HANDS-ON AI I

Reading, Handling and Visualization of Datasets

Andreas Schörgenhumer
Institute for Machine Learning

JOHANNES KEPLER
UNIVERSITY LINZ

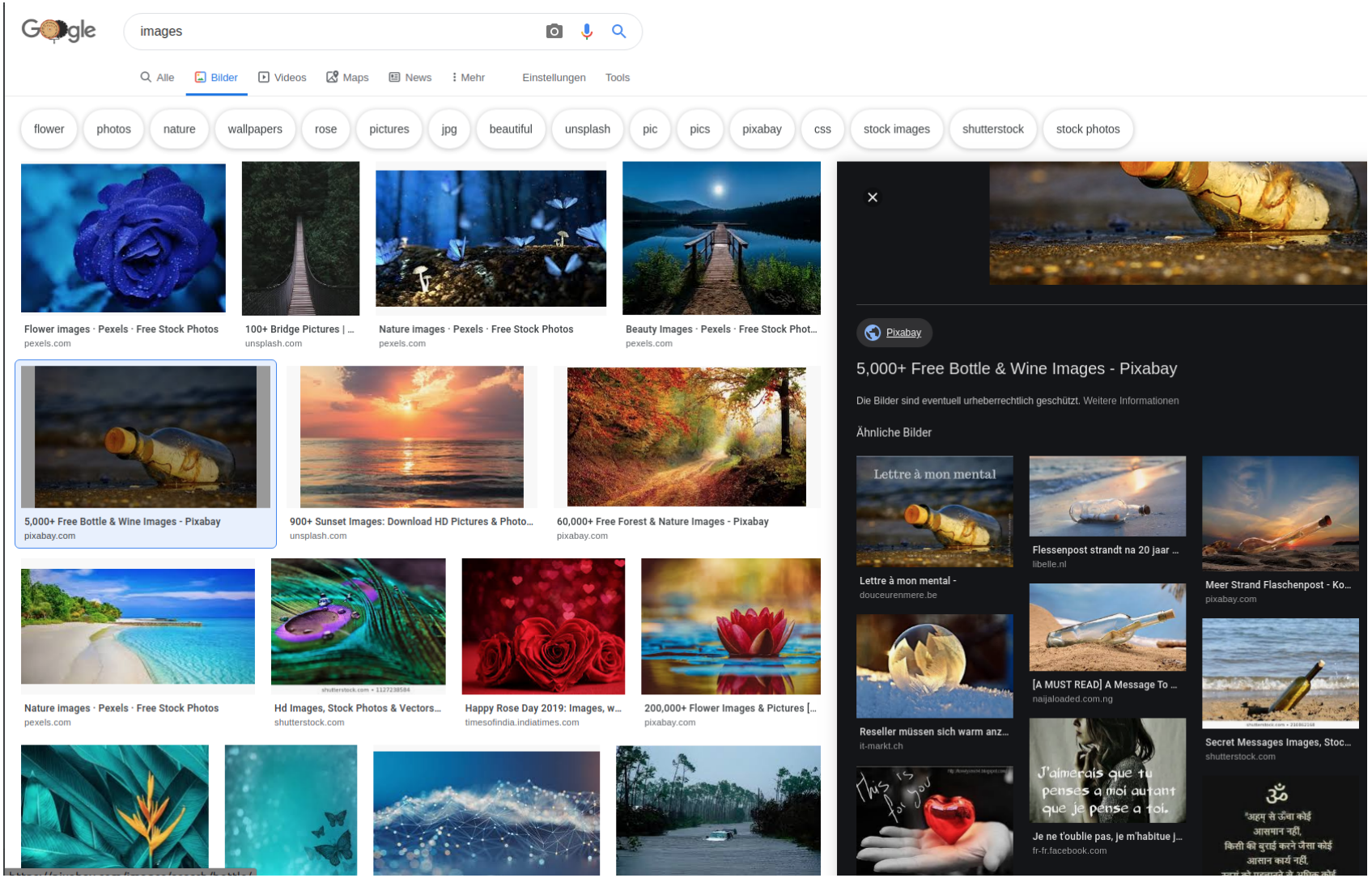
Institute for
Machine Learning

Copyright Statement

Content of Unit 2

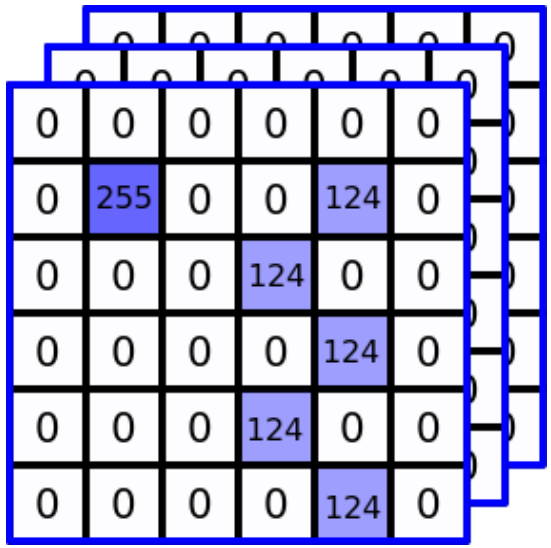
- Second data source: images
- Third data source: sequential data
  - Sound
  - Text

