Karlsruhe Institute of Technology

Fraunhofer IOSB

Logo retrieval in mass data using deep learning

MASTER THESIS

KIT - KARLSRUHE INSTITUTE OF TECHNOLOGY
FRAUNHOFER IOSB - FRAUNHOFER INSTITUTE OF OPTRONICS,
SYSTEM TECHNOLOGIES AND IMAGE EXPLOITATION

Andras Tüzkö

June 14, 2017

Main Advisor: Dipl.-Inform. Christian Herrmann
Co-Advisor: Dipl.-Inform. Daniel Manger

Interaktive Echtzeitsysteme

Statement of authorship

I hereby declare that I have produced this work by myself except the utilities known to the supervisor, that I have labeled all used utilities completely and detailed and that I have labeled all material that has been taken with or without modification from the work of others.

Karlsruhe, June 14, 2017 Andras Tüzkö Andras Tüzkö

Contents

1

Introduction

Advertising with static logos is one of the most important marketing methods. A very effective way to reach a lot of people with these static logos is to sponsor sport teams or to buy advertising spaces in sport events broadcasted on the TV. However, the prices of these surfaces mean huge expenses for the advertiser. This is the reason why the need for logo appearance statistics of sport videos arises. In particular there is a desire for quantitative measurement of the proportional size of the logo to the screen and of the time one particular logo is visible on the screen. This data is then used to judge the cost efficiency for the specific logo placement, i.e. to be able to decide on which sport event to advertise, with which size of logo and where to place it.

In this work a system for logo retrieval with proposal based object detection and classification will be presented. The system consists of a logo detector and a classifier used for feature extraction. The logo detector is a faster region based convolutional neural network [Ren15] trained to recognize logos on images. For feature extraction from the proposed region, there will be the performance of several classifier neural network tested. A similarity score will be calculated to decide the identity of the logo to be searched and the proposed region. To recognize logos in videos, a video will be cut into frames and then the system will be run on every image.

The challenge of this task is manifold. The first problem is that the logos in these videos are far from being perfectly clear. They can be partially occluded, blurred - if the camera moves fast, perspectively transformed, rotated and can have various coloring, suiting well to the design of the shirt or the arena. In addition, there is a problem with the ambient illumination variation just as for other computer vision tasks. The second challenge is the large number of different companies, having a legal prohibition of copying symbols. This already results in a huge appearance variety, which is further grown by the diversity of logos within a company. This makes the detection of logos very complex. In addition, a lot of company uses wordmarks i.e. a logo only with its name. It is a harder task to recognize complete words than only a letter or a simple graphic mark, symbol. Furthermore in classification tasks, the objects to be classified have usually a 3d shape in the reality, whereas logos have only a planar surface. This means, it yields only less information, if a logo is photographed from any other angle than perpendicular to the plane of that, not like other objects. Unfortunately, there are only a few publicly available small datasets with bounding box annotated logos. The majority of the images are adjusted to ensure a good visibility of the logos, not like on the frames of the sport videos.

To master these challenges, the available solutions and methods improved recently a lot. In the decade before, hand-crafted feature extraction was prevalent in computer vision tasks. It needed an expert to create a system and it yielded often only mediocre results. Deep learning methods for computer vision problems have been dominant since the success of convolutional neural networks in 2012 [Kri12]. Compared to earlier systems (e.g. SIFT [Low04], HOG [Dal05] features), the great improvement of deep learning methods is the capability of learning how to extract features automat-

ically. The enhancements are encouraged by continuous research. The development of deep nets is mainly powered by the annually organized ImageNet classification challenge [Rus15]. Since the aim of this contest is to classify an object, which fills out the majority of an image, the location of the particular object is irrelevant. To be able to classify and recognize objects, which have a much smaller size relative to the size of the whole image, region based classification can be utilized.

The rest of this thesis is organized as follows. Section 2 reviews the related works of image retrieval, object detection and logo retrieval. In Section 3 the proposal based object detection with convolutional neural networks will be introduced. Section 4 describes the logo retrieval system. Afterwards, Section 5 includes evaluation and comparison of the system with another logo retrieval methods. Finally, the last section concludes the work and gives prospects on future work.

2

Related Work

2.1 Image Retrieval

Many technics outside the scope of deep learning exists for images retrieval from videos. SIFT features [Low04] with bag-of-visual-words were used to efficiently get translation invariant descriptors around keypoints by Zisserman [Siv03]. HOG [Dal05] SIFT, HOG blockwise orientation histograms

2.2 Object Detection

Keypoint detection: translation, rotation invariant Viola Jones

2.3 Logo Retrieval

3

Proposal Based Object Detection and Classification

In this section the theoretical overview of the logo retrieval system will be presented. First of all, the advanced Convolutional Neural Networks (CNN) are detailed in section 3.1. The explained networks are later evaluated in chapter 5. Fully convolutional networks will be introduced in section 3.2. Section 3.3 explains region proposal systems for generating candidate object locations on an image. Afterwards, the section 3.4 describes region based convolutional neural networks for object detection and classification. The improvement of this method, the fast region based convolutional neural networks will be detailed in the section 3.4.2. Folowing this, the further development, the faster region based convolutional neural networks will be reviewed in the section 3.4.3.

3.1 Modern Convolutional Neural Networks

Convolutional neural networks (CNN) are inspired by the visual cortex of humans, introduced by LeCun et. al. [LeC89], for zip code recognition. Image classification challenges are dominated by convolutional neural networks since the success of AlexNet architecture [Kri12] in ImageNet LSVRC contest [Rus15]. ImageNet dataset contains of 15 million images, in the challenge only a subset of that is used. The task in this challenge is to classify 100.000 images of 1000 categories, after training with 1.2 million annotated images.

AlexNet was the first neural network after the conquest of support vector machines, achieving prominent performance, and it won the ImageNet challenge in 2012. It consists of 5 convolutional layers, each followed by a max-pooling, which counted for a very deep network at that time. As activation function ReLU (Rectified Linear Units) are utilized. This results in much shorter training times, than $tanh$, which is susceptible to saturate. It uses dropout layer [Hin12] as a regularization method against overfitting, which allows to train deeper networks. It simulates the presence of more neural networks, each inferring a different convolutional map from the input image. The network ends with 3 fully connected layers, having 1000 outputs for the 1000 classes to be classified in the last layer.

In the next year, in 2013, Zeiler and Fergus proposed ZFNet [Zei13], which takes AlexNet as basis, but outperforms it slightly. They achieved, to visualize feature activations, which helped to reveal several problems of AlexNet. They applied smaller filter sizes (7x7 instead of 11x11) and the stride is also lowered from 4 to 2 in the first convolutional layer.

Szegedy et.al. introduces GoogLeNet [Sze14], which contains much more layers and won the classification challenge in 2014. This network utilizes "Inception" blocks, which embodies three

convolutional layers with different filter sizes, beeing responzible for the features of diverse sizes in the same input conv map. In addition a block comprises also a max pooling branch, to obtain small translation invariance. The output of all contained layers are concatenated together, and serves as the output of the block. In the same year, VGG network was proposed too [Sim14], which won the first place in the localization part of the challenge. This network follows the general architecture of AlexNet, by utilizing also 5 max-pooling layers, and having 3 fully connected layers at the end. However, the number of convolutional layers are increased to 13, whereas the filter sizes are reduced further to 3x3. Convolutional srides are fixed to 1, to preserve any information and not loose on performance.

A medium sized network, called `VGG_CNN_M`, was proposed in [Cha14]. It is very similar to the ZFNet, but has reduced stride and receptive field in the first convolutional layer. For the majority of the experiments in chapter 5, a modified version of this network is used, having only 1024 outputs instead of 4096 in the second last fully connected layer `VGG_CNN_M_1024` [Cha14].

Residual networks (ResNet) [He15] utilize very deep network architecture of 50-151 layers, and won the contest in 2015. It adopts skip connections, which is a direct connection from the output of a lower layer (the one lying closer to the input image). This connections address the problem of degradation, arises when very deep networks are involved, causing performance dropping, by eliminating the need of learning to map the identity of earlier layer's output. It applies batch normalization (BN) layer for regularization and input normalization, proposed in [Iof15]. This is an intermediate normalization layer, eliminating the need of mean subtraction and variance division of the input images, buy learning shift and scaling parameters applied on the input map. This means, that the network can learn these parameters specific for every layers, and the normalization can be build in to the network in an end to end manner. Since the introduction of BN, it became the widely applied normalization technique in a lot of new proposed network e.g. GoogLeNet has also a variant, with utilizing BN.

In year 2016, ResNets were further improved by [Xie16], called ResNext - Residual networks with next dimension. This network won the second place on the challenge, where the first place is won by Team Trimps-Soushen with an ensemble of classifiers. ResNext splits a ResNet unit into a multi-branch convolutional neural network, achieving better performance, then earlier networks.

