Friedrich Schiller University Jena
Faculty of Economics and Business Administration
Department of Business Information Systems
[redacted]

Use of Image Classification in PIM Systems

Master Thesis

Submitted to attain the degree of Master of Science (M. Sc.)

in Business Information Systems

Submitted by:
[redacted]

Submitted to:
[redacted]

Jena, 20.07.2020

Table of Contents

I List of Figures

II List of Tables

III Abbreviations

AI	Artificial intelligence
API	Application Programming Interface
CNN	Convolutional Neural Network
CRISP-DM	Cross-Industry Standard Process for Data Mining
DAM	Digital Asset Management
ERP	Enterprise Resource Planning
FN	False Negative
FP	False Positive
KPI	Key Performance Indicator
MAM	Media Asset Management
MDM	Master Data Management
PDE	Product Data Enrichment
PDM	Product Data Management
PIM	Product Information Management
PLM	Product Lifecycle Management
ReLU	Rectified Linear Unit
TN	True Negative
TP	True Positive

1 Introduction and Motivation

Modern economies are characterized not only by the growing number of market participants but also by a steadily increasing range of product variations [36]. Therefore, the higher number of products offered is associated with a more significant amount of product data that must be managed [1]. To ensure efficient communication between market participants, meaning that information of different formats is collected, and no data is lost or duplicated, central information systems such as the *product information management* (PIM) systems are applied [78]. The primary purpose of these systems is to prepare product data for marketing purposes, which is to enrich necessary product data with additional attributes and descriptions so they can then be published for marketing purposes on the website, in social media, or the classic print catalog [110]. However, maintaining PIM systems can prove to be very time-consuming in practice, mainly because processes within product data enrichment must still be carried out manually, therefore, also causing inconsistencies in the product data [9].

A topic currently discussed in the media is the expansion and use of procedures from the field of *artificial intelligence* (AI). The developments on the market show that more and more companies are implementing AI-based solutions in their systems and organizations to open up new applications and make existing processes more efficient [15]. In this connection, deep learning as a part of AI seems to be very rewarding. Thus, many large tech companies already use deep learning for commercial purposes [43]. Furthermore, AI has arrived in manufacturing companies [2], and retail [15] and will be used even more widely in the future. In this context, dotSource GmbH[1], a German digital agency that advises customers from the e-commerce sector about PIM systems, has communicated the need for research to the extent to which the integration of an AI solution might generate optimization potentials within PIM systems.

Text-based AI processes already exist in PIM systems [113], [116]. Thus, of particular interest is the question of whether other AI procedures could also create advantages. Because a part of PIM systems deals with organizing product images for catalog creation, this thesis examines whether a computer vision technique based on deep learning, such as the image classification, may intervene in PIM processes and automate them at least

[1] www.dotsource.de

partially. Successful integration would, above all, promise a significant reduction in costs and an increase in efficiency [39]. Although there are plenty of different PIM systems available on the market [90], very little literature can be found on them [9]. On the other hand, in deep learning research, new scientific works can regularly be found in which scientists try to improve their deep learning constructs, yet, literature is scarce regarding the adoption of new AI procedures in existing business models [5]. Under the aspect that AI promises so many potentials and is used or at least planned to be used by many companies and industries in the next few years [2], [15], it would be interesting to take a closer look at the difficulties associated with such an application in a business context.

Since PIM systems are specially designed to support marketing activities in e-commerce by enriching product data, monitoring the results of an image classification procedure in this context is of importance, as incorrect information might be published. Rich and correct product information in online retailing can thus be regarded as a fundamental prerequisite for the success of a retailer as it influences the purchasing decisions of customers [30]. Thus, the integration of AI-systems raises explicitly the question of how the success and added value of these procedures should be measured and monitored in the operative business. This thesis intends to analyze these questions and close the gap in research concerning PIM systems and the business-related handling of AI-caused misclassifications. For this purpose, besides the theoretical development of the problem, an image classifier based on a convolutional neural network [69], [70] is tested in this study.

This thesis's content is structured as follows: The first section (chapter 2) gives a theoretical introduction of PIM systems, including their main features. Problems arising in the context of PIM systems are discussed in more detail, and the research question is defined. The next section (chapter 3) concentrates on the modeling of the application. The optimization potential of the procedure is pointed out, and specific monitoring measures are identified and developed. Chapter 4 covers the methodology, including the dataset and the structure of the image classifier, as well as the evaluation of the application. The methodology is based on the *Cross-Industry Standard Process for Data Mining* (CRISP-DM) framework, which was initially developed to manage data mining projects. Here, the dataset and the structure of the neural network are presented in more detail. Chapter 5 summarizes and discusses the results. A review of the methodology and possible prospects are also presented in this chapter. Finally, the conclusion is drawn, and an outlook for future work is given in chapter 6.

2 Product Information Management Systems

This section gives a brief overview of PIM systems, their main features, and how they differ from similar systems. Furthermore, the major issues of PIM systems are pointed out, followed by a first introduction to the objectives of this thesis. Since scientific research on the structure and the processes of PIM systems is very limited [9], the following description will primarily draw on the work of Abraham [1].

2.1 Main Processes and Features of PIM Systems

PIM is a rather new concept that primarily finds its application in e-commerce [1]. In this regard, PIM systems serve as centralized information systems for the optimal management of product information and related media data [1]. It is becoming more and more important, as online trade is forcing companies to offer a more significant number of product variations compared to physical business [1], [14]. Consequently, the broader product range follows a larger amount of product data that must be managed [1]. Since traditional spreadsheets are no longer efficient above a particular data volume, systems such as the PIM are needed to cope with the increased requirements [1].

With the expansion of e-commerce, the demand for complete and higher-quality product information is continuously growing as it can fundamentally influence the purchasing decisions of customers [21], [89]. In this context, comprehensive product information is one of the essential product features traded by webshops [30]. The benefits are a higher margin for companies and higher satisfaction for customers in the long run [78]. At the same time, the product information must be organized through more and more channels. Online trading in this connection also means distributing product information via websites, social media, and print, as well as on different end devices such as desktops, tablets, or mobile phones [50]. That is where PIM systems show their advantages since they are used in cases where product information has to be frequently synchronized, enriched for marketing purposes, and published via various channels and devices [110]. Because trade is becoming more and more complicated due to the stronger segmentation of customer groups, the continually increasing number of products and suppliers, and the growing internationalization of markets, integration of systems such as PIM, for enabling the effective organization of product data in the corporate context, is gaining in importance [1].

Abraham [1] offers an overview of the individual processes and components of PIM systems in this connection. Table 11 in the appendix summarizes the main features. According to him, all product-related data must first be collected from different sources in different formats, sorted, and transformed. That includes importing data from systems such as *enterprise resource planning* (ERP) systems, point of sale systems, procurement systems, product suppliers, data suppliers, or media agencies. He also explains that the sources of information used by a company differ in each case. Thus, in large companies, master data storage sometimes adds up to 20 different system. A common way to import this data is via catalogs or entire master category trees. Typically, structured formats for data imports include Text, CSV, XML, BMECat, or iDoc [1].

Often product information provided by the supplier is not enough to be used for further operational purposes. Suppliers keep on transmitting information in different unstructured formats, like PDF, CD, or even links, which generally makes automated data management difficult or even impossible [1]. The retailer himself must then create and integrate the basic information into the ERP as it is the central hub for the management of operational product data [1]. If the data import is to be automated, Java *application programming interfaces* (APIs) and web services can be utilized for synchronization. This way, direct connections between the supplier and the retailer's PIM system are established without additional effort to maintain the data [1]. Moreover, typical relational databases connected with PIM systems are Oracle, MS SQL, and MySQL, while standard APIs, in turn, control the interfaces with ERP systems [1].

The second step is the consolidation and cleansing of product data. Because different people and departments in companies often generate product data over several years, this step is needed to consolidate and sort out the data to avoid duplicates and to guarantee that each product is represented by only one instance [1]. Another part of consolidation is supplier consolidation. After all, multiple suppliers may provide the same product, yet information of varying quality. Thus it is necessary to bundle this diverging product information into one instance [1].

After the consolidation follows the data enrichment process, where products are divided into main and sub-categories using classification systems that support the management of product attributes [1]. Although it is possible to build individual classification systems, many PIM systems also support standardized or industry-specific variants such as

eCl@ss[2], ETIM[3], or UNSPSC[4] [1]. Because the imported product data is often limited to basic information such as identification numbers, prices, weights, and sizes, a process is needed to enrich this data with more detailed information and product descriptions. The applied product data enrichment process ensures to raise the information quality to a level where the data is utilizable for customer-oriented marketing purposes [1]. In this context, product-related media data such as photos, videos, manuals, or drawings are also linkable to the corresponding products [1]. However, the enrichment process is not limited to individual products, but it is extendable to entire product categories to which specific attributes and characteristics are assigned [1]. Also, PIM systems offer various possibilities to establish relationships between related products and thus carry out cross-selling activities. For quality assurance, the system does likewise support the execution of version controls and the assignment of access authorizations [1].

In the last step, according to Abraham [1], the product information is exported to the publication channels, depending on the final purpose, whereby a wide range of formats is available (e.g., Text, PDF, HTML, XLS, XML, CSV, JSON[5]). PIM systems are not publication channels themselves. Instead, they connect data to respective output systems like newsletters, websites, mobile apps, marketplaces, social media, and classic print catalogs. This way, the edited product data is presented to the customer for information purposes [1]. Similar to the data import, the export can function automatically, including real-time synchronization at predetermined time intervals via standard APIs and web services [1].

2.2 Distinction to Similar Systems

Since PIM systems and other product or process-related systems used in the corporate context share many properties, a distinction should be made between these individual concepts [1]. For example, PIM systems should not be confused with *product data management* (PDM) and *product lifecycle management* (PLM), whose concepts are hard to differentiate and thus mixed up in the literature [54], [55]. PDM supports design and de-

[2] www.eclass.eu
[3] www.etim.de
[4] www.unspsc.org
[5] The data format JSON is not mentioned directly in Abraham [1]. However, other sources show that this format is also generally supported [35], [127].

velopment processes and ensures storage of the relevant data, whereas, PLM has the purpose of maintaining the transition between product life cycle stages [55]. Thus, PLM may be considered process-oriented, while PDM has a more data-oriented approach instead [87]. Hence, PIM constitutes more of a sub-feature of the bigger PLM, whereas there is little difference with the PDM, as both are very similar in their function but serve different target groups [1].

Media assets are graphic information that does not consist of text [1]. The *media asset management* (MAM), also referred to as *digital asset management* (DAM), stands for the storage and management of marketing relevant media data like images, videos, drawings, or presentations [1]. Next to the upload and mass import of media data, their main tasks are the automatic linking to specific product categories or product instances as well as formatting media data for the use in different output channels. Thus, depending on the PIM system, MAM/DAM can represent an already integrated part of the PIM system itself or be a separate module linked via an interface [1].

ERP systems function as relevant data suppliers for PIM systems. The main difference between PIM and traditional ERP systems is that ERP systems mainly serve to manage operational processes (e.g., finance, logistics) [1]. In contrast, PIM is generally used to manage product information centrally for all kinds of customer-based marketing purposes [1]. Although ERP systems could theoretically take over the tasks of PIM systems, this would require an extraordinary technological effort, as these systems were initially not designed for such a purpose [1].

Master data management (MDM) systems go beyond PIM systems since they unite all company-related core data in a comprehensive view, including strategic, organizational, and technological data, thus providing benefits to various departments of a company [78]. Moreover, thanks to holistic data consolidation, new data-driven applications from the field of artificial intelligence are also possible [78].

In summary, PIM mainly differs from other similar systems by the centralized enrichment of product data for marketing and sales purposes. This property of PIM will be the basis for further theoretical elaboration and the use case considered in the following sections of this thesis.

2.3 PIM Systems' Problems

As the size of a company increases, so does the amount of data to be managed. In this relation, the number of manually performed working steps also rises, which poses a higher risk of error [65]. In the expanding area of e-commerce, retailers often only receive basic product data from their suppliers, like identification numbers, names, weights, sizes, and prices, which do not meet the required quality standards for online shops. In such cases, retailers themselves must ensure to upgrade the product data via the data enrichment process, i.e., to add additional attributes and descriptions to the individual products [1]. Some PIM systems already perform automated keyword extraction from supplied information and create product descriptions based on those keywords [91]. However, these procedures naturally demand the availability of product texts for functioning. If no product information is available, these text-based solutions may not work, and thus, entering attributes into the PIM database must again be done manually or at least requires manual preparation as in the classical enrichment process.

The influence of seasonal trends in the fashion industry plays a significant role in product management [34]. Nowadays, some retailers even deal with up to 24 collections per year [93]. The online retailer ASOS offers on its website between 2,500 – 7,000 new products per week from its production and third-party brands [106]. Another example from Adidas shows that they develop 20,000 new products each season, resulting in 40,000 new items by the end of the year [29]. Taking the last number and assuming that each new product is to be enriched by three attributes, this would result in a total amount of 120,000 attribute assignments every year. Adidas's example, moreover, demonstrated that the assignment of more than 20 attributes to every new product is a task manual work alone can no longer accomplish [29].

This simple calculation shows how effortful it can be to manage data within the PIM system. Adding new products thus requires a lot of manual work. It also entails a risk of data inconsistencies arising from incorrect entries, which adversely affects the total quality of the data [71], [65]. Yet, it seems that difficulties also stem from the correct handling of the heterogeneity of the information exchanged [33]. Different people can describe the same products in various fashions, which eventually leads to inconsistencies in the data [33]. Therefore, employees must have a uniform conception of the processes in the PIM [65]. Standardized methods can help in this connection to systemize the tasks and the way

of collaboration so that some of the mistakes resulting from missing knowledge or skills can be avoided [65].

In this context, a system would be desirable, which could take over a part of these manually executed working steps. Such a solution could eventually reduce human workload and human error at the same time and lower overall costs in connection with PIM administration. Since AI is currently regarded as one of the most demanded and promising technical solutions over the next years (see chapter 1), this work shall question the extent to which AI can be implemented beneficially in the PIM context. The PIM system Akeneo[6], for example, uses machine learning algorithms to build a product library that provides the user with access to publicly available product information. It automatically collects product features and supplies them to the PIM system during the product data enrichment process if required [44][113]. The PIM provider Contentserv[7], on the other hand, offers in cooperation with AX Semantics[8] a plug-in solution for automated text generation using natural language generation [116].

