RETAIL SYSTEM AUTOMATION WITH MACHINE LEARNING

by

Tanjimul Ahad Asif

A project report submitted in conformity with the requirements for the degree of
Master of Science in Information Technology

Department of Mathematical and Physical Sciences
Faculty of Graduate Studies
Concordia University of Edmonton

Retail system automation with machine learning

Tanjimul Ahad Asif

Approved:

Supervisor: Saha Baidya Date

Committee Member: Date

Dean of Graduate Studies: Rorritza Marinova, Ph.D. Date

RETAIL SYSTEM AUTOMATION WITH MACHINE LEARNING

Tanjimul Ahad Asif
Master of Science in Information Technology

Department of Mathematical and Physical Sciences
Concordia University of Edmonton
2021

Abstract

Lately, technological advancement has made it possible with the accessibility of enormous annotated datasets and artificial intelligence breakthroughs to have sparked a spectacular rise of precise object recognition and analysis. This thesis specifies the development and implementation considerations of an AI-enabled deep learning-based object detection system for the grocery retail industry. This thesis aims to automate the retail experience of fruits and vegetables. In this work, a data collection of images with appropriate pixel segmentation and bounding boxes has been compiled, the relevant theory is described, and we introduce the dataset of fruits images with the experiment results for training a neural network model. The thesis demonstrates the challenges for such a solution, a recommendation for the object detection model to be used, and future work reference

Keywords

Artificial Intelligence, Machine learning, Computer Vision, Retail automation

Acknowledgments

I would like to thank my Professor, Dr. Saha, whose insightful guidelines was invaluable throughout my studies. His wise counsel helped me to complete my thesis.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this paper, we chose the task of identifying groceries for several reasons. On one side, groceries have specific categories that are hard to differentiate, like the citrus genus containing oranges and grapefruits. Thus, we attempt to create a network that experiments with how well artificial intelligence can complete the task of classifying them. Another reason is that groceries are often found in stores, so they serve as a good starting point for automating the retail experience. This is to be achieved by tracking and identifying objects that the customer puts in their physical shopping cart and then using this information to build a virtual shopping cart, potentially removing the need for a cashier to scan the items and handle payments conveniently. Finally, development steps and considerations, as well as model choice and data dependencies, are discussed, concluding with a recommendation for which object detection model should be deployed for a reliable classifier. A successful classifier requires a high-quality dataset. Unfortunately, most of the available image datasets include both the object and the noisy background. This could result in cases where changing the background causes the item to be classified incorrectly. As a result, we trained a deep neural network capable of recognizing fruits from pictures. This is preliminary research aimed at developing a classifier capable of identifying a much broader range of objects from pictures. A trained network model like this will distinguish a wide range of grocery types, making it useful in a wide range of everyday retail consumer experience scenarios.

1.1 Background

1.1.1 Background of the retail system

This study aspires to automate the retail experience through the use of machine learning in grocery shopping. Automating the retail checkout process has been there for a while now [1], and it has worked in various ways, with the most prominent way being the self-scanning method. It requires the customers to scan the items themselves, eliminating the need for a salesperson at every checkout point. However, the retail business needs to trust customers to scan all of the items properly that they are purchasing. Routine checks are undertaken to ensure that all products were scanned, but this is not done for every customer because it would contradict the aim of enhanced convenience and decreased reliance on sales assistants. Furthermore, it indicates that profit loss is 2 probable due to customers failing to scan their things accurately. In order to overcome this, technologies like the 'Amazon Go project'[2] have used different sensors to completely automate the checkout process, without the need to scan bar codes rather than using cameras, scale and depth sensors to analyze and recognize when a customer takes an item from the store. The use of various sensors improves the accuracy of such a system as environmental information is known. However, this also imposes high costs, and those investments represent a significant barrier to the small retailers operating on narrow margins and without the means of huge investments. Therefore, this project seeks to develop a prototype that only uses a computer vision to recognize each consumer purchase item.

