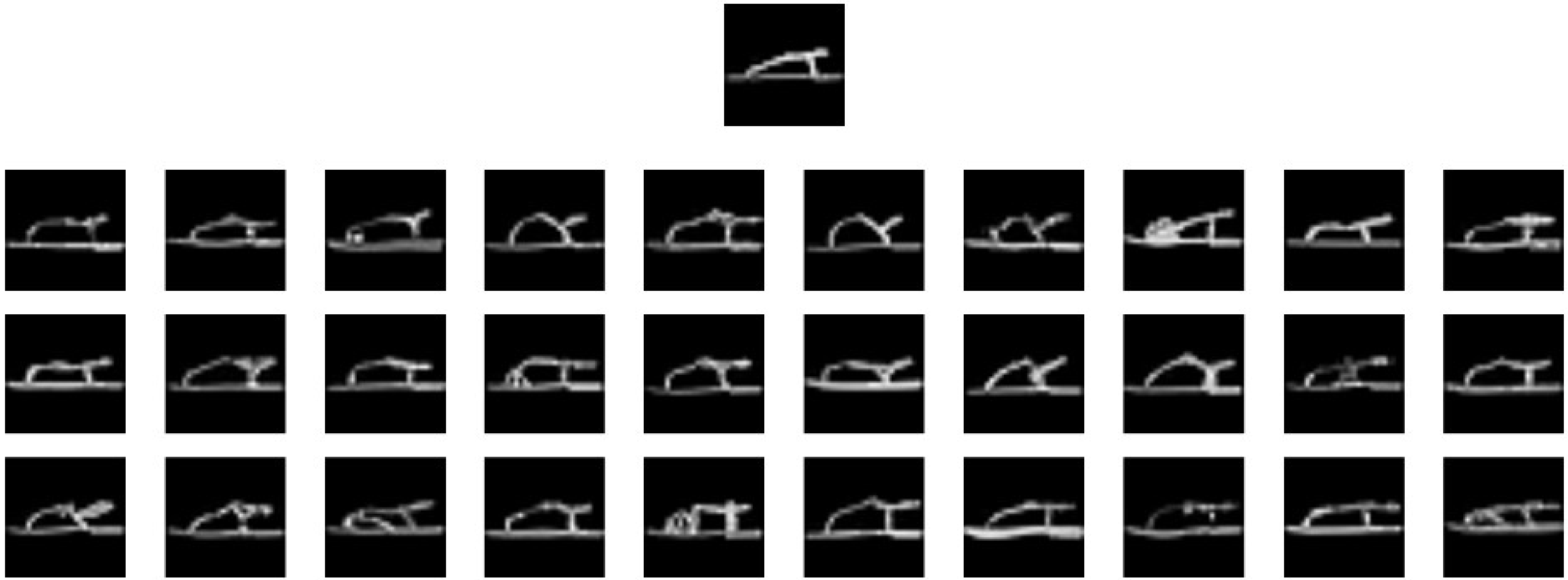
Example 5 (closest K=30 neighbors)



Example Queries

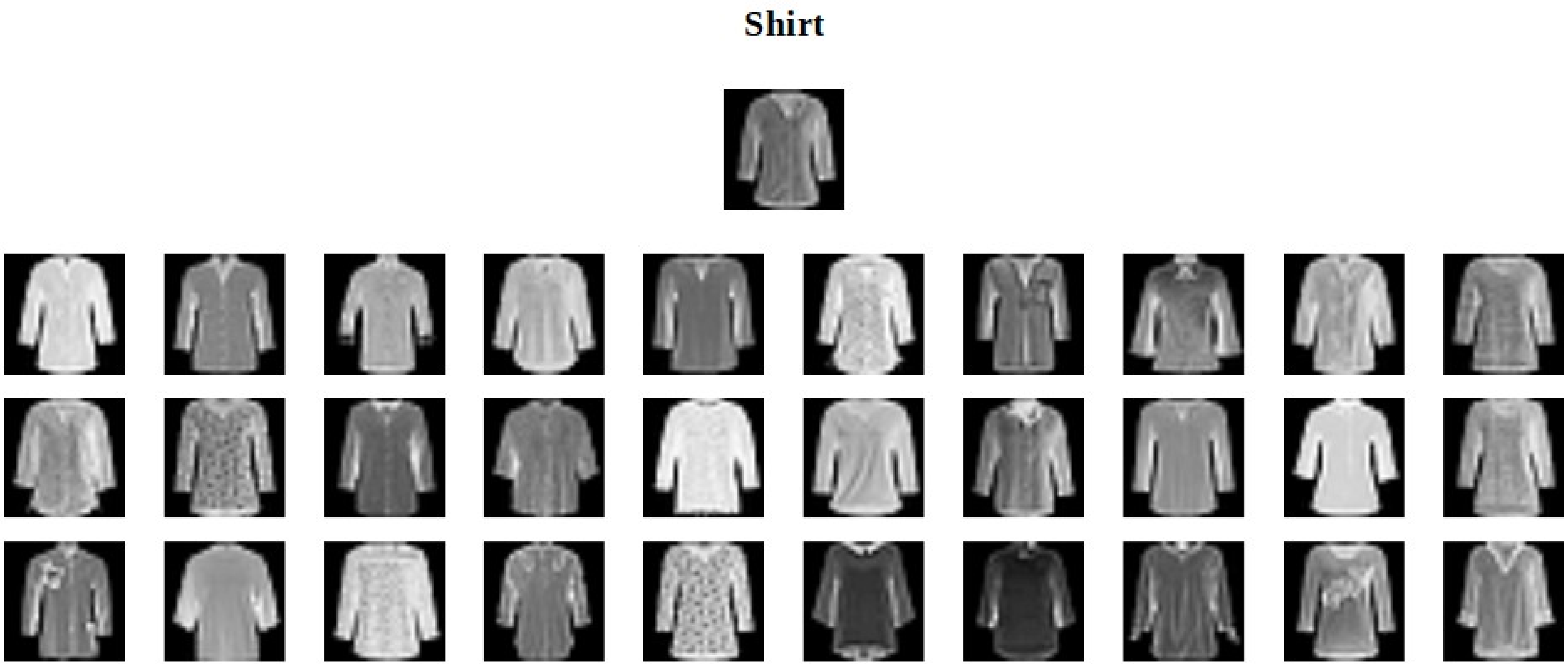
Example 6 (closest K=30 neighbors)

