Convolutional Neural Networks for Image Recognition in Mixed Reality Using Voice Command Labeling

Burkhard Hoppenstedt[1], Klaus Kammerer[1], Manfred Reichert[1], Myra Spiliopoulou[2], and Rüdiger Pryss[1]

[1] Institute of Databases and Information Systems, Ulm University, Ulm, Germany
[2] Faculty of Computer Science, Otto-von-Guericke-University, Magdeburg, Germany
burkhard.hoppenstedt@uni-ulm.de

Abstract. In the context of the Industrial Internet of Things (IIoT), image and object recognition has become an important factor. Camera systems provide information to realize sophisticated monitoring applications, quality control solutions, or reliable prediction approaches. During the last years, the evolution of smart glasses has enabled new technical solutions as they can be seen as mobile and ubiquitous cameras. As an important aspect in this context, the recognition of objects from images must be reliably solved to realize the previously mentioned solutions. Therefore, algorithms need to be trained with labeled input to recognize differences in input images. We simplify this labeling process using voice commands in Mixed Reality. The generated input from the mixed-reality labeling is put into a convolutional neural network. The latter is trained to classify the images with different objects. In this work, we describe the development of this mixed-reality prototype with its backend architecture. Furthermore, we test the classification robustness with image distortion filters. We validated our approach with format parts from a blister machine provided by a pharmaceutical packaging company in Germany. Our results indicate that the proposed architecture is at least suitable for small classification problems and not sensitive to distortions.

Keywords: Mixed Realiy · Image Recognition · Convolutional Neural Networks

1 Introduction

Image recognition [2] has become an important factor in the digitalization of industrial factories. Camera systems support the industrial production, e.g., by automatically detecting faulty parts of a machine. The current development of smart glasses [13] offers the possibility to utilize them as mobile cameras, with reduced resolution and expected noise due to the users' movements. Smart glasses offer the further possibility of location independent image classification. Interestingly, a paradigm change in the field of image classification could be observed in the last years. The excellent classification rates of *convolutional neural networks*

(CNNs) outperformed traditional approaches in many use cases [7]. The traditional approaches rely on the explicit definition of image features, while CNNs offer a more generic approach and are able to find complex relationships in images. In the broader context of supervised learning approaches like CNNs, each image needs a *label* to be classified. To tackle the labeling problem, in this work, we generate the labels by mapping voice commands to the smart glasses video stream. More specifically, three technical parts of a machine from an industrial company are classified, whereas the classification process is afterwards tested by using image distortion filters (i.e., blur, noise, and overexposure filters) and measuring the effect on the classification accuracy. In general, our approach tries to provide a simplified image recognition from scratch, for which a user scans his or her environment and all objects during a *calibration/labeling phase*. This input is processed in a machine learning pipeline using CNNs and, eventually, presented to the user through a web service for live classification.

The remainder of the paper is structured as follows: Section 2 discusses related work, while Section 3 introduces relevant background information for image recognition, mixed reality, and convolutional neural networks. In Section 4, the developed prototype is presented, in which the mixed-reality application, the processing pipeline, and the classification algorithm are presented. The results of the distortion algorithms are shown in Section 5. Threats to validity are presented in Section 6, whereas Section 7 concludes the paper with a summary and an outlook.

2 Related Work

Convolutional neural networks (CNN) are widely used in the field of image recognition. CNNs have been successfully tested in the context of face recognition with a high variability in recognizing details of a face [8]. Even though CNNs are widely used for image recognition, they can also be applied to other use cases, such as speech recognition and time series prediction [9]. Since the training of neural network is very time intensive, but the execution time is rather low, CNNs are also suitable for real-time object recognition [10]. Object recognition for augmented reality is mostly performed in a marker-based manner [14], which means that markers (e.g., barcodes) support the recognition process. Standard architectures for CNNs have been proposed, e.g., by AlexNet [7], GoogleNet [16], or InceptionResnet [15]. In large scale scenarios, deep convolutional neural networks incorporate millions of parameters and hundreds of thousands neurons[7], and therefore need an efficient GPU implementation. Furthermore, an evolving topic in the field of image recognition using CNNs is denoted as *transfer learning* [12]. Hereby, image representations from large-scale data sets are transferred to other tasks to limit the necessary training data. In general, not only the content of the image can be learned, but also the image style [3]. The latter offers the possibility of high level image manipulation. The aforementioned techniques denote a promising extension level of our approach. However, these techniques are, in our opinion, not suitable for a small scenario, as a larger computational power would be necessary. To the best of our knowledge, existing works do not combine image recognition, mixed reality, and voice commands as we have realized for the solution presented in the work at hand.

