BACHELORTHESIS
Eric Schaefer

Quality Classification and Evaluation of Bicycles Based on Images Using Deep Learning Approaches

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

HOCHSCHULE FÜR ANGEWANDTE
WISSENSCHAFTEN HAMBURG
Hamburg University of Applied Sciences

Eric Schaefer

Quality Classification and Evaluation of Bicycles Based on Images Using Deep Learning Approaches

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Bachelor of Science Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuende Prüferin: Prof. Dr.-Ing. Marina Tropmann-Frick
Zweitgutachter: Prof. Dr. Peer Stelldinger

Eingereicht am: July 19, 2021

Eric Schaefer

Thema der Arbeit

Automatische Bewertung von Fahrrädern mithilfe von Deep Learning und Zustandsbildern

Stichworte

Deep learning, Bilderkennung, Convolutional Neuronal Networks (CNN), Qualitätsbewertung von Fahrrädern, Region Based Convolutional Neuronal Networks (R-CNN), Objekterkennung, Semantische Segmentierung

Kurzzusammenfassung

Mit zunehmender Zahl von Online- und Gebrauchthändlern hat eine zuverlässige Bewertung der Qualität dieser Fahrräder und ihrer Preise an Bedeutung gewonnen. In dieser Arbeit konzentrieren wir uns auf die Bewertung von Fahrrädern anhand von Bildern ihres aktuellen Zustands mittels CNNs und semantischer Segmentierung. Die experimentellen Ergebnisse zeigen, dass die semantische Segmentierung hilfreich ist, Fahrräder vor der Auswertung in verschiedene Teile und Kategorien zu unterteilen. Die Bewertung mit diesem Ansatz ist praktikabel.

Eric Schaefer

Title of Thesis

Quality Classification and Evaluation of Bicycles Based on Images Using Deep Learning Approaches

Keywords

Deep learning, Visual recognition, CNN, Bike Quality Assessment, R-CNN, Object detection, Semantic Segmentation

Abstract

As the number of online and second-hand dealers increases, a reliable way to evaluate the quality of these bicycles and their prices has gained importance. In this work, we concentrate on the evaluation of bicycles based on pictures of their current condition using CNNs and semantic segmentation. The experimental results show that using semantic segmentation, separating bicycles into different parts and categories before the evaluation is helpful. The assessment using this approach is workable.

Contents

List of Figures

List of Tables

Accronyms

AI Artificial Intelligence.

AMT Amazon Mechanical Turk.

AP Average Precision.

CNN Convolutional Neuronal Networks.

FPN Feature Pyramid Networks.

ILSVRC ImageNet Large Scale Visual Recognition Competition.

R-CNN Region Based Convolutional Neuronal Networks.

RoI Region of Interest.

RPN Region Proposal Networks.

SSD Single Shot Multi-box Detector.

1 Introduction

1.1 Motivation

In recent times, the number of people who use bicycles or are interested in obtaining one has risen because of climate change concerns and traffic jams in the city center of bigger cities. Additionally, the number of online shops and online second-hand dealers has increased as well. Especially with online second-hand dealers, it is difficult to make a grounded decision on whether the price for the bicycle is reasonable. To help people and dealers make a grounded decision on the price or price range for the bicycle, it would be helpful if there is an automatic way to get this information based on the image of the bicycle in question.

Since there have been many advances in image and object recognition and object classification with deep learning, it would be interesting to know if we could create a model with deep learning to solve this problem. Deep learning is used with great success in similar situations, such as damage assessment of buildings in areas of natural disasters, as well as a helper in evaluating damage to cars for insurance. Hence, it would interest if we could use deep learning to assess the quality of a bicycle.

1.2 Goal

This work has two goals. The first one is creating a deep learning model that can evaluate the quality of a bicycle. The second goal is to check how good semantic segmentation works on images with bicycles and whether using semantic segmentation is helpful for the quality assessment of the bicycle. For example, it would be interesting to detect the bicycle in an image and highlight it from the background and whether this has any advantage on the quality assessment. Additionally, we want to try using semantic segmentation to detect certain bicycle parts and focus the quality assessment only on these parts.

1.3 Structure

The rest of the bachelor thesis is structured as follows.

In chapter 2 we present a review of related works, primarily based on the quality assessment and deep learning methods we want to use.

Chapter 3 explains the proposed method of our model, which tries to assess the quality of the bicycles.

Chapter 4 contains information on the dataset, explains the preparation of the dataset for the experiments, describes the experiments, and evaluates and discusses the results of the experiments.

Chapter 5 concludes the thesis and contains the conclusions. In addition, it gives an outlook on things that we could improve.

2 Previous Work

The quality assessment of objects using deep learning is a topic that garnered much attention in recent years, mainly due to the advancement in object classification and object recognition, and semantic segmentation. In the section 2.1, we will look into different fields where quality assessment is already used. We will mainly review the ways that have been used to solve the quality assessment. In the section 2.2, we check some of the datasets for object classification and look into the CNN method to solve object classification. In the section 2.3, we cover semantic segmentation datasets and algorithms that segment these datasets semantically, especially Faster R-CNN with RPN and FPN.

2.1 Quality Assessment in other Fields

In recent years, many fields have already experimented with an automatic assessment of images using neuronal networks to simplify the evaluation of pictures. Examples of areas in which this has already been tried are building damage assessment [17] and car damage analysis of insurance [1]. In both these examples, the insurance inspector has to go through several images to assess the damage to the object in question. Because of this, it is beneficial to get automatic suggestions beforehand.

Nia [see 17, p. 20-23] proposed three different feature streams to better assess the damage to buildings by natural disasters by only using postevent images (see Fig. 2.1). Nia uses three streams to get more and different information extracted from the pictures. The first stream is a color image feature stream to analyze the raw input data. The second stream is a color mask feature stream, used to remove the effect that the background or other factors like camera angle and camera position have by using semantic segmentation to detect the house and extract it from the image. The third stream is a binary mask feature stream that uses only a binary channel instead of the three RGB channels used by the colored streams. The usage of this stream is to learn the shape of the building. Finally, Nia's model combines the results of all three streams during the regression process to evaluate how severe the damage is.

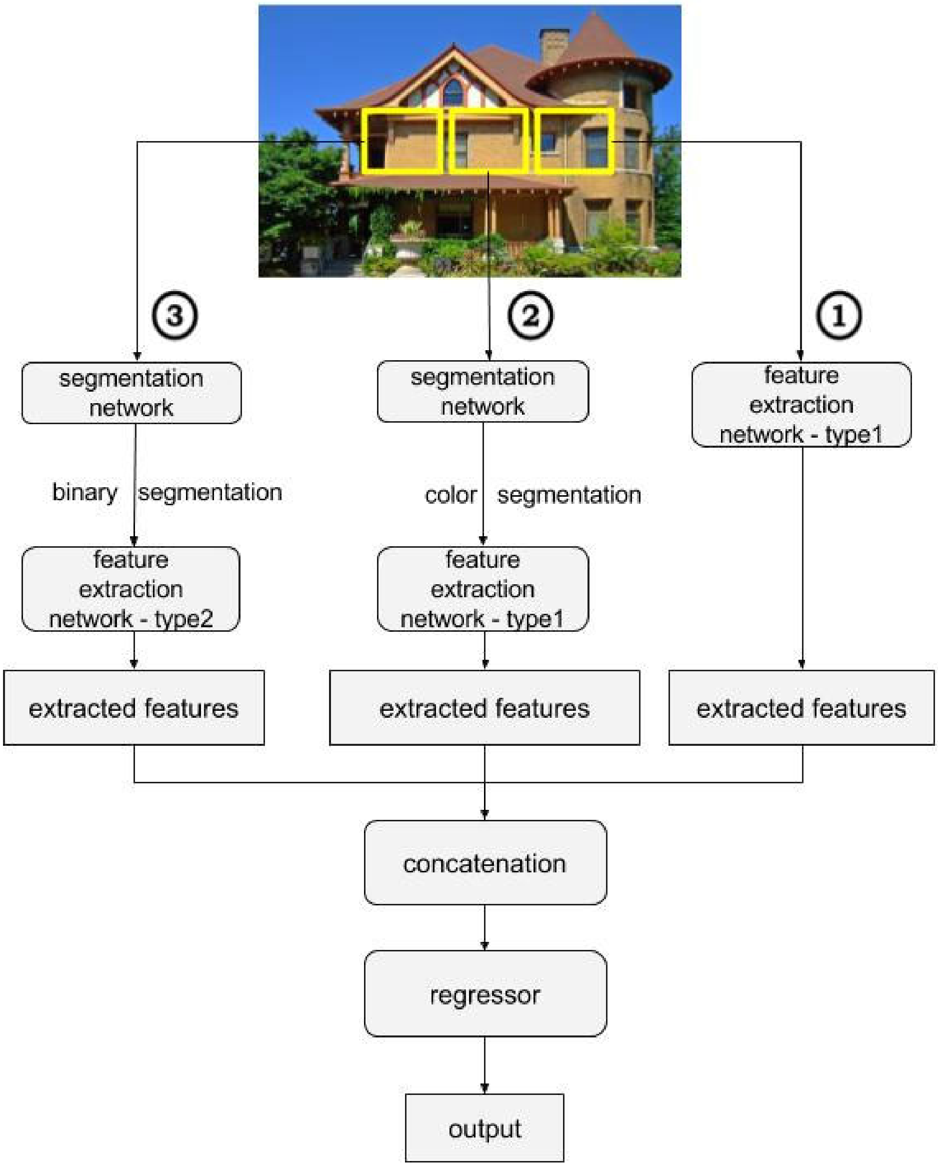

Figure 2.1: Overview over the model by Nia [17]. (1): Color image feature stream. (2): Color mask feature stream. (3): Binary mask feature stream.
Source: Nia [17, p. 19] (Copyright © 2017, SIMON FRASER UNIVERSITY).

Artan and Kaya [see 1, p. 316-317] realized during their research that they could divide the car into different areas that are more likely to have different kinds of damage. Because of this, the approach was to at first use object recognition to detect certain parts of the car and then analyze and categorize the damage that this part has taken. The advantage, therefore, is that it is easier to compare the damage costs of a smaller section of the car than the whole. Thus, we get more specific results. They also examined whether Faster R-CNN or Single Shot Multi-box Detector (SSD) shows better results and concluded that Faster R-CNN is better than SSD for this kind of task [see 1, p. 318].