Sandal



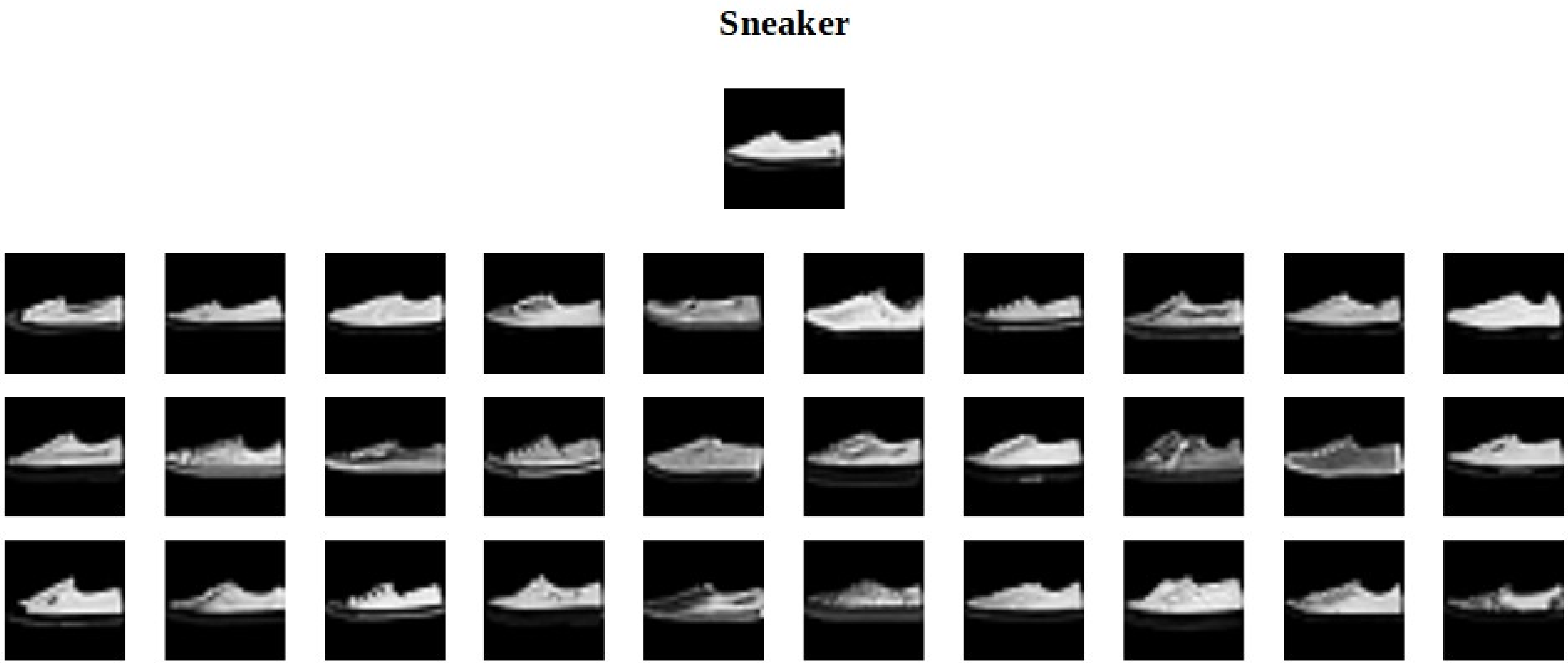
Example Queries

Example 7 (closest K=30 neighbors)



Example Queries

Example 8 (closest K=30 neighbors)