DenseNet [Hua16] has also a ResNet-like architecture, but it rather connects the output of a layer with every subsequent layer's input, This results in a much more dense network, than a conventional feed-forward network, having $L \cdot (L-1)/2$ connections, instead of L, where L is the number of layers or the number of "dense units". This is accomplished by rather concatenating instead of summarizing the output of a unit with the previous output.

3.2 Fully Convolutional Neural Networks

A neural network is fully convolutional, if it does not contain any fully connected layers. Firstly Matan et.al. used FCNs for recognizing strings of digits. Long et. al. proposed [Lon14] how to transform a deep neural network with fully connected classifier layers at the end, to a fully convolutional network. For this purpose the fully connected layers at the end of the network are to be converted to convolutional layers. Particularly, this is achieved, by considering a convolutional layer as a fully connected layer applied in sliding window fashion on the input map. For this purpose, the fully connected layers are replaced with convolutional layers, having a filter size equal to the size of the input feature map in the original network. As a result of this, if the new network is fed with an input

image, larger then the original input size of the network, the convolutional layer will process the feature map, and the output is not a vector, but a grid.

Since the number of weights of a neuron in a fully connected layer is defined by the shape of the data of the layer, the trained network, containing fully connected layers, can process only a fix-sized input image. As a fully convolutional network does not have fully connected layer anymore, it has the advantage of being able to train and test with images of arbitrary sizes.

The outputs of such a network are two dimensional feature maps, which can be used as heatmaps per class, whereas the higher activations mean the presence of that specific class. These convolutional maps can also be used directly for semantic segmentation, where each pixel of an image should be classified. Nowadays fully convolutional networks are essential part of state-of-the-art object detectors, yielding better performance, image size agnosticism, as well as shorter training and inference times as shown in section 3.4.

3.3 Region proposal systems

To recognize different objects on an image, like logos, small regions should be considered. The easiest way to search for these locations is the exhaustive sliding window search, applied on multiple scales. Although, as section **??** presents, this induces a lot of computational costs. In order to reduce this computational burden, region proposal systems can be utilized. Region proposals are possible object locations on an image.

Earlier computer vision solutions used external proposal systems. This means that the proposals of every image should be pre-calculated before training or inference. One of the most popular region proposal methods is Selective Search [Uij13]. It merges neighbor regions according to a similarity score in a bottom-up fashion. It processes an image under 2s on the CPU, which precludes the possibility of real time applications. Edge Boxes [Zit14] are efficiently calculating the number of contours in a box, and ranking them according to that almost real time. Recently, as section 3.4.3 introduces, the proposal system is already part of the neural network.

3.4 Region Based Convolutional Neural Networks

This section gives a brief overview about how faster region based convolutional neural networks evolved.

3.4.1 Regions with Convolutional Neural Network Features

Although this network is already historical, it is worth to mention it, because it helps to understand the improvements of the later systems. Region based convolutional neural networks [Gir13] consist of four separate systems. Firstly, region proposals are generated external with Selective Search. There will be altogether 2000 object positions considered. Secondly, each region of the possible object locations is warped to a size of 227x227, and then the feature vector of every single region is extracted with a CNN. The network [Kri12] is pretrained on the ImageNet dataset [Den09], and then fine-tuned on the final classes. The network is run on every region proposal bounding boxes, to extract vectors with a fixed-size. These vectors will to be written to the disk. Thirdly, a set of class-specific linear Support Vector Machines (SVM) [Cor95] is used to classify the specific region. At last bounding box regressions is run, to reduce the mislocalization of the object. This happens outside of the network.

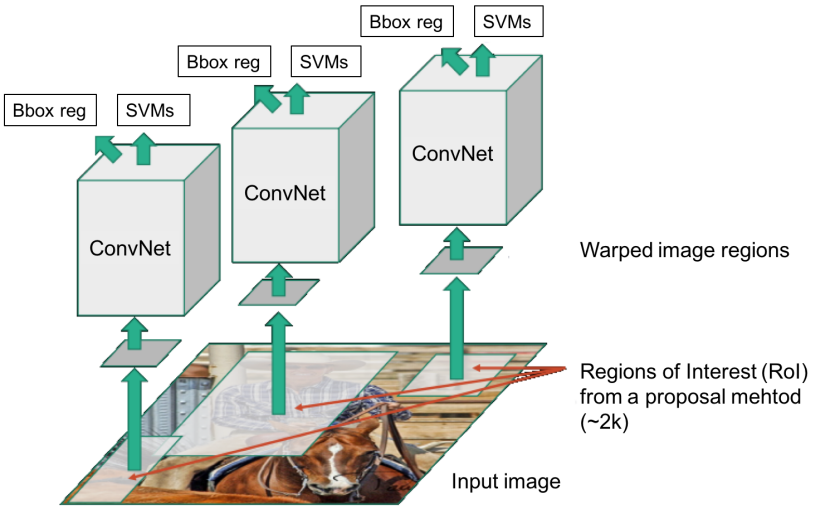

Figure 3.1: R-CNN takes external region proposals, warps the region to a uniform shape, and extracts the features separately. The extracted features are then used to classify the region, and calculate bounding box regression externally.

3.4.2 Fast Region Based Convolutional Neural Network

Fast Region based CNNs [Gir16] are aimed to improve the classification accuracy and feature vector extraction speed of the interest regions, generated also with selective search. An intermediate convolutional feature map is extracted from the whole input image with a fully convolutional neural network, also called as base network in [Ser13]. The output is a downscaled feature map, which is fed to the so called RoI (region of interest) pooling layer. This layer crops regions from the map according to the appropriate downscaled region proposals, and executes a modified version of max pooling on each regions, which results in a convolutional map with a fixed-shape, regardless the size of the region.

After the pooling, fully connected layers are used to calculate the final class probabilities and bounding box regressions for each region. The output of the bounding box regression are class specific small position and size adjustments, needed to refine the rough object locations.

The proposed network is trained for K+1 classes, where K is the number of object classes, and the background is also modelled as a separate class. During training, the positive examples are chosen, regarding the intersection over union (IoU) value to the groundtruth. This value is widely used for measuring the overlapping between regions, regardless the actual size of the regions. The calculation between two regions, R_1 and R_2 is as follows:

$$\frac{area(R_1 \cap R_2)}{area(R_1 \cup R_2)} \tag{3.1}$$

For positive training examples there are thoose regions applied, which have an IoU with the groundtruth at least 0.5. For the background class are the examples with IoU [0.1,0.5) used.

The network is trained with a multi-task loss function for classification and bounding box regression, defined as:

$$L(p,u,t^u,v) = L_{cls}(p,u) + \lambda[u \geq 1]L_{loc}(t^u,v) \tag{3.2}$$

where p is the computed class probabilities, u is the groundtruth class, t^u is the predicted bounding box offset vector for every classes, and v is the groundtruth bounding box position and size. The probabilities are calculated with softmax as usual, so the log loss is used for the classification error: $L_{cls}(p,u) = -log\,p_u$. Since the bounding box regressor cannot be trained with background images, its error is not added to the complete loss. This is achieved, by setting the label of the background class to zero, and using the term $[u \geq 1]$, which is 1 if $u \geq 1$, and 0 otherwise. For regression loss, a smooth version of L1 loss is used, which is defined as follows:

$$L_{loc}(t^u,v) = \sum_{i\in(x,y,w,h)} smooth_{L_1}(t_i^u, v_i) \tag{3.3}$$

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{3.4}$$

The improvements of this method compared to the previous region based CNN introduced in section 3.4 are as follows:

- **Joint feature extraction:** much shorter training and inference time is achieved by the lower computational redundancy of running convolutional layers on the whole image only once, rather then for every proposed regions.
- **One network:** the feature extraction and the classification happens in the same network. This has more advantages:
  - This results again in faster test and training times, due to the unnecessity of writing the extracted feature vectors to disk, which incidentally could require e.g. hundreds of gigabytes of storage [Gir16] for the VOC07 trainval set [Eve].
  - As the backpropagation is implemented through the RoI pooling layer, the whole network, together with the convolutional layers, can be trained jointly, against earlier implementations, like R-CNN [Gir13] or spatial pyramid pooling networks (SPPnet) [He14].
- **Minibatch from a few images:** Faster training speed is achieved by collecting a minibatch only from two images, rather than every region from different images. This method is proved to converging within similar times, despite the high correlated regions.

3.4.3 Faster Region Based Convolutional Neural Network

A great disadvantage of the Fast R-CNN method is, that the region proposals are generated externally. Girshick et.at. introduces Faster R-CNN [Ren15], which generates the interest regions within the neural network nearly cost-free (10ms pro image). This system consists of a region proposal system and a Fast R-CNN object detector.