1.1.2 Computer Vision and Object Detection

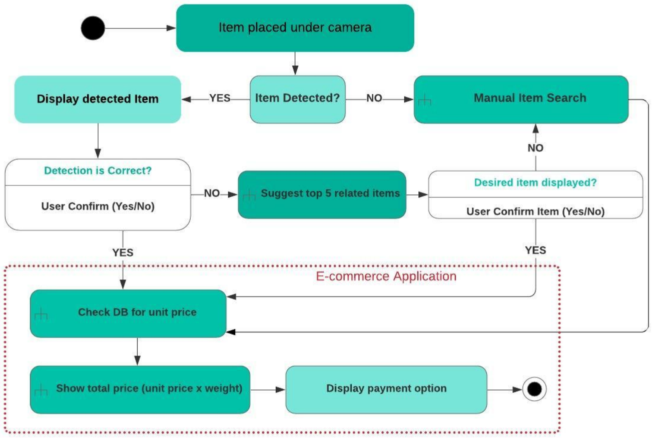

Figure 1.1: System Architecture

The system must analyze and track customer-bought items without using human input to automate the grocery retail experience. Computer vision is the technology and research area in which a computer may use images and other inputs in a pattern similar to the human visual sense to achieve a higher perception of its acting context. Recent improvements in statistical learning-based computer vision algorithms have enabled the precise detection and classification of items in a given picture. This thesis will cover a few such methods and apply them to the proposed problem formulation for the retail environment.

1.2 Aims and objectives

This thesis project's main deliverable is a functional object detection model that is efficient enough to demonstrate an automated grocery store system prototype. In addition, the model should be functioning in real-time to recognize any items purchased by the customer.

1.2.1 Model evaluation

The model will be trained and evaluated based on performance with the gathered data set to determine the architecture and parameters for object detection. The performance indicators are the same format described in Section 3.2.1 for object detection and runtime speed. In addition, an empirical assessment has also been made of conditions including FPS, amount of frame objects, and light that affect the system's precision.

1.2.2 Data gathering

Image data labelled with the correct descriptive information is necessary to use statistical learning-based computer vision models. Therefore, a data set of particular grocery products has been compiled and annotated.

1.3 Constraints and limitations

Implementing a fully automated grocery store prototype is a challenging task that falls outside the scope of this research. Therefore, a simple version based on the same technology will be developed as a proof of concept. The prototype can be regarded as an intelligent application that combines computer vision to detect the buyer's product. Thus, it improves the customer's convenience.

1.3.1 literature and intellectual property

Computer vision is a considerable aspect that is explosively increasing and promoted by both industry and the academy. The reason for this is the large variety of applications ranging from self-contained vehicles, smartphones, photo and video analysis, automation to defence systems. Even though these systems development is incredibly complex and time-consuming, the field has stayed essentially open, with plenty of research on this field and open-source code for many performing methods.

1.3.2 Hardware

The computer unit comprises an Nvidia GTX 980 GPU processor with which intensive calculations for deep learning models are processed in real-time. In future, this calculation can be done in the cloud.

Chapter 2

Literature Review

The content of the thesis implies that the reader understands how object detection works with deep learning-based computer vision systems. As a result, the most fundamental concepts in these topics will not be thoroughly described. However, the thesis will examine architectural principles and related theories that distinguish models.

2.1 Object Detection

The task of automatically predicting the existence of a particular object in a given input picture is known as object detection. The prediction should return the position and object class of the identified items. This information is frequently encoded in the image as a bounding box or pixel-precise mask with coordinates along with the associated item class. Traditional object identification algorithms focus on recognizing key points, edges, corners in an image and assembling a particular collection of features to represent an item. Feature detectors are frequently built as convolutional filters that are applied to the image that needs object detection. However, convolutional kernels were typically created to extract the necessary feature, and that was a time-consuming procedure that relied heavily on human instinct over which features would best describe a specific object.

