DHBW
Duale Hochschule
Baden-Württemberg

Amazon Go as a Use Case to Enable Image Recognition Using Convolutional Neural Networks

Seminar paper

Submitted on June 7, 2018

Faculty of Economics

Business Information Systems

Kurs WWI-2015I

by

Birkan Erki, Leonard Michalas, Hauke Thielert, Moritz Walther

Contents

List of Abbreviations

ANN Artificial Neural Network

CNN Convolutional Neural Network

SVM Support Vector Machines

NN Nearest Neighbour

RF Random Forests

ReLU Rectified Linear Unit

List of Figures

List of Tables

1 Introduction

In 1994 Jeff Bezos started a company that would grow into an empire that no one could have imagined then, Amazon.com Incorporation.[1] The idea was to set up an online marketplace to sell books. There was no need to set up an expensive retail store infrastructure to reach as many people as possible and unite a broad selection of books and papers. The internet helped him to fulfill that idea, but today with the technological evolution that has occurred in the last two decades, there is more to that business model of Amazon.com than selling books online. For example, today in the time of artificial intelligence and autonomous driving different visions and technologies are driving the way business models are positioned. Today's companies have to focus on how to use and incorporate those technologies in order to create a unique value proposition for the customer.

One of the most promising new technologies of today which is generating a lot of buzz is computer vision. Therefore, this seminar paper will be focusing on the technical evolution of the online retailer giant from Seattle and additionally will explore its use of technology of today. The business model that will be used to showcase this evolution of technology is Amazon Go, one of the newest ideas of the company which relies heavily on the principles of computer vision and image recognition that are used in other industries such as automotive or manufac-turing. Amazon Go is a new retail concept that offers the customer innovative and time efficient grocery shopping without any cashiers or check out points nor lines of any sort by using a combination of the latest technologies of the time to compete with today's traditional super-market brands.[2] The seminar paper will therefore focus on the use of technology and especially the incorporation of computer vision and image recognition. To do that as a first step, it will highlight the evolution of the company and look at the current business model of Amazon Go and the technical implementation to make it happen. Secondly, the objective of this paper will be to dive deeper into the use of technology and create a small scale prototype to showcase and explain the functionality, limitations, and possibilities of its usage. It shall examine different image recognition algorithms in depth. The paper will conclude its findings with a detailed reflection to better understand the work.

Looking at the objectives of this paper from a scientific perspective, the goal is to figure out the way the company incorporated the newest technologies by focusing on computer vision and image recognition to enable its newest value proposition. All required theoretical information will also be explained and summarizes as part of a literature review and will be added in the bibliographical literature overview at the end of the paper.

[1] cf. Sherman 2015
[2] cf. Amazon.com 2018

2 Computer Vision at Amazon.com Inc.

As mentioned before Amazon.com Inc. is an online retailer with a broad variety of services attached to it. To understand the intention of the company to come up with this new business idea, it is necessary to highlight its evolution from an economical and a technological point of view. Therefore, the upcoming two chapters will outline the different products and use cases the company conjured up and then focus on the use case of Amazon Go itself by looking at the different technical components that are combined to enable it.

2.1 Technical Evolution and Products

The technical evolution of the last 20 years has been changing tremendously and the company Amazon.com Inc. is one of the best examples to illustrate this development. To do so a short timeline[3] will be given to point out selected milestones of the company in both an economic and technologic manner.

- 1994: Jeff Bezos founded Amazon.com Inc. in Seattle.
- 1996: The company launched an affiliate program to increase reach and attention by focusing on people and different search engines.
- 1998: Amazon expanded its offerings to music and video sales.
- 2000: A marketplace for third-party orders was opened paired with the beginning of sales of photo and camera items and an expansion of operations in Japan
- 2002: The start of Amazon Web Services. The company entered the cloud computing market.
- 2005: Amazon launched Amazon Prime a specialized program to offer better services to customers such as a quicker delivery and free access to other services.
- 2007: The company launched the e-reader Kindle to expand its e-book presence.
- 2008: Many new companies were acquired to broaden the offering of the company. One of them was Audible a podcast and audiobook retail platform.
- 2014: Fire TV was introduced to the market expanding the offering of streaming ser-vices to TVs and music.
- 2015: The company started selling Amazon Echo a smart speaker using machine learning and speech recognition.

[3] cf. Sherman 2015

- 2016: In December 2016, the first Amazon Go store was opened to its employees in Seattle showcasing a highly developed combination of the latest technologies.
- 2017: Amazon acquired the organic food chain Whole Foods Market.[4]
- 2018: The first Amazon Go supermarket in Seattle was opened to the public.

As the timeline illustrates, the company has increased its digital offerings over time including many other well-known brands and services to offer their customers a unique selection of products.

Just this year the MIT Technology Review has published an article that was summarizing the latest comments on Amazon's technological future from CEO Jeff Bezos himself. During an invitation-only conference in Palm Springs, he was giving the public some information about the next adventures of the company under the name of MARS. In this case, MARS stands for "machine learning, home automation, robotics and space exploration" representing some of the most popular technologies that the company is already involved in and will increase its research in the future. But the company has several even more futuristic technological fields which it would like to investigate. One of them is research on gravitational waves and pulsars to be incorporate in a new project of the CEO, who is trying to build a 10,000-year clock. In addition to that, the CEO was trying out all kinds of robotic research programs such as ping-pong robots or so-called robo-dogs and showcasing insights in one of his private projects in Airspace called Blue Origin.[5] As of right now, there are not a lot of details discovered about those initiatives of the company and the CEO but the conference and its content give a hint of the technological roadmap the company is planning. To conclude Amazon.com Inc. is definitely one of the most exciting companies to watch in the next years as it has proven to incorporate the latest technologies quickly and efficiently to develop its value proposition. Computer vision is one such example.

2.2 Amazon Go

After having summarized the evolution of Amazon.com Inc. this chapter will focus on the technical set up of its latest business model Amazon Go. In order to that this chapter will be divided into three parts. The first part will look at the business model of Amazon Go. Secondly, the chapter will give a broad overview how this new shopping experience works and lastly show the technical components that are used to make it happen.

2.2.1 User manual

The upcoming chapter will focus on explaining how the new concept works and how the customer is involved in the whole process. Therefore the paper will refer to the Amazon Go tutorial from

[4] cf. Turner 2017
[5] cf. Snow 2018

the app that is available in the apple and android app store. The application offers an intuitive animated user guide before signing up. This manual will focus only on the in-store components of the model:

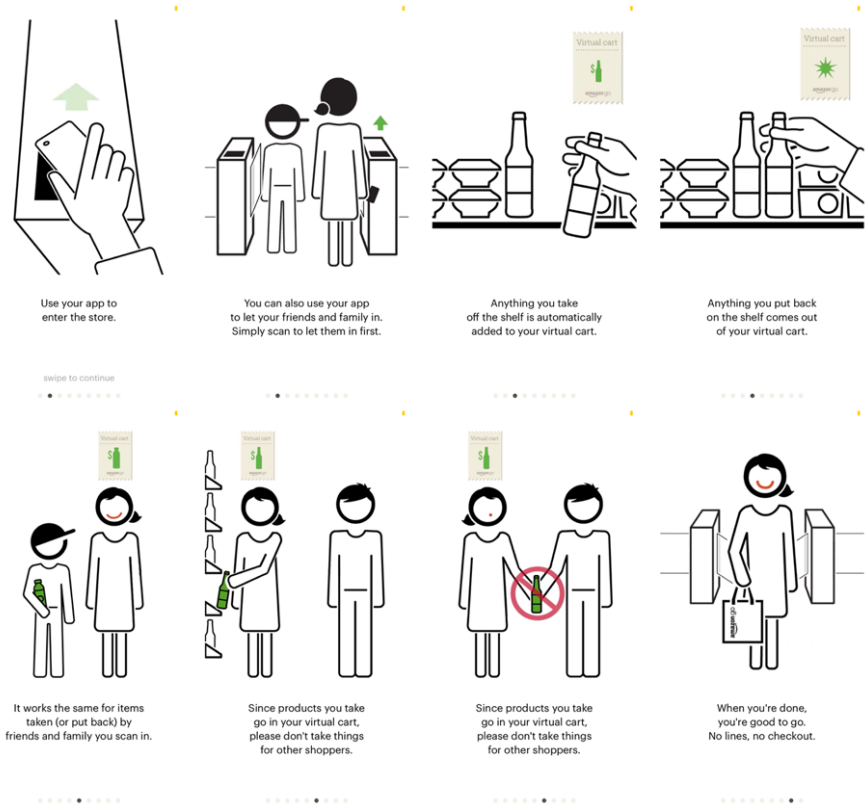


Fig. 1: Amazon Go[6]

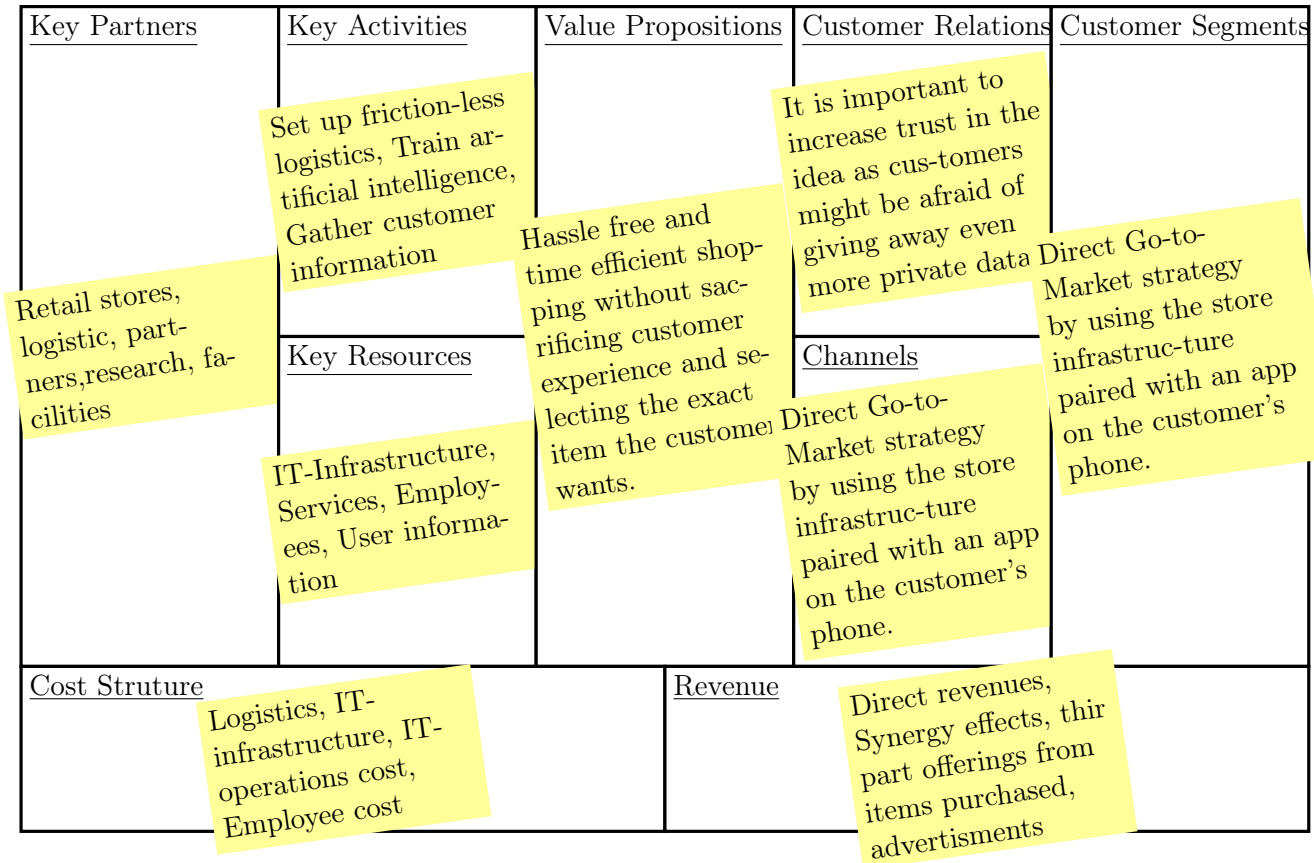
After the customer has downloaded the app and signed up an Amazon account, or potentially logged in to his account, the customer has to open the app and show the QR-code that will show on the display of his device and scan it at one of the entrance-barriers. As the tutorial from the app shows it is also possible to invite friends or other family members to come with you by simply scanning the QR-code every time one of them enters the store, then the customer has to enter last.[7] After that, the phone can be put away and the shopping can start. If a customer decides to take an item from the shelf a variety of sensors such as weight or pressure sensors in the shelf themselves and cameras, that use different methods of image recognition to identify the individual that took the item, work together to add that item to your virtual shopping cart. This works by a concept Amazon calls sensor fusion.[8] In this case sensor fusion stands for the use of different sensor data and camera images that are collected to figure out which item was taken and by whom it was taken. This information will then be transported via the store's network to a processing engine called the master. The master is basically an IT-infrastructure composed of servers that run some sort of machine learning algorithm. The algorithms will then run some calculations to identify the person and assign the item to the virtual shopping cart