This thesis, though, pursues a different objective. Since a substantial part of PIM systems is the linkage of products with corresponding images (see chapter 2.1), the focus shall lie on the extent to which image-based AI procedures can generate efficiency gains. Image classification is of particular interest here, as it has been extensively researched in the past few years and has also found practical application in the meantime (see chapter 3.1). Figure 1 illustrates a simplified concept of how AI-based image classification could be applied to optimize the processes within PIM.

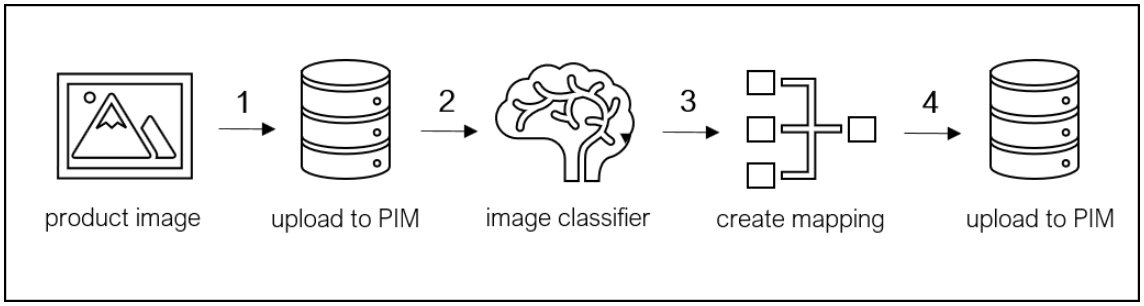

Figure 1: Basic idea of the AI application in PIM (own representation).

[6] www.akeneo.com
[7] www.contentserv.com
[8] www.ax-semantics.com

The primary goal is to build a neural network that independently classifies product images and therefore provides them with labels or product tags. For this purpose, a neural network is trained in advance with product images and labels corresponding to the individual company's industry.

When a new product instance, together with its product image, is uploaded to the PIM system *(1)*, the neural network is set to pick the product image up *(2)*. It classifies and outputs its associated labels (product attributes) *(3)*. These labels load back into the PIM system at the location of the corresponding product instance *(4)*. The image classifier ensures that products in the PIM system are automatically provided with appropriate attributes, meaning that the process of manual *product data enrichment* (PDE) is no longer required. Consequently, the method promises a faster and more consistent PDE as it avoids errors due to individual manual activities.

From this consideration, the central hypothesis of this work is derived:

> *H1: The automation of the PDE process using the AI-based image classification releases time and cost-saving potentials in the long term.*

3 Modeling the Integration of Image Classification into PIM Systems

The following chapter further develops the modeling of the application with a particular focus on PDE. First, more detailed information about the image classification technique is given (chapter 3.1). The second chapter discusses how AI-based image classification may be incorporated as a useful tool to support PDE (chapter 3.2). An additional goal is to identify any factors that might influence the process's general feasibility and economic efficiency. Therefore, key figures and monitoring measures are presented, which can help to manage the risk of misclassifications and associated costs (chapter 3.3).

3.1 Differentiation of Image Recognition Procedures

Chollet [22] explains that in deep learning, neural networks apply whose depth is determined by the number of information processing layers. According to him, the advantage of deep learning compared to traditional machine learning methods is that information is downsized to smaller parts without having to specify the structure of features in advance. This way, deep learning applies to more complex problems than machine learning, as laborious feature engineering is omitted, and feature learning is performed in one go [22].

Starting with the annual ImageNet Large Scale Visual Recognition Challenge[9] first held in 2010, Krizhevsky, Sutskever, and Hinton [64] presented a deep *convolutional neural network* (CNN) [69], [70], which surprised scientists with its performance in the field of computer vision[10]. Computer vision can thereby be roughly explained as the extraction and the understanding of visual information (e.g., images or videos) using AI models [60]. In this context, CNNs are used to a large extent for computer vision tasks [4]. Here, they have proven to be one of the best algorithms for image recognition applications [98], [60]. Various CNN-based methodologies and applications within the area of image recognition such as image classification [24], [118], object detection [112], [41], as well as segmentation [95], [7], exist. Even though there is no uniform definition within the literature [6], at least a rough outline of the main differences between individual computer vision sub-

[9] www.image-net.org/challenges/LSVRC

[10] The CNN achieved a top-5 test error rate of 15.3% in the classification context. For comparison, the next best error rate was 26.2% [64].

tasks shall be given to provide a better understanding of the later application part of this work.

Image classification, in its basic form, is about assigning a label to an image [95], [119]. Although CNNs have shown high performance in image classification on single-label problems [64], [104], research has shifted its focus towards multi-label classification, as it is supposed to capture better complex information and object relationships within real-world images [118], [119]. It should be clear regarding the terminology that the terms multi-class and multi-label are not equivalent. While the first case selects one class from a pool of mutually exclusive classes, the second case assigns several classes to one instance simultaneously [57]. Image classification is in this regard similar to object classification and object categorization, having the purpose of determining the presence of objects, regardless of their exact position on the image [73]. It should not be confused with object detection, which is more complicated than image classification as it combines the localization and classification of objects [74], [73].

Since visual properties such as the color or texture of an object associate with specific pixels of the image, the core idea of image segmentation is to distinguish individual image sections and objects on a pixel basis by either the clustering of similar regions or by detection of region boundaries [99]. Applications for image segmentation, for example, include scene understanding and autonomous driving [7]. According to Russakovsky et al. [98], there is a significant trend towards more complex procedures for image recognition applications. However, the following practical part of this thesis will be limited to the simplest case, the image classification, mainly since this study does not primarily aim to elaborate more sophisticated procedures, but rather for clarifying the applicability of a simple image classification task within PIM.

3.2 Optimization Potential of AI Integration

As already shown in chapter 2.1, the four main pillars of PIM are data collection, consolidation, product data enrichment, and data distribution. While the collection and distribution aim at organizing data flows and consolidation intends to avoid system-wide redundancies, PDE, in turn, targets the improvement of data quality for marketing activities.

Via the PDE, PIM distinguishes itself from similar systems such as PDM or MDM (see chapter 2.3). Because of that, this thesis concentrates only on the use case of PDE in the course of further modeling and practical application. Unfortunately, there are no instructions in the literature on how to organize PDE operations in PIM systems. Consequently, the following process description builds on aspects already mentioned in chapter 2 as well as on various real-world use cases, which are presented by Abraham [1]. Figure 2 illustrates the essential process steps of classic PDE.

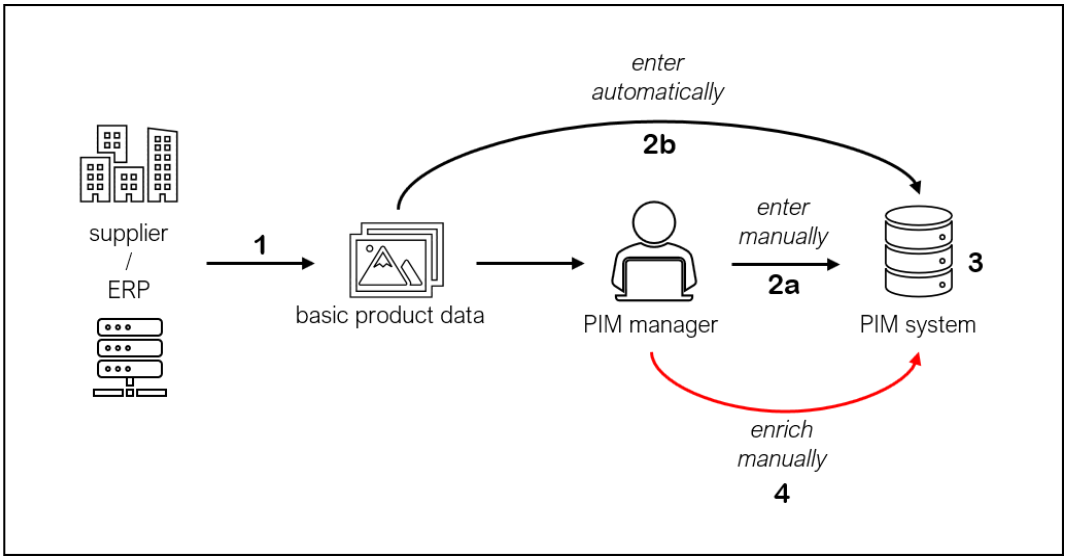

Figure 2: Classic PDE process (own representation).

Once the ERP or the supplier, via a direct link, delivers new data *(1)*, it is transferred either manually *(2a)* by an employee (e.g., PIM manager), or automatically *(2b)* to the PIM. The data is then consolidated and stored on the PIM system's server *(3)*, from where it can be retrieved and further processed within the scope of PDE. Since the information quality of the data is often insufficient to be presented to the customer, the employee must execute manual enrichment of the data *(4)*. The enrichment process includes attributes like product descriptions and media data, such as images (see chapter 2.1). The image data is stored in the MAM, which is either a direct part of the PIM or a separate module connected via an interface. The linkage between product instances and corresponding product images in the PIM is the essential prerequisite for the image classifier's deployment since it naturally requires input images to operate.

Figure 3 shows the visualized workflow of the new AI-based PDE process.

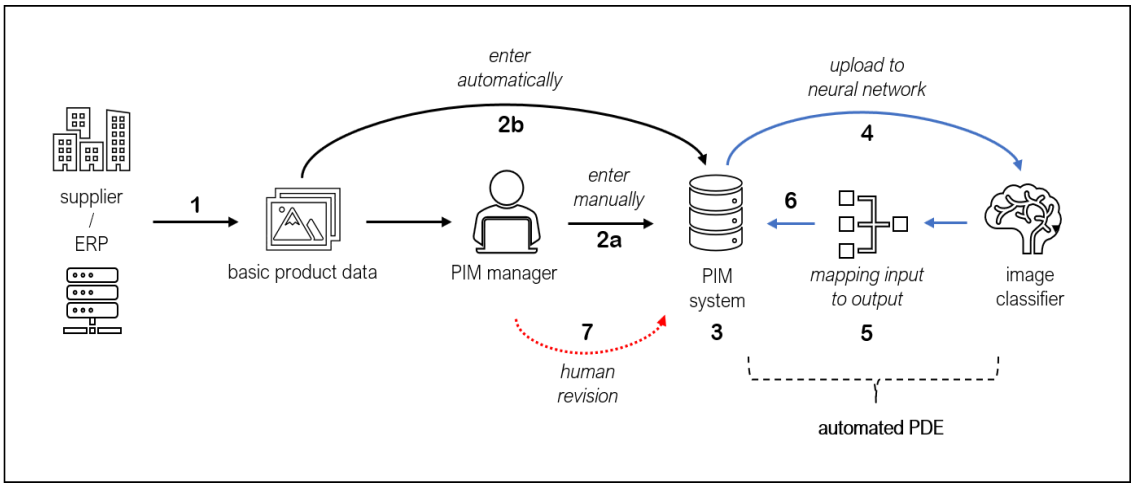

Figure 3: AI-based PDE process (own representation).

Steps *(1)* – *(3)*, i.e., data collection and consolidation, basically remain unaffected. What changes is the manual enrichment activity by transferring it on to the automated process of image classification. After the PIM server stores the new product image, it is retrieved by the image classifier for further analysis *(4)*. The input image is preprocessed and analyzed during the actual classification process, by mapping it to corresponding output classes (here: product attributes) *(5)*. The classification results are then stored in PIM-compatible formats (e.g., CSV, Excel, JSON)[11] and imported into the PIM the traditional way *(6)*. The automated PDE process itself is complete at this point. Steps *(4)* – *(6)* are carried out without human intervention, which means, that the manual working steps are now entirely transferred to the neural network.

A closer look at the process seems to underline the initial hypothesis that image classification in PIM saves manual workload and thus reduces working time and costs since the manually performed steps for PDE are left out. However, especially at the beginning, errors made by the image classifier cannot be excluded, which is why it is necessary to introduce a control measure such as human revision in an additional step *(7)*. The implementation of a new process for monitoring is also consistent with the recommendations of standardized analytical models from the field of data science such as the CRISP-DM [19] or Microsoft's Team Data Science Process (TDSP) [83].

[11] If the neural network is designed in Python, see, for example, pandas documentation [82].

The idea behind the human revision process is to check and, if necessary, to relabel incorrectly classified PIM entries. For the neural network to learn from these changes permanently, the new data instances could be collected and provided as additional training examples for later retraining measures (see, for example, [124]). Thus, through the revision process, the system could be optimized. Boosting the network's performance is therefore expected to steadily reduce the need for corrective actions within the process of revision as well as the overall manual workload.

There are various ways to execute the retraining of the network. One possibility would be to retrain everything from scratch. That could take place periodically, as a form of maintenance activity, or when adding new data to the system [124]. Another way to allow the system to be more dynamic is to use continual learning [86], [88], which also refers to continuous learning [80], incremental learning [18], or lifelong learning [72]. One thing these methods have in common is that they all ultimately attempt to adapt neural networks to cope with changing data conditions in real-time, without having to retrain the network's parameters from the ground up. Therefore, this technique is particularly suitable for real-time applications with limited memory and computing capacities, such as those found in smartphones or robotic systems [59].

The main problem is that this kind of training accompanies the risk of overwriting learned information with newly acquired knowledge. That is also known as catastrophic interference [81], [92], or catastrophic forgetting [38]. A solution for this issue would be to retrain the neural network from scratch with the original and new data. This measure would not be particularly efficient for real-time applications due to the amount of time involved [59]. It is questionable whether a real-time application regarding the introduced model of the image classification procedure in PIM systems would be overall useful. It might serve little purpose since the classified attributes, and their PIM entries are first to be reviewed manually before being approved for upload to an online shop or other publication channels. Thus, this work builds on the premise that it is sufficient to retrain the neural network with new product images at predetermined maintenance intervals.

It is also important to note that the final number of applied attributes may vary greatly depending on the industry or product type. For example, garments might only have 5-7 characteristics, while electronic items may have 30 or more attributes [1]. Yet, when considering that product features easily reach double-digit numbers within a PIM system,

multi-label classifiers seem to have more practical value than single-label classifiers, as they are more capable of capturing real-world object information [118], [119].