2.2 Deep learning

Artificial neural networks produce the best image identification and classification results. Most deep learning models are built on these networks. Deep learning is a set of algorithms used by machine learning with several layers of nonlinear processing units. The data is transformed into a much more abstract and composite illustration at each level. Deep neural networks succeeded in exceeding existing methods of machine learning. In certain areas, they also accomplished the first detection of superhuman patterns. The eminence of deep learning further strengthens this as a significant step toward achieving Strong AI. Deep neural networks and convolutional neural networks have proven to be successful, particularly in image recognition.

2.3 Convolutional neural networks

Deep learning models include convolutional neural networks (CNN). A model of this type can be made up of convolutional layers, pooling layers, ReLU layers, fully connected layers, and loss layers. Each convolutional layer comprises a Rectified Linear Unit (ReLU) layer followed by a Pooling layer, then one or more convolutional layers, and eventually one or more fully connected layers in a conventional CNN architecture. Considering the structure of the images while processing them is one way of distinguishing CNN from a typical neural network. A conventional neural network transforms the input into a one-dimensional array that reduces the sensitivity of the trained classifier to changes in position. Multi-column deep neural networks are used to get some of the highest outcomes on the MNIST dataset. They employ several maps per layer with many layers of nonlinear neurons, as detailed in the study. Even though the structure of such networks makes training them more complex, it is done with graphical processors and custom code. The network's structure employs winning neurons with max pooling to decide the successes. Another study corroborates the theory that convolutional networks have improved accuracy in the field of computer vision. An all-convolutional network with improved performance on CIFAR-10 is presented in-depth in the study. The study suggests that pooling and fully connected layers be replaced by convolutional layers that are similar. Although it may be minimized by utilizing smaller convolutional layers within the network, it increases the number of parameters and inter-feature correlations that work as regularization. The layers of a CNN network will be described in detail in the sections that follow.

2.4 Deep Learning for Computer Vision

Neural networks are not a new concept and have been around since the 1970s. In recent years, significant progress has been achieved in numerous computer vision tasks because of advances in processing power, the availability of massive publicly annotated datasets, and the creation of deep neural networks. It has triggered a transformation in computer vision when it demonstrated considerably more accurate results in classification test using deep neural networks.

2.5 Object Detection Model

Although this field evolves rapidly, a few alternative architectures have been adopted and improved upon repeatedly over the years. Therefore, this project's object identification model will be based on a state of neural network architecture similar to this.

2.5.1 YOLO

YOLO is a single-stage model, unlike Faster R-CNN and Mask R-CNN [11, 12]. The RPN first proposes various sizes in the Faster R-CNN architecture based on the retrieved image attributes to present only viable and realistic boxes. Each candidate box is classified in the model's second stage. A vast number of boxes are generated statically instead of offering these boxes without regard for their viability. It leads to class imbalances for classifying the binding boxes because the vast majority of boxes are part of the background, allowing the model training to be significantly influenced while the corresponding non-background data is not learnt. The model may have a modest loss, but it still works inadequately under challenging cases. This problem can be mitigated by the complex example mining that seeks to balance the training by selecting the non-background boxes and actively picking the complex examples which match actual objects in training pictures.

2.6 Other Object Detection Model

This research has concentrated chiefly on a well-known optimized model for object identification benchmarks, but this is an area that is rapidly evolving with a high level of research interest from academics and major companies alike. As a result, several intriguing experiments are already considered, such as R-CNN and Retina Net. Work that improves the training process and data augmentation stages are also of significant interest. Additional testing of the proposed model is required, including accuracy on a dataset with more classes, validation of the model's failure cases, and runtime optimizations to decrease the computing resources required. Finally, additional labelled data is required, especially for models with several classes and scenes of higher volume.