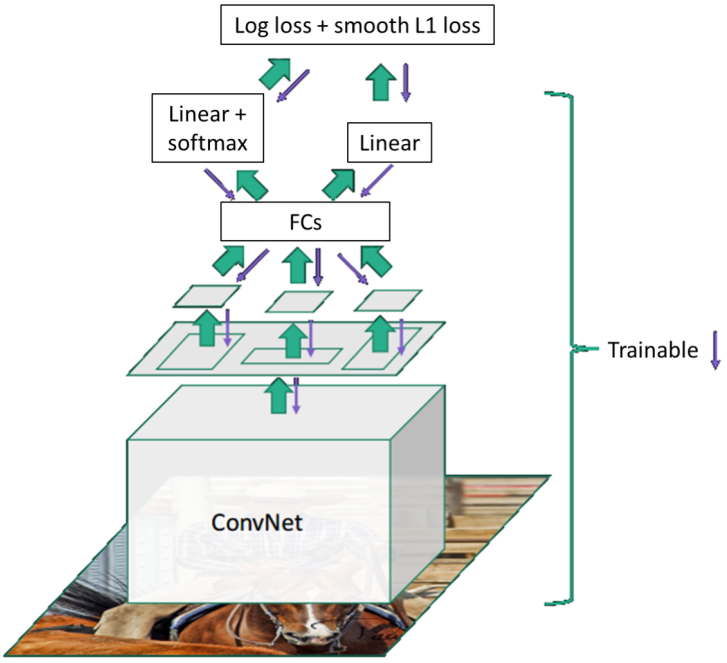

Figure 3.2: Fast R-CNN uses external proposals, then infers the complete image with a fully convolutional network. The proposals is then used to crop regions from the feature map with RoI pooling. The cropped region is then classified and the region coordinates are adjusted with a fully connected network.

Region Proposal Network

The convolutional feature map, extracted by the base network, is processed by the RoI pooling layer. Additionally a thin fully convolutional network, the region proposal network (RPN), is also utilized on the convolutional maps, to generate the proposals. Reference boxes, called "anchors" are generated at every position of the conv feature map, in different scales and different aspect ratios. This ensures the scale invariance of the objects. A convolutional layer with 3x3 kernel extracts a fixed-size vector from every window of the conv map, where in each window 9 anchors are considered. As the fully convolutional network iterates through the conv map in a sliding window fashion, translation invariance is granted. An objectness score and bounding box offset is then calculated for every anchor, by different classifiers and regressors, specialized in a specific scale and aspect ratio.

Training

As the RPN is a class agnostic object detector, it should detect all the types of objects, which the network is trained on. For this purpose, the class informations, during training of RPN, can be discarded. Positive examples are collected from the proposals with an IoU higher than 0.7 with any groundtruth. Negative label is assigned for regions, which have an IoU, lower than 0.3.

Since the objective of the RPN is the same as for Fast R-CNN, namely to classify regions and regress bounding box coordinates, the loss functions for training the RPN can the same multi-task loss, which are used for training a Fast R-CNN, detailed in section 3.4.2.

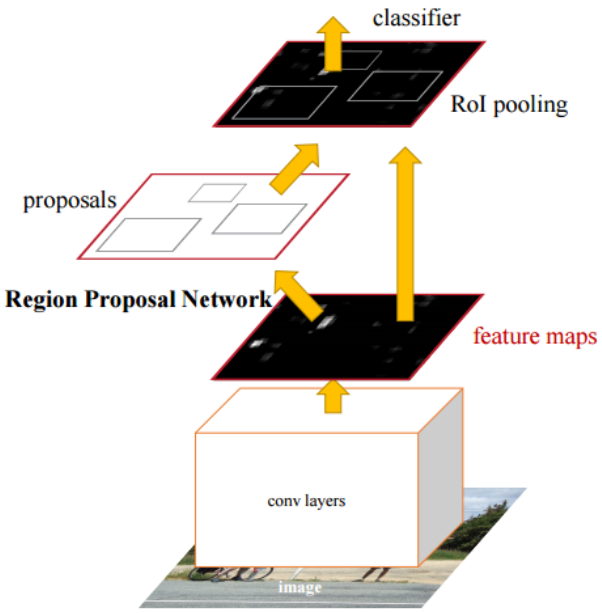

Figure 3.3: Faster Region Based Convolutional Neural Network consists of a Fast R-CNN and a region proposal network (RPN). Fast R-CNN is responsible for the feature map extraction from the whole image, and classify regions from that. The RPN is an in-network implemented proposal system, for generating candidate object locations in a fast way.

Earlier versions of the system suggested to train the complete system in an alternating way. First the RPN will be trained with a CNN, pretrained on e.g. the ImageNet dataset [Den09]. The trained RPN will then be used to propose regions, and train the Fast R-CNN part with them. The CNN of the fine-tuned Fast R-CNN is then used to train the RPN again, and this process is repeated iteratively. Later, it has been proven, that training the whole network of the faster R-CNN can be performed in an end to end manner, with marginal performance drop. This means, that all the parts of the system can be trained jointly.

4

Logo Retrieval System

The difficulties of the problem logo retrieval were highlighted in the introduction, in chapter 1. Nowadays, deep neural networks are utilized to solve such difficult computer vision problems. In this work, end to end neural network solutions will be used, to retrieve logos from videos. The set of logos to be searched is called the query set. Furthermore the search set is composed of the frames of videos, where the logos should be retrieved from.

4.1 Logo Datasets

The hunger of deep learning methods for training data is well-known. As the publicly available logo datasets are relatively small, a better training result can be achieved, if the datasets are merged. The different logo datasets with the number of brands, images and bounding box RoIs can be seen in table 4.1. The total number of brands means the number of different brands altogether.

There is also a trademark dataset available having a much greater cardinality, called METU Trademark [Tur17]. However, the images of this dataset contain only the logo of a company, without any context. This dataset turned out to have no use for region based deep learning methods, since this approach needs to learn to distinguish between objects to be learned and the background. The network was trained with the fusion of FlickrLogos-32 and METU trademark dataset. The training setup is described in section 4.4. Then it was tested with the evaluation method of py-faster-R-CNN [Gir17] [Ren15], described in chapter 5.

However, this training data is insufficient to train a network from scratch (with randomly initialized weights). Girshick et. al. showed [Gir13], that initializing the weights of the network from a CNN, which was trained on a not related dataset, and fine-tuning that on the target dataset, can boost on performance significantly. It is because of the hierarchical learning of shapes by the layers of the convolutional neural network. As a result the learning of the first several convolutional layers can even be turned off during fine-tuning. Thus, for all the training in this work, the weights of models are initialized from a network, pretrained on ImageNet classification dataset [Den09].

For train and evaluation purposes, there were four datasets created, by extracting them from sport videos. These data are needed, to be able to finetune the networks for the specific context. All the logos of these images were annotated despite occlusion and bad sight of them, along with company name. The collected datasets are summarized on table 4.2. The utilization of the datasets (training or evaluation) are indicated too. The one, used for testing, is from a different TV broadcasting company then the other three sets.

	Number of brands	Number of logo images	Number of RoIs
BelgaLogos [Jol09], [Let12]	37	1321	2697
FlickrBelgaLogos [Let12]	37	2697	2697
Flickr Logos 27 [Kal11]	27	810	1261
FlickrLogos-32 [Rom11]	32	$70 \cdot 32 = 2240$	3404
Logos-32plus [Bia17], [Bia15]	32	7830	12300
TopLogo10 [Su16]	10	$10 \cdot 70 = 700$	863
Total	**80 (union)**	15598	23222

Table 4.1: Publicly available logo datasets with bounding box annotations

	Phase	Number of brands	Number of logo images	Number of RoIs
Football-1	Train	104	331	3329
Ski		27	179	701
Ice hockey		19	410	3920
Football-2	Test	43	287	2143

Table 4.2: Collected logo datasets from sport videos

4.2 Logo Detection

There are a lot of possibilities to search for objects in an image as explained in section **??**. Keypoint detectors and external proposal systems are translation and rotation invariant, but usually these systems cannot be trained on a specific dataset. Girshick et.al. proposed the Faster R-CNN neural network, detailed in section 3.4.3, for end to end learning to detect and classify objects on an image. This network has a bounding box regressor for each trained class, thus it is capable to produce object type specific region proposals.

In the following sections, different detectors will be introduced.

4.2.1 Region Proposal Network

During the training of a Faster R-CNN network, a region proposal network will be trained to detect all the kind of objects, which it was trained on. Thus, after training a faster R-CNN with different logos, the trained RPN can be used alone as a logo detector. It has the advantage, that the detector can be extracted from every already trained faster R-CNN network, not like those in case of the following solutions.

4.2.2 Class Agnostic Faster R-CNN

Faster R-CNN can be trained for two classes: background and logo. For this purpose the classes of every annotation box will be neglected. This solution is expected to yield better performance then the RPN detector. Firstly, because of the fully connected layers preceding the final classifier and the bounding box regression layers. Secondly, it is a cascade of two detectors (RPN and FC), similarly as in [Vio04], for which it is expected to have a lower false positive rate.