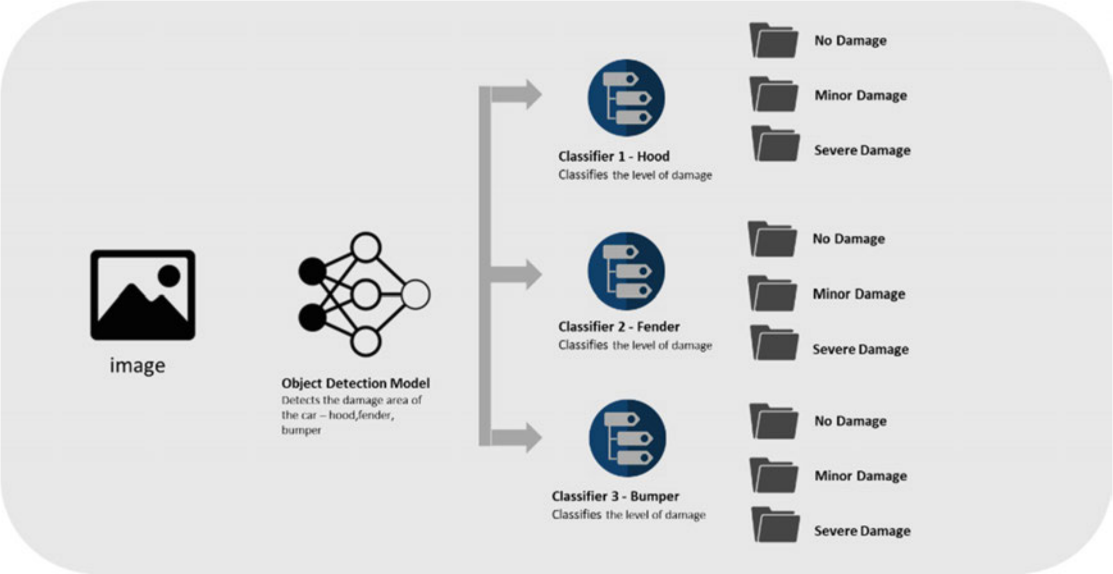

Figure 2.2: Damage analysis model framework by Artan and Kaya [1].
Source: Artan and Kaya [1, p. 317] (Copyright © 2020, Springer International Publishing).

2.2 Object Recognition and Classification

Many algorithms focus on object recognition and classification, but in recent years the advancements due to neuronal networks have led to a stronger focus on them, mainly CNN. In this work, we will focus on CNN and consider the bike as an object and get the quality of the bike by classifying it. This section will focus on already existing datasets for object classification and explain how CNN works.

2.2.1 Object Classification Datasets

Object classification distinguishes between Small and Large-Scale Image Datasets. Our solution can work with both small and large datasets. However, due to restrictions in graphics card memory and concerning simplicity, we will only work with a small dataset. In the following section, we will introduce some Small and Large-Scale Image sets that are already well labeled.

Small Image Datasets

We use Small Image Datasets most commonly for training and evaluation benchmarks because we get faster results than when we run an algorithm on a Large-Scale Image Dataset. After all, the Small Image Dataset has fewer images and can run more training epochs than with a larger dataset. The Caltech101 Dataset [14] contains 101 object categories with 40 - 800 images per category. It chooses the categories randomly and downloads the pictures using the Google Search Engine. Similar to Caltech101 is the Caltech256 Dataset [8], which contains 256 classes instead of 101. Each class has between 80 - 800 images. Therefore, the number of images per category has at least doubled. The MSRC [21] contains 591 images of 21 object classes and is already randomly split into 45% training, 10% validation, and 45% test sets.

Large-Scale Image Datasets

With the advances in computer vision research and computer hardware over the past decade, we need larger datasets to improve the development and testing of complex object classification algorithms. These advances focus in particular on GPU Performance and GPU Memory.

MNIST [13] is one of the older and better-known large-scale image datasets and contains single-digit handwritten numbers. These numbers are divided into a 60,000 image training set and a 10,000 image test set. It is one of the most commonly used datasets since it is easy to use for learning and testing, and there exist many published results which used different algorithms for learning. Thus, many algorithms have been tested on MNIST, and the results are accessible to check and compare.

ImageNet [4] is organized according to the WordNet [18] hierarchy. That means that the object categories are based on synonym sets of synsets. The goal of ImageNet is to provide 500-1,000 images on average per synset. These images are gathered through several search engines, using the WordNet synonyms and their translation in different languages as a query. The resulting images per category are then verified by several humans using the services of Amazon Mechanical Turk (AMT).

2.2.2 Object Recognition Methods

Object recognition enables Artificial Intelligence (AI)-Systems to detect and identify objects in images or video. It is part of the fields of robotics, machine learning, and computer vision. While object recognition is relatively easy for humans, it is a challenge for a computer system. That is because humans will look at the whole image and make their decision based on the detected structures, while computer vision systems can only base their decision on a certain number of pixels. Because of this, the viewpoint, the illumination, the scale, and the occlusion have a high impact on object recognition. In this section, we will review some of the deep learning methods that are used in object recognition and classification.

Variations of CNNs are currently the most used deep learning method in computer vision applications. These applications include classification and regression. The construction of CNNs follows a specific pattern: convolutional layers are followed by pooling layers and the whole application ends with fully connected layers. LeNet-5 [13] is one of the simplest examples of a network that follows this pattern, while ResNet [9] is one of the most complex structures up-to-date.

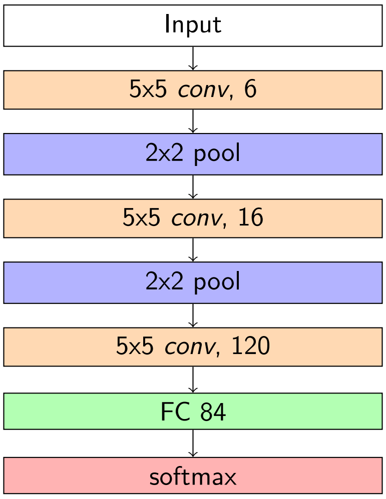

Figure 2.3: LeNet-5 Architecture

LeNet-5 (see Fig. 2.3) was developed by Lecun et al. [13] for better results learning the MNIST-dataset. It was the first try on convolutional networks and had an accuracy of 92%.
AlexNet [12], which was the winner of the ImageNet Large Scale Visual Recognition Competition (ILSVRC) [20] in 2012, popularized ConvNets in computer vision. AlexNet comprises five convolutional layers, some of which are followed by max-pooling layers. The three fully connected layers use dropout and data augmentation to reduce overfitting in the fully connected layers. The network finishes with a Softmax layer. AlexNet comprises around 60 million parameters. That makes it one of the most complex ConvNets to that date. In the following years of the challenge, the structure of AlexNet was further improved and used as inspiration by other researchers.
VGG [22], a participant in the ILSVRC 2014, used even more convolutional and fully connected layers and showed that the depth and complexity of the networks have a significant impact on the performance. It contains 13 convolutional layers and three fully connected layers. The convolutional layers use small-size convolutional filters (3×3) and 2×2 pooling. VGG roughly has 140 million parameters. The Construction of a deeper network structure leads to a more complex and accurate model. However, it is prone to overfitting and vanishing/exploding gradients due to the higher number of parameters. This model also requires more powerful hardware devices to train and evaluate.

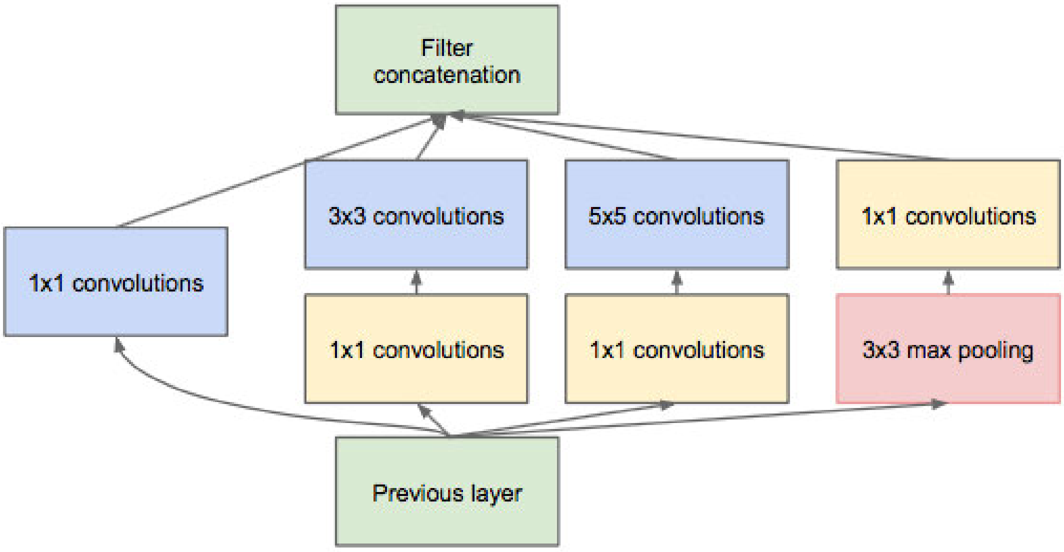

Figure 2.4: Inception module with dimension reduction by Szegedy et al. [23].
Source: Szegedy et al. [23, p. 5] (arXiv preprint arXiv:1409.4842, 2014).

Due to the exploding amount of parameters for deeper networks, GoogLeNet [23], the ILSVRC 2014 winner, developed a module called Inception (see Fig. 2.4). That reduced the number of parameters drastically to 7 million while simultaneously increasing the number of layers to 22. The beginning makes two traditional convolutional layers, while nine Inception modules follow in the deeper part of the network. Average pooling is used before the fully connected layer and Softmax layers. Each Inception module comprises two layers. The idea behind the Inception module is that even low-dimensional filters contain much information about an image. That allows the network to increase in width and depth of the Inception modules while simultaneously controlling the computational complexity of the network.

