Bike Price Prediction

Matthias Fast, Amos Dinh

Semester Project

Agenda

1. Problem formulation
2. Crawling for data
3. Filtering and cleaning the data
4. Loss function
5. Model Training
6. Visualization of activations (Conclusion)

1. Problem formulation and motivation

- Suppose you can identify bicycles which are underpriced based on an image of the bicycle
- You can resell the bicycle to make profit

-> Problem formulation:

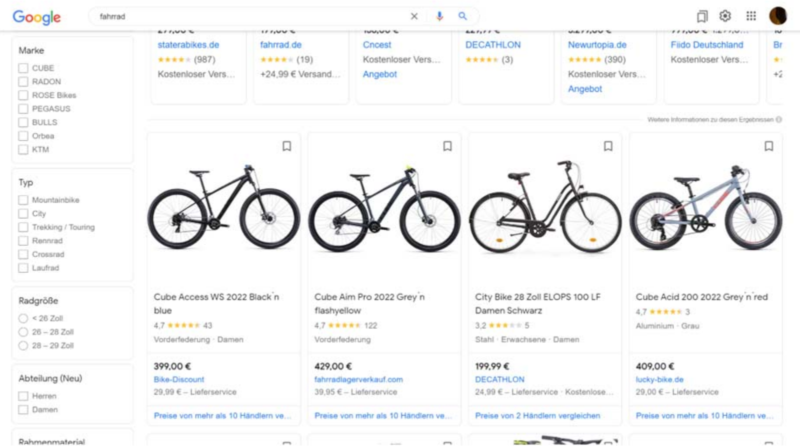
„Predict offer-price in euros, given an image of a bike”

2. Crawling

„Predict offer-price in euros, given an image of a bike"

- **(Image, price)** mapping can be consistently obtained with webscraping
- **scraping of bicycle-images**:
  - High Quality:
    - www.fahrrad.de (1500)
    - www.fahrrad-xxl.de (5700)
  - Low Quality:
    - www.google.com/... &tbm=shop (52000)

  ~60000 Images



3. Filtering and Cleaning

- Htsxywfnsyx%ktw%fs%nrflj
  - Ymj%unhyzwj%mfx%yt%htsyfns%tsj%gnh~hqj

  Wjizhj%ymj%frtzsy%tk%ifyf%sjjiji%-Sfwwt|ji2it|s%uwtgqjr%
  ktw%zqfynts.?
    - Ymj%gnh~hqj%mfx%yt%kfhj%xnij|f~x
    - Ymj%gfhplwtzsi%tk%ymj%nrflj%sjjix%yt%gj%rtstytstzx3

-> Aim to remove deviating images

3. Filtering and Cleaning

3.1. Filter out non-bicycle images

3.2. Remove non-monotonous and other images

3.3. Remove duplicates

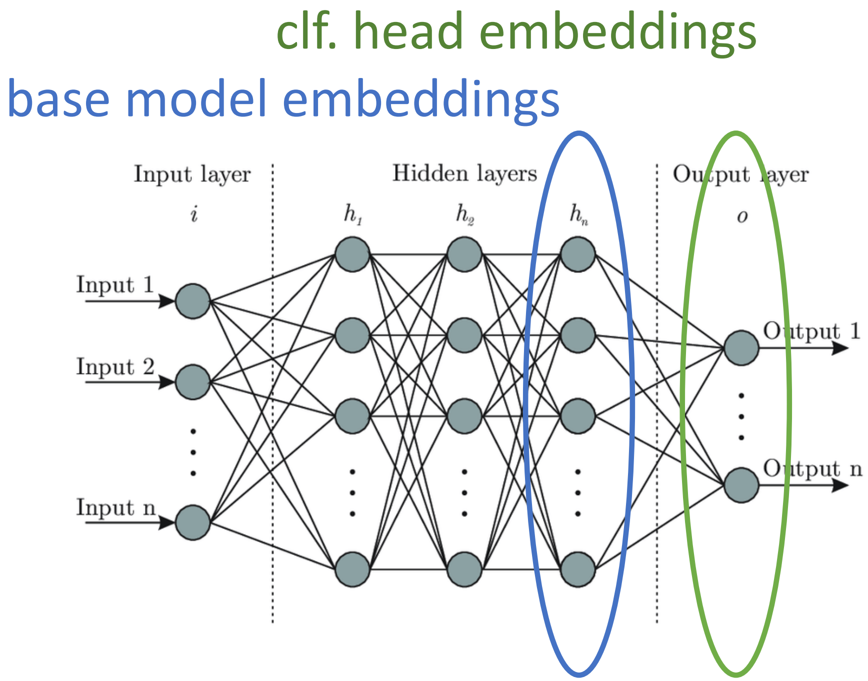
3.4. Is there bias in the images?

(3.5.) Preprocessing and mean image

3.1. Filter out non-bicycle images

Chosen method:

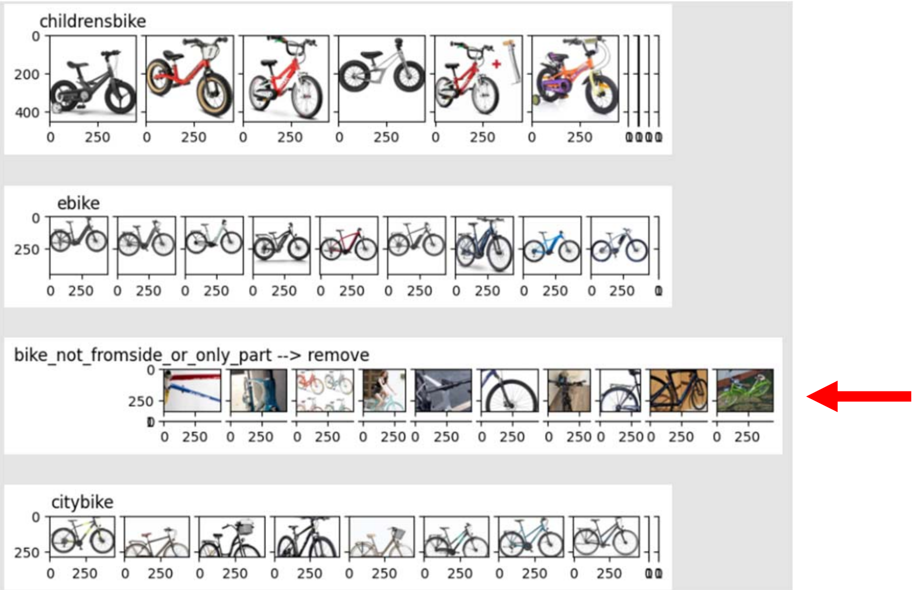
1. Compute image embeddings using ResNet50 CNN (ImageNet)
2. filter images: label “bad samples” by hand and use their embeddings to localize similar “bad” images
3. Filter images with visual verification

