FASHION AND APPAREL CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS

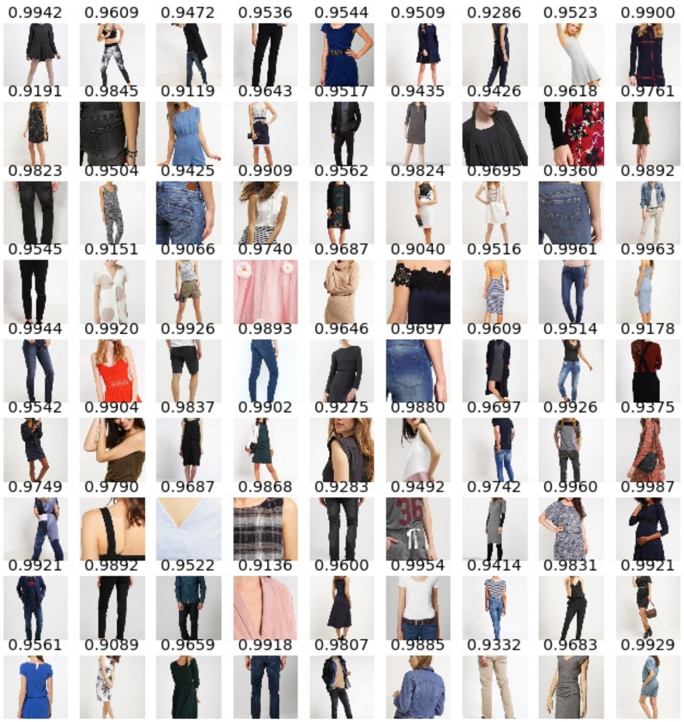
Alexander Schindler
alexander.schindler@ait.ac.at
Thomas Lidy
lidy@ifs.tuwien.ac.at
Stephan Karner
stephan.karner@monstyle.io
Matthias Hecker
matthias.hecker@monstyle.io

FASHION IMAGE CLASSIFICATION

- Online e-commerce access to product images
  - Asos-EU, Farfetch, Zalando
    - Images & metadat
- **Problem**
  - Metadata differs in
    - Quality, granularity, taxonomy
    - Taxonomy varies in depth of categorical hirarchy
- **Task**
  - use CNNs to
    - Consolidate Metadta
    - Enrich Metadata

BRIEF OVERVIEW

- Empirical study
- Applying deep Convolutional Neural Networks to fashion classification
- Evaluated five CNN architectures
- Custom and pre-trained models
- Evaluated on three tasks
    - Person detection
    - Product classification
    - Gender prediction

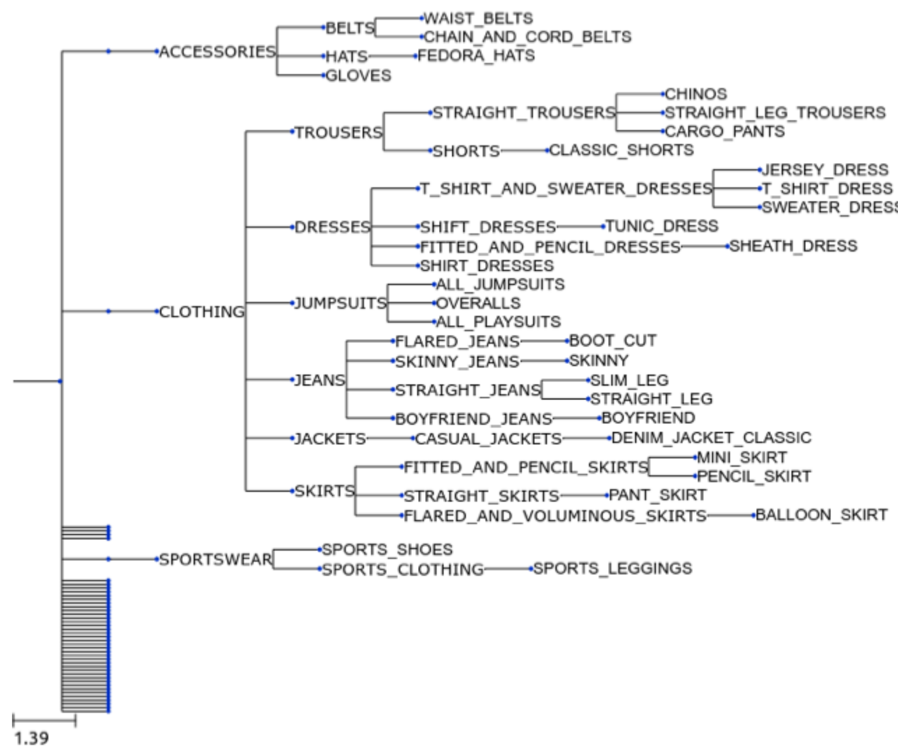


DATASETS

- **Person**
  - 7.833 images
    - 5.669 labeled as persons
    - 2.164 labeled as products
- **Products**
  - 234.884 images
  - 39.474 products
    - ~5,95 images per product
  - Ground-truth labels assignements
    - Product category
      - Label hirarchy
    - Gender
    - Age

