STYRIA
DATA SCIENCE

www.styria.ai

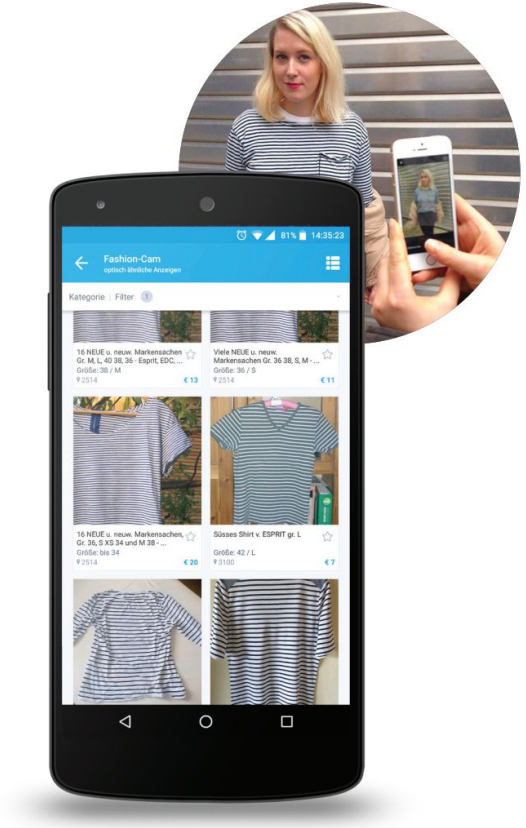
# About Us

- Styria
  - The biggest media company in the SE Europe
  - Reach up to 90% of Internet users
- Styria Data Science Team
  - Formed in early 2015.
  - 10 people
  - Working for clients within the group and many international companies

# Visual similarity search

# FashionCam - Visual Search on willhaben.at

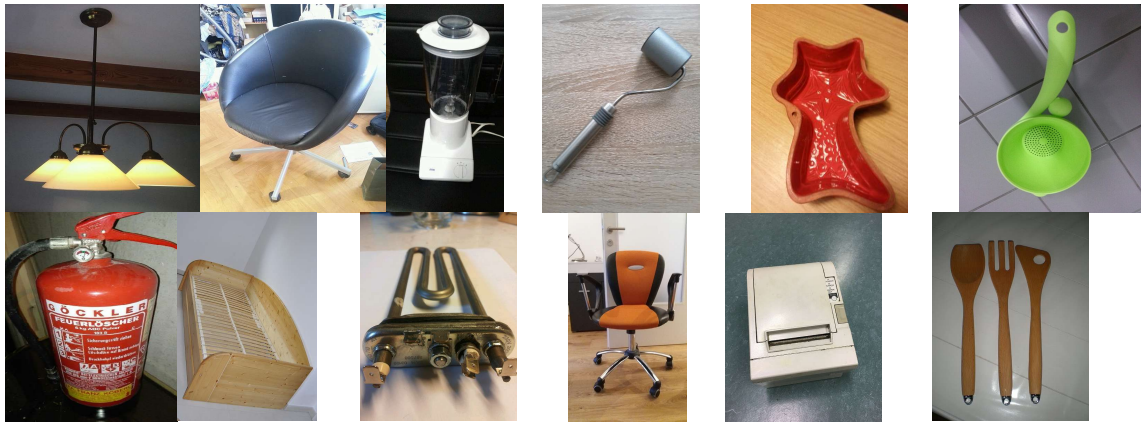
- Cutting edge R&D in the field of deep learning
  - mostly focused on classifieds industry (online 2nd hand marketplaces)
- Visual search
  - 230 fashion categories
  - 5 million images training set
  - 3.2 million neurons with 6.1 million weights
  - 200ms results returned to the user
- Classifieds visual search app
  - First such app in the world
  - Making Willhaben users happy since 2016
- Patent pending



# What is Visual Search?

## Query

## Dataset



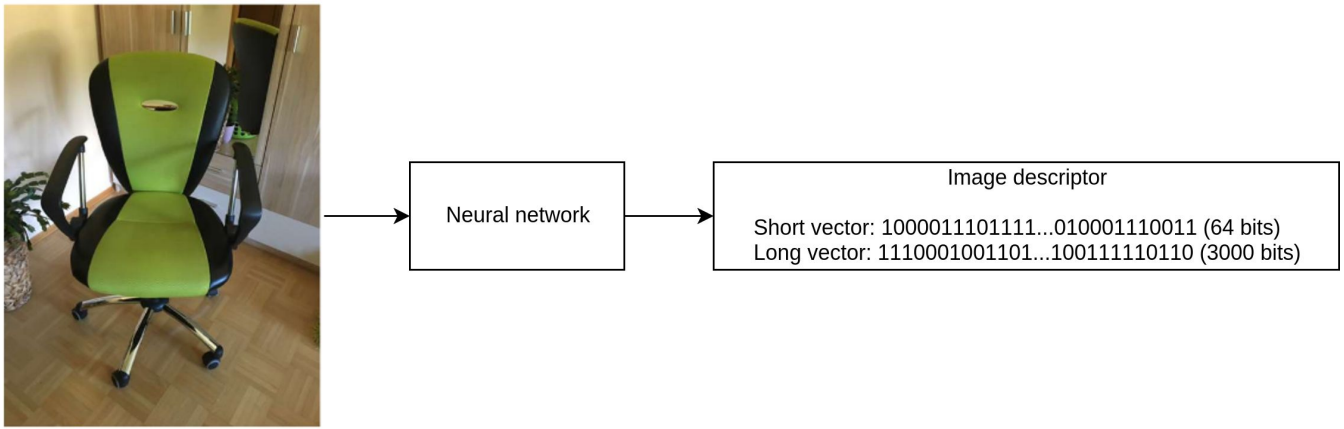

**Which image is the most similar to given image?**

# Building a model

- Train a deep CNN to classify objects
- GoogleNet v1, Darknet
- Use learnt features for semantic/visual similarity
  - Create descriptors from activations inside neural network
- Domain specifics - e.g. fashion:
  - Color - pre-train on generated triplets
  - Brand - without annotations - NLP

# Image descriptor

- Images are not comparable
- Use neural network for feature extraction
- Each image gets a descriptor
  - Similarity metric: cosine distance, hamming distance...



