Maximilian Witte

Image Data for Machine Learning Research in Marketing

EMAC Odense 2023

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Why Use Image Data with Machine Learning for Marketing Research?

1 More than 6.5 billion images shared every day[1]

2 More than 60% of US adults use instagram[2]

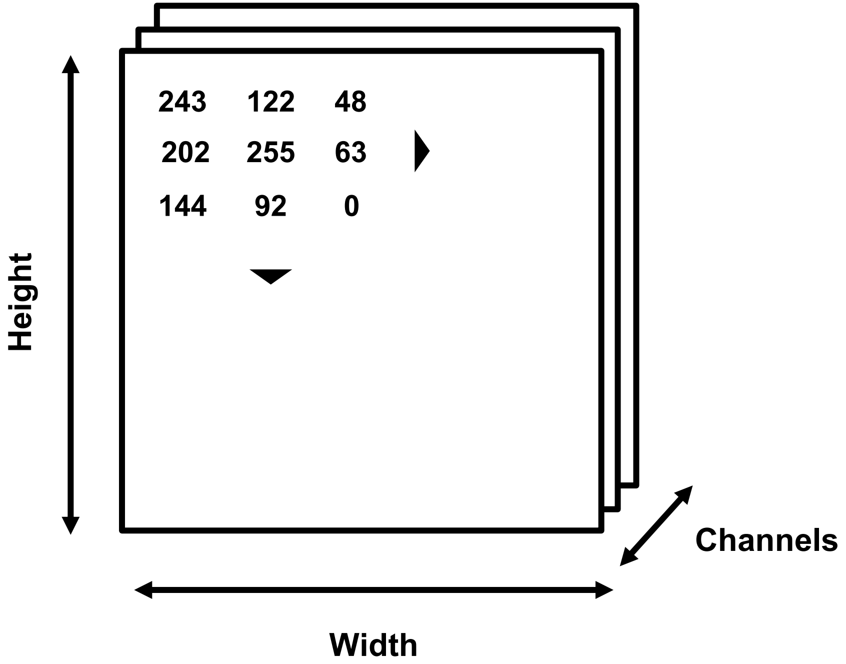
Automatically extract information for analysis

Predict behavior

Goal always generalization on unseen data

1. Hartmann, J., Heitmann, M., Schamp, C., & Netzer, O. (2021). The power of brand selfies. Journal of Marketing Research, 58(6), 1159-1177.
2. 31 Mind-Boggling Instagram Stats & Facts for 2022, https://www.wordstream.com/blog/ws/2017/04/20/instagram-statistics