Images


images
Alle
Bilder
Videos
Maps
News
Mehr
Einstellungen
Tools
flower
photos
nature
wallpapers
rose
pictures
jpg
beautiful
unsplash
pic
pics
pixabay
css
stock images
shutterstock
stock photos
Flower images · Pexels · Free Stock Photos
pexels.com
100+ Bridge Pictures | ...
unsplash.com
Nature images · Pexels · Free Stock Photos
pexels.com
Beauty Images · Pexels · Free Stock Phot...
pexels.com
5,000+ Free Bottle & Wine Images - Pixabay
pixabay.com
900+ Sunset Images: Download HD Pictures & Photo...
unsplash.com
60,000+ Free Forest & Nature Images - Pixabay
pixabay.com
Nature images · Pexels · Free Stock Photos
pexels.com
Hd Images, Stock Photos & Vectors...
shutterstock.com
Happy Rose Day 2019: Images, w...
timesofindia.indiatimes.com
200,000+ Flower Images & Pictures [...
pixabay.com
Pixabay
5,000+ Free Bottle & Wine Images - Pixabay
Die Bilder sind eventuell urheberrechtlich geschützt. Weitere Informationen
Ähnliche Bilder
Lettre à mon mental
Lettre à mon mental -
douceurenmere.be
Flessenpost strandt na 20 jaar ...
libelle.nl
Meer Strand Flaschenpost - Ko...
pixabay.com
Reseller müssen sich warm anz...
it-markt.ch
[A MUST READ] A Message To ...
naijaloaded.com.ng
Secret Messages Images, Stoc...
shutterstock.com
This is for you
J'aimerais que tu penses à moi autant que je pense à toi.
Je ne t'oublie pas, je m'habitue j...
fr-fr.facebook.com
"अहम् से ऊँचा कोई आसमान नहीं,
किसी को बुराई करने जैसा कोई आसान कार्य नहीं.

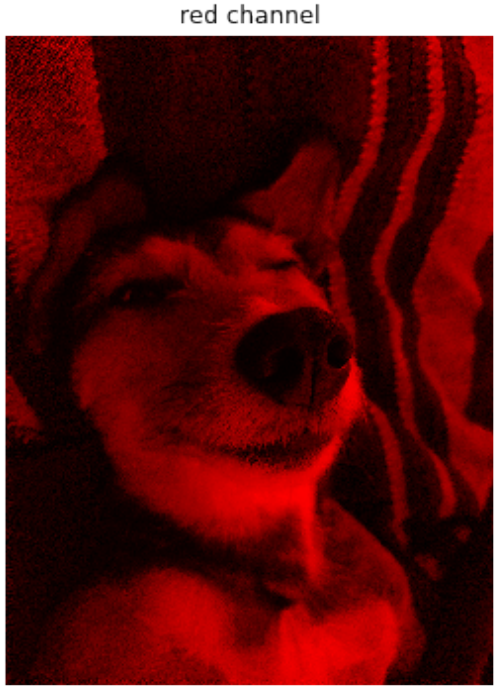
Representation of Images

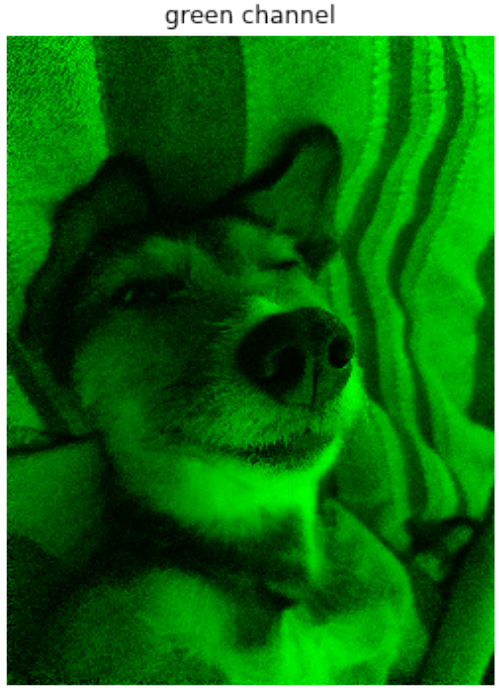
- Images are usually represented in **3 dimensions**:
  - Height
  - Length (or width)
  - Channels (often: red, green, blue)
- **Color depth** refers to the number of possible values for each channel of a pixel.
  - Color depths of **8-bit** ($2^8 = 256$ values) and **16-bit** ($2^{16} = 65536$ values) are common.
  - Higher values mean increased intensity.

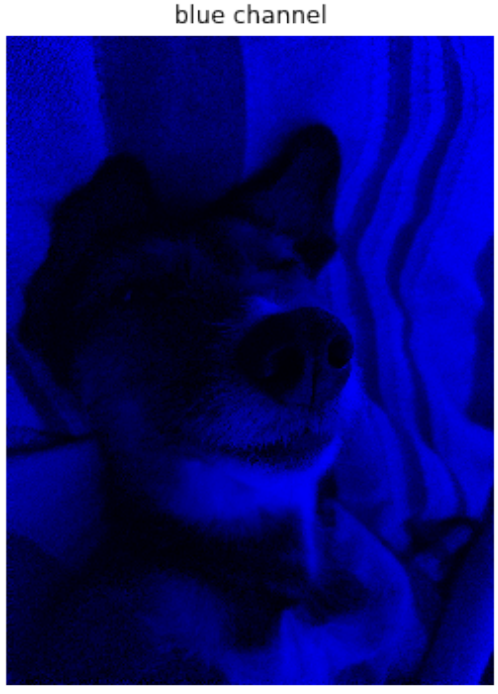


RGB Model

- The RGB (red, green, blue) model is the most important model for colored images.
- 3 channels: **red, green, blue**
- Adding up the 3 color channels results in the final image.







RGB Model

- The RGB (red, green, blue) model is the most important model for colored images.
- 3 channels: **red, green, blue**
- Adding up the 3 color channels results in the final image.