Example Queries

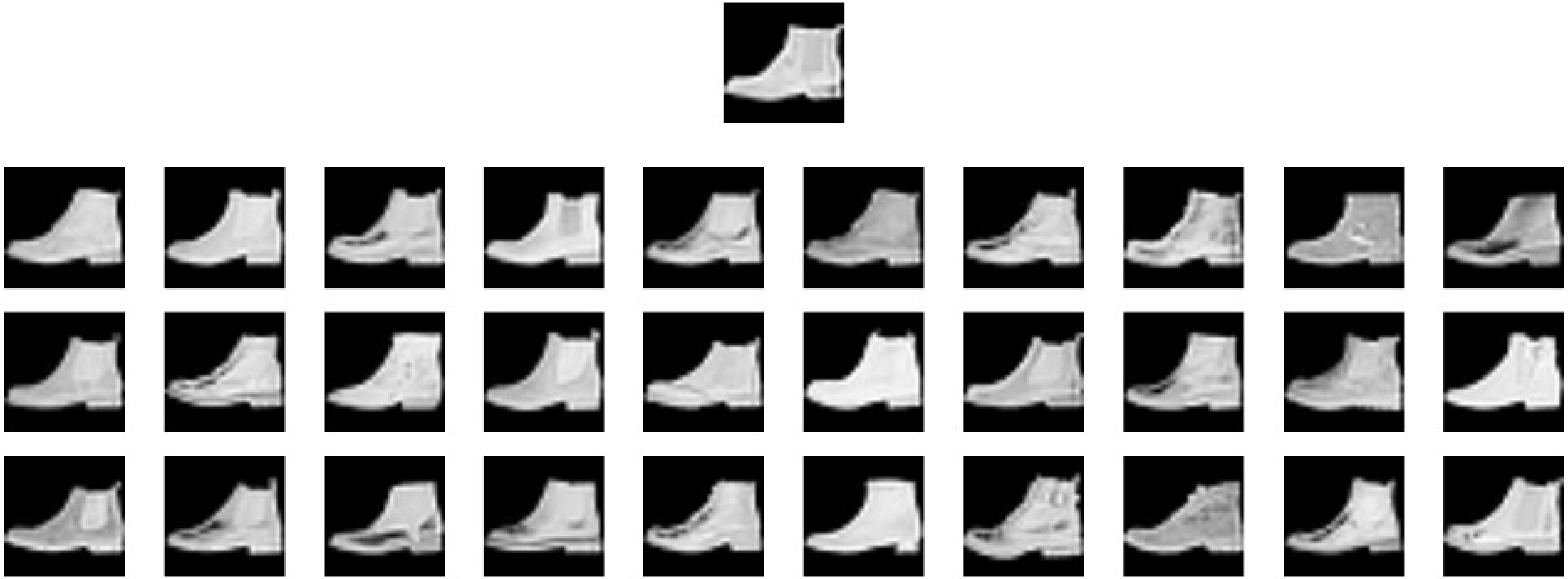
Example 9 (closest K=30 neighbors)

Bag

Example Queries

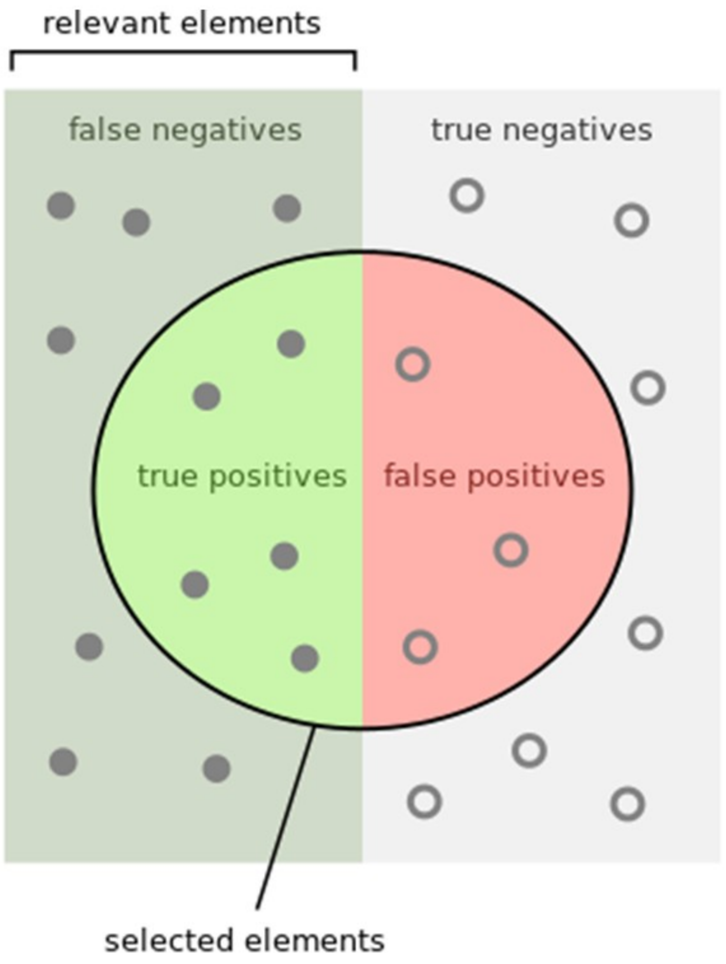
Example 10 (closest K=30 neighbors)