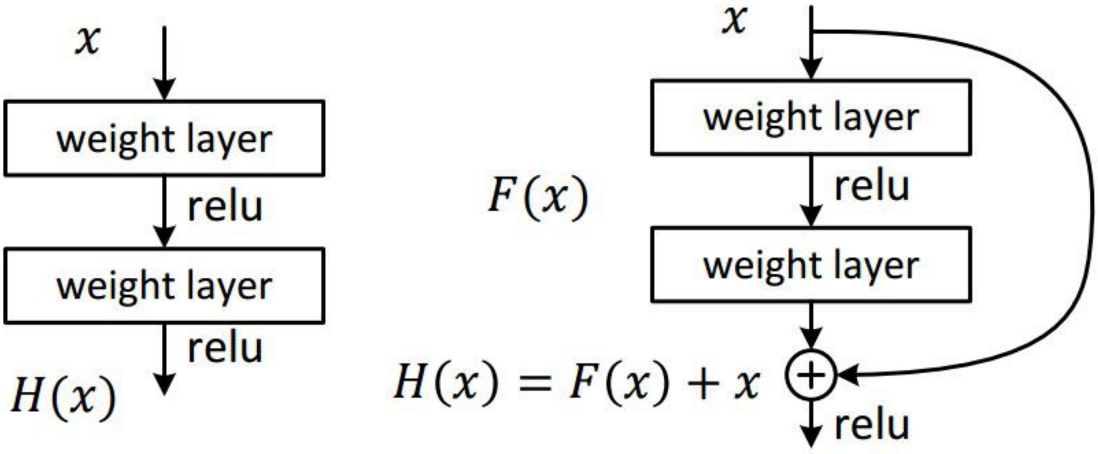

Figure 2.5: Residual learning. left: plain net, right: residual net [9].
Source: Nia [17, p. 11] (Copyright © 2017, SIMON FRASER UNIVERSITY).

Residual networks (ResNet) [9], the winner of the ILSVRC 2015, are the best performing networks in multiple computer vision tasks. These tasks include image classification and semantic segmentation. The ResNet, used for ILSVRC 2015, contained 152 layers, which is 8x more extensive than VGG [22], but it still had a lower time complexity than VGG. Simply stacking more layers on each other and constructing a deeper network leads to higher training and testing errors since the accuracy gets saturated and degrades rapidly.To solve the degradation problem, He et al. [9] suggested not to rely on the desired mapping $H(x)$ but create their mapping $F(x) = H(x) - x$ called residual mapping, shown in Fig. 2.5. After adding two new layers to the ResNet, the residual mapping comes into play, and the resulting and desired mapping will be $H(x) = F(x) + x$, with $F(x)$ resulting from the two layers and x being the original input for these two layers. This so-called shortcut connection does not add to the time complexity nor the parameters of the network.

Additionally, ResNet utilizes batch normalization [10]. Batch normalization improves the regularization, accelerates the training process, and makes the model less sensitive during the initialization process of the network.

2.3 Semantic Segmentation

The difference between object recognition and semantic segmentation is that object recognition recognizes the objects present in an image, while semantic segmentation classifies each pixel of the picture. Semantic segmentation classifies and segments a form at the pixel level, which makes it a challenging task. This work uses semantic segmentation to split the input image into pixels containing the bike and the background. It also uses semantic segmentation to extract specific parts of the bicycle for better classification. The following section focuses on semantic segmentation datasets and algorithms and models which solve them.

2.3.1 Semantic Segmentation Datasets

Many datasets have been collected and annotated for semantic segmentation. These datasets cover various types of object categories and provide a different level of detail quality. While we create our dataset for this work, we still use the annotation style of the Microsoft COCO dataset [15].

PASCAL VOC dataset [4] contains 500,000 images and 20 object classes used for classification, detection, and segmentation. Images for VOC2007 were obtained through the photo-sharing website Flickr[1]. For each object class, a set of keywords were used for the query on Flickr. Due to the usage of photos taken for personal interest instead of for computer vision research, Everingham et al. [5] call the dataset "unbiased". The annotation contains the class of the object, a bounding box that surrounds it, its viewpoint (front, rear, left, right, or unspecified), and its truncation. The truncation specifies whether the object is occluded or whether it extends outside the image. Some images also include pixel-wise information about segmentation, but until VOC2012, only 9,993 images include segmentation.

[1] https://www.flickr.com/

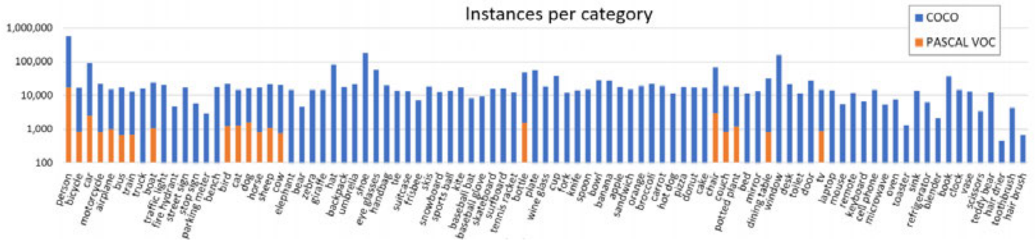

Figure 2.6: Comparison of number of annotated images per category for COCO [15] and PASCAL [5].
Source: Lin et al. [15, p. 7] (arXiv preprint arXiv:1405.0312, 2015).

Microsoft COCO dataset [15] contains 328,000 images of 91 objects. After including all categories of PASCAL VOC [5], children (between 4 - 8) were asked to name things they usually see. From the resulting 271 classes, 91 were chosen based on the usefulness and occurrence. The images were collected using Flickr and other image search engines, and for the annotations, AMTwas used. COCO has more instances and categories than PASCAL VOC (see Fig. 2.6) and fewer categories but more instances per category than ImageNet [12].

2.3.2 Semantic Segmentation Methods

Analogous to object classification algorithms, many algorithms solve semantic segmentation. Due to the strength of CNN, ConvNets are used to solve semantic segmentation tasks, especially VGG [22] and ResNet [9]. Additionally, there are also methods to accelerate the semantic segmentation to get real-time predictions. We will first introduce these methods and then follow up on the algorithms that use these methods.

Region Proposal Algorithms for Semantic Segmentation

The idea of region proposal algorithm for semantic segmentation is to get real-time results for semantic segmentation, so the detection must get faster. Region proposal algorithms are added as ‘attention’ [2] to the existing model.

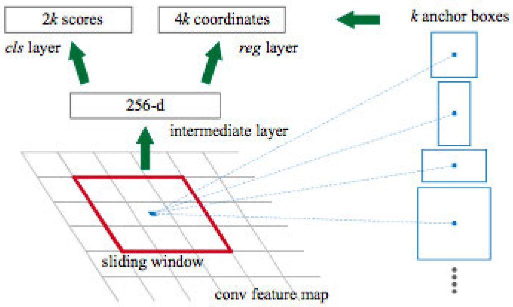

Figure 2.7: RPN
Source: Ren et al. [19, p. 4] (arXiv preprint arXiv:1506.01497, 2015).

RPN [19] was one of the first algorithms proposed. The RPN uses the last shared convolutional layer as its input and slides an $n \times n$ (in most cases 3×3) sliding window over each feature map. The resulting feature is forwarded to two sibling fully connected layers, one for box regression (*reg*) and one for box classification (*cls*). The fully connected level is used by all sliding windows.. Each sliding window location predicts multiple region proposals simultaneously, with the center of the window as an anchor. The maximum of possible proposals for each area is denoted as k. In the case of 3×3, this means we get nine anchors ($k = 9$). The *reg* layer returns two coordinates, the width and the height of the anchor box that means $4k$ values per sliding window, while the *cls* layer returns the probability for the class or the background, so $2k$ values. (see Fig. 2.7)

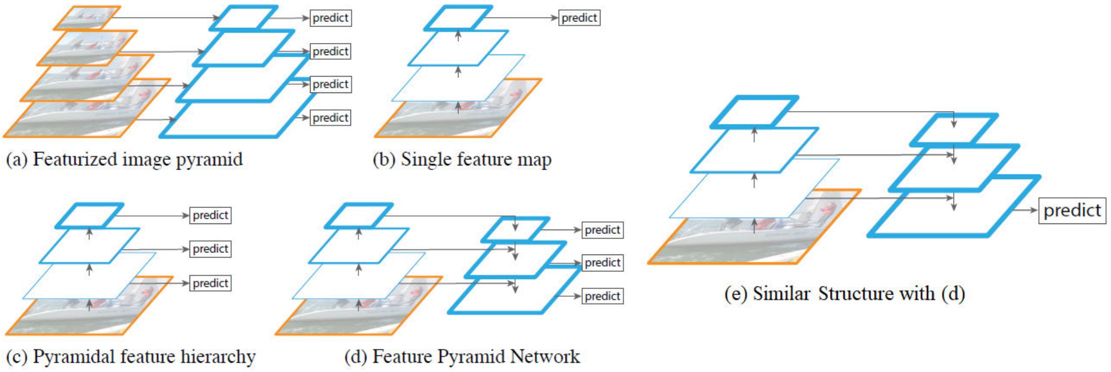
predict
predict
predict
predict
(a) Featurized image pyramid
predict
(b) Single feature map
predict
predict
predict
(c) Pyramidal feature hierarchy
predict
predict
predict
(d) Feature Pyramid Network
predict
(e) Similar Structure with (d)

Figure 2.8: Different architectures for Detection. (b) is used by RPN, (d) by FPN. Source: [24] (Copyright © 2019, Towards Data Science).

FPN [16] uses RPN as its basis, but instead of only using a single feature map (see (b) in Fig. 2.8), it uses a feature pyramid network (see (d) in Fig. 2.8). The bottom-up pathway of the FPN is a straightforward computation of a CNN, for example, ResNet [9]. If more than one convolutional layer with an identical resolution exists, only the last layer will be used as a pyramid level. For the top-down pathway, higher resolution features will be upsampled to have an equal resolution as the next level of the pyramid. That ensures that levels with a higher resolution are spatially coarser but semantically stronger. The results of each level will then be merged with the result from the level above. That is why both levels need the identical resolution.

