How LIME Explanation Models Can Be Used to Extend Business Process Models by Relevant Process Details

Myriel Fichtner[1], Stefan Schönig[2] and Stefan Jablonski[1]
[1]*University of Bayreuth, Germany*
[2]*University of Regensburg, Germany*

Keywords: Image Mining, Business Process Model Enhancement, Business Process Improvement, Relevant Process Detail, Convolutional Neural Network, LIME Explanation Models.

Abstract: Business process modeling is an established method to describe workflows in enterprises. The resulting models contain tasks that are executed by process participants. If the descriptions of such tasks are too abstract or do not contain all relevant details of a business process, deviating process executions may be observed. This leads to reduced process success regarding different criteria, e.g., product quality. Existing improvement approaches are not able to identify missing details in process models that have an impact on the overall process success. In this work, we present an approach to extract relevant process details from image data. Deep learning techniques are used to predict the success of process executions. We use LIME explanation models to extract relevant features and values that are related to positive process predictions. We show how a general conclusion of these explanations can be derived by applying further image mining techniques. We extensively evaluate our approach by experiments and demonstrate the extension of an existing process model by identified details.

1 INTRODUCTION

Business process modeling languages are used to describe business processes. Business process models provide insights into the structure of processes and potential for process improvement. Usually, process models are designed by process experts and thus contain all information that have been identified as important by experience. However, the results of process model executions might not be optimal. It may even be observed that the execution of the same process model results in different outcomes that differ with respect to various criteria, e.g., production time, quality or cost of the process output. Such observations lead to the assumption that the process is not modeled in sufficient detail or quality. In this research we explicitly focus on missing details of process models as cause for non-optimal executions. We identify three reasons why a process model is lacking details.

(i) Process modelers are often not sure how detailed a process has to be modeled in order to guarantee optimal executions. Although, there are a lot of modeling tools available, none of them provides modeling guidelines in this sense (Kluza et al., 2013). There are styling guidelines and modeling advice in order to keep aspects like clarity (e.g., (Becker et al., 2000), (Mendling et al., 2010)), but there is no instruction how detailed a process should be designed. Even process experts are not aware of all details or which details are relevant enough to model (Niedermann et al., 2010). Input constellations or prerequisites are taken as granted and are not considered in process descriptions. Due to lack of awareness, important details are missing in a process model.

(ii) Modelers know which details are relevant for process success, but they cannot integrate them appropriately into a model. Consider a task that allows multiple input constellations, but not all of them lead to successful process executions. This information is hard to represent in a process model. In such cases, the restricted expressiveness of established modeling languages leads to the lack of modeling details. To extend the usability and expressiveness of existing modeling languages, different approaches have been developed. For example, media data can be attached to modeling elements (Wiedmann, 2017). Nevertheless, the relevant information is missing in the model as long as no suitable representation method is available.

(iii) Some information can be intentionally omitted in the process model. Complex processes lead to complex models which contain hundreds of modeling elements. Large process models, in turn, lead

Fichtner, M., Schönig, S. and Jablonski, S.
How LIME Explanation Models Can Be Used to Extend Business Process Models by Relevant Process Details.
DOI: 10.5220/0011067600003179
In *Proceedings of the 24th International Conference on Enterprise Information Systems (ICEIS 2022) - Volume 2*, pages 527-534
ISBN: 978-989-758-569-2; ISSN: 2184-4992

to overload. In the worst case, process participants are unable to read and execute the tasks appropriately. Therefore, one goal of modeling is to avoid too complex models. This requirement is hard to meet, while process designers try to reduce the size of models by *(a)* omitting details during modeling that are known but do not occur often enough or *(b)* merging alternative cases due to their presumed irrelevance or rarity. By using abstraction mechanisms (e.g., (Polyvyanyy et al., 2008), (Reichert et al., 2012)), modeling elements are aggregated leading to coarser views that support comprehensibility. Then, process information is omitted for the benefit of abstraction. The loss of information may include relevant details that influence process success.