[6] cf. Amazon Mobile LLC 2016
[7] cf. Betters 2018
[8] cf. Kolbrück 2016

of the customer. The whole process is done in real time so the customer does not have to wait until a product is added to his cart. In the case that the customer decides to put the item back into the shelf the same procedure will delete this exact item from the shopping cart. This process will also work for every person that has been checked in with the same account. Also, it is important that the customer does not hand items to other persons that have not signed in with the same account otherwise the algorithms might assign this product to that account. After having completed the selection of items that the customer wants to purchase he can simply walk out through one of the QR-barriers. Shortly after that, the app will notify the costumer by sending him a receipt of the purchase on the app. The amount of his purchase will then be billed to the entered account information or credit card.[9]

2.2.2 Business Model

To better understand why companies try to address their customers the way they do it is necessary to have a profound look at their business model. This helps to identify the motivations and objectives of the use case and the financial potential they hope to gain by using them. To do that the business model canvas is a great way to showcase the different aspects that work together. Therefore the following canvas was set up trying to highlight the different fields that make Amazon Go unique:



[9] cf. Burgess 2018

The Business Model Canvas is a global standard to design, challenge and pivot the business models of their companies or customers. It can be divided into 9 different strategic manage-ment tools and processes which are: Key Partner, Key Activities, Key Resources, Value Proposition, Customer Relationship, Channels, Customer Segments, Cost Structure and Revenue Stream. In the case of Amazon Go, the value proposition focuses on the time as one of the most important values for the future.[10] The intention is to save as much time as possible without neglecting the traditional retail store experience. One of the biggest advantages of Amazon is the broad infrastructure and expertise the company already unites. Therefore the company doesn't require a lot of resources, partners and channels to operate this business model. One of the best examples is the acquisition of Whole Foods Market which provides the company with a big retail store infrastructure to incorporate the business model into. Also when looking at cost versus revenue it becomes visible that the costs are easy to be calculated and seem pretty logical and are easy to lower by looking to build the IT-infrastructure with a focus on computing power for the specific tasks they operate while also looking at energy efficient ma-chines and usage. And the other side the revenues are not only the direct profit from sales in the stores but also the value of the data that is generated. This set of data about the individuals can not only be sold but also used to feed into new business models or improve other retail channels of the company. For these reasons a Business Model canvas was used; however, it is important to state that this Business Model Canvas is only one way to illustrate the use case of Amazon and approaching it from different perspectives will lead to different conclusions.

2.2.3 Technical components

Lastly this chapter will highlight the different technical components that enable the process that was illustrated in the previous sections. For this purpose the components will be divided into three different areas of usage.

- **Inventory control features:**
  to control how many items are in the shelf and or in the basket of the customer Amazon uses primarily a set of weight/pressure sensors that are built into the shelf. Paired with the information of the weight of an item and its placement the sensors can notice any change in weight that is remaining. The information is than used to assign specific items to the cart of the customer. In addition to that a big network of digital cameras is set up in the shelf, the ceiling and possibly other areas of the store. These camera infrastructure will capture the people entering the shop and later the people that are taking items from the shelf. Similar to the sensors in the shelf the images will be used to identify the customer through shape and or facial recognition patterns and correctly assign the items he took to his shopping cart. Also to enhance the product identification all of the products have some sort of RFID tag that is printed on their package. This tag can also be captured by the camera system and will help to figure out what product was taken.

[10] cf. Osterwalder/Pigneur 2011

- **User experience and identification:**
  Next in order to provide a frictionless user experience the company provides the Amazon Go app for each customer to handle his shopping as intuitively as possible by also providing him information, special discounts and later his receipt. By scanning the QR code that is displayed on the device of the customer the infrastructure can identify the customer at the moment of his entrance. All additional information of the customer that might be useful for later identification will then be loaded and used to offer special discounts and assign the right items to his shopping cart.

- **IT-infrastructure and processing units:**
  Lastly in order to complete all these features the concept heavily relies on the IT-infrastructure in the background. As part of this infrastructure several network, storage and server components are combined to receive the date from the sensors, cameras, RFID tags, the app and other API's to databases from Amazon.com that contain valuable information about each individual customer. After that a set of machine learning algorithms are used to calculate which customer is in the store and what items he or she takes from the shelf. This information will then be saved and sent to the app to illustrate the shopping cart of the customer and later bill him the correct amount from his account.[11]

The given three component groups conclude the technical infrastructure of an Amazon Go store and underline its functionality from a technical point of view. The inventory control functions are particularly complex, and make heavy use of computer vision technologies such as image recognition. A deeper technical understanding of how these technologies work in general will be given in the next chapter.

[11] cf. Fungineers 2016

3 Technical Setup Overview

In order to realize the innovative concepts of Amazon Go, two main technical areas are being merged. This chapter will first elaborate on the setup of computer vision. Then, Machine Learning will be explored and how it enables image recognition algorithms.

3.1 Computer Vision

As part of this chapter, computer vision will be presented. This will include a general explanation, examples and limitations. In the end, details surrounding the usage of computer vision by Amazon Go will follow

3.1.1 Explanation

The term "computer vision" essentially describes the set of computer-based tasks, which orientate themselves on the human sense of visual sight. Furthermore, it can be defined as the science endowing computer or other machine with vision, or the ability to see.[12]

Core tasks of a seeing system are methods such as acquiring, processing, analyzing and understanding a digital image. In addition, the system has to perform an extraction of multidimensional data from the real world in order to produce usable information. In context, this means the transformation of visual images into descriptions of the world can interface with other thought processes and elicit appropriate action.[13]

Examples of methods used to achieve that, include color-classification (e.g. to classify skin tones), contrast-analysis (e.g. to recognize geometric shapes), optical flow (e.g. to extract movement) and various methods for edge detection such as Sobel-Operator, Gauß-Laplace-Pyramid and Wavelets. More complex recognition tasks use models. These rely on beforehand knowledge, which is crucial for the recognition of the object. These models can be two- or three-dimensional or statistical, which means that they can be deformed.[14]

The organizations of these methods is highly application dependent. However, there are typical functions which are common to many computer vision systems:

[12] cf. Moscatelli/Kodratoff 2017, p. 1
[13] cf. Learned-Miller 2011, p. 2
[14] cf. Moscatelli/Kodratoff 2017, p. 2

Method Name	Method Explanation
Image Acquisition	A digital image is produced by one or several image sensors, which, besides various types of light-sensitive cameras, include range sensors, tomography devices, radar, ultra-sonic cameras, etc. Depending on the type of sensor, the resulting image data is an ordinary 2D image, a 3D volume, or an image sequence. The pixel values typically correspond to light intensity in one or several spectral bands (gray images or color images).
Pre-Processing	In order to extract some specific piece of information, it is usually necessary to process the data in order to assure that it satisfies certain assumptions implied by the method.
Feature Extraction	Image features at various levels of complexity are extracted from the image data.
Detection / Segmentation	Decisions about which image points or regions of the image are relevant for further processing.
High-level Procession	This process deals with image recognition (classifying the image into categories) and image registration (comparing and combining two different views of the same image).
Decision Making	Pass / fail of the automation or flag for further human review.

Tab. 1: Important methods in Computer Vision in chronological order[15]

The methods explained above recognize characteristics and data out of images and order them by the objects contained of said image. This is based on the deductive application of known rules. An extension of this process is described as machine learning, which would use inductive methods to find these rules in the first place.

3.1.2 Examples

The usage of computer vision as an interdisciplinary field often occurs in the form of application programming interfaces (APIs). These APIs are published by a host of vendors with all kinds of focuses including Google, IBM and Microsoft.[16]

Furthermore, computer vision is currently used in two main fields. One of them is industrial development. Systems here are used for quality-control and the measurement of simple objects. Therein in the programmer determines the condition of the surroundings. Important factors for the algorithms to run smoothly are camera position, lighting, speed of movement of the object tracked and position of the object. Concrete application scenarios include:[17]

[15] cf. Learned-Miller 2011, p.9
[16] cf. Oberoi 2016, p. 1
[17] cf. Steger/Ulrich/Wiedemann 2018

- Components on a conveyor belt are monitored, in order to check accuracy and reduce the error rate of the final product.
- Welding-robots are being steered into the right welding-direction.

The other field in which computer vision is heavily used in, are natural environments. This are project much more complex requirements to the technical recognition of images. Mainly because the programmer has no influence on the surroundings, which can decrease the functionality immensely. For example, a black car on a bright background is easily recognizable. However, a green car driving past a grassland may cause problems. Scenarios of application include:[18]

- Automatic recognition of streets and pedestrians on the sidewalk.
- Recognition of human faces and their expressions.
- Recognition of people and their tasks.

3.1.3 Limitations

Situations that limit the usability of computer vision can occur in all the beforehand stages of method processing. However, there is one limitation called "scene understanding" which is probably the most significant of all limitations. It describes the system being given an image of the world (not a photo centered on an object) and it determining what is going on, what are the elements of the seen (visually and structurally) and how do they relate to each other in way that is relevant to the agent being supported with computer vision.[19]

One aspect of such a model will be the identification of surfaces. But surfaces are not the flat rectangles of man-made objects. A valley, a rock, or a row of trees can be a surface. Another aspect is two-dimensional parsing of the scene structure. In an interactive deployment, the statistical organization of the scene needs to evolve continuously as a robot explores and moves through it.[20] If scene understanding technology matures, then for example a number of real-time flexible computer vision applications will be possible.