Person

Product

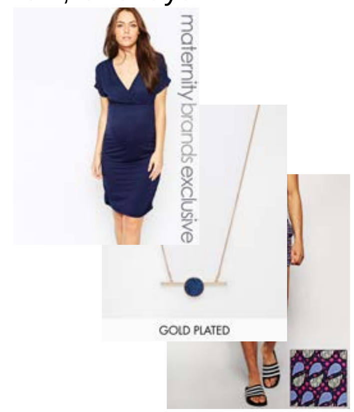


DATA QUALITY / ISSUES

White background

Worn by persons

Text, Overlays



Close-up texture

Close-up fit

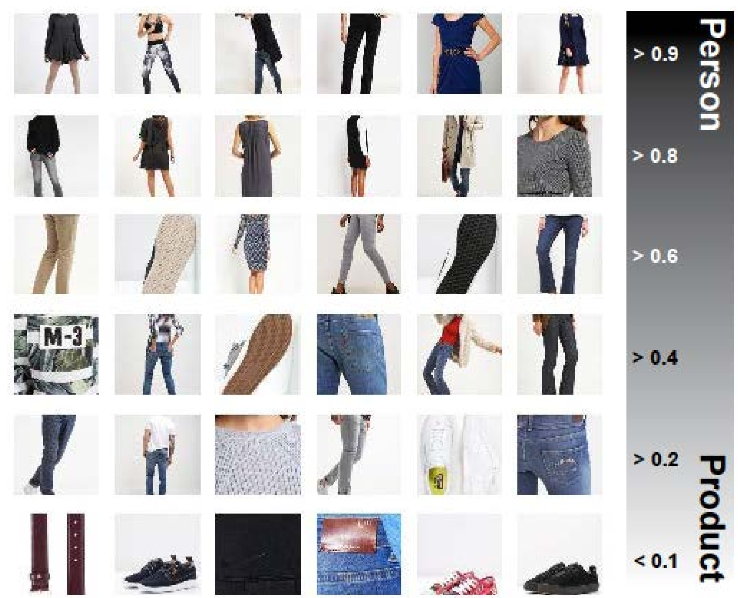
Multiple objects

Brand logo

Misc

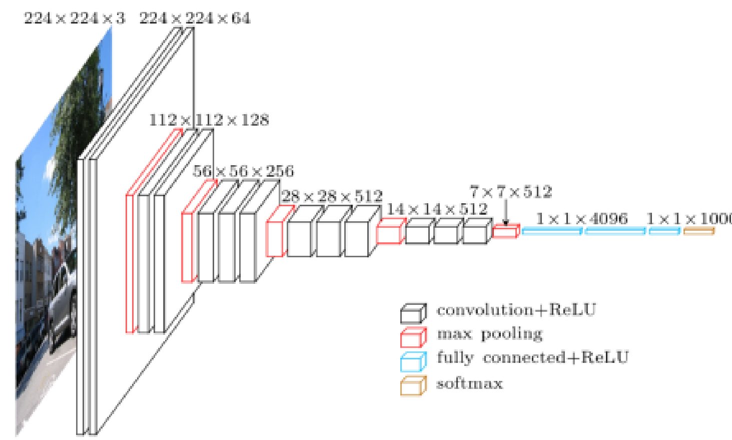
PERSON DETECTION

- **Products also presented by persons**
  - How they look when worn?
- **Problem**
  - Person wears multiple products
    - Single-label classification
    - Decission problem
- **Approach**
  - Train model to identify persons
  - Use model to filter images with persons
  - VGG-like custom model
- **Results**
  - **91.07%** accuracy on persons dataset

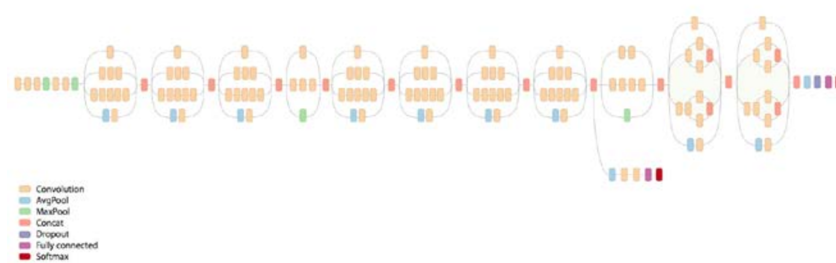


PRODUCT CLASSIFICATION

- **Deep Neural Network Architectures**
  - Vgg16 and Vgg19
  - InceptionV3
  - Custom CNN and Vgg-like
- **Experiments**
  - From-Scratch
  - Pre-Trained
- **Evaluation**
  - 4-Fold Crossvalidation
  - Grouped Stratification
- **Metrics**
  - Raw Accuracy
  - Max of Sum per product



David Fossard, https://www.cs.toronto.edu/~frossard/post/vgg16/



John Shlens, https://research.googleblog.com/2016/03/train-your-own-image-classifier-with.html

EXPERIMENTAL SET-UP

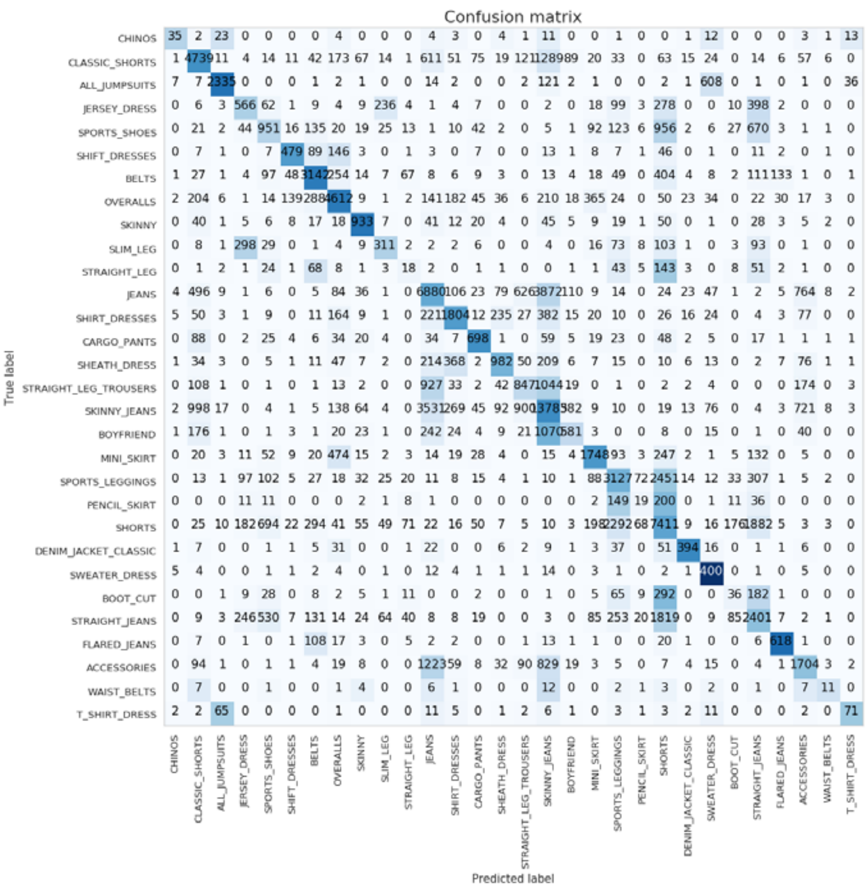
- **Small scale**
  - Subset of 23.305 images
- **Large scale**
  - 234.408 images
- **All Models**
  - Data Augmentation
    - 25% vertically and horizontally shifting
    - 25% zoom range
    - Horizontal flipping