3 Fundamentals

3.1 Convolutional Neural Networks

In general, neural networks are mathematical models for optimization problems, for which the influence of each neuron is expressed with a *weight*. The network constitutes a construct build from neurons that receive an input and compute its output via an *activation function* (e.g., *sigmoid*). A stack of neurons in a single line is denoted as a layer. The first layer constitutes the input, the last layer is called output and all layers in between are denoted as *hidden layers*. In the case of a Convolutional Neural Network (CNN), the neurons form convolutional layers. The most important parameter in a convolutional layer is the filter size, which denotes the window size of the convolution. Each convolution reduces the input's size, so that we use - for the example of image recognition - a padding at the image borders to keep the images dimensions. To reduce the spatial size, a pooling layer is applied. The most common pooling operation is denoted as *max pooling*, where a filter applies the maximum function on the image. The combined information of the neural network is denoted as model. Hereby, the prediction accuracy represents the quality of the model. To keep the computation simple, not all training data is loaded into the network at once. Instead, small batches with a predefined *batch-size* are used in each training iteration. In one *epoch*, the model is fed with all the training images. In general, three types of data sets exist: *Training data*, *validation data* and the *test set*. The model is trained with the training data and tested with the validation data. The test set consists of images from a separate data set to test the generalization of the network and prevent *overfitting* [4]. CNNs are classified as a supervised learning method, which means that they need a *label* that assigns the correct output for each input. In our approach, we try to simplify this labeling process via voice commands. Finally, the speed of the learning progress can be influenced via the *learning rate*, where a high learning rate enables the model to adapt the weight of each neuron quickly.

3.2 Mixed Reality

Mixed Reality tries to achieve the highest overlapping of reality and virtuality in the reality-virtuality continuum [11]. When using the Microsoft HoloLens, then, the latter performs spatial mapping [5] to generate a virtual model. Therefore, virtual objects can be placed in the real world and stay in a fixed position through tracking features. The HoloLens is equipped with various sensors, including a RGB camera, a depth sensor, and a Mixed Reality capture feature. Furthermore, the HoloLens offers the usage of individual speech commands based on natural language processing.

4 Prototype

4.1 Workflow of the Approach

In general, our approach (see Fig. 1) aims at a simple labeling for the image recognition. The first step is to define all names of the objects to be recognized. When starting the mixed-reality application, the HoloLens loads these names from the database and defines speech commands for these terms. Then, the *labeling phase* starts by recording a video. When an object enters the user's field of view, the user says its name. Thereby, the HoloLens logs the current timestamp and the name of the object into a file. The same procedure takes place when the objects leaves the user's field of view.