3.1.4 Usage by Amazon Go

As previously mentioned, the "Just Walk Out" concept of Amazon Go enables users to enter the store with the Amazon Go app, shop for products and walk out of the store without lines or checkout. This is mainly powered by a network of cameras in the store, which use computer vision to detect when an item has been taken from a shelf by a customer and who has taken it. The system is also able to remove an item from a customer's virtual basket if it is put back

[18] cf. Steger/Ulrich/Wiedemann 2018
[19] cf. Learned-Miller 2011, p. 9
[20] cf. King 2016, p. 1

on the shelves. Thereby, Amazon is able to track people and objects in the store at all times, ensuring it bills the right items to the right shopper when they walk out.

Making exact statements about how Amazon Go operates is impossible as the company hasn't published any official information. However, assumptions can be drawn from the recently published Amazon patent filings. These show that the cameras used in Amazon Go may include RGB cameras, depth sensing cameras and infrared sensors. Within the patent filings are some additional details that suggest simply using the app to enter may not be quite as straightforward as it sounds. It is noted that upon detecting a user entering and/or passing through a transition area, the user is identified, and that various techniques may be used to identify the user. This includes a camera that captures an image that is processed using facial recognition, and that in some implementations, one or more input devices may collect data that is used to identify when the user enters the materials handling facility.

Underpinning the computer vision systems is machine learning. Working together, the systems allow for advanced pattern recognition and allow for machines to draw conclusions from vast data sets. Moreover, Amazon says it uses sensor fusion during the shopping process. It's likely this systems involves combining data from many sensors – these include weight sensors in the shelves to track individual products. This mainly leads to a higher degree of accuracy.

3.2 Machine Learning

This chapters provides theoretical core knowledge around the topic of Machine Learning. First, fundamentals, characteristics and terminologies will be explained. Then, a more broad elaboration on different machine learning algorithms will be provided. Advancements in the field of artificial intelligence could mainly be realized by applying Machine Learning algorithms on large amounts of data. Machine Learning algorithms recognize patterns and learn to make predictions and recommendations by processing new and historical Data. This happens without explicit instructions through programming. In addition, algorithms adapt to new data sets and experiences, which increases the efficiency of predictions.[21]

The computation characteristic of machine learning is the generalization of a test-experience (or examples) and the output of a hypothesis, which predicts the target function. Generalization enables the system to achieve good results for invisible datasets that lie in the future. Unlike other problems with optimization, machine learning does not employ an accurately defined function that needs to be optimized. Instead, test-errors serve as catalysts, to research learning-errors. The process of generalization requires so called classifiers, which input discrete or continuous Vectors and output a class.[22]

The following graphic provides an overview on the components of Machine Learning:

[21] cf. McKinsey & Company 2018, p. 1
[22] cf. Awad/Khanna 2015, p. 1

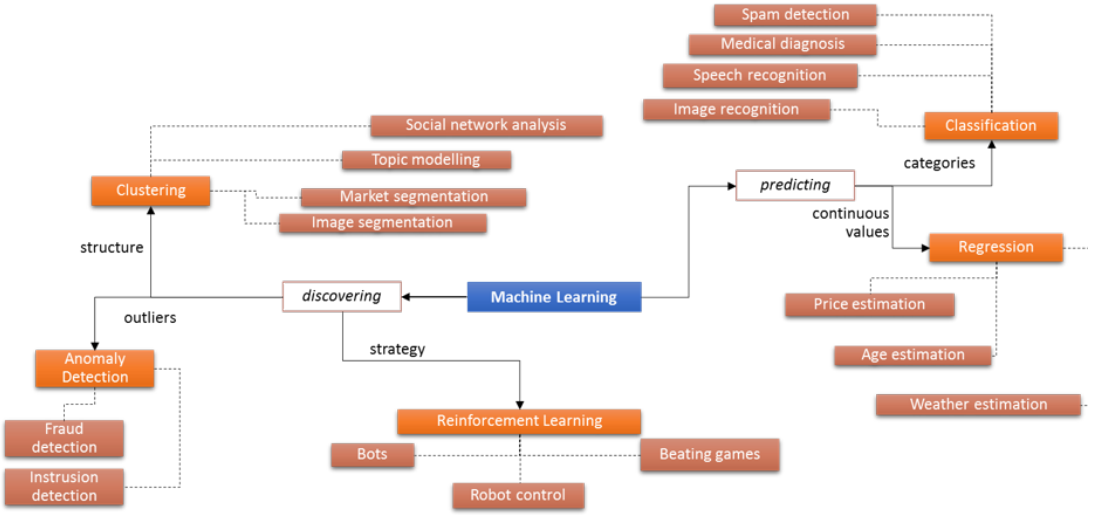


Fig. 2: Machine Learning - Overview

In order to have a better understanding about the concept of machine learning, the following subsection will deal with important terminologies:

- Classification: Process of summarizing objects into classes.
- Regression: Analysis, which help to assess, how the value of a dependent variable changes when an independent variable is changed.[23]
- Pattern recognition: Procedure, which automatically categorizes measured signals.[24]
- Accuracy: The rate of correct and incorrect predictions that the system calculates over a dataset. Normally, accuracy is measured on an independent test-dataset, which was not previously used for learning.[25]
- Dataset: An unsorted collection of data, which can fit into a scheme. In a typical dataset, every column represents an attribute and every row a member of the dataset.
- Instance: An object, which is characterized by vectors, either trained through generalization to the model or used for a prediction.[26]

Classification and regression are vastly different approaches, but they can both occur in the same algorithmic methods. A classification in an algorithm predicts a continuous value, but this continuous value has the probability-form of a discrete class. Moreover, classifications can be evaluated by error-rate, regressions cannot. In the case of the latter, a discrete value is predicted, but in the form of an integer amount.[27]

[23] cf. Ketkar 2017, p. 8
[24] cf. Franz 2017, p. 6
[25] cf. Awad/Khanna 2015, p. 2
[26] cf. Awad/Khanna 2015, p. 3
[27] cf. Ketkar 2017, p. 8

3.2.1 Types of Machine Learning Algorithms

After base-knowledge about learning Machines has been explained, a closer look regarding algorithms can be proposed. Generally, three different types of Machine Learning algortihms are distinguished: Supervised Learning, Unsupervised Learning and Reinforcement Learning.

- Supervised Learning
  This algorithms uses trained data and underlying knowledge from human, in order to learn to relationship between multiple inputs and one output. Supervised Learning is best used, when it is known how to classify the input values and which type of behavior is to be predicted. This knowledge can then be applied to new data sets.[28]

- Unsupervised Learning
  This type of algorithm explores input data, without having to be fed explicit output values. For example, the customer demographic is analyzed to detect patterns. If it is not known how to classify the datasets, this algorithm is probably the most fit to take charge of classification.[29]

- Reinforcement Learning
  This algorithm learns to execute a task, by simply receiving treats or points for results. For example, is maximizes received points for an increasing Return-On-Investment portfolio. It is best to use reinforcement learning, when little training data is available, the target value is not ideal to define or when interaction with environment is the only way to learn.[30]

[28] cf. McKinsey & Company 2018, p. 2
[29] cf. McKinsey & Company 2018, p. 2
[30] cf. McKinsey & Company 2018, p. 2

3.3 Image Classification

After having presented the three main machine learning subject areas (reinforcement learning, regression and classification), this paper mainly focuses on image classification, which is part of the subject area of classification.

The whole idea of image classification is to build a computer system, that given an input is able to predict the correct output.[31] Therefore Keifer and Lillesand defined image classification as the process of categorizing all pixels in a given image to obtain a given set of labels.[32] Thus, image classification includes the task of processing a given image (input) and assigning a label (output) to it. For example, an image classification task could be labeling and image as a dog or cat, assumed an image displays either a dog or a cat.[33] In real-world image classification enables many useful application scenarios for both society and enterprises. The most promising use cases are summarized in table 2 below.

Industry	Use Case
Automotive	Autonomous driving including scene analysis, automated lane detection, and automated road sign reading to set speed limits.[34]
Media	Applications that recognize images on social media to identify brands so companies can better position their brands around relevant content.[35]
Health Care	Applications for detecting disease (for example tumors) in MRI scans.[36]
Retail	Applications that analyze supermarket shelves and the shopping carts of shoppers to detect items and make recommendations in the store about what else they might want to buy.[37] And of course, the use case this paper is focusing on is using image classification and recognition to make a fully cashier less store possible.[38]

Tab. 2: Image Classification – Use Cases

In order to be able to classify an image, machines are using so-called classifiers. A classifier is an algorithm, which associates different objects that can be characterized by multiple features to one or multiple classes.[39] In practice many different algorithms can be used for image classification. However, this chapter will be highly focused on explaining Neural Networks and in particular the

[31] cf. Kapur 2017, ch. 5
[32] cf. Lillesand/Kiefer 1994
[33] cf. Shanmugamani 2018, ch. 1
[34] cf. Luckow et al. 2016
[35] cf. Begg 2017
[36] cf. Ali 2017
[37] cf. Yoruk/Etin 2017
[38] cf. Wingfield 2016
[39] cf. Bothe 1993

Convolutional Neural Network classifiers since Amazon is using this classifier inside the stores.[40] Other classifiers which can be used alternatively will be introduced briefly at the end of this chapter.

3.3.1 Neural Network

A Neural Network is a computational system that works in a similar way to the human brain. The brain attempts to remodel the interaction between its neurons and synapses.[41] Hence, neural networks often are also called Artificial Neural Network (ANN).[42] ANNs are versatile, powerful, and scalable which makes them the ideal solution to solve large and highly complex machine learning tasks, such as classifying billions of images (e.g. Google Images).[43]

a) Neurons
   ANNs define neurons (often also called nodes or units) as a processing unit, which performs a mathematical operation (the activation function) to generate one output from a set of multiple differently weighted inputs. The output of a neuron is equal to the weighted sum of the inputs plus the bias of the corresponding neuron.[44] To make this a little bit clearer let's say a neuron receives three inputs as shown in figure 3.

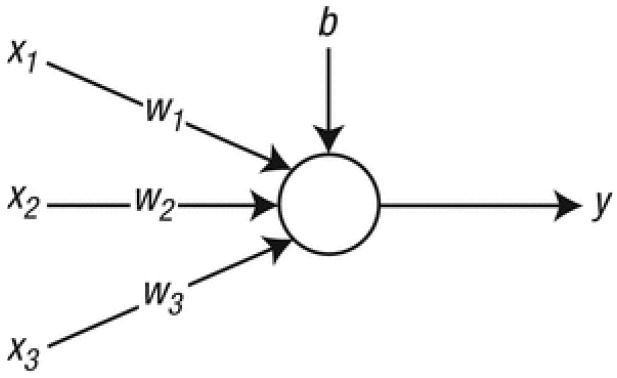


Fig. 3: A neuron that receives three inputs [45]

   The numerical input signals (x_1, x_2, and x_3) are either coming from other neurons or external sources. Each input has a corresponding weight (w_1, w_2, and w_3).[46] Weights are a very important factor in converting an input to impact the output. They are numerical parameters, that determine how vigorously each neuron affects the other ones.[47] Before an input signal reaches the neuron it is multiplied by the corresponding weight. Next to the inputs the neuron is also impacted by the bias (b). The main function of the bias is to provide every neuron with

[40] cf. Gutierrez 2018
[41] cf. Trinkwalder 2016
[42] cf. Yegnanarayana 2004
[43] cf. Géron 2018
[44] cf. Venkateswaran 2017
[45] cf. Kim 2017, ch. 4
[46] cf. Karn 2017
[47] cf. Venkateswaran 2017

a trainable constant value in addition to the normal inputs it receives.[48] Once all weighted input signals and the bias have reached the neuron, their values are added to be the weighted sum. Hence, the weighted sum is calculated as follows:

$v = (w_1 \times x_1) + (w_2 \times x_2) + (w_3 \times x_3) + b$

Lastly, the neuron applies an activation function on the weighted sum.[49] The activation functions are the core of ANNs. They perform a nonlinear transformation on the weighted sum. The usage of an activation function has two reasons. First of all, the result determines whether a particular neuron is activated or not, that is, whether the information received by the neuron is relevant or not.[50] Secondly, neural network needs to be able to learn and represent almost any arbitrary complex function. However, this would not be possible working only with linear functions. Therefore, it is important to apply a non-linear activation function on the weighted sum in order to be able to able to generate non-linear mappings from inputs to outputs.[51] The following table summarizes the three most commonly used activation functions briefly.