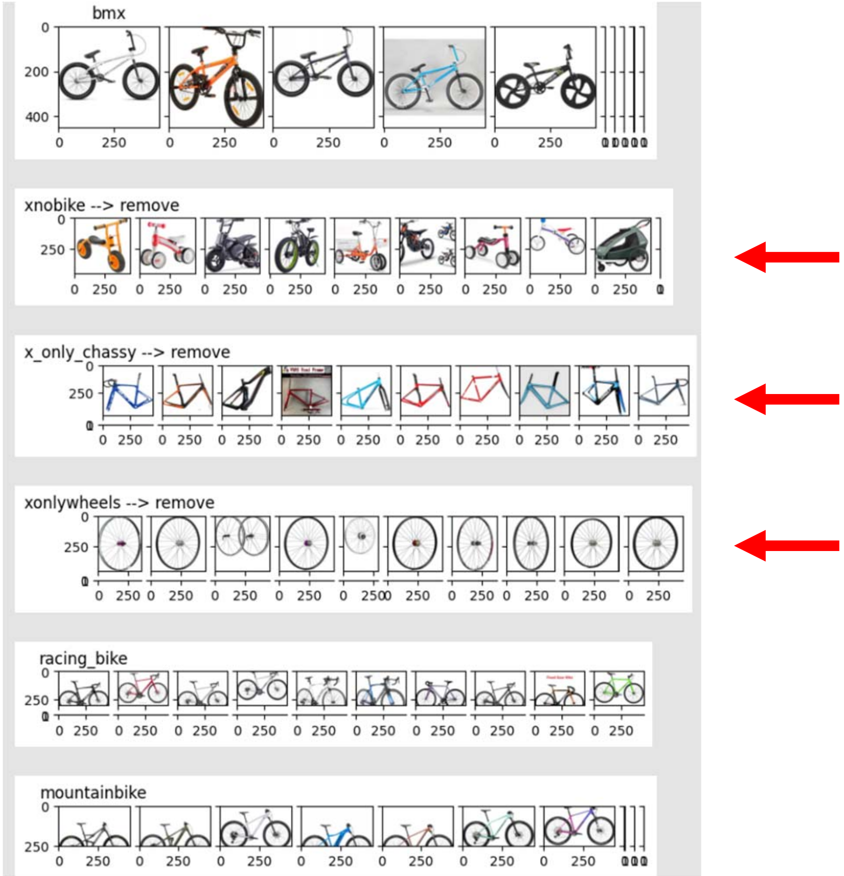


3.1. Filter out non-bicycle images

```
# list handlabeled classes
!ls /content/content/image_classes
```

```
bike_not_fromside_or_only_part        childrensbike   mountainbike   x_only_chassy
bike_with_nonmonotonous_background    citybike        racing_bike    xonlywheels
bmx                                   ebike           xnobike
```

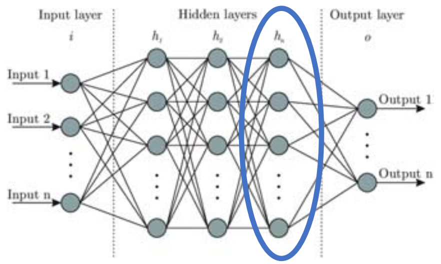


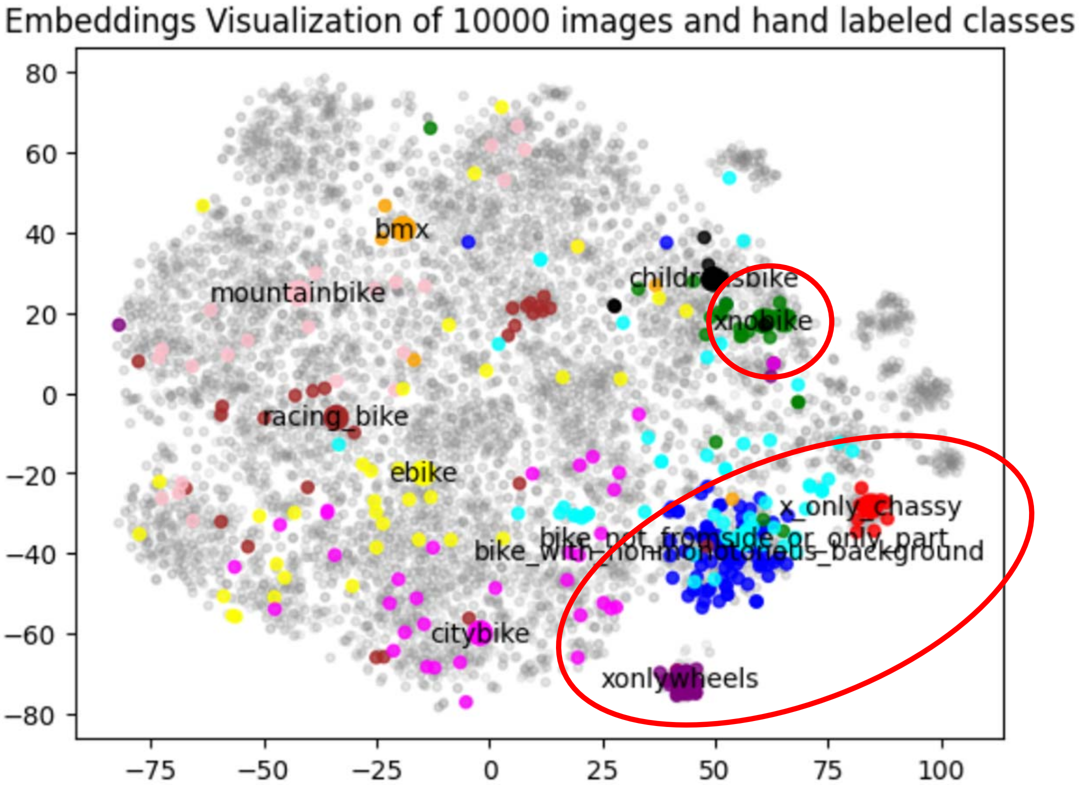


3.1. Filter out non-bicycle images

TSNE of ResNet base:

- Embeddings of classes to be removed are marked red.
- xnobike (green) is not well separated