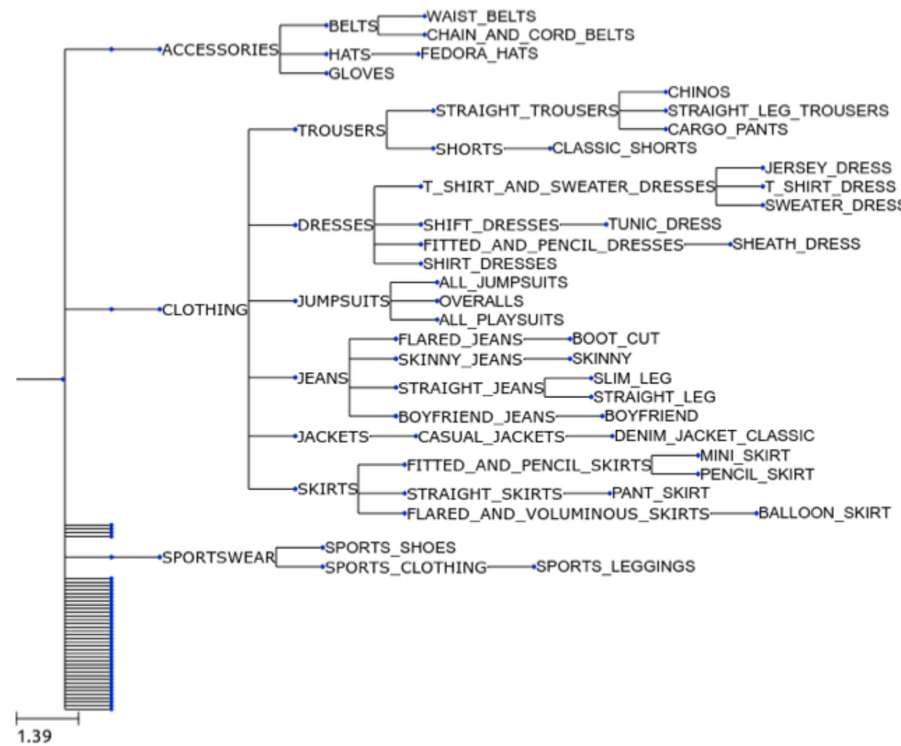
RESULTS – SMALL SCALE (24K)

- **Best results:** Pre-trained + fine-tune entire model
  - Freezing network + training only top layers not as good
- **Person filter** did not improve performance
- Small custom models have advantage of speed, but not as accurate

Description	best fold	best fold cum max	Mean cum max
InceptionV3, pretrained, fine-tuned	0.706	0.794	0.791
InceptionV3, pretrained, fine-tuned	0.658	0.729	0.716
VGG16, pretrained, fine-tuned	0.646	0.711	0.691
InceptionV3, pretrained, fine-tuned, person filter model as layer	0.569	0.685	0.658
VGG19, pretrained, fine-tuned	0.579	0.673	0.634
InceptionV3, pretrained, fine-tuned, no augmentation	0.564	0.673	0.647
VGG19, pretrained, train only top-layers	0.578	0.669	0.343
VGG16, pretrained, train only top-layers	0.603	0.652	0.368
InceptionV3, pretrained, train only top-layers	0.585	0.650	0.643
InceptionV3, pretrained, fine-tuned - person filtered metadata	0.640	0.636	0.614
InceptionV3, clean	0.492	0.594	0.580
Custom CNN, augmentation	0.506	0.568	0.538
Custom CNN	0.463	0.556	0.523
Custom VGG-like	0.438	0.549	0.519
VGG16, clean	0.439	0.455	0.443
VGG19, clean	0.437	0.447	0.430
VGG19, pretrained, train only top-layers	0.819	0.887	0.880
InceptionV3, pretrained, fine-tuned	0.798	0.863	0.836
VGG19, pretrained, fine-tuned	0.762	0.846	0.830

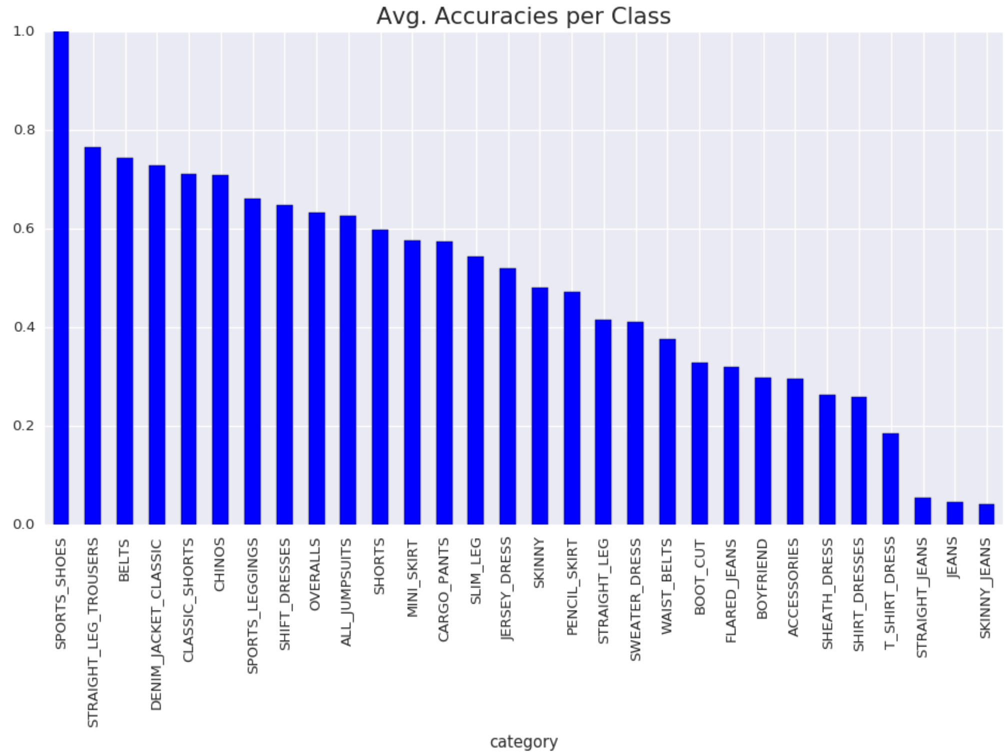
CONFUSIONS – SMALL SCALE (24K)





PER CLASS ACCURACIES – LARGE SCALE (234K)

- Problem of different granularity of provided ground truth
- Parent/Child nodes used interchangeably
  - Misclassification of child as parent is not wrong
  - Model does not consider hirarchy



CONCLUSIONS

- Despite large dataset and reduced number of classes
  - **Pretrained models outperform** from-scratch training
    - Product classification – 79.1%
    - Gender prediction – 88.0%

- **Custom small model** enough to learn **binary task**
  - person/product classification – 91.07%

- **Preprocessing** of **ground-truth** required
  - Flatten hierarchy, remove ambiguities and overlaps
  - Use **hierarchical CNNs**
  - Use **attention** (person images)

THANK YOU!