RGBA Model

- The RGBA model uses a fourth channel: **alpha**.
- Alpha controls the transparency of the image.
- Higher values mean increased opaqueness ($\alpha = 0 \rightarrow$ fully transparent, $\alpha = \max \rightarrow$ fully opaque).

Grayscale

- Grayscale images have only **one channel representing the brightness**.
- Color images can be converted to grayscale images (different grayscale conversions exist), but not vice versa (without additional information).

Data Augmentation

- The field of **computer vision** focuses on processing and understanding images.
- Strong shift towards machine learning as **a way to understand, manipulate and even generate images**.
- Example: We “show” a machine learning model a large number of images with and without Charlie.
  - Train it to distinguish between these two groups.
  - **Augment** the images to reduce the workload of getting a lot of Charlie images.
  - Feed original and/or augmented images to our model.

Data Augmentation

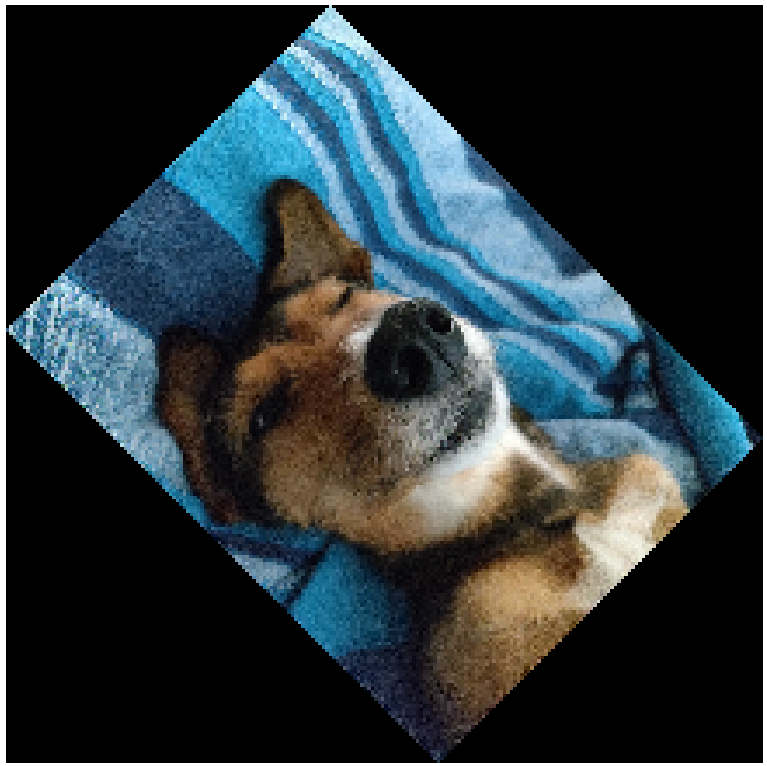
- In many cases, we can augment data artificially without collecting new real samples: Create "new" **artificial samples** by modifying existing samples.
- Pros:
  - Can increase the number of data points by a large factor with little effort (often on-the-fly).
  - Reduces overfitting, increases robustness of model (we will learn about this in a later unit).
- Cons:
  - Can introduce artifacts or change the task.
  - Heavily dependent on data, model and task.

Data Augmentation Techniques

- There exist a lot of data augmentation **techniques**.
- Some can be applied across different fields (e.g., adding noise), some are specific and only applicable in certain scenarios.
- Here are some common examples in the area of **image augmentation**:
  - Rotation
  - Flipping
  - Zooming/Cropping
  - Blurring
  - Noise
  - Input Dropout
  - Distortion Effects
  - Color Jittering
  - ...

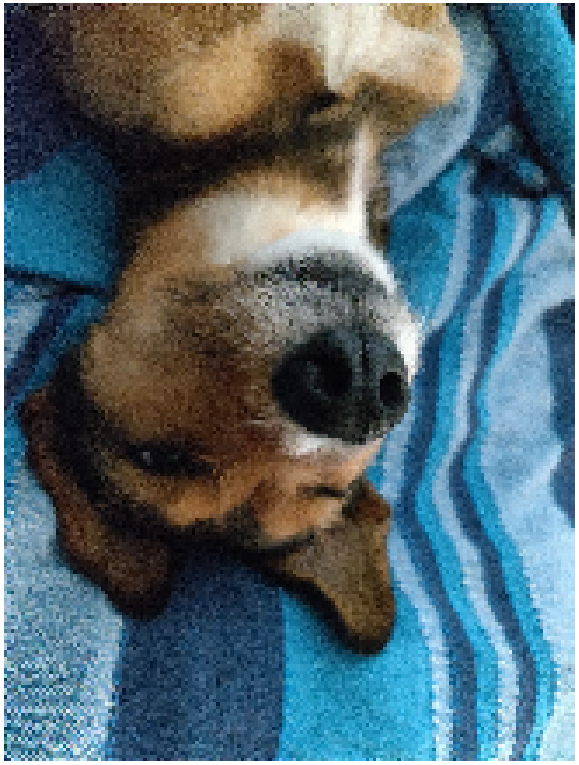
Rotation

- Rotate images around their center by a certain angle.
- It always depends on the task how much rotation is reasonable.



Flipping

- Mirroring the image across its vertical or horizontal axis.
- It depends on the task if flipping is reasonable.