4.3 Logo Comparison

After a logo is detected, a correspondence from the query set should be found. In order to retrieve as much objects from images as possible, the detectors should work with a high recall. Although, for difficult tasks, like open-set logo detection, high recall value induces low precision, thus a lot of false positive possible object locations are produced. These examples should be eliminated by the classifier.

However, in case of image retrieval the goal is not direct classification, but rather feature extraction of an RoI. Thus, the features of the logos of all the query images and the search set should be collected, which can be solved in several ways.

4.3.1 Faster R-CNN feature extraction

The features of both the query and the search set can be extracted by running a faster R-CNN network on them in a standard way, and utilizing the output vectors of the last or the second last layer as features. The advantage of this solution is its speed. This is the fastest solution among the detailed ones. But there are several drawbacks. It has low performance, because of the unknown classes, especially if the net is trained for a small number of classes. This results in a low dimensional feature vector e.g. 32 after training with FlickrLogos-32, by using the class probabilities as feature which yields often the best descriptor. A second issue, which discourages from applying this method for open-set retrieval, occurs during running the network on the query examples.

The region proposal network outputs more hundred possible object locations (default is 300 in test phase) for every input image. Processing so many query features would immensely increase the computational burden. Thus, it is advantageous to filter the detection list. Although, this cannot be done with the classification scores, since the images contain logos from unknown brands, thus may having an ensemble of brands as descriptor. For this purpose, the score output of the RPN can be used, which is an objectness indicator. This score can be adopted for the detections of the search images too. Therefore, one can take the RoI with the greatest score for query images, and set a low threshold for images of the search set.

However, there is often the case that the network does not estimate the complete logo with the greatest score, but only a part of that (e.g. the Registered Trade Mark symbol in figure 4.2). This mislocalization destroys the retrieval of that entire class.

Figure 4.1: Misplaced logo detection, with maximum RPN score

4.3.2 Fast(-er) R-CNN feature extraction

The drawbacks of the solution in section **??** imply, that the RPN should be turned off for the examples of the query set. This means, that the network is applied in fast R-CNN mode on one location, including the complete query image. However, it may yields worse descriptors, because of the loosely fitting bounding box. On the other hand, the logo positions and their features of the search set can be inferred with faster R-CNN, and filtered with a threshold as described in section **??**. The retrieved

vectors are then normalized and the similarities of them are calculated with cosine distance, which calculates the cosine value of the angle between two vectors. This distance is then used as probability score of the correspondence to a query image. The detections on a particular location, not having the largest score, are eliminated by non-maximum suppression, which searches for other bounding boxes with a minimum IoU of 0.3.

4.3.3 Logo Detector and Fast R-CNN feature extraction

The system in the section **??** may suffer from the inability of the RPN to localize unknown logos. The solution can be further improved, by using the best logo detector from the section 4.2. The output of the detector is then treated as external proposals, and thus the system run in fast R-CNN mode for the images of the search set. This setup is very similar to the one of fast R-CNN, detailed in section 3.4.2, but utilizing rather neural networks instead of selective search as external region proposal system to collect object locations.

4.3.4 Logo Detector and General feature extraction networks

Section 3.1 details the evolution of convolutional networks by going through the most important ones of today. Donahue et al. proposed [Don13], that convolutional networks can produce excellent descriptors of the input image, regardless of the absence of fine-tuning to the specific context of the image. For this purpose, a network is pretrained on very large datasets, after which it can be deployed for a broad set of computer vision problems. But it cannot localize objects on an image. For region proposal purpose, a trained logo detector can be utilized from section 4.2. This setup of networks is very similar to the R-CNN, detailed in section 3.4, but omits the use of selective search as external region proposal system. As such,it has the disadvantage of retrieving features of the regions separately from each other, by not sharing the computed feature maps. Thus it needs much more time, to infer a complete image. On the other hand, it is beneficial for the performance, because the complete network is only focused on a specific region. This part of the system can be easily swapped to reach the desired performance / time constraints, since all the kind of networks, explained in section 3.1 can be utilized for feature extraction.

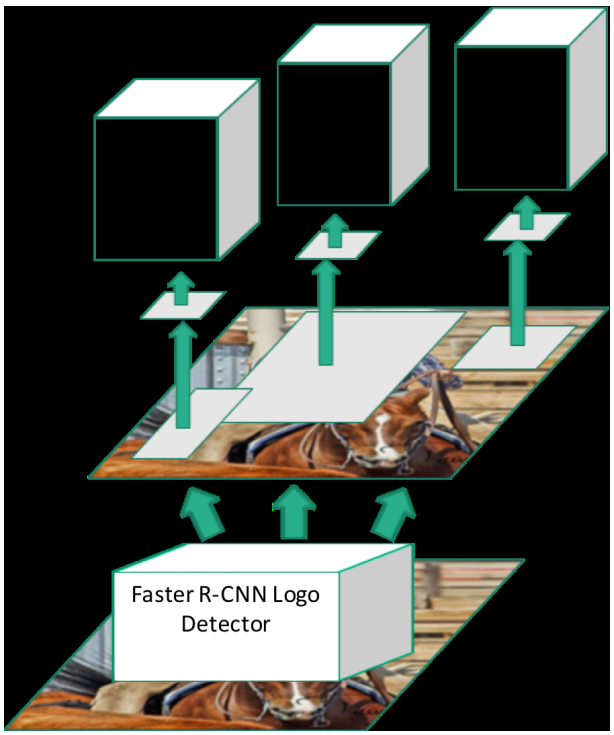

Figure 4.2: Faster R-CNN and Classifier based logo retrieval system

4.4 Jointly Trained Detector and Classifier

Both a class agnostic detector (section 4.2.2) and a classifier can be built together in a faster R-CNN network. For this purpose, either all fully connected layer or only the two at the end of the network, responsible for classification and bounding box regression should be duplicated. In doing so, one branch will be trained with brand label, the other only with logo object indication. In this network, the weights of the feature extraction layers and the region proposal network can be shared between the two tasks. The easiest way to train such a network is a siamese like faster R-CNN.

Siamese networks [Had06] are basically used for calculating similarity scores between inputs. If it is combined with an appropriate loss function e.g. contrastive loss [Had06] or max margin loss [Sim13][Her16], the network can be trained with an image pair and a label indicating whether the objects on the images are from the same category or not. The network than learns to project objects from the same category with a low-, otherwise with a large distance to each other, according to a specified metric. This is achieved by sharing the weights of the feature extraction layers.

For faster R-CNN, the parameters for region proposal network can also be shared. This setup has the advantage, that training of one task can also improve the performance of the other task, which is not currently trained. In particular for logo retrieval, the network can benefit from a bounding box annotated logo dataset, without specific brand indication. To annotate such a dataset it is much less human resource needed. A quantitative evaluation can be found in section 5.2. A schematic illustration of the training and test setup can be seen in figure 4.3.

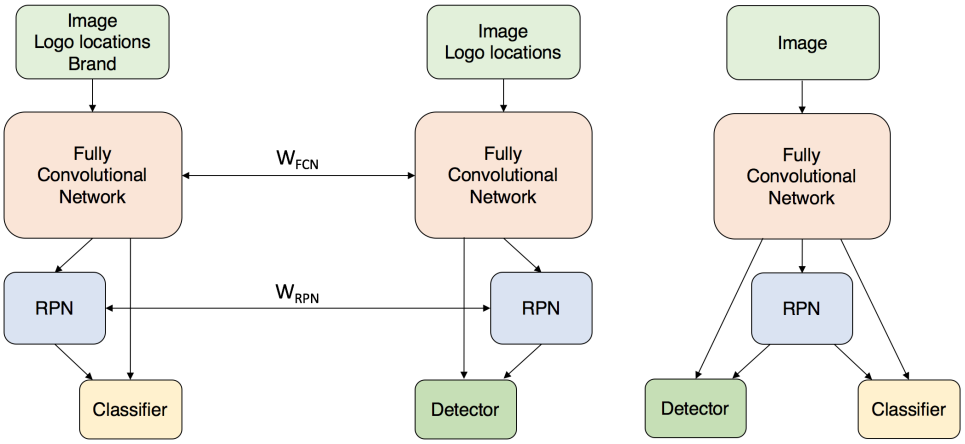

Figure 4.3: Network setups in train and test phases for learning detection and classification jointly

5

Experiments

Precision and recall are favoured values in image retrieval. Precision is the fraction of the number of relevant retrieved objects and the number of all the retrieved objects. Recall is the ratio of the number of relevant retrieved objects to the number of all the relevant objects. Although, many retrieval systems are capable to return a ranked list of the retrieved objects, precision and recall ignore this information. Thus, the trained models are evaluated with the nowadays very popular mean average precision (mAP) metric.