Alexander Schindler, 08.06.2018

Machine Learning Showcase

Mon Style GmbH

https://monstyle.io
https://asksophie.io

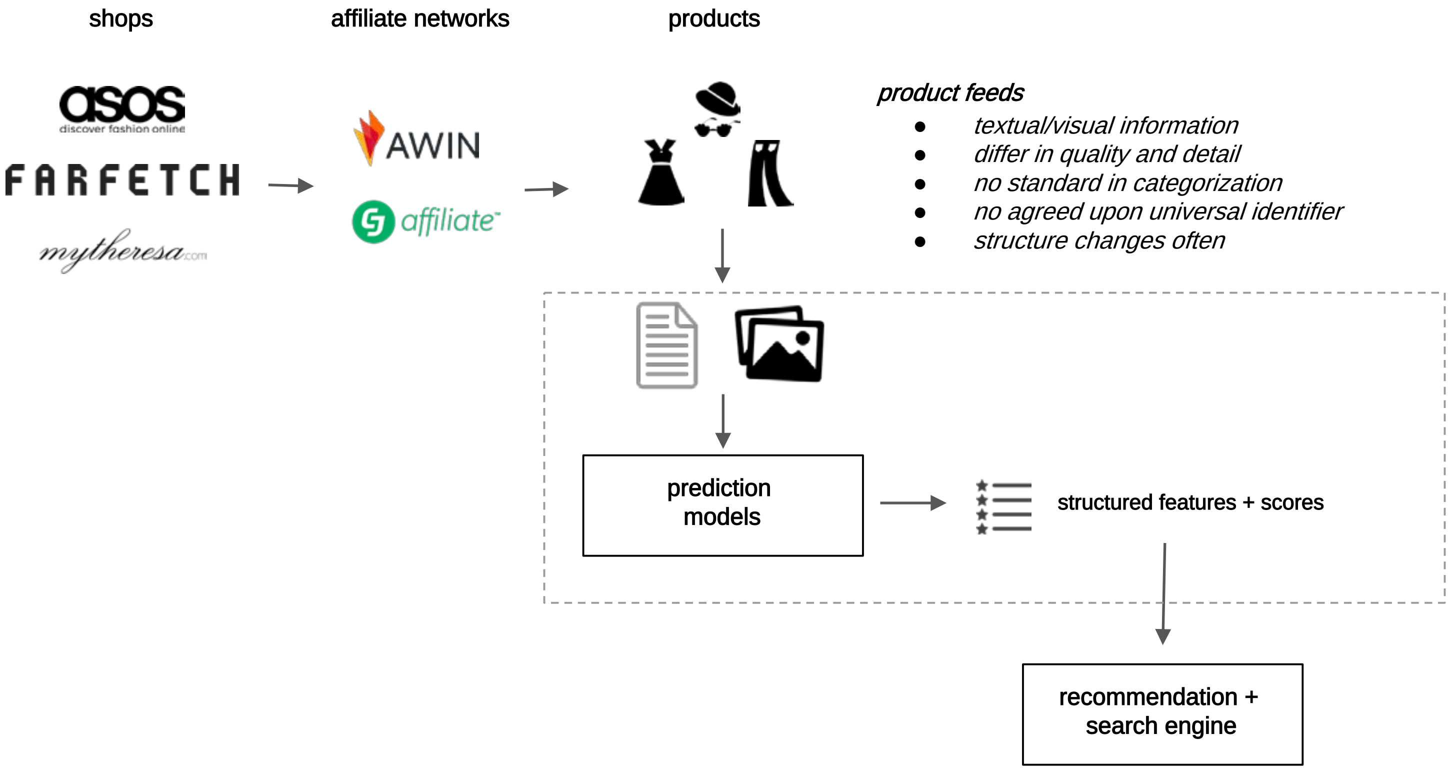
Presentation Agenda

1. Use Case / Problem Description
   (what are we talking about)

1. Prediction Pipeline
   (product data to curated catalog)

1. Datasets with Mechanical Turk
   (example dataset created with mturk)

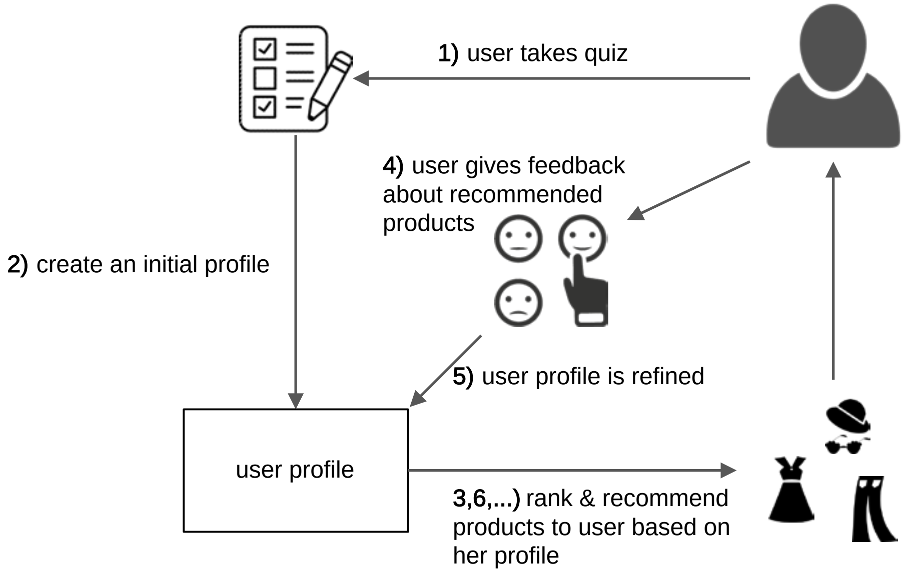
1. Affiliate Overview



1. Recommendation

recommendation + search engine → Customer-facing Platforms

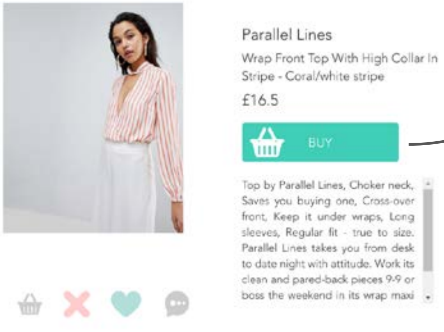
content-based recommendation engine



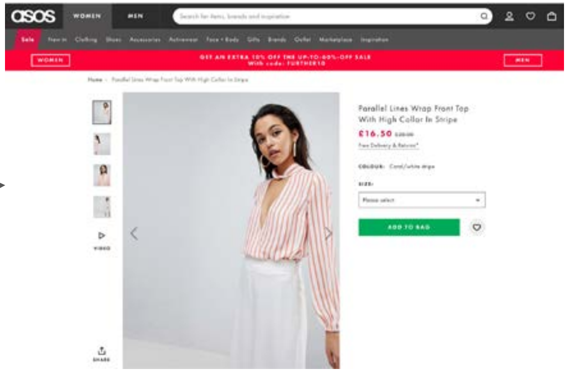
1. Website

recommendation + search engine → Customer-facing Platforms

web shopping platform & community



↴ *monstyle.io*

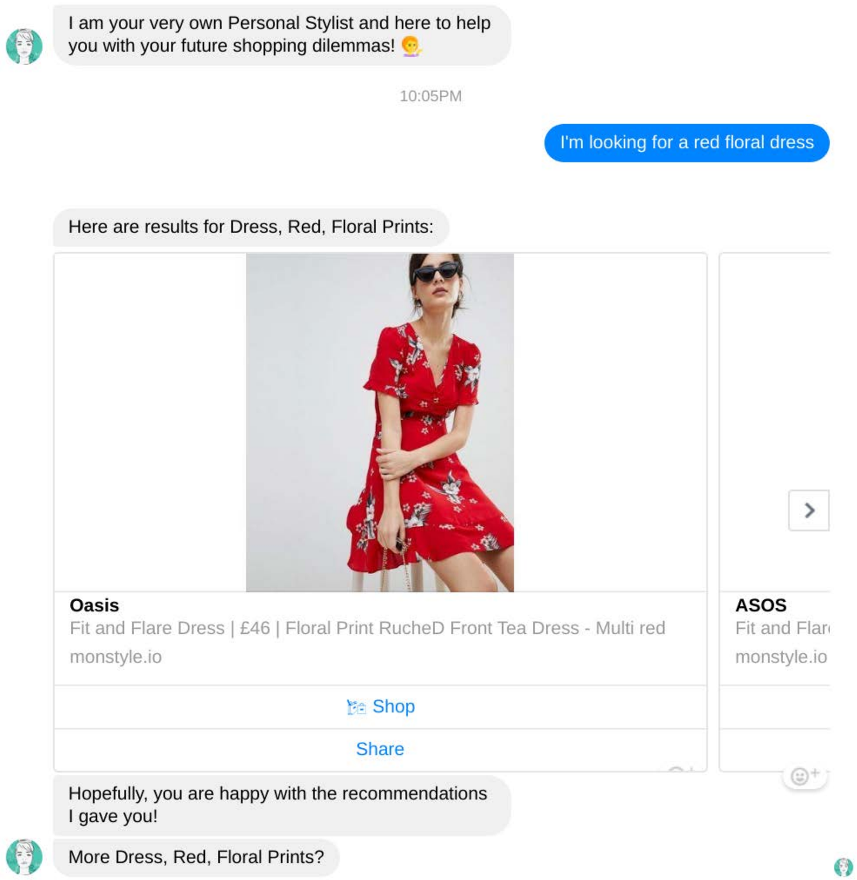


↴ *asos.com*

1. Chatbot