Deep-Learning Methods for Semantic Segmentation

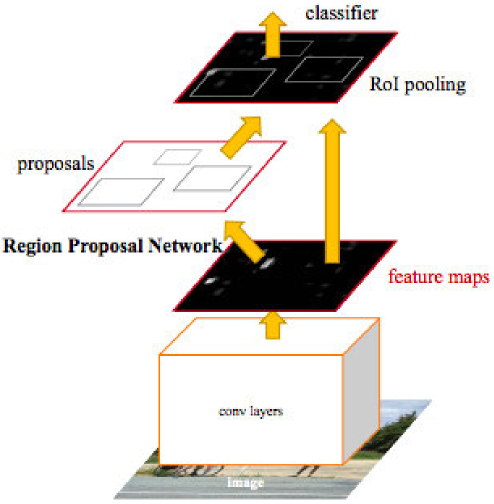

Figure 2.9: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the ‘attention’ of this unified network.
Source: Ren et al. [19, p. 3] (arXiv preprint arXiv:1506.01497, 2015).

R-CNN [7] was one of the first networks that extracted the bounding boxes and the features for the semantic segmentation in the same instance. Fast R-CNN [6] and Faster R-CNN [19] each improve R-CNNto get real-time results for semantic segmentation. Faster R-CNN is currently the best version of R-CNN adds a RPN as an ‘attention’ and Region of Interest (RoI) pooling layer to the network. The RoI pooling layer is used to transform all nonuniform inputs from the feature maps and the RPN to a fixed-size feature map.

An alternative to Faster R-CNN is Feature Pyramid Networks for Object Detection [16], using ResNet as the CNN and FPN as a RPN.

Dilated-C5 or Deformable ConvNet [3] is another algorithm for semantic segmentation. It uses deformable convolution and deformable RoI pooling. Deformable is based on the idea of augmenting the spatial sampling location with an additional offset. These offsets are self-learned without supervision. For an example of how deformable convolution works, see Fig. 2.10.

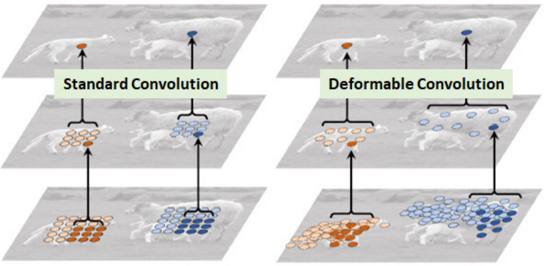

Figure 2.10: Standard Convolution (Left), Deformable Convolution (Right).
Source: Dai et al. [3, p. 4] (arXiv preprint arXiv:1703.06211, 2017).

3 Proposed Approach

Our goal in this work is to assess the quality of bicycles automatically. Several attempts have been made to classify the quality of an object using images as an input. For example, damage assessment on buildings with three different feature streams [17] or car damage analysis using object detection to identify parts of the car [1]. In this work, we will combine both the idea of multiple feature streams and object detection. Additionally, we will split the quality assessment into two parts. The first is to classify the bicycle into different categories since some categories have a higher value than others, despite being of lower quality. The second step then assesses the quality of the bicycles, but only using the bicycles of the same category as the reference.

Due to the recent advances in deep learning, we use multiple CNN and semantic segmentation networks to assess the quality of the bicycle. Our model comprises two steps, the first to categorize the bicycle, and the second to assess quality. Both levels of our model will use three feature streams for better results, and the three feature streams follow the same principle for both levels. Each feature stream represents a different attribute of the image data, which is significant for the quality assessment. These three pipelines all use one to two CNNs to classify the categories of the respective step from the input data.

The Color image stream employs a LeNet5 [13] like structure CNN to extract features from the raw input data. Color mask stream uses a semantic segmentation deep structure (R50-FPN [16]) to select objects of interest from the raw input image and then a LeNet5 [13] like structure CNN to obtain the classification from the segmented image. Since bicycles, unlike buildings, lack a simple to recognize structure, due to different angles of the picture, the third stream will not be a binary mask stream as proposed by Nia [17], but a color masked partial image stream. The color masked partial image utilizes a semantic segmentation deep structure (R50-FPN [16]) to divide the bicycle in the input data into different, for identification significant, bicycle parts. Due to the subdivision into partial images with meaningful information about the bicycle, it is easier to categorize them. For example, a Mountain-bike is easy to identify due to its more robust frame and wider tires, while a racing bicycle has a lighter frame and narrower tires. After having

extracted and analyzed the features of the three streams, we combine the results from the three streams to get one continuous value for each of the categories the current steps possess. An overview of the proposed model is illustrated in Fig. 3.2 and Fig. 3.3.
In the following sections of this chapter, we describe the behavior of each stream, its inputs, and its corresponding outputs. Section 3.1 focuses on the categorization of bicycles, section 3.2 on the assessment of the quality.

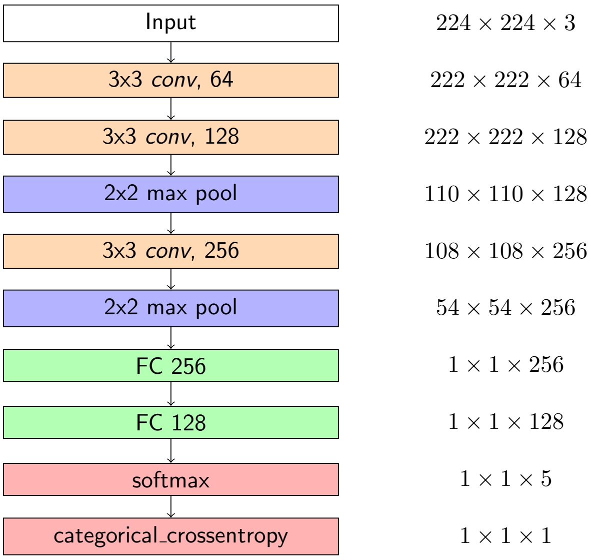

Figure 3.1: LeNet-5 [13] network structure with some modifications. The network contains 3 convolutional layers followed by 2 fully connected layers. A softmax and a categorical crossentropy loss layer are appended as classifier.

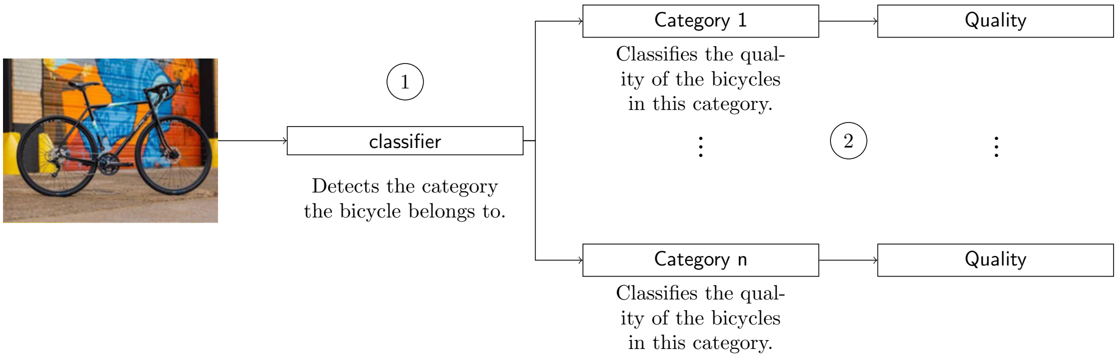
1
classifier
Detects the category the bicycle belongs to.
Category 1
Classifies the quality of the bicycles in this category.
Quality
2
Category n
Classifies the quality of the bicycles in this category.
Quality

Figure 3.2: Overview of out proposed model. (1): Classifier (section 3.1). Assigns categories to the bicycles. (2): Assessment (section 3.2). Assesses the quality.

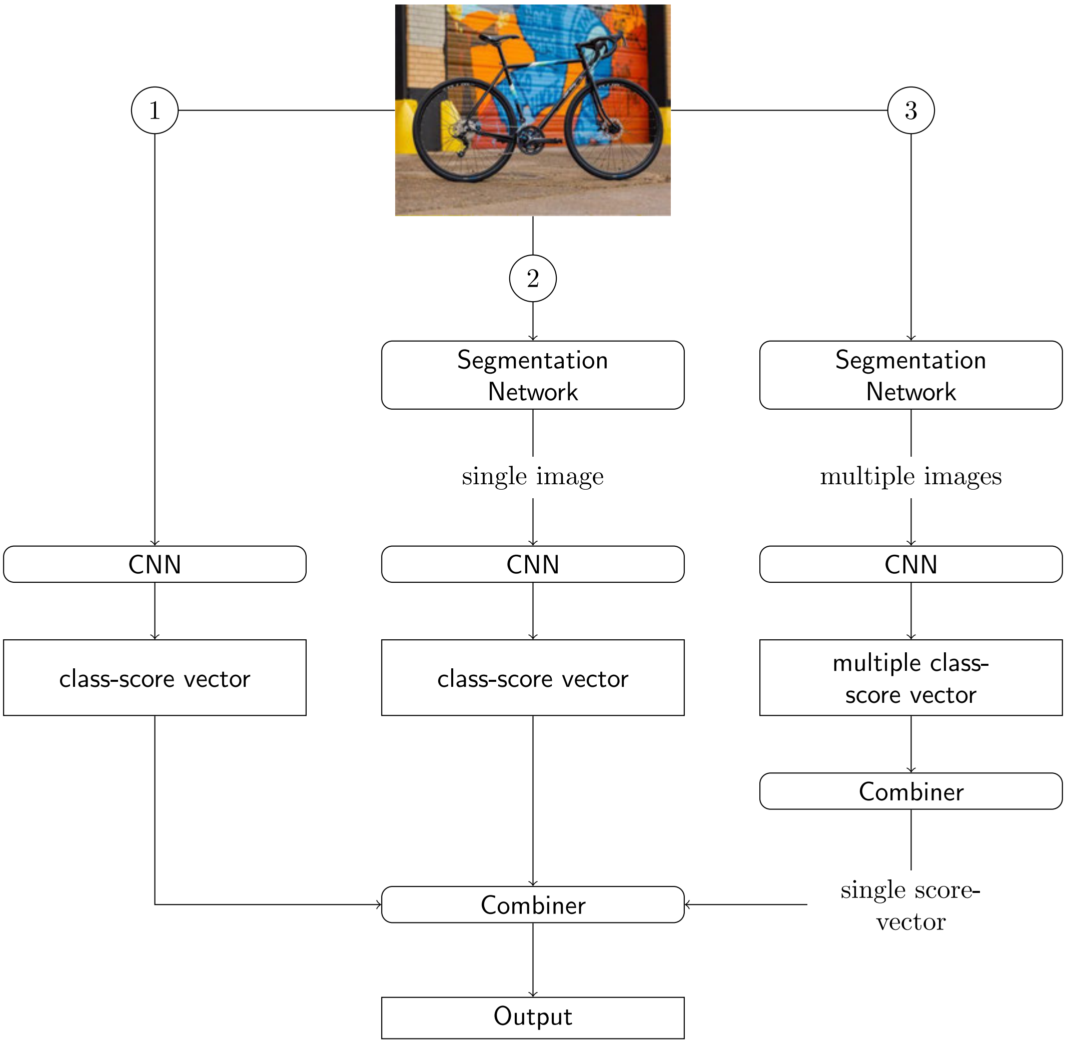

Figure 3.3: Overview of our proposed model for the two steps. (1): Color image stream (section 3.1.1). A LeNet5 like structure directly analyzing raw input image data. (2): Color masked stream (section 3.1.2). A LeNet5 like structure analyzing color masks of the image data. (3): Color masked partial image stream (section 3.1.3 and 3.2.1). A LeNet5 like structure employed on significant parts of the bicycle separately. The Combiner utilizes the class-scores and come to a combined result.