# Descriptor extraction

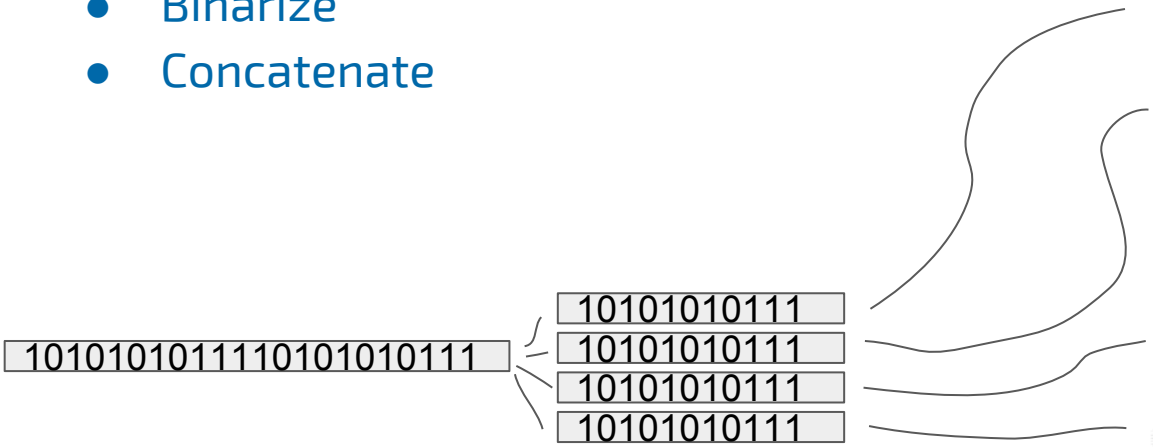
- Layers
  - Layers close to input extract simple features
    - Color blobs, lines, basic texture
  - Layers close to output extract semantic features
    - Complex shapes, image classes
  - Idea
    - Combine outputs from multiple layers
- Problems
  - Very high dimensionality, for example Darknet19:
    - Conv3: 56 x 56 x 128 =  401408 dimensions
    - Conv13: 14 x 14 x 512 = 100352 dimensions

# Descriptor extraction

- Global pooling
  - Reduce each feature map to a single number by averaging activations
    - 28 x 28 x 128 -> 1 x 1 x 128
    - 14 x 14 x 512 -> 1 x 1 x 512
- Autoencoders
  - Pros
    - Arbitrary dimension of latent space
    - Can learn good representations
  - Cons
    - Complicates the architecture
    - Hard to train

# Descriptor extraction

- Global average pooling of different layers of
- Binarize
- Concatenate



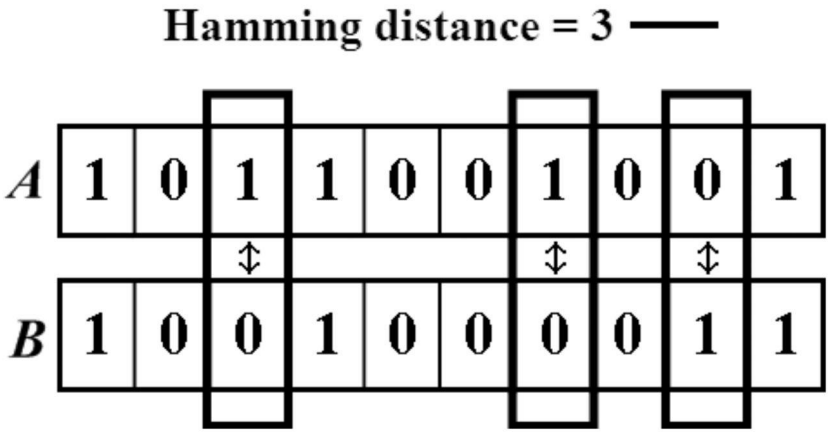

| Type | Filters | Size/Stride | Output |
|---|---|---|---|
| Convolutional | 32 | 3 × 3 | 224 × 224 |
| Maxpool | | 2 × 2/2 | 112 × 112 |
| Convolutional | 64 | 3 × 3 | 112 × 112 |
| Maxpool | | 2 × 2/2 | 56 × 56 |
| Convolutional | 128 | 3 × 3 | 56 × 56 |
| Convolutional | 64 | 1 × 1 | 56 × 56 |
| Convolutional | 128 | 3 × 3 | 56 × 56 |
| Maxpool | | 2 × 2/2 | 28 × 28 |
| Convolutional | 256 | 3 × 3 | 28 × 28 |
| Convolutional | 128 | 1 × 1 | 28 × 28 |
| Convolutional | 256 | 3 × 3 | 28 × 28 |
| Maxpool | | 2 × 2/2 | 14 × 14 |
| Convolutional | 512 | 3 × 3 | 14 × 14 |
| Convolutional | 256 | 1 × 1 | 14 × 14 |
| Convolutional | 512 | 3 × 3 | 14 × 14 |
| Convolutional | 256 | 1 × 1 | 14 × 14 |
| Convolutional | 512 | 3 × 3 | 14 × 14 |
| Maxpool | | 2 × 2/2 | 7 × 7 |
| Convolutional | 1024 | 3 × 3 | 7 × 7 |
| Convolutional | 512 | 1 × 1 | 7 × 7 |
| Convolutional | 1024 | 3 × 3 | 7 × 7 |
| Convolutional | 512 | 1 × 1 | 7 × 7 |
| Convolutional | 1024 | 3 × 3 | 7 × 7 |
| Convolutional | 1000 | 1 × 1 | 7 × 7 |
| Avgpool | | Global | 1000 |
| Softmax | | | |

**Table 6: Darknet-19.**

*[1] YOLO9000: Better, Faster, Stronger, https://arxiv.org/abs/1612.08242*

# Hamming distance

- Count of unequal bits in two binary represented numbers
  - Smaller distance -> more similar images
  - If two binary represented numbers are the same -> 0
- Fast to calculate - binary XOR using AVX2 instructions



# Selecting the Features

STYRIA DATA SCIENCE

- How much should each layer contribute the similarity?
  - Create a small dataset with desired clusters
    - 10 red shirts, 10 white shirts, 10 shirts with stripes …
    - 10 MTB, 10 road bikes …
  - For given selection of features find the weight of each layers contribution
  - Optimize so we get the desired clusters

