ESKİŞEHİR TECHNICAL UNIVERSITY JOURNAL OF SCIENCE AND TECHNOLOGY
A- APPLIED SCIENCES AND ENGINEERING

Estuscience – Se, 2024, 25 [4] pp. 602-616, DOI: 10.18038/estubtda.1545522

RESEARCH ARTICLE

SEMI-SUPERVISED CLASSIFICATION OF 2D MATERIALS USING SELF-TRAINING CONVOLUTIONAL NEURAL NETWORKS

Umut Kaan KAVAKLI [1] , Bahar GÖRGÜN [2] , Ayşegül TERZİ [3] , Cahit PERKGÖZ [4] *

[1] Department of Computer Engineering, Faculty of Engineering, Eskisehir Technical University, Eskişehir, Turkey
umutkaankavakli@gmail.com - *0009-0003-4968-4124*

[2] Department of Computer Engineering, Faculty of Engineering, Eskisehir Technical University, Eskişehir, Turkey.
bahargorgun9@gmail.com - *0009-0003-4299-4488*

[3] Department of Computer Engineering, Faculty of Engineering, Eskisehir Technical University, Eskişehir, Turkey.
terziaysegul9696@gmail.com - *0009-0009-3671-5562*

[4] Department of Computer Engineering, Faculty of Engineering, Eskisehir Technical University, Eskişehir, Turkey.
cahitperkgoz@eskisehir.edu.tr - *0000-0003-0424-7046*

Abstract

Deep learning algorithms require large amounts of data, and their accuracy rates are directly related to the amount and quality of the data. Moreover, supervised learning models require the data to be labeled. However, data labeling is always a time-consuming and laborious process. Labeling data obtained from microscope images can be more laborious. Molybdenum disulfide (MoS_2) in monolayer form, which can be produced on large surfaces with the chemical vapor deposition method (CVD) and has advantages for potential electronic applications, is a frequently studied material in the field of nanotechnology. However, MoS_2 produced on these large surfaces usually has defective surfaces and needs to be detected. This process is a difficult process to be performed with a microscope by an expert. Artificial intelligence-based supervised learning algorithms, which need labeled data, provide an effective solution for these detections. Furthermore, increasing the number of labeled data increases the accuracy of these algorithms.

In this study, a teacher-student model is explored using self-training, a semi-supervised learning technique, to effectively train a deep convolutional neural network to detect defects on MoS_2 samples. Initially, the teacher model is trained using a small amount of data labeled by an expert. This trained model is enriched by generating pseudo-labels for previously unlabeled data. Then, a student model is trained using these real and pseudo-labeled data. The trained model then replaces the teacher model, and the process repeats, gradually improving labeling accuracy. The results show that the self-training method increases accuracy from 77% to 82% compared to the CNN model trained only on the existing labeled data, and the defect regions in MoS_2 are effectively classified with minimal manual labeling.

Keywords

Self-training,
Deep learning,
CNN,
Two-dimensional materials,
MoS_2

Time Scale of Article

Received : 10 September 2024
Accepted : 02 December 2024
Online date :27 December 2024

1. INTRODUCTION

In recent years, one of the most compelling fields has been artificial intelligence (AI) and its applications, particularly due to its potential to offer solutions to complex and time-consuming problems [1, 2]. As computing power has increased, AI techniques have been applied to numerous areas where their use was previously unfeasible [3, 4]. One of these areas is nanotechnology, specifically two-

*Corresponding Author: cahitperkgoz@eskisehir.edu.tr dimensional (2D) materials, which have attracted significant attention due to their superior properties. Nanotechnology generally deals with the manipulation of materials at the atomic and molecular levels, focusing on the design, production, and application of materials at extremely small scales. Understanding the complex structures and properties of nanomaterials, optimizing them, and developing them for innovative applications have become more feasible with the integration of AI techniques.

Several studies have used machine learning methods to optimize the properties of graphene-based materials. Neural networks, a subclass of artificial intelligence, have been explored as a method to predict the behavior of materials using data obtained from laboratory instruments [5]. In another study, artificial intelligence methods have shown how they can be used to optimize the production processes of nanomaterials and support their characterization processes [6]. These studies highlight the importance of AI in the field of nanotechnology, but AI can be employed not only for the design and development of new nanomaterials but also to optimize their production processes. It can predict defects in the production phase, allowing for preventive measures to be taken, thereby increasing production efficiency, reducing costs, and improving product quality.