These reasons show that non-modeled but relevant process details have to be identified. It is not enough to determine missing aspects, but also to identify their characteristics that optimize process output. Existing process improvement approaches do not cover these requirements, since they are based on pre-known aspects. Most of them focus on how to optimally restructure already defined modeling elements (e.g., adjustment of execution paths). However, it is indeed possible that important process details are not taken into account during the modeling stage and thus cannot be analyzed by such techniques. Examples can be found in manufacturing environments, where many tasks are executed manually by process participants. In industrial projects we face placement scenarios where process models just prescribe that parts should be placed on a pallet. Further instructions on how to place the parts are missing. Observing process executions reveal that the process outcomes are deviating. Our research goal is motivated by this experience.

Existing process improvement approaches show that additional data sources and analysis concepts have to be taken into account. In manufacturing environments, production lines are usually monitored by collecting image data. According to (Schmidt et al., 2016), images are a powerful data source, which contain complex information and interrelations, where business processes can benefit from. Inspired by this, we aim for the deployment of image mining techniques. In this paper we focus on the implementation of an approach that extracts relevant process details from images to enhance process models and guarantee process success. We propose to collect image data of single task executions and to analyze them using LIME explanation models (Ribeiro et al., 2016). Based on these results, insufficient modeled details are identified. Examples on how these details may be attached to a process model are given. Our work complements previous process improvement approaches. It supports process designers even after the modeling phase in their efforts of designing a process success-oriented model.

2 RELATED WORK

Business processes describe a series of steps that must be performed to achieve a business goal, such as manufacturing a specific product. The modeling language *Business Process Model and Notation* (BPMN) is considered as standard in this field (Chinosi and Trombetta, 2012). It enables a graphical notation of business processes by mapping the procedure itself and involved process entities (e.g., documents) to modeling elements. Once modeled, processes are executed according to the model. A widely used approach to improve process models and executions is Process Mining. Process mining techniques target the automatic discovery of information from event logs which contain one or several process cases (Van der Aalst et al., 2009). Such techniques are able to identify, for example, that the execution of process steps in another order maximizes process success. This insight is then considered for future executions through redesigning the process model. Since event logs represent model executions, they contain exclusively information that was previously modeled.

We distinguish between two types of approaches. We classify techniques that analyze processes by known information that is already contained in the process model as approaches that work with *intrinsic* parameters. In contrast, approaches that work with *extrinsic* parameters incorporate further data sources or information not yet included in a model.

There are a lot of different approaches that optimize processes using merely intrinsic parameters, e.g., (Gounaris, 2016), (Polyvyanyy et al., 2008), (Reichert et al., 2012) (Schonenberg et al., 2008). Also approaches that consider multiple process perspectives to improve process analyzability have to be mentioned in the context of intrinsic techniques, e.g., (Front et al., 2017). According to (Radeschütz et al., 2008), most business analysis tools do not consider all relevant data sources or are restricted to a single data source. This confirms our observation, that there are only a few approaches based on extrinsic parameters. In (Niedermann et al., 2010) a (semi-) automated process optimization approach is suggested, which integrates process and operational data, as well as any other required data source, e.g., process participant specific data. Furthermore, we classify approaches that add information that cannot be expressed by the process modeling language as extrinsic. The authors of (Wiedmann, 2017) propose the BPMN extension BPM(N)^{Easy} to attach media annotations to modeling elements. This provides users additional data sources that contain information exceeding the expressive power of a BPMN model.

3 OVERALL CONCEPT

In previous work, we proposed an extrinsic parameter-based concept to enhance existing business process models by analyzing image data of process executions (Fichtner et al., 2020), (Fichtner et al., 2021). The overall concept consists of five steps which are briefly described in this section while the remainder of this paper is dedicated to the implementation of the Image Analysis step.