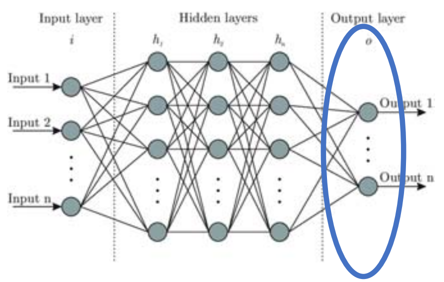


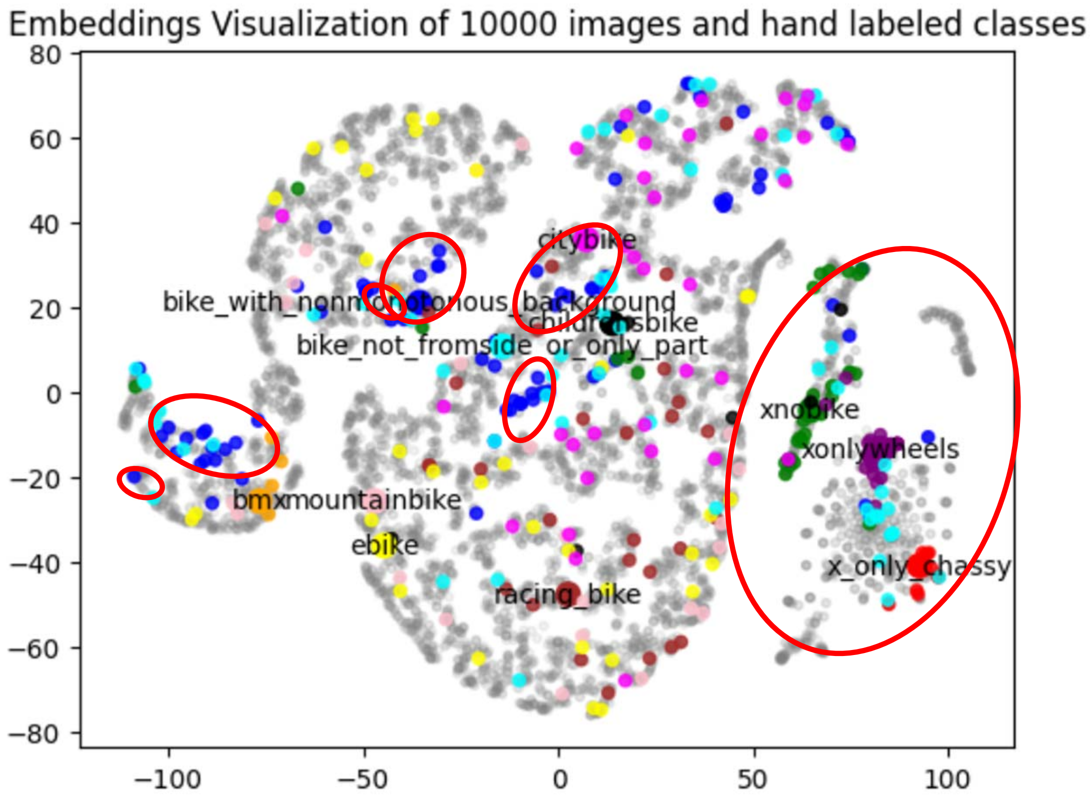


3.1. Filter out non-bicycle images

TSNE of ResNet with classification head:

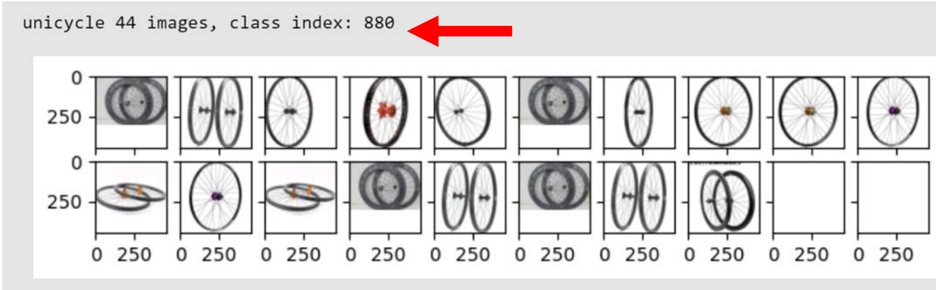
- xnobike (green) is well separated
- other classes (e.g. blue, cyan) are not well separated

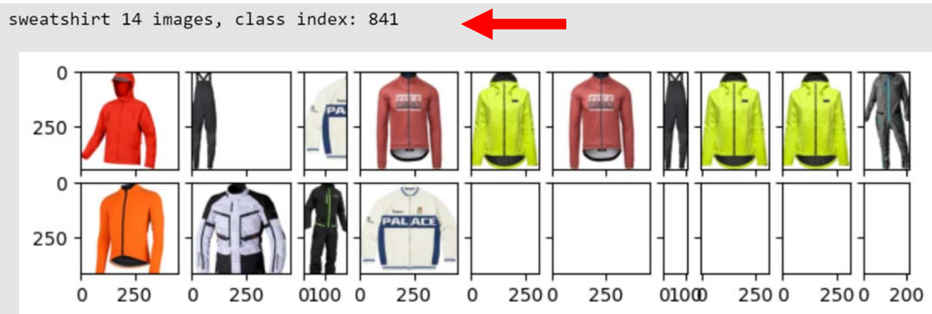




3.1. Filter out non-bicycle images

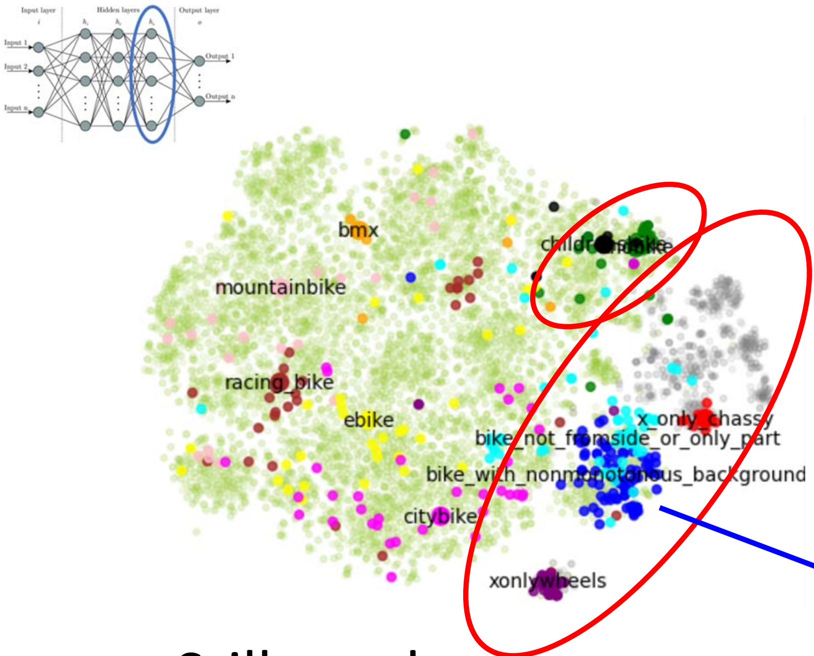
- Examination of top-1 predictions of the classification head images shows many “top-1 classes” to be removable