Activation Function	Functionality	Advantage	Disadvantage
Sigmoid $\sigma(x) = 1/(1 + exp(-x))$	Takes an input and brings it to range between 0 and 1.	Gets close to the firing rate of a real neuron.	It saturates and the gradient of the function becomes shallower at some point.
Tanh $tanh(x) = 2\sigma(2x) - 1$	Takes an input and brings it to the range between -1 and 1.	Unlike the sigmoid neuron its output is zero-centered	It saturates and the gradient of the function becomes shallower at some point.
ReLu $f(x) = max(0, x)$	Takes a real-valued input and thresholds it at zero.	Lower computational cost, fast training of large nets.	Can be fragile during training and can "die".

Tab. 3: Summary of the most commonly used activation functions [52]

b) Neural Network Architecture

After understanding the functionality of a single neuron, it is now important to understand how multiple neurons interact with each other within a neural network.

[48] cf. Karn 2017
[49] cf. Kim 2017
[50] cf. Dutta 2018
[51] cf. Walia 2017
[52] cf. Karpathy 2017

Neural Networks consist of a collection of neurons that are connected and organized in an acyclic graph. Hence, the outputs of some neurons can become inputs to other neurons.[53] Most commonly Neural Network models are organized into distinct layers of neurons as can be seen in figure 4.

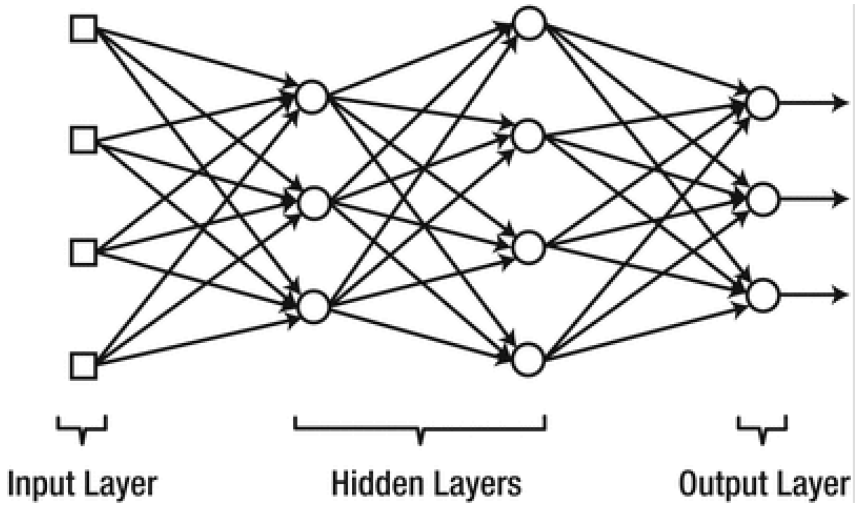


Fig. 4: Neural Network Architecture [54]

A neural network with a layered structure of neurons consist of three categories of layers: The input layer, the hidden layers and the output layers.[55] The functionality of the individual layers will be briefly explained in table 4.

Layer	Functionality
Input Layer	The input layer consists of multi-input neurons. The input neurons provide information from the outside to the network. No computation is performed in any of these neurons. Their only job is to pass on the information from the outside to the hidden neurons.
Hidden Layer	The hidden layer consists of multiple hidden neurons. They are called hidden since they don't have a direct connection to the outside. Hidden neurons perform computations and transfer information from the input neurons to the output neurons.
Output Layer	The output layer consists of multiple output neurons. The output neurons are responsible for computations and transferring information from within the network to the outside.

Tab. 4: Layers of a neural network [56]

c) Evolution of Neural Networks

[53] cf. Karpathy 2017
[54] cf. Kim 2017, ch. 4
[55] cf. Géron 2018
[56] cf. Kim 2017, ch. 4

Neural Networks have been developed from very simple architectures to more complex structures over the past years. The first neural networks were very simple and in contrast to the neural network in figure 4 they only consisted of an input and an output layer. These types of neural networks are called single-layer neural networks. Multi-Layer neural networks constitute a more modern architecture. These neural networks contain an additional layer (the hidden layer) between input and output layer as can be seen in figure 4. Multi-layer neural networks can be categorized into two groups. Networks with only one hidden layer are called shallow neural network. A multi-layer network with multiple hidden layers on the hand is called a deep neural network. Deep neural networks are mostly used in practical applications.[57]

d) Neural Network Preperation
After having a basic understanding on how neural networks work and how they are structured it is now time to take a look at how neural networks can be prepared in order to use them to solve a classification problem. Regarding this topic, Kim defined five process steps, which will be explained in the following:[58]

1) Initialize the weights of the neurons with adequate values.

2) Feed the neural network with training data. Obtain the output from the neural network and calculate the error from the correct output.

3) Adjust the weights to reduce the errors.

4) Repeat step 2 and 3 with as much training data as you have.

This process is also displayed in figure 5.

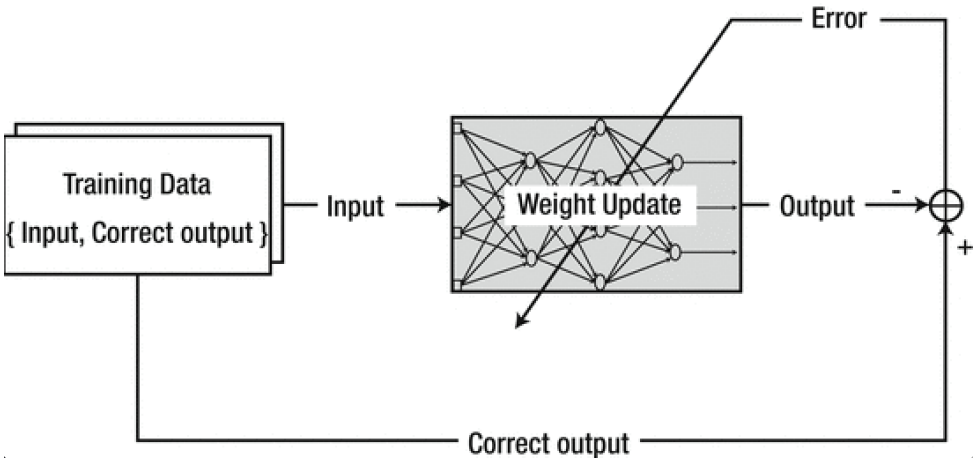


Fig. 5: Neural Network Preperation [59]

[57] cf. Kim 2017, ch. 4
[58] cf. Kim 2017, ch. 4
[59] cf. Kim 2017, ch. 4

Now time to take a deeper look into how exactly Neural Networks are used for image classification. Therefore in the next section, a specific manifestation of a neural network (Convolutional Neural Network), which is predestined for image classification, will be elucidated.

3.3.2 Convolutional Neural Network

The Convolutional Neural Network (CNN) is a variation of an ANN and is specifically suitable for image processing and classification.[60] In order to classify an image a CNN performs five steps (Convolution, ReLU, Pooling, Connecting, Backpropagation).[61] The goal of this section is to explain these steps so that the reader understands how a CNN is able to classify images. For better understanding, the classification process will be demonstrated with the aid of an example black and white image (s. figure 6).

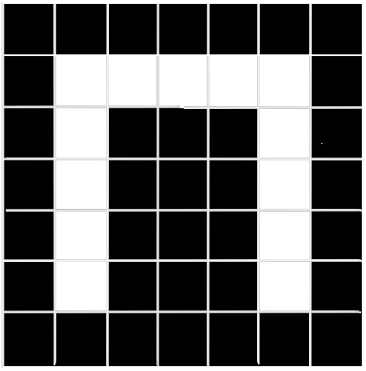


Fig. 6: Pixel image of a door

However, before taking a closer look at the five steps a CNN performs, it has to be clear how a computer perceives the given image, and where possible challenges may arise when classifying it.

To a computer, an image looks like a two-dimensional array of pixels with a number in each position.[62] As can be seen figure 7 in the referenced example a pixel value of 1 is white, and -1 is black.

[60] cf. Trinkwalder 2016, p. 130
[61] cf. Cakmak/Das 2018
[62] cf. Rohrer 2016

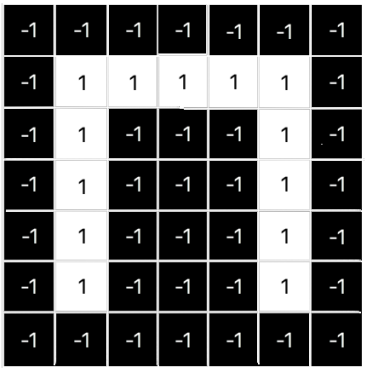
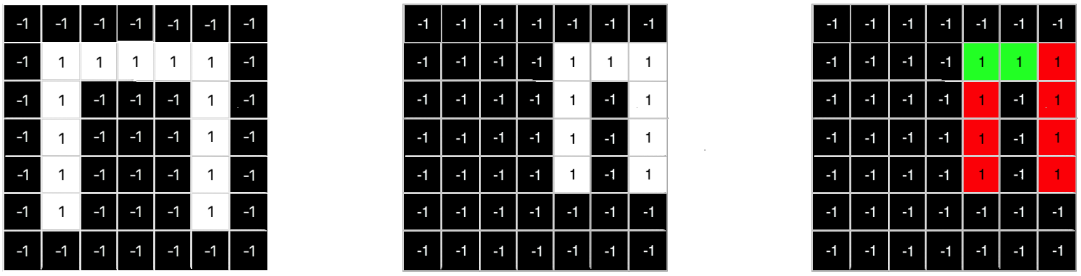


Fig. 7: Pixel image of a door with values

If an image of a door would always look exactly the same, classifying it would be an easy task. In this case, the computer would just need to loop through each pixel value in order to be able to make a decision whether the given image displays a door or not. However in real-world not every door looks the same. Sometimes the door can be shifted, smaller, bigger or even rotated. The same door pictured from different angles, with different lighting, or differing hardware will also look different. In this case, just looping through the pixel values would not be crowned with success, since the computer would identify a value mismatches as demonstrated in figure 8.