It is assumed that there exists a business process model for a given process. In the **Preparation** step, process tasks are identified in the model that have be executed manually by process participants. Each of these tasks will be analyzed successively for relevant process details in the overall procedure. In the **Definition** step, the process model is modified regarding the considered task, such that a video of the task execution is recorded or a picture of the initial situation of the task is taken. The process model is executed and image data of the task is recorded. After each execution, the data is labelled according to process success (**Execution and Labelling**). The image data is analyzed by using image mining techniques in the **Image Analysis** step. The goal is to identify the regions in all image data that are related to process success. The output of this step is the relevant process detail that contains process success related information and is missing in the process model. To consider the analyzed process detail in future process executions, the existing process model is modified. The modification can either be structure-related, text-related or the detail can be integrated as media annotation. In the **Validation** step, it is evaluated whether the preceding process model modification increases process success or if another task might be responsible for the reduced process success and has to be considered.

The objective of the approach is to ensure overall process success by *(i)* identifying tasks in a process model that were insufficiently modeled and *(ii)* extending the model by missing but relevant process details. In contrast to classical quality assurance approaches, e.g., (Prykäri et al., 2010), that evaluate the direct correlation between a task execution and its output, the presented concept considers the overall process success. This allows to identify whether process failures originate from single tasks, even though the execution itself seems to be correct. To realize the Image Analysis step, we identified the following requirements regarding its output:

First, the analyzed process details should not only contain information extracted from a single positive example. Although this may be sufficient for improvement, the action scope of process participants will be severely limited. This can be illustrated by an example: Consider a task in a process model with the instruction to place a single object on a pallet. Since the object position is not explicitly stated, the process participant place the object anywhere on the pallet. We assume that there are images showing this scene and that they are labelled according to overall process success. We further assume that task success depends on the position of the object on the pallet, i.e., the task is successful if the object is placed in a certain region. One single positive example contains the information that placing an object at a certain position leads to success. This is a valuable result, however, restricts the process participants too much and is impractical in real process environments. In addition, the proposed improvement strongly depends on the selected example. Single examples always include the risk of being non-representative. Approaches that are able to consider the full solution space, i.e., the valid region to place the object, are needed. The idea of our approach is to take all positive evaluations and to generalize them. Thus, out of a concrete placement information from one positive scenario, a region for placing the object is determined by collecting all positive examples. Placing the object anywhere in this region improves the process outcome. We formulate this procedure as finding a general conclusion.

Second, the analyzed information must be interpretable for process participants. In the example above, an analysis of all object positions in positive labelled images results in a list of coordinates. An instruction that contains this list is hard to interpret and can only hardly be followed. In contrast, a visualization where the valid placement region is highlighted is easier to understand. Therefore the analysis results have to be transformed in an adequate representation.

4 EXTRACTING RELEVANT PROCESS DETAILS

In this section we present an implementation for the Image Analysis step from our overall concept. We focus on tasks where input specifications are not modeled prescriptively enough. We identify features that are relevant for the success of process executions. We aim at finding regions of successful process executions in the feature space and at extracting criteria that separate them from unsuccessful ones. From these criteria we derive relevant process details which are integrated into a business process model. Separating a feature space in multiple classes is the well researched issue of classification. If the feature space is known, classical machine learning approaches can be used to determine the class boundaries. If the features have to be extracted automatically, deep learning mechanisms are used (Popescu and Lucian, 2014). In the context of images, convolutional neural networks (CNN) have proven to be a successful technique. CNNs are used for prediction, while in most cases the focus is on the final result of a prediction. However, knowing the reason for a prediction and making CNNs explainable and interpretable is important. Although the parameters that are connected to the decision of a CNN are difficult to interpret, there are some explanation techniques summarized in (Burkart and Huber, 2021). By using such an approach, we are able to *(i)* identify which features in images are related to successful process outcomes and *(ii)* which values of these features are required to guarantee process success.