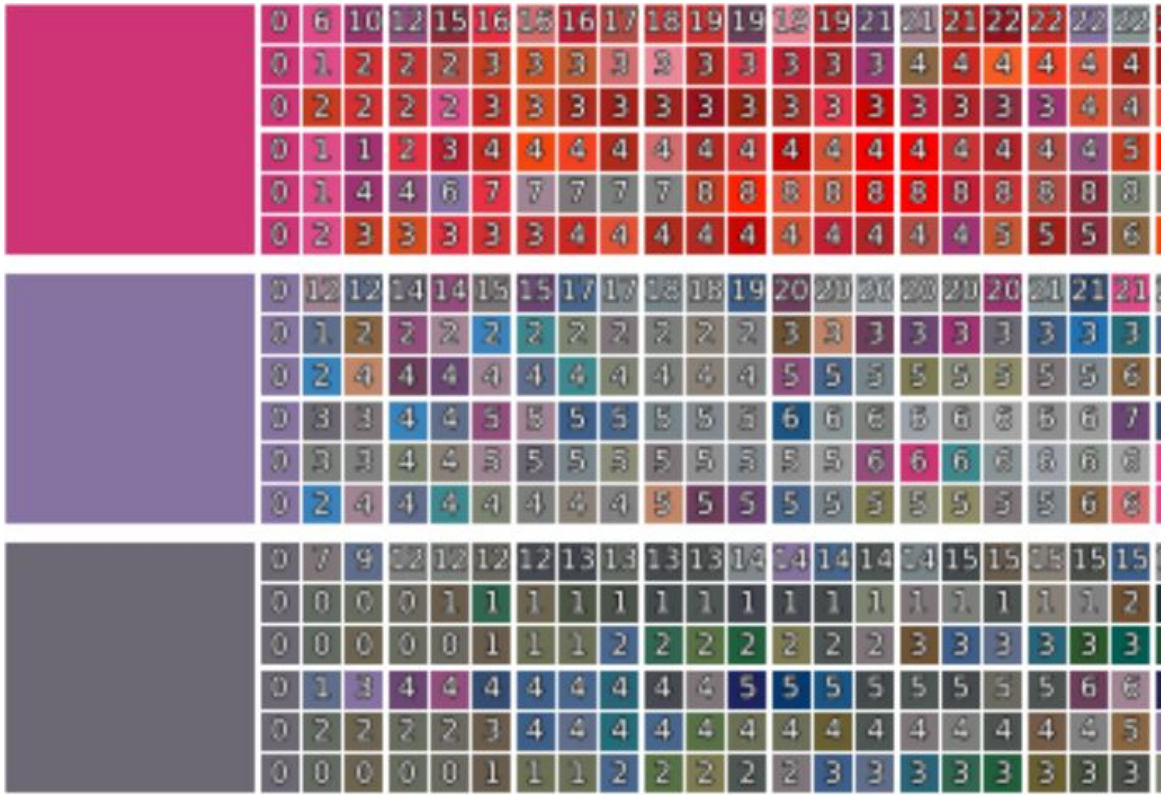
# Selecting the Features

STYRIA
DATA SCIENCE

# Adding custom features - color

- Color Space
  - Distances in RGB space do not correspond how we perceive color differences
  - Distances in CIELAB color space are more appropriate for human perception
- Models
  - In classifieds middle of an image is representative of an object color
  - Learn to rank the colors
  - Hand feature design
  - Triplet neural network architecture
- Evaluation
  - Ground truth of distances in CIELAB space

# Adding custom features - color



# Adding custom features - brands

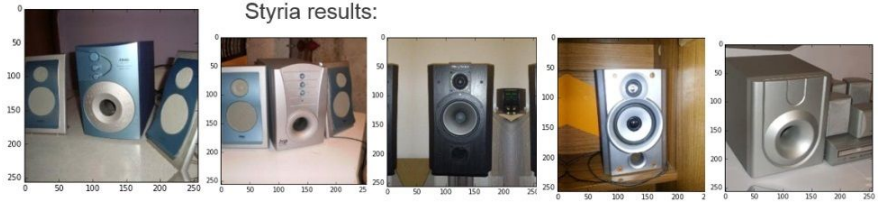
- In some categories brands are more important than visual features
- Use ad text to extract relevant features

| | stem | word | ig | word_count | max_ig | score |
|---|---|---|---|---|---|---|
| **3043** | salomon | salomon | 0.463040 | 1920 | 0.711390 | 0.329402 |
| **2825** | convers | converse | 0.469386 | 22762 | 0.689292 | 0.323544 |
| **4461** | puma | puma | 0.426182 | 28424 | 0.739303 | 0.315078 |
| **3207** | reebok | reebok | 0.458802 | 4106 | 0.635408 | 0.291526 |
| **6776** | nik | nike | 0.369508 | 45534 | 0.653182 | 0.241356 |
| **6555** | adidas | adidas | 0.374947 | 47462 | 0.622593 | 0.233440 |

# Performance & Quality

Query:

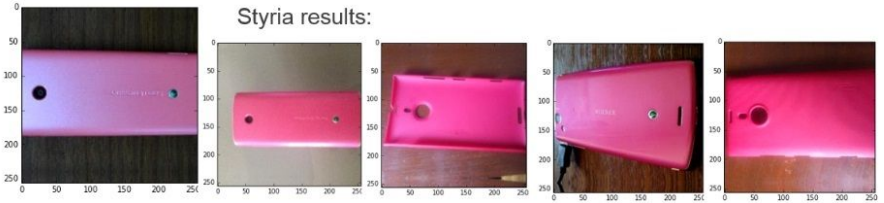
Styria results:



Google results:

Query:

Styria results:



Google results:

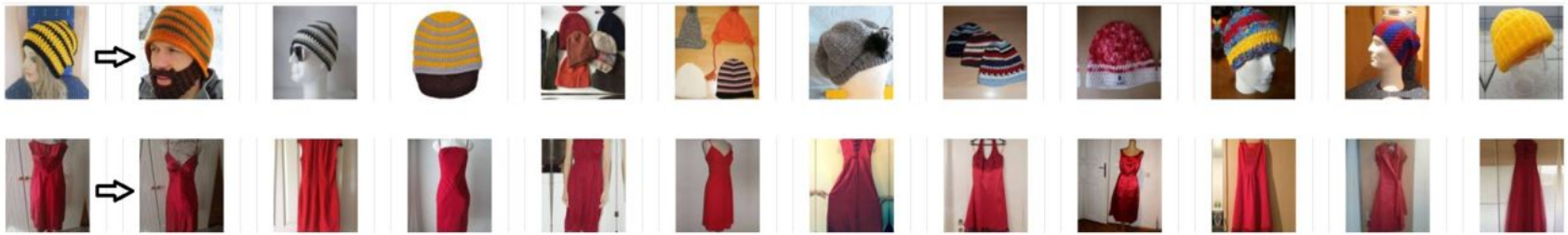
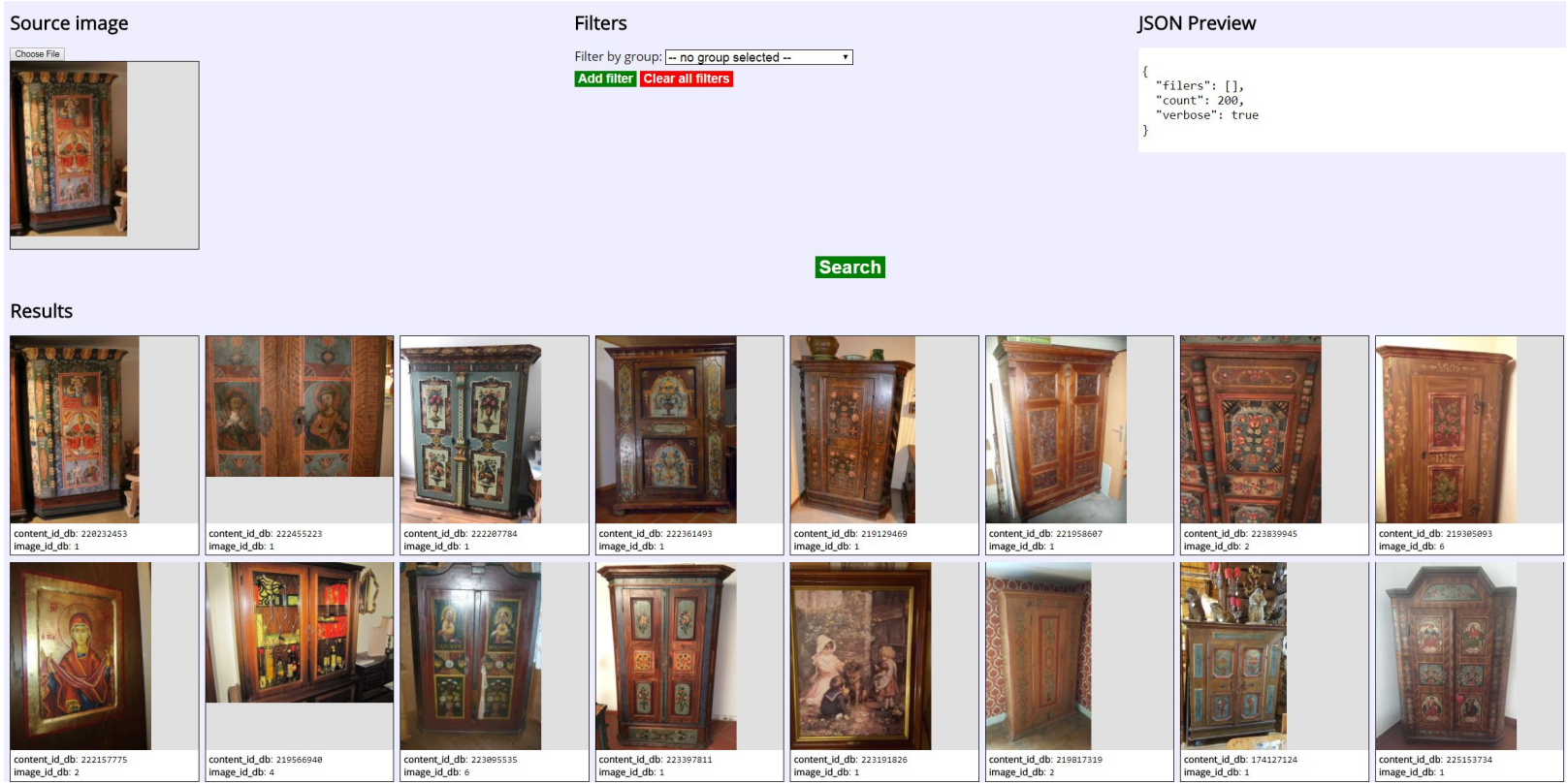


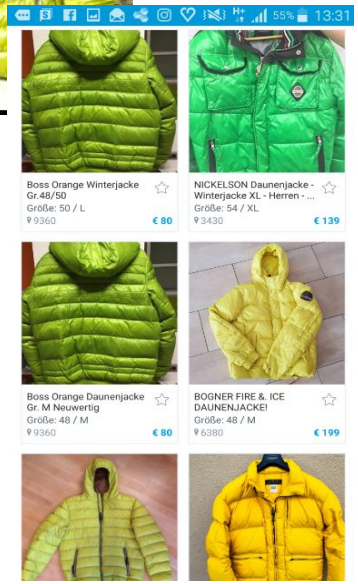

Image ID: 3967268
Group ID: 591
Similarity Score: 0
Image ID: 6323231
Group ID: 591
Similarity Score: 15
Image ID: 15859066
Group ID: 591
Similarity Score: 24
Image ID: 15807444
Group ID: 591
Similarity Score: 27
Image ID: 7168701
Group ID: 591
Similarity Score: 27
Image ID: 7894056
Group ID: 591
Similarity Score: 28
Image ID: 8067935
Group ID: 591
Similarity Score: 29
Image ID: 4855282
Group ID: 591
Similarity Score: 29

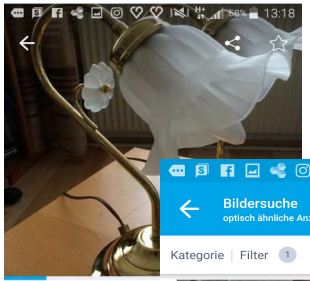
# Willhaben - expanding to other categories

