IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 3 December 2024, accepted 19 December 2024, date of publication 23 December 2024,
date of current version 31 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3521497

RESEARCH ARTICLE

A Novel Deep Learning-Based Approach for Defect Detection of Synthetic Leather Using Gaussian Filtering

CHRISTOPHER MAI, PASCAL PENAVA, (Member, IEEE), AND RICARDO BUETTNER, (Senior Member, IEEE)
Chair of Hybrid Intelligence, Helmut-Schmidt-University/University of the Federal Armed Forces Hamburg, 22043 Hamburg, Germany

Corresponding author: Ricardo Buettner (buettner@hsu-hh.de)

This publication has been funded by the Open-Access-Publication-Fund of the Helmut-Schmidt-University/University of the Federal Armed Forces Hamburg.

ABSTRACT Synthetic leather is a commonly used material, especially in the clothing industry. It is employed in the production of footwear, such as shoes and boots, as well as handbags and other accessories. Prior to exportation, the leather is subjected to a series of processing stages, which may result in the formation of surface defects. Quality assurance, which serves to identify defective leather in a timely manner, is still partially conducted by humans. This step is time-consuming and is associated with a higher error rate. To address these challenges, the deployment of image processing systems for the automated inspection of leather defects is becoming increasingly crucial. The objective of this study is to develop an effective deep learning model for the classification of defective synthetic leather, with the aim of implementing it in the quality control of synthetic leather production. To this end, a variety of architectural approaches are employed, including Xception, InceptionV3, ResNet50V2, VGG19, and VGG16. In addition to hyperparameter tuning, all architectures utilize transfer learning. Moreover, the impact of the Gaussian filter as a pre-processing step for images is examined. The results show that by using an innovative Gaussian filtering approach, an accuracy of 100% was achieved with the VGG16 model. Furthermore, the results show that the filtering approach has a positive influence on the accuracy of all other models. The high accuracies of the deep learning models show that the use of machine vision systems for the automatic inspection of defects in the synthetic leather industry is an effective economic step for companies, as it leads to cost and time savings in production due to the high classification performance. This study provides an overview of the results achieved and other key performance indicators for all the models used.

INDEX TERMS Deep learning, Gaussian filter, synthetic leather, defect detection, transfer learning, pre-trained architectures, data augmentation, hyperparameter tuning.

I. INTRODUCTION

Leather is one of the most traded commodities, with great economic importance [1]. It can be divided into genuine leather and synthetic leather. Genuine leather is derived from the rawhide and skin of animals such as cows, crocodiles, or goats. It is a natural material that is water-repellent, elastic, and durable. Due to its high quality and durability, it is a popular textile material that is utilized in the production of clothing, car accessories, and shoes, among other products [2]. Synthetic leather, also known as artificial leather or faux leather, is a widely used material due to its variety, mass production, and widespread manufacture [3]; it is also an alternative to leather of animal origin. It is made from a textile base coated with an application of polymer in the form of lacquer or emulsion, typically using polyurethane (PU), polyvinyl chloride (PVC), or polypropylene [4], that simulates the mesh and grain layers

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Ayoub Khan.

of genuine leather [5]. The advancement and manufacturing of textile synthetic leather have significantly addressed the rising consumer demand for leather products [5]. Synthetic leather is manufactured to imitate the appearance and, if desired, the texture of genuine leather [6], sometimes with better physical properties such as higher durability or better stitch tear [7]. Further advantages are that the production of synthetic leather does not necessitate the use of animal skins, thereby eliminating the need to kill animals for their use. Synthetic leather also has no typical defects that are common in genuine leather, such as scars, wire scratches, fire marks for identification of the animal, spots, burns, and marks caused by insects [8]. Another significant advantage is that the cost of synthetic leather is generally lower than that of genuine leather, which is due to the economies of scale associated with mass production [9]. Furthermore, it offers advantages in the context of production, as it provides standardized sizes and thicknesses and is available in roll form, resulting in a reduction of waste during cutting and facilitating more effective planning [8]. Synthetic leather can be used in many different ways, including making shoes, boots, clothing, and bags [10], furniture fabrics, clothing fabrics, automobile interior decoration, and medical equipment [11]. Additionally, modified graphene can enhance synthetic leather with mold protection efficient antibacterial and self-cleaning properties, and it can be used to apply functional polyurethane coatings in automotive and aircraft interiors [11]. Synthetic leather and genuine leather are produced using different processes. The production of genuine leather requires a higher degree of manual labor. The production of synthetic leather, on the other hand, is a highly automated process [12]. For both types, automatic quality control is a crucial aspect of leather production, as it enables the early detection and rejection of defective material that may result from the various processing steps. In certain instances, the quality control of synthetic leather is also conducted by human workers [12]. However, this manual approach is both time-consuming and costly. Despite its significant importance, however, automatic quality control in leather production remains largely limited to genuine leather. Nevertheless, it is of great importance in the highly automated production of synthetic leather.

In recent years, deep learning methods have achieved remarkable success in image recognition and classification. Among the various deep learning models, the convolutional neural network (CNN) is the most widely used one, which is most suitable for the binary classification of defective or non-defective leather material for quality inspection. Prior research has demonstrated the efficacy of deep learning techniques in object recognition, particularly in the context of defect detection in synthetic leather [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24]. Various approaches have been explored in this domain. One is the reconstruction-based method, which involves reconstructing all input images into a defect-free state that was previously trained. The presence of defects can then be identified by comparing the original image with the reconstructed image [17], [18]. In the work of Srilakshmi et al. [22], they developed an automated pipeline for visual inspection with the objective of detecting defects. In this process, they employed a customized VGG16 model, which had been pre-trained on the ImageNet [25] dataset.

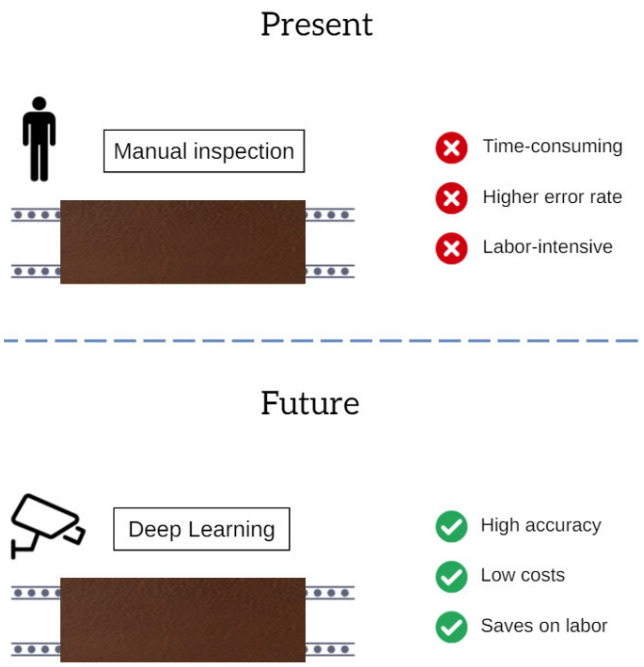

FIGURE 1. Comparison of the present manual quality inspection process in synthetic leather production and its potential future evolution, which could offer significant advantages through the integration of modern deep learning methods.

Despite the aforementioned methodologies, there is a continued necessity for an efficacious model that is able to accurately differentiate between defective and non-defective leather so that it can be employed in automated quality control within the synthetic leather industry. We address this problem by using different state-of-the-art pre-trained models and set a new benchmark in the detection of defects in synthetic leather. Figure 1 illustrates the existing manual inspection approach for synthetic leather and a prospective alternative based on deep learning, which offers several benefits. For the development and evaluation of our model, we employ the MVTec dataset [26], comprehensively detailed in Subsection III-D, comprising labeled images of synthetic leather. In the context of the classification task, the utilization of pre-trained architectures is advantageous, as it allows for the application of transfer learning, which is particularly beneficial in the case of the dataset under consideration. To the best of our knowledge, only a limited number of studies have employed supervised learning models for the detection of synthetic leather defects, such as [20] and [22].