(a) Pixel image of a door (b) Variation (c) Value mapping

Fig. 8: Comparison of two door images

Because of that problem CNNs try to match small parts of an image instead of the whole image at once. These small parts are also called features.[63] Some examples of possible features in our example are displayed in figure 9.

[63] cf. Athiwaratkun/Kang 2017

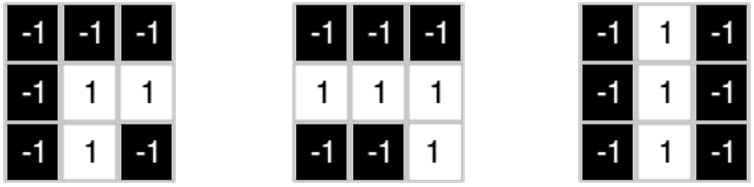


Fig. 9: Example image features

After understanding the concept of *features* and their importance for image classification, now the five steps a CNN performs to classify an image will be introduced. Each of the steps has a dedicated layer, which will be explained in the following:

1. Convolution Layer
   During the convolution, the CNN tries to figure out where and whether the defined features are matching with parts of the given image. To do this a concept called *filtering* is used. During the filtering four steps are being performed.[64]

   a) Selection of a feature, which should be used for filtering.

   b) Multiplication of each image pixel by the corresponding feature pixel.

   c) Addition of all results.

   d) Divide by pixel number of the feature.

   For better understanding in the following this process will be demonstrated. In the first step of this demonstration it will be assumed that the in image 10 displayed feature and image patch will be used.

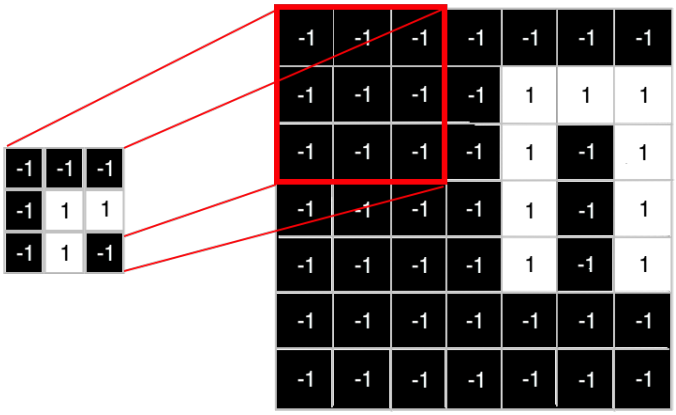


Fig. 10: Pixel image of a door with values (1)

[64] cf. Kim 2017, ch. 6; cf. Karn 2016

Now each image pixel will be multiplied with the corresponding feature pixel. All results will be added and finally divided by the number of pixels.

$$X = \frac{(-1)\times(-1)+(-1)\times(-1)+(-1)\times(-1)+(-1)\times(-1)+(1)\times(-1)+(1)\times(-1)+(-1)\times(-1)+(1)\times(-1)+(-1)\times(-1)}{9}$$

$$X = \frac{1+1+1+1-1-1+1-1+1}{9}$$

$$X = \frac{1}{3}$$

$$X = 0.33$$

The same process can be repeated on other patches of the image as well.

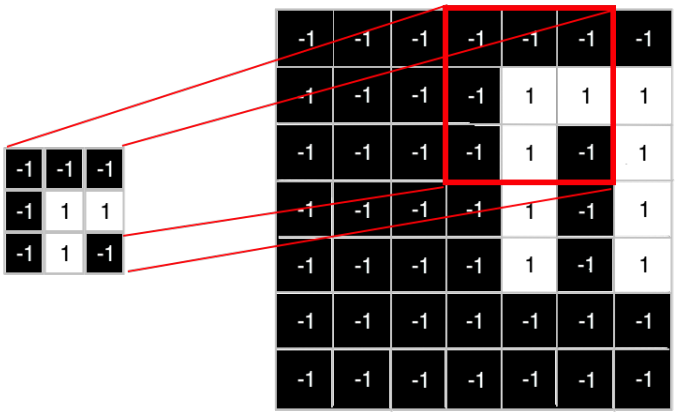


Fig. 11: Pixel image of a door with values (2)

$$X = \frac{(-1)\times(-1)+(-1)\times(-1)+(-1)\times(-1)+(-1)\times(-1)+(1)\times(1)+(1)\times(1)+(-1)\times(-1)+(1)\times(1)+(-1)\times(-1)}{9}$$

$$X = \frac{1+1+1+1+1+1+1+1+1}{9}$$

$$X = \frac{9}{9}$$

$$X = 1$$

This demonstration has shown how filtering works. Furthermore, it showed that if the feature pixels are matching with an image part the result will be 1.

During convolution, this filtering process is repeated for every possible position on the image and with all defined features. The results of each filtering iteration will be stored in a so-called *feature map*. The *feature map* gives a very accurate overview where the used feature occurs on the image. The *feature map* for the feature we used in the demonstration above is displayed in figure 12.[65]

The final result of the convolution step are multiple *feature maps* (one *feature map* for each identified feature).

[65] cf. Kim 2017, ch. 6

0.33	0.33	0.33	1	0.11
0.33	0.33	0.11	0.33	-0.77
0.33	0.33	0.11	0.55	-0.33
0.33	0.33	0.33	0.33	-0.11
0.33	0.33	0.11	0.11	-0.11

Fig. 12: Feature Map for one feature

2. ReLU Layer
   During the convolution, the machine computes the linear operations by doing element-wise multiplication and summations. However, as already explained previously (s. section 3.3.1) cascading linear operations produce another linear system Therefore it is important to add non-linearity after each convolution.[66]

   Hence, after each convolution, an activation function is applied to the outputs (the *feature maps*). Modern CNNs use Rectified Linear Unit (ReLU) as the activation function.[67] This activation function was already explained earlier in section 3.3.1.

   For demonstration purposes figure 13 shows how the feature map from figure 12 looks like after the ReLU function was applied on it.

0.33	0.33	0.33	1	0.11
0.33	0.33	0.11	0.33	0
0.33	0.33	0.11	0.55	0
0.33	0.33	0.33	0.33	0
0.33	0.33	0.11	0.11	0

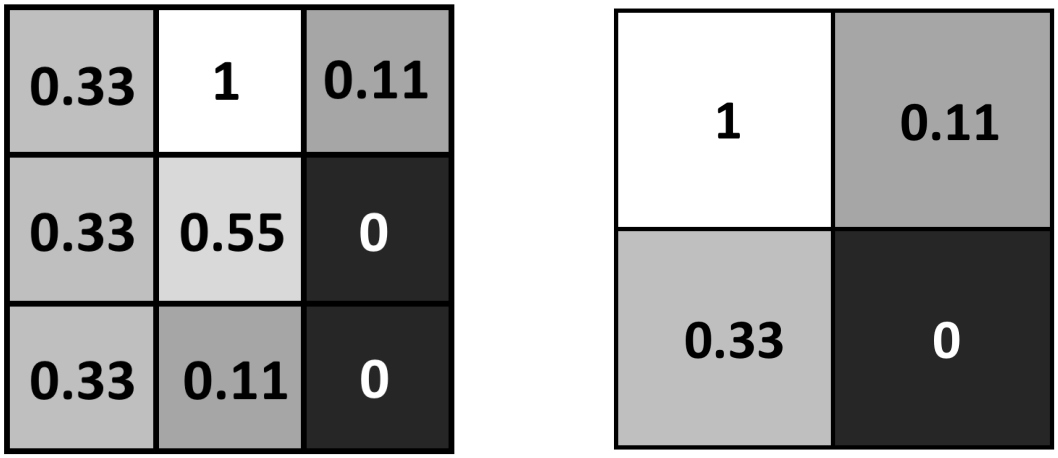
Fig. 13: Feature map after ReLu

3. Pooling Layer

[66] cf. Dev 2017, ch. 3
[67] cf. Dev 2017, ch. 3

The pooling function reduces the size of the image by combining neighboring pixels of a certain area (i.e $2x2$) of the image into a single representative value.[68] Hence, their goal is to shrink the feature map in order to reduce the computational load, the memory usage, and the number of parameters. Pooling is a common technique that is used in many other image processing schemes as well.[69]

Figure 14 shows how the feature map from our example looks like after pooling its values twice.



(a) Pooled feature map (1)

(b) Pooled feature map (2)

Fig. 14: Pooling

4. Fully connected Layer
   The output from the preceding layers is representing high-level features of the input image. The purpose of the fully connected layer is now to classify these inputs.[70] Therefore the basic idea on which a fully connected layer works is that it takes these inputs and identifies the specific feature that mostly correlates to a particular class. For example in our example model which tries to predict whether an image contains a door or not, it will identify high values in the feature maps, which represent some high-level features like door corners.[71]

   To do this, in the fully connected layer the values from all feature maps are processed and sorted into a list as can be seen in figure 15.[72] In the example above only one feature was taken into consideration and only one feature map was created. However, normally a machine takes multiple features into consideration and hence it is creating many more feature maps. In this list of values some values tend to be high when the input image contains a door or respectively will be low when it does not contain one. Hence, these values have a lot of weight (explained in section 3.3.1), when it comes to deciding whether

[68] cf. Kim 2017, ch. 6
[69] cf. Géron 2018, ch. 3
[70] cf. Karn 2016
[71] cf. Dev 2017
[72] cf. Rohrer 2016

the input image contains a door or not.[73]

In our example (s. Figure 15) all values in the list with a high weight (represented by thick lines) have the highest possible value 1. Therefore the model can predict that the input image contains a door with a very high certainty.

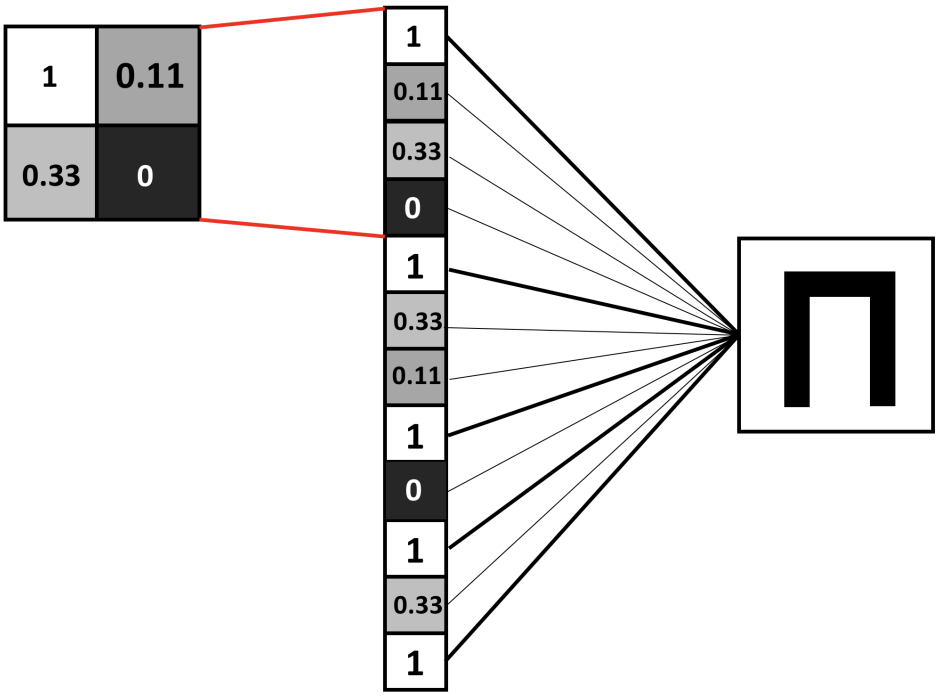


Fig. 15: Fully connected Layer

5. Backpropagation
   The final step is the so-called backpropagation. In this step the gradients of the error with respect to all weights in the network is calculated in order to use gradient descent to update all filter values, weights and parameter values to minimize the output error.[74]