3.1 Classification

Since the price of a bicycle drastically differs depending on the category, e.g., E-Bikes are way more expensive than other types of bicycles, the first step of our model is to identify the type of bicycle that is assessed. The following section explains the three streams we use and the combination process of the results at the end of the step.

3.1.1 Color Image

The Color image stream, which is the first pipeline of the classification step of our proposed model. It is designed to analyze the raw input data and requires no semantic segmentation as a preprocessing step. The raw pixel values of the image are represented as $X(Width \times Height \times Channels)$ and given to several convolutional layers. The convolutional layer computes a feature map using equation 3.1. The parameters of the equation are the pixels of a small region of the input (X), the weights of the pixels (W), and a bias offset (b). An element-wise activation function ($ReLU$), shown in equation 3.2, will be applied to the output of the neurons. After some of the convolutional layers, a downsampling operation (*pooling*) will be additional connected.

$$z_j = f(\sum_i w_i x_i + b) \tag{3.1}$$

$$f(x) = \max(0, x) \tag{3.2}$$

The output of the additional computations after the last convolutional layer will then be transformed to a 1-dimensional array and used as the input of the fully connected layers. The fully connected computes the class probability distribution using equation 3.3. The parameters are a small part of the input values (X) and their respective weights (W). The fully connected layers also use $ReLU$ as its element-wise activation function but no downsampling operation.

$$z_j = f(\sum_i w_i x_i) \tag{3.3}$$

The last layer of the network is a classifier. That is also a fully connected layer, but it uses the Softmax operation, shown in equation 3.4, as an element-wise activation function. The loss function used is categorical cross-entropy, as shown in equation 3.5, where y_i' is the i-th scalar value in the model output and y_i the corresponding target value.

$$\sigma(x_j) = \frac{e^{x_j}}{\sum_{k=0}^{K} e^{x_k}} \tag{3.4}$$

$$\text{Loss} = -\sum_{i=1}^{\substack{\text{output}\\\text{size}}} y_i * \log(y'_i) \tag{3.5}$$

The complete network structure and the dimensions of each layer are illustrated in Fig. 3.1.

3.1.2 Color Masked Image

The second pipeline of the classification step of our proposed model is the color masked image stream. Since most people would not take a picture of their bicycle in front of a green screen to sell it, several visual factors could affect the performance of a vision-based system. This factor could be the sky, trees, and other objects in the background. Because these factors either slow down the learning process or even have counteractive effects on the accuracy. For example, if one category is always photographed with a clear sky and the others are not, then the most significant feature to recognize this category would be the sky. However, if we then test the model and input an image of this category without a clear sky, it would be classified wrongly. To address this issue, we consider utilizing semantic segmentation algorithms to help focus on the relevant regions of the input data. We propose to resize the image using the bounding box to cut away the parts of the picture that hold no significant areas. Additionally, we suggest to grey-scale the background of the bounding boxes and only leave the masked segment of the bounding box in its original color.

The color masked image stream uses a deep structure (R50-FPN [16]) to preprocess the input image and segment the image into the objects of interest as a mask and the rest as background. Given an input image, R50-FPN collects all predefined instances of bicycles and outputs those as foreground. A couple of output instances are shown in Fig. 3.4.

Similar to section 3.1.1, several convolutional layers with $ReLU$ as the activation function followed by a pooling layer and several fully connected layers with $ReLU$ as the activation function traverse the masked images. The model extracts the categories using a fully connected layer with Softmax as the activation function. Categorical cross-entropy is used as the loss function. The resulting category scores will be combined (section 3.1.4) with the results from the color image stream.

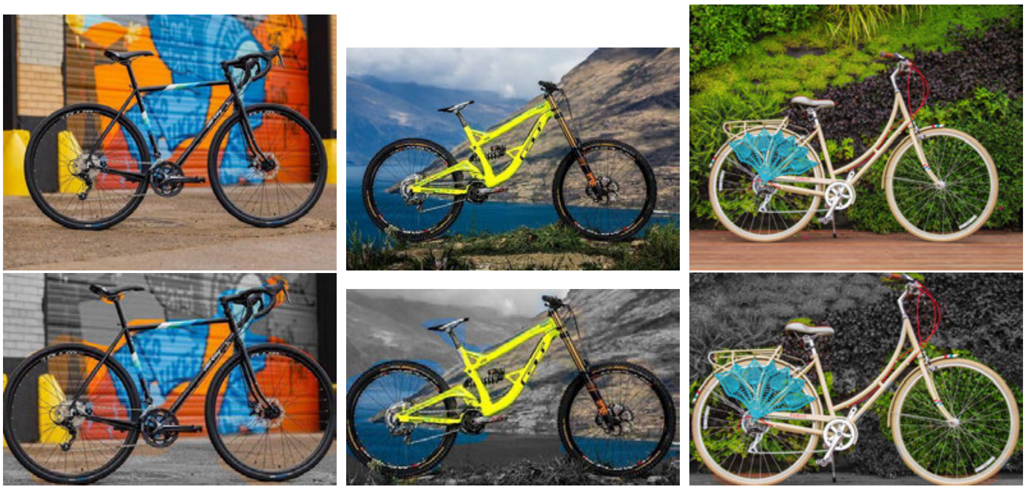

Figure 3.4: The first row: raw images which are used by the color image stream pipeline. Second row: corresponding color masked images as the output to the color mask pipeline. Instances of greenery and the sky are grey-scaled. Hence, the model is able to better focus on relevant parts.

3.1.3 Color Masked Partial Image

The third pipeline, the color masked partial image stream, follows the idea and model of the color masked image stream (section 3.1.2). A R50-FPN [16] is used to segment the input image in the preprocessing step, while a LeNet5 [13] like structured network for classification. The main difference between the color masked and color masked partial image stream is that the color masked partial image stream has more than one image as the output. It instead creates images according to regions of the bicycle that are significant for the classification (e.g., frame or tires). An example of the partial pictures that this will create can be seen in Fig. 3.5.
Not all of these partial image types will be utilized as a classification model since not all have a value in the categorization step (e.g., handlebar or saddle, because both are too similar in many different categories). The remaining partial images will afterward be classified by the CNN. The resulting category scores of all partial images will then be combined to one single category score, used as the result for the input image. This result will then be combined with the results from the two other streams. Both combination functions are described in section 3.1.4.

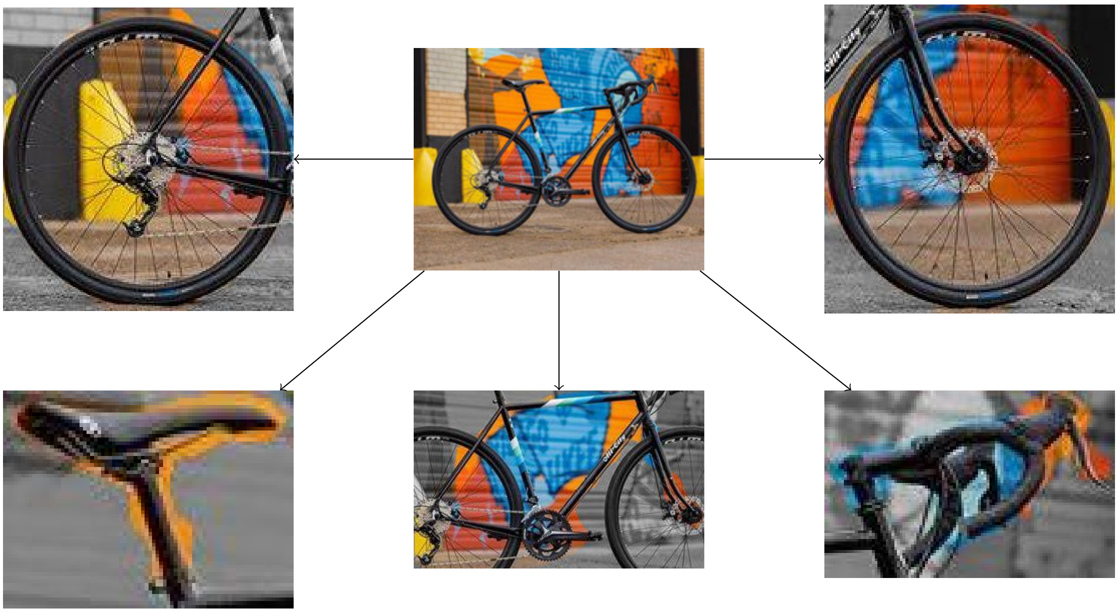

Figure 3.5: Partial images extracted from the input image.

3.1.4 Combination

The three aforementioned pipelines provide a continuous category score value for all possible categories, with the most likely category highlighted. The goal is to combine these category score values and conclude which category is the most likely category for the input image.