Ankle Boot



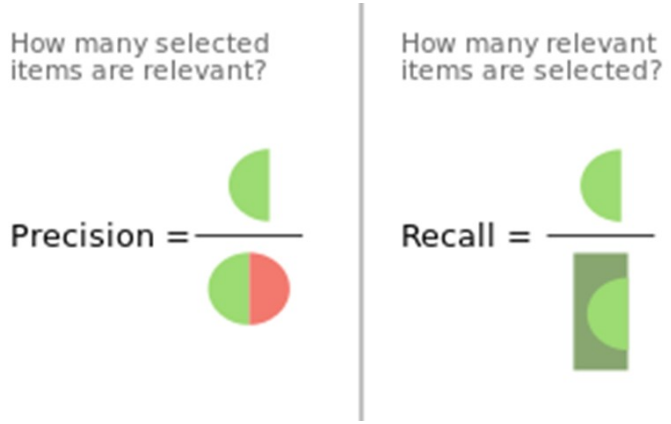
Metrics

Precision & Recall



$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$



Experiments

Using various similarity metrics, neural network layers, K sizes

We have calculated average precision and recall metrics for experiments using:

- 100 different random test images as search input queries
- various K sizes ranging from 1 to 500
- feature vectors extracted from various neural network layers
- various similarity metrics (Cosine, Euclidean, etc)

Results

CSV File Format

- The results are stored in a CSV file with the following format:

  Neural Network Layer, Similarity Metric, K size, Avg Precision (%), Avg Recall (%)

- Example data:

  flatten_1, euclidean, 1, 87.0, 0.015
  …
  flatten_1, cosine, 500, 71.6, 5.970
  …
  dense_1, euclidean, 1, 94.0, 0.016
  …
  dense_1, cosine, 500, 84.8, 7.074

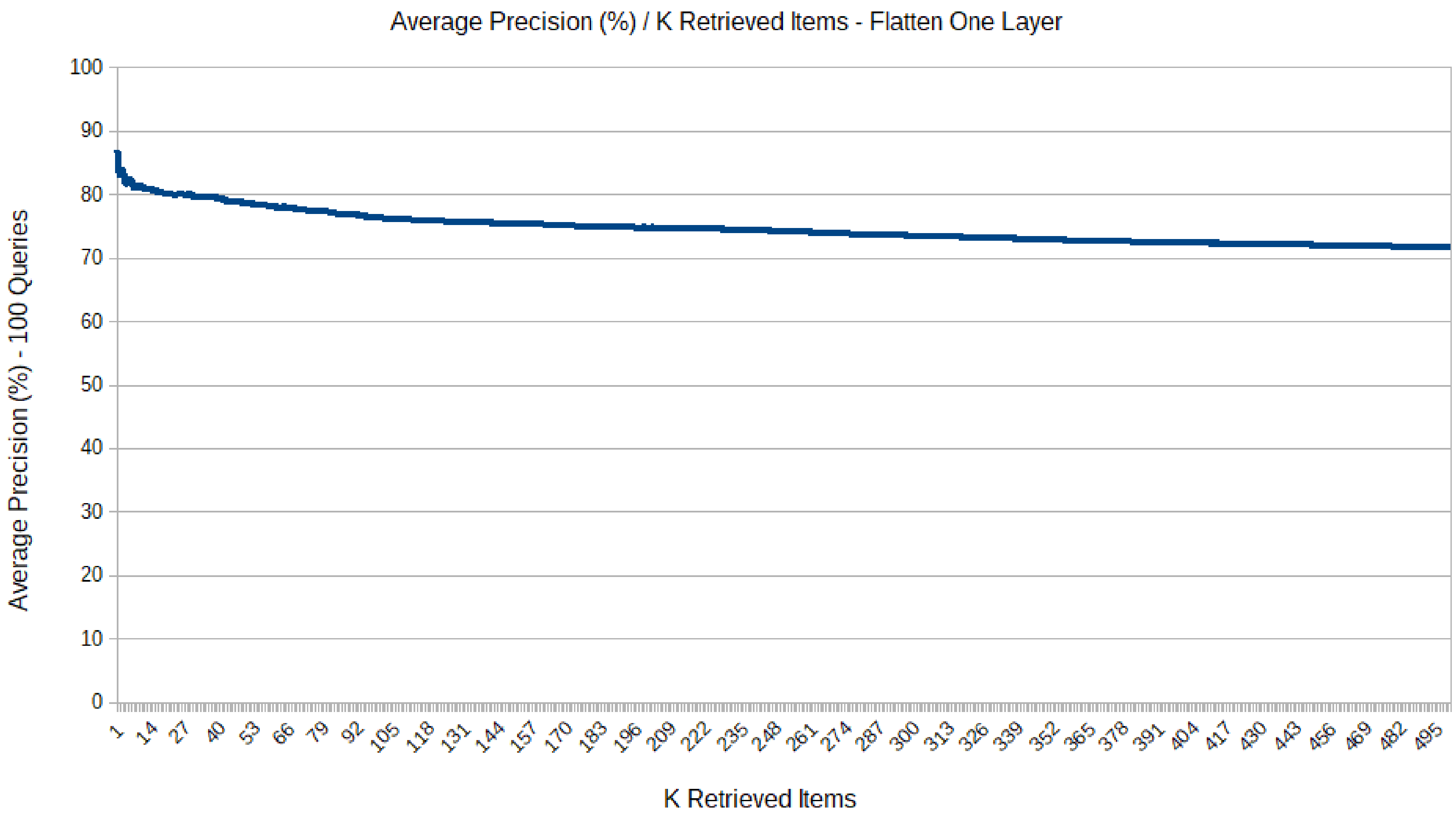
Results

Conclusions

- When K size increases the precision decreases
- When K size increases the recall increases
- The feature vectors extracted from 'dense_1' layer give better precision, recall results
- The cosine similarity gives better precision, recall results
- In general there is a trade-off between precision and recall metrics