   Figure 16 shows an overview of all five steps involved in classifying an image with a CNN. Note that all of the explained layers can be stacked in order to improve the network and that the following image is just a very simple example of anCNN.[75]

[73] cf. Kim 2017
[74] cf. Karn 2016
[75] cf. Li et al. 2017

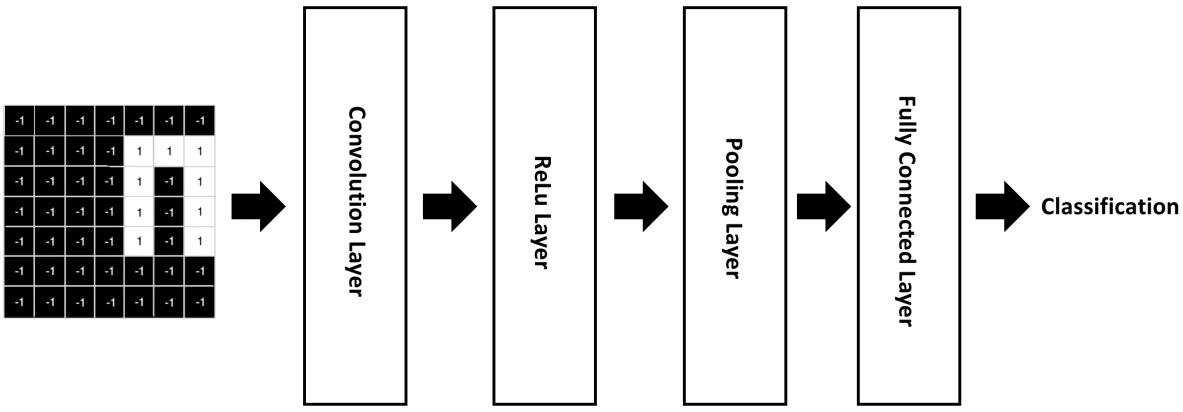


Fig. 16: CNN - Overview

3.3.3 Alternative Classifiers

Support Vector Machine

Support Vector Machines (SVM) can be used for both classification and regression problems. An SVM uses training data, splits them into different classes and represents them as vectors in a vector room.[76] In the binary classification case (only two classes), an SVM algorithm finds the hyperplane that gives the widest margin with respect to vectors of each class. Since the hyperplane is separating data objects from one class from the data objects of the other class by drawing a line between them (s. figure 17), this hyperplane sometimes is also called a classifying hyperplane.[77]

When new a data object is added, whatever side of the classifying hyperplane it lands on will decide the class which the SVM assigns to it. The further from the hyperplane the new data object lies, the more confident the SVM is that it made a correct classification.[78]

[76] cf. Ertel 2016, p. 298
[77] cf. Castaño 2018, ch. 9
[78] cf. Bambrick 2016

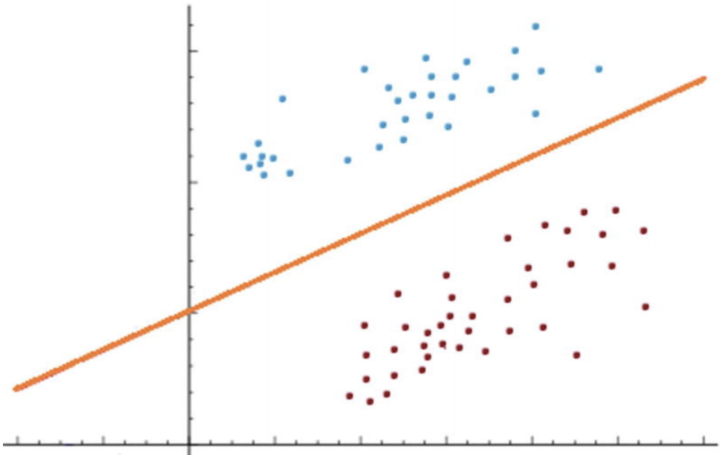


Fig. 17: Classifying hyperplane (SVM)[79]

Nearest Neighbour

Another method to classify images is to use the Nearest Neighbour (NN) approach. The NN is a classifier that does not need any parameter, since it processes and analyse the data directly without any prior training. Therefore the NN is a so-called *Lazy Learning Approach* as its only operation is to save categorized data.[80]

When classifying an image the NN calculates the difference between the input image and the already saved and categorized images. Subsequently, it performs probability calculation in order to make the classification.[81]

Decision Tree

The decision tree classifier is a simple but widely used classification technique.[82] A decision tree classifies inputs by sorting them in a tree structure from the root down to some leaf nodes, which provide the classification of the inputs. Every node in the tree specifies a test of a specific attribute of the given input, and each tree branch descending from the nodes corresponds to one of the possible values for this specific attribute.[83]

When classifying an input the algorithm starts at the root node, tests the input on the attribute specified by this node and will then move it down to the next branch. This process will be repeated branch after branch while the algorithm is always testing the corresponding attributes until a leaf node is reached and a classification can be made.[84]

[79] Castaño 2018, ch. 4
[80] cf. Ertel 2016, p. 206
[81] cf. Subramanian 2016, ch. 6
[82] cf. Kumar/Tan/Michael 1997, p. 150
[83] cf. Mitchell 2006, p. 50
[84] cf. Mitchell 2006, p. 50

Random Forest

The Random Forests (RF) method (s. figure 18) is a so called *Ensemble Learning Method*, which means that it uses multiple different algorithm in order to learn and make predictions.[85]

During the training, the method generates multiple simple structured decision trees. In order to make a classification all generated decision trees will be used. The result which occurs the most will be the final result of the classification.[86]

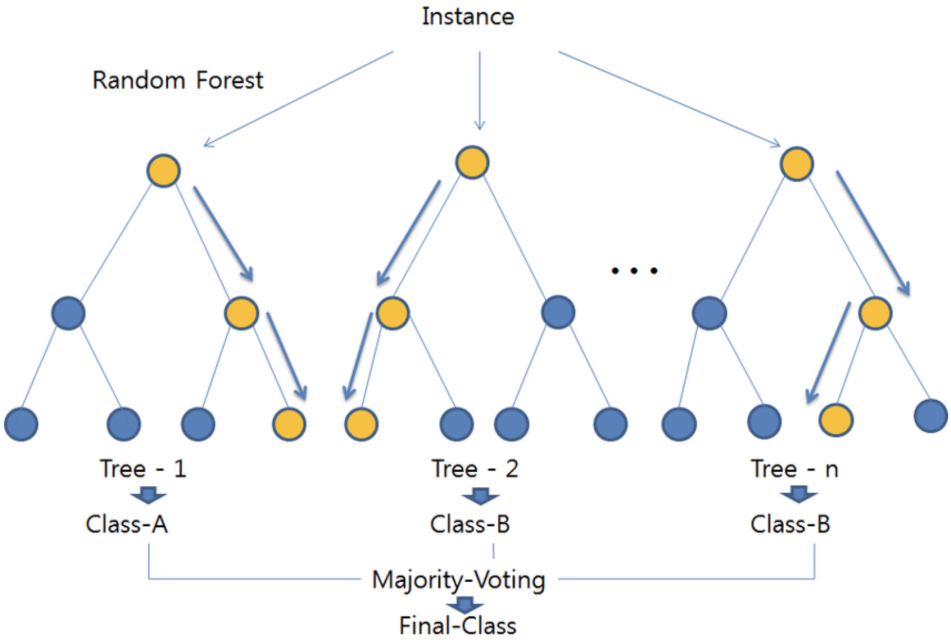


Fig. 18: Random Forest Classification[87]

[85] cf. Ertel 2016, p. 217
[86] cf. Suthaharan 2016, p. 273
[87] Lee et al. 2017

4 Implementation

This paper aims to implement a practical example to demonstrate the kind of technology used in Amazon Go's computer vision setup. Using the examined technologies and computer vision algorithms examined above the demonstrative implementation was undertaken. Amazon Go uses so called "sensor fusion"; a mixture of different sensor information.[88] In this case image data only shall be used. The envisioned result is to emulate Amazon's inventory control system. In the demonstration objects shall be visually identified by the sensor and added to the users shopping cart. In real life amazon uses several cameras to identify the objects customers have taken from shelves and adds it to their cart in order for them to pay for it. This required images to be taken analyzed and then based on the results a price shall be determined.

4.1 Technical Setup

The prior section examined various image recognition algorithms used in machine learning. Neural Network, Convolutional Neural Network, Nearest Neighbor, Support Vector Machines methods were examined. The functionality, pros, and cons of usage were examined. One of the most promising algorithms identified was convolutional neural networks. Convolutional Neural Networks have several advantages over standard Neural Networks when it comes to image processing, including lower processing power requirements and solving over-fitting issues.[89] It also extremely commonly used for image recognition, and several premade networks exist with excellent capabilities. A Convolutional Neural Network is used here in order to classify images.

In 2015 Google open sourced its deep learning framework under the name TensorFlow. Google had been developing TensorFlow since 2012 as an internal use framework called DistBelief. They had been winning several deep learning competitions using the framework. It became available to the public under the Apache license and has since been further developed by the community. As stated on their website "TensorFlow is a software library for numerical computation". It is often called a machine learning framework. TensorFlow arranges data into a structure called a "tensor" and uses a relational structure called a "graph" which relates nodes via edges. Simply put TensorFlow simplifies the process of creating your own artificial intelligence. Other similar frameworks exist such as Torch, Theano, and Infer.NET.[90] However, TensorFlow is quickly becoming the standard framework used. Its advantage is that it is well supported and documented. Additionally further simplifying layers such as Keras exist on top of TensorFlow making the process even easier. TensorFlow is used here.

[88] cf. Burgess 2018
[89] cf. Ravindra 2017
[90] cf. Sukaj 2017

TensorFlow API's exist for C++, Java, and Python. C++.[91] C++ is a low level programming language that promises lower processing power and time than high level alternatives. However it is less user friendly; syntax may not be intuitive, it requires managing memory, uses a compiler, etc.[92] Java is a high level object oriented programming language.[93] It is well documented and the standard language that is taught at the DHBW. Because of this it is well within the capabilities of the authors. However the TensorFlow documentation for Java is lacking. Most instructions are not laid out for this language. Python is another high level language.[94] It is known for being easy to learn with intuitive simple syntax.[95] The majority of TensorFlow documentation is laid out for Python as well.[96] For these reasons Python is used.