recommendation + search engine → Customer-facing Platforms

facebook messenger chatbot



↴ *messenger.com, @SophieStyleBot*

1. Example Shop Data

data feeds provided as XML/CSV files:

asos.com

```
<name>sleeve shirt - white</name>
<price>12.50</price>
<category>t-shirt</category>
<image>http://…</image>
```

farfetch.com

```
<title>sleeve shirt</title>
<buy_at>25.50</buy_at>
<category>shirts</category>
<color>white</color>
<thumbnail>http://…</thumbnail>
```

normalize,
deduplicate,
classify

monstyle

```
title: sleeve t-shirt
color: white
```

variants

monstyle

```
shop: farfetch
price: 25.50
```

monstyle

```
shop: asos
price: 12.50
```

1. Example Problem: Brand Names

some brand names include category descriptions, for instance:

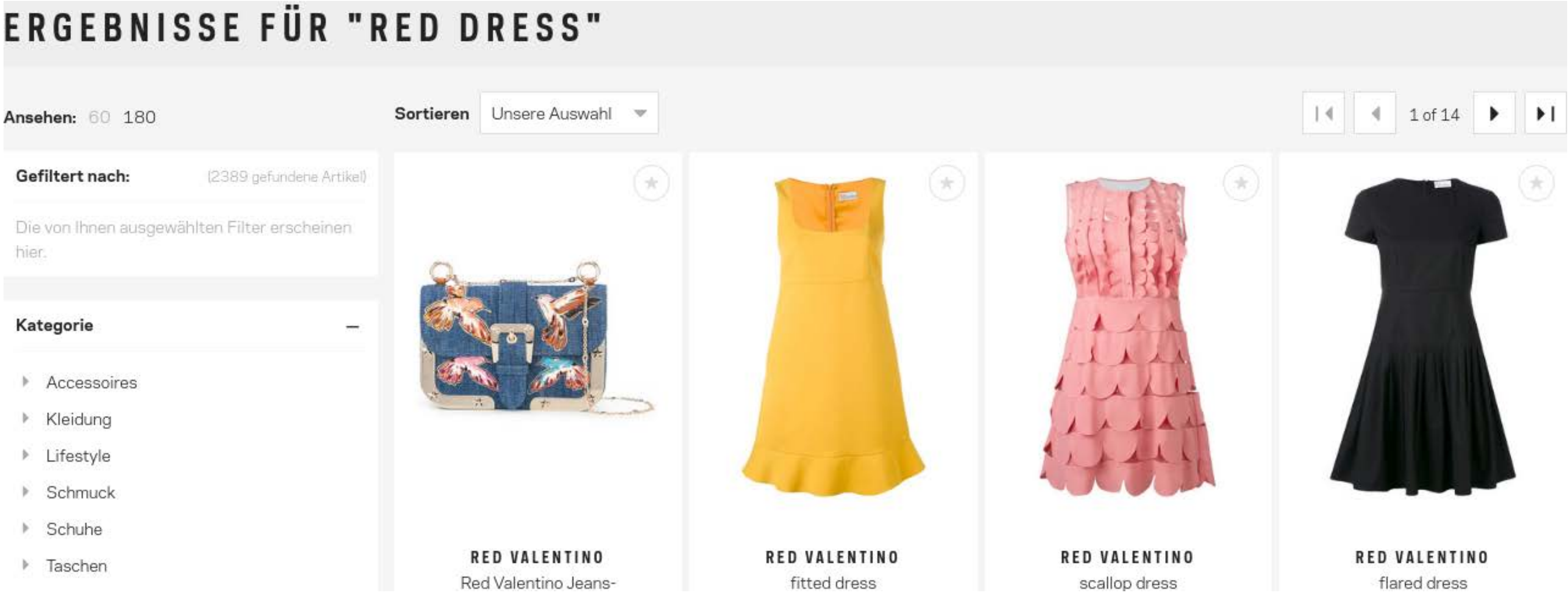
Product Title:
Versace **Jeans** Foil Logo Tank Dress with Cutout Back