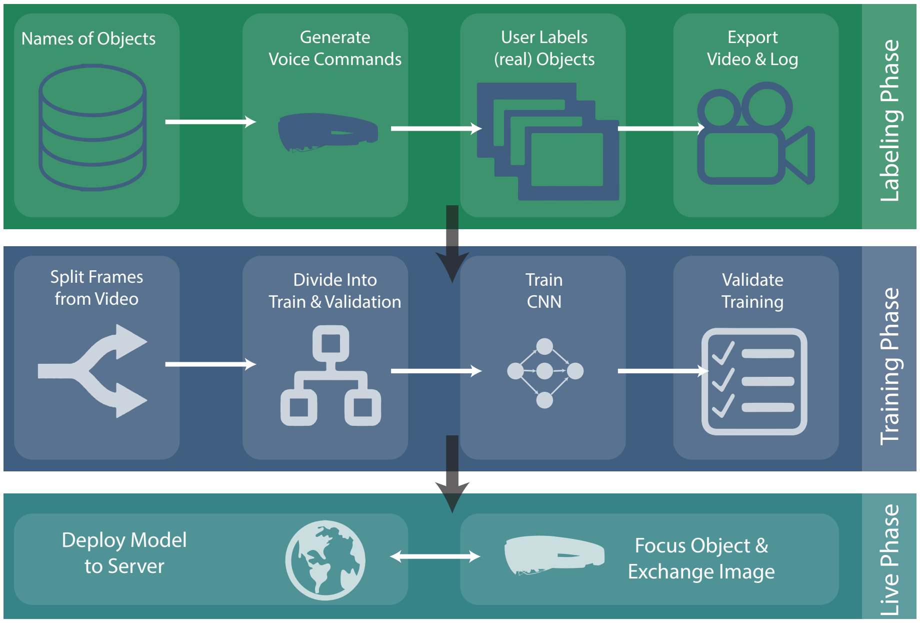

Fig. 1. Workflow of the Approach

As the next step, we generate a *mapping* of images to objects, defined by the period of time between the start and the end voice command. At the end of the labeling phase, the resulting video and log file are sent to an offline application. This latter divides the video into frames and separates all frames into the corresponding folders with images for each object or background. The images are chosen randomly to have the same number of images for each object class. Using the deep learning framework *Tensorflow* [1], the neural network is being trained (i.e., learning phase). As a very simple architecture is used, the network can be trained by using a normal CPU. The image input is automatically split up into 80% training data and 20% validation data. After the learning process is finished, the network is accessible via a RESTful API using a python server. Practically, the user operates with the smart glass, puts the focus to an object, and says the voice commands *classify*. The image is then sent to the server, predicted by the use of the network, and the prediction result is eventually sent back. Note that we needed to include a manual correction into the labeling process. Theoretically, the timestamp t of a voice command should fit exactly to the video frame where the user has seen the object. Unfortunately, there is a calculation time before the voice command gets recognized. We measured this delay and calculated a mean difference between timestamp and frame of 1.16 seconds with a variance of 0.13 seconds. Therefore, we included this delay as a static threshold in the processing pipeline. Altogether, the following technologies were used to realize this approach. As a database for all possible objects, we chose the document-based NoSQL database *MongoDB*. The web interface is provided by the python webserver *Flask*. All machine learning operations are provided by *Tensorflow*, which uses the image library *OpenCV* for image processing. We developed the mixed-reality application in *Unity* and used the Java library *JCodec* to split the video and map the recorded timestamps. Finally, all distortion filters were generated using the software *Matlab*.

4.2 Convolutional Neural Network

The network is implemented using a simple CNN structure (see Fig. 2). The input consists of a *4D tensor*, with the dimensions number of images, width, height, and number of color channels. The weight of the neurons, that will be adapted during the training through *back propagation*, are initialized with a random normal distribution. As an optimizer, we use the *Adam* algorithm [6] for gradient calculation and weight optimization. We choose 0.0001 as a learning rate and a batch size of sixteen. After each convolution step with 32 filters, a max pooling is applied to the result.

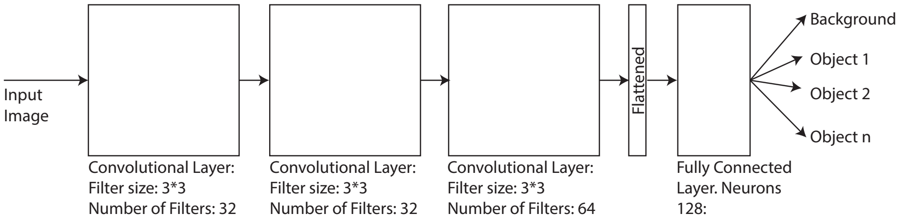
Input Image
Convolutional Layer: Filter size: 3*3 Number of Filters: 32
Convolutional Layer: Filter size: 3*3 Number of Filters: 32
Convolutional Layer: Filter size: 3*3 Number of Filters: 64
Flattened
Fully Connected Layer. Neurons 128:
Background
Object 1
Object 2
Object n

Fig. 2. Used CNN Architecture

4.3 Distortion Filters

To test the effects of bad image quality in our approach, we tested the three distortion filters blurring, noise, and overexposure (see Fig. 3). We applied one filter each time and tested the resulting images with our model in terms of accuracy. For the blurring, we used a box filter with dimensions 11x11. Moreover, a *salt & pepper* noise with a density of 0.2 is applied and, lastly, the brightness effect is achieved by increasing the RGB value by fifty.

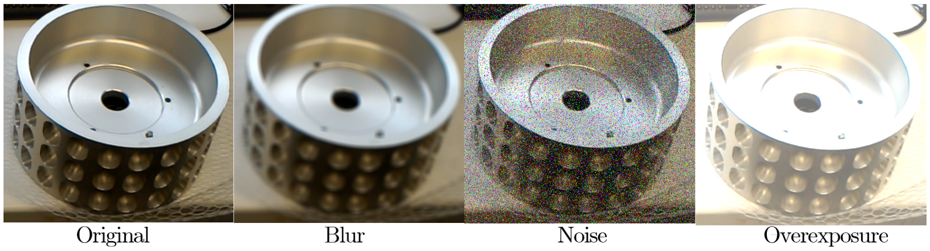

Fig. 3. Distortion Filters

5 Results

The prototype was tested with 500 images per class (i.e., 2000 images in total). Each object to be detected (three in total) and the background are represented by a training class. In general, the training process was stopped after five epochs to measure the accuracy. The training process of the images without any distortion revealed a validation accuracy of 90.6% (see Fig. 4). The noise on the image led to an accuracy of 86.2%, the blurring lowered the accuracy to 85.6% and, lastly, the images with an increase brightness were classified with an accuracy of 81.0%. Therefore, when performing image recognition in Mixed Reality, attention should be paid to a good illumination. The blurring effect, likely caused by fast head movements, was not critical for the classification. In general, the distortion filters did not disrupt the classification significantly.