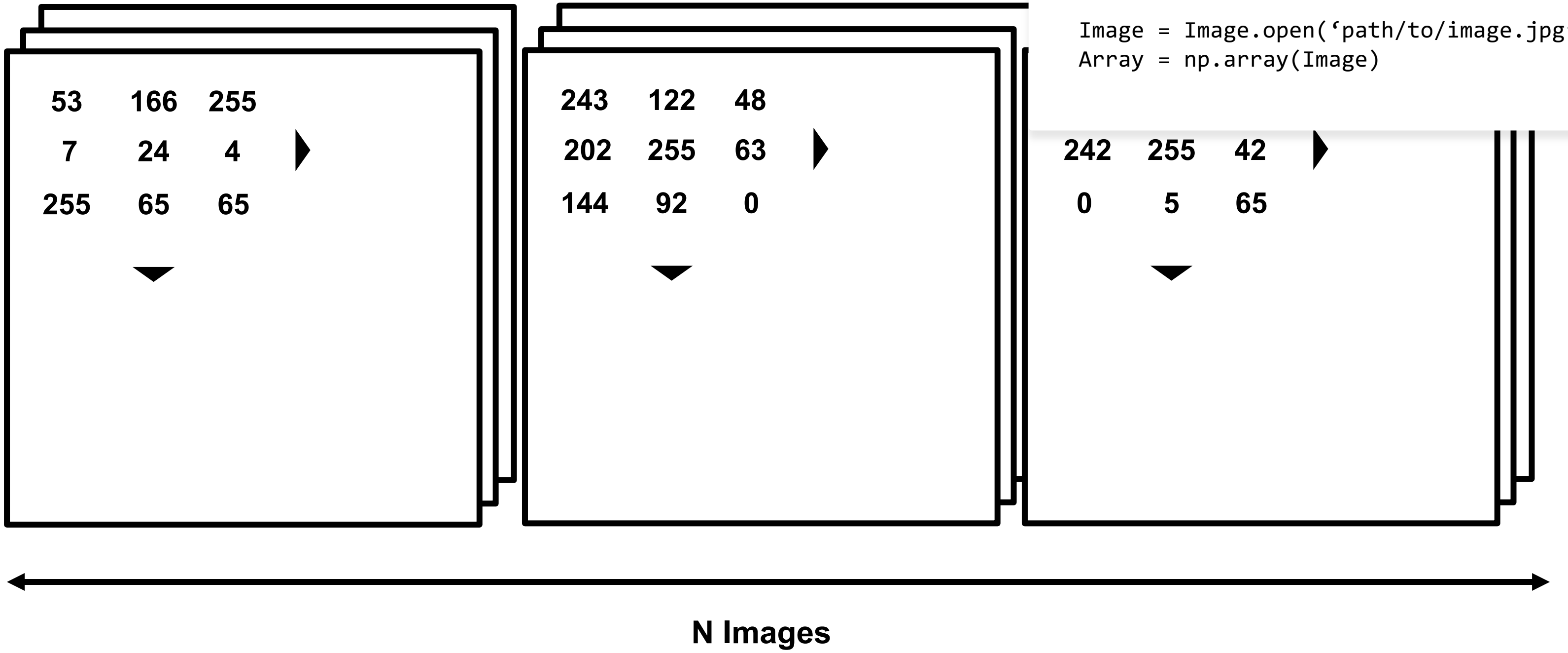
The 4 dimensions of image data



Source: Chollet, F. (2021). Deep learning with Python. Simon and Schuster. https://www.manning.com/books/deep-learning-with-python

The 4 dimensions of image data

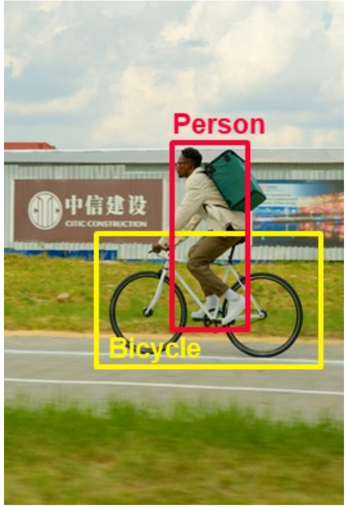
```python
from PIL import Image
import numpy as np

Image = Image.open('path/to/image.jpg')
Array = np.array(Image)
```



Source: Chollet, F. (2021). Deep learning with Python. Simon and Schuster. https://www.manning.com/books/deep-learning-with-python

A multitude of potential tasks for image data

Classification



Object Detection

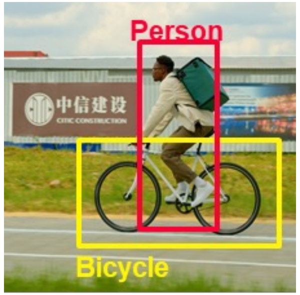
Segmentation

A multitude of potential tasks for image data

A Image Analysis

Classification

Object Detection



Segmentation

+ Depth estimation | Image-to-Text | …

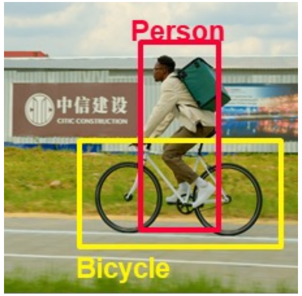
B Generative AI[1]

Text2Image

+ Img2Img

1 Fine-tuning generally unsupervised using generative AI.

Tools to annotate your data

	Classification, Generative AI	Object detection	Segmentation, depth detection
Task			
Format	Folders, Tables	JSON, XML	Image, JSON
Tools	1. Move images to folder 2. PhotoSift[1] 3. Use Python GUI	1. Labelme[2] 2. V7[3] 3. Labelbox[4]	1. Labelme[2] 2. V7[3] 3. Labelbox[4]

Python labeling notebook

https://github.com/Maximilianwte/Image-tutorial/blob/main/Image_labeling.ipynb

1 PhotoSift (freeware) available for windows: https://www.rlvision.com/photosift/about.php
2 Labelme (freeware) available for windows, mac and linux: https://github.com/wkentaro/labelme
3 V7 (free for education) available online: https://www.v7labs.com
4 Labelbox (freemium) available online: https://labelbox.com/

A Using Images for analysis

Table 3. Summary Statistics of Field Data.