In light of the dearth of literature examining defects in synthetic leather through the lens of supervised learning and pre-trained architectures, we also compare our results with

those of researchers who have used semi-supervised and unsupervised learning [13], [15], [16], [17], [18]. In addition to the models, we are investigating the influence of the Gaussian filter on synthetic leather. In areas where no pre-trained architectures are used, the Gaussian filter is used in the following works: [27], [28]. In these papers, the authors used the filter to emphasize the defect in the leather and suppress the meaningless information. However, a comparison of the effects of this filter on the overall result was not performed. Based on these two references, we hypothesize a positive impact of applying the Gaussian filter to synthetic leather. Given that synthetic leather emulates the surface structure of genuine leather, it seems reasonable to posit that the observed effects on genuine leather, such as the enhancement of defect visibility, are likely transferable to the synthetic leather context. As far as we know, there is no work investigating the influence of the Gaussian filter on synthetic leather in relation to pre-trained architectures in the context of supervised learning. The results of this study can be used to improve quality control in the synthetic leather industry. The aim is to detect defects in synthetic leather goods as effectively as possible and thus maintain the competitiveness of the sector. In this study, the following pre-trained CNN models are used: VGG16, VGG19, ResNet50V2, InceptionV3, and Xception. Hyperparameter tuning is performed for all models to determine the optimal parameters. Furthermore, the Gaussian filter is used to pre-process the images. The main results of this study are as follows:

1) Applying innovative Gaussian filtering, we were able to develop a VGG16-based model that achieved 100% accuracy in distinguishing between defect-free and defective synthetic leather images.

2) We show that applying the Gaussian filter as a pre-processing step led to improved accuracy for all pre-trained architectures used.

Our paper is structured as follows: Section II provides an overview of related work in the field of leather defect detection by supervised, semi-supervised, and unsupervised learning methods, the basics of transfer learning are also explained. Section III contains information about the deep learning approach used, a brief description of the deep learning architectures used, the Gaussian filter utilized as a pre-processing step, followed by a detailed explanation of the training and test process and the dataset used for the evaluation. In Section IV, we present the results, followed by the discussion in Section V. Finally, Section VI presents a summary of the key findings and highlights the study's contributions.

II. RELATED WORK

Deep learning methods for defect detection approaches can be divided into three categories, supervised, semi-supervised, and unsupervised learning, with the main difference being the extent of labeled image requirements: supervised methods need labels for all classes, semi-supervised only for defect-free instances, and unsupervised require no labels, it tends to be possible to say here that semi-supervised methods perform worse than those of supervised learning [29]. However, supervised learning methods tend to perform exceptionally well when the dataset includes a sufficient number of examples for each class, thereby enabling the model to learn more effectively from the comprehensive labeled data [29].

This section provides a concise overview of the evolution of contemporary convolutional neural networks and transfer learning. Subsequently, relevant studies on defect detection from recent years are presented, with the focus of this work being on the use of supervised learning with pre-trained architectures for classification.

A. EVOLUTION OF MODERN CONVOLUTIONAL NEURAL NETWORKS

CNNs are a deep learning architecture that is good at classifying images. They are able to do the feature extraction and final classification in a combined step [30]. The development of backpropagation in 1990 marked a significant milestone toward the CNNs we use today, initially demonstrated by a network capable of recognizing handwritten digits [31]. Today, CNNs are still state-of-the-art regarding image recognition and classification [32], but they have nothing in common with the CNNs of the past and are much more capable of more complex recognition tasks [33]. With their novel architecture, AlexNet, Krizhevsky et al. [34] achieved the first major breakthrough in solving more complex tasks and have created the first model ever to win the ImageNet competition. ImageNet represents one of the most used research image datasets. Over the next few years until today, VGGNet [35], GoogleNet (Inception Models) [36], and ResNet [37] were the architectures that were able to build on the success and achieve even better accuracies.

B. TRANSFER LEARNING BASICS

Transfer learning is classified in the field of machine learning. Due to its versatile applicability, transfer learning has become a popular and promising area in machine learning [38]. Using transfer learning strategies can significantly reduce the required amount of training data [39]. The goal of transfer learning is to improve the performance of learning models in specific target domains by transferring knowledge from different but related source domains. In this method, networks are trained in advance with a large dataset. Databases such as ImageNet can be employed for this purpose. The pre-trained network and convolutional layer weights are then copied and applied to the new classification task. This can drastically reduce the reliance on a large amount of data in the target domains for building learning models [40]. Transfer learning is a very efficient method to perform related tasks. An alternative approach is pre-training. The only difference to transfer learning is that this method has greater flexibility in designing the network architecture since this can be defined in advance [41]. The advantage of transfer learning, which enables the transfer of knowledge from large datasets,

is particularly advantageous in this work, which uses a small dataset that can also occur in a practical scenario.

C. DEEP LEARNING METHODS USING SYNTHETIC LEATHER AND SEMI- OR UNSUPERVISED LEARNING

Semi-supervised or unsupervised learning methods are often used for defect detection because they require few to no labeled images to detect defects. Li et al. [13] designed a novel multi-classification AD (MCAD) framework to achieve high accuracy in anomaly detection. They used a teacher-student model with ResNet18 as the teacher and ResNet10 as the student and trained the multi-classification model via transfer learning. In [14], Jiang et al. introduced an extension of the PatchCore method called Feature-Level Registration PatchCore (FR-PatchCore) to test anomaly detection. FR-PatchCore constructs a feature matrix, which is stored in the memory bank and continuously updated using the optimal negative cosine similarity loss. In another work [15], the authors present their method named DRÆM (discriminative joint reconstruction anomaly embedding method), which belongs to the reconstructive approach. The method simultaneously learns a combined representation of an anomalous image and its anomaly-free reconstruction while developing a decision boundary to discriminate between normal and anomalous examples. To detect defects, Rudolph et al. [23] utilized DifferNet, which applies normalizing-flow-based density estimation of image features at multiple scales to identify defects. This approach utilizes the likelihoods of several transformations of a single image to compute a robust anomaly score. The authors Yang et al. [16], used a memory-based end-to-end segmentation network (MemSeg) to detect surface defects. It introduces artificially simulated abnormal samples and memory samples. It introduces artificially simulated abnormal samples and a memory pool. During training, it explicitly learns the differences between normal and abnormal images while storing general patterns of normal samples. By comparing input data with memory samples, it identifies abnormal regions effectively. Luo et al. [17] used a reconstruction-based method called Adaptive Mask Inpainting Network (AMI-Net). AMI-Net restores defective images to their normal state and then detects defects by analyzing the differences between the restored images and the originals. In another work [18], the authors Napoletano, Piccoli, and Schettini also used a reconstruction-based method. They used a pre-trained CNN autoencoder with a statistical transformation to remove anomalies from input images, then compared the cleaned images to the originals for anomaly detection and localization. In [24], the authors present a framework for Patch Distribution Modeling (PaDiM) to simultaneously detect and localize anomalies in images within a one-class learning setting. PaDiM uses a pre-trained CNN for patch embedding, models the normal class with multivariate Gaussian distributions, and enhances anomaly localization by leveraging semantic correlations within the CNN.

TABLE 1. Summary of related work for the classification task of synthetic leather using pre-trained architectures.

Author	Pre-processing filter	Model	Highest Accuracy
Srilakshmi et al. (2022) [22]	not used	VGG16	94.66%
Adão et al. (2022) [19]	not used	Xception	96.00%
Zuluaga et al. (2023) [21]	not used	DarkNet19, SqueezeNet	100% 100%
Fuqin et al. (2024) [20]	not used	ResNet34	92.6% (defective) 92.8% (no defect)

D. DEEP LEARNING METHODS USING SYNTHETIC LEATHER, PRE-TRAINED ARCHITECTURES, AND SUPERVISED LEARNING

In order to benefit from the advantages of transfer learning, we focus on the use of pre-trained architectures in this paper. According to our research, only a few authors deal with the defect detection of synthetic leather using pre-trained architectures. Fuqin et al. [20] developed a defect generation method with multiple loss functions, named DG2GAN, to address issues such as insufficient variety and poor quality of the augmented data. This method utilizes cycle consistency loss to generate defect images from a vast number of defect-free synthetic leather images. The authors tested their method with a ResNet34 model and achieved accuracies of 92.6% (defective) and 92.8% (no defect). In their study, Adao et al. [19] employed a Xception model on distinct variants of their dataset. They utilized a grid to mark defects in synthetic leather. Two annotation styles were utilized to address a binary classification task aimed at differentiating between defective and non-defective images, achieving an accuracy of 96%. Zuluaga et al. [21] addressed the challenge of insufficient sample images by implementing federated learning. In their study, the DarkNet19 and SqueezeNet architectures were employed, achieving an accuracy of 100% with both models. Srilakshmi et al. [22] developed an automated visual inspection (AVI) pipeline for defect detection. A custom VGG16 architecture, pre-trained on ImageNet, was employed for the classification of leather as either non-defective or defective. The approach yielded an accuracy of 94.66%.