Since we propose that all three streams have the same impact on the result of the combination, the first step is to combine the results from the colored masked partial image stream to a single category score vector. To calculate this one category score vector, we add the values of the same category of all partial images together and divide the result through the number of partial pictures, as shown in equation 3.6. n is the number of partial images for one input image, while $x_{j,i}$ stands for the j-th category of the i-th partial image.

$$z_j = \frac{\sum_{i=1}^{n} x_{j,i}}{n} \tag{3.6}$$

After the results of the color masked partial image streams have been combined, we use a similar equation, shown in equation 3.7, to aggregate the results of the three streams and determine the category for the input image.

$$z_j = \frac{\sum_{i=1}^{3} x_{j,i}}{3} \tag{3.7}$$

The category with the highest resulting score is then chosen as the category for the input image.

3.2 Quality Assessment

The second step of our model assesses the quality of a bicycle after its category has been determined using only bicycles of the same category.
The quality assessment uses the same three streams as the classification step, but the color masked partial image stream is different for the quality assessment than the classification step. Therefore, this section will only contain the differences between the two steps.

3.2.1 Color Masked Partial Image

The color masked partial image pipeline for the quality assessment step follows the same structure as the classification step. The main difference is, while the classification step does not take all possible partial images into account (e.g., handlebar and saddle), the quality assessment step does. The reason for this is, for example, the handlebar does not differ significantly between different categories. That is because they all follow a very similar pattern. However, regarding the quality assessment, all recognized parts of the bicycle hold significant value.
Due to this, the color masked partial image stream of the quality assessment step observes all detected parts.

4 Experiments

This chapter focuses on the experiments we did to check whether our proposed approach works or not. In section 4.1, we explain the dataset we use and how the images are annotated. While in section 4.2, we focus on the experiments divided into two steps.

4.1 Dataset

We used an existing dataset by Khan [11] containing 1380 bicycle images (1080 training, 300 testing). The dataset we use only includes a part of the original dataset, with 523 pictures for training and 131 for testing. The collected images have different sizes. We do not use downsampling for the segmentation network to bring all images to a fixed size. However, for the classification with CNN, we downsample all instances to 224×224 pixels.

4.1.1 Data Collection

The Bicycle Image Dataset [11] is composed of 1080 training images and 300 testing images and was published in 2020 to provide a dataset that can detect bicycles in any situation. It is most suitable for parking spaces and bicycle competitions. Due to this, the dataset contains many images which show races or more than one bicycle. These images do not help us determine the quality of the bicycles shown in the pictures because they are incomplete or partially hidden. It could also be unclear which bicycle is the main focus of the picture. Additionally, some of the images show bicycles that are unique kinds of bicycles (see Fig. 4.1). Finally, 523 training and 131 testing images remain in our dataset.

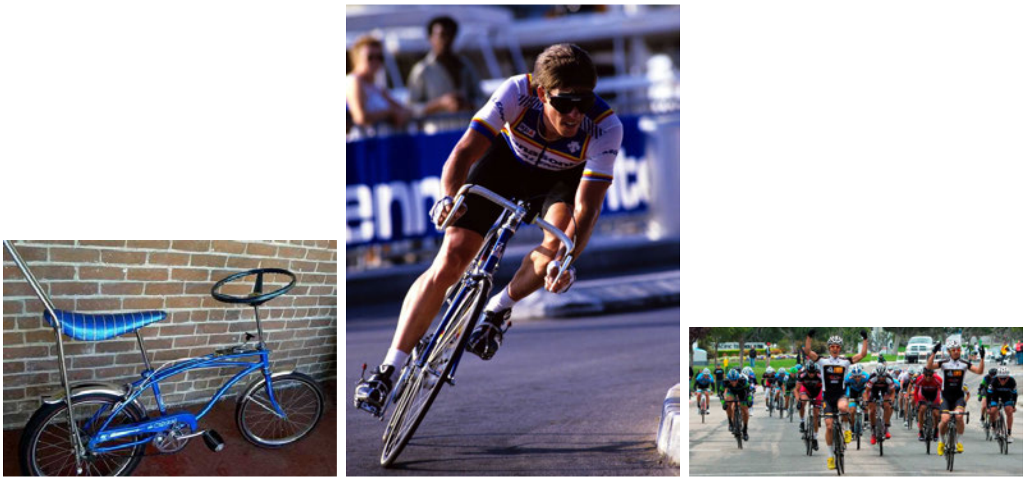

Figure 4.1: Left: example for special kind of bicycle (see handlebar), Middle: important characteristic are hidden, Right: images with no focused bicycle.

4.1.2 Data Annotation

The data annotation is divided into the annotation used for object classification and annotations used for semantic segmentation. The part for object classification describes the classes used for classification, while the part for semantic segmentation includes examples of the annotations.

CNN Annotation

As described in section 3, our goal is to classify the quality of a bicycle using two steps, first category classification and then quality assessment. For the category classification, we use five categories. These five categories are **motor**, bicycles with a secondary drive, **mountain bike**, bicycles with wider tires, **old**, bicycles notable older or without the latest technology (e.g., dynamo), **race**, bicycles used for races, and **rest**, bicycles not fitting in the other categories.

	motor	mountain bike	old	race	rest
training	58	97	115	69	184
test	19	16	28	20	48

Table 4.1: Annotation for the category classification (number indicates the number of images per category).

There are three categories for the quality assessment **high, medium, and low**. We only choose three as the number of classes because the number of images for some categories is not that high, and dividing them into more classes would be difficult and would make testing more complicated.

	motor		mountain bike		old		race		rest	
	train	test	train	test	train	test	train	test	train	test
high	30	10	47	9	22	10	30	6	69	16
medium	10	3	40	2	69	15	24	5	77	21
low	18	6	10	5	24	3	15	9	38	11
overall	58	19	97	16	115	28	69	20	184	48

Table 4.2: Annotation for the quality classification (number indicates the number of images per category).

Segmentation Annotation

For each image of the dataset, two segmentation annotations exist, one to distinguish between the bicycle and the background of the image (Color Masked Image, see 3.1.2), and one to highlight different parts of the bicycle (Color Masked Partial Image, see 3.1.3). We only focus on the **handlebar**, the **tires**, the **saddle**, and the **frame** of the bicycle.

We use the graphical image annotation tool **lableme** by Wada [25] for the annotation of the images. The resulting annotations can be converted to a PASCAL VOC-format dataset or a COCO-format dataset with the corresponding converter script, which is part of lableme's GitHub repository.

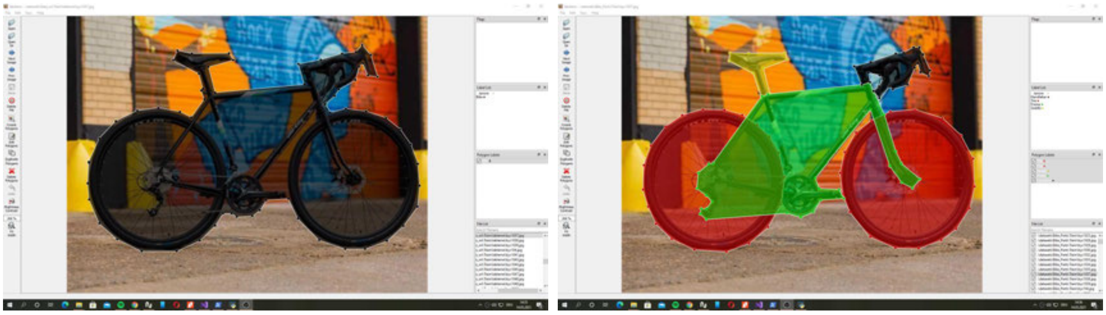

Figure 4.2: Left: Lableme annotation of bicycle
Right: Lableme annotation of bicycle parts

4.2 Training Procedures and Results

In this section, we provide details of the training procedures and the results of the training procedures. We trained different network structures (R50-FPN [16] and LeNet5 [13]) for separate tasks and combine various approaches described in Chapter 3 to find the best combination of the different inputs (color image, color mask, and color mask partial).

4.2.1 Data Preparation

Before evaluating if our approach works, we have to prepare the datasets to run our R50-FPN and LeNet5 networks. For the R50-FPN, there are two COCO Annotations, one for bicycle detection and one for bicycle part detection. Both use the original dataset as their base. The LeNet5 network, on the other hand, uses three different input datasets. The first is the original dataset, the same as for R50-FPN, and the other two are the results of the output of the semantic segmentation.

CNN

To let our LeNet5 Network process the image and its corresponding label, we use Keras ImageDataGenerator() and flow_from_directory() functions. ImageDataGenerator utilizes a directory approach where the images are saved in the directory with the corresponding label (see Figure 4.3). This method lets us automatically resize every image to 224 × 224 pixels and split the train directory into a training set and a validation set. In the case of the Color Masked and Color Masked Partial Images, we create a new directory following the same structure as the main directory for the category detection and quality assessment. Both the training and test images for the Color Masked and Color Masked Partial Images are modified using the resulting network of semantic segmentation. Every pixel that lies outside of the detected boundary box is ignored. The remaining background in the boundary box is grey-scaled. We modified the original visualizer used by the framework detectron2 [26] to ignore the added colored masks used for better visualization and leave the detected object in its original color.

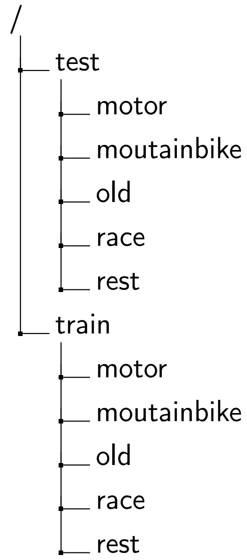

Figure 4.3: Example for the category directory tree.

Segmentation Annotation

The framework detectron2 by Yuxin Wu et al. [26] supports the PASCAL VOC and COCO annotations as the base for the training, validation, and testing set, but only the COCO annotation is supported for semantic segmentation. As mentioned in section 4.1.2, the annotation tool lableme contains a script that allows us to convert the annotations made by lableme into COCO Annotations. The COCO Annotation contains information on the boundary box surrounding the segment, the area describing the segment, its category, and whether or not it is part of a crowd. Detectron2 can then use the COCO Annotation files to generate the dataset and handle the needed modifications for the R50-FPN.

```
{
"info": info,
"images": [image],
"annotations": [annotation],
"licenses": [license],
}

image{
"id": int,
"width": int,
"height": int,
"file_name": str,
"license": int,
"flickr_url": str,
"coco_url": str,
"date_captured": datetime,
}

annotation{
  "id": int,
  "image_id": int,
  "category_id": int,
  "segmentation": RLE or [polygon],
  "area": float, "bbox": [x,y,width,height],
  "iscrowd": 0 or 1,
}

categories[{
  "id": int,
  "name": str,
  "supercategory": str,
}]
```

Figure 4.4: Annotation style for COCO.
Info and License contains information for dataset, but not for the detection (see https://cocodataset.org/#format-data for more information).