	Twitter		Instagram		Display Ads	
Variable	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
Dependent Variables						
Likes	5.06	233.79	255.33	4,380.30	—	—
Comments	.43	7.00	9.85	65.53	—	—
Purchase Intentions	.02	.15	.51	2.72	—	—
Brand-Image Type						
Brand Selfie[a]	28.73	—	19.63	—	72.68	—
Consumer Selfie[a]	7.93	—	11.57	—	18.80	—
Packshot[a]	63.35	—	68.80	—	8.51	—
Image Characteristics						
Logo Size	.12	.14	.07	.10	.33	.21
Logo Centrality	.63	.23	.63	.19	.81	.14
Visual Complexity	−1.96	.43	−2.09	.55	−1.40	.94
Brightness	.51	.15	.56	.13	.60	.13
Brightness Contrast	.25	.05	.26	.05	.24	.05
Post Characteristics						
Number of Words	2.20	.55	3.24	.82	2.29	.22
Number of Hashtags	.39	.59	2.29	.92	—	—
Number of Handletags	.23	.39	.25	.47	—	—
Branded Caption[a]	6.23	—	12.10	—	—	—
Branded Tag[a]	13.02	—	61.65	—	—	—
Ad Tag[a]	.35	—	2.35	—	—	—
First-Person Pronoun[a]	35.66	—	28.97	—	.13	—
Second-Person Pronoun[a]	14.59	—	16.79	—	35.25	—
Question Word Share	.01	.04	.01	.03	.01	.03
Netspeak Word Share	.02	.06	.01	.03	.00	.01
Positive Sentiment[a]	42.43	—	50.84	—	6.12	—
Neutral Sentiment[a]	46.25	—	45.24	—	92.73	—
Negative Sentiment[a]	11.32	—	3.92	—	1.15	—
Post Age	5.04	.64	2.39	1.48	—	—

Source: Hartmann, J., Heitmann, M., Schamp, C., & Netzer, O. (2021). The power of brand selfies. Journal of Marketing Research, 58(6), 1159-1177.

Choosing a library for deep learning

TensorFlow

Developed by Google

Keras High-level API

Support for GPU's and TPU's

PyTorch

Developed by Meta

Pythonic syntax

Steep learning curve

HUGGING FACE

Large amount of pre-trained models

Easy-to-use

Support for PyTorch and TensorFlow

Preparing the data

The goal of ML is generalization → Simulate situation of having unseen data

1. **Load data**

```python
from datasets import load_dataset

dataset = load_dataset("imagefolder",
data_dir=”path/to/directory")
```

2. **Data Splitting**

- **Train-test-split**: Split off percentage of training data for testing
- **K-fold cross-validation**: Split data in k equal datasets. Iterate over all subsets using each as validation dataset
- **Bootstrapping**: Create artificial datasets by randomly sampling observations

```python
from sklearn.model_selection import
train_test_split

train_dataset, test_dataset =
train_test_split(dataset, test_size=0.2,
random_state=42)
```

3. **Feature extraction**

```python
from transformers import  ViTFeatureExtractor

feature_extractor = ViTFeatureExtractor.
 from_pretrained('google/vit-base')

def prepare(example_batch):
    inputs = feature_extractor([x for x in
    example_batch['image']],
    return_tensors='pt')
    inputs['labels'] = example_batch['label']
    return inputs

prepared_ds =   dataset.with_transform(prepare)
```

Increasing data variability by using Image Augmentations

Use augmentations to efficiently increase variability of the training data

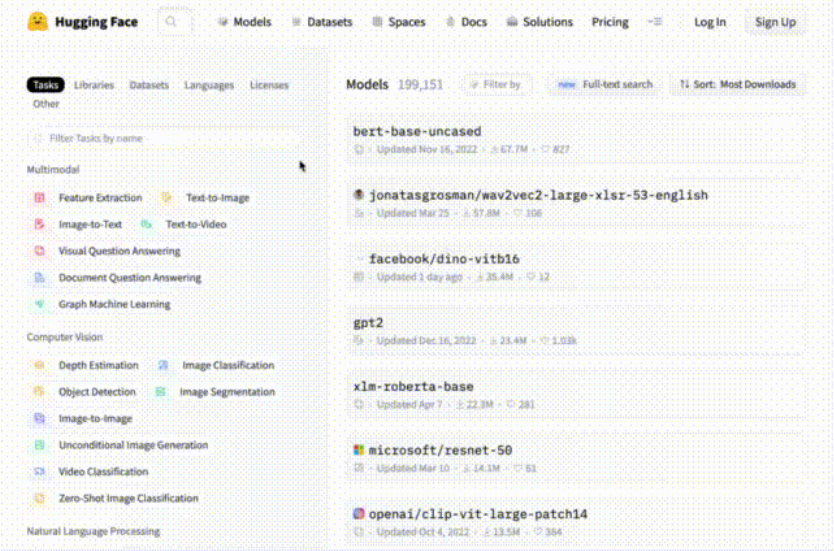
More variability in data leads to better generalization