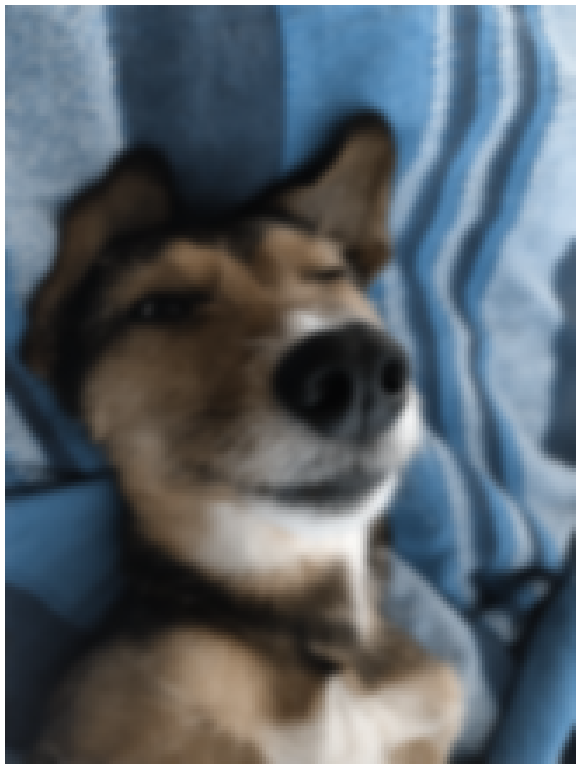


Zooming/Cropping

- Zoom into the image (different sizes of Charlie).
- Often, the same size of the image is required:
  - Selecting a smaller part and resizing it to the original resolution.

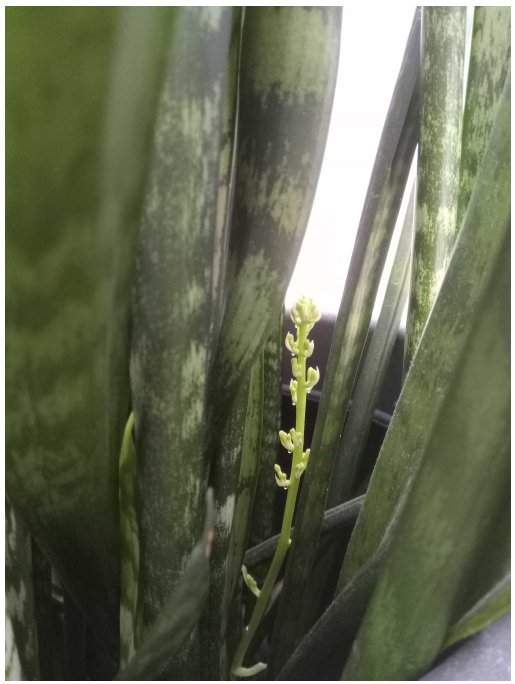
Blurring

- Usually not all images are super sharp.
  - Blurred images help the machine learning model to detect blurred Charlies as well.
- Gaussian blurring is most common blurring technique.



Noise

- Add random noise such that input feature values (pixels) are still plausible.
- Noise strength can be parameterized.



original

light noise

heavy noise

Input Dropout

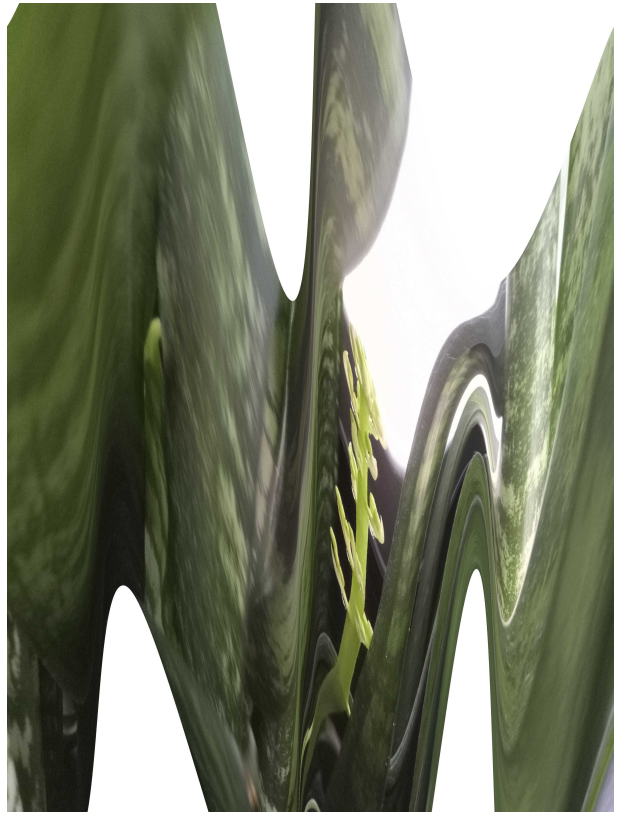
- Temporarily remove input features to mimic data errors or hidden image content.
- Different dropout techniques exist, e.g., removing individual R, G or B values, or even entire (adjacent/neighboring) pixels over all channels.