Although the image classification procedure is capable of achieving human accuracy in specific application areas [24], [48], the classification results should not be adopted without further verification. Especially in e-commerce, where product assortments quickly reach several thousand different items and product variations (see chapter 2.3), the demand for the image classification procedure can be very high. The entire procedure is challenging because product images can vary greatly in their visual representation. That includes image-related problems such as different perspectives and angles [31], the high similarity of products and material-related deformations [31], [46], as well as distorting factors such as blur and noise [28], might significantly affect the training and performance of the classifier. This circumstance makes it difficult to include all the necessary factors right from the first training process. Still, it highlights the need for the introduced human revision process and retraining of the classifier again.

A critical point to note is that even with a very high classification accuracy, a small percentage of incorrectly identified cases might remain. Nevertheless, as in Donati et al. [29], the achievement of sufficiently high classification performance is expected, allowing to further model the use of the application in a business context. That is also in line with information provided by the Fraunhofer Institute [27], which ascribes a high level of development to the image classification method based on deep neural networks. Thus, this thesis does less consider how the application of image classification can be solved technically, but rather how the advantages of the method can be exploited from an economic point of view. For this reason, it must be clarified how to deal with misclassifications that occur in the business environment as well as investigate to what different extend types of key figures may help to keep track of the overall performance and profitability of the entire application.

Within the literature, relatively little evidence and work on how to deal with the downside of AI applications, that is, potentially occurring misclassifications was found. That is problematic considering that in online trade, information quality is supposed to be a decisive factor in the buying behavior and satisfaction of customers, as already explained in chapter 2.1. One way to address this problem is to develop and implement appropriate monitoring measures to detect errors and control the classifier's performance. Such actions would ensure to exploit any gains of the AI in the longer run by the company [109].

3.3 AI Monitoring and KPI Development

A way to evaluate the performance of a classification method is to measure the accuracy of its predictions. Besides, the results can be displayed using a confusion matrix and corresponding metrics like precision, recall, or F-score. The confusion matrix is one of the first performance indicators to address in connection with multi-class problems since it illustrates the prediction accuracy of the classes as well as where confusion occurs [62].

Jiang & Cukic [56], for example, have specified the confusion matrix as follows: For a two-class problem, for example, where it is only a question of classifying a test as positive or negative, the confusion matrix consists of four categories. *True positives* (TP) are cases correctly classified as positive. *False positives* (FP) are cases incorrectly classified as positive. *True negatives* (TN), in turn, represent results that were correctly classified as negative, whereas *false negatives* (FN) are cases incorrectly classified as negative. Depending on the confusion matrix, precision and recall are also calculated.

Recall is the probability of detection of positive or negative modules.

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

Precision, on the other hand, is the proportion of correctly predicted positive or negative modules.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

That also covers multi-class problems, which are then treated as a two-class problem, where one class is examined as true or false against the others [76]. Besides, the F-score unifies the results of precision and recall and converts them into a single value, which is the harmonic mean of both values.

$$F - score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \tag{3}$$

In this respect, the F-score can help determine the optimal balance between precision and recall values for a model [105]. Figure 4 depicts the structure of the confusion matrix and the related metrics.

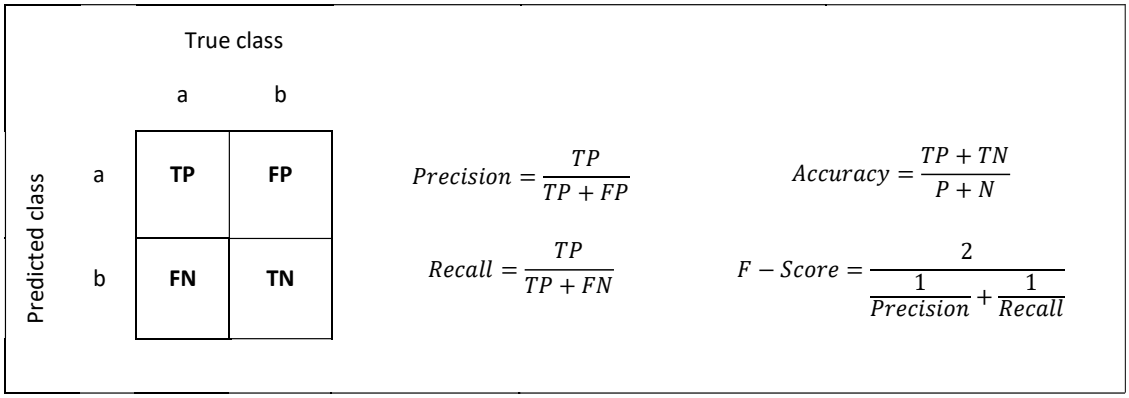

Figure 4: Confusion matrix and related performance metrics (modified from Fawcett [32], p. 862).

How serious the effects of misclassifications ultimately are must be considered concerning the specific application area. In critical fields such as medicine or fraud detection, misclassifications may have severe consequences [76]. In contrast, misclassifications in a PIM system lead to labels wrongly assigned to a product. That may result in the potential release in publication channels, which can later cause economic damage to the company. Yet, in comparison, this type of error seems to have less critical implications than, for example, misclassifications in medical applications, where in the worst-case lives may be at stake [76].

Nonetheless, misclassification costs are sometimes not considered, nor is a distinction between different cost characteristics. The classifiers are then primarily increasing the accuracy or minimizing error rates without recognizing that different types of misclassifications may have negative impacts in varying degrees [56]. Therefore, increasing accuracy alone is often not an adequate means, especially since it cannot generally be expected to reduce the costs of misclassifications or let costs be outweighed by any benefits [76].

According to Weiss & Provost [120], unbalanced data significantly impedes the classifier's performance. That is because underrepresented classes cannot be adequately learned in the training process and therefore have higher error rates in prediction. They also conclude that these minority classes are also the ones that cause higher misclassification costs. This happens when the imbalance in the database is due to the nature of the problem, for example, in fraud detection, where only a small proportion is fraud cases or in medicine where only a small number of cases is a rare disease [120]. However, within the

business environment, this problem is of less importance, as minority classes often are not the focus of attention [76]. Here, imbalances in the data sets can also result from constraints in data collection measures, such as legal restrictions [20]. Lombardi et al. [76] also mention, that particularly in the business context, the situation is more complicated since multiple factors make it difficult to assess costs accurately in connection with the classification results. They say that the costs depend not only on the actions a company takes but also on how customers react to these, for example, through changed purchasing behavior.

A cost matrix, according to Jiang & Cukic [56], is structured similarly to the confusion matrix and serves by taking different consequences of misclassifications into account. Correctly classified cases involve no costs, i.e., the costs for true positives and true negatives are equal to zero. A misclassification, on the other hand, is penalized with a corresponding cost value whose amount depends on the severity of the consequences of the respective error. In this context, the costs for false positives and false negatives may differ in their amount [56]. Chapter 2.1 already indicated that high-quality information, especially in e-commerce, is essential for the retailer's success. In this respect, it is important to differentiate which type of misclassification is more likely to be related to severe consequences: if a class is misclassified and the corresponding product is assigned an incorrect attribute or if a class is not recognized at all and, therefore, no PDE occurs.

With the growing popularity of online trade, an increased range of niche products, which would otherwise not be available in local stores, is becoming more critical. That is because stationary businesses can only offer a limited number of products due to area restrictions and because online search enables customers to look for particular products [14]. The image of both the product and the niche company is of vital importance for the retailer's success, whereby niche products tend to have a higher value, which may express itself in higher prices [25]. A niche is moreover of smaller size and tailored to meet specific customer requirements and wishes [103]. Because a smaller customer base is being addressed, the company needs to build long-term relationships with its customers [103].

In contrast to the advantage mentioned above of easier product search in online retailing, incomplete product information may lead to customers not being able to find particular products [10], causing the retailer to lose potential revenues and thus creating opportunity costs, as, for example, described in [94]. Incorrect product information could raise expec-

tations and motivate customers to buy specific items [100]. This may lead to dissatisfaction at the end and a higher return risk if the customer realizes that the purchased product is not in line with his or her expectations [11], [84].

Overall, niche products tend to meet higher quality requirements and demand higher prices, as described by Dalgic & Leeuw [25]. Therefore, it would be particularly critical not to achieve the desired quality standard by offering false information. Consequently, customer satisfaction is at risk, which may lead to negative reviews that significantly influence the general perception of the product [53], causing customer churn in the worst case, potentially associated with economic damage for the retailer [42].

Comparing the points mentioned above, missing information due to FNs may result in opportunity costs in terms of lost sales. Incorrect information in the PIM due to FPs, on the other hand, may lead to permanent damage, manifested by a loss of customers, bad publicity, and decreased sales. Therefore, FPs deserve a higher cost factor than FNs. The cost evaluation should moreover establish a reference to direct business performance, equivalent to what Studer et al. [109, p. 4] have called “Economic Success Criteria” in their framework. They explain that both the algorithm's performance and its influence on the company's success should be displayed and analyzed with various *key performance indicators* (KPIs). These KPIs include, for example, the amount of time and cost savings as well as increases in sales or product quality [109].

If sales figures or product quality are affected by the image classification procedure cannot be determined within the scope of this work. However, the performance indicators presented in this study could at least serve as a basis for business-related success measurements directly linked to the image classification processes, such as the number of misclassifications and the effort involved in correcting them. Regarding the monitoring of classification results during ongoing business operations, it is also not possible to create a confusion matrix since the ground truth labels of the predicted product images are not known to the classifier. A way to set up monitoring and relabel new image data is to implement the process of human revision, as described in chapter 3.2.1. Human revision in this regard includes manual quality control of the classification results and serves as a prerequisite for continuous optimization of the classifier. If misclassifications are present and corrective measures are needed, the wrong entries are relabeled, making them, along with the correctly classified entries, available as new and correct training data for later

training sessions. This way, for each successive training session, new data will be included in the training process. The training results can then be displayed in the subsequent evaluation using confusion and cost matrices to compare them with previous performances.

The following listing summarizes some suggestions, partially derived from the mentioned proposal of Studer et al. [109], as to which KPIs could reflect the performance of the classifier as well as the extent to which it causes business-related effects and expenses in the context of PIM (Table 1).

Table 1: KPIs and measures used after the image classification procedure.

	Measure	**Goal**	
a	Confusion matrix (incl. precision, recall, F-score, accuracy)	The traditional way to measure prediction accuracies and visualize the performance	(I) Focus on classifier performance
b	Cost matrix	Inclusion and emphasis of different misclassification costs (FP/FN)	
c	Time expenditure for human revision activities	Measurement of how much time was spent on human revision tasks; serves for total cost calculation	(II) Focus on business performance
d	Proportionate salary costs	Salary allocation depending on the time spent on human revision activities; serves for total cost calculation	
e	Total risk	Settlement of salary and misclassification costs; represents potential maximum of costs	Comparison between (I) and (II)

The key figures *(a)* - *(b)* represent classic measures connected with classification applications, as described above. The statistics *(c)* - *(d)*, on the other hand, provide information about the influence of occurring misclassifications on the success of the company. Figure *(c)* measures the amount of time spent on revision tasks. It is, in turn, the base for *(d)*, which adds actual costs in terms of salaries to the time spent. This figure shows how expensive the human revision process ultimately was. The ratios from *(I)* represent the performance and, thus, the potential risks of the classifier in terms of follow-up costs regarding corrective measures, while the values from *(II)* reflect the actual costs incurred. The last figure *(e)*, in turn, combines the classification risks with business-related performance measurements and enables different cost aspects to be represented in one number. This ratio represents the entire risk of the image classifier within the PIM application.

The purpose of these indicators is to keep an overview of the risk of the application, i.e., the incurred and potentially incurring costs. The whole point is to ensure that the costs do not exceed the benefits of the application. Using the entire scope of the metrics makes it easier to assess whether the procedure continues to yield cost advantages or whether increasing costs, either through poor classifier performance or through excessively long and expensive human revision measures, are leading to inefficiency of the entire procedure.

4 Methodology

CRISP-DM is a standardized and freely accessible process model intended to support the search for patterns and correlations during data mining projects [19]. It was developed in 1996 in cooperation with several big companies and with financial support from the European Union. The goal was to present an industrial standard that allows harmonization of the processes of data mining projects as well as to increase the quality of knowledge discovery in general. The six levels of CRISP-DM serve as a guide for the structure of the following practical part of this work, although the model has been adapted a little for this purpose (Figure 5).

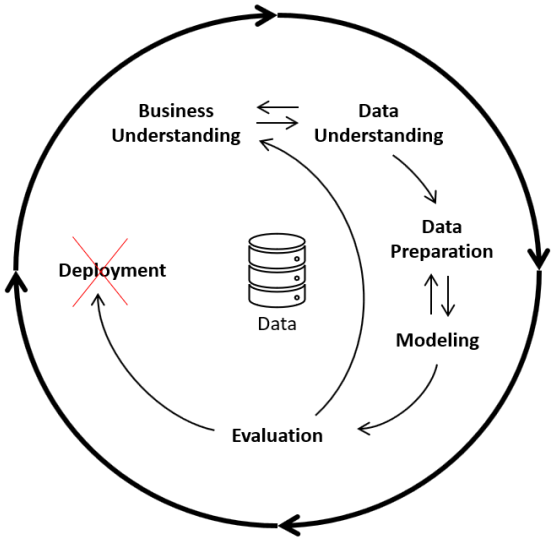

Figure 5: Adapted CRISP-DM model (modified from Chapman [18], p. 13).

The business understanding (chapter 4.1) contains a description of the purpose and the objective of this work. It is followed by the development of the data basis, meaning the data understanding (chapter 4.2) and the data preparation (chapter 4.3). After that, the image classifier's technical modeling is explained in more detail (chapter 4.4). The evaluation (chapter 4.5), on the other hand, includes the assessment of the classification procedure, where the classifier and business-related metrics developed for monitoring purposes (chapter 3.3) are applied and analyzed. As this work does not intend to perform a real-life implementation of the image classification procedure in a PIM system, the last step of the CRISP-DM (deployment), is excluded.

4.1 Business Understanding

The need for further investigation was communicated by the dotSource GmbH, regarding the extent to which the procedure of AI-based image classification is applicable in conjunction with PIM systems. As a digital agency, the company deals, amongst others, with the use of PIM and MDM systems in e-commerce. The interest in an AI-based PIM solution that generates efficiency and cost advantages is thus correspondingly high.

