Electronic Letters on Computer Vision and Image Analysis 21(1):59-77, 2022

Material Classification with a Transfer Learning based Deep Model on an imbalanced Dataset using an epochal Deming-Cycle-Methodology

Marco Klaiber* and Sophia Kaerger* and Yannik Minsch*

* *Aalen University, Research Center of Data Science, Beethovenstraße 1, 73430 Aalen, Germany*
Received 5th of October, 2021; accepted 10th of April 2022

Abstract

This work demonstrates that a transfer learning-based deep learning model can perform unambiguous classification based on microscopic images of material surfaces with a high degree of accuracy. A transfer learning-enhanced deep learning model was successfully used in combination with an innovative approach for eliminating noisy data based on automatic selection using pixel sum values, which was refined over different epochs to develop and evaluate an effective model for classifying microscopy images. The deep learning model evaluated achieved 91.54% accuracy with the dataset used and set new standards with the method applied. In addition, care was taken to achieve a balance between accuracy and robustness with respect to the model. Based on this scientific report, a means of identifying microscopy images could evolve to support material identification, suggesting a potential application in the domain of materials science and engineering.

Keywords: Material Identification, CNN, Transfer Learning, Deep Learning, Material Science

1 Introduction

Daily human interaction with novel scenes or objects relies on the perception and identification of material properties of surfaces and objects, resulting in the enormous importance of correct identification [1]. The domain of material identification represents a broad spectrum, ranging from identification in macroscopy to nano- and microscopy, recognizing the materials that make up our environment is elemental to any interaction, whether human or machine [2, 3]. In recent years, materials scientists have been able to make immense progress in the acquisition, analysis, and comparison of structural images, proving the father of nano- and microtechnology Feynman (1960) correct in his statement "There's Plenty of Room at the Bottom" and allowing the technology to advance deep into the microscales [4, 5]. Detection of materials is proving to be enormously beneficial in macroscopy, e.g., robotics, waste separation, or autonomous driving, while it is important in microscopy to detect microcracks or minute defects, as those are inextricably linked to material properties [2, 3, 6–8]. Especially microtechnology and the ever-improving recognition of minute structures within various materials is defined

Correspondence to: <marco.klaiber@studmail.htw-aalen.de>

Recommended for acceptance by Luis M. Goncalves
Ehttps://doi.org/10.5565/rev/elcvia.1517
ELCVIA ISSN: 1577-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

as the hope of the economy in the twenty-first century as it changes human lifestyle and industrial structure, so that nano- and microtechnology is also considered as the fourth industrial revolution [5, 9].

In the addressed domain of microtechnology, a materials scientist must pay attention to an abundance of minute details when viewing microscopic images of materials to achieve optimal identification [3, 8]. This is strongly complicated by certain shape variations, phase boundaries, or a wide range of appearances that can also vary by illumination and shape [3, 8]. Fault diagnosis requires intensive prior knowledge on the part of an expert, which can also lead to misinterpretation, since some forms of faults are not visible to the human eye [10]. Therefore, the detection and diagnosis of materials is a complex and challenging task that is currently performed by humans, leading to misinterpretation and preventing correct classification as well as diagnosis [10, 11]. These issues of possible misinterpretation or non-identification can lead to incorrect use, addressing, or interaction with the materials, resulting in monetary losses on the one hand, but also posing risks to the users and environments in the particular domain on the other hand [12, 13]. In addition, the attention span of humans is limited and explicitly in repetitive tasks this attention decreases rapidly, which can also lead to serious errors [14, 15]. There is also the risk of identifying materials that may cause harm to human health without any indication of potential risk from the respective material [14].

The interpretation of microscopic material image data is subject to the "intuition" of experienced researchers, which often leaves many of the deep graphical features unused because of difficulties in processing the data and finding the correlations, that can be optimally addressed by deep learning (DL) in particular [16, 17]. In the domain of materials science, the guiding ideology that can be summarized into four paradigms prevails: The first paradigm defines the empirical trial-and-error method, the second paradigm embraces physical and chemical laws, the third paradigm includes computer simulation, and the fourth paradigm is Big Data-driven materials science [9]. Explicitly, the fourth paradigm can perfectly unite the other three paradigms in the aspects of theory, experiment, and computer simulation through the continuous development of data mining technology and artificial intelligence through machine learning (ML) in closer consideration of DL [9, 16]. DL is a way to enable an automatic optical identification and/or characterization system for materials, requiring an algorithm that is reliable for different materials, operates with high accuracy, is fast enough for real-time processing, and can be easily adapted to different optical setups and different user requirements with minimal additional human effort [17]. Through DL capabilities, generic microstructural signatures can be used to automatically find relationships in large and diverse microstructural image datasets [4, 16]. This application form of DL falls into the rapidly developing domain of computer vision and offers a variety of techniques that have shown tremendously promising performance in practice, where a trained network was able to extract deep graphical features such as contrast, edges, shapes, flake sizes and their distributions, based on which key physical properties of materials were identified [4, 17].

Training a deep neural network from scratch is often not feasible because it requires a dataset of sufficient size, and it may take a long time for convergence to be achieved than to make the experiments worthwhile. Even if a large enough dataset is available and convergence does not take that long, it is often helpful to start with pre-trained weights instead of randomly initialized weights [18]. Transfer learning (TL) is often beneficial when a limited amount of annotated data is available, where manual annotations are time-consuming and require a high level of expertise. The usefulness of TL is due in part to the fact that the first convolutional neural network (CNN) layers capture low-level features, that are often used between different types of images such as edges and blobs. When there are only a small number of labeled examples, TL can help to reduce the error in the test set through a regularization effect [19, 20].

This report is structured as follows: We begin by providing an overview of the research background in the field of materials science and related work. We then describe the methodology, including the integration of the epochal Deming-cycle in the ML context, the dataset used for the research, and the preprocessing applied. We then show the results of the method we implemented using an evaluated final model. Finally, the results are discussed and brought to the scientific context.