2.6.1 R-CNN

UC Berkeley and Facebook AI Research has created and revised the Faster R-CNN architecture [7] for object identification over several years. R-CNN [8] stands for region-based convolutional network. The model accepts an image as input and outputs bounding boxes (segmentation masks in Mask R-CNN [9]) and the class label for each identified object. The R-CNN architecture's region predictions are created using Selective Search that combines pixels of similar colour and texture. For feature identification, the suggested region is fed via a CNN structure based on AlexNet [5], which also is subsequently fed through an SVM for multiple classifications. The prediction is given a class label and a confidence level by the SVM. Finally, the suggested bounding box is refined using a regressor that inputs the bounding box coordinates and the object's anticipated class. Roi Pooling is then added to this prototype to improve the architecture. Region of Interest pooling substantially decreases the number of forwarding passes required for the suggested areas by sending the entire picture across the network and storing the different convolutions. The model was also enhanced by combining the several phases into a single model. Fast R-CNN [10] makes it considerably easier to train because all of the model steps are optimized simultaneously in one phase. The Faster R-CNN [7] research was the successive refinement for this architecture that cleared the path for lowering the cost of the region proposal plod. According to Ren et al, the exciting areas in a particular image rely on the retrieved characteristics from the CNN layers. As a result, they develop a regional proposal network that takes as input feature maps calculated from CNN layers. It implies that just one CNN needs to be taught to lower the cost of the region proposal stage. Finally, pixel-by-pixel segmentation masks were added to enhance the Faster R-CNN architecture. The architecture is enhanced by adding a Fully parallel Convolutional Network that produces a binary segmentation mask. Each pixel is categorized as a one or a zero depending on whether it relates to an identified item.

2.6.2 Retina Net

This technique only helps to a certain degree, and single-stage models generally restrict border boxes to guarantee that the model continues to be trained on the relevant complex cases. The primary enhancement with Retina Net is to provide a focus loss feature to resolve the class imbalance problem[13]. The focal loss function provides an adaptive total loss by dynamically balancing the simple and more complex instances. The likelihood of the prediction being ground truth class with a lower value of pt. When pt is high, the focus loss is near zero for simple instances and does not contribute to model training. However, the Focal Loss is considerable and contributes significantly to model training when pt is less than 0.5. The focus setting controls how forceful the focal loss is when simple instances are disregarded. The focused loss becomes conventional Cross-Entropy loss when the focusing parameter is set to 0.

2.7 Segmentation and Bounding Box

The identified object's position and size are provided by bounding boxes, while a segmentation mask label provides its pixel accuracy. It enhances object detection since it is trained on more accurate data.

Chapter 3

Methodology

This section outlines the methodologies used for the data collection procedure, how this study assessed object detection and explains the experiments and reasons why they were performed.

3.1 Data collection

The dataset used in this study is known as Fruit-360 and is available on Kaggle. The dataset has 82110 images of fruits and vegetables spread across 120 labels. Each image contains a single fruit or vegetable. Separately, the dataset contains another 103 images of multiple fruits. In this study, we have used three hundred data of three different fruits for the labels and the images for training—however, the more significant number of images for training results in more accurate recognition.

3.1.1 Labelling

Each object instance in every photo was labelled with a precise class of the product. The labelling tool VoTT Image Annotator was used for labelling. The complete dataset consists of 300 images with an average of three to five objects per image. Each image is labelled with the location, size and object class for all instances of objects in the corresponding image. The labels are in the form of pixel-by-pixel segmentation and can easily be converted to bounding boxes to train the models where the output is just bounding boxes. The dataset is stored in the CSV format in excel files, and scripts for conversion to other data formats have been utilized.

3.1.2 Data augmentation

More training data leads to more accurate models for recognition. As a result, the dataset is enhanced by the addition of changed versions of the pictures. The experiment was conducted to determine whether the model trained on various data sets corresponding to various augmentations is successful. The final prototype will incorporate the augmentations that have shown to be effective in increasing model accuracy.

3.2 Evaluation

There are various methods to evaluate an object detection pattern depending primarily on specifications. Generally, a precise model is always better, yet it is always a compromise between correctness and complexity, which usually means precise models are more cognitive. Given this, the models are tested for precision, resilience, and speed when the model is running digitally.