While this research question has not yet been explored in the literature and only a few reference cases on AI integration in PIM can be found on the World Wide Web, this thesis aims at analyzing the issue in an explorative way. The results of the theoretical analysis revealed that it is less the technical implementation one should worry about, but rather the question of how to profitably implement such an AI solution in the business context without causing additional costs or risks in the long term.

Because PIM systems mainly serve e-commerce activities, the use case is based on an industry that profits particularly strongly from the advantages of online trading. The fashion industry is selected for this purpose, since it is the largest business-to-customer segment in e-commerce and, according to forecasts, expected to continue to grow [108]. Over the last few years, various papers have dealt with the topic of image classification in the field of fashion, such as [67], [75], [31], [102], [34], [29], providing a solid theoretical and technical basis.

Lao & Jagadeesh [67] have differentiated between clothing type and clothing attribute classification. In their opinion, both variants represent sub-problems within the field of fashion classification, with the first being a multi-class procedure and the second a multi-label problem. To reduce complexity, the model within this thesis is designed as a multi-class problem, where only one class is assigned to each input image, similar to the clothing type classification of Lao & Jagadeesh [67].

For solving a multi-label problem, product images must be tagged with several labels at the same time. Yet, the applied data collection method only allowed the download of the product images in connection with their web link, and therefore, only coupled with their main product category. However, the chosen application design should be sufficient since the goal of this thesis is to examine whether there is an increase in the classifier's performance and a general reduction in costs over time as the number of training examples rises.

Regarding the hypothesis of cost-effectiveness put forward in this thesis, the question of whether the chosen classes ultimately represent product categories or specific product attributes such as color or patterns should not be of importance at first. The extent to which limitations arise in the context of real-life business applications is in more detail examined in the discussion of this thesis (chapter 5).

To further investigate the research question, the following presents a sample application where a CNN-based image classifier is trained and tested with data sets of various sizes containing fashion images.

The whole procedure is divided into three runs: in the first run, the network is trained with 250 images per class and then tested. In the second run, the training is done with the same parameters but with 500 training images per class. In the third run, the network is retrained with 1,000 images per class. The validation and test images remain the same in order not to cause distortion. The results of the three test runs will then be analyzed and compared using the performance indicators identified in chapter 3.3. The goal is to determine the extent to which the performance of the classifier can be optimized in the (here fictitious) time course by regular retraining with new product data. The results shall provide information on whether such an AI application in the PIM context, combined with human revision activities, leads to efficiency advantages in the long run in terms of time and cost savings.

4.2 Data Understanding

The data used in the methodology shall represent a cross-section of a typical online fashion store. The whole dataset consists of 12,000 images and is evenly composed of 10 different product classes: (0) bags, (1) coats, (2) dresses, (3) skirts, (4) jeans, (5) shorts, (6) sweaters, (7) T-shirts, (8) sneakers, (9) sandals. The selected classes are inspired by the Fashion-MNIST dataset published by Zalando[12] [122]. However, the Fashion-MNIST dataset was not used for this work, although it contains a broad set of prestructured data. That is because its images are only in grayscale and because the products are displayed in an isolated manner, i.e., without disturbing elements such as other product sections or body parts. Looking at images currently used in online fashion stores, it is evident that

[12] www.zalando.de

such a reduced representation does not quite meet the actual standards of online shops at the time of this work. To get a more realistic picture of the classification results and to uncover general practical problems in image classification, a new dataset was created.

For this purpose, a total of 11,478 women's fashion images was collected and downloaded from Zalando using R[13]. The remaining 522 images (mainly the category shorts) were taken from the DeepFashion dataset [75]. This step was necessary to avoid unbalanced class representations, which could otherwise negatively influence the training process, as described in [120]. Yet, the DeepFashion dataset was not entirely chosen since it includes a significant part of inferior quality images, meaning low resolution, blurred representation, diverse backgrounds, or wrong labeling. For this reason, a completely new data set was created in the end.

4.3 Data Preparation

Since algorithms need quality data to obtain good results and real-world data is often subject to errors [125], some methods like those used in classic data mining were applied to the data preparation. Overviews of general data preparation procedures are provided by Witten et al. [121] and Aggarwal [3].

The images were manually checked for duplicates and cleaned up, to prevent the classifier's results from being influenced by repetitive attributes [121]. The inspection for errors or possible outliers, i.e., for products that do not belong to the respective class, was carried out manually. Since clothing exhibits many different characteristics, the manual labeling of product images, especially for very similar products, is often associated with a risk of subjective evaluation, which leads to inconsistencies regarding the corresponding labels [67]. Therefore, disparities, in connection with the assigned product classes, may still exist. Subjective decision rules were also used to check the overall image quality. The criterion for these decisions was the visual assessment, i.e., pictures that the viewer could not identify since being too small or unsharp were sorted out. It was considered that no imbalances in the class distribution arise by deleting images, meaning that the same number of images was assigned to each product class.

[13] www.r-project.org

For the final dataset, care has been taken to ensure that the products within the images are centered, displayed in their entirety, and easily identifiable. Outliers in terms of product images that could not be assigned to one class by a human annotator (e.g., multiple products depicted, or no product in the foreground) were sorted out. However, this step was largely limited as in some classes, such as skirts, a significant part of product displays showed a combination of several products.

The results are comparable to those of Schindler et al. [102]. They distinguish between two different types of images: Whereas some images show single products against a white background, others depict multiple products presented or worn by a model. In this context, the latter causes semantic noise as a distorting factor since only one label is assigned to the image. In this thesis, it is expected that with diffuse product representations, as in the second case described by Schindler et al. [102], the classification process becomes more difficult. Therefore, misclassifications occur more frequently.

Finally, the data were divided into training, validation, and testing data, with the only difference that for each of the three training runs, the number of training data was increased. Table 2 shows the final ratio of the distribution.

Table 2: Distribution of training, validation and testing data for each training run.

Run		**Training**	**Validation**	**Test**	∑
A	Number of instances for each class	250	100	100	-
	Total number of instances	2,500	1,000	1,000	4,500
B	Number of instances for each class	500	100	100	-
	Total number of instances	5,000	1,000	1,000	7,000
C	Number of instances for each class	1.000	100	100	-
	Total number of instances	10,000	1,000	1,000	12,000

4.4 Modeling

4.4.1 Theoretical Background

When it comes to visual recognition tasks, CNNs are a common choice since they mainly specialize in dealing with grid-like structures such as images [4], [43]. Good introductions to this topic provide Goodfellow [43] and Aggarwal [4].

In this context, the term CNN suggests that these networks use convolution, a linear mathematical operation instead of the usual matrix multiplication, as in traditional neural networks [43]. The convolution layer applies a filter matrix that searches individual image pixel areas in an overlapping manner to detect and extract significant features. The result of this convolution operation is the feature map, which, in comparison, is equivalent to the values of hidden layers in traditional feedforward networks [4]. This form of fractional image processing ultimately results in fewer connections and parameters and makes training more manageable than with comparably sized standard feedforward networks [64].

According to Goodfellow [43], besides the just described concept of sparse interactions, convolution explicitly puts parameter sharing and equivariant representations in the foreground. Parameter sharing in this connection states that the weight matrix does not just use the elements once, but instead, uses the same parameters for multiple functions within the network [43]. This linkage leads to the equivariance to translation, meaning that as long as the pixel values of an image object remain constant, they are always calculated in the same way by the convolution operation [43].

CNNs differentiate in their structure between the input layer, hidden layer, and output layer. The main difference compared to traditional neural networks lies in the hidden layer, which itself is a construction of three different layers that repeat alternately: the convolution layer, the ReLU layer, and the pooling layer [4], [43]. The function of the convolution layer has already been described above. This layer is followed by the ReLU layer, which passes the feature maps through a non-linear activation function [4], [43]. Nowadays, one of the most popular activation functions is the *rectified linear unit* or ReLU [85], as ReLU has proven to be faster than other classic activation functions such as sigmoid or tanh [64], [68]. Therefore, this part of the hidden layer is often directly referred to as the ReLU layer [4].

In the next step, the activation maps produced by the ReLU are transferred to the pooling layer and replaced by a condensed representation of outputs [43]. Different methods such as max pooling [126] or average pooling are used, whereby the former being more popular due to its higher degree of translation invariance, which is the higher robustness against small variations of input features [4]. Therefore, pooling primarily helps distinguish whether a feature in the image is present at all, rather than determining its exact location [43].

After the information has passed the hidden layer, it moves to the fully connected layer, where it is mapped to a predefined set of outputs according to the original application problem [4]. The activation function is either the softmax or sigmoid function, depending on whether a multi-class or multi-label problem is present [34]. This layer has the same construction and functionality as a traditional feed-forward network, indicating that since the nodes are interconnected, most of the parameters are located here. Thus, the fully connected layer is crucial for problem-specific training of the entire network [4]. This property shall be adopted now. For this purpose, a generally pretrained neural network is being used and adapted to the defined use case of fashion image classification.

4.4.2 Technical Modeling

The model was created and tested in the Jupyter[14] notebook with Python (3.7.3)[15] using the PyTorch[16] (1.3.0) library. Since no suitable GPU was available, the program was run via the CPU. Table 3 summarizes the final parameters of the model.

The CNN is based on the method of transfer learning, where a pretrained neural network and its weights are adopted, and only the fully connected layer is modified to match the particular classification problem [4]. Even with high-quality data sets, the combination with pretrained models provides better results than if the neural network was trained for one problem-specific data set only [102]. Over time, various pretrained CNN architectures have evolved: AlexNet [64], Inception/GoogLeNet [111], VGG [104], or ResNet [49], to name a few. Since it had shown slightly better performance in the area of fashion classification compared to other architectures (Inception, VGG) [34], ResNet was chosen

[14] www.jupyter.org
[15] www.python.org
[16] www.pytorch.org

as the pretrained model, which is based on the ImageNet data set [26]. Various modifications of ResNet with different numbers of layers were tested (ResNet-50, ResNet-101, ResNet-152) [49], whereby, in this case, ResNet-152 (152 layers) showed the best performance.

The neural network's fully connected layer was adjusted to 10 output classes according to the original problem. It performs the activation via the ReLU function [85] and uses Dropout as a regulation method [51]. Since this case is a multi-class problem, the activation function for the output units is the softmax function (see chapter 4.4.1). The training process itself was carried out by the backpropagation algorithm as in classic neural networks [97], [69]. The cost function is the negative log-likelihood, minimized in the training process via gradient descent [43]. Therefore, ADAM [61] was chosen as an optimizer for the execution of the gradient descent, i.e., for the optimization of the weights.

Table 3: Selected parameters for the applied convolutional neural network.

Purpose	Parameter	Specification
Model architecture	• Pretrained model • Fully connected layer	• ResNet-152 • 10 output classes, ReLU, softmax for last layer
Training parameters	• Batch size • Learning rate • Number of epochs	• 32 • 0.0001 • 50
Regularization methods	• Image transformations • Dropout • Early stopping	• crop, rotation (15°), horizontal flip, normalization • 0.5 • Patience = 5
Technical components	• Programming language • Libraries • Environment • GPU use	• Python 3.7.3 • PyTorch 1.3.0 • Jupyter notebook • none

For the reduction of overfitting during training, various regulatory mechanisms were applied. An easy and computationally relatively inexpensive possibility is to increase the training data and its manifestations via transformations artificially [64]. Thereby, even small shifts of a few pixels are supposed to improve the network's generalization performance significantly [43]. For this purpose, the input images were rotated (by 12 degrees) and mirrored horizontally, as suggested by [43] and [64], respectively. Another method is the already mentioned dropout, by which individual hidden units are randomly excluded from the network, thus simulating different network structures [51]. For the dropout value, 0.5 was chosen [51], [107]. Finally, early stopping was implemented as a meta-algorithm. It operates like a threshold that determines after how many epochs, without further improvement of the validation error, the training stops [43]. Several different combinations of the learning rate, batch size, and other parameters relevant to the training were tested.

4.5 Evaluation

4.5.1 Classifier-related Performance

Both loss and accuracy were considered for training and validation purposes. The respective curves are shown in Figures 6 - 8, and the overall results are summarized in Table 4.

What is noticeable in Figures 6 - 8 is that the validation accuracy is always higher than the training accuracy, the validation loss, in turn, is lower than the training data. The ratio between both graphs seems to be stable. A possible explanation for this would be that dropout as a regulation method was applied to the training data but, on the other hand, was not used on the validation set. Thus, additional noise arose during training, distorting the training and validation results [40], [96]. A second explanation might be that the validation set's images are easier to learn or that the distribution of the validation and training set is different [40], [96]. Last but not least, the training and validation loss might have been calculated at different times within an epoch [40], [96]. The latter can be excluded, though, since both values were determined within the same epoch in the applied classifier. As the training loss is not smaller than the validation loss, overfitting is also excluded for now [58].

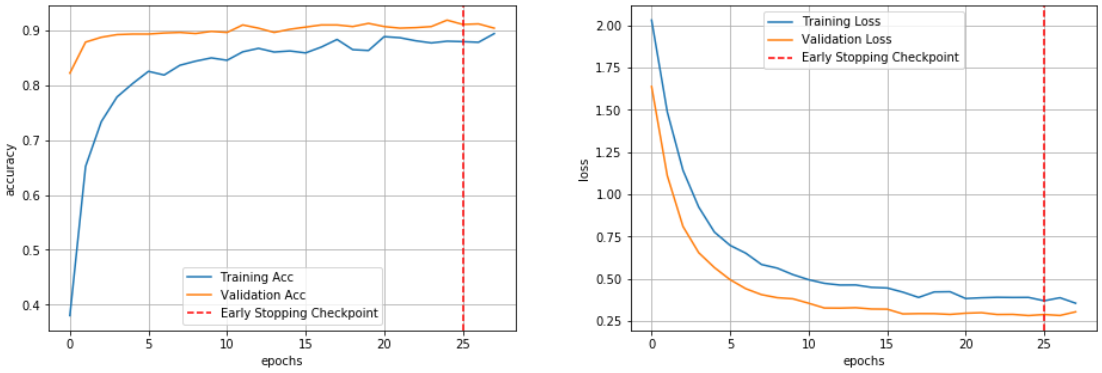

Figure 6: Accuracy and loss curves for A (2,500 training images) with early stopping point (here: epoch 25).