Related Work

In order to present the current state of material identification in the domain of ML and especially DL, relevant work and its results are presented below to the best of our knowledge. DeCost and Holm [4] used a novel support vector machine (SVM) in the domain of microstructure science of computer vision conceptualization with accuracy greater than 80% at 5-fold cross-validation. Based on the knowledge gained from the previous work, DeCost et al. [7] presented further SVMs supported by two image representation approaches, one is CNN and the other is a middle image feature based on Bag of Visual Words (BoW). Cross-domain deep CNN representations were used to achieve classification accuracy and thus material identification of more than 95%. Xia et al. [13], on the other hand, focused on the extraction of road information, with material identification elementary to the interaction with the conditions, designing a two-level structure to both extract the road surface and classify materials. A deep CNN was applied, and by using a post-processing approach, it was more consistent with that of real-world situations and achieved good results [13]. Whereas Kerr et al. [11] developed a system that outperforms humans in the task of material recognition by analysing thermal conductivity and surface texture using SVM and ANN. In addition, Bircanoglu et al. [6] evolved a CNN for waste sorting and achieved an accuracy of 75% by training from scratch to even 95% accuracy with TL and fine tuning. Similar image structures to our work were analyzed in the work of Kandel et al. [21] analyzed with CNNs, including a detailed comparison of optimizers, and the findings have also found resonance in this work. The dataset was based on histopathology images, revealing microscopic manifestations of disease, placing the work in the microscopic domain as well, and the outcome was measured using AUC, with ResNet achieving 93.84%. Another deep-learning-oriented approach was taken by Wei et al. [22] to determine isotropy from images of architectural materials. The efficiency of the algorithm was further enhanced by the use of TL and achieved 90% accuracy, providing an effective approach for mechanical design detection.

Methodology

For our research, we resorted to proven frameworks and packages in the domain of DL. We used the programming environment of Google Colab in the Pro variant to have a sufficiently strong computing power, namely a Tesla 100 GPU including 27 GB RAM, as well as to use the extensive library possibilities of Python [23]. As a basis, we used TensorFlow, which is a framework for sequential programming and is very popular in the domain of ML [24]. For a better understanding of the classification of our models, the model-independent LIME [25] was additionally applied, ensuring local model interpretability.

The goal of this work was to develop an effective model for the classification of microscopic images, which can be an efficient tool for real-world applications. Our methodology has been to compare different model and data processing approaches over recurring phases, where in the course the most effective model in terms of predictive performance should emerge. Within and across these phases, the respective models should be continuously improved, which is why we adapt the plan-do-check-act cycle according to Deming within the research [26, 27]. This is also known under the name Deming-cycle and represents a methodology from the management domain for the control and continuous improvement of processes [27]. This approach is transferable to the evaluation of our models, where the focus is also on the continuous improvement of process steps within the phases that are directly related to the desired predictive performance. The cycle will finally be terminated by the final model, which will determine the final result.

We have compared different TL approaches with different base models that bring the advantage of pre-trained weights. This allows us not to develop the model from scratch, but to work with the knowledge of established models and add the final layers that should perform the final classification. In the first respect, we have chosen state-of-the-art architectures from different years of the ImageNet large scale visual recognition challenge (ILSVRC) for the comparisons, namely VGG16 [28], Xception [29], EfficientNetB0 [30], ResNet50V2 [31], and InceptionV3 [32], which have shown exceptional performance and popularity in the literature. In addition, a custom model architecture, named APANnet, was also designed to ensure the effectiveness of TL, training

the model from scratch. In the following of this paper, the individual steps within our research are described in chronological detail to give the reader the best possible insight. This work was developed following the methodology of the design science research approach of Hevner et al. [33], which aims to improve problem-solving capabilities by creating innovative artifacts, which in this work argues for the ML algorithm according to the developed guidelines. The methodology evaluated and finally used is shown in figure 1, which builds sequentially on each other.

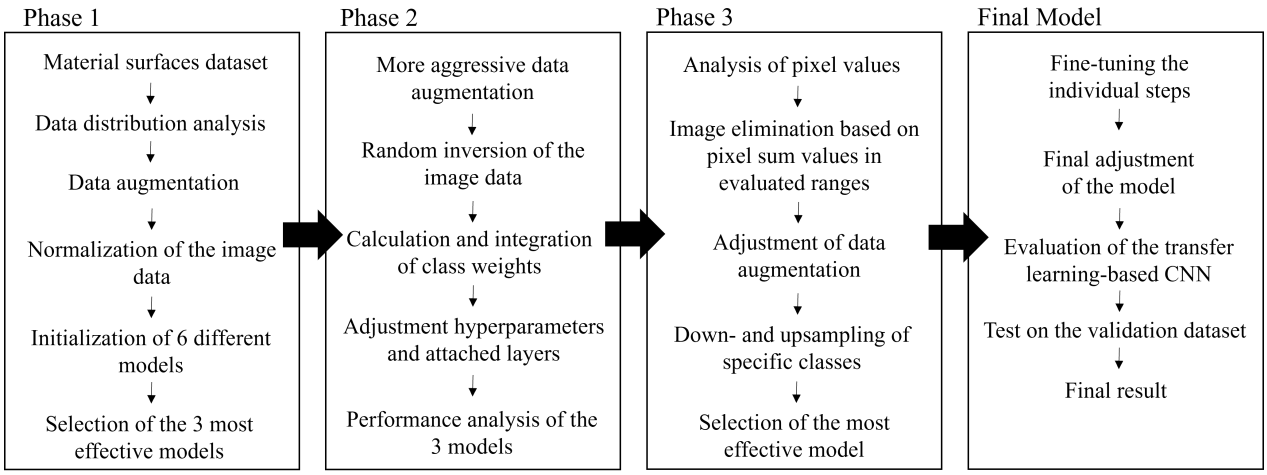

Figure 1: Overview of the methodology used

Dataset

In this work, we used a self-collected dataset consisting of a total of approximately 306,000 microscopic material surface images that have a size of 300 × 300 pixels. These are assigned to 15 different classes, with 12% of the total retained for the validation dataset for each class. In advance, one class was removed, which was composed of four other classes, which would negatively affect the classification of the model, since a clear assignment would no longer be possible. The individual classes include microscopic images from different domains, such as bainite, which represents a microstructure that can form during heat treatment of carbon steel, or AlSi9Cu3, a cast aluminum alloy with good strength properties, such as high thermal conductivity and chemical resistance [34, 35]. A complete listing of all classes, including their explanations and the available sets of training and validation samples, is shown in table 1.