3.2.1 Mean average precision

Precision is a metric that determines what percent of the overall projected successes are positive. The average precision for an object class is a measure of precision across all class instances in the test dataset. The mean average precision is the average precision for all object classes in the model.

3.2.2 Runtime

The running time in object detection research is less addressed when the model is the primary benchmark. The runtime of each model is essential in real-life applications. The model must run online, in a secure way, and without being vulnerable to consumers attempting to deceive the system.

3.3 Resolution requirement

Various resolution levels are evaluated to obtain the optimized resolution for our system. The images utilized in this experiment comprised sequences of things being retrieved and replaced at various resolutions. The aim is to detect the item as it passes through the detector. Because this is a reasonable approach to detect how well the system operates through object recognition. It should be noted that employing the most basic method of detecting the number of objects would eliminate the requirement for a high resolution even more. It would need the system to concurrently recognize all image elements based on the image and the number of object in each image.

3.4 Model optimization

It is advantageous to utilize an existing architecture based on the models previously stated because they have already been proved to function effectively on the standardized metrics used in research. However, the models are pretty generic, and the architecture and model parameters may be adjusted to our requirements and restrictions to adapt them to our specific application

Chapter 4

Results

This section provides the findings of the tests carried out during the system development process. The impact of data augmentation on the original dataset is given for various procedures, the system's requirement was assessed using the test, and the results of the benchmarking of the model are provided. The accuracy, runtime, and variation resilience are all included in the results. The accuracy of the model mentioned in section 2.3 is shown in the mean average accuracy of the model classes. Results of the benchmarking Convention [3] are provided. The benchmark presents correct models for various thresholds at 0.50:0.05:95 instead of standardizing accuracy at a 50 percent intersection. It also reports the typical metrics. The aim is to differentiate how well the models are constructed.

4.1 Data augmentation test

The findings to assess the influence of various augmentation methods are based on the assumption that the various augmentations will produce relative outcomes for the model. YOLO was used to test the various data sets at a resolution of 1024x720 pixels. The various augmentations were used in the following order: Scale - Resize each image and associated annotations between 50 and 150 percent of its original scale. Noise - Add gaussian noise to an image using a normal distribution sampled once per pixel. Brightness - To make a picture darker or brighter, multiply all pixels in it with 0.5 to 1.5. Shear - Shear the pictures and annotations by a factor of -20 to 20. Mirror - Vertically and horizontally mirrored pictures and annotations.

4.2 Model results

The benchmark used to assess the accuracy and runtime of the tested model was based on the metrics mentioned in Section 3.2. The benchmark results are described in the following section. The model was pre-trained on the fruit-360 dataset, and the final layers were assessed on the collected dataset. The total number of training pictures was 300, with an average of multiple instances per image.

4.2.1 Accuracy

The accuracy of the tested model is reported below, with the same test dataset. The results of the model are given as percentages.

Model	Test images	Resolution	Positive	Negative
YOLOv3	20	1920x1080	14	6
YOLOv3	20	024x720	12	8
YOLOv3	20	800x600	9	11

Table 4.1: Table to accuracy test

As seen in the table above, YOLO obtains the maximum 72 percent accuracy using the native image resolution of 1920x1080. The results are nearly equivalent at 1024x720. However, the accuracy is significantly lower at 800x600. The accuracies of the models seem to be affected roughly to the same degree when the input resolution is downscaled.

4.2.2 Failure cases

Typical model failure scenarios consist of pictures with overlap between objects, dark pictures, differentiation between two very similar items, numerous glare photos and cases of the very compact camera to the object covering significant frame sections. Handling occlusions and overlapping items is usually a complicated task, but certain failure situations might be avoided with a more extensive dataset spanning all the product dimensions. When testing the prototype in real-time, some reflection difficulties were unexpected given that the training data was collected using white backgrounded clear images. It could be easily avoided by training the model with more precise data and testing with clear images.