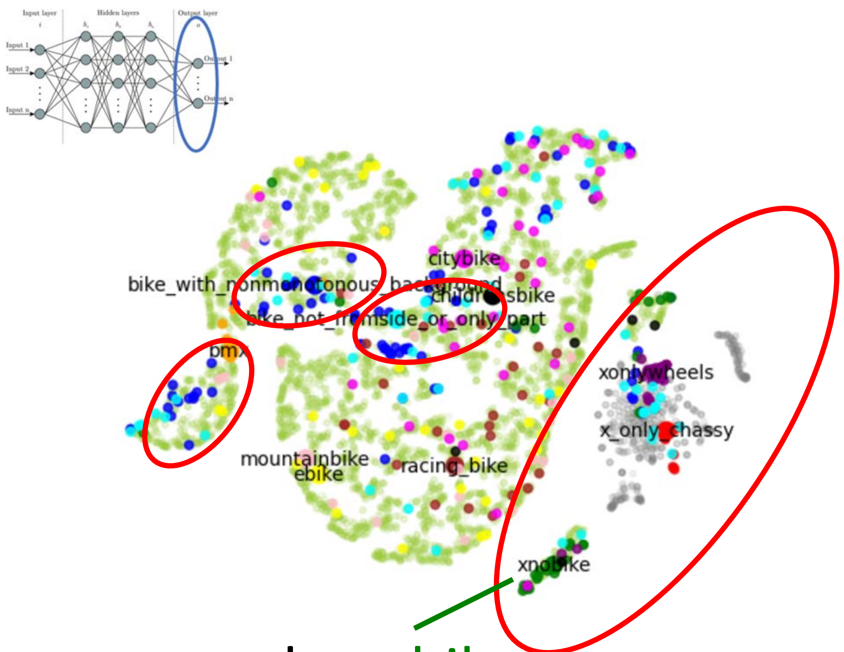




3.1. Filter out non-bicycle images

- Embeddings after removal, 5200 images are removed, 54000 are kept

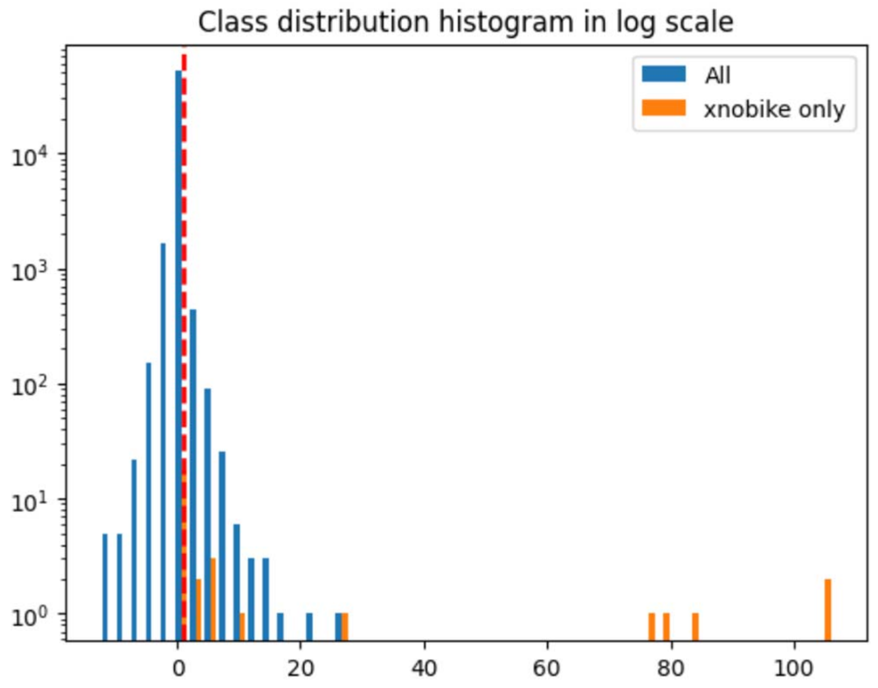




-> Still need to remove non-monotonous and xnobike

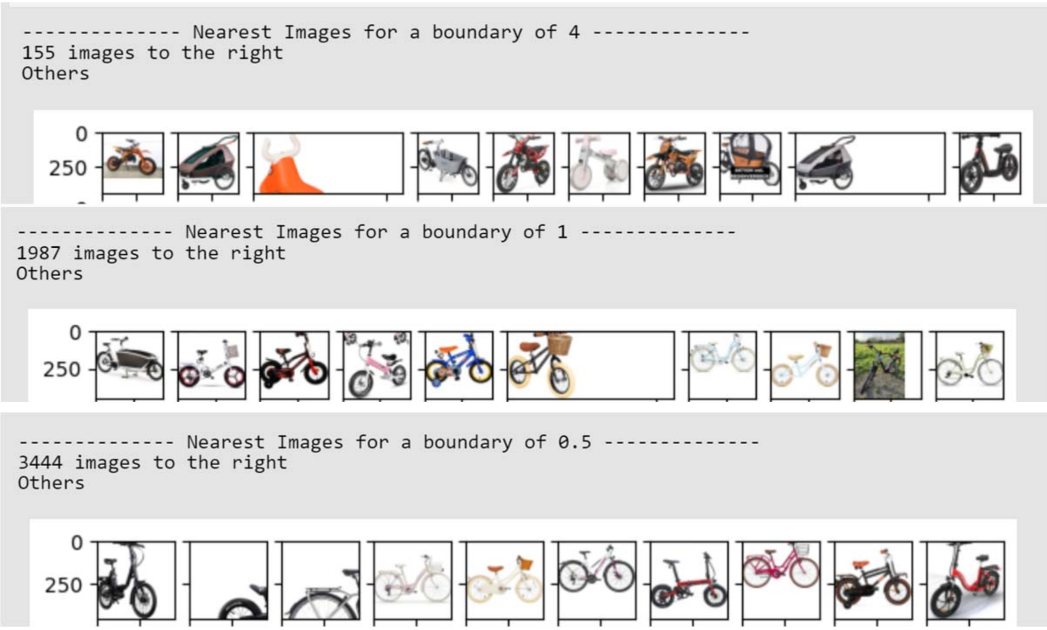
3.2. Remove non-monotonous and xnobike

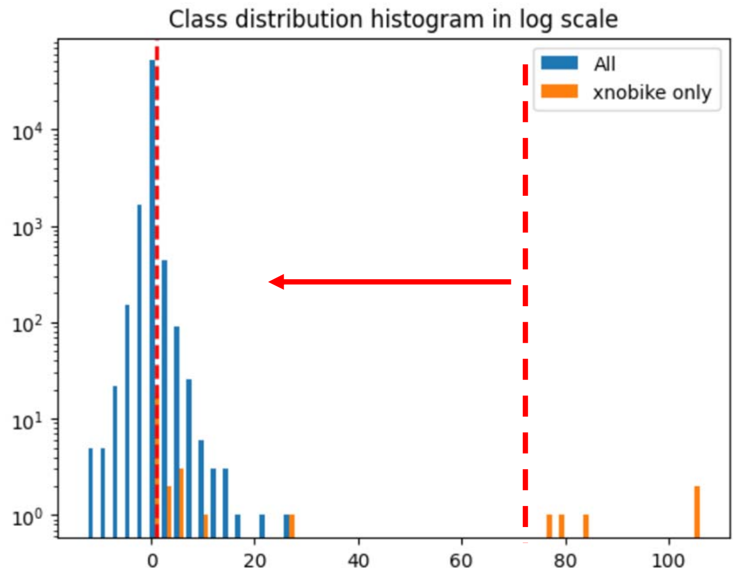
1. Perform LDA on embeddings of hand-labeled images vs. all other images