In our implementation approach we use local interpretable model-agnostic explanations (LIME) provided by (Ribeiro et al., 2016). The concept of LIME enables the explanation of predictions of any classifier in an interpretable and faithful manner (Ribeiro et al., 2016). An interpretable model locally around the prediction is learned and identified. The authors of LIME propose a method to explain models through representative individual predictions. The development of LIME was motivated by problem statements related to trust in the context of system decisions. It is important to understand the reasons of a decision and to recognize wrong ones in order to avoid mistakes (Dzindolet et al., 2003). LIME complements existing systems allowing users to assess trust even when a prediction seems to be correct but is made for the wrong reasons. Non-experts are enabled to identify irregularities when explanations are present. These aspects inspired us to use LIME in the context of Business Process Management. Knowing the reasons behind a prediction can improve business processes fundamentally instead of providing temporary and case-dependent suggestions for improvement. To provide an interpretable representation, LIME uses binary vectors indicating the presence or absence of a contiguous patch of similar pixels. The explanations are visualized by highlighting decision-relevant parts in the original images. The authors of LIME published promising experiments and the source code of their research [1] what supports the use of LIME. Since LIME only provides local explanations, we analyze the resulting images to derive a global explanation. Giving a global understanding of image explanations is an open research problem (Ribeiro et al., 2016). We present an idea to tackle this issue for our experiments.

Furthermore, we address the question of *(iii)* how the process model can be extended by analyzed information. We give an example by modifying a process model designed with BPMN.io [2]. This is an established toolkit to view and model BPMN 2.0 diagrams. Modeled diagrams can easily be imported and exported via XML files. To enrich an existing model with details, we modify this file.

4.1 Implementation

Our implementation consists of three parts: First, a CNN is trained with labelled image data and LIME is used to explain the classification model. The outputs of the explanation step are images highlighting pros for the prediction. Second, these images are analyzed regarding different features and a general conclusion is derived. Third, analysis results are integrated into a business process model.

To realize the first part, we follow an open source implementation presenting the usage of LIME [3]. We adopt the basic architecture of the CNN which is sufficient for the complexity of our experimental images. We adapt the parameters for the expected image size to our data and reduce the value of the batch size and epochs for system compatibility reasons.

In the second part, we further analyze the results after using LIME on positive labelled images to derive a general conclusion. The LIME results are copies of the original images but contain only those regions that explain the decision to the positive class. Irrelevant parts are colored black. The experiments in the next section show that the remaining image content is in most cases an object that is involved in the task. We define this object as contiguous set of pixels that have similar colors but that do not have the background color. A global explanation can be computed based on local explanations by finding a possibility to compare super-pixels in different images (Ribeiro et al., 2016). In our experiments, we address this issue by analyzing the object visible in each LIME result with respect to a set of features (color, shape, size and position (centroid)). We define this step as finding a

[1] https://github.com/marcotcr/lime-experiments (last accessed 16 Dec 2021)

[2] https://bpmn.io/ (last accessed 15 Dec 2021)

[3] https://github.com/marcellusruben/All_things_medium/blob/main/Lime/LIME_image_class.ipynb (last accessed 16 Dec 2021)

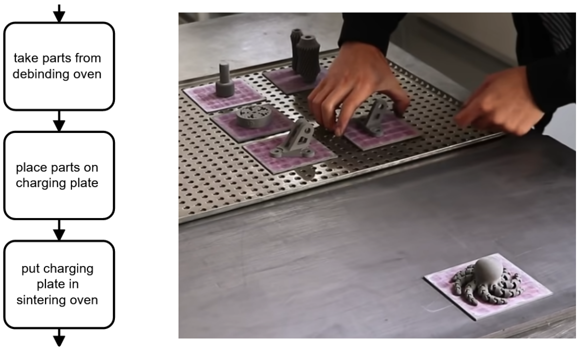


Figure 1: The metal injection molding or extrusion process.

general conclusion. For this purpose, the analysis results are summarized in feature specific sets. There is one set for each feature containing all possible values, e.g., a color set contains all object colors that were analyzed across all LIME results. All features and their corresponding values are relevant process details which have to be considered during task execution to enhance process success.