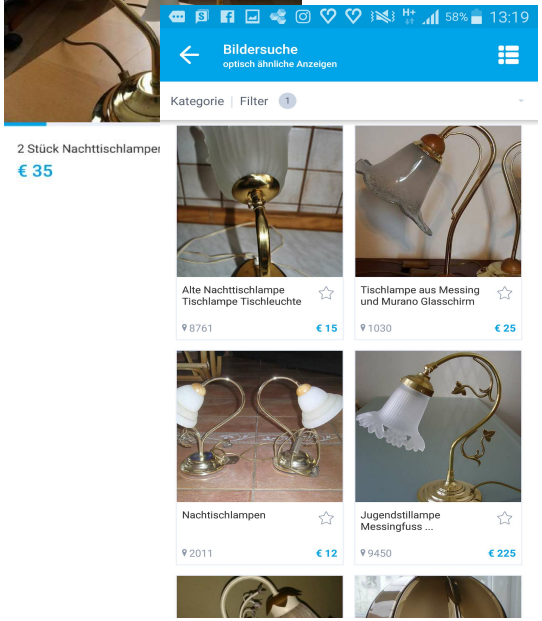
STYRIA DATA SCIENCE

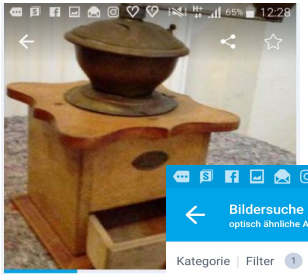


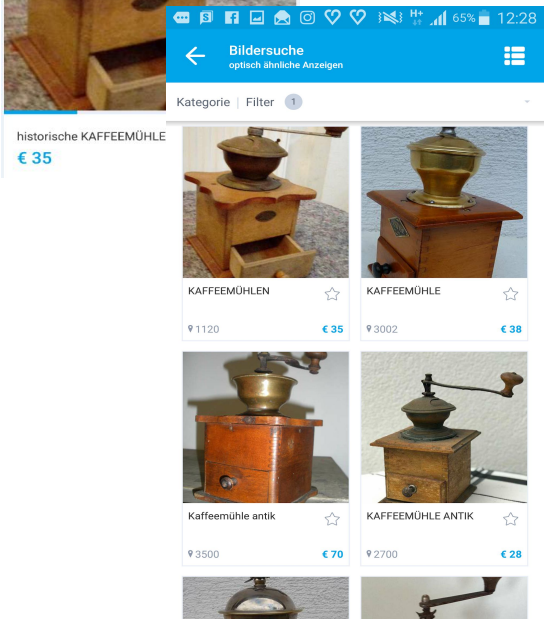
# Examples - Fashion & Antiques







## Featured image

Image ID: 870460

Group ID: 3282

Group name: Hauben

## model1


Image ID: 870460
Group ID: 3282
Similarity Score: 0


Image ID: 662447
Group ID: 3398
Similarity Score: 177


Image ID: 772647
Group ID: 3282
Similarity Score: 193


Image ID: 693154
Group ID: 3282
Similarity Score: 198


Image ID: 772648
Group ID: 3282
Similarity Score: 209


Image ID: 83815
Group ID: 3282
Similarity Score: 210


Image ID: 676999
Group ID: 3282
Similarity Score: 211


Image ID: 549229
Group ID: 3398
Similarity Score: 212


Image ID: 887818
Group ID: 3484
Similarity Score: 215


Image ID: 870440
Group ID: 3282
Similarity Score: 217

## model2


Image ID: 870460
Group ID: 3282
Similarity Score: 0


Image ID: 662447
Group ID: 3398
Similarity Score: 704


Image ID: 83814
Group ID: 3282
Similarity Score: 816


Image ID: 662449
Group ID: 3398
Similarity Score: 839


Image ID: 677000
Group ID: 3282
Similarity Score: 850


Image ID: 78490
Group ID: 3282
Similarity Score: 858


Image ID: 852467
Group ID: 3282
Similarity Score: 862


Image ID: 763696
Group ID: 3282
Similarity Score: 870


Image ID: 825771
Group ID: 3282
Similarity Score: 870


Image ID: 893282
Group ID: 3282
Similarity Score: 870

## model3


Image ID: 870460
Group ID: 3282
Similarity Score: 0


Image ID: 662447
Group ID: 3398
Similarity Score: 237


Image ID: 83815
Group ID: 3282
Similarity Score: 291


Image ID: 549229
Group ID: 3398
Similarity Score: 293


Image ID: 772647
Group ID: 3282
Similarity Score: 293


Image ID: 83814
Group ID: 3282
Similarity Score: 294


Image ID: 693154
Group ID: 3282
Similarity Score: 297


Image ID: 662449
Group ID: 3398
Similarity Score: 300


Image ID: 887818
Group ID: 3484
Similarity Score: 307


Image ID: 676999
Group ID: 3282
Similarity Score: 310

## model4


Image ID: 870460
Group ID: 3282
Similarity Score: 0


Image ID: 662447
Group ID: 3398
Similarity Score: 200


Image ID: 772647
Group ID: 3282
Similarity Score: 214


Image ID: 772648
Group ID: 3282
Similarity Score: 234


Image ID: 549229
Group ID: 3398
Similarity Score: 236


Image ID: 693154
Group ID: 3282
Similarity Score: 236


Image ID: 83815
Group ID: 3282
Similarity Score: 239


Image ID: 870440
Group ID: 3282
Similarity Score: 239


Image ID: 83814
Group ID: 3282
Similarity Score: 243


Image ID: 83816
Group ID: 3282
Similarity Score: 249

# Deployment

# The Search Algorithm

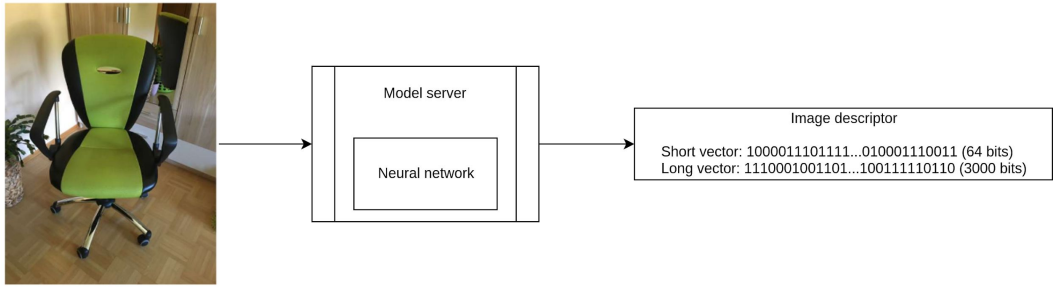
- Inputs
  - Source image descriptor
  - Number of desired results
- First phase of the search
  - Calculate short vector Hamming distance for all images
  - Sort results
  - Take first 256 * <number of desired results> images
- Second phase of the search
  - Calculate long vector Hamming distance for first phase results
  - Sort results
  - Take first <number of desired results> images