Table 1 provides a summary of the aforementioned works. A review of the literature reveals a paucity of research on the defect detection of synthetic leather using supervised learning and pre-trained architectures. It is worth noting that the known pre-trained architectures, such as InceptionV3 or VGG19, have yet to be subjected to examination. Furthermore, it is evident that no pre-processing filters were employed in the aforementioned prior work. Nevertheless, the application of pre-processing filters has the potential

to enhance the efficacy of the model. For instance, the Gaussian filter has been demonstrated to be effective in the context of genuine leather [27], [28]. Specifically related to the two references [27] and [28], it is assumed that the application of the Gaussian filter to synthetic leather could lead to positive results. In the aforementioned references, the filter is employed to emphasize defective areas and attenuate meaningless regions. Given that synthetic leather emulates the surface structure of genuine leather, it seems reasonable to posit that the observed effects on genuine leather, such as the enhancement of defect visibility, are likely transferable to the synthetic leather context. To the best of our knowledge, no prior research has examined the impact of the Gaussian filter on synthetic leather in conjunction with pre-trained architectures in the context of supervised learning. With these considerations in mind, this work explores the utilization of diverse pre-trained architectures for the binary classification of synthetic leather, utilizing the Gaussian filter as a pre-processing filter for the images and examining its effects on model accuracy. The findings are presented in this study.

III. METHODOLOGY

The objective of this study is to develop an effective deep learning model for the classification of defective synthetic leather. In preparation for training, the images of the test set for each category of defective images are combined into one category from the original leather dataset. This results in two categories being available for the binary classification problem: one class where only defective images are included and one where defect-free images are included. The entire dataset is first converted to a NumPy array, then resized to 150 × 150 px and pre-processed with a Gaussian filter for the sequential data manipulation. The architectures VGG16, VGG19, ResNet50V2, Xception, and InceptionV3 were used for defect detection. A stratified 5-fold cross-validation is then applied to all five models during the training process. For each split of the cross-validation, the best hyperparameters for the model are first searched. Before training the full model, the additional dense layer is trained using the transfer learning approach. For each training, the original class weights of the specific object are used. Finally, the performance is evaluated using images from a previously unused portion of the data.

A. MODEL ARCHITECTURE

In this study, we use several pre-trained architectures that have the advantage of being able to use the weights of larger datasets, such as those from ImageNet. The selection of models in the context of supervised learning is a crucial decision.

In this study, the choice was informed by prior research, as summarized in Table 1, as well as by studies conducted beyond the specific application of synthetic leather. Table 1 further highlights that numerous models have not yet been applied to synthetic leather. Taking this fact into account, alongside insights from the literature review, we selected the models to be utilized in this study. Furthermore, it was decided to include only a single variant of each architecture in the study. An exception was made for VGG16 and VGG19, as these architectures are frequently employed in the literature. It was therefore considered beneficial to incorporate both VGG architectures in the analysis. The ResNet variant ResNet50V2 was selected due to its structural similarity to ResNet50 while being less deep and achieving higher accuracy on the ImageNet dataset. A schematic representation of the adapted VGG16 architecture, which achieved the highest values for balanced accuracy in this study, can be seen in Figure 2. VGG16 and VGG19 were both introduced in 2015 by Simonyan and Zisserman [35] and represent a deep CNN. VGG16 consists of 16 layers, including 13 convolutional and three fully connected layers. Specifically, the model consists of five convolutional blocks: the first and second blocks each have two layers, while the remaining blocks each contain three convolutional layers. These layers extract features from the input image, while the 2 × 2 max-pooling layers, one after each of the five blocks, downsample the feature maps and reduce their spatial dimensions. With 3 × 3 kernel-sized filters, two of them in the first two blocks and three in the last three blocks, they show an architecture with very small convolution filters. The model has a total of 138.4 million parameters. VGG19 has a similar design, and it consists of 19 layers, including 16 convolutional and three fully connected layers. Like VGG16, VGG19 also consists of five convolutional: the

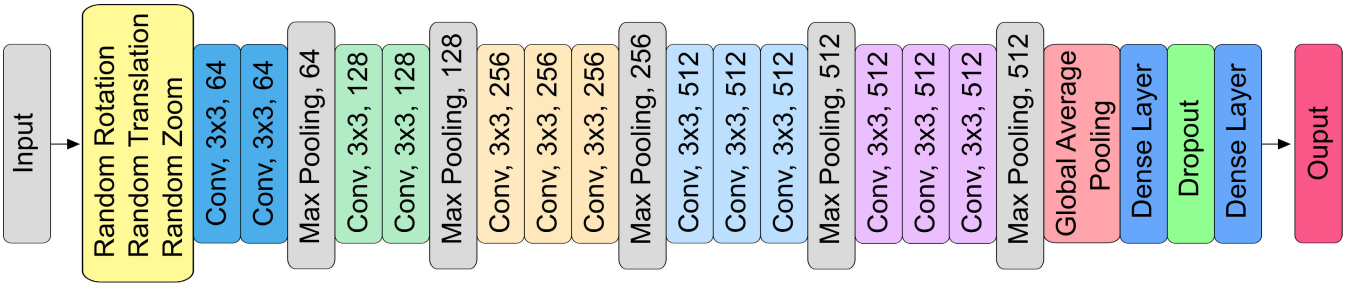

FIGURE 2. Schematic representation of the adapted VGG16 architecture, which achieved the highest balanced accuracy in this study.

first and second blocks each have two layers, and the three remaining blocks each contain four convolutional layers. Because of this, VGG19 has approximately 143.7 million parameters, 5.3 million more than VGG16.

When training deep neural nets, there comes a point where the accuracy does not improve or deteriorate with increasing depth despite adding more layers. This phenomenon is not due to overfitting, as adding more layers paradoxically increases the training error [37]. This is known as the degradation problem. To solve this problem, He et al. [37] introduced an architecture named Residual Network (ResNet). ResNet has been proposed in different depths, such as ResNet18, ResNet34, ResNet50, ResNet101, and ResNet152, where the numbers indicate the number of layers in each model. A residual block consists of a series of layers that are connected to each other by so-called shortcut connections. These connections allow activations to be passed directly from one layer to the next, improving the flow of information and the efficiency of modeling [37]. So if the current layer is not needed, it can be bypassed thanks to this identity, which reduces overfitting issues [37].

Szegedy et al. [36], [42] introduced InceptionV3 as Google's third version of their inception CNN. The idea of the inception models is to use different kernel-sized filters for the convolution on the same level. Thereby, a variation in the location of information can be intercepted, and thus overfitting problems sink, and classification results increase for images with distributed information. The Inception models thus increase accuracy not by getting "deeper" but "wider" [36], [42]. The architecture of InceptionV3 consists of several convolutional layers, some of which are combined in so-called Inception modules. Compared to its predecessors V1 and V2, InceptionV3 is more efficient, less computationally intensive, has a deeper network, uses auxiliary classifiers as regulators, and factorized convolutions to break up large convolutions, e.g., a 7×7 convolution is broken down into two 3×3 convolutions [42].

In 2017, Chollet [43] introduced the Xception architecture, which is based on an extension of the Inception architecture principle, where Inception modules have been replaced with depthwise separable convolutions with residual connections [43]. The in total, 36 convolutional layers are organized into 14 modules, each featuring linear residual connections, except for the first and last modules. Xception has almost as many parameters as InceptionV3, but outperforms InceptionV3 on the ImageNet dataset [43].