Table 1: Explanation of the data set including the respective number of training and test data

Class	Explanation	Train	Validation
AlSi9Cu3	Cast aluminium alloy	15,056	1,883
Bainite	Crystalline microstructure within steel	12,191	1,525
Boron Nitride	Material that consists of boron and nitrogen covalent bonds	12,284	1,537
Fabric	Not further specified	44,236	5,531
Films Coated Surface	Not further specified	1,454	183
Graphene	Carbon allotrope where the carbon atoms are arranged in a two-dimensional honeycomb lattice	5,201	651
LiB	Lithium-ion battery	68,792	8,600
Magnet	Metal with magnetic field	6,955	870
Molybdenum Disulfide	Compound consisting of molybdenum and sulfur	16,146	2,019
Non-Crimp Fabric	Basalt fibres	3,640	456
Palladium	Palladium nanoparticles on carbon surface	4,780	599
Particles	Not further specified	18,864	2,358
Patterned Surface	Not further specified	17,467	2,184
Powder	Not further specified	6,328	792
Tungsten Ditelluride	Inorganic chemical compound formed by the reaction of tungsten with tellurium	38,581	4,824

After the initial manual review of the data, it was recognized that the Graphene, Molybdenum Disulfide, Boron Nitride, and Tungsten Telluride classes had largely identical features to the human eye, as did images containing only green coloration with no other discernible patterns. The green coloration is due to a silicon dioxide (SiO_2) matrix in which the materials were embedded to identify certain elements and phases on microscopy images. Depending on the thickness of the layers of the embedded material, the color changes, indicating a similar oxide thickness of the addressed classes. These immense similarities crystallized early on as the main challenge in classification, which will be discussed in detail later in the work through the approaches chosen to solve it. A comparison of the classes addressed and their visual discrimination issues is shown in figure 2.

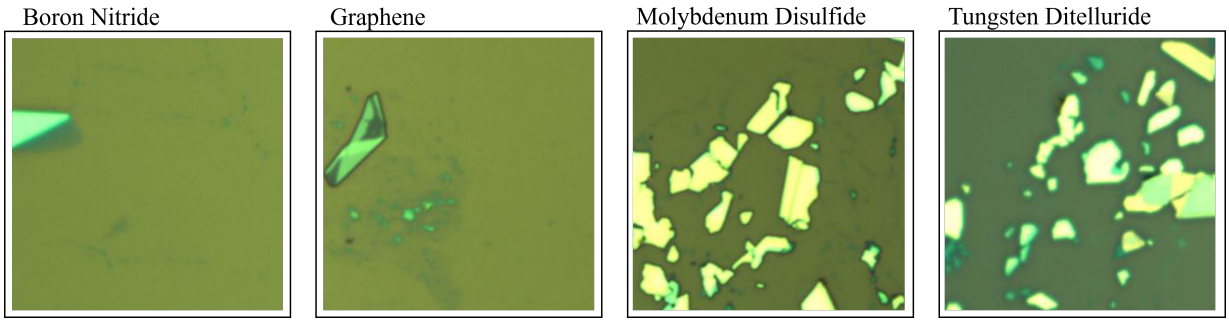

Figure 2: Comparison of the visual similarities of the four addressed classes

Preprocessing

For the preprocessing, it was primarily important to be able to optimally estimate the training data, which was approached by analyzing the distribution of the classes to provide an effective basis for the models. The result of this consideration was the distribution in figure 3, which represents an unbalanced dataset. This distribution was intentionally used for the baseline results to test self-regulation and behavior in the presence of under- and over-representation of classes.

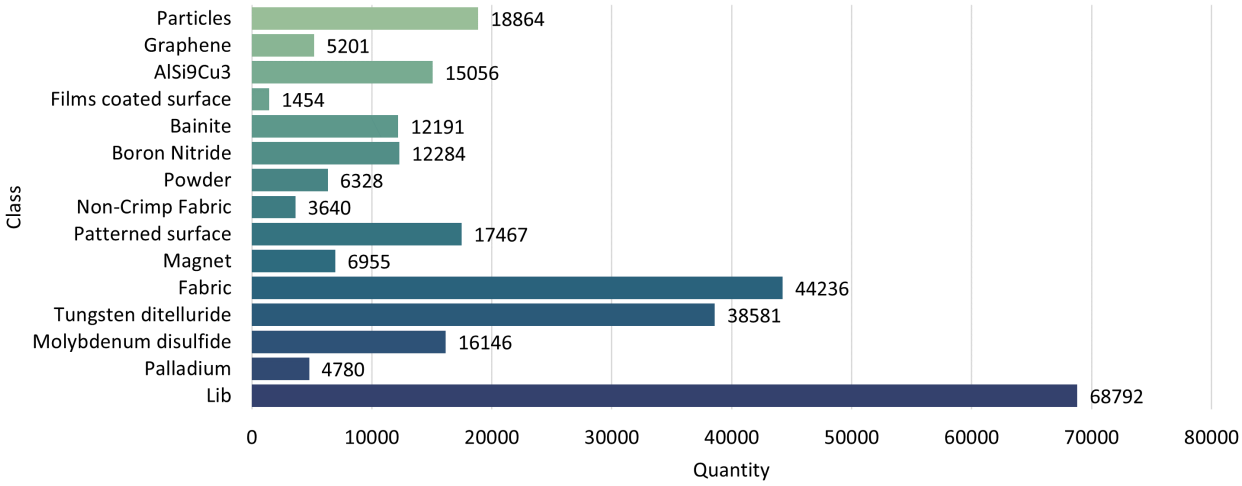

Figure 3: Number of data from the training dataset per class

A key component for training with image data is data augmentation, which expands the existing dataset within the training by adding modified copies of the original data [36]. Added to this is the benefit that augmentation thereby regulates overfitting and increases the amount of data available for training, allowing the model to recognize more specific patterns [36]. Explicit overfitting is a fundamental problem in the domain of DL, where the model learns a very high variance function to perfectly model the training data, but has a significantly weak generalization capability and would not work in a real-world environment [36]. In addition to the size of the dataset, data augmentation promotes quality, using a number of techniques, such as geometric transformations, colour space augmentations, kernel filters, shuffling images, or randomly deleting sections [36]. In all models, we used the augmentation layers of the Keras API [37], defining the settings per phase identically for each to ensure comparability. Over the time of the research, the following settings emerged as most effective for our final model in terms of predictive performance, as well as training efficiency: random rotation (factor = 0.4), random translation (height factor = 0.4; width factor = 0.7), random horizontal and verti-

cal flip, as well as random contrast (factor = 0.5), and random zoom (height factor = 0.4), with all values being experimentally justifiable in terms of effectiveness across the phases. Another important aspect for training our models in terms of predictive performance, but also training speed, was the normalization of the data before it is passed into training. In this case, the goal was to have the input values centered in a distribution around 0 with a standard deviation of 1 by calling the mean and variance of the data pre-computed at runtime, which is a efficient statistical approach [37, 38]. Explicitly, for our image data, this means that those were normalized from (0, 255) to a range (-1,+1).