4.2.2 Evaluation

We conducted several experiments to evaluate the performance of our proposed algorithm. As illustrated in Fig. 3.2, the proposed method consists of two steps, the classification step and the quality assessment step. Both steps consist of three pipelines, as illustrated in Fig. 3.3, the color image stream, the colored masked stream, and the colored masked partial image stream. For both steps, the color image stream utilizes a LeNet-5 network, while the two streams that employ masks use a R50-FPN for semantic segmentation, providing the masks additionally. Each of the three pipelines performs a distinct visual analysis. For finding the best-performing combination of these three streams, various combinations of the three pipelines are combined. First, we will focus on the standalone evaluation of the two semantic segmentation networks, which are utilized by both the category classification and the quality assessment step. Afterward, we will evaluate the category classification step first and the quality assessment second.

Semantic Segmentation (R50-FPN)

Since the modifications of the original images using the R50-FPN [16] are deployed for both steps of our proposed algorithm, we decided to evaluate the semantic segmentation as a standalone part of the evaluation. They furthermore contain information on the feasibility of the detection of the bicycle as a whole and the detection of specific parts of the bicycle from different angles, making it wiser to evaluate them individually. Additionally, this gives a better way for others to decide whether they want to use object detection to detect specific parts of a larger object or expand on the approach.
To compare some of the results of our models, we compare them with the R50-FPN example Mask R-CNN model by [26][1] and the winner of the Coco Challenge 2020[2]. Additionally, we also include some metrics which are not used by default but still contain fascinating information.

Both the bicycle and bicycle part detection use a modified version of the original visualizer by detectron2. Detected objects of the same category possessing overlapping boundary boxes are combined to one object to reduce the detection of a partial object, e.g., the handlebar. Furthermore, should the detection generate a resulting image already containing part of the detected object in the other boundary box, e.g., tire, the

[1] https://github.com/facebookresearch/detectron2/blob/master/MODEL_ZOO.md
[2] https://cocodataset.org/#detection-leaderboard

first bounding box is extended to include both bounding boxes. For the bicycle part detection, we additionally removed the masks of a different object connected to the object we want to evaluate. E.g., the frame is connected to all other detected objects. And the frame's mask thus will be ignored in the other objects.
Both networks also follow the suggestion for one GPU usage for training done by the developers of detectron2 to speed up the evaluation and handle VRAM restrictions. These contain setting the images per batch to 2 and base_lr to 0.0025. The base_lr stands for base learning rate and helps restrain overfitting. Additionally to these two settings, we also set the warmup phase to 1000 iterations and train for 10000 iterations. After every 250 iterations, we conduct an extra evaluation to evaluate the AP, which is the standard comparison metric used by PASCAL VOC and COCO.

network	AP	AP^{50}	AP^{75}	AP^{S}	AP^{M}	AP^{L}
COCO 2020	0.525	0.764	0.580	0.359	0.559	0.665
R50-FPN on COCO	0.352	0.563	0.375	0.172	0.377	0.503
bicycle detection	0.783	0.984	0.958	NaN	NaN	0.783
bicycle parts detection	0.695	0.962	0.824	0.660	0.647	0.692

Table 4.3: Comparison with COCO2020 Winner and R50-FPN on COCO dataset.

	AP	AP^{50}	AP^{75}	AP^{S}	AP^{M}	AP^{L}
boundary box	0.888	0.984	0.984	NaN	NaN	0.888
segmentation	0.783	0.984	0.958	NaN	NaN	0.783

	accuracy	false-positive	false-negative
faster R-CNN	1.0	-	0.0
masked R-CNN	0.971	0.048	0.017

Table 4.4: Evaluations for bicycle detection on boundary box and segmentation. 1. Table contains the standard AP and 2. Table additional metrics.

	AP-Frame	AP-Handlebar	AP-Saddle	AP-Tire
boundary box	0.821	0.733	0.755	0.972
segmentation	0.671	0.470	0.698	0.943

	AP	AP^{50}	AP^{75}	AP^{S}	AP^{M}	AP^{L}
boundary box	0.813	0.996	0.938	0.784	0.747	0.811
segmentation	0.695	0.962	0.824	0.660	0.647	0.692

	accuracy	false-positive	false-negative
faster R-CNN	0.987	-	0.024
masked R-CNN	0.963	0.052	0.023

Table 4.5: Evaluations for bicycle parts detection on boundary box and segmentation. 1. Table contains the AP for the different objects, 2. Table the standard AP and 3. Table accuracy metrics obtained during the training phase.

CNN category classification

This section focuses on the first of the two parts of our proposed model, and it uses untrained LeNet-5 [13] deep networks to identify the class belonging to the bicycle. We use three different models to get better overall results by combining the results of the three models. Each of the three models is trained for 200 iterations with a batch size of 32. 20% of the training set is used as a validation set. Additionally, the training set is shuffled with a seed of 42 to get better training. Adam is used as the optimization method, which focuses on increasing the accuracy. All networks use a Softmax layer as the last layer and categorical cross-entropy as the loss function.

The first model is our baseline, named color image, and uses the unmodified version of our dataset. It gives us a clue whether the categorization is feasible and works as a reference for the other results. All images are completely colored.
The second model, named color masks, combines the bike detection semantic segmentation model and the LeNet-5 network to classify the images. The images for this model only contain the calculated boundary boxes from the bike detection and are partially grey-scaled to better focus on the bicycle itself.
The third model, named color masked parts, is the most complex model of the three. It contains the R50-FPN tasked with bicycle parts detection and up to four LeNet-5

networks. The number of LeNet-5 networks depends on how many of the parts help us with a better detection. Each of the four LeNet-5 networks is responsible for one of the four different parts of the bicycle we try to detect. Especially the classification of the tires is interesting. Since in most cases, the tires are split into two different images by the bicycle part detection. To give every part the same weight, we combine the predicted results of the tires into one result. Subsequently, we combine the results of up to four models used by the color masked parts model. Finally, the aggregated results of the four LeNet-5 networks decide the category of the bicycle together. Whether all four networks will be used or not for the final model depends on whether the addition of the networks helps with a better classification.
After the best performing color masked parts model is determined, we try combining the results of the three models to decide which of the combinations is the best. For the color masked parts model, look at table 4.7 and for the complete classification at table 4.6.

model components			accuracy	wrong out of 131
color image	color masks	color masked parts		
✓	✗	✗	0.794	27
✗	✓	✗	0.702	39
✗	✗	✓	0.786	28
✓	✓	✗	0.779	29
✓	✗	✓	0.794	27
✗	✓	✓	0.710	38
✓	✓	✓	**0.801**	**26**

Table 4.6: Accuracy of different combinations between the three networks for the category classification.

bike parts				accuracy	wrong out of 131
Frame	Handlebar	Saddle	Tires		
✓	✗	✗	✗	0.740	34
✗	✓	✗	✗	0.580	55
✗	✗	✓	✗	0.600	52
✗	✗	✗	✓	0.743	30
✓	✗	✗	✓	**0.786**	**28**
✓	✓	✗	✓	**0.786**	**28**
✓	✗	✓	✓	**0.786**	**28**
✓	✓	✓	✓	0.740	34

Table 4.7: Accuracy of different combinations between the three bike parts for the category classification.

CNN quality Assessment

The quality assessment is the second step of our proposed model and focuses on assessing the quality of a bicycle depending on its category. That means we have five different models, one for each category. These models are also divided into three models: the color image, color mask, and color masked parts. The evaluation follows the same procedure as the category classification. Unlike the category classification, the quality assessment will only use 50 iterations since the number of images is lower by a large margin. Additionally, only the category rest will have a validation set because, due to the small sample size, no validation by TensorFlow will be performed.

model components			rest	
color image	color masks	color masked parts	accuracy	wrong out of 48
✓	✗	✗	0.792	10
✗	✓	✗	0.813	9
✗	✗	✓	0.781	11
✓	✓	✗	**0.854**	**7**
✓	✗	✓	0.813	9
✗	✓	✓	0.792	10
✓	✓	✓	0.833	8

Table 4.8: Accuracy of different combinations between the three networks for the quality assessment of the rest category.

bike parts				rest	
Frame	Handlebar	Saddle	Tires	accuracy	wrong out of 48
✓	✗	✗	✗	0.625	18
✗	✓	✗	✗	0.667	16
✗	✗	✓	✗	0.667	16
✗	✗	✗	✓	**0.781**	**11**
✗	✓	✓	✗	0.688	15
✓	✓	✓	✗	0.688	15
✗	✓	✓	✓	0.729	13
✓	✗	✗	✓	0.729	13
✓	✓	✓	✓	0.708	14

Table 4.9: Accuracy of different combinations between the three bike parts for the quality assessment of the rest category.

category	accuracy	wrong	components
motor	1.0	0 out of 19	color & color mask & parts (frame, saddle, tire)
mountain bike	1.0	0 out of 16	color & color mask & parts (handlebar, saddle, tire)
old	0.821	5 out of 28	color & parts (tire)
race	1.0	0 out of 20	color mask & parts (frame, handlebar, saddle, tire)
rest	0.854	7 out of 48	color & color mask

Table 4.10: This table shows the accuracy of the quality assessment for the different categories of bicycles and which of the three pipelines where used to get the best performance.

4.3 Discussion

After we described the training procedures and listed the results of our experiments in section 4.2, we will discuss the results in this section and find the best-performing model.