All these pre-trained models were used for transfer learning. Before the images of the selected dataset can be fed into the CNN models, they were pre-processed using the built-in ImageDataGenerator function of Keras. Because of the usage of transfer learning, the base model is switched to inference mode so that it does not update the weights it has already learned. All pre-trained models were pre-trained with the ImageNet dataset. The top layer of each of the base models is excluded, as it is specific to the dataset used to train the respective model. For deep learning models, larger datasets lead to better accuracies. However, the problem with large datasets is that they require a lot of effort to acquire and label the data. For this reason, data augmentation is used to generate good deep learning models even with smaller datasets [44]. After the base model, three pre-processing layers for random rotation, random zoom, and random translation (horizontal and vertical) are added. The parameter values of these layers are determined by the hyperparameter tuning; the intervals can be seen in table 2. Following the pre-processing layers, a global average pooling layer was implemented. After this pooling layer, the model consists of a dense layer, which is followed by a dropout layer designed to randomly disable some neurons during training to prevent overfitting. The model concludes with a final dense layer. The number of neurons in the first dense layer is not fixed; instead, it is determined within the range of [32, 128] based on hyperparameter tuning to identify the optimal configuration. The activation function for the first dense layer is set to ReLU, which introduces non-linearity into the model. The final dense layer, responsible for producing the output, contains a single neuron with a sigmoid activation function, which is appropriate for binary classification tasks.

B. GAUSSIAN FILTER

In image processing, a Gaussian low-pass filter, also called Gaussian blur, is commonly used to smooth images by removing noise, effectively eliminating the high-frequency components of the image [45]. The Gaussian filter uses a Gaussian function to calculate the transformation applied to each pixel in the image [46]. This function in two dimensions is defined as follows [45]:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{1}$$

where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis and σ represents the standard deviation of the Gaussian distribution. To apply the Gaussian filter to the image, a kernel is created by sampling the Gaussian function at various distances from the center point. This processing technique blurs the edges of an image by smoothing out the pixel values, as the differing edge pixels, having lower weights, are replaced by the surrounding pixel values [47]. When smoothing through the filter, the edges are better preserved than with a median filter [48]. The size of the kernel and the variance of the Gaussian function determine the amount of blurring applied to the image.

Gaussian filter is applied to the input image using the *cv2.GaussianBlur*() function from the OpenCV library. In this paper, the kernel size is defined as 5×5, and the Kernel standard deviation in x and y direction is set to 0. This means σ is calculated by the kernel and has no fixed value [49]. The result of using the Gaussian filter with the above settings for σ and kernel size can be seen in Figure 3. A clear blurring to the original image is recognizable.

FIGURE 3. Effect of the Gaussian filter on a defect leather image from MVTec dataset [26].

C. PROCESS OF TRAINING

A schematic representation of the training process is provided in Figure 5, while the basic structure of all the models utilized can be seen in Figure 4. Before training the model, a stratified 5-fold cross-validation from the scikit-learn [50] library was used to split the entire dataset into training and test sets to ensure that each fold of the dataset had the same proportion of classes. In addition, the parameter *random_state* was used to make all models comparable. For each fold, 10% of the training data is selected for validation for hyperparameter tuning.

TABLE 2. Overview of the hyperparameters to be optimized with their search interval.

Hyperparameter	Minimum Value	Maximum Value	Step
Hyper-parameter	**Minimum Value**	**Maximum Value**	**Step**
Random Rotation	0.05	0.2	0.05
Random Zoom	0.05	0.2	0.05
Random Translation Height	0.05	0.2	0.05
Random Translation Width	0.05	0.2	0.05
Dimensionality of Dense layer	32	128	16
Dropout Rate	0	0.5	0.1
Learning Rate	10^{-5}	10^{-2}	calculated logarithmi-cally
Solver	-	-	Adam, SGD, RMSprop

To train the model, the batch size is set to 16 and the total number of epochs to 100. The input images for the pre-trained models are scaled to a resolution of 150 × 150 px with three RGB color channels. The training process of the model created from scratch and the transfer learning model

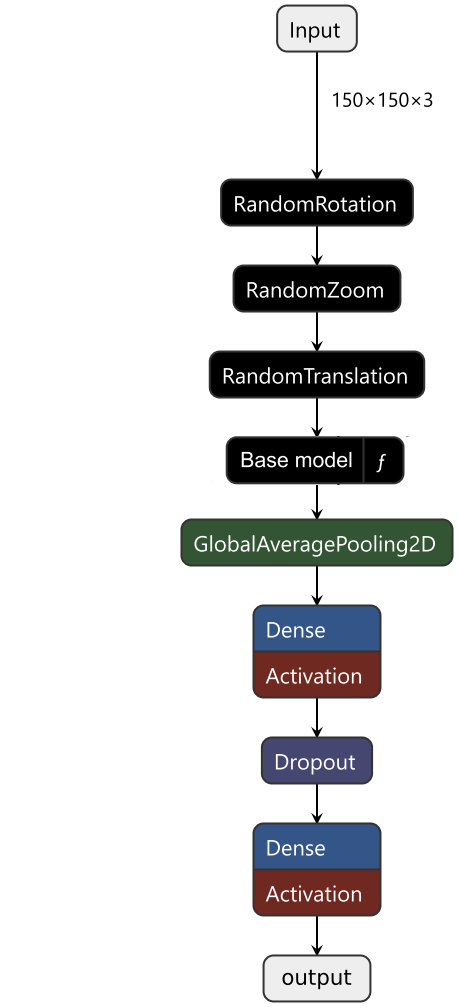

FIGURE 4. Basic structure of the applied models.

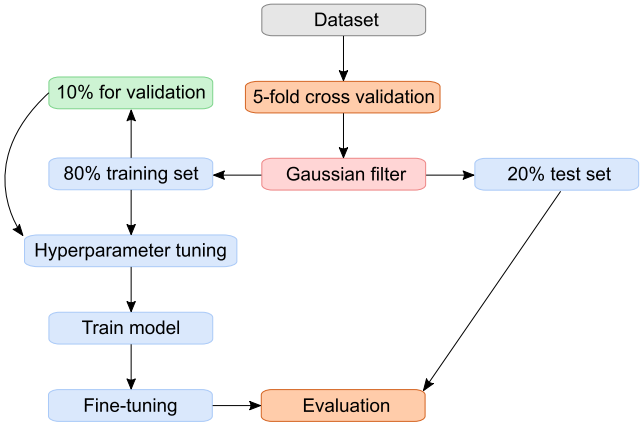

FIGURE 5. An illustration about the training and test procedure using a Gaussian filter.

differ only slightly. While three steps are used for the transfer learning model, only the first two steps are used for the model created from scratch. First, the training, test and validation data are pre-processed with a Gaussian filter with a 5 ×

5 kernel and a standard deviation $\sigma = 0$ for the horizontal and vertical directions, where $\sigma = 0$ means that the standard deviation is calculated from the kernel size [49]. Second, a Bayesian optimization with KerasTuner [51] is deployed to identify the best hyperparameters. The provided options for the Bayesian optimization can be found in Table 2. For the Bayesian optimization process, 30 trials are conducted to identify the parameters that yield the lowest validation loss. Each of these 30 trials is permitted to run for a maximum of 40 epochs. Additionally, the *earlys topping* callback function is employed, whereby the process is terminated if the validation loss does not decrease over ten consecutive epochs. Once the optimal parameters are identified through this process, they are then used to train the final model. For the individual pre-trained models, the input files must be available in a certain way, for example, the images must be converted from RGB to BGR for VGG19. For ResNet50V2, on the other hand, the pixel values of the input data must be scaled in the range between $[-1, 1]$. This pre-processing step is carried out before hyperparameter tuning. To enhance classification performance and minimize overfitting, the training set was expanded using common data augmentation techniques such as zoom, rotation, and translation. In the third step of the training, the custom layers are trained using patience of ten at 100 epochs. Once the validation loss has not increased for ten epochs, the training is terminated through callbacks. The model of the epoch with the best results is automatically restored. During the hyperparameter tuning and Bayesian optimization steps, the base model is frozen. The fourth and final step involves applying transfer learning to the model. During this step, the base model is unfrozen, allowing it to be fine-tuned for enhanced performance. The learning rate is set to a minimum of 1×10^{-6}. The number of epochs for this phase is determined by adding 30 to the best epoch from the previous training. Additionally, the same conditions and parameters that were employed in the initial training step are utilized for this fine-tuning process. After completing the final step, the model undergoes an evaluation. The performance indicators and metrics are calculated using the associated test subset. This evaluation employs common functions such as *accuracy_score*() and *f* 1*_score*() from the scikit-learn library. These functions are used to assess various aspects of the model's performance. All relevant metrics are computed and stored for each fold in the cross-validation process. At the conclusion of this process, the average values of all five folds are calculated and reported, providing a comprehensive summary of the model's overall performance.