3.2. Remove non-monotonous and xnobike

2. Manually verify images kept for different thresholds



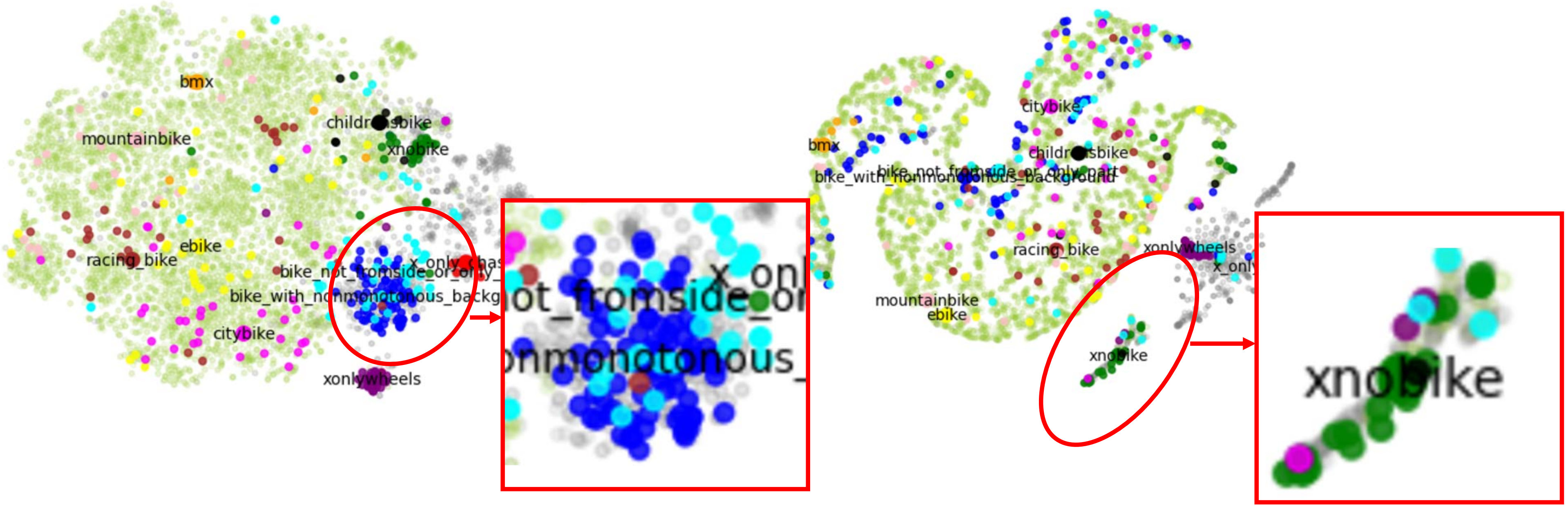


-> xnobike: Choose threshold 1, remove 1987 images, keep 52021

-> non-monotonous: threshold 0.97, remove 3194 images, keep 48825

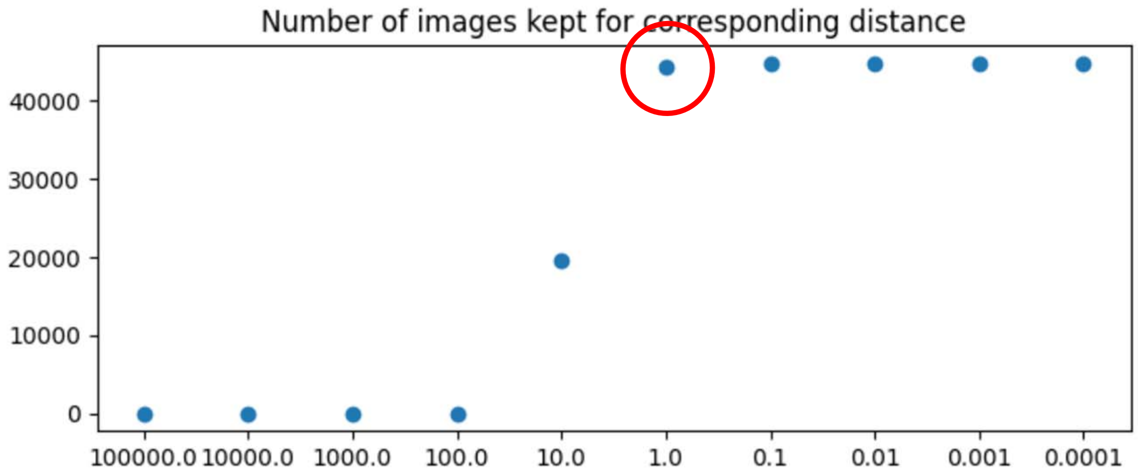
3.2. Remove non-monotonous and xnobike

- Removal appears to be mostly successful



3.3. Remove duplicate images

- Many duplicate images (Google)
- Reuse embeddings, since they also represent “similarity” to a certain degree
- Group images by Euclidean distance, groups of “distance x”

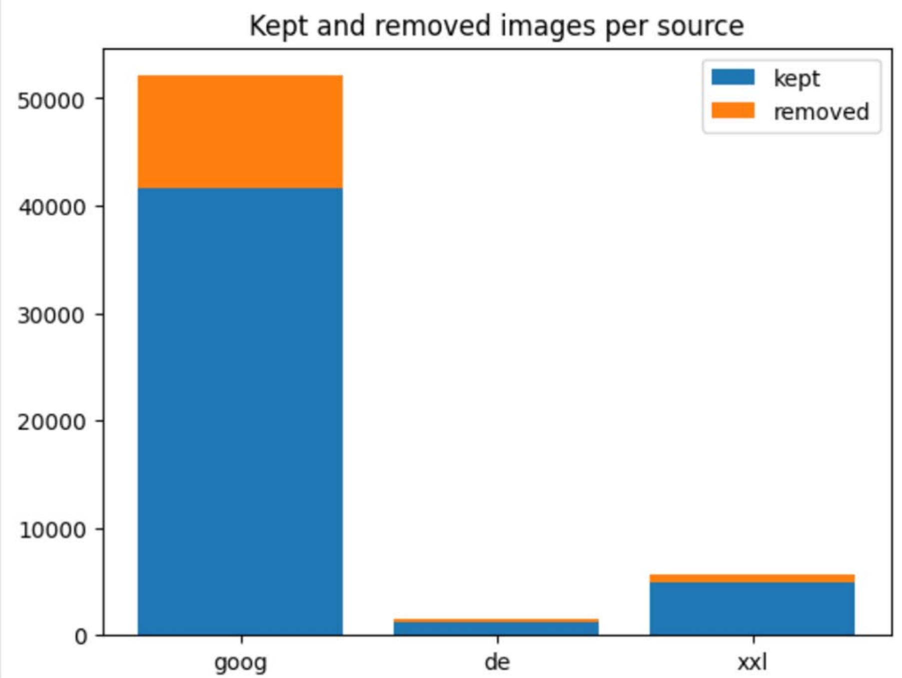


-> Choose threshold 1 , remove 3989 images, keep 44836

3.4. Is there bias in the images?

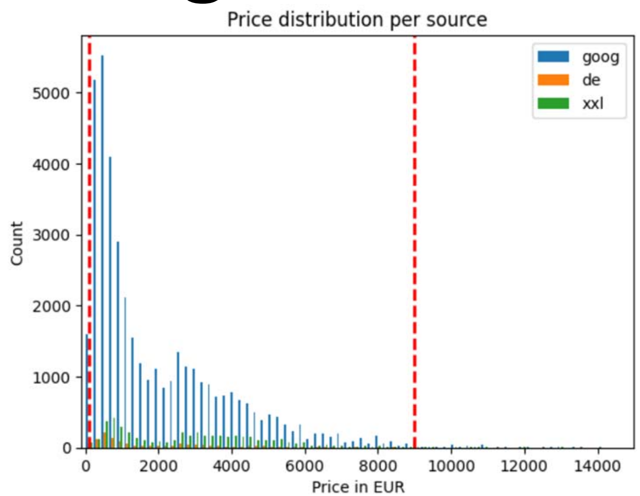
- Many duplicate images (Google)

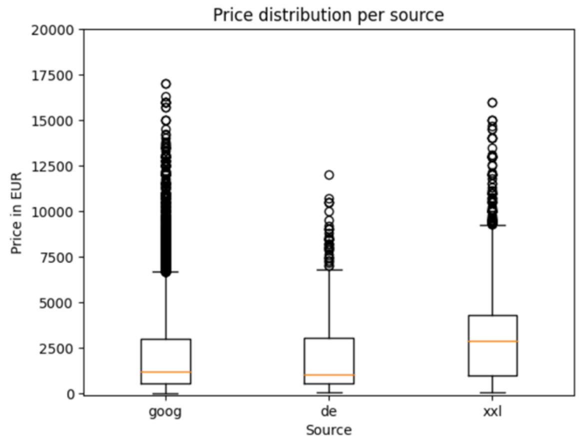
```
Kept image-count: [41594, 1278, 4887]
```



3.4. Is there bias in the images?

- Price distribution per source
- The bicycle-sellers tend to have higher priced bicycles
- Generally, the number of images from Google outweighs this fact
- Bicycles priced < 100 and > 10000 are removed