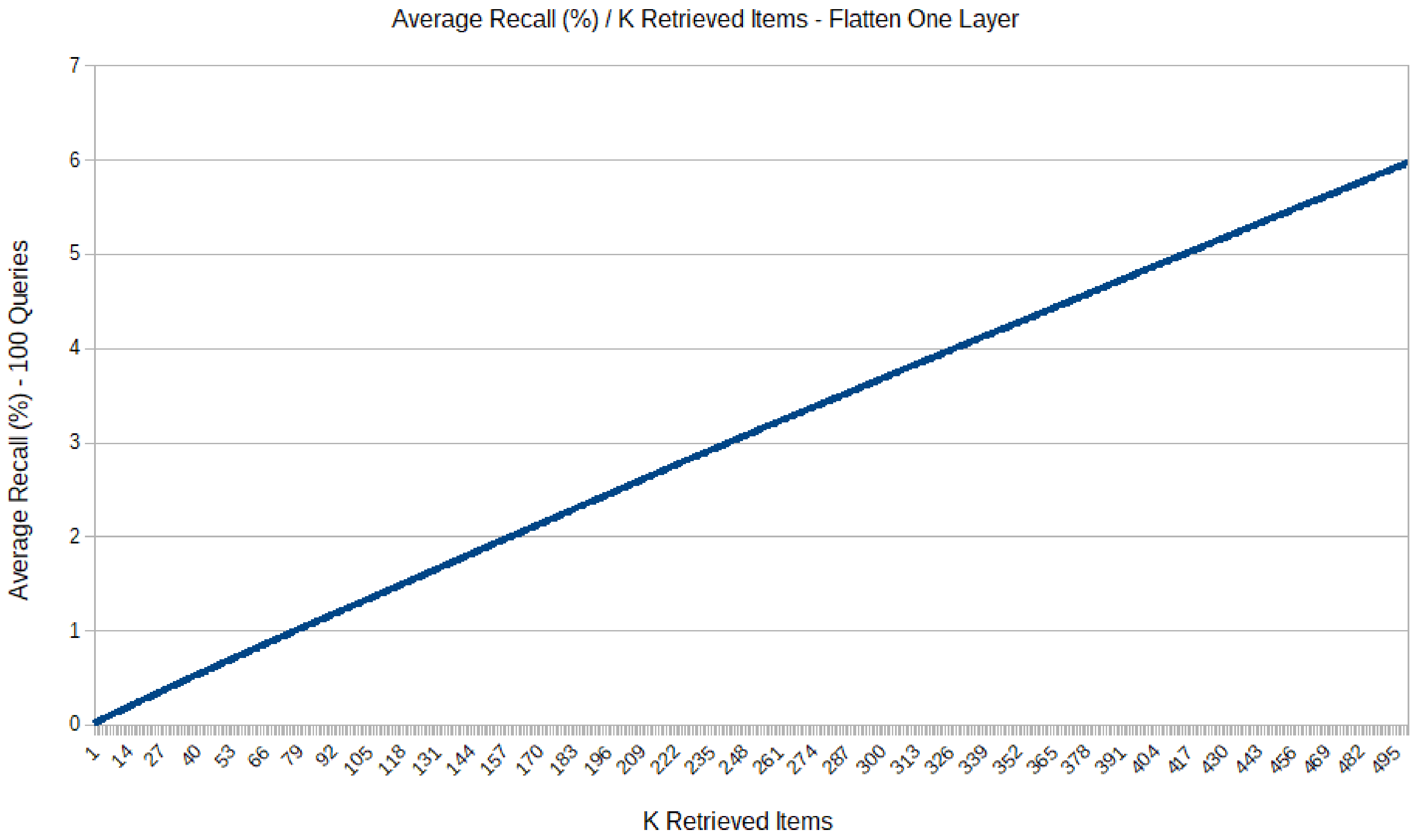
Curves

Average Precision / K Retrieved Items (for flatten_1 layer)