D. DATASET

This study utilizes the MVTec Anomaly Detection dataset [26], which comprises 15 classes and a total of 5,354 unannotated color images of 15 distinct objects, with resolutions ranging from 700 × 700 px to 1024 × 1024 px. The dataset is divided into 3,629 images designated for training and 1,725 images for testing. The training set exclusively contains anomaly-free images, whereas the test set includes both normal and anomalous images, along with anomaly ground truth mask labels for segmentation evaluation. On average, each class in the training set includes 242 images.

In the synthetic leather study, the dataset comprises a total of 369 images, with 245 allocated for training and 124 for testing. Among the 124 test images, 32 are defect-free and classified under the *good* category, while the remaining 92 images contain defects. The test set is further categorized into six groups: Color (19 images), Cut (19 images), Fold (17 images), Glue (19 images), Good (32 images), and Stitch (18 images). The folder names in the test dataset indicate the type of defect, with the numbers in parentheses specifying the quantity of images available for each category. All images in this dataset have a resolution of 1024 × 1024 px. The categories of the synthetic leather dataset are shown in Figure 6.

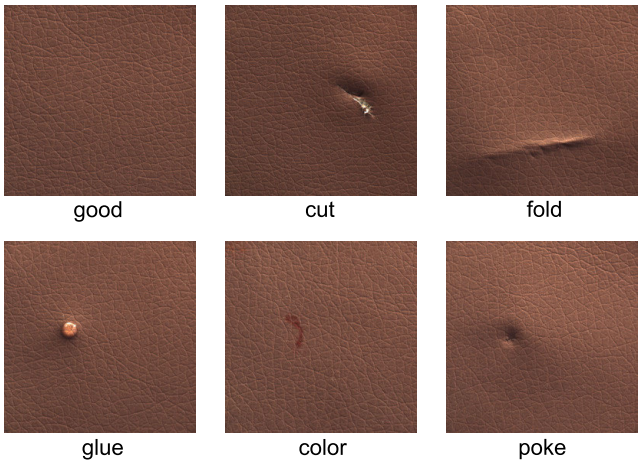

FIGURE 6. Sample images from each class found in the leather test set from MVTec [26].

E. EVALUATION METRICS

To evaluate and interpret the model's performance, we employ the following performance indicators: Accuracy, Balanced accuracy, True positive rate (sensitivity or recall), True negative rate (Specificity), Positive predictive value (precision), Negative predictive value, Cohen's Kappa, F1-Score and at least Area Under the Curve (AUC).

The Accuracy determines the overall effectiveness of a model [52]. To mitigate inflated performance estimates on imbalanced datasets, balanced accuracy can be utilized, defined as the arithmetic mean of sensitivity and specificity [53], [54]. The True positive rate metric indicates the accuracy of classifying the positive class, and maximizing it increases the likelihood of correctly identifying true members of the positive class [55]. The True negative rate, on the other hand, indicates how effectively a classifier identifies negative classes [52]. The Positive predictive value metric assesses prediction accuracy for the positive class by indicating the proportion of positive predictions that correctly match

TABLE 3. Results of the performance indicators for the proposed approach in the notation: no Gaussian filter | with Gaussian filter.

Performance indicator	Xception	InceptionV3	ResNet50V2	VGG19	VGG16
Accuracy	97.02% | 99.46%	97.29% | 98.92%	98.92% | 99.46%	99.46% | 99.73%	99.73% | 100.0%
Balanced accuracy	95.83% | 98.92%	97.44% | 97.78%	97.91% | 98.89%	98.92% | 99.44%	99.44% | 100.0%
True positive rate	98.20% | 100.0%	97.11% | 100.0%	100.0% | 100.0%	100.0% | 100.0%	100.0% | 100.0%
True negative rate	93.45% | 97.84%	97.78% | 95.56%	95.61% | 97.78%	97.84% | 98.89%	98.89% | 100.0%
Positive predictive value	97.90% | 99.29%	99.27% | 98.62%	98.60% | 99.29%	99.29% | 99.65%	99.65% | 100.0%
Negative predictive value	94.64% | 100.0%	92.52% | 100.0%	100.0% | 100.0%	100.0% | 100.0%	100.0% | 100.0%
Kappa	0.9190 | 0.9853	0.9307 | 0.9695	0.9701 | 0.9850	0.9853 | 0.9925	0.9925 | 1.0
F1-Score	98.03% | 99.64%	98.15% | 99.30%	99.29% | 99.64%	99.64% | 99.82%	99.82% | 100.0%
AUC	98.41% | 99.98%	99.80% | 99.96%	99.84% | 100.0%	100.0% | 100.0%	99.98% | 100.0%

true positive instances [55]. The Negative Predictive Value, equivalent to the precision for the negative class, is the ratio of correctly classified negative samples to all samples classified as negative [56]. Cohen's kappa describes the reliability of a model by measuring the agreement between two judgments. It ranges from -1 to 1, where a Cohen's kappa of -1 indicates complete disagreement, and a Cohen's kappa of 1 signifies perfect agreement [57]. The harmonic mean of precision and recall, called F1-Score, ranges from 0 to 1, with the minimum (0) occurring when all positive samples are misclassified (true positives $= 0$), and the maximum (1) occurring for perfect classification (false negatives $=$ false positives $= 0$) [58]. AUC means Area Under the (ROC)-Curve. The ROC curve, obtained by plotting sensitivity against the false positive rate at all possible threshold values, is a monotonically increasing function within the unit square from (0, 0) to (1, 1), where closer proximity to (0, 1) indicates better predictions [59]. AUC, a scalar number between 0 and 1 representing the expected performance of the ROC curve (with 1 being the best), has the important statistical property of equating to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance [60].

IV. RESULTS

Our approach has yielded the results presented in Table 3. As previously described, the models were evaluated using stratified 5-fold cross-validation, and the average values of the metrics can be found in this table. The pre-trained architectures were evaluated in terms of accuracy, the class-averaged sensitivity (true positive rate), Positive predictive value (precision), and Cohen's Kappa. Additionally, the F1-score, balanced accuracy, true negative rate, negative predictive value, and area under the receiver operating characteristic curve (AUC) are reported.

This shows that for the architectures of Xception, ResNet50V2, VGG19 and VGG16, the use of the Gaussian filter with the settings $\sigma = 0$ and a 5×5 kernel does not lead to a deterioration of the metrics but rather causes them to be positively influenced. Only InceptionV3 shows a deterioration in the values for the true negative rate (-2.22%) and positive predicted value (-0.65%) when using the Gaussian filter compared to no use. Due to the imbalance of the dataset (ratio defective to non-defective $\approx$ 1:3), the balanced accuracy is of great importance. Very good results are also achieved here. As illustrated in Table 3, the application of the Gaussian filter resulted in a notable enhancement in the balanced accuracy of the Xception architecture, reaching 98.92%. This represents a 3.09% improvement over the model that did not employ the Gaussian filter. However, the kappa value of Xception exhibited the most significant enhancement through the application of the filter, attaining a value of 0.9853, which signifies an increase of 0.0663. It is particularly noteworthy that the VGG16 model achieves a perfect score of 100% in all metrics when using the Gaussian filter.