# Two types of search

- Search by ID
  - Use one of the existing images as a source image for search
  - Search input is its pre-calculated descriptor
- Search by image
  - Upload custom image for visual search (used by FashionCam)
  - Image Search Runtime first sends the image to Model Server to get its descriptor
  - That descriptor is used for search input

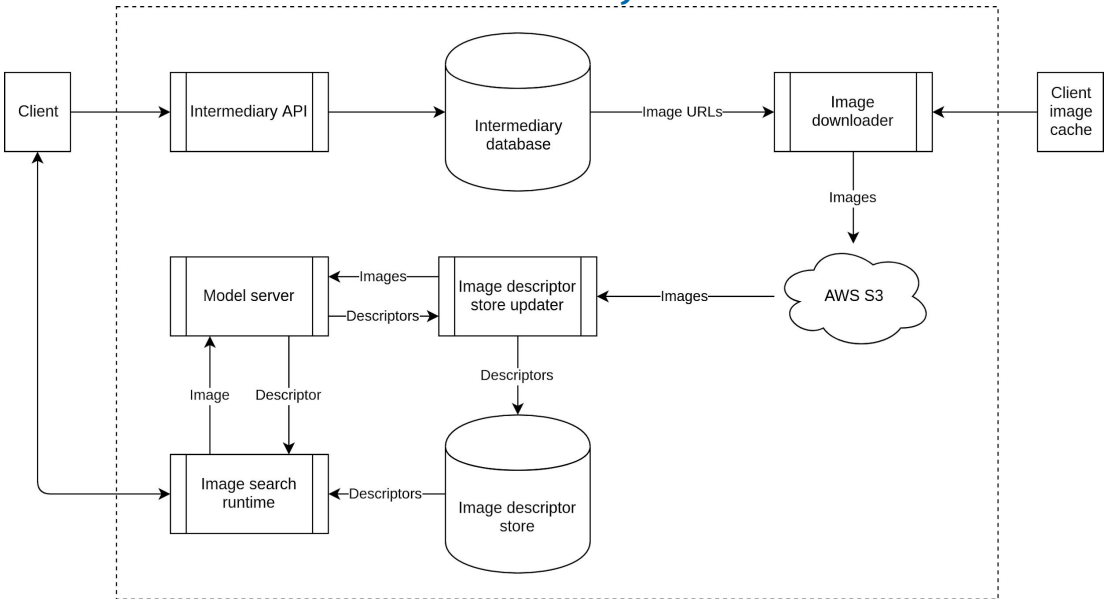
# TensorFlow Model Server

- TensorFlow Serving component
- Exposes a trained model through API
- Converts image into descriptor
- Runs on GPU



# Image Search

- Runtime loads all image descriptors into memory
  - Around 2GB per million descriptors
  - Stays in sync with Image Descriptor Store
- Search is executed on dense in-memory structures



# Deployment

- Entire system deployed on AWS
- Over 50M images on S3
- EC2 instances
  - Dockerized components
- Model server on GPU EC2 instance
  - GPU can process batch of requests in parallel
  - Gets more efficient with more concurrent requests
  - Up to 480 req/s with batch size of 50
- Image search runtime:
  - Single instance: 40 req/s with 50 ms avg response time
  - Scales up to required capacity

# Technology

- TensorFlow
  - Machine learning framework by Google
  - Open source
  - Model training decoupled from production
  - C++ API for top performance in production
  - Multi-platform support
  - Tensorflow Serving - model as a service
- NumPy
  - Used for dense in-memory data structures
- Cython
  - Performance-critical parts of code written in C
  - Links C functions with Python