In the third part, the analyzed process details are integrated in an existing process model. As stated in Section 3, textual modification, structural modification or extension by media annotations are possible ways to enhance a task instruction. For textual modifications, the model is exported as XML file and the value of the appropriate task attribute is extended by the process detail. For the structural modification, additional tags have to be created. In case of media annotations, an image that represents the process detail is created and attached to the task.

4.2 Experiments and Results

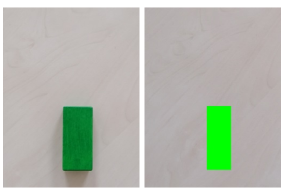
For our experiments, we design a simple process model representing a part of the metal injection molding or extrusion process. These processes are similar, with the latter replacing molding by extrusion. The left image in Figure 1 shows this process designed with BPMN.io. The picture on the right illustrates the task of manually placing parts on a charging plate in a real process environment [4].

To evaluate the applicability of LIME in the context of Business Process Management, we recorded image data showing this placement task. For a first proof of concept, we restrict to simple object shapes instead of taking parts from industrial environments. We prepare our images by using object recognition techniques. This excludes disturbing factors regarding image quality, such as noise or uncontrollable lighting conditions. We recorded 1000 images per experiment to ensure a sufficiently large database to compute a representative result. The classification problem is limited to two classes, resulting in 500 negative examples (*unsuccessful process executions*) and 500 positive examples (*successful process executions*). Each image shows an input constellation of the task. We know the criteria that distinguishes positive and negative labelled image data. In real applications these criteria are not known but should be analyzed in the Image Analysis step. The analyzed criteria correspond to our definition of relevant process details. However, we exploit the knowledge of the criteria in our experiments to validate the classification results.

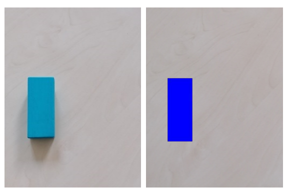
An explanation of all LIME parameters can be found in the official code documentation. We adopt the default settings for most of them except two parameters: We restrict the number of labels for which an explanation is made to two classes ($top\_labels = 2$). The parameter ($num\_features$) is adjusted for each experiment. It defines the number of similar pixel regions to be included in the explanation.

Experiment 1: Color

In a first experiment, we use a simple scenario where the color of an object is the decisive criterion. Each image shows one rectangular object on a plane while images with blue objects are labelled positive and images with green ones are labelled negative. We show examples of both classes including their processed variants after using object recognition techniques in Figure 2. The object positions are determined randomly and, just as the shape, should not be a relevant feature. While the decision criteria is obvious for a human being, a CNN has to be trained in order to recognize that only one characteristic is important. In contrast, we will see later that the system is able to extract relevant features in more complex scenes. This is an important aspect when it comes to real process environments. For this experiment we use 800 images for the training and 200 for the validation of the CNN. Based on the positive labelled images (cf. Figure 2b), local explanations are computed using LIME. Examples of results with $num\_features = 4$ can be seen in Figure 3. The LIME results show that the CNN's decision strongly depends on the object and its



(a) Negative example.



(b) Positive example.

Figure 2: Image data for experiment 1. Per example: original images (left) and their processed images (right).

[4]https://www.youtube.com/watch?v=QaMdjKE7vT8 (last accessed 17 Dec 2021)

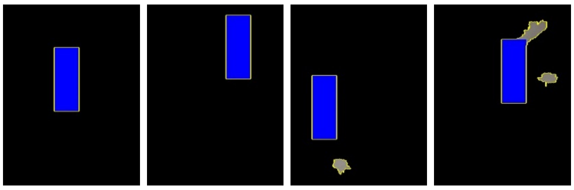


Figure 3: LIME results of experiment 1.

characteristics, while the background has no impact. The other computed results are comparable to those shown here. In 22% of all cases, the images contain only the object, while the rest of the image is blackened (cf. left images in Figure 3). In the remaining cases, some small parts of the background are visible (cf. right images in Figure 3). To derive a general conclusion, the characteristics of the remaining image content, i.e., the object, are analyzed. Pixels that do not have the background color are considered. Therefore, images where small parts of the background can be seen are processed as well. The analysis results in a list of valid positions and sizes as well as single values for the features shape and color. If these values are considered during task execution, the process is successful. For this purpose, the business process model is extended by this information. Therefore we edit the XML file describing the process model and change the name of the task in the respective line. We show an excerpt of the modified file in Listing 1.

```
<bpmn:task id=”A1” name=”place parts on charging plate;
        Position:{(51, 159), ...};
        Size:{(44, 116), ...};
        Shape:{’rectangle’};
        Color:{’blue’}”>...</bpmn:task>
```

Listing 1: Modification of the task-related tag.