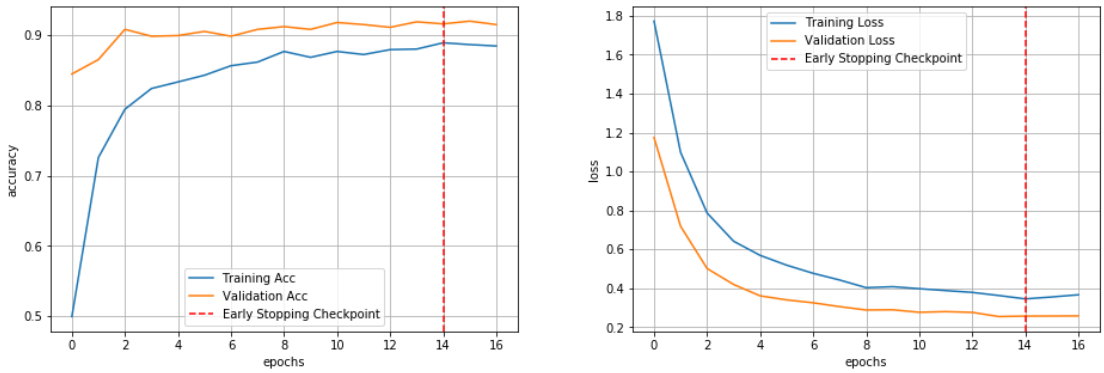

Figure 7: Accuracy and loss curves for B (5,000 training images) with early stopping point (here: epoch 14).

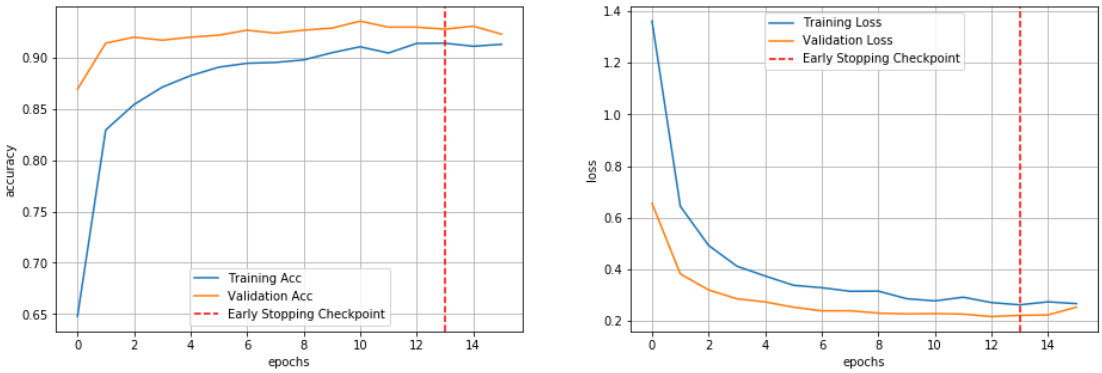

Figure 8: Accuracy and loss curves for C (10,000 training images) with early stopping point (here: epoch 13).

Table 4: Training time and test accuracy for data sets of different sizes (A: 250, B: 500, C: 1,000 training examples per class).

Model	Epochs	Time for Training (CPU only)	Test Accuracy (%)
A	25	ca. 16h 39min	92.4
B	14	ca. 16h 29min	92.9
C	13	ca. 1d 4h 10min	94.2

The results show that training for model A was stopped after 25, for B after 14 and for C after 13 epochs. Thus, the number of epochs tends to decrease as the number of training examples increases. The decreasing number of epochs would explain why model B completed faster than model A, even though training time per epoch was significantly higher (for epoch 1 in A: 2215.81s; in B: 3800.45s). However, this effect does not seem to repeat for the training time of C, which has the highest duration with slightly more than one day.

Overall, the number of epochs required seems to decline as the number of training data increases, whereas the training time needed per epoch in turn rises. To a certain extent, these effects balance each other out, resulting in faster convergence of model B. However, if the number of training data exceeds a certain point, the training duration seems to increase disproportionately compared to the reduction of epochs, as in model C.

The test accuracy has shown continuous improvement as the number of training data increased, with differences of 0.5% between models A and B and 1.3% between models B and C. At this point, these results already indicate that the classifier's performance improves with an increasing number of training examples (here, doubling the number of training images). For each model, the corresponding confusion matrix is provided in the appendix (Figure 14). The misclassification results from the confusion matrices are included in Table 5. The precision, recall, and F-score values derived from the confusion matrices are listed in Table 6.

Table 5: Number of misclassifications for each run (values derived from the confusion matrices A - C).

	A		**B**		**C**	
Class	FPs	FNs	FPs	FNs	FPs	FNs
Bags	1	0	1	0	1	0
Coats	1	9	3	5	3	5
Dresses	22	17	13	22	10	17
Jeans	1	1	1	2	0	0
Sandals	5	2	2	2	2	2
Shorts	5	7	9	5	6	7
Skirts	14	21	13	19	14	13
Sneaker	1	5	3	3	0	3
Sweater	12	9	12	9	10	7
T-Shirt	14	5	14	4	12	4
∑	76	76	71	71	58	58
∑	**152**		**142**		**116**	

Table 6: Precision, recall, and F-score (F1) values for each class and each run.

	A			**B**			**C**		
Class	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
Bags	0.99	1.00	1.00	0.99	1.00	1.00	0.99	1.00	1.00
Coats	0.99	0.91	0.95	0.97	0.95	0.96	0.97	0.95	0.96
Dresses	0.79	0.83	0.81	0.86	0.78	0.82	0.89	0.83	0.86
Jeans	0.99	0.99	0.99	0.99	0.98	0.98	1.00	1.00	1.00
Sandals	0.95	0.98	0.97	0.98	0.98	0.98	0.98	0.98	0.98
Shorts	0.95	0.93	0.94	0.91	0.95	0.93	0.94	0.93	0.93
Skirts	0.85	0.79	0.82	0.86	0.81	0.84	0.86	0.87	0.87
Sneaker	0.99	0.95	0.97	0.97	0.97	0.97	1.00	0.97	0.98
Sweater	0.88	0.91	0.90	0.88	0.91	0.90	0.90	0.93	0.92
T-Shirt	0.87	0.95	0.91	0.87	0.96	0.91	0.89	0.96	0.92
Ø	**0.93**	**0.92**	**0.92**	**0.93**	**0.93**	**0.93**	**0.94**	**0.94**	**0.94**

The values show that the classes skirts and dresses perform particularly poorly compared to the other classes. By looking at the confusion matrix, it becomes clear that the category dress is often confused with skirts, sweaters, and t-shirts. Also, a relatively high risk of confusion between skirts and dresses seems to exist. This observation is consistent with that of Hara et al. [46]. They found that such confusion may arise due to too similar characteristics between classes or when classes, as in the case of "dress", unite attributes of other classes (e.g., the lower part of a dress similar to skirt, while the upper part similar to T-shirt). Figure 9 shows examples of misclassifications, most likely due to too high similarity. Within model C bags and jeans, on the other hand, were both reliably identified with a total recall of 1. An explanation might be that both categories show significant features and shapes that are clearly identifiable and almost always present in the data set (jeans: two legs, bags: handles and mostly square shape).

(1) (2) (3)

Figure 9: Examples of misclassifications for the category 'dress' due to too high similarity (Examples taken from model A. Predicted classes: (1) skirt, (2) sweater, (3) t-shirt).

Another error source seems to be more related to the semantic noise mentioned by Schindler et al. [102], which is caused when several classes are present on an image, but the classifier is only built for assigning one label at a time. From a practical point of view, misclassifications in this context are not an error but are rather to blame on the general modeling, since only a multi-class problem is applied here. Yet, a multi-label classifier would be required to assign several attributes to a specific product. Figure 10 shows examples of misclassifications in this context, which were probably only caused by the test design.

(1) (2) (3) (4)

Figure 10: Examples of misclassifications due to multiple representations ((1) + (2) skirt predicted, sandals true; (3) + (4) sweater predicted, skirt true).

Despite the growing number of training examples, some cases were continuously misclassified after each run. Figure 11 presents some examples of this. The misclassifications regarding the depicted images could be due to image-related problems such as high similarity of products [31], [46], textile-related material deformations [31], [46], or difficulties due to different perspectives and angles [31]. At this stage, only assumptions can be made about which problem led to the respective misclassification: in the image *(1)*, the reason could be the side view of the model, which suggests a box-like shape to the classifier. The shiny material and striking color could also imply an accessory rather than outdoor clothing. For image *(2)*, the small cords at the bottom of the sweater might resemble typical jacket cords, while for image *(3)*, the pattern in the middle of the image might be similar to buttons on a cardigan.

Regarding precision and recall in the models, it is apparent that sometimes an increase in recall is accompanied by a decrease in precision (and vice versa). This effect is especially evident when comparing models A and B. Between models B and C, this effect seems to be much weaker. This observation is in line with Skopal and Moravec [105]. They justified the shift between precision and recall by arguing that for the detection of more relevant objects of a class, the classifier must broaden its focus and thus tends to misclassify more often. To verify this effect, the F-score is applied. The results show that the F-score is continually growing, which points to a general improvement of the application. The fact that the average precision in model A slightly differs from the average recall and F-score is probably due to a rounding error.

(1) (2) (3)

Figure 11: Examples of repeated misclassifications ((1) bag predicted, coat true; (2) coat predicted, sweater true; (3) sweater predicted, t-shirt true).

For checking of how the results reflect the potential costs caused by FPs and FNs, the cost matrix is analyzed. Christley [23] refers in this context to FPs as type I errors and to FNs as type II errors. Since no real cost factors are identifiable for the application example, the whole process is illustrated using fictitious values. For the calculation, a cost factor of 5 is chosen for FPs and a lower cost factor of 3 for FNs. In this step, the cost factors are multiplied by the number of misclassifications per class. Table 7 shows the output of the resulting cost matrix.

Again, the total costs associated with misclassifications that occur decrease as the number of training data grows. However, it is noticeable that in model B, the costs for the specific classes jeans, shorts, and sneakers slightly increase. The corresponding precision and recall values show that their relationship has tended to shift in opposite directions compared to model A. The impact of the cost factors changes accordingly. However, the overall effect is relatively small. It is also compensated for by lower costs in the following training round C, even for the classes such as dresses and skirts, which were previously identified as particularly critical. The cost matrix also indicates that the classifier's performance improves over time as the number of training examples increases and that the associated misclassification risks in terms of possible costs for both type I error and type II error are steadily reduced.

Table 7: Cost matrix for type I error (FPc) and type II error (FNc) (the cost factors were randomly chosen: for FP=5 and for FN=3).

	A			B			C		
	FP_c	FN_c	Cost p. C.	FP_c	FN_c	Cost p. C.	FP_c	FN_c	Cost p. C.
Bags	5	0	5	5	0	5	5	0	5
Coats	5	27	32	15	15	30	15	15	30
Dresses	110	51	161	65	66	131	50	51	101
Jeans	5	3	8	5	6	11	0	0	0
Sandals	25	6	31	10	6	16	10	6	16
Shorts	25	21	46	45	15	60	30	21	51
Skirts	70	63	133	65	57	122	70	39	109
Sneaker	5	15	20	15	9	24	0	9	9
Sweater	60	27	87	60	27	87	50	21	71
T-Shirt	70	15	85	70	12	82	60	12	72
∑	380	228		355	213		290	174	
Total cost			**608**			**568**			**464**

4.5.2 Business-related Performance

The following section applies the business performance measures identified in chapter 3.3 for further evaluation. For correct calculation, exact parameters such as the time required to make entries within PDE or to carry out corrective actions as part of human revision are needed. As in the previous case, sample parameters were chosen for demonstration purposes.

It is assumed that 1,000 new products are added to the PIM after each classification run, which shall then be analyzed and enriched by the image classifier. The 1,000 test images represent this number. The corresponding amount of corrective actions required after each classification run equals the number of misclassifications previously conducted. If each product needs about 5 minutes for manual enrichment within the PDE, this means that for the enrichment of all 1,000 products, 5,000 minutes must be planned in total (case I). That equals around 83.3 hours. However, if the PDE is carried out automatically, these 5,000

minutes are omitted. Instead, the human revision process is added, for which 2 minutes per product are specified. Yet, that does not include any corrective measures that must be carried out in an extra step if necessary. Here, a time effort of 5 minutes is expected, which is equal to the time needed for manual PDE per product. Note that both the time for the inspection and the time for manual corrective action is considered in the case of a misclassified instance. If 2 minutes pro product are estimated for simple inspection of the results and 5 minutes, as in classic PDE, for corrective manual adjustments, then the total time required for model A equals: (1,000 * 2 min) + (152 * 5 min) = 2,760 min. This is equivalent to 39.7 hours, which is less than half the original time required. The time needed for models B and C is determined in the same way. The results are shown in Table 8.

Table 8: Time expenditure for manual PDE (case I) and automated PDE (case II) (as parameters, 2 min were assumed for control activities in the context of human revision and 5 min for classic PDE actions per product).

Model	**Estimated amount of time for PDE (in hours)**	
	Case I (manual PDE)	Case II (automate PDE)
A	83.3	46.0
B	83.3	45.2
C	83.3	43.0

The calculation shows that a significant reduction in time is achieved by using AI-supported PDE. However, the results depend on the initial parameters identified and selected and the time difference between simple control activities in human revision and actual corrective measures or manual PDE actions. Nevertheless, in the numerical example, a time improvement of 51.6 % has been achieved for the AI process.

The wage costs are calculated in proportion to the time spent. According to Abraham [1], no answer to the question of the responsibility for the PIM can be given since it differs from company to company. In many cases, however, it is the marketing department as they have the most significant benefit from PIM systems and, therefore, often bear the largest share of the costs. For the calculation, a salary for a marketing assistant with an

average of about 3000 € per month for a classic 40-hour week was selected. This value was derived from the Xing career platform [123]. To simplify matters, four business weeks per month, i.e., a total of 160 hours, were considered, which results in an hourly wage of 18.75 €.

The case I in Table 8 shows that around half of the monthly working time is spent on PDE. This amounts to a wage share of (83.3 h * 18.75 €) = 1,561.9 €, spent on manual PDE only. In return, the wage share in case II for automated PDE is (46 h * 18.75 €) = 862.5 € for model A, 847.5 € for model B, and 806.25 € for model C. From a business perspective, savings of (1,561.9 € – 862.5 €) = 699.4 € in terms of wage costs for one employee would have been achieved in model A. The results for all models are summarized in Table 9.