All the mentioned results are then calculated with the evaluation implementation of py-faster-R-CNN [Gir17] [Ren15]. Firstly, it creates a descending sorted list for every company (class), based on the probabilities of being logos from the specific brand on given positions of the images. Then, the precision curve, as a function of recall, is acquired for every list. The average precision is then calculated as the area under the curve. The average of these values gives the mean average precision.

The models trained for logo detection are evaluated beside of mAP, with free-response receiver operating characteristic (FROC) curve [Mil69]. This metric was first used for cancer localization in medical images. On this curve the detection rate aka. recall (the fraction of the number of the true positive detections and the number of all the positive locations in the dataset) is plotted over the average number of false detections per image. Since the detectors should be optimized to have a recall, as high as possible, this curve gives an intuitive interpretation about the performance of a detector.

If not stated otherwise, all the models are trained for 80k iterations with a base learning rate of 0.001, which is reduced to its one-tenth after every 50k iterations. All the training and testing are performed in Caffe deep learning framework [Jia14].

5.1 Training with Synthetic Data

In this section, the effect of synthetic data to the logo detection performance will be examined. For this purpose, the detector was trained on existing and generated datasets.

Synthetic data was already used to test its impact to logo retrieval by [Su16]. However, they reported the performance improvements, by extending a very scarce real training data with synthetic data (10 training images of FL-32 pro class). It is questionable whether the synthetic images would have helped so much if more real training data had been used (e.g. 40 images pro class, by using the validation set of FL-32 too).

5.1.1 FlickrBelgaLogos dataset

A dataset, which is annotated manually, may contain logos, which stay unannotated. If a system is evaluated on such a dataset, and detects the unannotated logo, it counts as false positive. Thus, a synthetic dataset, called FlickrBelgalogos [Let12] is created for evaluation purpose by pasting the logo annotations of the dataset BelgaLogos [Jol09] on images from Flickr to random positions. One could argue with the correctness of evaluating a detector with this dataset, because alone the contrast difference may make the logos easier to detect on these images.

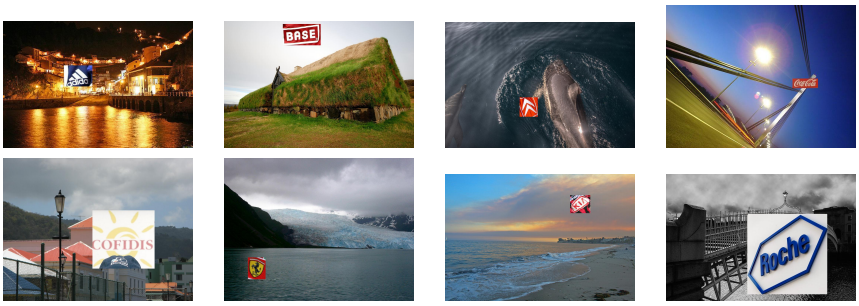

Figure 5.1: FlickrBelgaLogos examples

This dataset was evaluated for training purposes. Therefore a small subset of BelgaLogos was chosen as test set. The logos on these images were leave out from the FlickrBelgaLogos dataset. The rest of the images is used as the train set to train a Faster R-CNN model with the 37 classes of BelgaLogos. The trained models are tested on the chosen test set of BelgaLogos. Then the train set of BelgaLogos was also trained with the same network, to compare the results. After that the datasets were fused to examine the possibility of achieving better performance. Lastly, the model was trained with curriculum learning [Ben] (CL) as it was done with logos in [Su16]. CL is a learning process, whereas the examples are gradually becoming more difficult during training. In this context, it is realized by training the network first merely with synthetic logos and then with real images.

The synthetic dataset alone could achieve moderate results, the obvious advantage of a real dataset can be seen in figure 5.2. Unfortunately, the fusion of datasets does not bring extra performance, neither with a simply fusion, nor with curriculum learning. The latter has the advantage of achieving convergence much earlier, compared to other training scenarios. In [Su16], 10% relative extra performance was achieved by training with CL, to recognize a restricted number of classes. In case of FlickrBelgaLogos, this is probably not successful, because the same logos of BelgaLogos are reused in FlickrBelgaLogos. This means, while trying to achieve better performance, the transfer of logos to another context does not give additional information. As base network the `VGG_CNN_M` [Cha14] was chosen, pretrained on ImageNet [Den09], because of its much shorter training times, compared to VGG-16 (about a quarter as much time), still having a performance good enough for the experiments.

The different points of the curves are collected by moving the decision boundary threshold on the scores of a region being object or not in the interval [0.01; 1), with 0.01.

After that, the open-set logo detection capability of the model, trained only on FlickrBelgaLogos, was tested. A faster R-CNN was trained purely for logo detection, without logo classes. For evaluation, a self annotated dataset of a sport video was used, that has logos merely from such companies, with which the net has not been trained before. Despite the small size and the unreality of this dataset, it is able to generalize from the learned logos and detect some of the logos unknown for the network. The detection performance evaluation can be seen in figure 5.3 as well as an example detection on 5.4

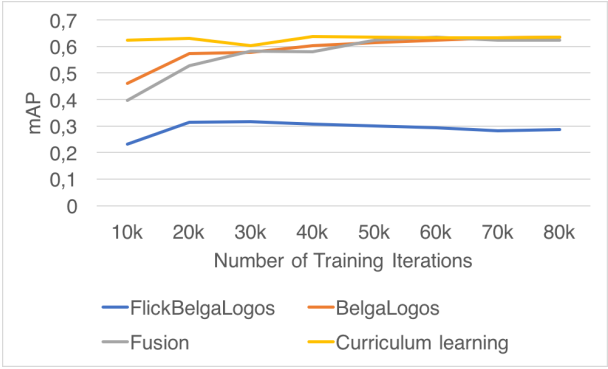

Figure 5.2: Logo recognition performance after training with real (BelgaLogos) and synthetic data (FlickrBelgaLogos)

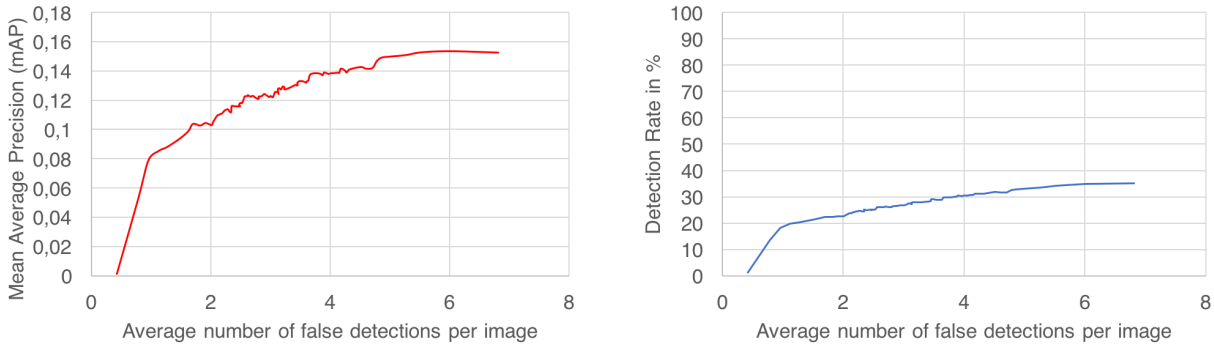

Figure 5.3: Logo detection performance

Figure 5.4: Logo detection example

5.1.2 METU Trademark dataset

In order to extend the training dataset, a synthetic dataset was generated, where one logo from the METU Trademark dataset [Tur17] was placed on an image. As basis, photos from Tripadvisor were used to fulfill the requirement of being practically logo-free. The majority of the logo's background has a white color. But logos usually do have some background, other than white. Therefore some transformations were applied on the logo images before as follows. One third of the dataset was left unchanged. The brightness of the rest of them was adjusted to the brightness of the image, on which the logo is placed. Furthermore for one third of the logos, the mean HSV value of each logo was calculated and it was rotated with 90 degree chosen randomly. In addition, Gaussian blur was applied on the edges of the logos, in order to suppress the sharp contrast changes. The table 5.1 summarizes the applied transformations.

	Brightness adjustment	**Hue rotation**
33%	-	-
33%	yes	yes
33%	yes	yes

Table 5.1: Applied transformations

The created dataset is then used to train a two class faster R-CNN for logo detection. The base network is again `VGG_CNN_M`, as in section 5.1.1. For evaluation, the same self annotated sport video dataset was used, as in section 5.1.1. Unfortunately, the trained network could not yield appropriate performance. The problem is probably that the dataset consists of logos mainly with a white background and a black text on them. The other source of problem could be the unreal appearance of the logos themselves or the transformations applied to them. Figure 5.5 shows some generated examples.