Phase 1: First comparison of the models

The goal of the first phase was to compare the selected five TL approaches, as well as the APANnet, and use the first results to select the three best architectures for further research. The APANnet consisted of three convolutional layers and subsequent max pooling layers. In addition, several dense layers and a flat layer were integrated. Due to the fact that we are using an unbalanced dataset, balanced accuracy is used as the primary comparison metric for the evaluation, although other common metrics are not ignored, which is maintained within the research. To this end, simple accuracy was intentionally captured in the paper, despite the presence of an unbalanced data set, in order to show the difference towards balanced accuracy. The training procedure for each model was an initial training over the attached layers to the base model, and a subsequent fine-tuning, again only over the attached layers over 5 epochs each. Here we used the premise that the base models have sufficient prior knowledge and therefore our added layers should reach this level of knowledge. The results of each model are described in table 2.

Table 2: Results of phase 1

	VGG16	Xception	EfficientNetB0	ResNet50V2	InceptionV3	APANnet
Accuracy	**74.1%**	73.0%	73.1%	73.1%	70.7%	65.8%
Bal. Accuracy	**83.4%**	81.9%	83.2%	81.5%	78.8%	71.8%
Kappa	**71.3%**	70.0%	70.1%	70.1%	67.5%	62.3%
Precision	**71.5%**	77.0%	75.7%	74.9%	73.4%	69.4%
Recall	**74.1%**	73.0%	73.1%	73.1%	70.7%	65.8%

The VGG16-based TL approach resulted as the current most effective architecture with 83.4%, followed by the EfficientNetB0 with 83.2%. The analysis of the training showed that the ResNetV50 has more optimization potential compared to the Xception model, why we shortlisted the ResNetV50 as the third architecture and will thus no longer consider the Xception, the InceptionV3, and the in-house APANnet for the further course based on the results. As suspected, the main problem across all models was the comparatively weak correct assignment of the Graphene, Molybdenum Disulfide, Boron Nitride, and Tungsten Telluride classes, which will be addressed primarily in the following steps. In addition, the unbalanced distribution of classes across all classes could degrade performance, which should also be fixed and would highlight a common problem of DL with imbalance.

Phase 2: Optimization of the 3 best models

The second phase focused on solving the challenge of the four posed classes, which was addressed through more aggressive augmentation. The hypothesis was that further transformations would highlight the differences within the classes, meaning that larger parameters were defined for each method, for example expanding the horizontal flip to a vertical and horizontal flip. Additionally, a random inversion was defined in the augmentation, which is a pixel operation that determines the opposite color of the respective color space, thus inverting the colors [39]. The hypothesis was that the inversion could cause more patterns to stand out, which could help to distinguish them better.

To compensate for the unbalanced distribution of each class, class weights were included to computationally balance the dataset, which has led to successful results in other work and is a popular method for improving predictive performance in these cases [40, 41]. To calculate the appropriate class weights, starting from the class with the most representation relative to each of the other classes, the relation was calculated that those have the same level and thus each class is assigned the same representation for training. In addition, our approach in this phase was to test the models with the base models VGG16, EfficientNetB0, and ResNetV50 for various hyperparameters, as well as to further specify their attached layers. Primarily addressed were the optimizer, the batch size and the learning rate (in pre-training and fine-tuning). Therefore, that the optimizer has a significant impact on the result, SGD [42], Adamax [43], RMSprop [44], and FTRL [45] models were considered for comparison. For the training, 5 epochs each were used in the pre-training and fine-tuning, which led to the results in table 3.

Table 3: Results of phase 2

	VGG16	EfficientNetB0	ResNet50V2
Accuracy	**85.7%**	85.3%	75.2%
Bal. Accuracy	**84.2%**	81.4%	81.8%
Kappa	**83.6%**	83.0%	72.5%
Precision	**83.1%**	85.1%	78.6%
Recall	**85.7%**	85.3%	75.2%
AUC	**99.1%**	98.9%	97.6%
Optimizer	**Adamax**	Adam	Adam
Learning Rate	**0.001**	0.001	0.001
Batch Size	**128**	128	64
Augmentation	**augm. & invert**	augm.	augm.
Class Weights	**calculated**	calculated	calculated

The individual changes significantly improved the performance of all models compared to the first phase, with the VGG16-based model performing best with a balanced accuracy of 84.2%. Derived from the training process, the models still offer potential for improvement, which will be exploited in the upcoming phase. Comparing the optimizers, here using the VGG16 based model, Adamax turned out to be the most effective. However, Adam also outperformed the other models with a balanced accuracy of 82.99%. In comparison, RMSprop with 81.54%, and SGD with 80.81% remained significantly behind, whereas FTRL performed enormously weak. This pattern was largely detected in the other models, so Adam and Adamax were used for the next phase. The challenging nature of the Graphene, Molybdenum Disulfide, Boron Nitride, and Tungsten Telluride classes could be partially improved, yet it became clear that harder sample eliminations may be a likely solution to achieve better performance. Inversion could not bring the desired effect, which is why this issue should be reconsidered. In addition, LiB and Fabric (not further specified) are perfectly assigned, involving an enormous amount of training data, which does not seem to be necessary to this extent due to the very good differentiability. Due to this, a decimation of the two classes is considered, which should effectively reduce the training time and could improve the classification performance of the four challenging classes, as more emphasis could be put on those.