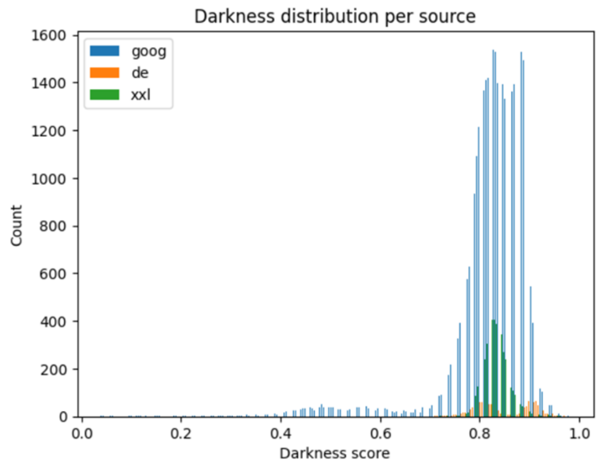


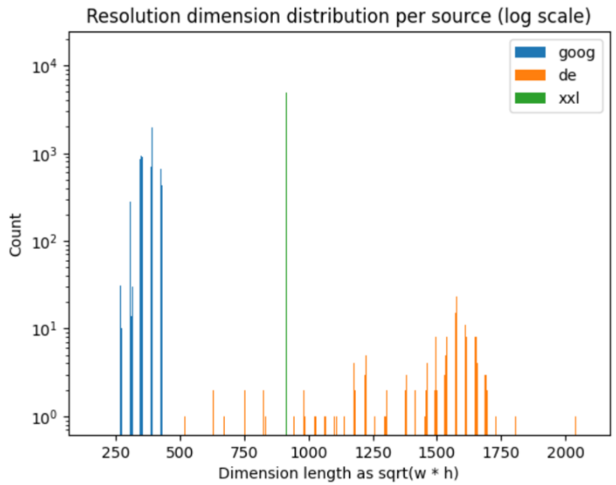
3.4. Is there bias in the images?

- Darkness score is the

"average greyscale pixel value"

- Fahrrad.de shows a deviating distribution but this is not further pursued





3.4. Is there bias in the images?

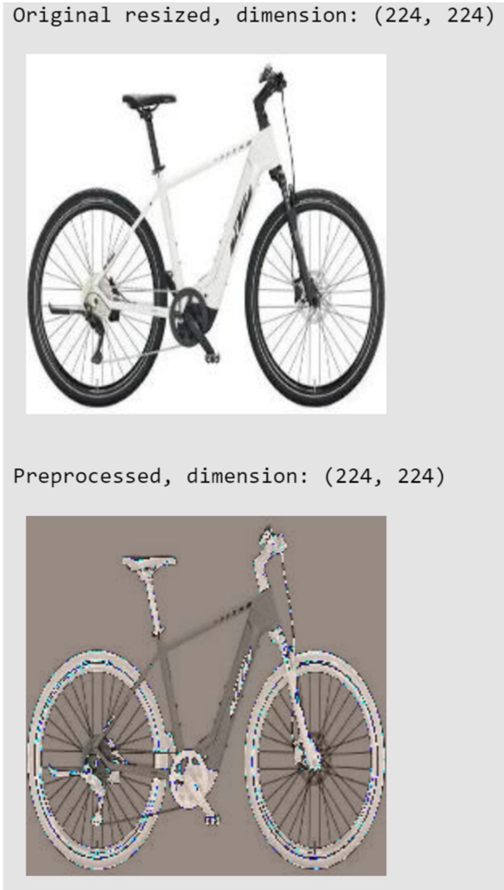
- Linear correlation is calculated
- As the price distribution looks similar to log-normal, this is explored
- Price is most linearly correlated with the square root of resolution and darkness (0.022 and 0.069)
- No large correlation can be determined

	price	resolution	darkness
price	1.000000	0.049333	0.017014
resolution	0.049333	1.000000	0.091306
darkness	0.017014	0.091306	1.000000

	price	sqrt_resolution	sqrt_darkness
price	1.000000	0.069241	0.022039
sqrt_resolution	0.069241	1.000000	0.098316
sqrt_darkness	0.022039	0.098316	1.000000

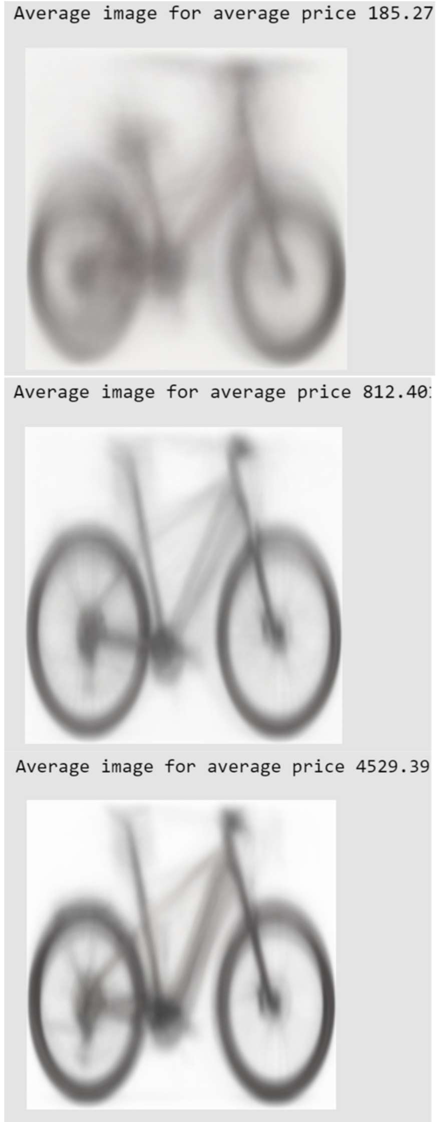
3.5. Preprocessing

- Images are rescaled and preprocessing applied (for ResNet)
- Cropping of images (Not part of the NB)
- Visualization immediately before training



3.5. Mean Image

- Compute the mean image for different price ranges
- Bicycles are generally right facing
- Most frequent shape seems to be “Mountainbike” (Non-horizontal mid-section)



3.5. Datasplit

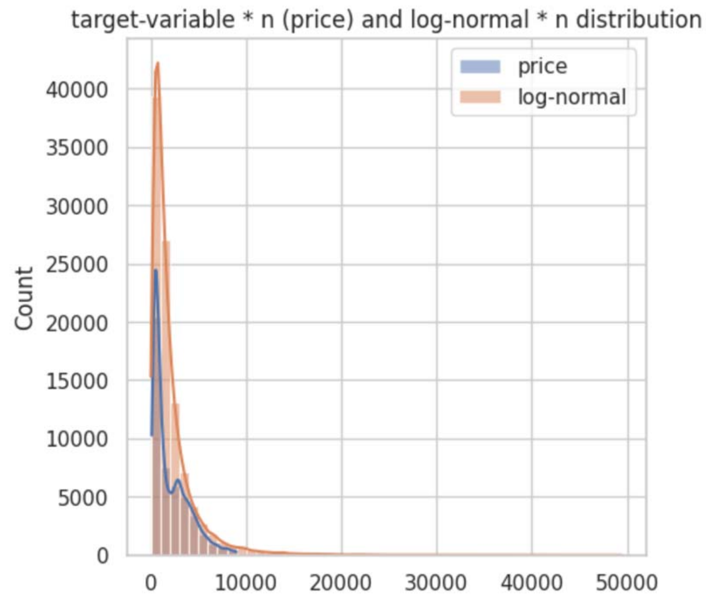
- Random Shuffle of all 3 sources,
  - Google
  - Fahrrad.de
  - Fahrrad-xxl.de
- Train = 45000
- Dev =   1000
- Test =   1000 (Originally hp. optimization was intended)
- Split is permanently saved and loaded via .json

4. Loss function

- The algorithm will minimize the loss function
- The choice of loss function has to match the problem to be solved
- Otherwise unsatisfactory results will be obtained

4. Loss function

- RMSE: $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$
- MAE: $\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$