-> generate list of brand names
-> use list to remove brands from product title

Product Title: Foil Logo Tank Dress with Cutout Back
Product Brand: Versace Jeans

-> free text search engine: ranking by field, rank title matches higher than brand

1. Example Problem: Brand Names



1. Tags/Keyword Vocabulary

different terminology used around shops, needs normalization:
-> extensive list of synonyms/aliases in order to normalize data
-> keep synonyms for user search

For Example: Lingerie and Nightwear » Sleepwear » Robe

kimono robe, robe, night kimono, nightgown, dressing gown, night robe, peignoir, sleeping gown, bath robe, bathing robe

For Example: Floral Print

floral, leaf, floral print, leaf print

1. Product Catalog

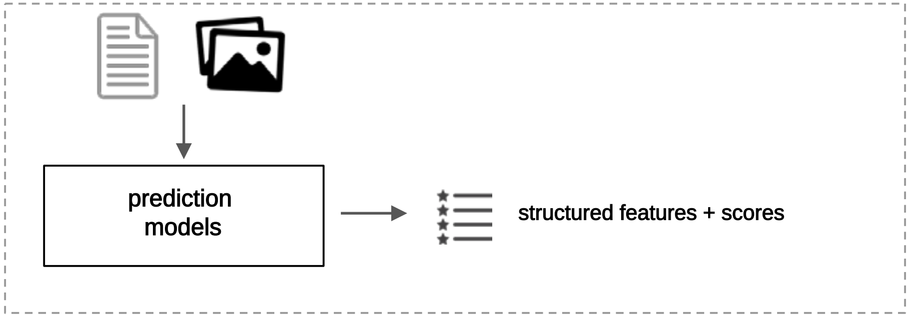
for example:

id there exists no universal standard, ASIN, EAN, UPC, etc.
title normalized version can be generated from brand, categories and other attributes
brand brands are an important indicator for style, size/fit-differences, etc.
price including currency
availability to not show unavailable products
category hierarchical, fashion-domain specific
color normalized color labels
pattern floral print, …
material leather, fur, cotton, vegan?
attributes some attributes are category-specific, such as collar type or skirt length

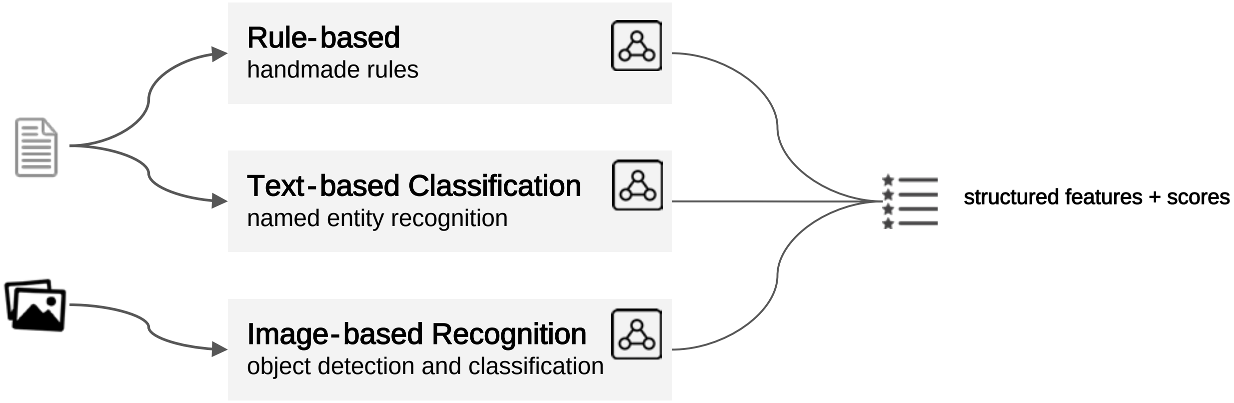
1. Product Catalog Example

id	2791782 -7049-4784-be6b-a2eb1865994e
title	leaf-print flared midi dress
brand	OSCAR DE LA RENTA
price	2.621 €
availability	available
category	clothing » dress » fit and flare dress
color	white, blue » navy
pattern	leaf print
material	cotton
attributes	sleeveless, square angular neckline, kneelength, ...

2. Prediction Pipeline



2. Prediction Models



2. Example Input

For Example

```
<name>leaf-print flared midi dress</name>
<buy_price>2621</buy_price>
<gender>female</gender>
<description>Navy blue and white cotton-blend
leaf-print flared midi dress from Oscar de la
Renta.</description>
```

Rule-based
handmade rules

Text-based Classification
named entity recognition

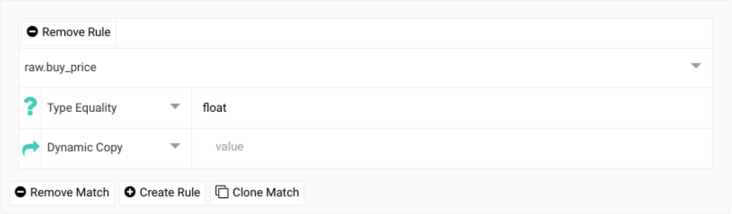
Image-based Recognition
object detection and classification

2. Rule Model

Rule-based Recognition

```
<name>leaf-print flared midi dress</name>
<buy_price>2621</buy_price>
<gender>female</gender>
<description>Navy blue and white cotton-blend
leaf-print flared midi dress from Oscar de la
Renta.</description>
```