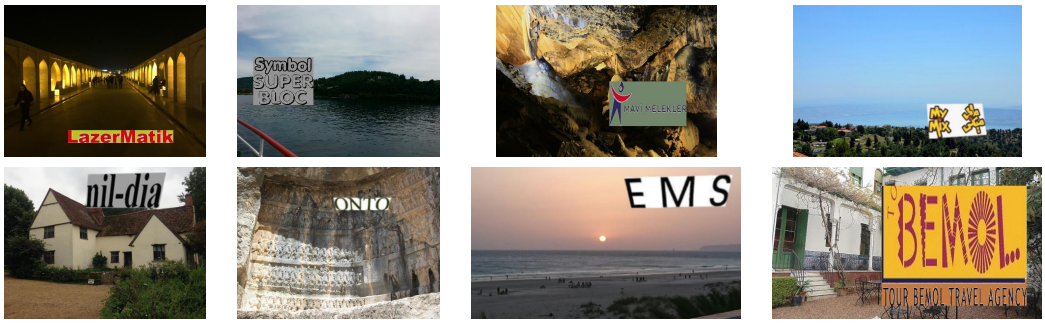

Figure 5.5: Generated synthetic logo images

5.1.3 Synthetic data with shape based logos

As section 5.1.2 shows, the mainly text based logos are incapable to train a model to detect real logos. In order to successfully extend the training dataset, logo images with more shape and color [Log] were collected. The logos were simply copied and pasted to the same Tripadvisor dataset, as in section 5.1.2. Unfortunately, also this dataset was not able to train a model to generalize and to recognize real logos. As a next trial, the white background of the logos was set to transparent, as suggested in [Su16], but in the context of open-set logo detection, it does not appear to be useful either. The success of FlickrBelgaLogos, seen in section 5.1.1, was probably because the logo and its direct surrounding come from a real scene.

Figure 5.6: Generated synthetic logo images

5.2 Logo Detection

Although there is not any known earlier work in the field of open-set logo detection and retrieval, it was still attempted to take a solution as baseline, which is already published. As section **??** details, there exists a lot of research in improving the retrieval performance on the FlickrLogos-32 dataset using faster R-CNN. Thereby, the region proposal network of a faster R-CNN is chosen for logo detector baseline, which is trained only on the train and validation set of FlickrLogos-32. The network itself achieves state-of-the-art performance on the test set of the dataset. In particular with VGG-16 as base network, it has 85.4 mAP, whereas the best already published result is 84.2 mAP proposed by [Bao16], using the multiscale fast R-CNN approach and AlexNet as base network.

All the results of the following experiments are compared in figure 5.7. The data points are gathered by moving the threshold value on the region objectness probability emitted by the network. Firstly, the RPN of the baseline network is evaluated on very challenging, self-annotated images, extracted from a football video. Afterwards, the network is trained on all the publicly available logo datasets with bounding box annotations, introduced in section 4.1, and the improvement of the RPN network is tested. This network has already a better recall performance, but it retrieves much more false locations for lower threshold values. Next, a class agnostic faster R-CNN network is trained again on all public datasets, but now only with "logo" class. Both the RPN and the classifier with regression layer at the end of the network are tested. The latter output is referred as "FC" on the figure 5.7, because of the fully connected layers preceding them. A quite interesting result is here, that by training the same model with the same dataset without the specific brand labels, the performance of the region proposal network improves (see "Public Datasets RPN" versus "Public Datasets Class Agnostic RPN").

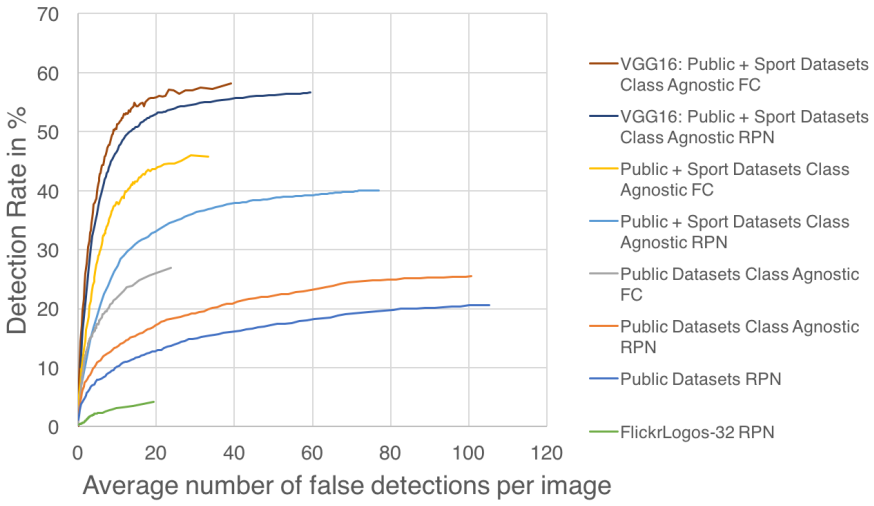

Figure 5.7: Logo detection evaluation

In order to fine-tune the networks for the specific task, the datasets from similar context were used, introduced in section 4.1. The training is saturated after about one epoch learning. This means, that the initial network is good pretrained for the task. This data has a large effect on the performance.

After training with so many logo brands, it is hard to find a sport video, with logos, which cannot be already found in the training set. E.g. Adidas is one of the brand, which occurs both in the training set and with a large number on the video. Nevertheless, the effect of that is questionable, because Adidas logos are coming from the datasets FlickrLogos-32 and Logos32Plus. The networks, trained already on these datasets achieve much weaker results.

Finally, a VGG-16 based faster R-CNN was trained on all the training data, used earlier. Although, the gap between the RPN and the FC logo detector of this network became smaller, due to the much deeper base network, before the RPN, the FC solution yields still superior performance both in recall and precision. One could argue, that the marginal improvement of FC to RPN does not worth the increased computational complexity. However, the RPN network has about double as much false positive on the same level of recall. This can cause much more additional computational burden for the classifier, then what the FC layers do, especially if it needs large computational load.

The VGG-16 detector can generalize quite well, as figure 5.9 shows it is able to operate under extreme light conditions too.

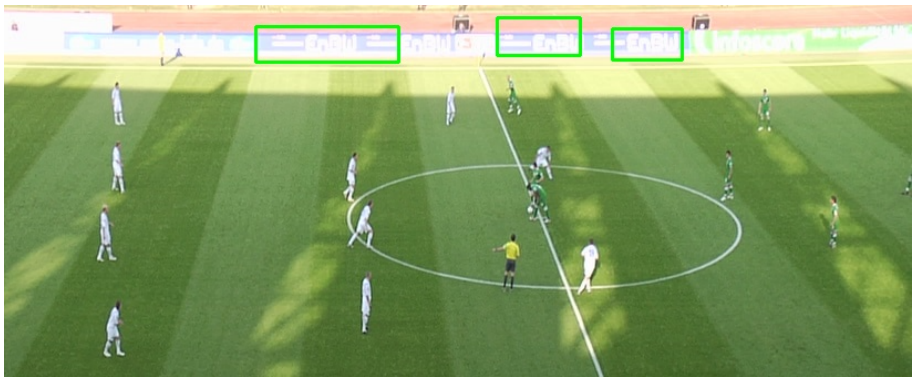

Figure 5.8: Logo detection under extreme light conditions

5.3 Evaluation on FlickrLogos-32

The effect of additional data and different architectures to the performance on the test set of FlickrLogos-32 (FL-32) is examined. The trained networks are the following ones:

1. A faster R-CNN, with `VGG_CNN_M_1024` as base network, trained on the training and validation set of FL-32.

2. A jointly trained detector and classifier as introduced in section **??**, whereas the impact of further by training the classifier the same way as in point 1, and the detector on every other public datasets without specific brand label.

3. A faster R-CNN is trained, based on `VGG_CNN_M_1024`, trained on the same datasets as the detector and classifier together in point 2, but this time with complete brand specification.

4. Point 1 and 2 is repeated with VGG-16

5. A faster R-CNN, trained on all public datasets, and annotated sport videos

The test set of FL-32 is evaluated with the configurations from points 1, 2 and 3, and compared in figure **??**. The networks are evaluated based on the mAP. It can be seen, that additional logo dataset, even without brand indication can be utilized, to detect and recognize known logos with better performance, by using the Siamese-like architecture, proposed in **??**. The network, trained on complete class information starts with a worse performance. It is because of the more number of iterations, needed to be able to distinguish between the much more learned classes. Later on, it has obviously superior performance.