Semantic Segmentation (R50-FPN)

The results for the Semantic Segmentation for bike and bike part detection, as seen in tables 4.3, 4.4, and 4.5, are very good with an average accuracy of over 80% for an accuracy of 75% or higher detecting whether a pixel is included in the mask. That is especially surprising since, due to a fixed learning rate of 0.0025, we have no fine-tuning in the later stage of the training process. While the learning rate rises slowly in the beginning due to the slow start option, it remains at the fixed learning rate of 0.0025 afterward, compared to the comparison metrics provided by detectron2, where the learning rate drops again at the end. The missing entries of the bicycle detection network for the average precision of the small and medium areas are expected since we are trying to detect a bicycle, which means we are always trying to detect large contiguous objects. The low average precision for the segmentation of handlebars is again a result of the lack of fine-tuning. Since we are good at recognizing the center of an object but not so good at the outer parts of the object and the handlebar is not a vast continuous entity but rather a combination of small short parts sticking out in any direction from the central section. We can see this clearly in Fig. 4.2, where we see the annotated mask for the bicycle parts.

CNN category classification

The fact that the handlebar and saddle are not excellent indicators for the category of a bicycle is not that surprising since especially the saddle of many bicycles has a similar look regardless of the kind of bicycle. The same goes to a certain extent for the handlebar, especially since the segmentation of the handlebar is the worst of the segmentation (see table 4.5), being the only segmentation below 60% average accuracy. Since including the categorization based on the handlebar or saddle does not increase the accuracy of the color masked parts model, the best-performing model for the color masked parts model only contains the information of the frame and tires.

The most powerful model for the entire bicycle categorization is the combination of three different categorization methods, the color image, color mask, and color masked parts (see table 4.6). We reach an accuracy of 80% despite not focusing on fine-tuning for the different LeNet-5 networks that we use for categorization and using the same model for all of them.

CNN quality classification

Due to the unique quality types for the motor category, which focuses more on the fact that we have high-class, first-generation, and modified bicycles as quality classes, it is not surprising that we achieve very high accuracy.

For the mountain bike category, the color masked parts model is only included since it is nearly as accurate as the color image model and color mask model, both of which get an accuracy of 100%.

The old category shows that using more than one model to achieve higher success is a good approach. The color image model and the color masked parts model, only focusing on the tire, have both a weaker accuracy than 82%, the color image model has 71%, and the tire 75%. Therefore, one would assume that if one combined the two models, we would get at best 75%, but instead, we get 82%. That proves that using different approaches and merging them is good because one model could be good at detecting one thing and bad at another, while it is precisely the opposite for the other model. In that case, both models would help each other by finding a middle ground better at detecting both aspects. The opposite could also happen, where the combined model gets weaker and not stronger.

In the race category's case, all color masked parts models combined lead to the best result for the color masked parts component, showing the power of dividing the bicycle into parts to get a better quality assessment.

That the best-performing model for the rest category does not contain any bicycle parts is not that surprising since the rest category contains several different bicycles. That makes it easier to assess the quality with the whole bicycle in mind and not by focusing on parts. Still, the inclusion of bicycle parts allows results of over 80%, and with 78% as a standalone decider, it is not weak. Finally, the best-performing model for the rest category reaches 85%.

5 Conclusion and Future Work

In this work, we tried to build a model that could assess the quality of bicycles. To achieve that, we proposed to divide the bicycles according to specific categories to simplify the assessment. We also used object detection with R-CNN to detect specific bicycle parts and use them to help the categorization and quality assessment. The results show that using semantic segmentation to identify bicycles or parts of bicycle work and focusing on bicycle components enhances the overall performance of the categorization and quality assessment. Additionally, dividing bicycles according to categories before assessing the quality is helpful since three out of the five categories reach a 100% accuracy by first categorizing them. The other two have an accuracy of over 80%. The last part of the proposed approach was using different bicycle features to strengthen the overall performance of our model, focusing on these features and then combining the results of the various pipelines. Table 4.10 shows that all five categories use different combinations of our three proposed pipelines, proving that this approach leads to better results.

5.1 Limitations and Future Directions

For this work, we used a not annotated small dataset because there is no publicly available dataset for quality assessment on bicycles, as far as we know. That leads to the fact that the categorization of bicycles follows our subjective view of bicycles. Additionally, the quality assessment only divides between high, medium, and low quality. Because of this, we neither get the recommended price for the bicycle nor the exact quality of the bicycle. Therefore for future directions, it would be helpful to work with a dealer who sells second-hand bicycles to get a better dataset with more images and better annotations.

Additionally, it would be interesting to see how the average accuracy for the semantic segmentation changes if we use more than one GPU and have more VRAMM since we could then remove the added setting for the config files for the R50-FPN. Especially since the fixed learn rate limits the possible optimizations for the neurons in the model.

Another task for future development would be to optimize the CNNs. Furthermore, adding weights to the three pipelines and five different models of the color mask partial image pipeline could lead to better results.

Bibliography

[1] C. T. Artan and Tolga Kaya. Car Damage Analysis for Insurance Market Using Convolutional Neural Networks. In Cengiz Kahraman, Selcuk Cebi, Sezi Cevik Onar, Basar Oztaysi, A. Cagri Tolga, and Irem Ucal Sari, editors, *Intelligent and Fuzzy Techniques in Big Data Analytics and Decision Making*, Advances in Intelligent Systems and Computing, pages 313–321, Cham, 2020. Springer International Publishing. ISBN 978-3-030-23756-1. doi: 10.1007/978-3-030-23756-1_39.

[2] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-Based Models for Speech Recognition. *arXiv:1506.07503 [cs, stat]*, June 2015, http://arxiv.org/abs/1506.07503 (Accessed: 2021-03-22).

[3] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable Convolutional Networks. *arXiv:1703.06211 [cs]*, June 2017, http://arxiv.org/abs/1703.06211 (Accessed: 2021-03-21).

[4] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009. doi: 10.1109/CVPR.2009.5206848.

[5] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *Int J Comput Vis*, 88(2):303–338, June 2010. ISSN 1573-1405. doi: 10.1007/s11263-009-0275-4, https://doi.org/10.1007/s11263-009-0275-4 (Accessed: 2021-03-18).

[6] Ross Girshick. Fast R-CNN. *arXiv:1504.08083 [cs]*, September 2015, http://arxiv.org/abs/1504.08083 (Accessed: 2021-03-22).

[7] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv:1311.2524 [cs]*, October 2014, http://arxiv.org/abs/1311.2524 (Accessed: 2021-03-22).

[8] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 Object Category Dataset. Report or Paper, California Institute of Technology, March 2007, https://resolver.caltech.edu/CaltechAUTHORS:CNS-TR-2007-001 (Accessed: 2021-02-04).

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, December 2015, http://arxiv.org/abs/1512.03385 (Accessed: 2021-02-25).

[10] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167 [cs]*, March 2015, http://arxiv.org/abs/1502.03167 (Accessed: 2021-03-18).

[11] Shahid Khan. Bicycle Image Dataset – MaviIntelligence, June 2020, http://maviintelligence.com/bicycle-image-dataset/ (Accessed: 2020-12-14).

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, May 2017. ISSN 0001-0782, 1557-7317. doi: 10.1145/3065386, https://dl.acm.org/doi/10.1145/3065386 (Accessed: 2021-03-13).

[13] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, Nov./1998. ISSN 00189219. doi: 10.1109/5.726791, http://ieeexplore.ieee.org/document/726791/ (Accessed: 2021-02-24).

[14] Li Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, April 2006. ISSN 1939-3539. doi: 10.1109/TPAMI.2006.79.

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context. *arXiv:1405.0312 [cs]*, February 2015, http://arxiv.org/abs/1405.0312 (Accessed: 2021-03-18).

[16] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. *arXiv:1612.03144 [cs]*, April 2017, http://arxiv.org/abs/1612.03144 (Accessed: 2021-03-18).

[17] Karoon Rashedi Nia. *Automatic Building Damage Assessment Using Deep Learning and Ground-Level Image Data*. PhD thesis, SIMON FRASER UNIVERSITY, Canada, January 2017, https://core.ac.uk/download/pdf/80537574.pdf (Accessed: 2021-01-29).

[18] Peter Oram. WordNet: An electronic lexical database. Christiane Fellbaum (Ed.). Cambridge, MA: MIT Press, 1998. Pp. 423. *Applied Psycholinguistics*, 22(1):131–134, March 2001. ISSN 1469-1817, 0142-7164. doi: 10.1017/S0142716401221079, https://www.cambridge.org/core/journals/applied-psycholinguistics/article/wordnet-an-electronic-lexical-database-christiane-fellbaum-ed-cambridge-ma-mit-press-1998-pp-423/8A9F540FB453B327C1AF0AC74E2F7D4D (Accessed: 2021-02-24).

[19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv:1506.01497 [cs]*, January 2016, http://arxiv.org/abs/1506.01497 (Accessed: 2021-03-18).

[20] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *Int J Comput Vis*, 115(3):211–252, December 2015. ISSN 1573-1405. doi: 10.1007/s11263-015-0816-y, https://doi.org/10.1007/s11263-015-0816-y (Accessed: 2021-03-13).

[21] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-class Object Recognition and Segmentation. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006*, Lecture Notes in Computer Science, pages 1–15, Berlin, Heidelberg, 2006. Springer. ISBN 978-3-540-33833-8. doi: 10.1007/11744023_1.

[22] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]*, April 2015, http://arxiv.org/abs/1409.1556 (Accessed: 2021-03-16).

[23] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions. *arXiv:1409.4842 [cs]*, September 2014, http://arxiv.org/abs/1409.4842 (Accessed: 2021-03-16).

[24] Sik-Ho Tsang. Review: FPN — Feature Pyramid Network (Object Detection) | by Sik-Ho Tsang | Towards Data Science, January 2019, https://towardsdatascience.com/review-fpn-feature-pyramid-network-object-detection-262fc7482610 (Accessed: 2021-03-18).

[25] Kentaro Wada. Wkentaro/labelme, April 2021, https://github.com/wkentaro/labelme (Accessed: 2021-04-28).

[26] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Facebookresearch/detectron2. Facebook Research, December 2020, https://github.com/facebookresearch/detectron2 (Accessed: 2020-12-14).

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „— bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] — ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen."

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: [redacted]

Vorname: [redacted]

dass ich die vorliegende Bachelorarbeit – bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Quality Classification and Evaluation of Bicycles Based on Images Using Deep Learning Approaches

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

[redacted]	[redacted]	[redacted]
Ort	Datum	Unterschrift im Original