dropout random values

adjacent pixels

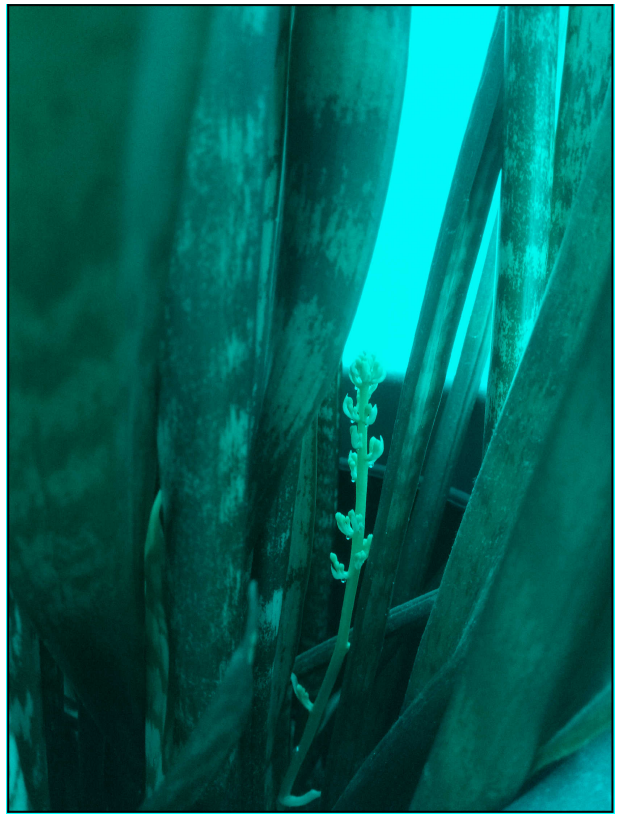
Distortion Effects

- Warp image using, e.g., lens-distortion-like effects.



Color Jittering

- Modify color channels, brightness, hue, saturation or value.



Some Notes on Data Augmentation

- Very useful to get more data without having to explicitly "collect" it from the source.
- Increases variation and thereby allows for a better generalization.
- **Careless augmentations** can actually have a **negative impact**. Example:
  - Extreme zoom on Charlie's nose.
  - Only black pixels remain with no resemblance of Charlie whatsoever.
  - Training a machine learning model with this input (while still saying this is Charlie) will yield worse results.
- Data augmentation is typical in image processing but not restricted to it.
  - Example: Add random noise to time series signal.

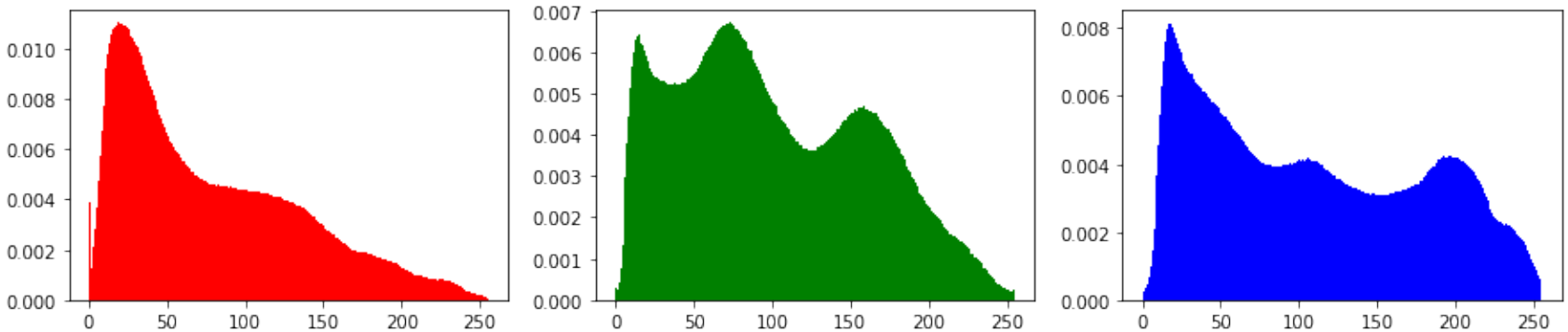
Image Segmentation

- A common problem in computer vision is to mark an object within an image, e.g., a dog with some background.

- We segment the image into **object** and **no object**: **image segmentation**.
- Wide field of research.

Simple Solution to Segment Charlie

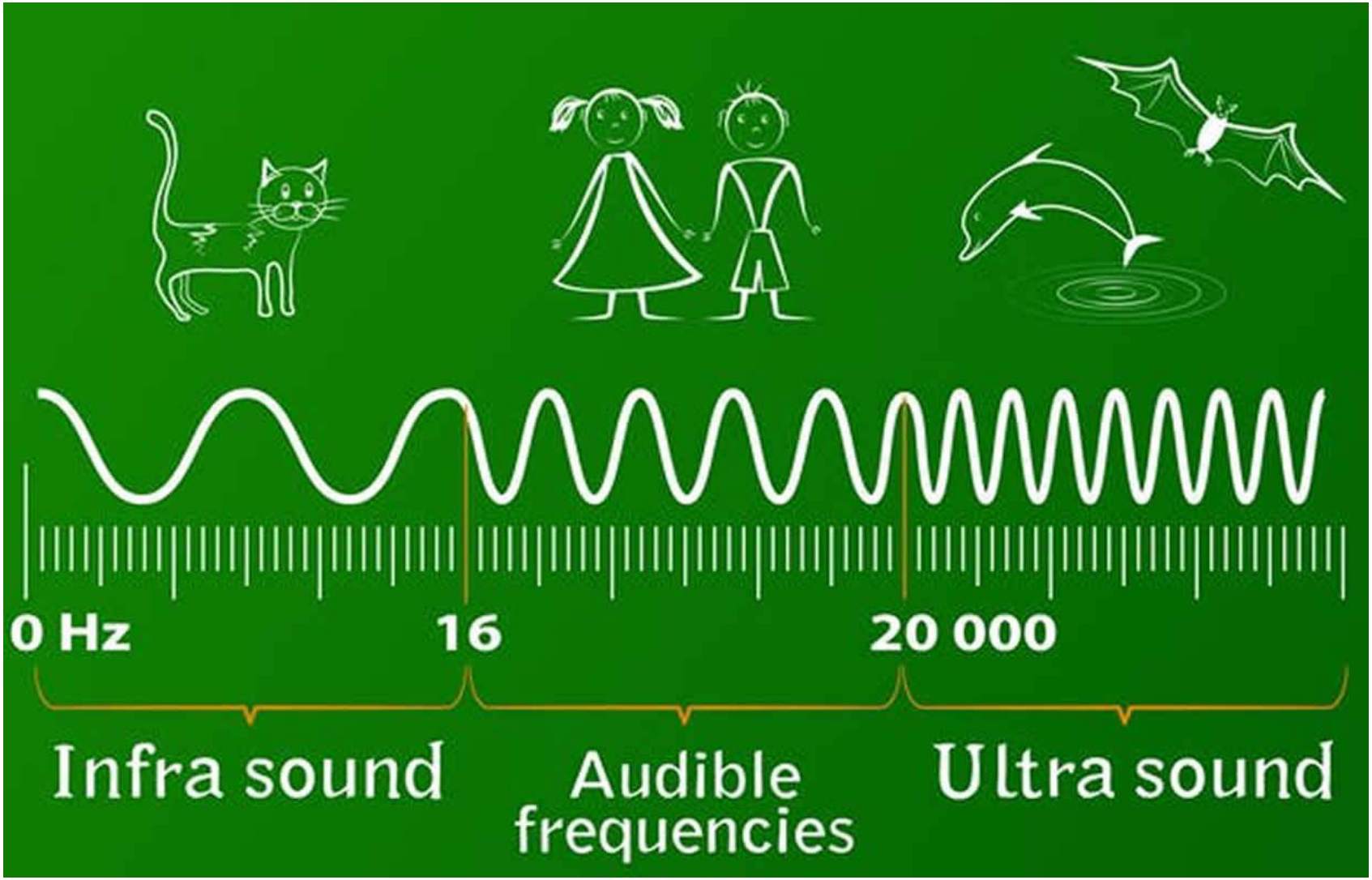
- Occurance of individual color values of an image.
- Pixels with RGB values ranging from 0 to 255.
- Setting thresholds to segment between Charlie and background.