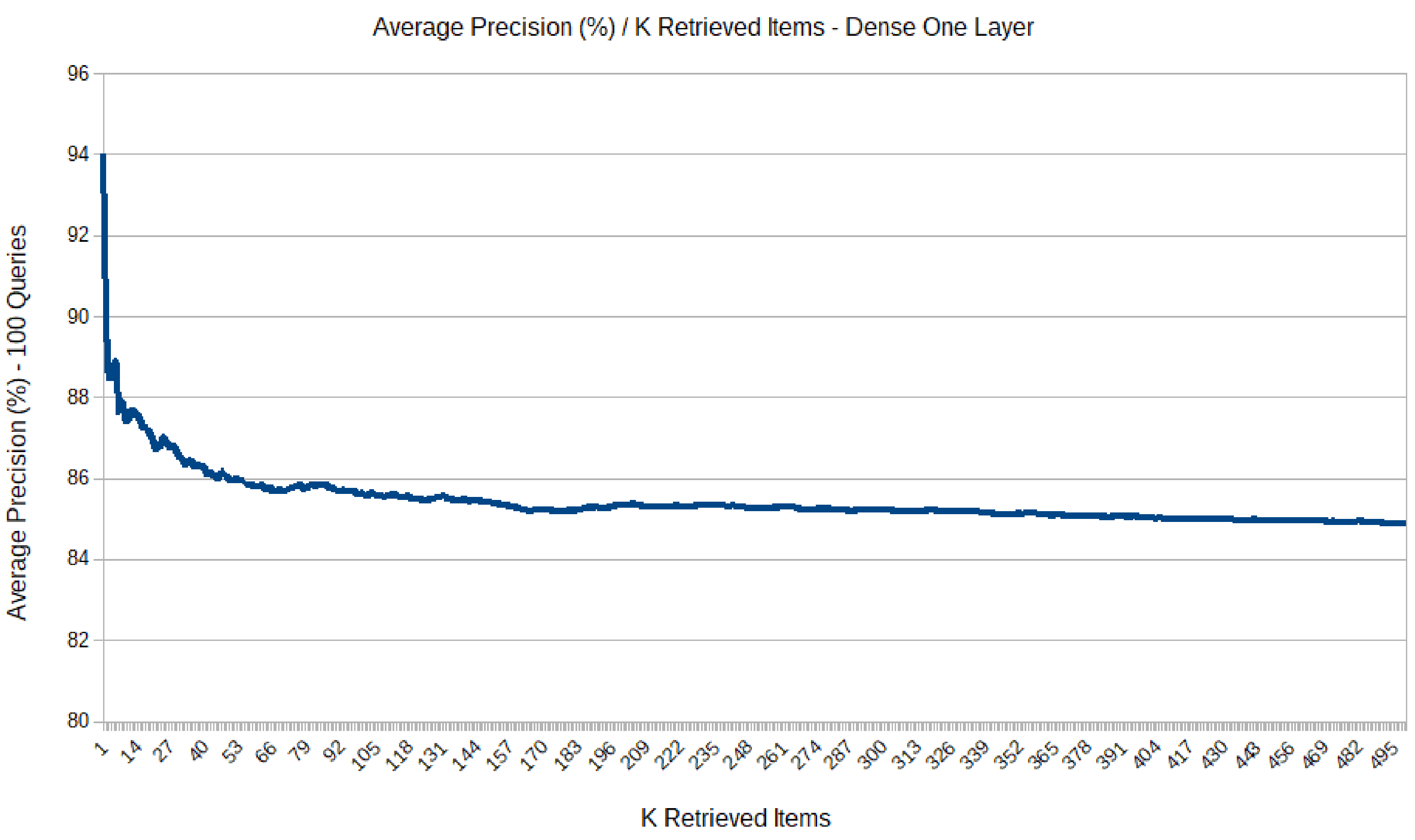
Curves

Average Recall / K Retrieved Items (for flatten_1 layer)