- MAPE: Mean Absolute Percentage Error

$$\frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

4. Loss function

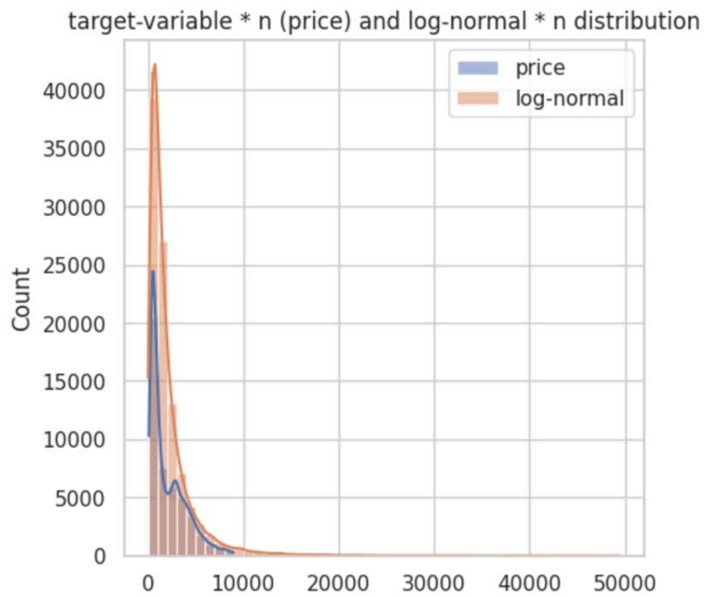
- RMSE: The tail, high priced image, would be over-emphasized
- MAPE: „Scaled MAE“ is more sensitive to small prices

y = 200, ŷ = 300, MAPE = 50%

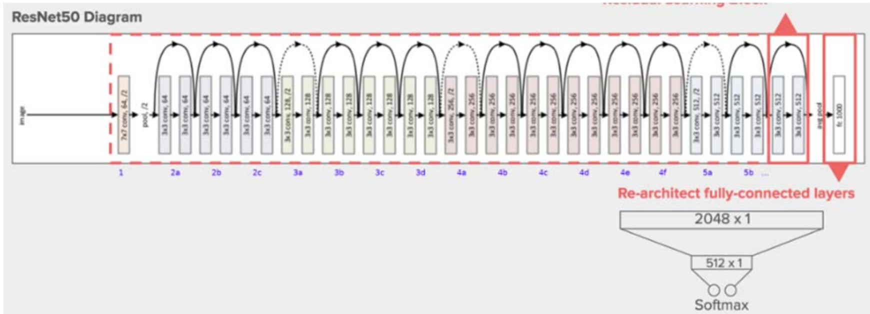
y = 2000, ŷ = 2100, MAPE = 5%

The original problem: Reselling Profit:

- Allows reselling cheap bicycles as well
- Allows for limited capital inve



5. Model training: Baseline



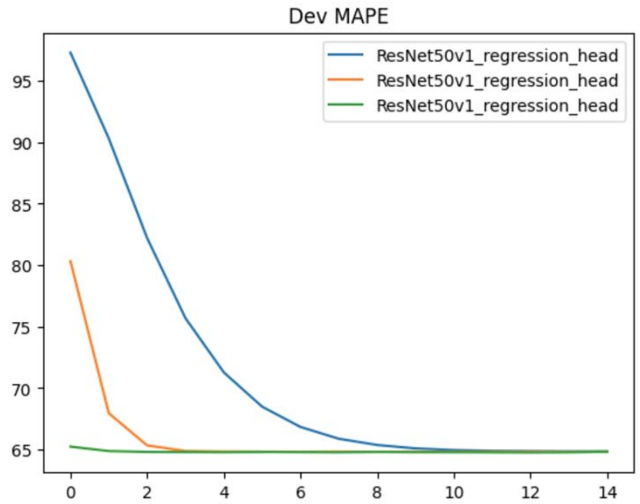
- The model used is a 50-block ResNet

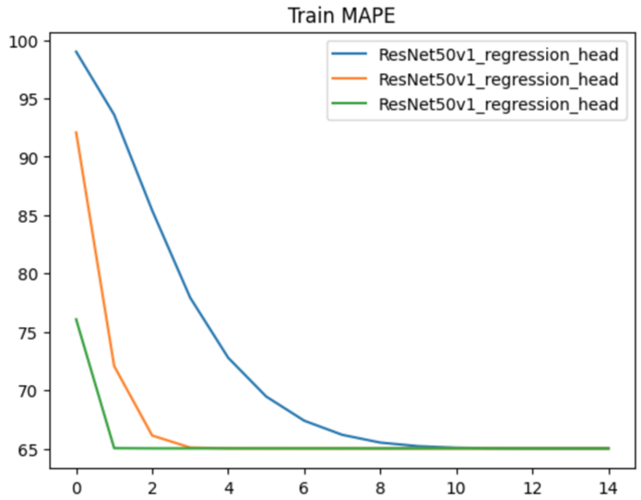
Regression Head:

- Hidden: 1024 Fully Connected Neurons
- Output: 1 ReLu/Linear Neuron

5. Model training: Baseline

bs: 32
lr: [
    0.0001,
    0.0003,
    0.001
]
epochs: 15





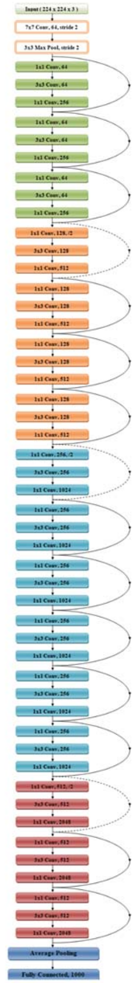
The baseline model can not achieve good performance,
MAPE = 65%
on average: 400€ or 1600€ for a 1000€ bicycle

5. Model training: Conv5-Model

- The model used is a 50-block ResNet

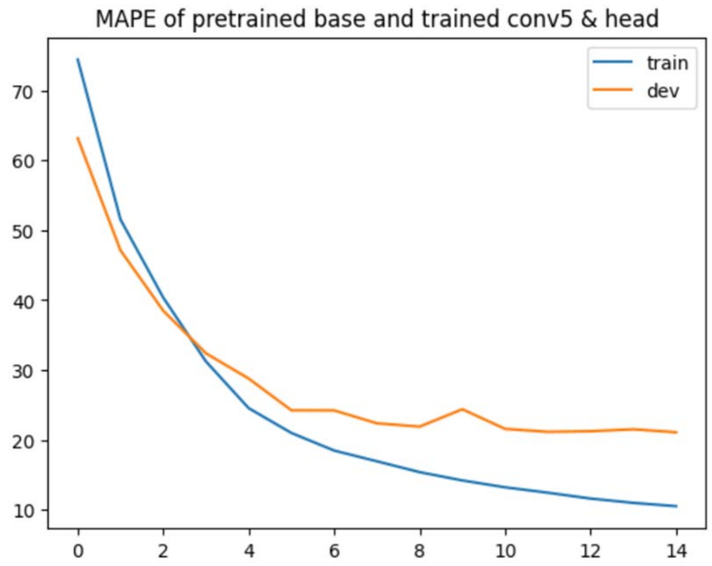
Regression Head:

- Hidden Layer: 1024 Neurons
- Output: 1 ReLu/Linear Neuron



5. Model training: Conv5-Model

bs: 32
lr: 0.0001
epochs: 15



The performance is better, MAPE = 21%
Prediction on average: 800€ or 1200€ for a 1000€ bicycle
Improvements: Hyperparameter Tuning, simply using ResNet V2