Sequences

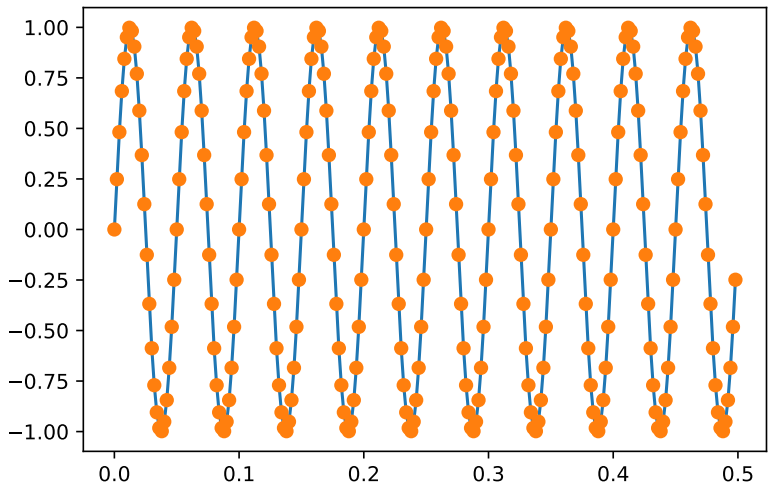
- A **sequence** is a type of data where the order of the values matters: $(a_1, a_2, a_3, a_4, \ldots, a_n)$
- Some examples:
  - Text
  - Sound (or time series in general)
  - Strictly speaking: images (order of pixels is essential)
  - ...

Sequences: Sound



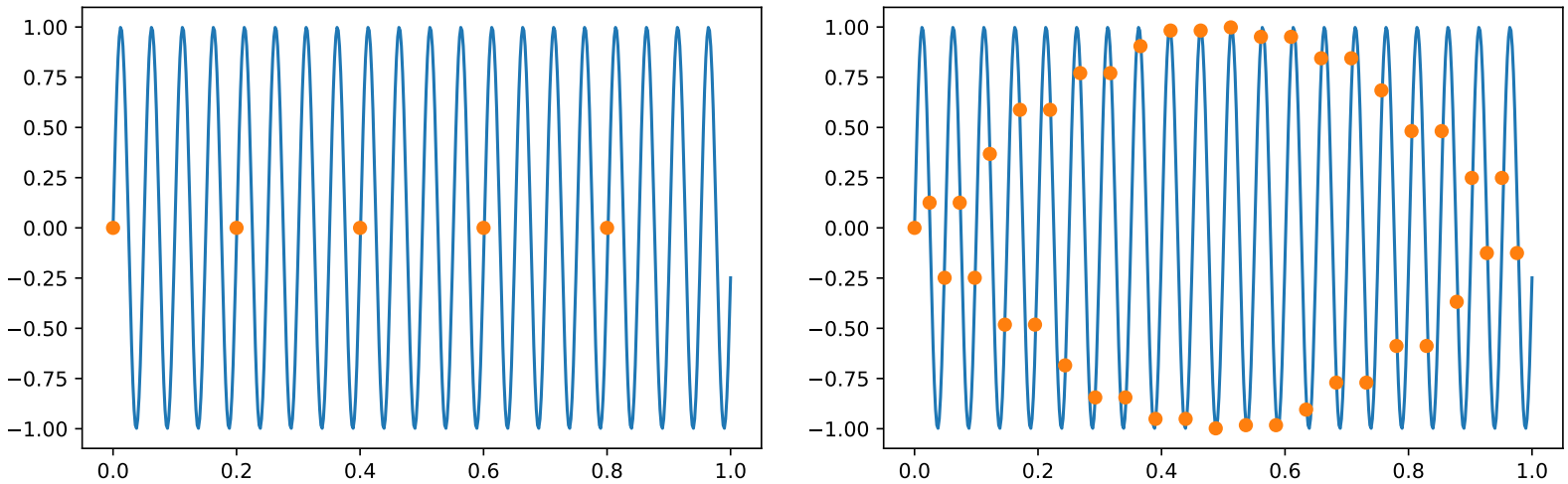
Continuous Signal

- Sound waves are a continuous signal.
- In physics, the **sound frequency** is measured in Hertz (Hz).
- In data analysis, the **sampling rate** from the continuous signal (per second) is also measured in Hz.
  - For example, a sampling rate of 48 kHz means retrieving 48000 samples per second.