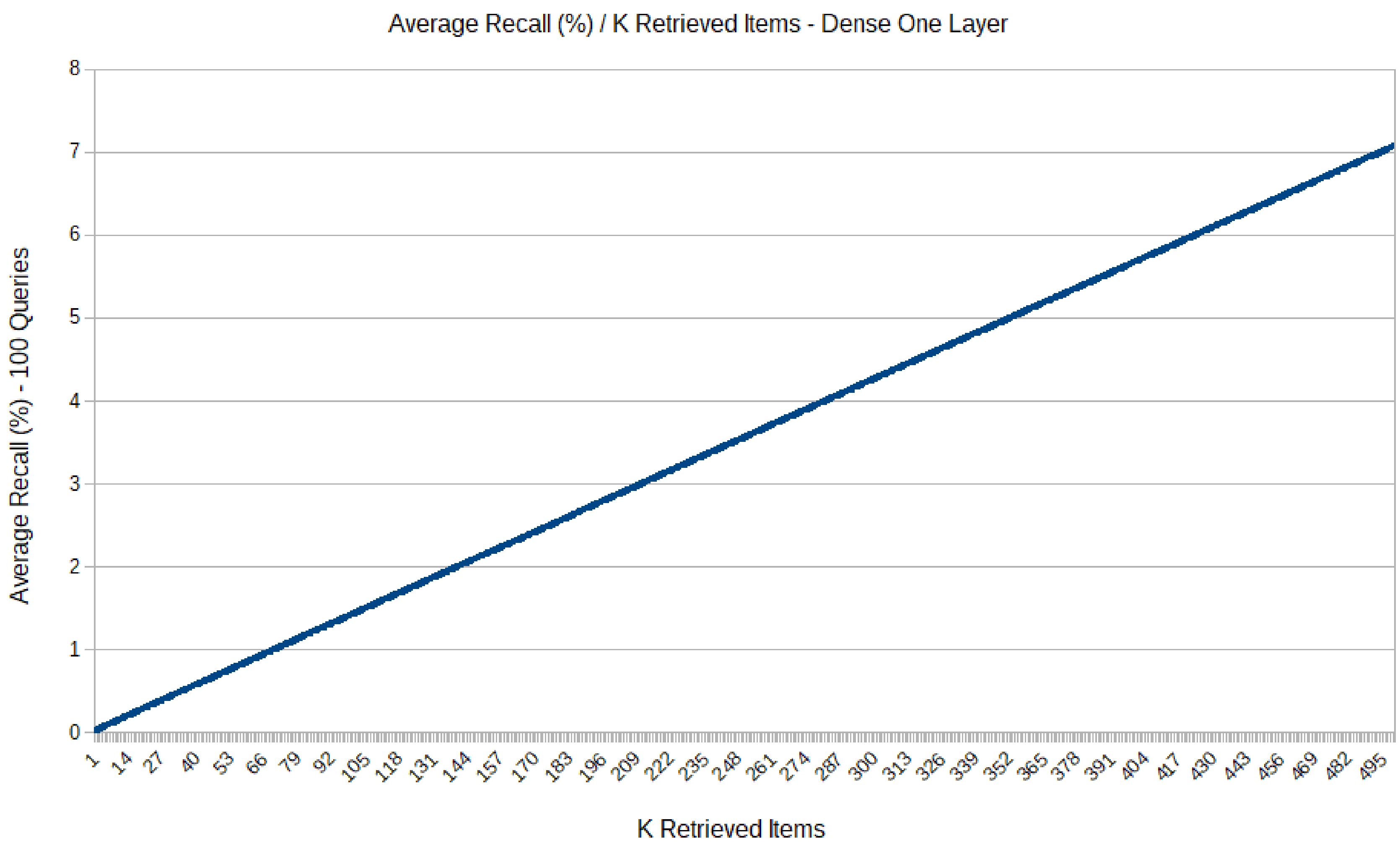
Curves

Average Precision / K Retrieved Items (for dense_1 layer)