4.2.3 Runtime

The model's runtime is evaluated on multiple test scenarios with various levels of complexity and at different picture resolutions. The outcomes are given in milliseconds,

Model	1920x1080	1024x720	800x600	600x400
YOLOv3	127	83	71	68

Table 4.2: Table to run-time test

Chapter 5

Future Work

5.1 Future work and next steps

The findings of this study are encouraging, but there is still a lot of work and optimization to be done before the prototype can be launched. The accuracy needs to be maintained since the system's security is impacted by modifying their selected items. The system's convenience is lost if the client may add or delete goods on their own in the application. Therefore, the solution becomes a self-scan solution with significantly greater implementation costs, which is not very helpful. It would be desirable to conduct further study on single-shot models, such as optimizing the YOLO model. Another unresolved issue might be why YOLO's accuracy decreased so much in the test with lower input images.

5.1.1 Model optimizations

The YOLO architecture's primary strength is its fast execution time. YOLO scores the highest FPS across all resolutions tested. It is much quicker than R-CNN and has more than double the frame rate of Retina Net. The drawback is that the accuracy obtained on our dataset has to be increased, and the precision varies depending on the resolution.

Scaling

Because data is static according to the application, our data set comprises images of approximately the same type in every instance. As a result, the entities in our data set are about the same size and do not need to adjust for incredibly large or extremely small objects. This information may be utilized to simplify and improve the model by ensuring that it suggests areas of an acceptable scale. It may be accomplished by adjusting the size of the anchor boxes used for region suggestions.

Image size

The size of the input image is another critical element of model performance. A larger image will preserve more detail, and the model will have to do more calculations as a result. The number of calculations required for a forward pass determines run-time; thus, this is a concern that has to be analyzed, and different picture resolutions are tested. The resolution is handled natively by the program, although it may be rescaled. The picture rescaling for this system must be done locally, which implies that this operation will have no impact on the system's run-time performance.

Non-max suppression

Multiple bounding boxes for the same object instance are problematic with YOLO. For obvious reasons, this is undesirable. Thus, non-max suppression is performed to include just the bounding box with the maximum expected validity score. The non-max suppression level has been fine-tuned to enable multiple bounding boxes in the case of occlusions while avoiding two or more boxes over a single item. If they overlap over a particular number, the level ignores the box with the lower score.

5.1.2 Speed and Accuracy

A model gets slower as it gets precise. It is a quantitative model since reducing the system's processing capacity while retaining adequate accuracy is critical. The present accuracy losses are around 18 percent; therefore, the model's total accuracy should be close to 98 percent if it aims to reduce the losses. It must consider that the user can add or remove items that they consider the system has made an error with, giving the customer the benefit of the doubt as with existing systems.

5.1.3 Sources of errors

The test outcomes are reasonable compared with research in the same area[16]. Incorrectly labelled pictures, class imbalances in the dataset are possible sources of error. However, YOLO has its specific deterioration to address this issue exceptionally well[13]. Deep learning algorithms may always involve overfitting [19], where the model preserves the training data instead of learning a generic mapping that operates beyond the training data set. It is mitigated by weight regularizations, validation sets and early breakdowns of the tested model.

Chapter 6

Conclusions

A model demonstrating the implementation of deep learned-based object recognition in the retail environment has been developed and demonstrated as the outcome of this research. The use of cutting-edge object detection algorithms adapted to this specific use case has shown to be accurate enough to illustrate a better perspective, but it still needs additional research and optimization before it can be deployed. Data collection, annotation, image processing, hardware solutions, and end-user interaction are all part of the project's scope. Full segmentation mask models do not provide better outcomes than bounding box models, with YOLO being the best overall.