Low Sampling Rates

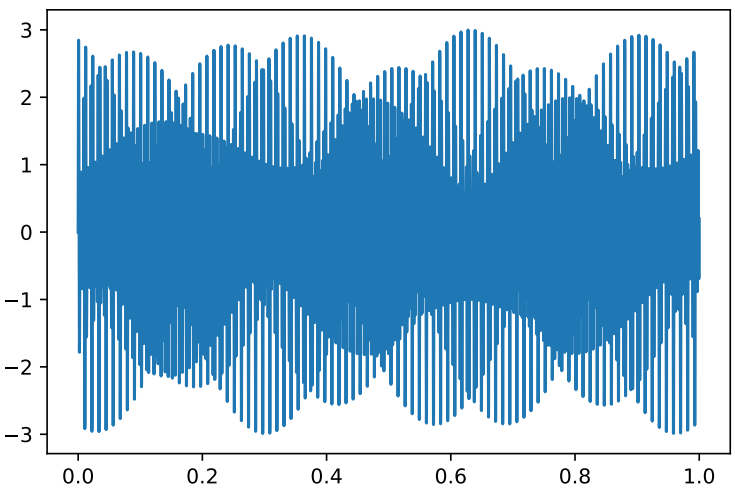
- If the sampling rate is too low, it is not possible to exactly reconstruct the signal.
- **Nyquist-Shannon theorem**: A signal is completely determined by a series of points that are $< \frac{1}{2f}$ seconds apart (f is the highest frequency in the signal), or in other words, the sampling rate f_s must be $> 2f$.

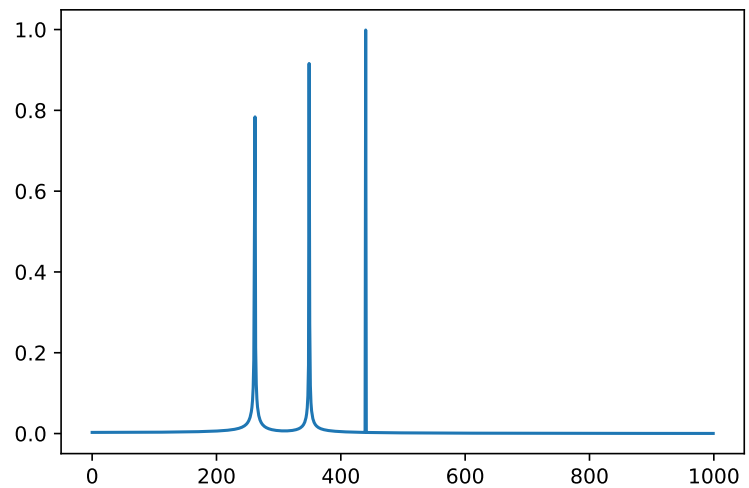


- Higher sampling rates mean higher audio quality.

Fourier Transformation

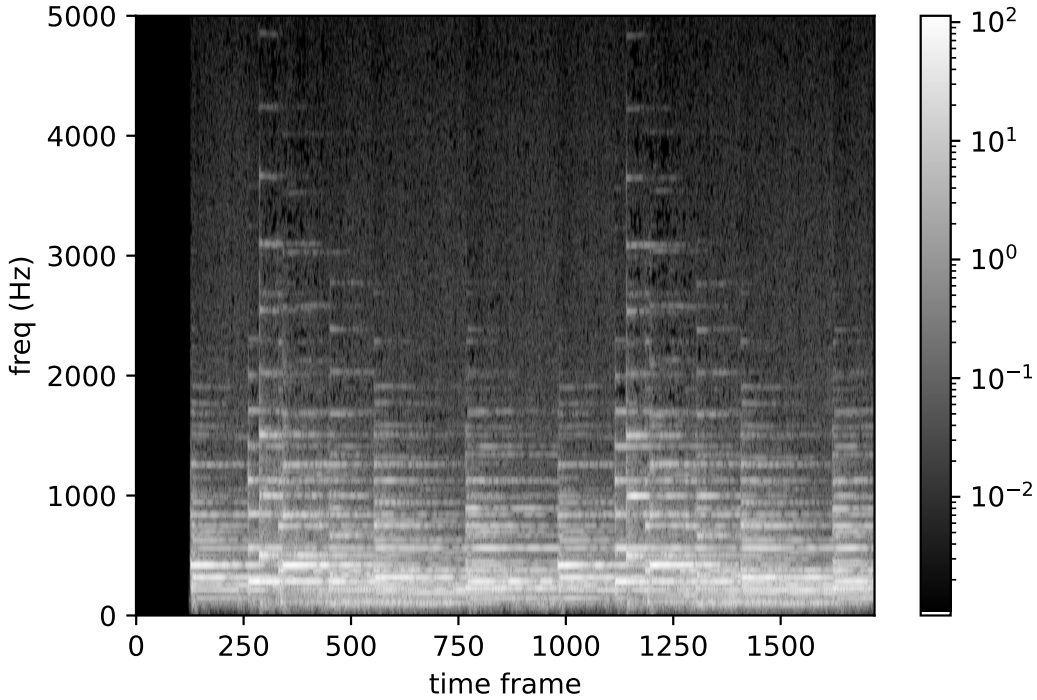
- The **Fourier transformation decomposes a signal** (a function of time) **into its constituent frequencies** (a spectrum).
- The Fourier transformation is not limited to functions of time, but the domain of the original function is commonly referred to as the **time domain** (example left image).
- The domain of the results is typically called the **frequency domain** (example right image).