All in all, the amount of savings ultimately depends on the total number of enriched products, the classifier's prediction performance, and the time difference between classic manual enrichment activities and human revision tasks (the less time needed for revision and correction, the better). It should be noted, though, that the main cost-saving effect is generated by the fact that the image classifier is implemented and used at all, meaning that further cost savings only marginally improve by progressive training.

Table 9: Total savings of the PDE in proportion to the wage costs.

Model	**Pro-rata personnel costs (in €)**		**Total savings (in €)**
	Case I (manual PDE)	Case II (automate PDE)	
A	1,561.9	862.5	**699.4**
B	1,561.9	847.5	**714.4**
C	1,561.9	806.25	**755.65**

For identification of the overall risk of the application, the classifier's total costs are composed of actual and potential follow-up costs due to misclassifications. Here, classification risks are merged with business-related performance indicators into one key figure.

This ratio is considered as the total risk of the application within the PIM context. Table 10 presents the final results.

Since all determined cost indicators have decreased, logically, the combination of these ratios must also show a downward trend over time. The results show that even with artificially added costs, which reflect the potential risk of undetected misclassifications in monetary terms, the procedure appears to be more cost-efficient than the classic manual PDE process. Thus, at least from the perspective of the PDE-related effort involved, the hypothesis has been confirmed.

Table 10: Total risk assessment using classifier and business-related cost metrics (values rounded).

	Cost indicators for total risk assessment (in €)			**Cost comparison (in €)**	
Model	Classifier-related (I)	Business-related (II)	Combination of (I) and (II)	Costs for manual PDE	Cost advantage of automated PDE
A	608	863	1,471	1,562	91
B	568	848	1,416	1,562	146
C	464	807	1,271	1,562	291

5 Discussion

The results of the evaluation equally showed that with an increasing number of training examples, the overall performance of the image classifier was improved, and the costs related to necessary manual revision and correction tasks reduced. The experiment also revealed a cost deterioration of the classes jeans, shorts, and sneakers within model B. No further cost increase was observed in model C. Instead, the overall performance of the classes improved again. Therefore, this shift in costs is only considered to be of a temporary nature.

An explanation for the confusion in model B is that in connection with the new images, product features were introduced for which the classifier was not trained due to the lack of training data. That would be in line with Walczak [117], who emphasizes that the training data must follow the same distribution as the test data. By adding more training examples in model C, however, these features were sufficiently covered, improving the classification performance after repeated training.

Regarding the observed misclassifications, three different error types were identified:

a) misclassifications due to a high degree of similarity between the products and the appearance of similar features,
b) misclassifications due to factors, such as changes in perspective,
c) misclassifications caused by the modeling.

In connection with *a)*, one can presume that with an increasing number of training examples, the classification performance improved even for critical classes that were otherwise often confused. The results point to a continuous improvement of the predictions. Nevertheless, it should be further investigated which specific features the image classifier is focusing on. According to Castelvecchi [17], neural networks act as a kind of black box, where it is not clear how the relevant features for training are selected and processed. One way to address this problem would be to run the neural network in the reverse direction and create a visual representation of what the system considers to be a reflection of the learned class [79]. In the case of fashion classification, such a technique could verify whether the classifier has learned on product-typical attributes or whether it has focused on other features unrelated to the product.

For case *b)*, the network is expected to handle pose variations with a generally larger number of training examples as long as the training data sufficiently cover these. A more concrete solution to this problem is offered by Liu et al. [75], who use landmarks to define the locations and shapes of clothing items before the actual training. The structure of their neural network is divided into three strands, each of which considers global features, determines landmarks, and predicts local attributes depending on these landmarks. By analyzing the local attributes in dependence on factors such as the location and the shape of the objects (or garments), misclassifications caused by pose variations are mitigated.

The last case *c)* is the most critical aspect of the method within this paper. For the sake of simplicity, the classification process was tested in terms of a multi-class problem. However, as already mentioned in chapter 3.2, the multi-label classification would be the desired method for an effective real-world application of image classification in the PIM. Because of that, a distinction must be made between product type (multi-class) and attribute classification (multi-label) since both methods rely upon different modeling [67], [45].

In this context, product type classification often applies as a preliminary stage or prefilter, on which the subsequent attribute classification is based [45]. Multi-class methods such as product type classification tend to achieve higher accuracy values as it is easier to differentiate between distinct product categories than between product attributes whose spectra often overlap and thus may be present in multiple product classes [45]. In connection with the PIM application, however, there is no need for a product type classification, since the images are drawn directly from the PIM. Therefore, it is clear to the classifier to which product class the image corresponds. The advantage with this is that the direct contribution of the product class might then positively influence the attribute selection in the way described above.

Regarding multi-label problems, different approaches have developed in the literature. In some papers, multi-label problems were treated as extensions of multi-class problems that do not consider any dependencies or relationships between labels [34]. Donati et al. [29], for example, used different computer vision and machine learning methods, depending on the level of complexity of the considered features. These methods range from simple image processing methods for attributes such as colors to deep learning methods for shape and material analysis. The problem here is that, although the classification tasks considered separately deliver good results, they cannot be efficiently realized above a certain

number of features, since too many models would need coordination [34]. Recent work in the context of multi-class [102], [34], as well as multi-label problems [52], [34], addresses the issue of integrating dependencies and hierarchies of features into the classifier from the outset to avoid the necessity of constructing multiple classifiers.

Another problem of the application is that the classifier only selects the class with the highest probability, even if it is low and marginally higher than the probability of the second-best class. Such cases were particularly evident when confusion between similar classes like dresses, T-shirts, and skirts was too high during the classification process. A simple tool that could help avoid misclassifications in particularly difficult cases is the introduction of a threshold, similar to Ferreira et al. [34], where a probability limit is determined up to which the neural network is free to perform classifications on its own. If the probability of a class falls below this threshold, the process passes to a human decision-maker for manual selection. This way, the risk of misclassifications might be avoided or at least reduced. Yet, the number of manual interventions would increase for the most critical cases, and likewise, the corresponding costs.

While the confusion matrix provides information about the number of instances where human revision is required, the cost matrix additionally highlights the entrepreneurial risk that must be anticipated if human revision activities are faulty or completely neglected. Lower costs in this context mean a generally lower economic risk. It should be clear that the values calculated using the cost matrix are not real costs, but rather risk indicators. A major criticism that arises in this context is that fictitious values were the basis for the exemplary determination of the business-related costs. As a consequence, the results may not be as representative as desired. The entire calculation must, therefore, be carried out again based on real-world values to draw definitive conclusions.

In practice, however, determining the essential cost factors can prove difficult, as many variables affect the business. Yet, the automated process's final effectiveness depends on the time spent on classic manual PDE activities per product and how large its margin is compared to the time required for human revision tasks. These issues vary significantly between companies and industries, as they depend on, among other things, the technical means used, how experienced or trained the employees are, but also on how many attributes per product typically need to be entered into the PIM system.

Criticism is also expressed in connection with the modeling, as it does not yet include any technical costs. Again, costs may vary in each case, as they depend on whether the system is manufactured in-house and operated on their own hardware or in conjunction with a cloud-based service. How often and how quickly retraining should occur and how often maintenance work needs to be carried out is also relevant. According to Studer et al. [109], local systems are generally limited in size and performance and often require extensive updating and maintenance. Khan et al. [60], in turn, points out that significant problems of a deep learning-based application also include high computational costs and memory restrictions and that procedures in this area should not be implemented on capacity-limited systems since the application's performance could be impeded in this way.

In contrast, cloud solutions nowadays offer enormous computing power, with the main problem being to guarantee stable connections between interconnected services. According to Breck et al. [13], the model staleness, which describes how robust a model is to environmental changes over time (e.g., a shift in the data set), must also be considered. They suggest to not only continuously monitor the classification performance of the algorithm but also the computational power of the system, as this may provide information on existing problems or significant deviations. Yet, no clarity is reached on how the image classifier's technical implementation should be designed within the framework of PIM.

However, a growing number of academic studies seem to deal with questions regarding the optimal deployment of neural networks. The topics of these studies range from energy-efficient [16] and serverless deployment of neural networks [114] to the use of deployment tools such as Docker and AWS GreenGrass [63] and the implementation of modules via APIs [115], to name a few. Since a wide variety of PIM systems are on the market nowadays, future work should also examine PIM systems' general compatibility with different deployment technologies. These include, for example, Flask, Docker, GraphPipe, ONNX, or TensorFlow Serving [66].

Another point not addressed yet is where the training data should come from. According to Walczak [117], the choice of an adequately large training data set is of high relevance, especially since not only the gathering of the data set causes direct costs but also the performance of the network is influenced by the number of training data in terms of required computing time and capacity, which again results in indirect costs to consider. The results from the evaluation section are quite in line with Walczak's [117] statement. The analysis of the models' training duration demonstrated that model B needed less time in

total than model A, despite having twice as much training data. A better generalization performance and a faster achievement of low validation loss values by using more training examples might be the reason for this [8]. However, this effect did not repeat in model C. One might assume here that the training process exceeded some threshold and has become less effective. Meaning, that the higher the amount of training data from this point on, the less likely any additional data is to accelerate the convergence of the model.

The exact amount of required training data can only be estimated at this point. According to Walczak [117], a general guideline is to use a minimum of four times the amount of weighted connections within the network, on the condition that the database sufficiently represents all relevant classes. In this study, the neural network is implemented into an already existing PIM system. That means that PIM entries and their images are available from which the neural network can draw for training purposes. The advantage is that the training data directly reflect the specific business of the retailer. In such a case, the image classification procedure would come close to the standardization process mentioned in chapter 2.3, as future predictions would only build on attributes that are already in use, thus avoiding additional data inconsistencies caused by manual PDE activities.

On the other hand, this thesis assumed that the image classification procedure extracts the required images from an already existing PIM. However, this does not apply to cases where not enough training images are available within a company's PIM. If the image classification procedure is supposed to be used in such a context, an already prepared data set should be available for pretraining purposes. That, however, turns out to be a difficulty. Meaning, it would be challenging to create a universal data set that covers individual companies' product features and classes sufficiently in one. Again, product attributes are traded differently depending on the company. Therefore, labels from the training data do not necessarily have to match the labels usually used by the retailers.

However, this classifier's weakness may become a strength again if the retailer explicitly expresses the need for a standardized procedure. Like the classification systems eCl@ss, ETIM and UNSPSC mentioned in chapter 2.1, which are used in the context of PIM to standardize product categorization (i.e., the product family tree), it might be a solution to establish a similar framework for a uniform attribute classification system employed by an image classifier. According to this framework, training would then result in the classifier only generating outputs within a predetermined scope. As a benefit, the retailer

would no longer have to carry out any additional pretraining with its data, yet the procedure could be employed directly. This procedure would be useful, for example, in cases where a new online shop is launched with the initial filling of product attributes being left entirely to the classifier in the sense of automated PDE.

Finally, organizational consequences are to be considered as well regarding the application. In this context, the question arises to what extent the procedure affects the organization's structures and objectives. Fensel and Ding [33] describe the scope to which the problem of heterogeneous product and customer information complicates content management (in their case in business-to-business e-commerce). In this context, they also explain that the standardization of product descriptions is essential for customer communication and for customers to find the products they are looking for. Image classification might help to carry out or at least support this standardization process. Schindler et al. [102] confirm this. According to them, the fashion image classification they use serves the unification and enrichment of product data by standardizing product labels. Similarly, the image classification procedure in the PIM should also serve the standardization of product information and a higher quality of data.

It has already been mentioned earlier that the question of responsibility is not to be answered directly, as it depends on the structures and goals of an organization and the degree to which single departments or persons must deal with PDE issues. However, it is inevitable that to guarantee high data quality in connection with the procedure in the long term, the assignment of clear responsibilities and control routines must take place within the organization [47]. More precisely, setting clear responsibilities is a prerequisite for the introduction of standardized processes [87]. According to Abraham [1], PIM systems are often used by several departments of an organization simultaneously, so the need for a clear division of responsibilities is particularly high. He also proposed the integration of separate PIM teams into the organizational structure, whose members would deal exclusively with the maintenance of PIM systems. In this context, it would be appropriate to leave the responsibilities for human revision activities to this team, since competencies and knowledge about the individual products and their characteristics are primarily necessary. However, the question of responsibility for the technical design and maintenance of the classifier cannot be answered as quickly as this depends on the individual skills of the organizational members or on whether sufficient IT resources and expertise exist in the respective company.

The introduction of a new IT system is usually associated with organizational change. The extent to which innovations are accepted and used in the company can be roughly broken down into two levels, the individual and organizational level [37]. However, when it comes to the specific introduction of internal technologies, such as image classification in the PIM context, employee-focused measures such as training and coaching seem to be a good first choice for integrating the technology [101]. In summary, for a successful system introduction, comprehensive process management and change management are required, where the impact of the system on the company's activities and business processes, as well as any consequences for individual employees and departments, are assessed and taken into account [78].

If all the points mentioned in the discussion are now taken up and evaluated in their entirety, the underlying hypothesis that the use of AI-based image classification in the PIM generates time and cost-saving potentials can be answered – at least partially – with a "yes". Regarding the time expenditure for manual PDE activities, a saving potential of almost 50% has been demonstrated, based on the calculation example. The lower labor input also results in proportionately lower costs in terms of wages. However, the results have shown that the cost savings are primarily due to the use of the image classifier itself and that continuous retraining has resulted in steady improvements, yet they are significantly smaller in scope. The image classification also entails economic risks in terms of occurring misclassifications, which could be mistakenly published. But even if the associated risks are considered, the new process generally appears to be more beneficial in terms of cost savings than the traditional method.

Various companies are meanwhile researching image classification for cataloging purposes. Zalando, for example, uses image classification for image tagging to improve fashion item encoding and retrieval [12]. Donati et al. [29], in turn, examined to what extent the process may be applied to handle the large number of products that are produced in the company each year. The classifier they used is a combination of various classification methods that are partly trained only to categorize company-specific product characteristics such as logotypes. This thesis, however, deals with image classification in the specific context of PIM systems. It distinguishes from the just mentioned studies in a way that here the primary purpose is to get the product data ready for publication reasons and not just for internal use. Thus, the demand for the classification results is a different one. It

has already been pointed out how important the use of high-quality data is, i.e., complete and correct product information in the context of marketing activities.