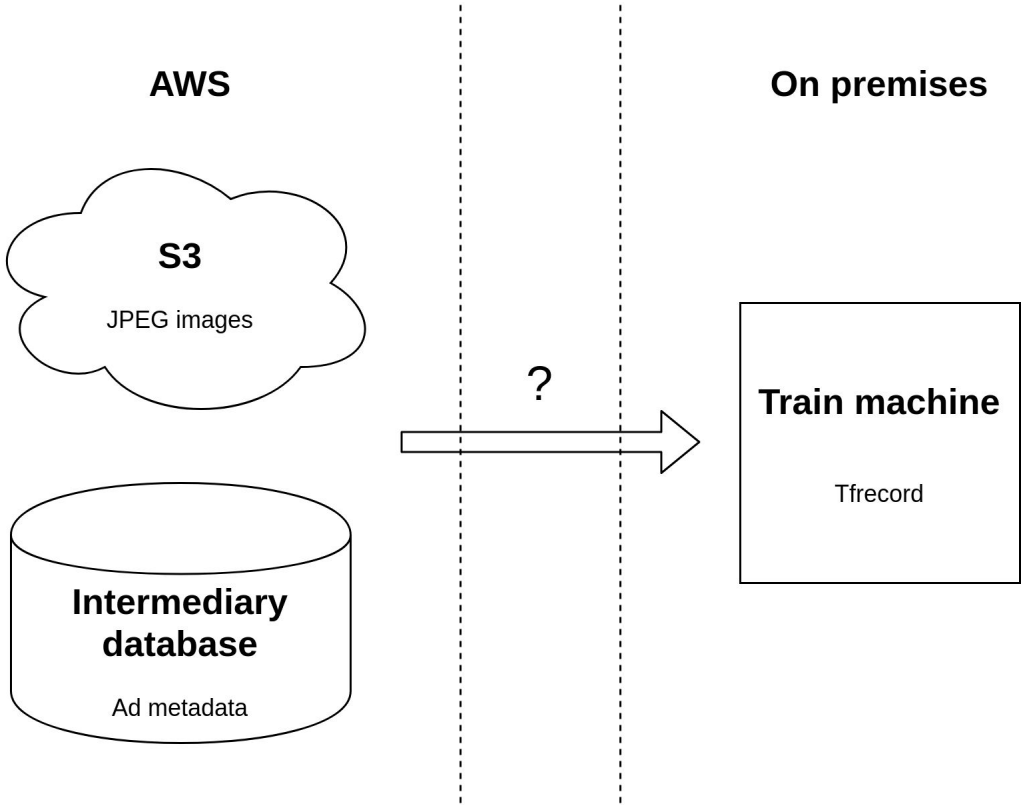




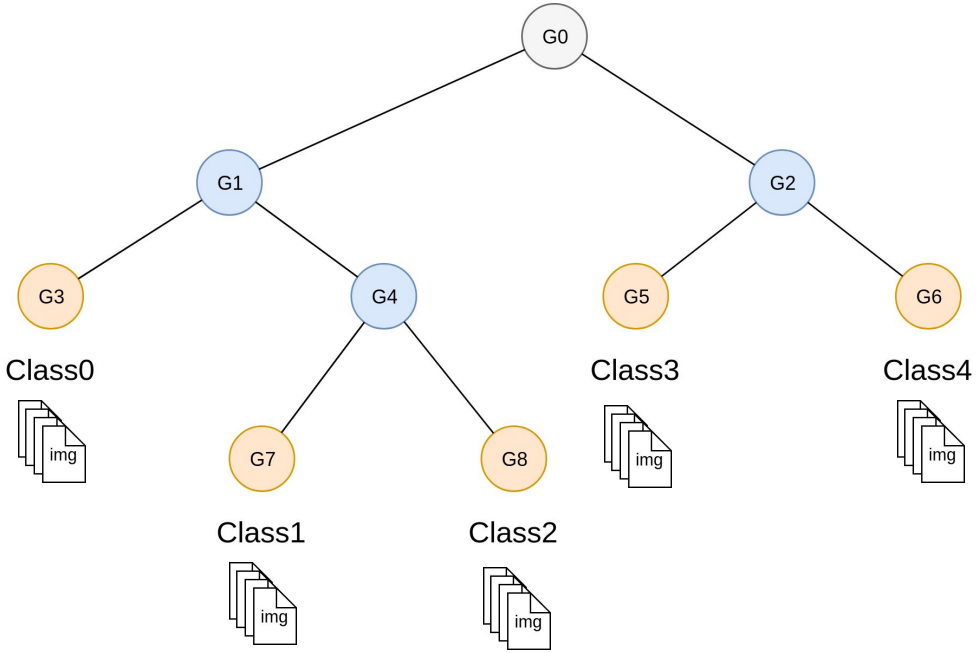
# Dataset preparation

# Problem

- Source
  - Images on AWS S3
  - Metadata in relational database
- Target
  - On-prem train machines
  - Multiple tfrecords
    - train
    - validation
    - ...

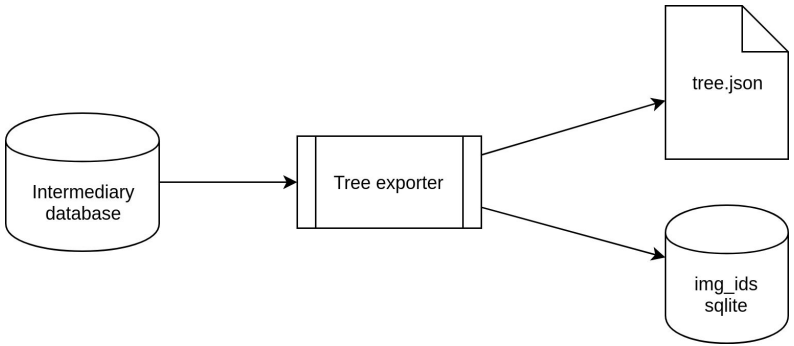


# Data structure



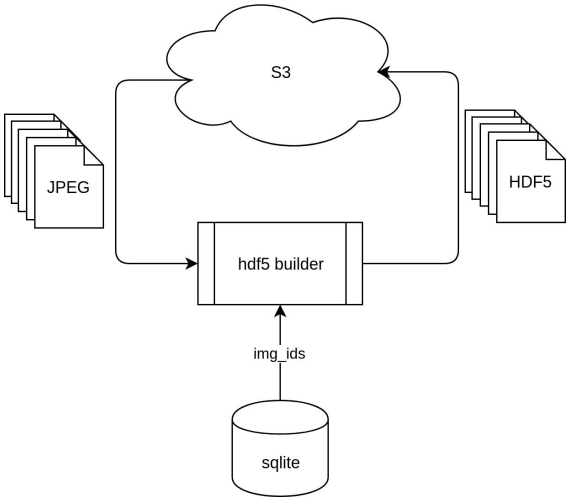
# Tree exporter

- On premises
- Load category tree and image ids from DB
- Perform tree operations (merge nodes, blacklist nodes, …)
- Perform sharding (train, validation, …)
- Export final category tree to JSON
- Export image ids for each leaf to sqlite



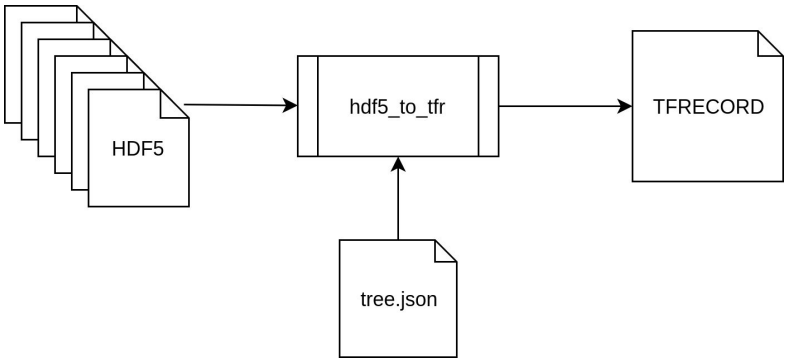
# Hdf5 builder

- On AWS
- Store multiple images in one file
- Conversion source
  - Individual JPEG images, 1024px larger side
  - Millions of small files (~1MB)
- Conversion target
  - One HDF5 per group per shard
  - Preprocessed images (scaled down)
  - Hundreds of large files (~1GB)