A decision is also made regarding the Convolutional Neural Network used. Building a Convolutional Neural Network is relatively simple using Python and TensorFlow. Multiple tutorials exist for this process and the complex mathematical basis is handled by TensorFlow.[97] Creating a Convolutional Neural Network is doable. However the network is relatively simple. A Convolutional Neural Network relies on multiple layers in which features are detected. A homemade network only has a few layers. Additionally the network needs to be trained to recognize objects. Collecting the images to form a comprehensive dataset for training can be cumbersome. The training process is computationally and time intensive. The made-from-scratch image recognition software is less accurate, recognizes less categories, and is requires more computing power. Several pre-trained Convolutional Neural Networks exist that can be used. These are image recognition algorithms that have already been trained with data and many classes. This allows for the training process to be skipped. Inception is one such pre-trained network. Google created this image recognition software to deliver world class results.[98] It is capable of identifying 1000 classes and has won several competitions for its ability and low computational load in relation to its performance. Inception works with many layers that have been optimized for this functionality. It uses layers within layers, the name is a reference to the similarity of this functionality to the movie. Figure 19 gives a brief overview of the Inception system.

[91] cf. Tensorflow 2018
[92] cf. Horne 2018
[93] cf. Jahari 2018
[94] cf. Yegulalp 2017
[95] cf. Rongala 2015
[96] cf. Brownlee 2016
[97] cf. Sachan 2018
[98] cf. Mulc 2018

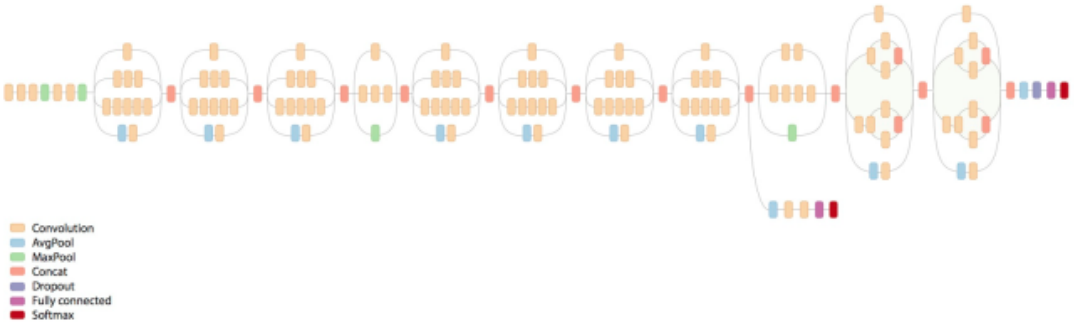


Fig. 19: Overview of the Inception Architecture[99]

Inception can also be retrained to identify additional classes, in case the pre-trained ones are not sufficient. Using Inception allows attainment of much better results than creating a Convolutional Neural Network from scratch. Finally a Python module is required for handling the capturing and processing of images. The Python Imaging Library or PIL is generally the standard used. Since PIL development has stagnated a fork of the software, Pillow, is generally used. However a more sophisticated image manipulation library exists that is specifically designed for computer vision. This library is called Open Source Computer Vision or OpenCV. This software is implemented in many languages including Python in the opencv-python package.[100] OpenCV is used here.

4.2 Demonstration

Two programs where created in order to achieve the desired results. The first handles the image capture and storage. It uses the webcam in order to take a picture and then stores this. The second program implements TensorFlow and Inception in order to process and classify the image taken in the first step. In order to take and image the first program begins by importing dependencies required. Chief among these is the OpenCV package. This must first be installed via pip. From there it is imported into the Python script:

```
1 import cv2
```

This is only one of several dependencies imported in a similar way. The others are used to handle tasks as required for example handling math functions etc. In order to capture images a cam object is created using the OpenCV package modules and the video feed from the webcam read in. A new window is opened in which the video feed is displayed.

```
1 cam = cv2.VideoCapture(0)
2 cv2.namedWindow("Amazon Go Capture")
```

[99] cf. Chu 2018
[100] cf. Reitz 2018

The program is written so that an image is taken whenever the spacebar is pressed. When the image is captured it is saved using a name which appends the image counter number. An image object called "newest" is created which is set to the latest image in the directory. When escape is hit the image capture window closes.

```
img_counter = 0

#Invoke image caputure

while True:
    ret, frame = cam.read()
    cv2.imshow("Amazon Go Capture", frame)
    if not ret:
        break
    k = cv2.waitKey(1)

    if k%256 == 27:

        #ESC pressed

        print("Escape hit, closing...")
        break
    elif k%256 == 32:

        #SPACE pressed

        img_name = "opencv_frame_{}.jpg".format(img_counter)
        cv2.imwrite(img_name, frame)
        print("{} written!".format(img_name))
        img_counter += 1
        newest = max(glob.iglob('C:/path.[Jj][Pp][Gg]'), key=os.path.getctime)
```

Now that the image has been taken it is ready for classification. The second script "classify_img.py" has already been imported as ci and its inference method is now invoked passing it the image "newest" as an argument.

```
ci.run_inference_on_image(newest)
```

The code used for the image classification is largely reused from the TensorFlow Models GitHub repository which includes several tutorials. The script will download and extract Inception if not done already. It will import TensorFlow. From there it sets up various tensors and define the relations via a graph. These are all premade in the tutorial and function as a black box. The important code that is called upon is the "run_inference_on_image" function.

```
def run_inference_on_image(image):
  """Runs inference on an image.

  Args:
    image: Image file name.

```

```
  Returns:
    Nothing
  """
  if not tf.gfile.Exists(image):
    tf.logging.fatal('File does not exist %s', image)
  image_data = tf.gfile.FastGFile(image, 'rb').read()

  create_graph()

  with tf.Session() as sess:

    softmax_tensor = sess.graph.get_tensor_by_name('softmax:0')
    predictions = sess.run(softmax_tensor,
                           {'DecodeJpeg/contents:0': image_data})
    predictions = np.squeeze(predictions)

    node_lookup = NodeLookup()

    top_k = predictions.argsort()[-1:][::-1]

    for node_id in top_k:
      human_string = node_lookup.id_to_string(node_id)
      score = predictions[node_id]
      print('%s (score = %.5f)' % (human_string, score))
```

The code reads in the image creates a tensor containing the Inception algorithm and checks the image against the classes contained therein. It transforms the class into a human readable string and returns the confidence percentage.

5 Conclusion

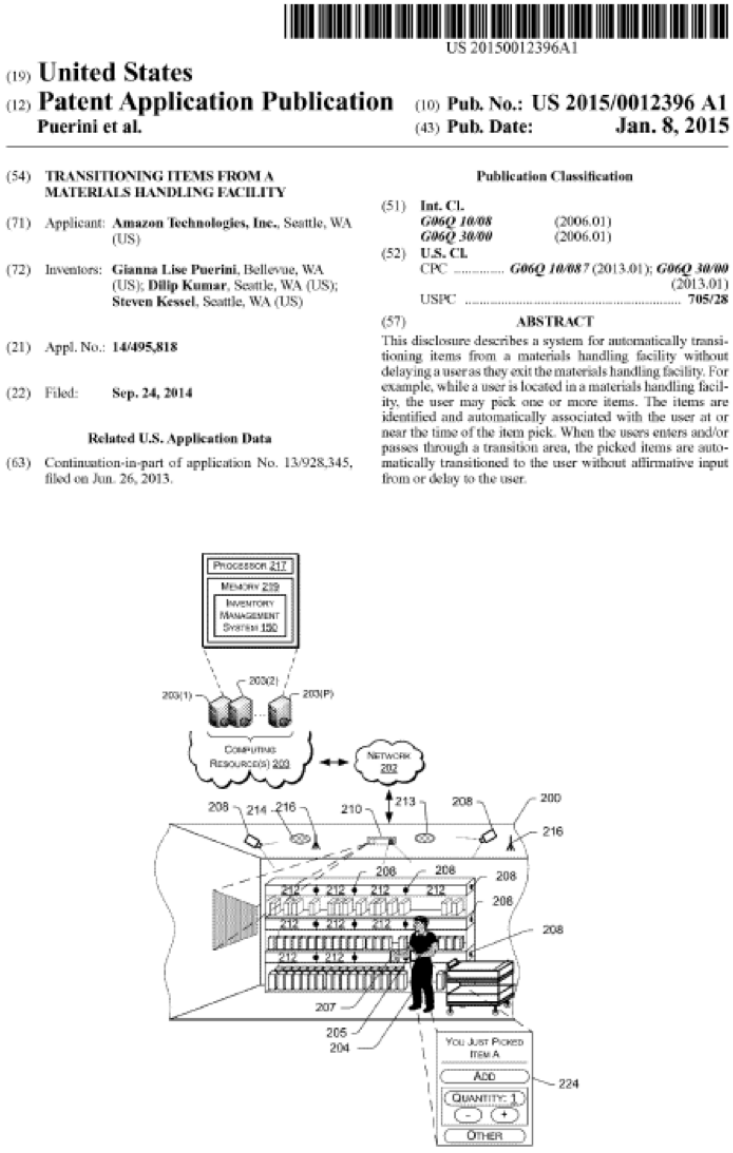
This whitepaper successfully explored the enablement of image recognition using convolutional neural networks. The scientific backdrop to carry this undertaking was in the form of the Amazon Go use case. Identifying Amazon Go as an innovative and therefore deeply disruptive company in the years to come, helped to justify why the use case was chosen. A large part of this undertaking was to clearly point out, just how the Amazon Go stores operate. This was done by examining three perspectives: the view of the user, the business model and the technical components. Moreover, the theoretical construct, consisting of computer vision, machine learning and image classification, led to a balanced foundation to continue the research. After presenting the Amazon Go use case, theoretical knowledge about computer vision was transferred. This not only explained the technology at hand, but also came up with examples in day to day scenarios and dealt with limitations. Furthermore, machine learning and its algorithm landscape were introduced theoretically. The final step in the technical setup overview, deep-dived into convolutional neural networks and alternative classifiers. The aim of the theoretical part was to convey insight into computer vision and machine learning, but more importantly to interlace these two technologies into image classification. Finally, the implementation chapter thoroughly validated the research by using TensorFlow and Inception in order to process and classify taken images as well as the Python script. This step by step demonstration created context and offered tangible results to the theoretical foundation laid beforehand.

With Amazon taking the lead in the immensely transformative potential of machine learning in consumer environments, it is only a question of time until other market players are forced to adjust and follow. The enhancement of technological performance and the reduction of political as well as cultural barriers, will mark a turning point in the way goods are consumed. Then, the market will be flooded with innovative store concepts and one great solution will follow the next. This whitepaper broke down just what it takes to enable image recognition using convolutional neural networks. However, the future promises to scale this foundation exponentially.

Appendix

Appendix listing

Appendix 1: Amazon Go Patent Application Publication



US 20150012396A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2015/0012396 A1**

Puerini et al. (43) **Pub. Date: Jan. 8, 2015**

(54) **TRANSITIONING ITEMS FROM A MATERIALS HANDLING FACILITY**

(71) Applicant: **Amazon Technologies, Inc.**, Seattle, WA (US)

(72) Inventors: **Gianna Lise Puerini**, Bellevue, WA (US); **Dilip Kumar**, Seattle, WA (US); **Steven Kessel**, Seattle, WA (US)

(21) Appl. No.: **14/495,818**

(22) Filed: **Sep. 24, 2014**

Related U.S. Application Data

(63) Continuation-in-part of application No. 13/928,345, filed on Jun. 26, 2013.