V. DISCUSSION

The objective of this study was to develop a model with high accuracy that can be used in the automatic quality control of synthetic leather production. To this end, the influence of the Gaussian filter on the classification performance of the architectures used was also investigated. The results, which are shown in Table 3, show that this filter has a positive effect on the accuracy of all architectures. In InceptionV3, however, the true negative rate (TNR) and positive predicted value (PPV) showed a negative effect. This may be due to the low number of images, especially in the defective classes in the dataset. The positive developments of most metrics could be due to the fact that the Gaussian filter blurs the images. As can be seen in Figure 3, thereby leather structure is generally shown in less detail. Furthermore, possible influences from shading or light are also softened, while the edges of defects are largely preserved. This fact could emphasize the defects of the leather in the models, leading to a better classification. Upon examination of the architectures without the application of the Gaussian filter, it becomes evident that the VGG models attained accuracies exceeding 99%. When the VGG models are hidden, the ResNet50V2 model demonstrates the highest accuracy value (without filter), which may serve to illustrate the efficacy of the residual skip connections. In comparison to Xception, Inception, and ResNet50V2, VGG16 and VGG19 employ a comparatively straightforward architectural approach, utilizing only convolutional filters of

TABLE 4. Overview of supervised, semi-supervised and unsupervised learning methods for defect detection on synthetic leather.

Author	Year	Category	Model	Metric (in %)
Srilakshmi et al. [22]	2022	supervised	VGG16	ACC: 94.66
Adão et al. [19]	2022	supervised	Xception	ACC: 96.00
Zuluaga et al. [21]	2023	supervised	DarkNet19	ACC: 100.00
			SqueezeNet	ACC: 100.00
Fuqin et al. [20]	2024	supervised	ResNet34	ACC: 92.60 (defective)
				ACC. 92.80 (no defect)
Li et al. [13]	2024	unsupervised	Teacher-student model	AUC: 95.24
Yang, Wu and Feng [16]	2023	semi-supervised	MemSeg	AUC: 100.00
Luo et al. [17]	2024	unsupervised	AMI-Net	AUC: 100.00
Napoletano, Piccoli and Schettini [18]	2021	semi-supervised	pretrained CNN autoencoder	AUC: 98.00
Jiang et al. [14]	2024	unsupervised	Feature-Level Registration PatchCore	AUC: 99.84
Zavrtanik, Kristan and Skočaj [15]	2021	unsupervised	DRÆM	AUC: 100.00
Our Model		supervised	VGG16	ACC: 100.00
				AUC: 100.00

size 3×3. Unlike other architectures, they do not incorporate depthwise separable convolutions, parallel convolutional layers, or residual skip connections. Only Xception is also based on the exclusive use of 3×3 convolution filters, but also uses depth-separable convolutions. In this particular case, it can be assumed that simple architectures for surfaces without a background are better suited for detecting defects than those with a more complicated structure.

In addition to (balanced) accuracy, the two metrics NPV and TNR are of great importance for defect detection. NPV evaluates the accuracy of a model in relation to the prediction of negative (defective) instances. A high value indicates the accuracy of the negative prediction. Due to the influence of the filter, this prediction accuracy could be increased to 100% (+5.36%) for Xception and also to 100% (+7.48%) for InceptionV3.

TNR indicates the efficacy of a model in correctly identifying defective leather. Low values encourage misclassification, resulting in defective leather being recognized as non-defective. This not-recognized defective leather is exported and subsequently leads to customer complaints, causing financial loss to the company. With the Gaussian filter, the TNR of models up to InceptionV3 could be enhanced as follows: Xception (+4.39%), ResNet50V2 (+2.17%), VGG19 (+1.05%) and VGG16 (+1.11%). The VGG16 model achieved a TNR of 100%, indicating that defective leather is consistently and accurately identified. While the results are impressive, it's worth mentioning that VGG16 might not yield the same results when used with different datasets or in real-world industry settings. A potential reason for this could be differences in image quality, the number of images, or the type and number of defects.

Table 4 presents the current work of using supervised learning architectures from section II. Additionally, the table includes methods of semi- and unsupervised learning methods, which are employed for comparison with the VGG16 model, among other models. It should be noted that all authors utilize the MVTec dataset.

First of all, it is noteworthy that in addition to our VGG16 model, the authors Zuluaga et al. [21] also achieved an accuracy of 100% with SqueezeNet and DarkNet19. They employed data augmentation techniques to expand the number of datasets, although the exact number of augmented datasets is not specified. However, the authors do not provide any information regarding the evaluation process of their models. In contrast, we maintained the original number of images in our dataset and utilized data augmentation solely to enhance the variation within the dataset rather than to increase its size. Furthermore, we evaluated all our models using stratified 5-fold cross-validation, which highlights the robustness and reliability of our models. This comprehensive evaluation approach ensures that our results are consistent and reliable, offering a more thorough validation of our models' performance.

Examining the performance of the remaining models from the supervised learning approach, it is evident that all the models we used in this study demonstrate superior accuracy compared to them. This superiority holds true also for models that do not incorporate the Gaussian filter. For instance, the Xception model without the Gaussian filter achieves an accuracy of 97.02%. When compared directly with the results of Adão et al. [19], who employed the same architecture, our model is 1.02% more accurate, and with the use of the Gaussian filter, the accuracy improvement extends to 3.46%. A similar pattern is observed with the VGG16 model as utilized by Srilakshmi et al. [22]. Our VGG16 model achieves a 5.07% higher accuracy than theirs, and when the Gaussian filter is applied, the accuracy increases by 5.34%, representing a substantial improvement. These findings underscore the effectiveness of our approach. The enhanced accuracy of our models, both with and without the Gaussian filter, highlights the robustness and efficacy of our methods.

Given the prevalent use of semi- and unsupervised learning methods in defect detection, we have also compared our model with these approaches. The most commonly used evaluation metric in this context is the Area Under the

Receiver Operating Characteristic Curve (AUC). A total of six studies from the field of semi-supervised and unsupervised methods were included in the table for comparison. Three of these methods achieved an AUC of 100%. Our models, ResNet50V2, VGG19, and VGG16, also achieve 100% AUC when the Gaussian filter is applied. Xception and InceptionV3 come very close, with AUC values only 0.02% and 0.04% lower, respectively. However, using AUC as the sole evaluation metric poses challenges when comparing with our model. It is important to note that an AUC value of 100% does not necessarily correspond to an accuracy of 100%, as demonstrated in Table 3 for InceptionV3 and VGG19. Despite achieving a perfect AUC, these models do not attain 100% accuracy. Our VGG16 model, on the other hand, not only achieves an AUC of 100% but also attains an accuracy of 100% and a balanced accuracy of 100%. This indicates that our model excels in perfectly classifying defective and non-defective leathers, making it comparable to unsupervised approaches. The comprehensive performance of our model across multiple metrics underscores its robustness and efficacy in defect detection, providing a more reliable measure of its capabilities compared to solely relying on AUC.

Furthermore, the VGG16 architecture offers additional advantages over semi- or unsupervised methods, though the extent of these benefits varies depending on the specific application. Supervised learning approaches, such as VGG16, can achieve high accuracy when a sufficient amount of labeled image data is available. Companies with access to extensive image datasets are therefore advised to utilize pre-trained architectures like VGG16 due to their superior accuracy. In cases where an adequate quantity of labeled image data is unavailable, this limitation can be mitigated through various techniques. Through the application of transfer learning, pre-trained weights derived from extensive and diverse datasets can be utilized, thereby significantly reducing the amount of training data required [39]. Moreover, the generation of artificial image data through advanced techniques such as Generative Adversarial Networks (GANs) further addresses the challenge of limited data availability. By creating realistic and diverse synthetic images, GANs can effectively supplement existing datasets. When combined with data augmentation strategies, which involve transformations such as rotation, zoom, and translation, this approach significantly enhances both the variety and the quantity of training data. In the context of real-time detection, VGG16 may be the more suitable choice due to its typically faster inference times. In contrast, semi- or unsupervised methods, which often rely on memory pools [14], [16] or involve additional computational processes, generally require longer classification times. These approaches also tend to demand greater memory and computational resources, necessitating advanced hardware capabilities.