To support readability, this information can also be appended as text annotation (cf. Figure 4). However, this information does not only contain the relevant detail, i.e., the blue color. Furthermore, the position and size lists are given which are hard to interpret. At this point, the analyzed information has to be related to the process context. In our experiment, all involved objects are rectangular and of same size. So the values for the features shape and size may be neglected. Considering the position list and object size, it can be derived that all positions on the plane are valid. Therefore also the position is no decisive feature. However, all objects are either blue or green confirming the color as relevant feature. The final result can be interpreted as instruction to place only blue parts on the charging plate.

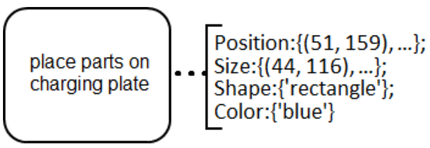
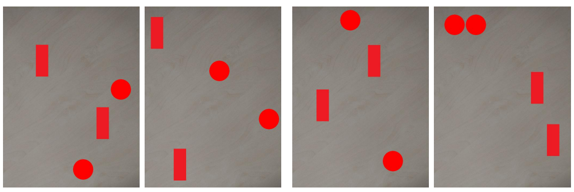


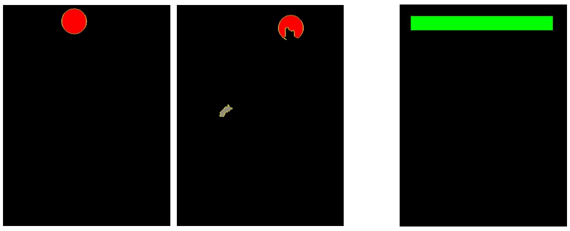
Figure 4: Analyzed details added as text annotation.

Experiment 2: Position and Shape

In a more complex placing experiment, each image shows four objects. Two of them are rectangular and the other two are circular. The color of all objects is the same. The object positions are determined randomly. We exclude intersections and ensure that no object protrudes beyond the image border. An image is labelled as positive, if at least one circular object is positioned in the upper seventh of the scene. Thus, the features shape and position are the relevant ones. Figure 5 shows examples leading to unsuccessful and successful process executions. Due to space limitations we show only the resulting images after using object recognition techniques. Out of the total 1000 images, 800 are used for training of the CNN and 200 are used for validation. Results after applying LIME with parameter $num\_features = 2$ are presented in Figure 6a. The left image shows the result after applying LIME on the positive labelled image presented in the left of Figure 5b. In all resulting images, one circular object is highlighted in the upper part of the plane. Among them, 29% of the results are comparable to the left image of Figure 6a. The others are comparable to the right image and either do not contain the circular object completely or additionally show small areas of the background. However, none of the results contain a rectangular object or both circular objects. This is an ideal result since all LIME images represent the condition that one circular object has to be placed in the upper part of the scene. Besides a list of positions and sizes, the analysis step outputs ”red” as single color value and ”circle, pentagon” as shape values. The wrong shape ”pentagon” occurs in few cases. It results from LIME explanations where the circular object is not completely visible (cf. right image in Figure 6a). Since the position list is hard to interpret, we suggest another representation. We suggest to create an image that shows the region of valid positions for placing an object (cf. Figure 6b). This representation allows to efficiently get an overview of all placement options. The region is computed by finding the minimal bounding box that encloses all object position values that are analyzed from LIME



(a) Negative examples. (b) Positive examples.
Figure 5: Image data for experiment 2.