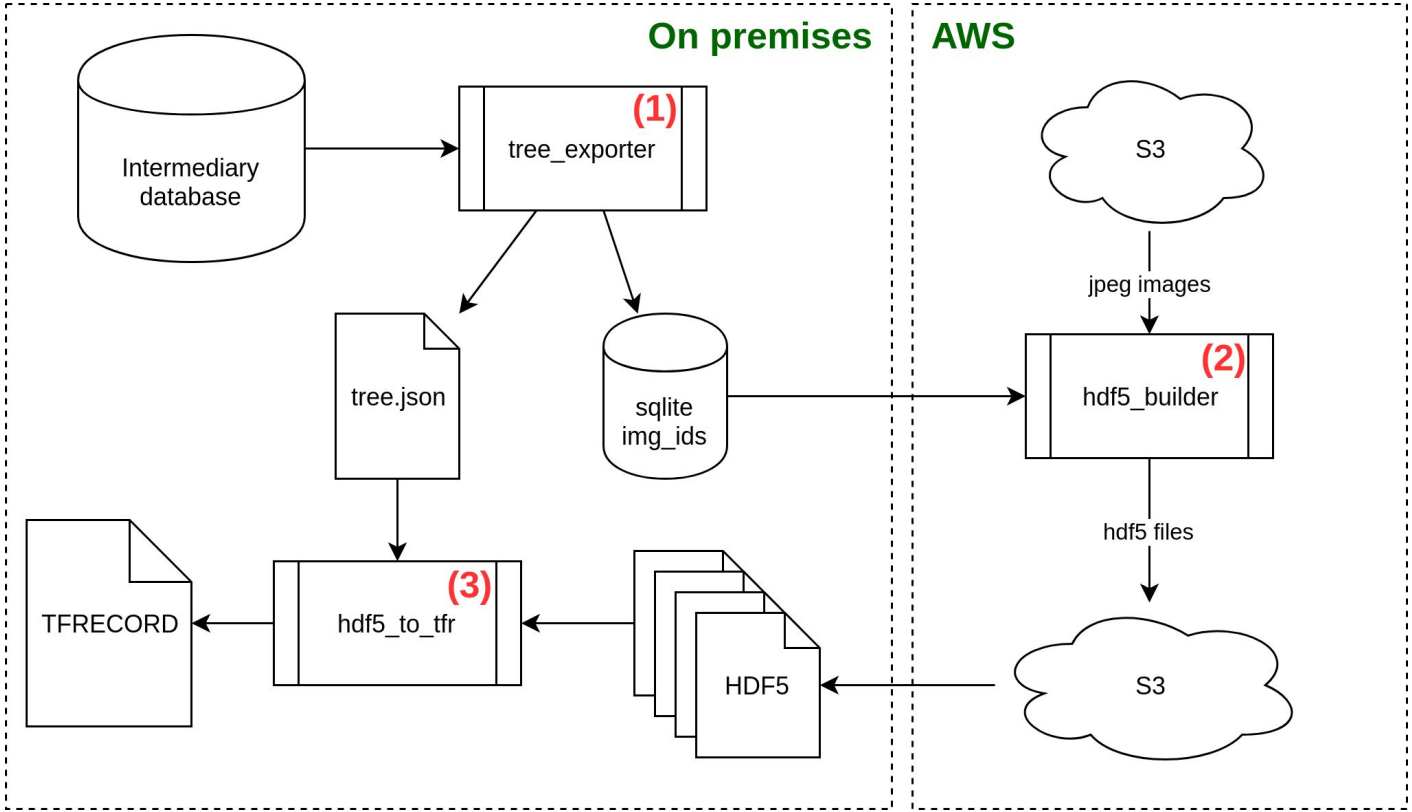


# Hdf5 to tfrecord

- On premises
- Read all HDF5 files simultaneously
- Ensure equal distribution of each group
- Store all images with metadata (class index, group_id, …) to tfrecord
- One tfrecord per shard



# Complete pipeline overview



# Training pipeline

# Goals

- Utilize GPU resources (4x) as efficiently as possible to increase training speed
- Perform online image augmentations to train more robust models
- Support general input
  - Single image
  - Multiple images
  - Image + text
- Support general input preprocessing

# Challenges

- Compute-heavy augmentations are performed on the CPU
- Augmentations are performed one image at a time
- While copying batches to GPU devices, they remain idle
- During a training step, subsequent batches need to be prepared

# Solutions

1. Our custom data input pipeline & training module
    - Queue runners -> Dataset API (cache, prefetch)
    - No StagingArea, GPU idle time
2. Tf_cnn_benchmarks-like pipeline and training
    - StagingArea, no GPU idle time
    - NCCL all-reduce
    - Heavy coupling, code quality issues
3. Horovod
    - Simple wrapper for tf.train.Optimizer
    - NCCL all-reduce
    - No StagingArea

# Nvidia GTC award

- NVIDIA GTC Conference 2017 (Munich)
- **Best poster award** & NVIDIA inception program access
- https://blogs.nvidia.com/blog/2018/01/11/ai-to-help-you-buy/

# Questions?

# Apache Lucene™ based visual similarity search

# Objectives

- **Unification** of visual similarity search with semantically dissimilar search types, e.g. geospatial and textual
- **Distributed** indexing and millisecond query execution
- **Fault recovery** & partition tolerance
- **Compatible** with Elasticsearch™ and Apache Solr™
- **Ease of integrating** into existing production search workloads



# Search re-implemented

STYRIA DATA SCIENCE

- Making two-phase search optional
  - Approximation phase matches a subset of images using coarse-grained binary feature vectors,
  - validation phase responsible for sorting the hit images in ascending similarity score order using fine-grained binary feature vectors.
- Weighting fine-grained binary feature vector components
  - E.g. higher weights assigned to the semantic component then colour to images of category X
- Millisecond execution time of two-phase search
  - $99^{th}$ percentile execution time 86ms using a single EC2 m5.xlarge instance on ~20.000.000 images