Spectrogram

- Spectrograms are visual **representations of the variation of the frequencies over time**:
  - This can be done by computing a spectrum for the first few milliseconds (time frame), another one for the next, etc.
  - Stitching these spectra together, we obtain a spectrogram.
  - It tells how much energy (=gray value) is in the signal at each time frame and frequency bin (=pixel).



Sequences: Text

Vivamus vehicula leo a justo. Quisque nec augue. Morbi mauris wisi, aliquet vitae, dignissim eget, sollicitudin molestie, ligula. In dictum enim sit amet risus. Curabitur vitae velit eu diam rhoncus hendrerit. Vivamus ut elit. Praesent mattis ipsum quis turpis. Curabitur rhoncus neque eu dui. Etiam vitae magna. Nam ullamcorper. Praesent interdum bibendum magna. Quisque auctor aliquam dolor. Morbi eu lorem et est porttitor fermentum. Nunc egestas arcu at tortor varius viverra. Fusce eu nulla ut nulla interdum consectetuer. Vestibulum gravida. Morbi mattis libero sed est.

Natural Language Processing (NLP)

- NLP is a subfield of:
  - Computer science
  - Linguistics
  - Deep Learning
- Challenges in NLP involve:
  - Speech recognition
  - Language understanding
  - Language generation
  - Language translation
- Nowadays, NLP is strongly driven by modern **Deep Learning** methods. We will learn more about this in Hands-on AI II.

Text as Categorical Data – Theory

- Normally, text can be considered qualitative data without mathematical meaning.
- Mathematical operations (e.g., $\sum$ or comparisons such as “less than”) do not make sense.
- Example: “Dog”, “Rat”, “Cat”

Text as Categorical Data – Practice

- Assume we consider n different categories ($= n$ different words), which we can also call **vocabulary** or **dictionary**.
- We could represent each category as an integer value:
  - Requires n different integer values, one for each category.
  - Example: "Dog"$= 0$, "Rat"$= 1$, "Cat"$= 2$
- Problem:
  - We would introduce new (probably false) information.
  - Example: In our ranking, "Dog" $<$ "Rat" and "Rat" $\cdot 2 =$ "Cat".

  $\rightarrow$ Not suitable for us.

Text as Categorical Data – Practice

- Solution: Represent a categorical feature as **one-hot-encoded** binary vector $\boldsymbol{v} = (v_1, \ldots, v_n)$:
  - Categorical data with n different values is enumerated from $1 \ldots n$ given a **fixed order**.[1]
  - $v_i = 1$ if category i is true, otherwise $v_i = 0$.
  - Each element in the vector represents one category $\rightarrow$ no false information![2]
- Example:
  - Possible values: “Dog”, “Rat”, “Cat” ($n = 3$)
  - Sample is “Dog” $\rightarrow \boldsymbol{v} = (1, 0, 0)$
  - Sample is “Cat” $\rightarrow \boldsymbol{v} = (0, 0, 1)$

[1] The order itself is not important, just that it is fixed.

[2] Only applies if information about order in feature vector is not used.

One-Hot Encoding Example

- Vocabulary with $n = 7$ and the following fixed order:

Word	an	awesome	cat	Charlie	dog	is	rat
Position	1	2	3	4	5	6	7

- Recall: For each word, we have a vector of size n where all numbers are zero but the element representing a specific word is set to 1.
- Example one-hot-encoded word “awesome” (position $= 2 \rightarrow v_2 = 1$, all other $v_i = 0$):
  $\boldsymbol{v}_{\text{awesome}} = (0, 1, 0, 0, 0, 0, 0)$

One-Hot Encoding Example

- Entire one-hot encoded vocabulary:

	an	awesome	cat	Charlie	dog	is	rat
an	1	0	0	0	0	0	0
awesome	0	1	0	0	0	0	0
cat	0	0	1	0	0	0	0
Charlie	0	0	0	1	0	0	0
dog	0	0	0	0	1	0	0
is	0	0	0	0	0	1	0
rat	0	0	0	0	0	0	1

Word Embedding

- Problem: For large vocabularies, the usage of one-hot encoding is not ideal. For example, a vocabulary size of $n = 30\,000$ would yield vectors of the same size ($= 30\,000$ dimensional feature vectors)!
- **Word embedding**: Represent each word by a much smaller vector but with real numbers.
- Ideally, a word embedding captures additional information compared to a simple encoding, e.g., (semantic) similarities between words.
- How to get a "good" embedding?
  - Not straightforward.
  - Often learned from a big corpus of words (in Hands-on AI II, we will actually learn an embedding ourselves).
  - Prelearned embeddings exist.

Downprojection of Word Embedding

- As always, visualizing our data can help in gaining insights.
- While embeddings are typically much smaller than simple encodings (e.g., 300 features per vector), this is still too much to visualize.
- Solution: **Reduce dimensionality** to **downproject** the embedding.
- If the embedding incorporates similarities, the downprojection of embedded words enables us to identify those.

Downprojection of Word Embedding