In summary, it can be stated that our approach, which involved the application of transfer learning with fine-tuning, hyperparameter optimization through Bayesian optimization, and data augmentation, has already yielded positive results. The additional application of the innovative approach of using the Gaussian filter further enhanced the values for the majority of performance indicators. The robustness of the approach was evaluated through the use of 5-fold cross-validation. Overall, our approach demonstrates the advantages of the above techniques. In particular, the effectiveness of using the Gaussian filter in combination with pre-trained architectures is emphasized by this study. Moreover, no additional artificial images were generated in this study, thereby demonstrating the effectiveness of our approach on the original dataset. These achieved results represent a significant improvement in comparison to the manual quality control that is currently still in place. The transition to an image processing-based system for companies that still have a manual process seems to be a logical next step, as this would enable accelerated quality control. The high values of TNR indicate that defective leather is correctly recognized as defective. Consequently, companies export high-quality leather, which significantly reduces customer complaints and leads to cost savings. It is hoped that the results of our study will encourage other authors to test this filter on their datasets.

VI. CONCLUSION

Deep learning is a powerful tool in the field of defect detection on the surface of synthetic leather, which is rapidly gaining momentum for the quality detection of industrial products. In this work, we used the five different pre-trained models, Xception, InceptionV3, ResNet50V2, VGG19, and VGG16 to develop a powerful model that can be used for automatic quality control in synthetic leather production. All models were pre-trained on the ImageNet dataset. For reasons of robustness and reliability, each architecture was cross-validated five times. Each of these five models per architecture underwent hyperparameter tuning with data augmentation and subsequent fine tuning. We used traditional data augmentation methods such as rotation, zoom, and translation. In addition to the aforementioned methods, we also investigated the influence of the Gaussian filter on the classification performance. To this end, the aforementioned process was performed for two cases, each comprising a dataset without and with a Gaussian filter as pre-processing of the images. The results demonstrated that the application of a Gaussian filter to each of the pre-trained models resulted in enhanced performance, particularly in terms of accuracy. The integration of a Gaussian filter with a pre-trained architecture indicates its potential as a valuable tool for the synthetic leather industry, facilitating the identification of defective material. This approach can also contribute to cost savings as substandard material is better identified and rejected, resulting in fewer customer complaints. Furthermore, this reduces reputational damage.

A. LIMITATIONS

Although this work provides good results in the detection of defective and non-defective leather images, it also has its limitations, which should be taken into account. The analysis

of the results presented is limited to the evaluation metrics used. Consequently, the interpretation of why the Gaussian filter works is also based on these metrics. Grad-CAM could be a potential solution for visualizing the influence of this filter in future work, allowing a deeper understanding of the effect of the Gaussian filter.

B. FUTURE WORK

In the future, it is planned to check the results of the classification models presented here with further and larger datasets in order to increase external validity. In addition, the influence of the Gaussian filter is to be investigated in more detail, so that different standard deviations σ and kernel sizes are to be examined. Additionally, Grad-CAM will be employed to visualize the influence, with the aim of gaining a deeper understanding of the filter. Furthermore, the models will be tested for a range of defect types, extending beyond the current scope of binary classification. Moreover, the existing architecture will undergo further development into a hybrid model, as proposed by Hax et al. [61], with the objective of reliably detecting defects in synthetic leather within videos.

REFERENCES

[1] O. Omoloso, K. Mortimer, W. R. Wise, and L. Jraisat, "Sustainability research in the leather industry: A critical review of progress and opportunities for future research," *J. Cleaner Prod.*, vol. 285, Feb. 2021, Art. no. 125441.

[2] S. Iqbal, T. M. Khan, S. S. Naqvi, and G. Holmes, "MLR-net: A multi-layer residual convolutional neural network for leather defect segmentation," *Eng. Appl. Artif. Intell.*, vol. 126, Nov. 2023, Art. no. 107007.

[3] F. Wang and K. J. Kyoung, "Leather defect detection method in clothing design based on TDENet," *IEEE Access*, vol. 11, pp. 104890–104904, 2023.

[4] N. Gilon, M. Soyer, M. Redon, and P. Fauvet, "Separation of leather, synthetic leather and polymers using handheld laser-induced breakdown spectroscopy," *Sensors*, vol. 23, no. 5, p. 2648, Feb. 2023.

[5] J. Ma, K. Cai, C. Yang, M. Li, X. Pan, Y. Huang, J. Yao, J. Zheng, and J. Shao, "Synthesis and properties of photocurable polyurethane acrylate for textile artificial leather," *Prog. Organic Coatings*, vol. 171, Oct. 2022, Art. no. 107017.

[6] H. Wang, "Advantages of animal leather over alternatives and its medical applications," *Eur. Polym. J.*, vol. 214, Jun. 2024, Art. no. 113153.

[7] T. B. Sudha, P. Thanikaivelan, K. P. Aaron, K. Krishnaraj, and B. Chandrasekaran, "Comfort, chemical, mechanical, and structural properties of natural and synthetic leathers used for apparel," *J. Appl. Polym. Sci.*, vol. 114, no. 3, pp. 1761–1767, Nov. 2009.

[8] A. M. Neiva and E. R. Pereira-Filho, "Evaluation of the chemical composition of synthetic leather using spectroscopy techniques," *Appl. Spectrosc.*, vol. 72, no. 6, pp. 921–932, Jun. 2018.

[9] D. Gurera and B. Bhushan, "Fabrication of bioinspired superliquiphobic synthetic leather with self-cleaning and low adhesion," *Colloids Surf. A, Physicochemical Eng. Aspects*, vol. 545, pp. 130–137, May 2018.

[10] Y. Tian, J. Wang, S. Zheng, X. He, and X. Liu, "Research on the preparation and application of synthetic leather from coffee grounds for sustainable development," *Sustainability*, vol. 14, no. 21, p. 13971, Oct. 2022.

[11] X. Zhu, Q. Li, L. Wang, W. Wang, S. Liu, C. Wang, Z. Xu, L. Liu, and X. Qian, "Current advances of polyurethane/graphene composites and its prospects in synthetic leather: A review," *Eur. Polym. J.*, vol. 161, Dec. 2021, Art. no. 110837.

[12] J. Lin, L. Pan, C. Lin, Z. Chen, X. Ye, and L. Chen, "Design of crimping on-line detection system for wet coating of synthetic leather," in *Proc. IEEE 5th Inf. Technol. Mechatronics Eng. Conf. (ITOEC)*, Jun. 2020, pp. 448–452.

[13] Z. Li, Y. Ge, X. Yue, and L. Meng, "MCAD: Multi-classification anomaly detection with relational knowledge distillation," *Neural Comput. Appl.*, vol. 36, no. 23, pp. 14543–14557, Aug. 2024.

[14] Z. Jiang, Y. Zhang, Y. Wang, J. Li, and X. Gao, "FR-PatchCore: An industrial anomaly detection method for improving generalization," *Sensors*, vol. 24, no. 5, p. 1368, Feb. 2024.

[15] V. Zavrtanik, M. Kristan, and D. Skocaj, "DRÆM—A discriminatively trained reconstruction embedding for surface anomaly detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8310–8319.

[16] M. Yang, P. Wu, and H. Feng, "MemSeg: A semi-supervised method for image surface defect detection using differences and commonalities," *Eng. Appl. Artif. Intell.*, vol. 119, Mar. 2023, Art. no. 105835.

[17] W. Luo, H. Yao, W. Yu, and Z. Li, "AMI-net: Adaptive mask inpainting network for industrial anomaly detection and localization," *IEEE Trans. Autom. Sci. Eng.*, early access, Feb. 26, 2024, doi: 10.1109/TASE.2024.3368142.

[18] P. Napoletano, F. Piccoli, and R. Schettini, "Semi-supervised anomaly detection for visual quality inspection," *Exp. Syst. Appl.*, vol. 183, Nov. 2021, Art. no. 115275.

[19] T. Adão, D. Gonzalez, Y. C. Castilla, J. Pérez, S. Shahrabadi, N. Sousa, M. Guevara, and L. G. Magalhães, "Using deep learning to detect the presence/absence of defects on leather: On the way to build an industry-driven approach," *J. Phys., Conf.*, vol. 2224, no. 1, Apr. 2022, Art. no. 012009.