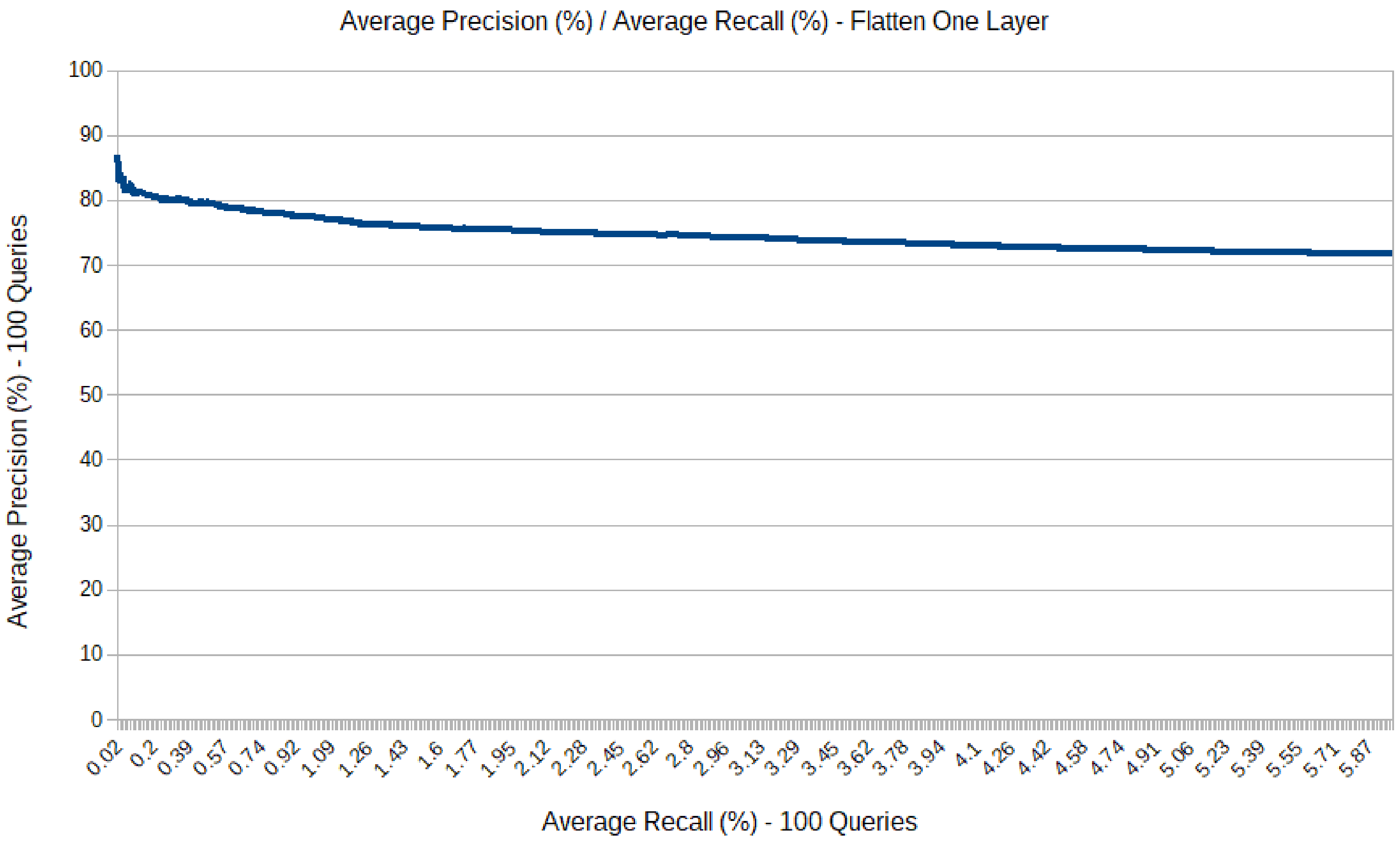
Curves

Average Recall / K Retrieved Items (for dense_1 layer)



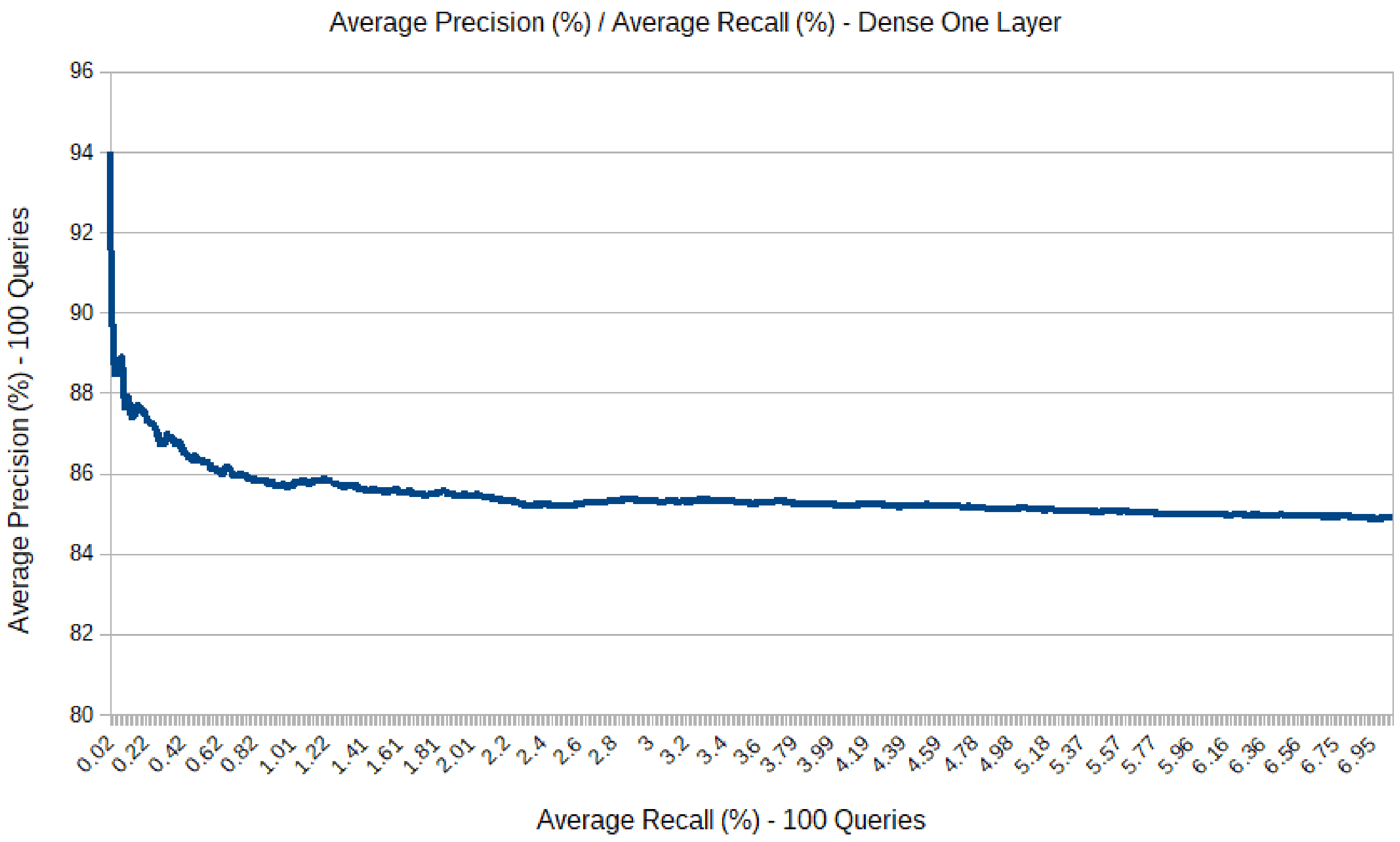
Curves

Average Precision / Average Recall (for flatten_1 layer)



Curves

Average Precision / Average Recall (for dense_1 layer)



Software Requirements

- Programming Languages : Python
- IDEs : Spyder, JetBrains PyCharm
- Application-level Dependency Managers: Anaconda
- Python Modules :

Module	Version
sklearn	0.19.1
numpy	1.14.3
matplotlib	2.2.2
keras	2.1.6
tensorflow	1.8.0

Thank you a lot and have a nice day!