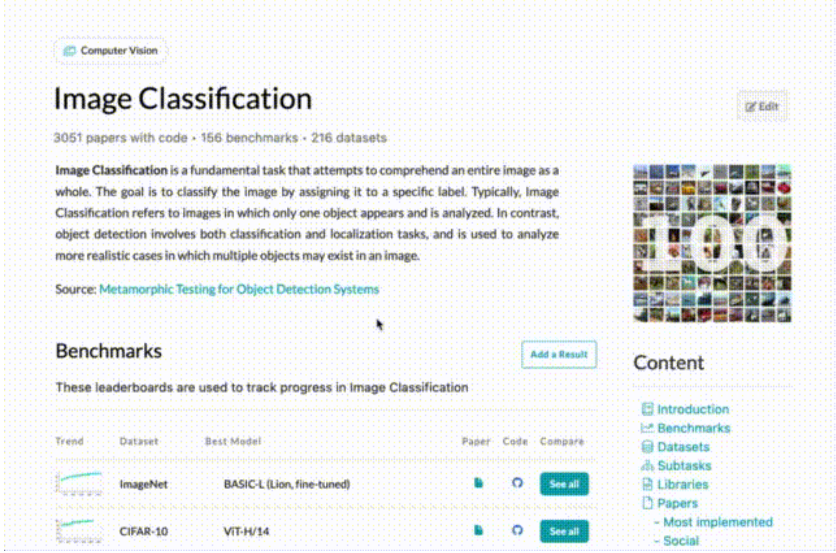
```python
from torchvision import transforms

transform = transforms.Compose([
transforms.ColorJitter(hue=0.5),
transforms.RandomRotation(30),
transforms.RandomHorizontalFlip(),
transforms.Resize((512, 512))
])

img_transformed = transform(img)
```


Resource: https://pytorch.org/vision/main/transforms.html

Resources to find a good model

Papers With Code





Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Resources: https://huggingface.co/models; https://paperswithcode.com/sota

Training the model

- **Process of training the model**
- **Question of optimal hyperparameters for training**

▼

- **Grid search, Random search, Bayesian Optimization**
- **Automated libraries for tuning hyperparameters**

Training (without automated Hyperparameter tuning)

```python
from transformers import
ViTForImageClassification

model =
   ViTForImageClassification.from_pretrai
   ned("google/vit-base").to('cuda')

training_args = TrainingArguments(
per_device_train_batch_size=64,
evaluation_strategy="steps",
num_train_epochs=20,
logging_steps=50,
learning_rate=1e-4,
)

trainer = Trainer(
     model=model,
     args=training_args)

train_results = trainer.train()
```

Training (with automated Hyperparameter tuning)

```python
import optuna
from transformers import
ViTForImageClassification

model =
   ViTForImageClassification.from_pretrai
   ned("google/vit-base").to('cuda')

def objective(trial):
   training_args = TrainingArguments(
   per_device_train_batch_size=trial.suggest_c
      ategorical('batch_size', [16,32,64]),
   evaluation_strategy="steps",
   num_train_epochs=20,
   logging_steps=50,
   learning_rate=trial.suggest_float('lr', 1e-
      5, 1e-3),
   load_best_model_at_end=True,
   )
   trainer = Trainer(
      model=model,
      args=training_args)
   train_results = trainer.train()
   return train_results['eval_accuracy']

study =
   optuna.create_study(direction='maximize')
study.optimize(objective, n_trials=100)
study.best_params
```

Python image classification

https://github.com/Maximilianwte/Image-tutorial/blob/main/Image_Classification.ipynb