Phase 3: Evaluation of a final model

To solve the challenge of the four classes that are difficult to classify, a machine elimination of the overlapping images and the predominantly green colored content was performed. For this purpose, the property of images was used, with the help of the fact that each image consists entirely of pixels. These pixels contain color information that varies in value depending on the coloring and allows the visible properties to be uniquely

determined using numerical values. The upstream definition of each image to the size of 224 x 224 ensured that each image contains a total of 50,176 pixels in sum. For this purpose, each image was provided as an RGB (red, green, blue) model so that the color depth, which represents the number of bits in a pixel, is equal to 24 bits evenly distributed among the red, green, and blue components, resulting in 8 bits per color component. This results in the calculation per image according to the respective color intensity: height x width x channel x color depth. Based on this, the sum of all pixel values can be used to calculate whether an image is almost primarily green and thus likely for potential misclassification due to the highly similar appearance within all four classes. These pixel sum values can vary due to slight color changes in the image or barely visible fragments. Therefore, thresholds (upper and lower limits) were defined through manual observations to remove the cross-class images for the addressed classes from the dataset once the pixel sum values were within the domain. This self-evaluated innovative method allows for the cleaning of data that could eventually potentially belong to any of the four classes and bias a result, decimating the dataset, but still providing sufficient training data through data augmentation. It is important to mention that only images were removed from the training dataset and the predefined validation dataset remains unchanged. This is important because in practice challenging similar images can occur, which makes it important that for the validation the model learns to classify such images. This procedure can reduce the final accuracy, which is nevertheless consciously accepted in order to create a more robust model for the application. This trade-off between robustness and accuracy is a well-known phenomenon for DL, where we aim to balance these properties [46, 47].

In addition the LiB and Fabric classes were decimated by approximately 58% and 45%, respectively, randomly deleting images to bring the sum of images to a similar level. Compared to this downsampling measure, the images of Graphene were quadrupled and Films Coated Surface doubled, using duplications of the existing image data, with a 180°-rotation and a vertical and horizontal flip. The increased sample number was intended to increase training effectiveness and improve the balance of the dataset, which was used alongside the class weights to increase their effectiveness as well. A dynamic learning rate through an exponential decay schedule was considered as another optimization option. This is based on the consideration that in training it may be useful to reduce the learning rate with increasing duration by an exponential decay function on the optimization steps, where initial and final learning rates were determined [48]. At this stage of the research, the training was extended and 10 epochs were chosen for pre-training and 5 epochs were chosen for fine-tuning, which led to the results in table 4.

Table 4: Results of phase 3

	VGG16	EfficientNetB0	ResNet50V2
Accuracy	85.0%	85.8%	**85.7%**
Bal. Accuracy	89.7%	89.8%	**91.5%**
Kappa	83.0%	83.8%	**83.8%**
Precision	91.1%	88.4%	**91.6%**
Recall	85.0%	85.8%	**85.7%**
AUC	99.0%	99.0%	**99.1%**
Optimizer	Adam	Adamax	**Adamax**
Learning Rate	0.001	0.001	**0.001**
Batch Size	128	64	**64**
Augmentation	augm.	augm.	**augm.**
Class Weights	calculated	calculated	**calculated**
Pixel-Value	used	used	**used**

The changes regarding the green image-classes resulted in a significant increase in accuracy, with the ResNetV50 providing the best results and not the VGG16 as before. A balanced accuracy of 91.5% was

achieved with the displayed hyperparameters. It is important to mention that the augmentation without an inversion gave better results for all models, which is why this methodology was discarded and clearly outperformed by the elimination based on the pixel sum values. Despite the improved values, we believe there is still potential to improve predictive performance, with the help of further adjusted and evaluated pixel sum ranges, as well as partial adjustment of hyperparamaters, which will be tired into our final model for this research. The use of dynamic learning rates across training has also been able to perform well, yet a learning rate of 0.001 for each model has produced the best results. In addition, LIME was integrated to analyze the relevant domains for the classification of the four problem classes and to obtain a model interpretability [25]. The addressed classes were sequentially evaluated which generally showed a similar pattern with respect to the mainly green images. The domains for the classification of the classes are shown in green and the domains that speak against the class are shown in red. As shown in figure 4, the model, in this case the VGG16-based, did not use the pattern recognizable to humans for the classification, but draws the information from other regions, which is a surprising finding. Predominantly, the right domain of the images was defined as contra, which is an interesting discovery and is addressed with a centered vertical reflection in the augmentation for the final model. For the resulting ResNetV50-based model, the individual methodological steps were finally fine-tuned. It was significantly important to use the gained knowledge of the research to be able to further improve the result.

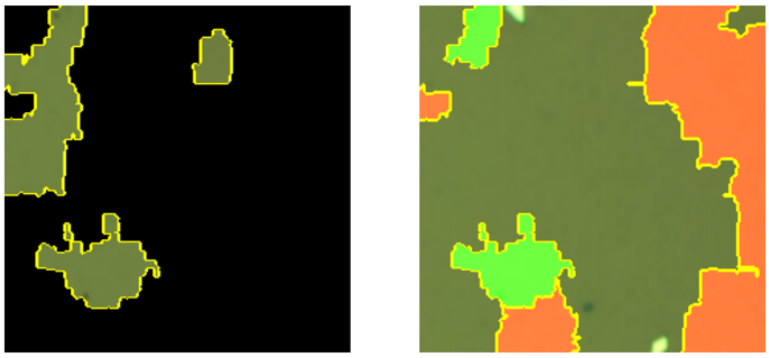

Figure 4: Model interpretation of the classification

Final Model

For our final model, we used a TL approach with the popular ResNetV50 [31]. In this approach, the fully connected layer at the top of the ResNetV50 network was truncated and replaced with layers evaluated via research to integrate specific knowledge and make the final classification. The addition of further layers was important, both to take advantage of TL, but also to be able to train specifically for the problem of multiclassification. The structure of the entire model is shown in figure 5.