(a) The resulting images highlight one circular object.

(b) Valid region to place objects.

Figure 6: LIME results of positive samples from experiment 2 (left). Bounding box of all valid object positions (right).

results. In this scenario, we suggest to attach this image to the respective task in the business process model by using media annotation concepts proposed by (Wiedmann, 2017). Process participants can view the attached image during process execution to get a visual extension of the task instruction. Furthermore, the task instruction itself is adjusted to include the analyzed and relevant features (color, shape). The XML file of the process model is modified according to experiment 1.

4.3 Evaluation and Discussion

The experiments confirm that our approach is able to identify relevant process details from images of process tasks. The relevant feature space can be determined by using CNNs without making any further assumptions about features. From a Business Process Management perspective, this is an important aspect. Since relevant details and features are usually not known in advance (cf. Section 1), assumptions regarding the feature space cannot be made. The concept of LIME is able to identify and visualize regions in the computed feature space that are relevant for successful process execution. In our experiments, the images created by LIME highlight single objects that take part in the scene. Considering the criterion that distinguishes negative from positive labelled images per use case, LIME produces expected and correct results in both experiments. The LIME results explain that the relevant information regarding process success is somehow related to the highlighted object. The rest of the image is irrelevant for the decision of the CNN. In the second experiment, pre-filtering to relevant content is essential to derive a general conclusion. It indicates that the rectangular objects are not relevant for process success. It further explains that only one circular object is relevant for the CNN's decision for the positive class. By analyzing the features of this remaining object across all LIME results, we are able to compute a general conclusion. Finding such a conclusion meets an important in the context of Business Process Management. As explained at the end of Section 3, a local explanation based on a single positive example is not sufficient as it restricts the scope of action of a process participant unnecessarily far.

In the experiments, we demonstrate the applicability of the concept in simple scenarios and with preprocessed image data using object recognition techniques. In real process environments, scenes can become much more complex. Although the presented approach also works with more complex input data, we propose to optimize it for more complex scenes. To meet associated requirements, the presented image analysis technique should be interchanged with more powerful image analysis techniques. Complex images require robust techniques that are able to explore large feature spaces. Furthermore, derived general conclusions have to be optimized in case of more complex scenes. To automate this issue the implementation has to be extended as follows. The general conclusion contains lists of object features and a set of values per feature. Each set has to be checked whether it contains all possible values that can occur in the process task. If this is true, the related object feature is not a relevant detail. For example, in experiment 1 the object shape is not a relevant feature since all objects involved in the process task are rectangular. Finally, a sufficient number of image data for training and validating the CNN may not be available in real process environments. In the case of limited data, we propose to integrate data augmentation techniques into the implementation.

5 CONCLUSION AND FUTURE WORK

In this work, we present an approach to identify missing process details of business process models that are relevant for an overall process success. In the proposed implementation, a CNN is trained with image data showing task scenes. The images are recorded during task execution and are labelled according to overall process success. We use LIME to explain the prediction of the CNN for samples of successful process executions. The results are images containing prediction-relevant regions. Across all results, these regions are analyzed using image analysis techniques to derive a general conclusion. It contains relevant features and values to achieve process success. This information is integrated into the process model. We evaluate our method with experiments using image data showing simplified scenarios from manufacturing process environments. Our experiments confirm that LIME and image mining techniques can be used to improve business processes. The results show that our approach is able to identify relevant process details. The process of metal injection molding or extrusion is taken as an example from real-world environments. The presented concept is applicable for a large set of similar processes that involve process participants in pick-and-place tasks. However, its flexibility allows to extend its scope of application to other tasks.

Future work should focus on using more powerful image mining techniques that increase the robustness and accuracy of the Image Analysis step. We aim to extend our experiments with more complex image data from real process environments. We further plan to evaluate the proposed ways of integrating process details in existing process models in a user study. This includes the investigation of how a certain feature must be represented in order to be interpreted correctly and efficiently by process participants.