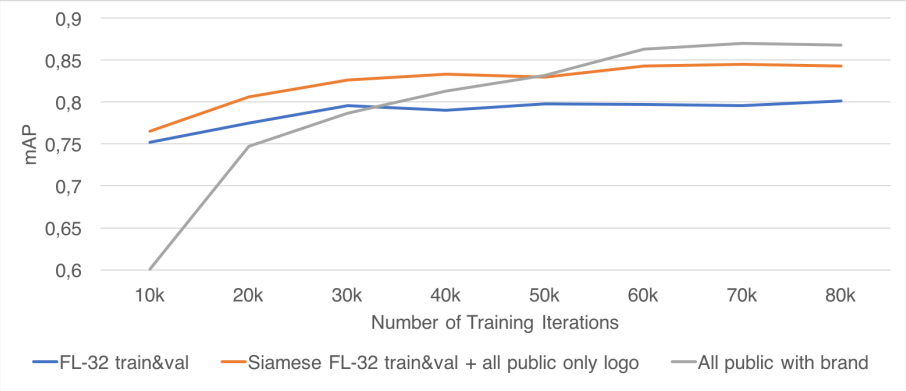

Figure 5.9: FL-32 test evaluation with `VGG_CNN_M_1024` base network

The improvement of additional data on the performance is then tested with VGG16 base network, having more capacity. Due to the more data, it achieves naturally state-of the-art performance on FL-32 test set.

5.4 Logo Retrieval

After the best logo detector is found, the system should decide, which detections are relevant, and classify them, to the instances of the query set.

As a baseline system for logo retrieval, the solution detailed in section **??** was chosen, because this system holds the state-of-the-art performance in closed set logo retrieval. It consists of a faster R-CNN, trained for logo detection on the train and validation set of FL-32.

As detailed in section **??**, this solution may have a hard time detecting the complete logo, especially if the logo has more distinct parts. The effect of mislocalization and the low number of output features probabilities of the last layer is further investigated. To this end, the performance of the class probabilities was qualitatively tested. In particular, three logo images were tested from the same, unknown brand, having lesser or greater appearance variation:

- a logo, cropped from a video
- a logo with very similar background color, but in high resolution
- a logo with a completely different color scheme also in high resolution

The logos fill out the majority of the images as figure **??** shows. The assumption was, that despite the unknownness of the brand, the descriptor features of the three logos will contain similar portion of probabilities of known brands, thus having low cosine distance to each other. Unfortunately, the network was incapable to yield such feature vectors, the logos and the associated logo from the known set with the greatest probability can be seen in figure 5.11. The composition of the normed features for the three query images is shown by figure **??**

Joint

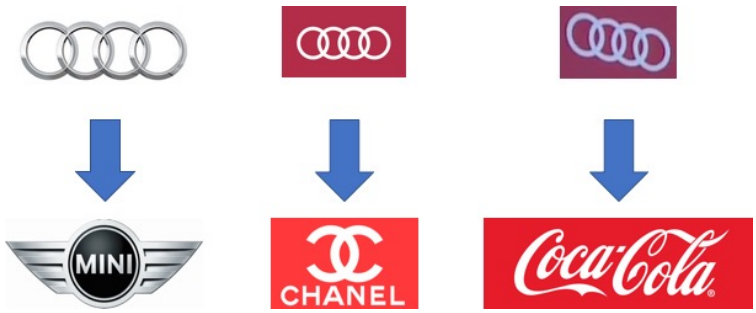

Figure 5.10

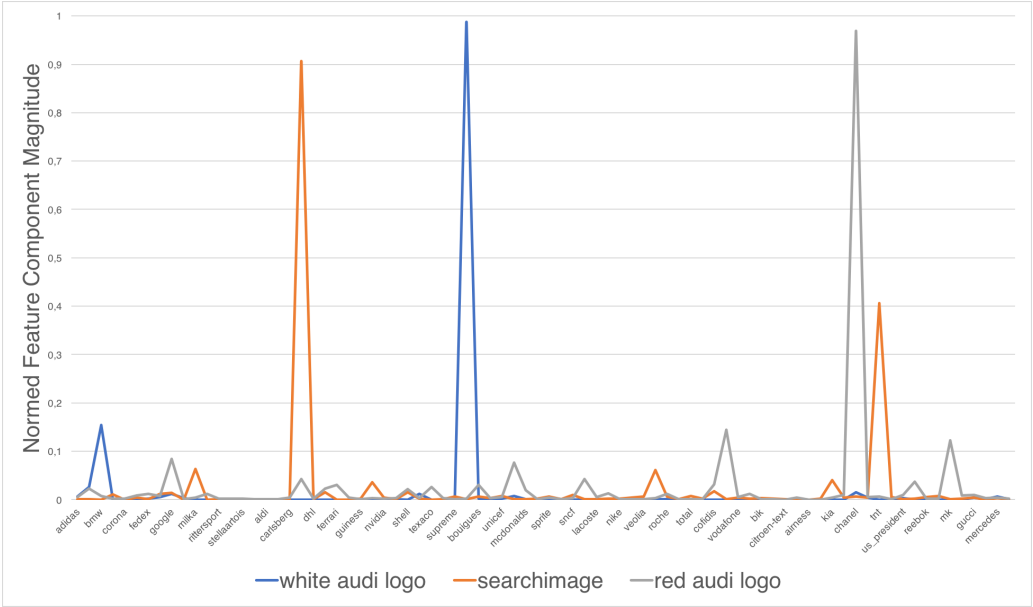

Figure 5.11

General convnets The proposed bounding box regions are warped to 224x244 pixels, which is the conventional input size of these networks. No padding were used for the crops, so the aspect ratio was not preserved. This should not induce errors, because the query images undergo this aspect ratio change too.

The retrieval performance was then tested, with query images, cropped from the video. But because of the are large intraclass variances it still a challenging task for the classifier. Figure 5.12 shows groundtruth examples.

Figure 5.12: Samples for intraclass variation in one sport video. Images from same columns have the same class.

Occluded images are not included in the test set, because perhaps neither a TV-viewer would find the correspondence for unknown logos. Images, which have an edge size both in width and height

smaller or equal than 25 pixel are also omitted, because those escape the viewer's attention too very likely. In addition, classes, having only one example were removed, because in this case the query set would cover the groundtruth.

The similarities between the extracted feature vectors were calculated by cosine distance.

Audi logo crop classification highres, crop, white

Classification performance by chance

5.5 Retrieval times

6

Conclusion

6.1 Summary

6.2 Future Work

Bibliography

[Bao16] BAO, Yu; LI, Haojie; FAN, Xin; LIU, Risheng und JIA, Qi: Region-based CNN for Logo Detection, in: *Proceedings of the International Conference on Internet Multimedia Computing and Service*, ICIMCS'16, ACM, New York, NY, USA, S. 319–322, URL http://doi.acm.org/10.1145/3007669.3007728

[Ben] BENGIO, Yoshua; LOURADOUR, Jérôme; COLLOBERT, Ronan und WESTON, Jason: Curriculum Learning

[Bia15] BIANCO, Simone; BUZZELLI, Marco; MAZZINI, Davide und SCHETTINI, Raimondo: Logo recognition using cnn features, in: *International Conference on Image Analysis and Processing*, Springer, S. 438–448

[Bia17] BIANCO, Simone; BUZZELLI, Marco; MAZZINI, Davide und SCHETTINI, Raimondo: Deep learning for logo recognition. *Neurocomputing* (2017), Bd. 245: S. 23–30, URL http://www.sciencedirect.com/science/article/pii/S0925231217305660

[Cha14] CHATFIELD, K.; SIMONYAN, K.; VEDALDI, A. und ZISSERMAN, A.: Return of the Devil in the Details: Delving Deep into Convolutional Nets, in: *British Machine Vision Conference*

[Cor95] CORTES, Corinna und VAPNIK, Vladimir: Support-Vector Networks. *Mach. Learn.* (1995), Bd. 20(3): S. 273–297, URL http://dx.doi.org/10.1023/A:1022627411411

[Dal05] DALAL, Navneet und TRIGGS, Bill: Histograms of Oriented Gradients for Human Detection, in: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, IEEE Computer Society, Washington, DC, USA, S. 886–893, URL http://dx.doi.org/10.1109/CVPR.2005.177

[Den09] DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K. und FEI-FEI, L.: ImageNet: A Large-Scale Hierarchical Image Database, in: *CVPR09*

[Don13] DONAHUE, Jeff; JIA, Yangqing; VINYALS, Oriol; HOFFMAN, Judy; ZHANG, Ning; TZENG, Eric und DARRELL, Trevor: DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *CoRR* (2013), Bd. abs/1310.1531, URL http://arxiv.org/abs/1310.1531

[Eve] EVERINGHAM, M.; VAN GOOL, L.; WILLIAMS, C. K. I.; WINN, J. und ZISSERMAN, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results, http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html

[Gir13] GIRSHICK, Ross B.; DONAHUE, Jeff; DARRELL, Trevor und MALIK, Jitendra: Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR* (2013), Bd. abs/1311.2524, URL http://arxiv.org/abs/1311.2524

[Gir16] GIRSHICK, Ross; DONAHUE, Jeff; DARRELL, Trevor und MALIK, Jitendra: Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* (2016), Bd. 38(1): S. 142–158, URL http://dx.doi.org/10.1109/TPAMI.2015.2437384