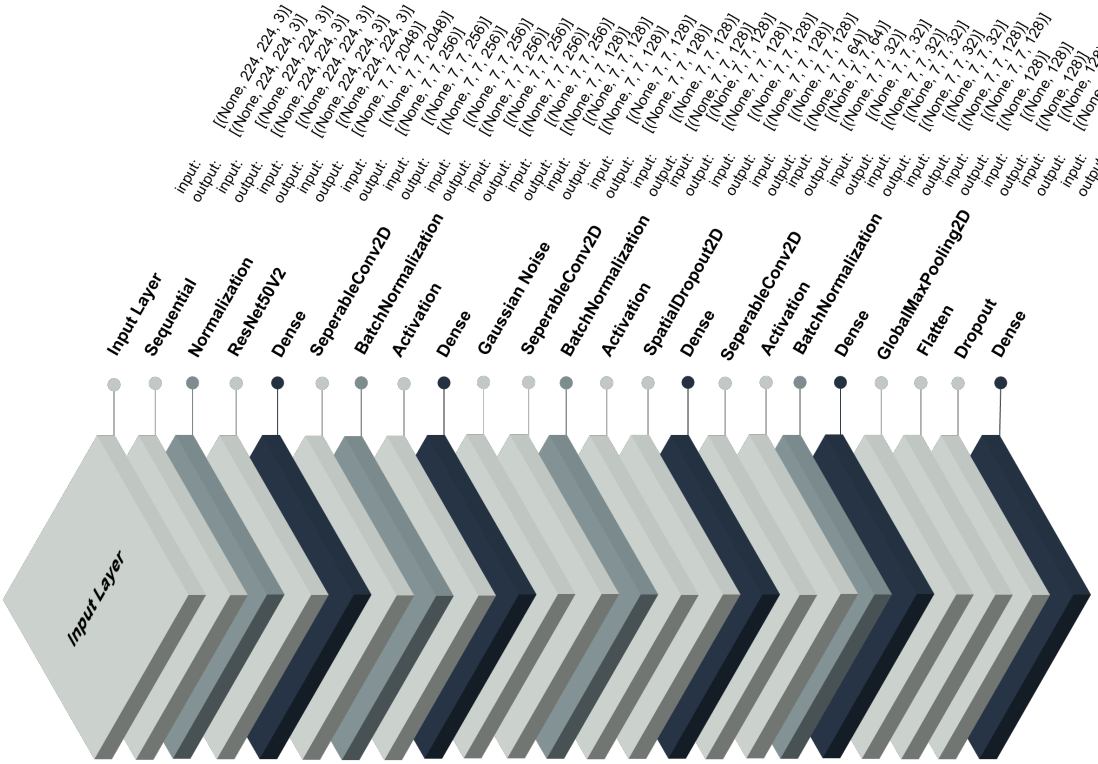

Figure 5: Structure of the final model

In the process of research, the individual layers were experimentally adjusted based on theoretical foundations. Due to the fact that overfitting is a big problem for DL, we integrated dropout layers, within the training step, whereby random neurons, according to the chosen rate, are set to 0, which is supposed to curb overfitting [49, 50]. The remaining inputs are not set to 0, to be eventually scaled up by (1/(1 - rate)), so that the sum over all inputs remains unchanged [49]. In addition, a spatial-dropout 2D-layer was used, which performs the same function as Dropout, but with the difference that full 2D feature maps are deactivated instead of individual elements [51, 52]. This was deliberately incorporated because if neighboring pixels of feature maps are highly correlated, then a regular dropout will not optimally regulate activations and will tend to reduce the effective learning rate. In such a case, which may typically occur in a network, a spatial-dropout 2D-layer is intended to promote independence between feature maps [52]. In addition to this, an additive zero-centered Gaussian noise was integrated to mitigate overfitting as well. Gaussian noise acts as a regularization layer that is only active at training time and represents a natural choice as a corruption process [37].

Only depthwise separable convolutional-2D-layers were used in our model instead of “traditional” ones. This has the background that separable convolutional-2D-layers first perform depthwise spatial convolution and then pointwise convolution, which promises a fast computation and is understood as an extreme version

of an inception block [37]. The activation function ReLu [53] was used for each of those layers, due to the fast computational possibility, the speedy convergence, and the non-negativity of the outputs [54]. In addition, the respective padding ensured that the output had the same height and width as the input [37]. Batch normalization was performed before each activation function, which should make the model faster and more robust by normalizing within mini-batches to a mean of 0 and a standard deviation of 1 for the next input [55, 56]. The exception was the last activation function, where batch normalization was added downstream to make the effect of ReLu additionally more robust [56]. Also integrated is a global max-pooling 2D layer, a sampling-based discretization process used to shrink the feature map by sliding an H × W block with a pool size according to the input size over the data, where H is the height and W is the width of the block [57]. The goal is to shrink the input representation to reduce dimensionality and allow assumptions to be made about the features and patterns it contains [57]. This is followed by a flatten layer, which finally converts the matrix to a 1-dimensional array before the final classification [58]. For the final classification, a softmax activation function was chosen that outputs a probability vector where the elements are non-negative and sum to 1, effectively solving a multiclassification problem [59].

Elementary for the performance of a model are the chosen hyperparamaters, explicitly the batch size and the learning rate can bring a significant performance difference, whereby both are in a high correlation to each other, which should be considered in the definition [60, 61]. This means that, in general, a high learning rate (Learning Rate ¿ 0.01) performs better when a large batch size (Batch Size ¿ 256) is chosen, while conversely, a low learning rate should be chosen for a small batch size [61]. Furthermore, the stack size should be a power of 2 to fully utilize the computational capabilities of GPUs [61]. Considering this correlation, we chose a batch size of 64 and a learning rate of 0.001 for initial training of our model, thus supporting the theory. In this process, all layers of ResNetV50 were frozen and only the appended layers were trained, which is in accordance with the maxim of TL [62]. For fine-tuning, we chose a lower learning rate of 0.00001 to fit all layers of the entire model to the task. In combination with the learning rate, the choice of optimizer is also elementary, and in this case Adamax [43] was chosen, based on Adam but generalized by the infinity norm, which gave the best performance.

Results

The final model achieved the following results. Training included initial training – freezing the baseline model – and fine-tuning, each performed over 10 epochs, resulting in a total of 20 epochs of training with varying focus. The model was able to achieve a balanced overall accuracy of 91.54%, with the hyperparameters defined to the values described, which is the best possible result. The algorithm also achieved an Accuracy of 86.02%, a Kappa of 84.11%, and an AUC of 99.11%. Table 5 summarizes additional performance indicators that provide a deeper insight into how the classifier performs.

Table 5: Results of the final model

	ResNetV50-based Model
Accuracy	86.02%
Balanced Accuracy	91.54%
Kappa	84.11%
Precision	92.07%
Recall	86.02%
AUC	99.11%

Discussion

DL classification of microscopic surface images provides an effective way to enable interpretation of the material image efficiently without human interaction through the learned patterns. Based on the results of the presented model, conclusions about its effectiveness can be drawn, which may allow its application in real-world. The accuracy of 91.54% was achieved by a balance between accuracy and robustness, which was ensured on the one hand by the preprocessing and on the other hand by the model architecture. In the preprocessing, only the training dataset was deliberately adjusted with respect to the elimination by the pixel sum values and the previously retained validation dataset was retained. This resulted in a diminished final accuracy, but a more robust model [46, 47]. This robustness was also addressed by dropout and noise layer, extending the trade-off between robustness and accuracy, which is common to current methods for training robust networks [46].