5. Model training: Conv5-Model

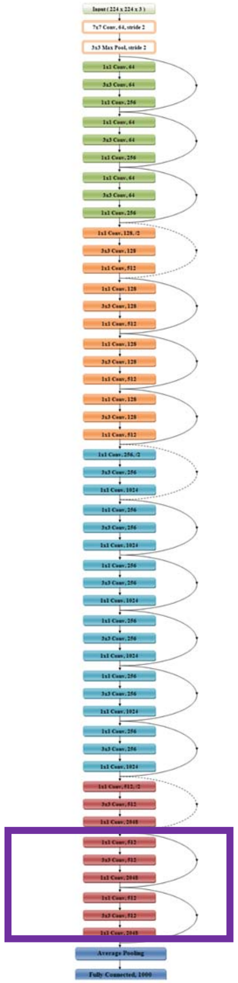
- The model used is a 50-block ResNet

Retraining:

- The last 6 of the total 53 convolutional layers

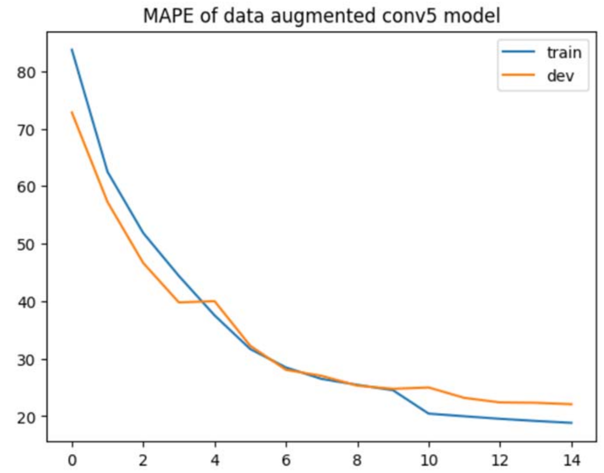
Regression Head:

- Output: 1 ReLu/Linear Neuron



5. Model training: Robust Conv5-Model

- The data is augmented by:
  - Flip
  - Rotation: 2.5 degrees
- MAPE = 22%



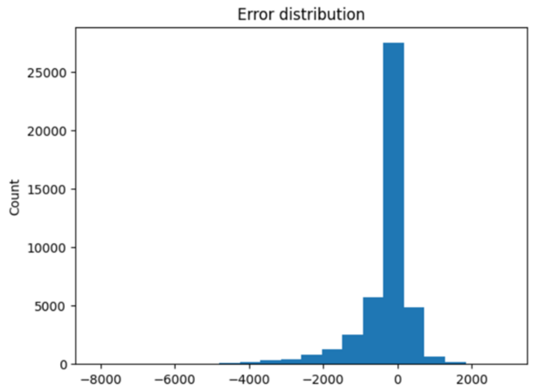
- Comparing MAPE

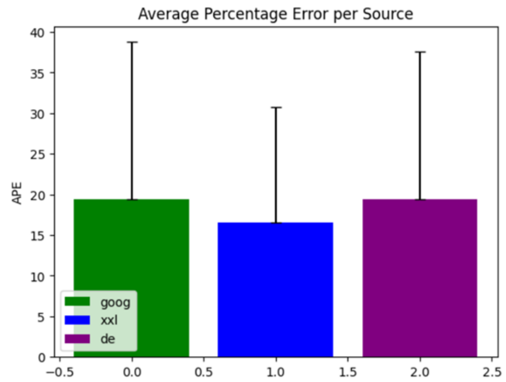
	Baseline	Conv5	Conv5 aug.
Dev	65%	21%	22%
Aug. Test	-	37.9%	22.5%

5. Recap: “Is there bias in the data?”

```
errors = predictions.reshape(-1) - np.array(train_prices)
```

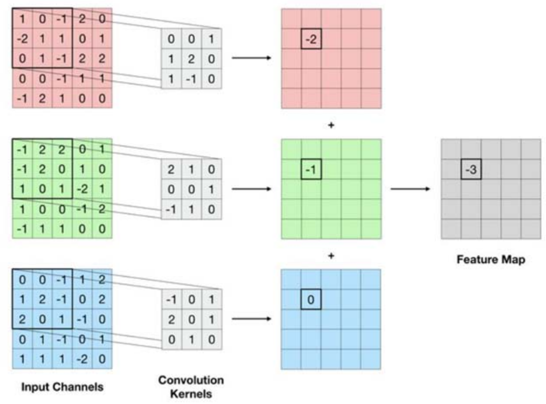
- Error approximately normal distributed
- No significant error difference between the sources





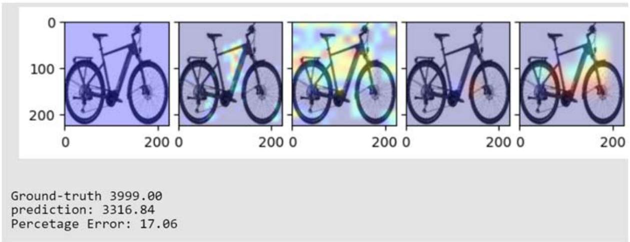
6. Visualization of activations

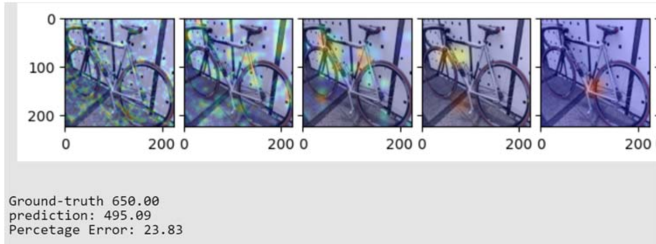
- Gradient Cam:
  - Gradient of
    1. the predicted output with respect to
    2. the output-featuremap of
    3. each convolutional layer

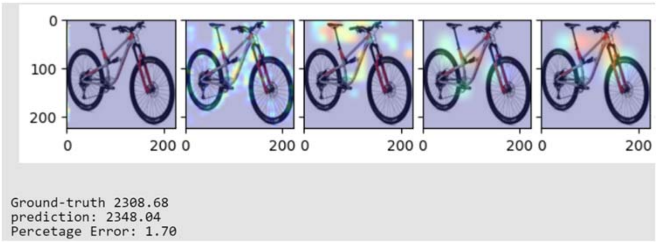


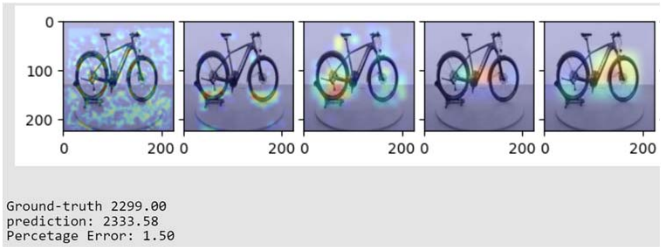
6. Visualization of activations

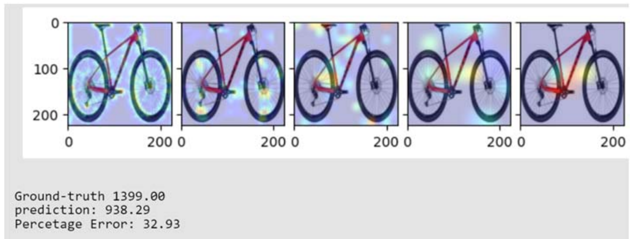
- The algorithm generalizes:
  - unclean images
  - “non-sideview” images as well
- Grad Cam is not 100% exact in locality (resolution is the feature map size of the respective Conv. Layer)











6. Visualization of activations

- The algorithm picks up on some random white background pixelation

-> train-dev-test data-leakage

- Only some images are affected
- Only some layers are affected
- The overall impact is not investigated

Thank you for your attention

Are there any questions, feedback or suggestions?