

Notes to Image Processing and Modeling

Contents

1	Strategy	1
2	Webscraping	1
3	Preprocessing	2
4	Model	2
5	Results	3

1 Strategy

- Use number of available images for each apartment as well as price predictions solely based on the image *contents* as numeric features for the final model

2 Webscraping

- Dataset contains link to websites of each listing
- Use the `requests` library to get HTML Source Code of each website
- Use the `beautifulsoup` library as HTML parser to find and extract embedded weblinks to all images located on this front page
With this strategy we could extract the first 5 images (if 5 or more were

available) that could be directly accessed from the front page source code

- Use `requests` again in combination with the `pillow` library to decode the content of all image addresses to two dimensional images which serve as input to a Convolutional Neural Network

3 Preprocessing

- In contrast to classification tasks where *Data Augmentation* is commonly used in order to expand the training set and improve generalization, this approach is not immediately transferable to a regression context since we have to guarantee that the label (i.e. the price) remains unchanged for each image transformation
- Thus, we decided against rotating the images or manipulating the color composition
- We **did** use image *cropping* however, which is in our opinion one of the few reasonable augmentations for regression
- After resizing all images to 256×256 pixels we randomly cropped a square area of 224×224 out of each image in the training set and cropped an equally sized area out of the image **center** in the validation set to avoid any randomness during inference
- At a final step all images were normalized for each color channel separately with the same values that were used during training on **ImageNet**. The mean values and standard deviations for each color channel are provided in the **PyTorch** [documentation](#)

4 Model

- Strategy: Fit one small custom CNN as benchmark model and one large pretrained Model with some custom final layers as main model
- Idea: The large model is pretrained on **ImageNet** and thus capable of extracting common features, use these features as input to custom final

layer(s) that output a price prediction, i.e. the last layer directly maps to a single node

- Potential Issue: If the pretrained network is very deep, the learned features before the final layer could be very specific to the *Output Classes* of **ImageNet** and not generalize well to our images. Some possible options:
 - Out of the collection of available very large pretrained models, choose one that is not extremely deep:
We chose **ResNet18** with roughly 11 million parameters
 - Do not freeze the weights of the pretrained model completely but fine tune them (i.e. modify the weights by backpropagating through the entire network):
We did not investigate this option further due to its high computational cost
 - Cut the pretrained model before the last layer (with the hope that at this point very generic and widely applicable features of images are extracted which therefore generalize better) and append the custom output layer
This option did not improve our results significantly
- It turned out that a single custom layer mapping from 512 to a single neuron was not expressive enough
- The performance improved slightly by adding a Fully Connected Network with three layers and *ReLU* activation functions at the end

5 Results

- Using only the content of the available images, the pretrained **ResNet18** achieved a Mean Absolute Error of 579 NOK (approx. 58 Euros) on the Validation Set
- The 'Null' Model of always predicting the mean price achieved an *MAE* of 630 NOK without a log-transformation of the price and a *MAE* of 569 NOK with a log-transformation, so the predictive power of the images alone was very small

- However, the correlation with the CNN predictions with the true price was 0.41:
This indicates some limitations of the correlation as useful metric on the one hand but positive tendencies of the CNN predictions on the other hand
- In fact, the network struggled the most with capturing the wide range of prices and almost always predicted values close to the center of the (log) price distribution
- Considering the difficulty of the task it is actually highly doubtful that humans could provide much more accurate predictions
- Show **Figure** of sample images with true and predicted price