Even the most precise models do not give accuracies close to 100 percent, which a consumer and retail outlets would want to apply this system in a workplace context. This research emphasizes the importance of a large amount of annotated data, where data augmentation seems to be an efficient way for enhancing the dataset. There is a lot of potential for future work in object recognition models, data augmentation methods, and optimizations in the retail industry

Reference List

[1] TrigoVision. Evolution of the retail checkout experience. url: https: / / www . trigovision . com / evolution - of - the - retail - checkout -experience/

[2] Amazon Go". url: https://www.amazon.com/b?ie=%UTF8&node=16008589011

[3] Tsung-Yi Lin et al. "Microsoft COCO: Common Objects in Context". In: CoRR abs/1405.0312 (2014). arXiv: 1405.0312. url: http://arxiv. org/abs/1405.0312.

[4] Open Source Initiative. MIT License. url: https://opensource.org/ licenses/MIT

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: Advances in Neural Information Processing Systems 25. Ed. by F. Pereira et al. Curran Associates, Inc., 2012, pp. 1097–1105. url: http://papers. nips.cc/paper/4824-imagenet-classification-with-deep-convolutionalneural- networks.pdf.

[6] J. Deng et al. "ImageNet: A Large-Scale Hierarchical Image Database". In: CVPR09. 2009.

[7] Shaoqing Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: CoRR abs/1506.01497 (2015). arXiv: 1506.01497. url: http://arxiv.org/abs/1506.01497.

[8] Ross B. Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: CoRR abs/1311.2524 (2013). arXiv: 1311.2524. url: http://arxiv.org/abs/1311.2524.

[9] Kaiming He et al. "Mask R-CNN". In: CoRR abs/1703.06870 (2017). arXiv: 1703.06870. url: http://arxiv.org/abs/1703.06870.

[10] Ross B. Girshick. "Fast R-CNN". In: CoRR abs/1504.08083 (2015). arXiv: 1504.08083. url: http://arxiv.org/abs/1504.08083.

[11] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: CoRR abs/1506.02640 (2015). arXiv: 1506.02640. url: http://arxiv.org/abs/

[12] Joseph Redmon and Ali Farhadi. "YOLOv3: An Incremental Improvement". In: CoRR abs/1804.02767 (2018). arXiv: 1804.02767. url: http: //arxiv.org/abs/

[13] Tsung-Yi Lin et al. "Focal Loss for Dense Object Detection". In: CoRR abs/1708.02002 (2017). arXiv: 1708.02002. url: http://arxiv.org/ abs/1708.02002.

[14] Alexander Ljung. imgaug library. https://github.com/aleju/imgaug. Version 0.2.9. 2019.

[15] Lex Fridman et al. "MIT Autonomous Vehicle Technology Study: Large- Scale Deep Learning Based Analysis of Driver Behavior and Interaction with Automation". In: CoRR abs/1711.06976 (2017). arXiv: 1711. 06976. url: http://arxiv.org/abs/.

[16] Patrick Follmann et al. "MVTec D2S: Densely Segmented Supermarket Dataset". In: ECCV. 2018.

[17] IMCO-Berlin. Wake Up Call for Retail: Organized Crime Winning the "Shoplifting War". url: https : / / www . imco - berlin . de / en / blog / article / view - article / ein - weckruf - fuer - den - einzelhandel - organisierte-bandenkriminalitaet-gewinnt-oberhand/ (visited on 06/07/2019).

[18] Trading Economics. Germany Retail Sales YoY. url: https://tradingeconomics.com/germany/retail-sales-annual

[19] Rich Caruana, Steve Lawrence, and C Lee Giles. "Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping." In: vol. 13. Jan. 2000, pp. 402–408.

[20] Wei Liu et al. "SSD: Single Shot MultiBox Detector". In: CoRR abs/1512.02325 (2015). arXiv: 1512.02325. url: http://arxiv.org/abs/1512.02325.

[21] Philipp Jund et al. "The Freiburg Groceries Dataset". In: CoRR abs/1611.05799 (2016). arXiv: 1611.05799. url: http://arxiv.org/abs/1611.05799.

[22] Congcong Li et al. "Data Priming Network for Automatic Check-Out". In: CoRR abs/1904.04978 (2019). arXiv: 1904.04978. url: http:// arxiv.org/abs/1904.0497.