[Gir17] GIRSHICK, Ross: py-faster-rcnn, https://github.com/rbgirshick/py-faster-rcnn (2017)

[Had06] HADSELL, Raia; CHOPRA, Sumit und LECUN, Yann: Dimensionality reduction by learning an invariant mapping, in: *In Proc. Computer Vision and Pattern Recognition Conference (CVPR?06*, IEEE Press

[He14] HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing und SUN, Jian: Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *CoRR* (2014), Bd. abs/1406.4729, URL http://arxiv.org/abs/1406.4729

[He15] HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing und SUN, Jian: Deep Residual Learning for Image Recognition. *CoRR* (2015), Bd. abs/1512.03385, URL http://arxiv.org/abs/1512.03385

[Her16] HERRMANN, Christian; WILLERSINN, Dieter und BEYERER, Jürgen: Low-Quality Video Face Recognition with Deep Networks and Polygonal Chain Distance, in: *Proceedings of the 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, IEEE, Gold Coast, Australia, S. 244–250

[Hin12] HINTON, Geoffrey E.; SRIVASTAVA, Nitish; KRIZHEVSKY, Alex; SUTSKEVER, Ilya und SALAKHUTDINOV, Ruslan: Improving neural networks by preventing co-adaptation of feature detectors. *CoRR* (2012), Bd. abs/1207.0580, URL http://dblp.uni-trier.de/db/journals/corr/corr1207.html#abs-1207-0580

[Hua16] HUANG, Gao; LIU, Zhuang und WEINBERGER, Kilian Q.: Densely Connected Convolutional Networks. *CoRR* (2016), Bd. abs/1608.06993, URL http://arxiv.org/abs/1608.06993

[Iof15] IOFFE, Sergey und SZEGEDY, Christian: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *CoRR* (2015), Bd. abs/1502.03167, URL http://arxiv.org/abs/1502.03167

[Jia14] JIA, Yangqing; SHELHAMER, Evan; DONAHUE, Jeff; KARAYEV, Sergey; LONG, Jonathan; GIRSHICK, Ross; GUADARRAMA, Sergio und DARRELL, Trevor: Caffe: Convolutional Architecture for Fast Feature Embedding, in: *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, ACM, New York, NY, USA, S. 675–678, URL http://doi.acm.org/10.1145/2647868.2654889

[Jol09] JOLY, Alexis und BUISSON, Olivier: Logo retrieval with a contrario visual query expansion, in: *MM '09: Proceedings of the seventeen ACM international conference on Multimedia*, S. 581–584

[Kal11] KALANTIDIS, Y.; PUEYO, LG.; TREVISIOL, M.; VAN ZWOL, R. und AVRITHIS, Y.: Scalable Triangulation-based Logo Recognition, in: *in Proceedings of ACM International Conference on Multimedia Retrieval (ICMR 2011)*, Trento, Italy

[Kri12] KRIZHEVSKY, Alex; SUTSKEVER, Ilya und HINTON, Geoffrey E: ImageNet Classification with Deep Convolutional Neural Networks, in: F. Pereira; C. J. C. Burges; L. Bottou und K. Q. Weinberger (Herausgeber) *Advances in Neural Information Processing Systems 25*,

Curran Associates, Inc. (2012), S. 1097–1105, URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[LeC89] LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W. und Jackel, L. D.: Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.* (1989), Bd. 1(4): S. 541–551, URL http://dx.doi.org/10.1162/neco.1989.1.4.541

[Let12] Letessier, Pierre; Buisson, Olivier und Joly, Alexis: Scalable mining of small visual objects, in: *Proceedings of the 20th ACM international conference on Multimedia*, ACM, S. 599–608

[Log] Clearbit - Free logo API, https://clearbit.com/logo, accessed: 2017-02-17

[Lon14] Long, Jonathan; Shelhamer, Evan und Darrell, Trevor: Fully Convolutional Networks for Semantic Segmentation. *CoRR* (2014), Bd. abs/1411.4038, URL http://arxiv.org/abs/1411.4038

[Low04] Lowe, David G.: Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision* (2004), Bd. 60(2): S. 91–110, URL http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94

[Mil69] Miller, Harold: The FROC Curve: a Representation of the Observer's Performance for the Method of Free Response. *The Journal of the Acoustical Society of America* (1969), Bd. 46(6(2)): S. 1473–1476

[Ren15] Ren, Shaoqing; He, Kaiming; Girshick, Ross und Sun, Jian: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, in: C. Cortes; N. D. Lawrence; D. D. Lee; M. Sugiyama und R. Garnett (Herausgeber) *Advances in Neural Information Processing Systems 28*, Curran Associates, Inc. (2015), S. 91–99, URL http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf

[Rom11] Romberg, Stefan; Pueyo, Lluis Garcia; Lienhart, Rainer und van Zwol, Roelof: Scalable logo recognition in real-world images, in: *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, ICMR '11, ACM, New York, NY, USA, S. 25:1–25:8, URL http://www.multimedia-computing.de/flickrlogos/

[Rus15] Russakovsky, Olga; Deng, Jia; Su, Hao; Krause, Jonathan; Satheesh, Sanjeev; Ma, Sean; Huang, Zhiheng; Karpathy, Andrej; Khosla, Aditya; Bernstein, Michael; Berg, Alexander C. und Fei-Fei, Li: ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* (2015), Bd. 115(3): S. 211–252

[Ser13] Sermanet, Pierre; Eigen, David; Zhang, Xiang; Mathieu, Michaël; Fergus, Rob und LeCun, Yann: OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. *CoRR* (2013), Bd. abs/1312.6229, URL http://dblp.uni-trier.de/db/journals/corr/corr1312.html#SermanetEZMFL13

[Sim13] Simonyan, K.; Parkhi, O. M.; Vedaldi, A. und Zisserman, A.: Fisher Vector Faces in the Wild, in: *British Machine Vision Conference*

[Sim14] Simonyan, K. und Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* (2014), Bd. abs/1409.1556

[Siv03] SIVIC, Josef und ZISSERMAN, Andrew: Video Google: A Text Retrieval Approach to Object Matching in Videos, in: *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, IEEE Computer Society, Washington, DC, USA, S. 1470–, URL http://dl.acm.org/citation.cfm?id=946247.946751

[Su16] SU, Hang; ZHU, Xiatian und GONG, Shaogang: Deep Learning Logo Detection with Data Expansion by Synthesising Context. *CoRR* (2016), Bd. abs/1612.09322, URL http://arxiv.org/abs/1612.09322

[Sze14] SZEGEDY, Christian; LIU, Wei; JIA, Yangqing; SERMANET, Pierre; REED, Scott E.; ANGUELOV, Dragomir; ERHAN, Dumitru; VANHOUCKE, Vincent und RABINOVICH, Andrew: Going Deeper with Convolutions. *CoRR* (2014), Bd. abs/1409.4842, URL http://arxiv.org/abs/1409.4842

[Tur17] TURSUN, Osman; AKER, Cemal und KALKAN, Sinan: A Large-scale Dataset and Benchmark for Similar Trademark Retrieval. *CoRR* (2017), Bd. abs/1701.05766, URL http://arxiv.org/abs/1701.05766

[Uij13] UIJLINGS, J.R.R.; VAN DE SANDE, K.E.A.; GEVERS, T. und SMEULDERS, A.W.M.: Selective Search for Object Recognition. *International Journal of Computer Vision* (2013), URL http://www.huppelen.nl/publications/selectiveSearchDraft.pdf

[Vio04] VIOLA, Paul und JONES, Michael J.: Robust Real-Time Face Detection. *Int. J. Comput. Vision* (2004), Bd. 57(2): S. 137–154, URL http://dx.doi.org/10.1023/B:VISI.0000013087.49260.fb

[Xie16] XIE, Saining; GIRSHICK, Ross B.; DOLLÁR, Piotr; TU, Zhuowen und HE, Kaiming: Aggregated Residual Transformations for Deep Neural Networks. *CoRR* (2016), Bd. abs/1611.05431, URL http://arxiv.org/abs/1611.05431

[Zei13] ZEILER, Matthew D. und FERGUS, Rob: Visualizing and Understanding Convolutional Networks. *CoRR* (2013), Bd. abs/1311.2901, URL http://arxiv.org/abs/1311.2901

[Zit14] ZITNICK, Larry und DOLLAR, Piotr: Edge Boxes: Locating Object Proposals from Edges, in: *ECCV*, European Conference on Computer Vision, URL https://www.microsoft.com/en-us/research/publication/edge-boxes-locating-object-proposals-from-edges/

List of Figures

List of Tables

Glossary

BN Batch Normalization

CL Curriculum Learning

CNN Convolutional Neural Network

FC Fully Connected Layer

FCN Fully Convolutional Neural Network

FROC Free

GT

IoU

R-CNN

ReLU

ResNet

ROC

RoI

SPP