REFERENCES

Van der Aalst, W. M. P., Van Dongen, B. F., Günther, C. W., Rozinat, A., Verbeek, E., and Weijters, T. (2009). Prom: The process mining toolkit, in bpm (demos). In *CEUR Workshop Proceedings*, volume 489.

Becker, J., Rosemann, M., and von Uthmann, C. (2000). *Guidelines of Business Process Modeling*, pages 30–49. Springer Berlin Heidelberg, Berlin, Heidelberg.

Burkart, N. and Huber, M. F. (2021). A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70:245–317.

Chinosi, M. and Trombetta, A. (2012). Bpmn: An introduction to the standard. *Computer Standards & Interfaces*, 34(1):124–134.

Dzindolet, M. T., Peterson, S. A., Pomranky, R. A., Pierce, L. G., G, L., and Beck, H. P. (2003). The role of trust in automation reliance. *International journal of human-computer studies*, 58(6):697–718.

Fichtner, M., Schönig, S., and Jablonski, S. (2020). Process management enhancement by using image mining techniques: A position paper. In *Proceedings of the 22nd International Conference on Enterprise Information Systems*, volume 1 of *ICEIS*, pages 249–255.

Fichtner, M., Schönig, S., and Jablonski, S. (2021). Using image mining techniques from a business process perspective. In *Enterprise Information Systems*, pages 62–83, Cham. Springer International Publishing.

Front, A., Rieu, D., Santorum, M., and Movahedian, F. (2017). A participative end-user method for multi-perspective business process elicitation and improvement. *Software & Systems Modeling*, 16(3):691–714.

Gounaris, A. (2016). Towards automated performance optimization of bpmn business processes. In *East European Conference on Advances in Databases and Information Systems*, pages 19–28. Springer.

Kluza, K., Baran, M., Bobek, S., and Nalepa, G. J. (2013). Overview of recommendation techniques in business process modeling. In *Proceedings of 9th Workshop on Knowledge Engineering and Software Engineering (KESE9)*, pages 46–57. Citeseer.

Mendling, J., Reijers, H. A., and van der Aalst, W. M. P. (2010). Seven process modeling guidelines (7pmg). *Information and Software Technology*, 52(2):127–136.

Niedermann, F., Radeschütz, S., and Mitschang, B. (2010). Deep business optimization: A platform for automated process optimization. *INFORMATIK 2010–Business Process and Service Science–Proceedings of ISSS and BPSC*.

Polyvyanyy, A., Smirnov, S., and Weske, M. (2008). Process model abstraction: A slider approach. In *2008 12th International IEEE Enterprise Distributed Object Computing Conference*, pages 325–331. IEEE.

Popescu, M. C. and Lucian, M. S. (2014). Feature extraction, feature selection and machine learning for image classification: A case study. In *2014 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM)*, pages 968–973. IEEE.

Prykäri, T., Czajkowsk, J., Alarousu, E., and Myllylä, R. (2010). Optical coherence tomography as an accurate inspection and quality evaluation technique in paper industry. *Optical review*, 17(3):218–222.

Radeschütz, S., Mitschang, B., and Leymann, F. (2008). Matching of process data and operational data for a deep business analysis. In *Enterprise Interoperability III*, pages 171–182. Springer.

Reichert, M., Kolb, J., Bobrik, R., and Bauer, T. (2012). Enabling personalized visualization of large business processes through parameterizable views. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 1653–1660.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Schmidt, R., Möhring, M., Zimmermann, A., Härting, R.-C., and Keller, B. (2016). Potentials of image mining for business process management. In *Intelligent Decision Technologies 2016*, pages 429–440. Springer.

Schonenberg, H., Weber, B., van Dongen, B., and van der Aalst, W. M. P. (2008). Supporting flexible processes through recommendations based on history. In *International Conference on Business Process Management*, pages 51–66. Springer.

Wiedmann, P. C. K. (2017). *Agiles Geschäftsprozessmanagement auf Basis gebrauchssprachlicher Modellierung*. PhD thesis, University of Bayreuth, Germany.