Rule-based
handmade rules



Ruleset Management UI

```
return raw.buy_price
```

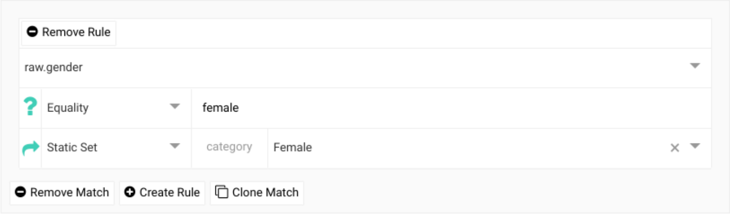
Price: 2,621.00
→ 100.0

2. Rule Model

Rule-based Recognition

```
<name>leaf-print flared midi dress</name>
<buy_price>2621</buy_price>
<gender>female</gender>
<description>Navy blue and white cotton-blend
leaf-print flared midi dress from Oscar de la
Renta.</description>
```

Rule-based
handmade rules



Ruleset Management UI

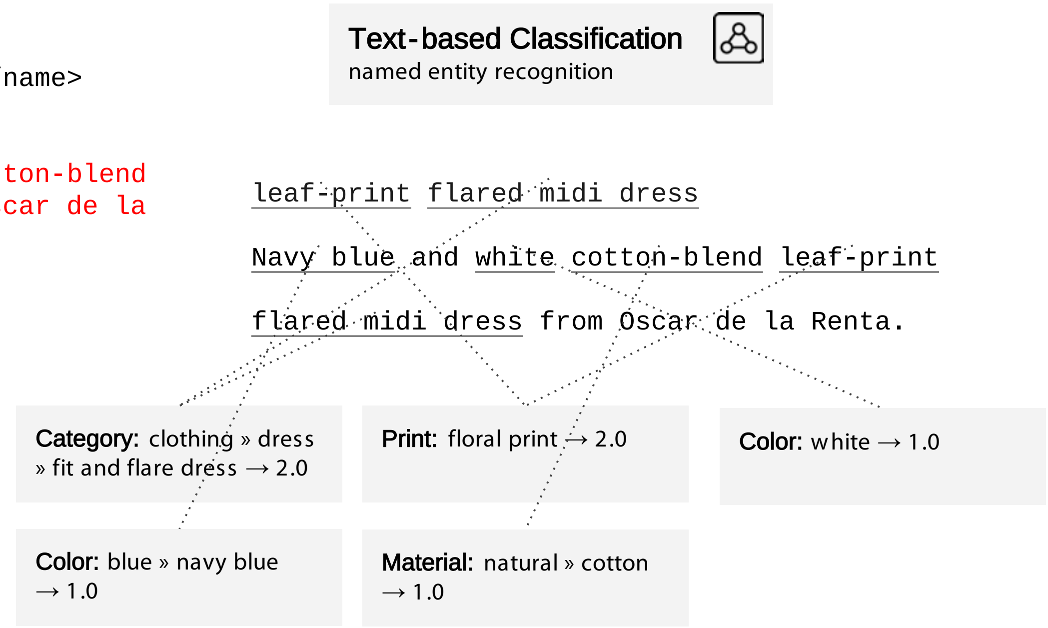
```
if raw.gender=='female':
    return GENDER.FEMALE
```

Gender: female
→ 100.0

2. NLP Model

Named Entity Recognition

```
<name>leaf-print flared midi dress</name>
<buy_price>2621</buy_price>
<gender>female</gender>
<description>Navy blue and white cotton-blend
leaf-print flared midi dress from Oscar de la
Renta.</description>
```



2. Vision Model

Image Classification

```
<name>leaf-print flared midi dress</name>
<buy_price>2621</buy_price>
<gender>female</gender>
<description>Navy blue and white cotton-blend
leaf-print flared midi dress from Oscar de la
Renta.</description>
```

Image-based Recognition
object detection and classification

Gender: female
→ 0.9289139

Category: clothing » dress
→ 0.9762532

2. Feature Assembling

Assemble Final Predicted Features

Gender: female
→ 0.9289139

Category: clothing » dress
→ 0.9762532

Gender: female → 100.9289139		**Material:** natural » cotton → 1.0
Price: 2,621.00 → 100.0	**Category:** clothing » dress » fit and flare dress → 2.97	
Print: floral print → 2.0	**Color:** blue » navy blue → 1.0	**Color:** white → 1.0

2. Example: Incorrect Input

Wrong Data

title wrong

```
<name>leaf-print skirt</name>
<buy_price>2621</buy_price>
<gender>female</gender>
<description>Navy blue and white cotton-blend
leaf-print flared midi dress from Oscar de la
Renta.</description>
```

Rule-based
handmade rules

Text-based Classification
named entity recognition

Image-based Recognition
object detection and classification

2. Example: Incorrect Input

Named Entity Recognition

```
<name>leaf-print skirt</name>
<buy_price>2621</buy_price>
<gender>female</gender>
<description>Navy blue and white cotton-blend
leaf-print flared midi dress from Oscar de la
Renta.</description>
```

Text-based Classification
named entity recognition

leaf-print skirt

Navy blue and white cotton-blend leaf-print

flared midi dress from Oscar de la Renta.