On the other hand, the overall assessment of the procedure has not yet been completed, as additional cost factors, particularly in connection with technical measures, are still open. That includes questions on the choice of hardware and software, whether the system landscape is built locally or via a cloud provider, how compatible the application is concerning different PIM systems, and how the exact technical workflow is structured. After covering the technical issues, organizational questions on staff deployment and associated expenses should also be addressed. If the total costs for the creation, initialization, and maintenance of the classifier exceed the identified cost advantages by the time savings regarding the PDE, the application would not be any better than the traditional manual process. Under such circumstances, the image classifier's development and implementation would be without any additional economic value. Thus, it should be kept in mind that the use of the AI should not be for the sake of the new technology but should have a clear added value for the company and the processes within the company [77].

Overall, the process and its cost-saving potential seem very promising. Especially in connection with the PDE, a lot of working time can be saved by replacing the manual tasks. The time gained could then be invested elsewhere in the organization, for example, in marketing or customer service to improve the online presence or customer experience. Thus, the final recommendation is to examine the unanswered questions in future studies to use the image classification procedure within the PIM profitably in the future.

6 Conclusion

This thesis aims to investigate whether the use of AI-based image classification in PIM systems leads to any cost or efficiency advantages. The main objective of PIM systems is product data enrichment, which enhances product data for marketing purposes. In this connection, particular emphasis is placed upon the quality of the data as it is a crucial criterion for the success of online retailers. To meet the requirement of high information quality and to optimize the classifier's performance in the long run, the human revision process was identified as a central controlling instrument. Besides, supplementary performance indicators were determined to help to assess the performance of the classifier and its economic viability.

The image classification implementation showed significant savings potential, specifically in the reduction of manual PDE tasks. In this respect, the evaluation revealed a decrease in time expenditure and costs of almost 50%. Nevertheless, the final assessment also showed several limitations regarding the modeling, missing technical and organizational costs, and explanations regarding the technical implementation of the whole procedure. Future studies must address the question of the total cost assessment of the procedure as well as the technical feasibility in this context. They must also analyze in detail to what extent extensive retraining efforts are necessary and worthwhile. Overall, however, the hypothesis was confirmed since the implementation of the image classification procedure can lead both to significant cost-saving potentials and efficiency advantages due to released labor.

Since research on the topics of PIM systems and the management of AI-based misclassifications in the business context are both only sparsely covered in the literature, this thesis contributes to the research of both issues equally. Nevertheless, future projects must still be conducted in the context of the points mentioned earlier before a final evaluation can be made. The results can be summarized as very promising, though. Further research in this context may prove rewarding for the future application of PIM systems and is therefore highly recommended.

IV References

1. Abraham, J. (2014). *Product information management: Theory and practice* (Management for professionals). Cham: Springer International Publishing.
2. Achim, B. (2019). *Industrie 4.0 - jetzt mit KI*. Bitkom Research.
3. Aggarwal, C. C. (2015). *Data Mining*. Cham: Springer International Publishing.
4. Aggarwal, C. C. (2018). *Neural Networks and Deep Learning: A Textbook*. Cham: Springer International Publishing.
5. Alsheibani, S., Cheung, Y., & Messom, C. (2018). Artificial Intelligence Adoption: AI-readiness at Firm-Level. In M. Tanabu, & D. Senoo (Eds.), *Proceedings of PACIS2018: Pacific Asia Conference in Information Systems (PACIS)*.
6. Andreopoulos, A., & Tsotsos, J. K. (2013). 50 Years of Object Recognition: Directions Forward. *Computer Vision and Image Understanding*, 117(8), 827–891.
7. Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12), 2481–2495.
8. Banko, M., & Brill, E. (2001). Scaling to Very Very Large Corpora for Natural Language Disambiguation. *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, 26–33.
9. Battistello, L., Kristjansdottir, K., & Hvam, L. (2018). Scoping a PIM System: A Supporting Framework. *2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 1831–1835.
10. Beach, G. (August 20, 2019). How high-quality product data powers phenomenal on-site search. https://www.productsup.com/blog/high-quality-product-data-on-site-search/. Accessed: July 15, 2020.
11. Bechwati, N. N., & Siegal, W. S. (2005). The Impact of the Prechoice Process on Product Returns. *Journal of Marketing Research*, 42(3), 358–367.
12. Bracher, C. Improving Fashion Item Encoding and Retrieval. https://research.zalando.com/welcome/mission/research-projects/improving-fashion-item-encoding-and-retrieval/. Accessed: July 15, 2020.
13. Breck, E., Cai, S., Nielsen, E., Salib, M., & Sculley, D. (2016). What's your ML test score? A rubric for ML production systems. *30th Conference on Neural Information Processing Systems (NIPS 2016)*.

14. Brynjolfsson, E., Hu, Y. J., & Smith, M. D. (2006). From Niches to Riches: The Anatomy of the Long Tail. *Sloan Management Review,* 47(4), 67–71.
15. Bundesministeriums für Wirtschaft und Energie (BMWi). (2019). Perspektiven der künstlichen Intelligenz für den Einzelhandel in Deutschland. Bundesministeriums für Wirtschaft und Energie (BMWi), Berlin.
16. Cai, E., Juan, D.-C., Stamoulis, D., & Marculescu, D. (2017). NeuralPower: Predict and Deploy Energy-Efficient Convolutional Neural Networks. *arXiv preprint arXiv:1710.05420*.
17. Castelvecchi, D. (2016). Can we open the black box of AI? *Nature,* 538(7623), 20–23.
18. Castro, F. M., Marín-Jiménez, M. J., Guil, N., Schmid, C., & Alahari, K. (2018). End-to-End Incremental Learning. *Proceedings of the European conference on computer vision (ECCV),* 233–248.
19. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *CRISP-DM 1.0: Step-by-Step Data Mining Guide*. SPSS Inc.
20. Chawla, N. V., Japkowicz, N., Kołcz, A., Chawla, N. V., & Kotcz, A. (2004). Editorial: Special Issue on Learning from Imbalanced Data Sets. *ACM SIGKDD Explorations Newsletter,* 6(1), 1–6.
21. Chiu, C.-M., Wang, E. T. G., Fang, Y.-H., & Huang, H.-Y. (2014). Understanding Customers' Repeat Purchase Intentions in B2C E-Commerce: The Roles of Utilitarian Value, Hedonic Value and Perceived Risk. *Information Systems Journal,* 24(1), 85–114.
22. Chollet, F. (2018). *Deep Learning with Python*. Shelter Island, NY: Manning Publications.
23. Christley, R. M. (2010). Power and Error: Increased Risk of False Positive Results in Underpowered Studies. *The Open Epidemiology Journal,* 3(1), 16–19.
24. Cireşan, D., Meier, U., & Schmidhuber, J. (2012). Multi-Column Deep Neural Networks for Image Classification. *IEEE Conference on Computer Vision and Pattern Recognition,* 3642–3649.
25. Dalgic, T., & Leeuw, M. (1994). Niche Marketing Revisited: Concept, Applications and Some European Cases. *European Journal of Marketing,* 28(4), 39–55.
26. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

27. Döbel, I., Leis, M., Molina Vogelsang, M., Neustroev, D., Petzka, H., Riemer, A., et al. (2018). *Maschinelles Lernen. Eine Analyse zu Kompetenzen, Forschung und Anwendung*. Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V., München.
28. Dodge, S., & Karam, L. (2016). Understanding How Image Quality Affects Deep Neural Networks. *Eighth International Conference on Quality of Multimedia Experience (QoMEX)*, 1–6.
29. Donati, L., Iotti, E., Mordonini, G., & Prati, A. (2019). Fashion Product Classification through Deep Learning and Computer Vision. *Applied Sciences*, 9(7), 1385–1406.
30. Episerver. (2018). Reimagining Commerce: Erfolgreiche Strategien im Handel setzen den Fokus auf Erfolgreiche Strategien im Handel setzen den Fokus das Einkaufserlebnis statt auf Conversions. Episerver GmbH, Berlin.
31. Eshwar, S. G., Gautham Ganesh Prabhu, J., Rishikesh, A. V., Charan, N. A., & Umadevi, V. (2016). Apparel classification using Convolutional Neural Networks. *2016 International Conference on ICT in Business Industry & Government (ICTBIG)*, 1–5.
32. Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861–874.
33. Fensel, D., Ding, Y., Omelayenko, B., Schulten, E., Botquin, G., Brown, M., et al. (2001). Product Data Integration in B2B E-Commerce. *IEEE Intelligent Systems*, 16(4), 54–59.
34. Ferreira, B. Q., Baía, L., Faria, J., & Sousa, R. G. (2018). A Unified Model with Structured Output for Fashion Images Classification. *arXiv preprint arXiv:1806.09445*.
35. Fischer, L. (May 28, 2018). Product Information Management (PIM) mit Drupal. https://www.netnode.ch/blog/product-information-management-pim-mit-drupal. Accessed: July 19, 2020.
36. Forza, C., & Salvador, F. (2006). *Product Information Management for Mass Customization: Connecting Customer, Front-office and Back-office for Fast and Efficient Customization*. United Kingdom: Palgrave Macmillan.
37. Frambach, R. T., & Schillewaert, N. (2002). Organizational Innovation Adoption: A Multi-level Framework of Determinants and Opportunities for Future Research. *Journal of Business Research*, 55(2), 163–176.

38. French, R. M. (1999). Catastrophic Forgetting in Connectionist Networks. *Trends in Cognitive Sciences*, 3(4), 128–135.
39. Gentsch, P. (2019). *Künstliche Intelligenz für Sales, Marketing und Service: Mit AI und Bots zu einem Algorithmic Business – Konzepte und Best Practices* (2nd edn). Wiesbaden: Springer Fachmedien Wiesbaden.
40. Geron, A. [@aureliengeron]. (March 27, 2019). Sometimes validation loss < training loss. Ever wondered why? 1/5. [Tweet]. Twitter. https://twitter.com/aureliengeron/status/1110839223878184960. Accessed: July 19, 2020.
41. Girshick, R. (2015). Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1440–1448.
42. Goldenberg, J., Libai, B., Moldovan, S., & Muller, E. (2007). The NPV of Bad News. *International Journal of Research in Marketing*, 24(3), 186–200.
43. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. Cambridge, Massachusetts, London, England: MIT Press.
44. Guillon, N. (February 7, 2019). Akeneo stellt KI-basierte Product Data Intelligence vor - Akeneo - The Open Source PIM. https://www.akeneo.com/de/press-release/akeneo-stellt-ki-basierte-product-data-intelligence-vor/. Accessed: July 14, 2020.
45. Gutierrez, P., Sondag, P.-A., Butkovic, P., Lacy, M., Berges, J., Bertrand, F., et al. (2018). Deep learning for Automated Tagging of Fashion Images. *Proceedings of the European Conference on Computer Vision (ECCV)*.
46. Hara, K., Jagadeesh, V., & Piramuthu, R. (2016). Fashion Apparel Detection: The Role of Deep Convolutional Neural Network and Pose-dependent Priors. *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1–9.
47. Haug, A., & Stentoft Arlbjørn, J. (2011). Barriers to Master Data Quality. *Journal of Enterprise Information Management*, 24(3), 288–303.
48. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *2015 IEEE International Conference on Computer Vision (ICCV)*, 1026–1034.
49. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.
50. Heinemann, G. (2017). *Der neue Online-Handel: Geschäftsmodell und Kanalexzellenz im Digital Commerce* (8th edn). Wiesbaden: Springer Gabler.

51. Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors. *arXiv preprint arXiv:1207.0580v1*.
52. Hu, H., Zhou, G.-T., Deng, Z., Liao, Z., & Mori, G. (2016). Learning Structured Inference Neural Networks with Label Relations. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2960–2968.
53. Huang, J.-H., & Chen, Y.-F. (2006). Herding in online product choice. *Psychology and Marketing*, 23(5), 413–428.
54. Huhtala, M., Lohtander, M., & Varis, J. (2012). Confusing of Terms PDM and PLM: Examining Issues From the PDM Point of View. *The 22nd International Conference on Flexible Automation and Intelligent Manufacturing (FAIM 2012)*.
55. Huhtala, M., Lohtander, M., & Varis, J. (2013). The Role of Product Data Management (PDM) in Engineering Design and the Key Differences Between PDM and Product Lifecycle Management (PLM). *The 1st PDM forum for Finland-Russia collaboration, Lappeenranta 25th and 26th April 2013*.
56. Jiang, Y., & Cukic, B. (2009). Misclassification Cost-Sensitive Fault Prediction Models. *Proceedings of the 5th International Conference on Predictor Models in Software Engineering*, 1–10.
57. Kang, F., Jin, R., & Sukthankar, R. (2006). Correlated Label Propagation with Application to Multi-label Learning. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2, 1719–1726.
58. Karpathy, A. (April 30, 2016). char-rnn. https://github.com/karpathy/char-rnn#tips-and-tricks. Accessed: July 16, 2020.
59. Kemker, R., & Kanan, C. (2018). Fearnet: Brain-inspired Model for Incremental Learning. *International Conference on Learning Representations (ICLR)*.
60. Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). A Survey of the Recent Architectures of Deep Convolutional Neural Networks. *Artificial Intelligence Review*, 1–62.
61. Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *arXiv:1412.6980*.
62. Koço, S., & Capponi, C. (2013). On Multi-class Classification Through the Minimization of the Confusion Matrix Norm. *Asian Conference on Machine Learning*, 277–292.

63. Krishnamurthi, R., Maheshwari, R., & Gulati, R. (2019). Deploying Deep Learning Models via IOT Deployment Tools. *2019 Twelfth International Conference on Contemporary Computing (IC3)*, 1–6.
64. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems, (*2), 1097–1105.
65. Kropsu-Vehkapera, H., Haapasalo, H., Harkonen, J., & Silvola, R. (2009). Product Data Management Practices in High-tech Companies. *Industrial Management & Data Systems*, 109(6), 758–774.
66. Kurovski, M. (October 18, 2018). From Exploration to Production — Bridging the Deployment Gap for Deep Learning (Part 2). https://towardsdatascience.com/from-exploration-to-production-bridging-the-deployment-gap-for-deep-learning-part-2-9e33cc8dfe5e. Accessed: July 17, 2020.
67. Lao, B., & Jagadeesh, K. Convolutional Neural Networks for Fashion Classification and Object Detection. *CCCV 2015: Computer Vision*, 2015, 120–129.
68. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, 521(7553), 436–444.
69. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., et al. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4), 541–551.
70. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.
71. Lee, T., Lee, I.-h., Lee, S., Lee, S.-g., Kim, D., Chun, J., et al. (2006). Building an Operational Product Ontology System. *Electronic Commerce Research and Applications*, 5(1), 16–28.
72. Liu, B. (2017). Lifelong Machine Learning: A Paradigm for Continuous Learning. *Frontiers of Computer Science*, 11(3), 359–361.
73. Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., et al. (2020). Deep Learning for Generic Object Detection: A Survey. *International Journal of Computer Vision*, 128(2), 261–318.
74. Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A Survey of Deep Neural Network Architectures and their Applications. *Neurocomputing*, 234, 11–26.