[20] F. Deng, J. Luo, L. Fu, Y. Huang, J. Chen, N. Li, J. Zhong, and T. L. Lam, "DG2GAN: Improving defect recognition performance with generated defect image sample," *Sci. Rep.*, vol. 14, no. 1, p. 14787, Jun. 2024.

[21] E. Zuluaga, S. Jaziri, A. Tellaeche, and I. Pastor-López, "A practical approach on performance assessment of federated learning algorithms for defect detection in industrial applications," *IEEE Access*, vol. 11, pp. 116581–116593, 2023.

[22] V. Srilakshmi, G. U. Kiran, Y. Guntupalli, C. N. Gayathri, and A. S. Raju, "Automatic visual inspection–defects detection using CNN," in *Proc. 6th Int. Conf. Electron., Commun. Aerosp. Technol.*, Dec. 2022, pp. 584–589.

[23] M. Rudolph, B. Wandt, and B. Rosenhahn, "Same same but DifferNet: Semi-supervised defect detection with normalizing flows," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 1906–1915.

[24] T. Defard, A. Setkov, A. Loesch, and R. Audigier, "PaDiM: A patch distribution modeling framework for anomaly detection and localization," in *Pattern Recognition. ICPR International Workshops and Challenges*, A. Del Bimbo, R. Cucchiara, S. Sclaroff, G. M. Farinella, T. Mei, M. Bertini, H. J. Escalante, and R. Vezzani, Eds., Cham, Switzerland: Springer, 2021, pp. 475–489.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[26] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD—A comprehensive real-world dataset for unsupervised anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9584–9592.

[27] Y. S. Gan, W.-C. Yau, S.-T. Liong, and C.-C. Chen, "Automated classification system for tick-bite defect on leather," *Math. Problems Eng.*, vol. 2022, pp. 1–12, Feb. 2022.

[28] Y. S. Gan, S.-S. Chee, Y.-C. Huang, S.-T. Liong, and W.-C. Yau, "Automated leather defect inspection using statistical approach on image intensity," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 10, pp. 9269–9285, Oct. 2021.

[29] A. Saberironaghi, J. Ren, and M. El-Gindy, "Defect detection methods for industrial products using deep learning techniques: A review," *Algorithms*, vol. 16, no. 2, p. 95, Feb. 2023.

[30] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[31] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems*, vol. 2, D. Touretzky, Ed., San Francisco, CA, USA: Morgan Kaufmann, 1989, pp. 396–404.

[32] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022.

[33] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, Oct. 2017.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25. Red Hook, NY, USA: Curran Associates, 2012, pp. 1–9.
[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, Jan. 2015, pp. 1–14.
[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[38] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021.
[39] X. Li, Y. Grandvalet, and F. Davoine, "Explicit inductive bias for transfer learning with convolutional networks," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80, Jan. 2018, pp. 2825–2834.
[40] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2009.
[41] K. R. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, no. 1, pp. 1–40, May 2016.
[42] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.
[43] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.
[44] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, p. 60, Dec. 2019.
[45] F. Ding, Y. Shi, G. Zhu, and Y.-Q. Shi, "Real-time estimation for the parameters of Gaussian filtering via deep learning," *J. Real-Time Image Process.*, vol. 17, no. 1, pp. 17–27, Feb. 2020.
[46] A. Ravishankar, S. Anusha, H. K. Akshatha, A. Raj, S. Jahnavi, and J. Madhura, "A survey on noise reduction techniques in medical images," in *Proc. Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA)*, vol. 1, Apr. 2017, pp. 385–389.
[47] X. He, H. Li, Z. Huang, X. Xiong, Y. Liu, and Y. Yao, "Multi-scale feature fusion extraction structure for the leather defect detection algorithm," in *Proc. IEEE 6th Int. Conf. Pattern Recognit. Artif. Intell. (PRAI)*, Aug. 2023, pp. 254–259.
[48] N. Veni and J. Manjula, "Gaussian denoising by time domain and frequency domain filters for MRI brain images," in *Proc. IEEE IAS Global Conf. Emerg. Technol. (GlobConET)*, May 2022, pp. 817–821.
[49] (Apr. 2014). *The OpenCV Reference Manual*. OpenCV. [Online]. Available: http://docs.opencv.org/
[50] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Jan. 2011.
[51] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, and L. Invernizzi. (2019). *Kerastuner*. [Online]. Available: https://github.com/keras-team/keras-tuner
[52] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manage.*, vol. 45, no. 4, pp. 427–437, Jul. 2009.
[53] Q. Wei and R. L. Dunbrack, "The role of balanced training and testing data sets for binary classifiers in bioinformatics," *PLoS ONE*, vol. 8, no. 7, Jul. 2013, Art. no. e67863.
[54] D. Chicco, N. Tötsch, and G. Jurman, "The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation," *BioData Mining*, vol. 14, no. 1, p. 13, Feb. 2021.
[55] C. Sharpe, T. Wiest, P. Wang, and C. C. Seepersad, "A comparative evaluation of supervised machine learning classification techniques for engineering design applications," *J. Mech. Design*, vol. 141, no. 12, Dec. 2019, Art. no. 121404.
[56] S. A. Hicks, I. Strümke, V. Thambawita, M. Hammou, M. A. Riegler, P. Halvorsen, and S. Parasa, "On evaluation metrics for medical applications of artificial intelligence," *Sci. Rep.*, vol. 12, no. 1, p. 5979, Apr. 2022.
[57] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, Apr. 1960.
[58] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, p. 6, Dec. 2020.
[59] O. Rainio, J. Teuho, and R. Klén, "Evaluation metrics and statistical tests for machine learning," *Sci. Rep.*, vol. 14, no. 1, p. 6086, Mar. 2024.
[60] T. Fawcett, "ROC graphs: Notes and practical considerations for researchers," *Mach. Learn.*, vol. 31, no. 1, pp. 1–38, 2004.
[61] D. R. T. Hax, P. Penava, S. Krodel, L. Razova, and R. Buettner, "A novel hybrid deep learning architecture for dynamic hand gesture recognition," *IEEE Access*, vol. 12, pp. 28761–28774, 2024.

CHRISTOPHER MAI received the B.S. and M.S. degrees in mechanical engineering from Brandenburg University of Technology Cottbus-Senftenberg, Germany, in 2019 and 2023, respectively. He is currently pursuing the Ph.D. degree with the Chair of Hybrid Intelligence, Helmut-Schmidt-University/University of the Federal Armed Forces Hamburg, Hamburg. His research interest includes machine learning and deep learning.

PASCAL PENAVA (Member, IEEE) received the B.S. degree in information systems from Friedrich Alexander University, Erlangen-Nuremberg, Germany, in 2021, and the M.S. degree in digitalization and entrepreneurship from Bayreuth University, Bayreuth, Germany, in 2023. He is currently pursuing the Ph.D. degree with the Chair of Hybrid Intelligence, Helmut-Schmidt-University/University of the Federal Armed Forces Hamburg, Hamburg. His research interests include the development of EEG-based BCIs and machine learning based analyses of time-series data.

RICARDO BUETTNER (Senior Member, IEEE) received the Dipl.-Inf. degree in computer science and the Dipl.-Wirtsch.-Ing. degree in industrial engineering and management from the Technical University of Ilmenau, Germany, the Dipl.-Kfm. degree in business administration from the University of Hagen, Germany, the Ph.D. degree in information systems from the University of Hohenheim, Germany, and the Habilitation (venia legendi) degree in information systems from the University of Trier, Germany. He is currently a Chaired Professor of Hybrid Intelligence with Helmut-Schmidt-University/University of the Federal Armed Forces Hamburg, Germany. He has published over 150 peer-reviewed articles, including articles in *Electronic Markets*, *AIS Transactions on Human–Computer Interaction*, *Personality and Individual Differences*, *European Journal of Psychological Assessment*, *PLOS ONE*, and IEEE Access. He has received 18 international best paper, the best reviewer, and service awards and award nominations, including the Best Paper Awards by *AIS Transactions on Human–Computer Interaction*, *Electronic Markets*, and HICSS, for his work.