It can be positively emphasized that the material images, which are visually elementary different, can be classified almost perfectly, which, in contrast to a human scientist, is significantly more efficient, in terms of time and cost. This can be attributed in part to the fact that the model takes a different focus when viewing the image data and can recognize patterns alternatively, unlike the human brain [63]. Fundamental to this is the nature of pattern recognition as an inexact science, allowing different approaches, sometimes complementary, sometimes competing, to the approximate solution of a given problem [63].

The chosen approach to the methodology – dividing the research into sequential phases – made it possible to effectively evaluate a final model from a selection of models, which also makes the individual steps comprehensible to fundamentally support research based on them and allow further artifacts to emerge [33]. This continuous improvement is due to a cycle of planning, executing, checking, and acting, which took place for each model, and was modeled after the Deming-cycle (PDCA-cycle) [26]. Although the PDCA-cycle originated from the management domain and was defined for the control and continuous improvement of processes, this procedure can be transferred into the ML domain, whereby also the characteristics are forced, as pointed out in the research. The continuous pursuit of improvement based on the Deming-cycle must nevertheless be finalized at a justifiable stage of research, which we define by the final model based on the good results.

In terms of the different base models, the performance was similar when compared, with VGG16 outperforming the other models for the most part, until elimination by pixel sum values, which allowed ResNetV50 to achieve the best results. This indicates that deeper data cleaning by the evaluated methodology has a significant impact on predictive performance. Thus, the hard elimination of the probably noisy data by the pixel value sums has a significant part in the good result. The evaluated method is an effective option in the present case because the Graphene, Molybdenum Disulfide, Boron Nitride, and Tungsten Telluride classes have a strongly similar visual appearance and the basic coloration of the classes produces similar pixel value sums. The innovative methodology yielded an information gain of about 8% accuracy from phase 2 to 3, underscoring its effectiveness.

Despite the extensive processing, the four classes addressed pose a difficulty for the model, although it was still possible to classify with more than 50% correctness for each class and basically achieve an initial predictive gain better than random guessing. In the domain of microstructures, it is difficult for humans to distinguish between materials, while our model makes possible initial real-world applications across classes. Explicitly for distinguishing the addressed four classes, our model can provide valuable information that can support humans, with which, in case of uncertainty, a collaboration of human and machine comprehension should lead to an even better decision. Thus, our model is effective without human interaction, but can also be used valuable in combination with humans for discrimination. This interplay may come into play most strongly once attention span in humans begins to wane and potentially fail to recognize patterns and become prone to misinterpretation, allowing the model based on machine classification to contain this danger [15].

In the domain of multiclassification with deep architectures, as performed in this work, the models are often prone to degradation problems that can degrade performance [31]. This also occurred in this work, which is why the mesh depth was not extended beyond a certain point, always examining this with respect to balanced

accuracy. This issue will be explicitly addressed in future work, as it may further improve overall performance.

Conclusion

In this work, an effective transfer learning-based deep learning model was developed for the classification of microscopic material surface images. The model achieved an overall balanced accuracy of 91.54%, which is the first of its kind to set new standards and define a first predictive gain in this domain with the selected dataset. An innovative pixel sum value methodology was used, allowing effective elimination of noisy samples. It was also shown that a TL approach for specific images can be a very effective methodology that should become more relevant in the future. Going further, our model's ability to automatically identify materials from image content can conceivably have applications in materials research as well as construction, as accurate material recognition is a fundamental component to success in design, construction, and maintenance of small and large-scale construction projects [64]. By providing an objective, image-based assessment using the in-house dataset, this work contributes to materials research and technology acceptance of a new IS artifact [65]. This novel ML approach is accurate, inexpensive, fast, and robust, and extends material identification analysis in the domain of materials research and engineering.

References

[1] H. Liu and F. Sun, "Material identification using tactile perception: A semantics-regularized dictionary learning method," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 3, pp. 1050–1058, 2018.

[2] A. Aggarwal and M. Kumar, "Image surface texture analysis and classification using deep learning," *Multimedia Tools and Applications*, vol. 80, no. 1, pp. 1289–1309, 2020.

[3] S. Bell, P. Upchurch, N. Snavely, and K. Bala, "Material recognition in the wild with the materials in context database," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3479–3487.

[4] B. L. DeCost and E. A. Holm, "A computer vision approach for automated analysis and classification of microstructural image data," *Computational Materials Science*, vol. 110, pp. 126–133, 2015.

[5] W. Tarng, C.-F. Tsai, C.-M. Lin, C.-Y. Lee, and H.-H. Liou, "Development of an educational virtual transmission electron microscope laboratory," *Virtual Reality*, vol. 19, no. 1, pp. 33–44, 2014.

[6] C. Bircanoglu, M. Atay, F. Beser, O. Genc, and M. A. Kizrak, "Recyclenet: Intelligent waste sorting using deep neural networks," *2018 Innovations in Intelligent Systems and Applications (INISTA)*, pp. 1–7, 2018.

[7] B. L. DeCost, T. Francis, and E. A. Holm, "Exploring the microstructure manifold: Image texture representations applied to ultrahigh carbon steel microstructures," *Acta Materialia*, vol. 133, pp. 30–40, 2017.

[8] F. A. A. Soares, M. I. Q. Junior, and R. Salvini, "Metallographic specimen imaging classification: A machine learning approach," in *2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)*, 2018, pp. 1–4.

[9] J. Wei, X. Chu, X.-Y. Sun, K. Xu, H.-X. Deng, J. Chen, Z. Wei *et al.*, "Machine learning in materials science," *InfoMat*, vol. 1, no. 3, pp. 338–358, 2019.

[10] W. Choi, H. Huh, B. A. Tama, G. Park, and S. Lee, "A neural network model for material degradation detection and diagnosis using microscopic images," *IEEE Access*, vol. 7, pp. 92 151–92 160, 2019.

[11] E. Kerr, T. McGinnity, and S. Coleman, "Material recognition using tactile sensing," *Expert Systems with Applications*, vol. 94, pp. 94–111, 2018.

[12] J. Weiß and A. Santra, "Material classification using 60-ghz radar and deep convolutional neural network," in *2019 International Radar Conference (RADAR)*, 2019, pp. 1–6.

[13] W. Xia, Z. Chen, Y.-Z. Zhang, and J. Liu, "An approach for road material identification by dual-stage convolutional networks," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, 2018, pp. 7153–7156.

[14] D. Bonello, M. A. Saliba, and K. P. Camilleri, "An exploratory study on the automated sorting of commingled recyclable domestic waste," *Procedia Manufacturing*, vol. 11, pp. 686–694, 2017.