Finally using your model to predict on new data

Model

Inside

3.75

Outside

4.52

```
from transformers import A
from transformers import
   ViTForImageClassificatic
from PIL import Image

image_processor =
   AutoImageProcessor.from_pretrained("go
   ogle/vit-base")

model =
   ViTForImageClassification.from_pretrai
   ned("/path/to/finetuned/model")

image = Image.open("/path/to/image")
inputs = image_processor(image,
return_tensors="pt")
with torch.no_grad():
    logits = model(**inputs).logits

predicted_label = logits.argmax(-1).item()
print('Label: '+
   model.config.id2label[predicted_label])
```

```
Label: 'Car'
```

Python image classification

https://github.com/Maximilianwte/Image-tutorial/blob/main/Image_Classification.ipynb

B Using Generative AI

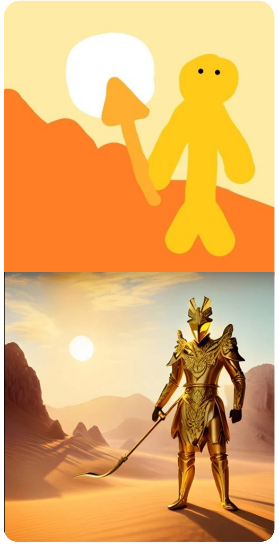
Diffusion models revolutionizing generative computer vision since 2022

Closed Source & Paid		Open Source & Free	
DALL-E 2	Midjourney	Stable Diffusion	DeepFloyd
http://labs.openai.com/auth/login	https://discord.com/invite/midjourney	Python	Python

Generating images can be done from text prompts or with an input image to transform

Prompting

Img2Img



Inpainting

Outpainting

Generating Images using Stable Diffusion

Stable diffusion notebook

https://github.com/Maximilianwte/Image-tutorial/blob/main/Stable_Diffusion.ipynb

```
!pip install diffusers
import torch
from diffusers import StableDiffusionPipeline

pipe = StableDiffusionPipeline.from_pretrained("CompVis/stable-diffusion-v1-5",
      revision="fp16", torch_dtype=torch.float16, use_auth_token=True).to("cuda")

images = pipe(prompt=“A dog writing a novel, cartoon”, height=512, width=512,
guidance_scale=7.5).images

images
```

Getting your data into the model

A Adapting an Input Image

Possible with all models

B Fine-tuning the weights and embeddings of the model

```python
pipe(prompt="Photo of xyzperson holding a handcreme", height=512, width=512,
guidance_scale=7.5).images[0]
```

Currently only with Stable Diffusion and DeepFloyd

Fine-tuning SD Dreambooth

https://github.com/Maximilianwte/Image-tutorial/blob/main/Finetuning_Stable-Diffusion_Dreambooth.ipynb

Resource: OpenAI Dalle-2 Web Interface.

A few principles on working with machine learning models

1 Generalization is key

2 Leverage pre-trained models

3 Better data trumps a larger model

4 Get fitting architecture and loss function

5 Prefer the simpler model

Resource: Domingos, P. (2012). A few useful things to know about machine learning. Communications of the ACM, 55(10), 78-87.

Resources

Presentation GitHub Repository

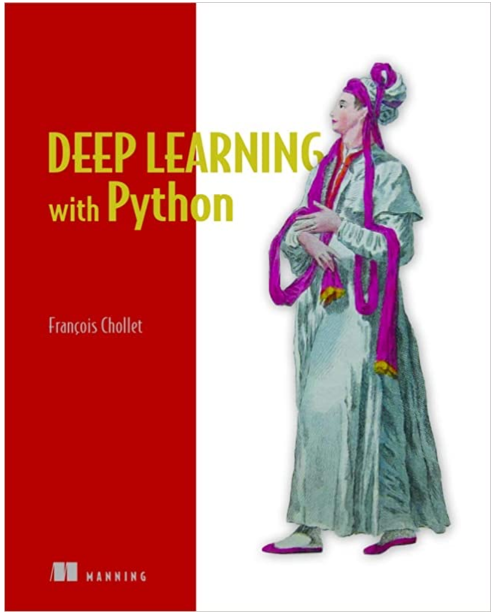
https://github.com/Maximilianwte/Image-tutorial

Blogs

https://huggingface.co/docs

https://towardsdatascience.com/

Book Recommendation



https://www.manning.com/books/deep-learning-with-python

Contact

Maximilian Witte

Chair of Marketing &
Customer Insight

University of Hamburg

#Unstructured data **#Generative AI**

#Decision Aids

maximilian.witte@uni-hamburg.de

linkedin.com/in/maximilianwte