Category: clothing » skirt → 1.0

Category: clothing » dress » fit and flare dress → 1.0

Print: floral print → 2.0

Color: white → 1.0

Color: blue » navy blue → 1.0

Material: natural » cotton → 1.0

2. Example: Incorrect Input

Resolve Conflicts

Category: clothing » skirt →1.0

Category: clothing » dress → 0.9762532

Price: 2,621.00 → 100.0

Material: natural » cotton → 1.0

Gender: female → 100.0

Category: clothing » dress » fit and flare dress → 1.97

Print: floral print → 2.0

Color: blue » navy blue → 1.0

Color: white → 1.0

2. Example: Incorrect Input

Assemble Final Predicted Features

Gender: female → 100.0

Material: natural » cotton → 1.0

Price: 2,621.00 → 100.0

Category: clothing » dress » fit and flare dress → 1.97

Print: floral print → 2.0

Color: blue » navy blue → 1.0

Color: white → 1.0

3. Data Labelling

Text and Image Annotation for Machine Learning Datasets

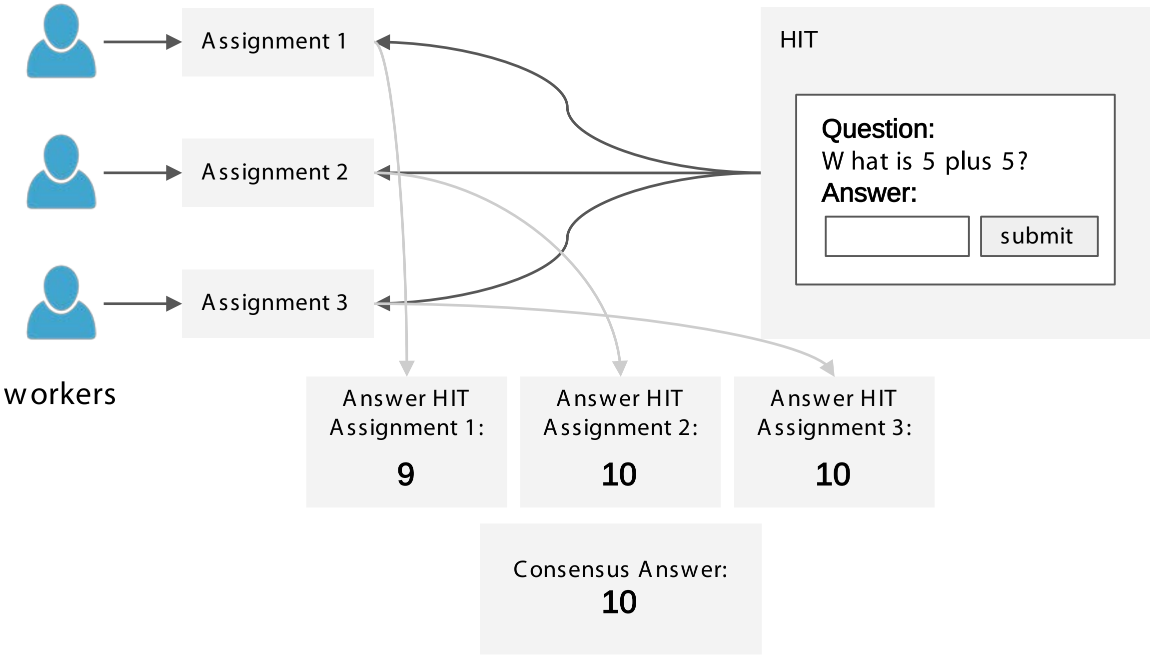
Tools and Services:

- Web Service: LabelBox (https://www.labelbox.io/)
- Web Service: Supervisely (https://supervise.ly/)
- Web Service: Sequence.work (https://sequence.work/)
- Mac Application: RectLabel (https://rectlabel.com/)
- **Mechanical Turk**

Excellent Resource:
https://en.wikipedia.org/wiki/List_of_manual_image_annotation_tools

3. Mechanical Turk (MTurk) - Overview

Assignments and Human Intelligence Tasks (HITs) - Consensus

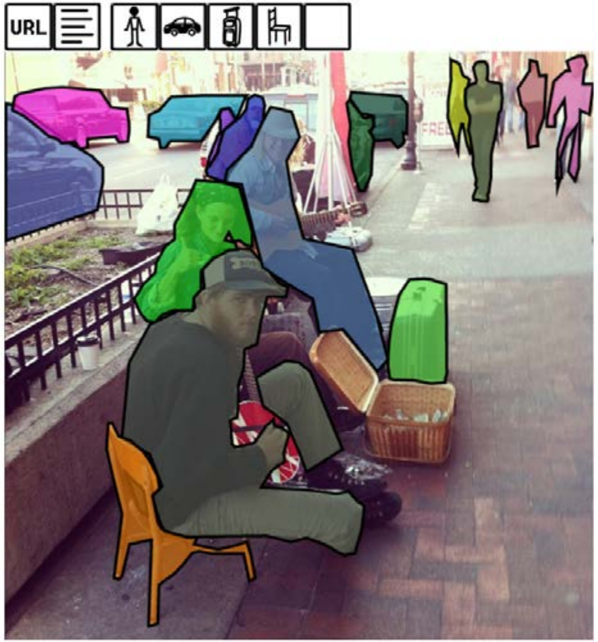


3. COCO Dataset

Example: COCO (Common Objects in Context)

A large-scale object detection, segmentation and captioning dataset with:

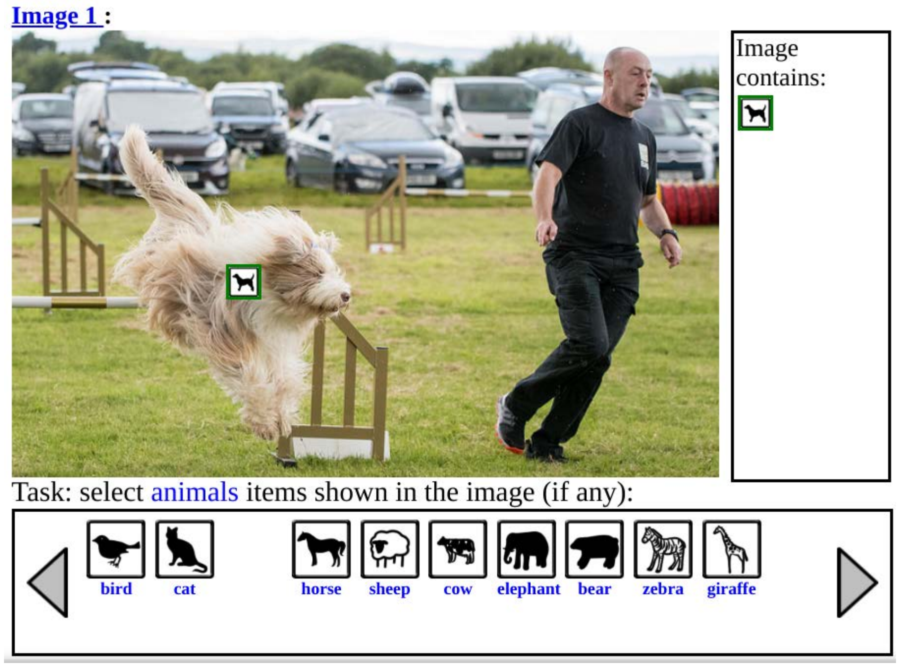
- 123,287 images
- 886,284 instances
- 91 object types
- created using mechanical turk



↴ *MS COCO 2017 Dataset Explorer, http://cocodataset.org/#explore*

3. COCO Dataset

Example: COCO (Common Objects in Context)



↴ MS COCO Annotation UI, https://github.com/tylin/coco-ui

Supported By

TECHNISCHE UNIVERSITÄT WIEN Vienna | Austria

wirtschafts agentur wien

Ein Fonds der Stadt Wien

FFG Österreichische Forschungsförderungsgesellschaft