[15] K. Wilson and J. H. Korn, "Attention during lectures: Beyond ten minutes," *Teaching of Psychology*, vol. 34, no. 2, pp. 85–89, 2007.

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

[17] B. Han, Y. Lin, Y. Yang, N. Mao, W. Li, H. Wang, K. Yasuda *et al.*, "Deep-learning-enabled fast optical identification and characterization of 2d materials," *Advanced Materials*, vol. 32, no. 29, pp. 1–10, 2020.

[18] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, "A survey on deep learning techniques for image and video semantic segmentation," *Applied Soft Computing*, vol. 70, pp. 41–65, 2018.

[19] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning - a new frontier in artificial intelligence research [research frontier]," *IEEE Computational Intelligence Magazine*, vol. 5, no. 4, pp. 13–18, 2010.

[20] A. Kensert, P. J. Harrison, and O. Spjuth, "Transfer learning with deep convolutional neural networks for classifying cellular morphological changes," *SLAS DISCOVERY: Advancing the Science of Drug Discovery*, vol. 24, no. 4, pp. 466–475, 2019.

[21] I. Kandel, M. Castelli, and A. Popovič, "Comparative study of first order optimizers for image classification using convolutional neural networks on histopathology images," *Journal of Imaging*, vol. 6, no. 9, pp. 1–17, 2020.

[22] A. Wei, J. Xiong, W. Yang, and F. Guo, "Deep learning-assisted elastic isotropy identification for architected materials," *Extreme Mechanics Letters*, vol. 43, pp. 1–15, 2021.

[23] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[24] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[25] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, pp. 1–9, 2018.

[26] W. D. Pittman and G. R. Russell, "The deming cycle extended to software development," *Production and Inventory Management Journal*, vol. 39, no. 3, pp. 1–6, 1998.

[27] R. A. Reid, E. L. Koljonen, and J. B. Buell, "The Deming Cycle Provides a Framework for Managing Environmentally Resposible Process Improvements," *Quality Engineering*, vol. 12, no. 2, pp. 199–209, 1999.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR 2015*, 2015, pp. 1–14.

[29] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 1251–1258.

[30] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, 2019, pp. 6105–6114.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1–9.

[32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1–9.

[33] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 1–32, 2004.

[34] H. K. D. H. Bhadeshia and J. Christian, "Bainite in steels," *Metallurgical transactions A*, vol. 21, no. 3, pp. 767–797, 1990.

[35] M. Panušková, E. Tillová, and M. Chalupová, "Relation between mechanical properties and microstructure of cast aluminum alloy AlSi9cu3," *Strength of Materials*, vol. 40, no. 1, pp. 98–101, 2008.

[36] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[37] F. Chollet, "Keras," 2015. [Online]. Available: https://keras.io/

[38] T. Jayalakshmi and A.Santhakumaran, "Statistical normalization and back propagationfor classification," *International Journal of Computer Theory and Engineering*, vol. 3, no. 1, pp. 1–5, 2011.

[39] F. Murdiya, A. Hamzah, and D. Andrio, "The application of non-sinusoidal resonance inverter on an ozone generator," in *2019 IEEE Conference on Energy Conversion (CENCON)*, 2019, pp. 1–5.

[40] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, pp. 1–54, 2019.

[41] M. Zhu, J. Xia, X. Jin, M. Yan, G. Cai, J. Yan, and G. Ning, "Class weights random forest algorithm for processing class imbalanced medical data," *IEEE Access*, vol. 6, pp. 4641–4652, 2018.

[42] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[43] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR*, pp. 1–15, 2015.

[44] M. C. Mukkamala and M. Hein, "Variants of RMSProp and Adagrad with logarithmic regret bounds," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, 2017, pp. 2545–2553.

[45] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, *et al.*, "Ad click prediction: a view from the trenches," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 1222–1230.

[46] Y.-Y. Yang, C. Rashtchian, H. Zhang, R. Salakhutdinov, and K. Chaudhuri, "A closer look at accuracy vs. robustness," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1–22, 2020.

[47] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. E. Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, 2019, pp. 7472–7482.

[48] J. Zhang, F. Hu, L. Li, X. Xu, Z. Yang, and Y. Chen, "An adaptive mechanism to achieve learning rate dynamically," *Neural Computing and Applications*, vol. 31, no. 10, pp. 6685–6698, 2018.

[49] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[50] H. Wu and X. Gu, "Towards dropout training for convolutional neural networks," *Neural Networks*, vol. 71, pp. 1–10, 2015.

[51] G. Kang, K. Liu, B. Hou, and N. Zhang, "3d multi-view convolutional neural networks for lung nodule classification," *PLOS ONE*, vol. 12, no. 11, pp. 1–21, 2017.

[52] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 648–656.

[53] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 15, 2011, pp. 315–323.

[54] C. Banerjee, T. Mukherjee, and E. Pasiliao, "An empirical study on generalizations of the relu activation function," *Proceedings of the 2019 ACM Southeast Conference*, pp. 164–167, 2019.

[55] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?" in *Proceedings of the 32nd international conference on neural information processing systems*, 2018, pp. 2488–2498.

[56] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37. PMLR, 2015, pp. 448–456.

[57] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is object localization for free? - weakly-supervised learning with convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 685–694.

[58] Y. Wang, B. Bai, X. Hei, L. Zhu, and W. Ji, "An unknown protocol syntax analysis method based on convolutional neural network," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 5, pp. 1–14, 2020.

[59] D. Yu, "Softmax function based intuitionistic fuzzy multi-criteria decision making and applications," *Operational Research*, vol. 16, no. 2, pp. 327–348, 2015.

[60] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Lecture Notes in Computer Science, Springer Berlin Heidelberg*, 2012, pp. 437–478.

[61] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," *ICT Express*, vol. 6, no. 4, pp. 312–315, 2020.

[62] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[63] J. C. Bezdek, "On the relationship between neural networks, pattern recognition and intelligence," *International Journal of Approximate Reasoning*, vol. 6, no. 2, pp. 85–107, 1992.

[64] I. K. Brilakis, L. Soibelman, and Y. Shinagawa, "Construction site image retrieval based on material cluster recognition," *Advanced Engineering Informatics*, vol. 20, no. 4, pp. 443–452, 2006.

[65] P. F. Wu, "A mixed methods approach to technology acceptance research," *Journal of the AIS*, vol. 13, no. 3, pp. 172–187, 2012.