6 Threats to Validity

Our approach is tested in only one room and with a low number of objects. The higher the number of objects is, the more likely it is that the classification accuracy will decrease. Moreover, as every user is responsible for the labeling process him- or herself, the classification will fail if the objects were not focused precisely or the voice commands are not correctly synchronized with the gaze.

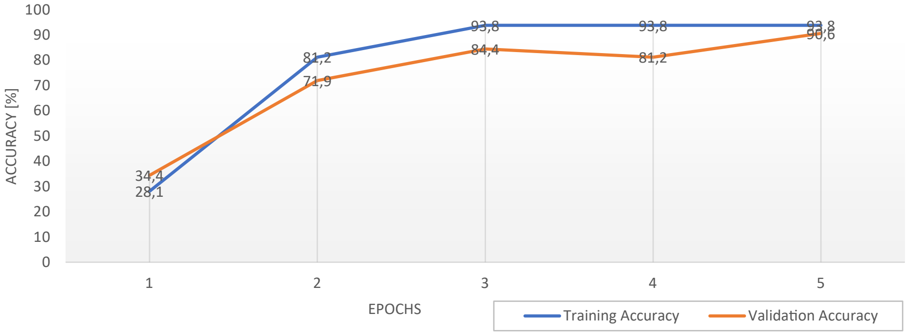

Fig. 4. Learning Progress

Furthermore, neural networks are not a transparent method of machine learning. Therefore, it will be hard to find failure reasons in case of a low classification rate. Despite these limitations, we consider our approach as an easy-to-use object recognition process with high accuracy rates on small data sets.

7 Summary and Outlook

We provided an approach in Mixed Reality that allows users to train objects by labeling frames in the recorded video via voice commands. The generated output is then processed and put into a convolutional neural network. The classification of an image during the use of the HoloLens is achieved by sending the image to a web server. Here, the image is classified with the previously trained model and the response is sent back to the HoloLens. This information can further on be used to monitor additional information for the recognized object. New types of mixed-reality glasses might introduce new possibilities for object recognition (e.g., better image resolution) and could improve this approach. Furthermore, the approach could be tested versus approaches, for which the objects are labeled manually. Moreover, the scalability of this approach should be further investigated. The neural network architecture is conceived in such a way that everyone can provide the computational power for the training phase. When tackling more complex problems, more convolutional layers could be introduced. Currently, the workflow demands that the user names all objects at the beginning. However, the user may consider some objects as more important than others and concentrate on them first. Hence, a future step could be to have the user add labels gradually. This would turn the static learning task into a stream learning task, in which the CNN must be adapted to new classes. In conclusion, we consider convolutional neural networks in combination with a labeling based on voice commands in Mixed Reality as an appropriate approach for object detection, especially for scenarios in the context of the Industrial Internet of Things (IIoT).

References

1. Abadi, M., et al.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), https://www.tensorflow.org/, software available from tensorflow.org
2. Fu, K.S., Young, T.Y.: Handbook of pattern recognition and image processing. Academic press (1986)
3. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2414–2423 (2016)
4. Hawkins, D.M.: The problem of overfitting. Journal of chemical information and computer sciences **44**(1), 1–12 (2004)
5. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., et al.: Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In: Proceedings of the 24th annual ACM symposium on User interface software and technology. pp. 559–568. ACM (2011)
6. Kingma, D.P., Ba, J.L.: Adam: A method for stochastic optimization. In: Proc. 3rd Int. Conf. Learn. Representations (2014)
7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
8. Lawrence, S., Giles, C.L., Tsoi, A.C., Back, A.D.: Face recognition: A convolutional neural-network approach. IEEE transactions on neural networks **8**(1), 98–113 (1997)
9. LeCun, Y., Bengio, Y., et al.: Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks **3361**(10), 1995 (1995)
10. Maturana, D., Scherer, S.: Voxnet: A 3d convolutional neural network for real-time object recognition. In: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on. pp. 922–928. IEEE (2015)
11. Milgram, P., Takemura, H., Utsumi, A., Kishino, F.: Augmented reality: A class of displays on the reality-virtuality continuum. In: Telemanipulator and telepresence technologies. vol. 2351, pp. 282–293. International Society for Optics and Photonics (1995)
12. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1717–1724 (2014)
13. Rauschnabel, P.A., Ro, Y.K.: Augmented reality smart glasses: An investigation of technology acceptance drivers. International Journal of Technology Marketing **11**(2), 123–148 (2016)
14. Rekimoto, J.: Matrix: A realtime object identification and registration method for augmented reality. In: Computer Human Interaction, 1998. Proceedings. 3rd Asia Pacific. pp. 63–68. IEEE (1998)
15. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI. vol. 4, p. 12 (2017)
16. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)