75. Liu, Z., Luo, P., Qiu, S., Wang, X., & Tang, X. (2016). DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1096–1104.
76. Lombardi, S., Gorgoglione, M., & Panniello, U. (2013). The Effect of Context on Misclassification Costs in E-Commerce Applications. *Expert Systems with Applications*, 40(13), 5219–5227.
77. Loshin, D. (2013). *Business Intelligence: The Savvy Manager's Guide* (2nd edn). Waltham, MA: Morgan Kaufmann.
78. Machts, T., & Grosch, T. (2019). PIM- und MDM-Software auswählen: Produktinformations- und Masterdaten-Management-Systeme im Vergleich. dotSource GmbH, Jena.
79. Mahendran, A., & Vedaldi, A. (2016). Visualizing Deep Convolutional Neural Networks Using Natural Pre-images. *International Journal of Computer Vision*, 120(3), 233–255.
80. Maltoni, D., & Lomonaco, V. (2019). Continuous Learning in Single-incremental-task Scenarios. *Neural Networks*, 116, 56–73.
81. McCloskey, M., & Cohen, N. J. (1989). Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. *The Psychology of Learning and Motivation*, 24, 109–165.
82. McKinney, W., & Pandas Development Team (June 17, 2020). pandas: powerful Python data analysis toolkit: Release 1.0.5. https://pandas.pydata.org/docs/pandas.pdf. Accessed: July 17, 2020.
83. Microsoft (2020). What is the Team Data Science Process? https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/overview. Accessed: July 14, 2020.
84. Minnema, A., Bijmolt, T. H.A., Gensler, S., & Wiesel, T. (2016). To Keep or Not to Keep: Effects of Online Customer Reviews on Product Returns. *Journal of Retailing*, 92(3), 253–267.
85. Nair, V., & Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 807–814.
86. Nguyen, C. V., Li, Y., Bui, T. D., & Turner, R. E. (2018). Variational Continual Learning. *International Conference on Learning Representations (ICLR)*, 2018.

87. Otto, B. (2012). Managing the Business Benefits of Product Data Management: The Case of Festo. *Journal of Enterprise Information Management,* 25(3), 272–297.
88. Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wermter, S. (2019). Continual Lifelong Learning with Neural Networks: A Review. *Neural Networks,* 113, 54–71.
89. Park, C.-H., & Kim, Y.-G. (2006). The Effect of Information Satisfaction and Relational Benefit on Consumers' Online Shopping Site Commitments. *Journal of Electronic Commerce in Organizations (JECO),* 4(1), 70–90.
90. Parsionate GmbH. PIM-Systeme im Überblick. https://pim-auswahl.de/pim-systeme/#pim-systeme. Accessed: July 14, 2020.
91. Pohl, M. (September 04, 2018). 5 Use-Cases - Wie künstliche Intelligenz bereits heute in MDM- und PIM-Projekten unterstützt. https://parsionate.com/de/magazin/5-use-cases-wie-kuenstliche-intelligenz-bereits-heute-in-mdm-und-pim-projekten-unterstuetzt/. Accessed: July 14, 2020.
92. Ratcliff, R. (1990). Connectionist Models of Recognition Memory: Constraints Imposed by Learning and Forgetting Functions. *Psychological Review,* 97(2), 285–308.
93. Remy, N., Speelman, E., & Swartz, S. (October 20, 2016). Style that's sustainable: A new fast-fashion formula. https://www.mckinsey.com/business-functions/sustainability/our-insights/style-thats-sustainable-a-new-fast-fashion-formula. Accessed: July 14, 2020.
94. Reyniers, D. J. (1989). Interactive High-Low Search: The Case of Lost Sales. *The Journal of the Operational Research Society,* 40(8), 769–780.
95. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *Medical Image Computing and Computer-assisted Intervention - MICCAI 2015* (pp. 234–241, Lecture Notes in Computer Science Image Processing, Computer Vision, Pattern Recognition, and Graphics, Vol. 9351). Cham: Springer.
96. Rosebrock, A. (October 14, 2019). Why is my validation loss lower than my training loss? | PyImageSearch. https://www.pyimagesearch.com/2019/10/14/why-is-my-validation-loss-lower-than-my-training-loss/. Accessed: July 16, 2020.

97. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning Representations by Back-propagating Errors. *Nature,* 323, 533–536.
98. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision,* 115(3), 211–252.
99. Russell, S. J., & Norvig, P. (2016). *Artificial intelligence: A modern approach* (Always learning). Edinburgh Gate, Harlow, Essex, England: Pearson Education Limited.
100. Rust, R. T., Inman, J. J., Jia, J., & Zahorik, A. (1999). What You Don't Know About Customer-Perceived Quality: The Role of Customer Expectation Distributions. *Marketing Science,* 18(1), 77–92.
101. Schillewaert, N., Ahearne, M. J., Frambach, R. T., & Moenaert, R. K. (2005). The Adoption of Information Technology in the Sales Force. *Industrial Marketing Management,* 34(4), 323–336.
102. Schindler, A., Lidy, T., Karner, S., & Hecker, M. (2018). Fashion and Apparel Classification using Convolutional Neural Networks. *arXiv preprint arXiv:1811.04374*.
103. Shani, D., & Chalasani, S. (1992). Exploiting Niches Using Relationship Marketing. *Journal of Consumer Marketing,* 9(3), 33–42.
104. Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*.
105. Skopal, T., & Moravec, P. (2005). Modified LSI Model for Efficient Search by Metric Access Methods. In D. E. Losada & J. M. Fernández-Luna (Eds.), *Advances in Information Retrieval*: *27th European Conference on IR Research, ECIR 2005, Santiago de Compostela, Spain, March 21 - 23, 2005* (pp. 245–259, Lecture Notes in Computer Science, Vol. 3408). Berlin: Springer.
106. Smith, K. (August 16, 2017). Five Reasons Behind ASOS's Incredible Retail Success. https://risnews.com/five-reasons-behind-asoss-incredible-retail-success. Accessed: July 14, 2020.
107. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research,* 15(1), 1929–1958.
108. Statista Digital Market Outlook (2020). eCommerce Report 2020. Statista GmbH, Hamburg.

109. Studer, S., Bui, T. B., Drescher, C., Hanuschkin, A., Winkler, L., Peters, S., et al. (2020). Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology. *arXiv preprint arXiv:2003.05155*.
110. Swerdlow, F., & Angel, N. (2016). *The FORRESTER Wave™: Product Information Management Solutions, Q4 2016: The 10 Providers That Matter Most And How They Stack Up*. Forrester Research, Cambridge, US.
111. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). Going Deeper with Convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–9.
112. Szegedy, C., Toshev, A., & Erhan, D. (2013). Deep Neural Networks for Object Detection. *Advances in Neural Information Processing Systems*, 2553–2561.
113. Trotter, C. (July 22, 2019). Why Akeneo believes AI is the future of PIM solutions - Insider Trends. https://www.insider-trends.com/why-akeneo-believes-ai-is-the-future-of-pim-solutions/. Accessed: 14.07.2020.
114. Tu, Z., Li, M., & Lin, J. (2018). Pay-Per-Request Deployment of Neural Network Models Using Serverless Architectures. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, 6–10.
115. Verenich, E., Velasquez, A., Murshed, M. G., & Hussain, F. (2020). FlexServe: Deployment of PyTorch Models as Flexible REST Endpoints. *arXiv preprint arXiv:2003.01538*.
116. W&V Redaktion (November 19, 2019). Automatisierte Texterstellung für Onlineshops: So einfach geht's. https://www.wuv.de/tech/automatisierte_texterstellung_fuer_onlineshops_so_einfach_geht_s. Accessed: July 12, 2020.
117. Walczak, S. (2019). Artificial Neural Networks. In M. Khosrow-Pour (Ed.), *Advanced Methodologies and Technologies in Artificial Intelligence, Computer Simulation, and Human-computer Interaction* (pp. 40–53). Hershey, PA: IGI Global.
118. Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., & Xu, W. (2016). CNN-RNN: A Unified Framework for Multi-Label Image Classification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2285–2294.
119. Wei, Y., Xia, W., Lin, M., Huang, J., Ni, B., Dong, J., et al. (2016). HCP: A Flexible CNN Framework for Multi-label Image Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(9), 1901–1907.

120. Weiss, G. M., & Provost, F. (2003). Learning When Training Data are Costly: The Effect of Class Distribution on Tree Induction. *Journal of Artificial Intelligence Research*, 19, 315–354.
121. Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques* (Morgan Kaufmann Series in Data Management systems). Amsterdam: Elsevier/Morgan Kaufmann.
122. Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv preprint arXiv:1708.07747*.
123. Xing. Dein Gehalt als Marketing Assistent. https://www.xing.com/salary/jobs/Marketing+Assistent/?country=DE. Accessed: July 18, 2020.
124. Yochze. (July 11, 2019). How to apply continual learning to your machine learning models. https://towardsdatascience.com/how-to-apply-continual-learning-to-your-machine-learning-models-4754adcd7f7f. Accessed: July 17, 2020.
125. Zhang, S., Zhang, C., & Yang, Q. (2003). Data preparation for data mining. *Applied Artificial Intelligence*, 17, 375–381.
126. Zhou, Y. T., & Chellappa, R. (1988). Computation of optical flow using a neural network. *IEEE 1988 International Conference on Neural Networks*, 2, 71-78.
127. Zinchenko, A. (August 02, 2018). Produktinformationsmanagement mit System. https://treopim.com/de/blog/produktinformationsmanagement-mit-system. Accessed: July 17, 2020.

V Appendices

Table 11: Features and processes of PIM systems according to Abraham [1], p. 4 ff.

Process	Goals	Core Features
Collection	The collection of product information, which is created and stored across the organization and beyond	• Data import from multiple sources (e.g. ERP, procurement, product suppliers, data suppliers, media agencies) in various structured formats (e.g. Text, Excel, CSV, XML, BMECat, iDoc) • Mapping data to product attributes • Transformation of data if necessary (e.g. changing weight units)
Consolidation	Preventing multiple entries, ensuring that each product is represented by only one instance	• Cleaning, merging and consolidating product information • Often only partially supported by automated operations (e.g. if all organizational units use the same product identifiers)
Enrichment	Improving product data quality for customer-oriented purposes by adding additional information to the products	• Management of attributes supported by classification system • Creation of relationships between products (e.g. for cross-selling) • Adding media assets (e.g. photos, videos, manuals, CAD/CAM drawings) • Automated data validation, verification and version control • Data access control for security • Workflow control for efficient task allocation between users
Distribution	Delivering data once or in regular intervals for different publication channels	• Export data to numerous channels (e.g. print, website, mobile apps, social media, marketplaces, email) in multiple formats (e.g. DOC, PDF, XML, HTML, CSV, Excel, FTP, web service)

A B C

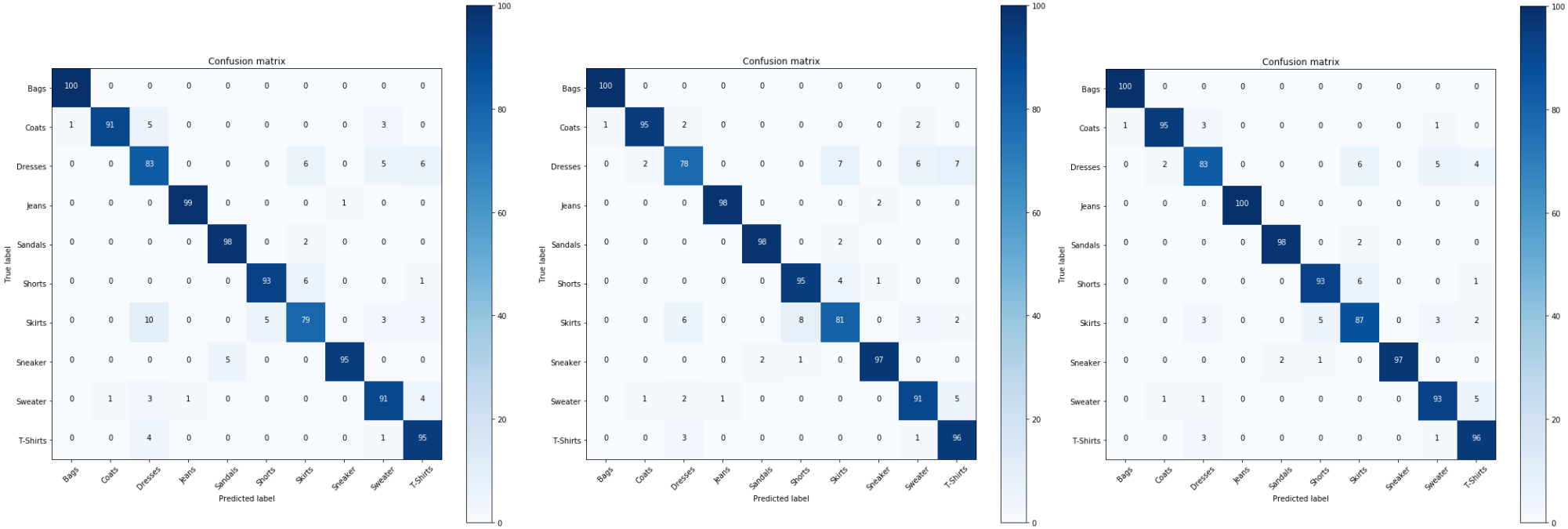

Figure 12: Confusion matrix for run A (2,500), B (5,000), and C (10,000 training images).

Statement

I ensure: I wrote the master thesis myself without the use of any means or sources other than indicated. The paper has not yet been submitted to any other examination committee and has not been published. I am aware that any false statements can have legal consequences.

Confidentiality Clause

This master thesis contains confidential data of the dotSource GmbH. This work may only be made available to the first and second reviewers and authorized members of the board of examiners. Any publication and duplication of this master thesis is prohibited. An inspection of this work by third parties requires the expressed permission of the author and the dotSource GmbH.