Publication Classification

(51) **Int. Cl.**
G06Q 10/08 (2006.01)
G06Q 30/00 (2006.01)

(52) **U.S. Cl.**
CPC ***G06Q 10/087*** (2013.01); ***G06Q 30/00*** (2013.01)
USPC **705/28**

(57) **ABSTRACT**

This disclosure describes a system for automatically transitioning items from a materials handling facility without delaying a user as they exit the materials handling facility. For example, while a user is located in a materials handling facility, the user may pick one or more items. The items are identified and automatically associated with the user at or near the time of the item pick. When the users enters and/or passes through a transition area, the picked items are automatically transitioned to the user without affirmative input from or delay to the user.

Fig. 20: Amazon GO - Patent

Bibliography

Literaturverzeichnis

Ali, A.-R. (2017): *Deep Learning Applications in Medical Imaging.* URL: https://www.techemergence.com/deep-learning-applications-in-medical-imaging/ (visited on 05/28/2018).

Amazon Mobile LLC (2016): *Amazon Go - App.* URL: https://play.google.com/store/apps/details?id=com.amazon.ihm.richard&hl=de (visited on 05/28/2018).

Amazon.com (2018): *Amazon Go, Frequently Asked Questions.* URL: https://www.amazon.com/b?ie=UTF8&node=16008589011 (visited on 05/28/2018).

Athiwaratkun, B./Kang, K. (2017): *Feature Representation In Convolutional Neural Networks.* Scientific Paper. Cornell University - New York.

Awad, M./Khanna, R. (2015): *Efficient Learning Machines, Theories, Concepts, and Applications for Engineers and System Designers.* Berkley, CA: Apress.

Bambrick, N. (2016): *Support Vector Machines: A Simple Explanation.* URL: https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html (visited on 05/28/2018).

Begg, R. (2017): *How AI and image recognition are transforming social media marketing.* URL: https://martechtoday.com/ai-image-recognition-transforming-social-media-marketing-202838 (visited on 05/28/2018).

Betters, E. (2018): *What is Amazon Go, where is it, and how does it work?* URL: https://www.pocket-lint.com/phones/news/amazon/139650-what-is-amazon-go-where-is-it-and-how-does-it-work (visited on 05/28/2018).

Bothe, H. (1993): "Mustererkennung". In: *Fuzzy Logic*, P. 174 - 191.

Brownlee, J. (2016): *Introduction to the Python Deep Learning Library TensorFlow.* URL: https://machinelearningmastery.com/introduction-python-deep-learning-library-tensorflow/ (visited on 05/30/2018).

Burgess, M. (2018): *The technology behind Amazon's surveillance-heavy Go store.* URL: http://www.wired.co.uk/article/amazon-go-seattle-uk-store-how-does-work (visited on 05/28/2018).

Cakmak, U. M./Das, S. (2018): *Hands-On Automated Machine Learning.* Birmingham, United Kingdom: Packt Publishing.

Castaño, A. P. (2018): *Practical Artificial Intelligence: Machine Learning, Bots, and Agent Solutions Using C#.* New York, United States: Apress.

Chu, V. (2018): *We Need to Go Deeper: A Practical Guide to Tensorflow and Inception.* URL: https://medium.com/initialized-capital/we-need-to-go-deeper-a-practical-guide-to-tensorflow-and-inception-50e66281804f (visited on 05/30/2018).

Dev, D. (2017): *Deep Learning with Hadoop.* Birmingham, United Kingdom: Packt Publishing.

Dutta, S. (2018): *Reinforcement Learning with TensorFlow.* Birmingham, United Kingdom: Packt Publishing.

Ertel, W. (2016): *Grundkurs Künstliche Intelligenz. Eine praxisorientierte Einführun*. Wiesbaden: Springer.

Franz, M. (2017): *Mustererkennung und Klassifikation*. Lecture. Hochschule Konstanz.

Fungineers (2016): *How does Amazon Go work, Amazon Go store explained*. URL: https://www.youtube.com/watch?v=d9kj45Of4jw (visited on 05/28/2018).

Géron, A. (2018): *Neural networks and deep learning*. California, United State: O'Reilly Media.

Gutierrez, D. (2018): *Amazon Go – Deep Learning Conquers Retail*. URL: https://insidebigdata.com/2018/02/15/amazon-go-deep-learning-conquers-retail/ (visited on 05/28/2018).

Horne, B. (2018): *A Comparison of Programming Languages*. URL: https://fusion809.github.io/comparison-of-programming-languages/ (visited on 05/30/2018).

Jahari, A. (2018): *What Is Java? A Beginner's Guide to Java and Its Evolution*. URL: https://www.edureka.co/blog/what-is-java/ (visited on 05/30/2018).

Kapur, S. (2017): *Computer Vision with Python 3*. Birmingham, United Kingom: Packt Publishing.

Karn, U. (2016): *An Intuitive Explanation of Convolutional Neural Networks*. URL: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/ (visited on 05/28/2018).

– **(2017)**: *A Quick Introduction to Neural Networks*. URL: https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/ (visited on 05/28/2018).

Karpathy, A. (2017): *Convolutional Neural Networks for Visual Recognition*. Lecture. Stanford University.

Ketkar, N. (2017): *Deep Learning with Python, A Hands-on Introduction*. Berkley, CA: Apress.

Kim, P. (2017): *MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence*. New York, United States: Apress.

King, P. (2016): *What are the major open problems in computer vision?* URL: https://www.quora.com/What-are-the-major-open-problems-in-computer-vision (visited on 05/28/2018).

Kolbrück, O. (2016): *So funktioniert Amazon Go: Die Technik hinter dem Zauberwort „Sensor Fusion"*. URL: https://etailment.de/news/stories/Technologie-So-funktioniert-Amazon-Go-Die-Technik-hinter-dem-Zauberwort-Sensor-Fusion-20194 (visited on 05/28/2018).

Kumar, V./Tan, P.-N./Michael, S. (1997): *Introduction to Data Mining*. Minnesota, United States: McGraw-Hill Science.

Learned-Miller, E. G. (2011): *Introduction to Computer Vision*. Amherst: University of Massachusetts.

Lee, C. et al. (2017): "Informal Quality Data Analysis via Sentimental analysis and Word2vec method". In: 45, pp. 117–128.

Li, X. et al. (2017): "Deep Convolutional Neural Network and Multi-view Stacking Ensemble in Ali Mobile Recommendation Algorithm Competition". In: *Signal Processing and Communications Applications Conference - IEEE*.

Lillesand, T. M./Kiefer, R. W. (1994): *Remote Sensing and Image Interpretation*. New Jersey, United States: John Wiley & Sons.

Luckow, A. et al. (2016): "Deep learning in the automotive industry: Applications and tools". In: *IEEE International Conference on Big Data*.

McKinsey & Company (2018): "An executive's guide to AI". In: *McKinsey Analytics*.

Mitchell, T. (2006): *Machine Learning*. New York, United States: University of Minnesota.

Moscatelli, S./Kodratoff, Y. (2017): *Advanced Machine Learning Techniques for Computer Vision*. Scientific Paper. Université Paris.

Mulc, T. (2018): *Inception modules: explained and implemented*. URL: https://hacktilldawn.com/2016/09/25/inception-modules-explained-and-implemented/ (visited on 05/30/2018).

Oberoi, G. (2016): *Comparing the Top Five Computer Vision APIs*. URL: https://goberoi.com/comparing-the-top-five-computer-vision-apis-98e3e3d7c647 (visited on 05/28/2018).

Osterwalder, A./Pigneur, Y. (2011): *Ein Handbuch für Visionäre, Spielveränderer und Herausforderer*. Frankfurt: Campus Verlag.

Ravindra, S. (2017): *How Convolutional Neural Networks Accomplish Image Recognition?* URL: https://www.kdnuggets.com/2017/08/convolutional-neural-networks-image-recognition.html (visited on 05/30/2018).

Reitz, K. (2018): *Image Manipulation*. URL: http://docs.pythonguide.org/en/latest/scenarios/imaging/ (visited on 05/30/2018).

Rohrer, B. (2016): *How do Convolutional Neural Networks work?* URL: http://brohrer.github.io/how_convolutional_neural_networks_work.html (visited on 05/28/2018).

Rongala, A. (2015): *Benefits of Python over Other Programming Languages*. URL: https://www.invensis.net/blog/it/benefits-of-python-over-other-programming-languages/ (visited on 05/30/2018).

Sachan, A. (2018): *Tensorflow Tutorial 2: image classifier using convolutional neural network*. URL: http://cv-tricks.com/tensorflow-tutorial/training-convolutional-neural-network-for-image-classification/ (visited on 05/30/2018).

Shanmugamani, R. (2018): *Deep Learning for Computer Vision*. Birmingham, United Kingom: Packt Publishing.

Sherman, E. (2015): *20 years of Amazon's expansive evolution*. URL: https://www.cbsnews.com/news/20-years-of-amazons-expansive-evolution/ (visited on 05/28/2018).

Snow, J. (2018): *Jeff Bezos gave a sneak peek into Amazon's future*. URL: https://www.technologyreview.com/s/610607/jeff-bezos-gave-a-sneak-peak-into-amazons-future/ (visited on 05/28/2018).

Steger, C./Ulrich, M./Wiedemann, C. (2018): *Machine Vision Algorithms and Application*. Weinheim: Wiley-VCH.

Subramanian, G. (2016): *Python Data Science Cookbook*. Birmingham, United Kingdom: Packt Publishing.

Sukaj, E. (2017): *What is the best alternative to TensorFlow?* URL: https://www.slant.co/options/14689/alternatives/~tensorflow-alternatives (visited on 05/29/2018).

Suthaharan, S. (2016): *Machine learning models and algorithms for big data classificationn*. Boston: Springer.

Tensorflow (2018): *API Documentation.* URL: https://www.tensorflow.org/api_docs/ (visited on 05/30/2018).

Trinkwalder, A. (2016): "Netzgespinste, Die Mathematik neuronaler Netze: einfache Mechanismen, komplexe Konstruktion". In: *c't* 6, P. 130.

Turner, N. (2017): *Amazon Acquire Whole Foods for $13.7 Billion.* URL: https://www.bloomberg.com/news/articles/2017-06-16/amazon-to-acquire-whole-foods-in-13-7-billion-bet-on-groceries (visited on 05/28/2018).

Venkateswaran, B. (2017): *Neural Networks with R.* Birmingham, United Kingdom: Packt Publishing.

Walia, A. S. (2017): *Activation functions and it's types-Which is better?* URL: https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f (visited on 05/28/2018).

Wingfield, N. (2016): *Amazon Moves to Cut Checkout Line, Promoting a Grab-and-Go Experience.* URL: https://www.nytimes.com/2016/12/05/technology/amazon-moves-to-cut-checkout-line-promoting-a-grab-and-go-experience.html/ (visited on 05/28/2018).

Yegnanarayana, B. (2004): *Artificial Neural Networks.* India: Prentice-Hall of India Pvt.Ltd.

Yegulalp, S. (2017): *What is Python? Everything you need to know.* URL: https://www.infoworld.com/article/3204016/python/what-is-python.html (visited on 05/30/2018).

Yoruk, E./Etin, M. (2017): "Retail product recognition with a graphical shelf model". In: *Signal Processing and Communications Applications Conference - IEEE.*

Decleration

We hereby insure that we have personally authored this seminar paper with the topic: »Amazon Go as a Use Case to Enable ImageRecognition Using Convolutional Neural Networks «and have used no sources and aids other than those indicated.

I also insure that the submitted electronic version corresponds to the printed version.

(Place, Date) (Signature)