Since the discovery of graphene [7], interest in 2D materials has surged, and various 2D materials have been extensively studied [8]. Among these, transition metal dichalcogenides (TMDs) stand out as one of the most popular groups. In their monolayer form, TMDs possess unique optoelectronic properties, such as a direct bandgap, making them highly attractive for both material science and optoelectronic device applications [9]. Additionally, 2D materials offer a solution to reducing power loss in electronic chips by lowering the off-state current in transistors and enabling further miniaturization. [10]. Due to these advantages, 2D TMDs hold promise as channel materials in MOSFETs, offering wider bandgaps compared to graphene. Among the most studied TMDs is molybdenum disulfide (MoS_2), composed of covalently bonded Mo and S atoms, with weak Van der Waals forces holding the layers together [11, 12]. However, to fabricate devices from 2D materials, they must be produced on large surfaces without defects. While techniques such as mechanical exfoliation [13] and other methods [14] have been proposed, chemical vapor deposition (CVD) is the most suitable method for compatibility with microelectronics [15]. The CVD method shows promise for producing large-area, monolayer materials [16].

A crucial step in research is the analysis and characterization of 2D materials produced in the laboratory, which presents several challenges. One of the most significant challenges is determining whether the produced materials are defect-free or defective. While techniques such as Raman scattering spectroscopy, atomic force microscopy (AFM), and photoluminescence (PL) spectroscopy [17, 18] are commonly used, microscopic analyses are the most cost-effective [19], though they require expert operators. However, automating the characterization of large surfaces under a microscope using AI could make the process more practical and efficient. AI models can predict the properties of new nanomaterials from experimental data and theoretical calculations, analyzing images to more precisely characterize the structure and properties of nanomaterials, thereby minimizing trial-and-error in experiments. Various deep learning methods have been tested for this purpose [20-22].

Deep learning, a subset of AI, has gained popularity in nearly every scientific field over the past two decades, with convolutional neural networks (CNNs) making significant advancements, particularly in image processing and classification [23-25]. However, like all deep learning methods, CNN-based approaches require a large amount of labeled data. In the case of MoS_2 samples obtained through experiments that take hours in the laboratory, gathering and labeling sufficient data is both time-consuming and labor-intensive. In cases where labelled real data is scarce, we previously used Fresnel equations to generate artificial datasets and trained a CNN model using transfer learning to classify real MoS_2 flakes as normal or defective [26].

To address challenges with limited labeled data, self-training methods have been developed, utilizing both labeled and unlabeled data to improve model performance. In self-training [27, 28], a 'teacher' model is initially trained on a small, expert-labeled dataset known as 'true labels.' The teacher model then predicts 'pseudo-labels' on a large unlabeled dataset. A 'student' model is subsequently trained using a combination of both true and pseudo-labels. The student model is iteratively replaced by a new teacher model, and pseudo-labels are generated iteratively. As iterations progress, better pseudo-labels are produced, and the model's accuracy gradually improves with each iteration. Self-training methods have been applied to segmentation problems within the context of deep learning, offering a semi-supervised approach to enhance model performance with limited labeled data [29, 30]. In this study, a semi-supervised self-training deep learning method is proposed to identify defects in 2D materials when most of the data is unlabeled. Instead of manually labeling data, which is time-consuming and requires expert input, the approach first labels the unlabeled data and then performs classification using a CNN structure. MoS_2 was selected as a representative 2D material because of its widespread use in nanotechnology studies, attributed to its semiconductor properties and layered structure.

This article proceeds as follows: Section 2 introduces the proposed methodology and CNN concept. Section 3 discusses the experimental work and results, while Section 4 presents the conclusions.

2. MATERIALS AND METHODS

The goal of this study is to use deep learning methods to classify whether MoS_2 flakes in microscope images are defective or not (Figure 1). Initially, a small set of labeled data and a teacher-student semi-supervised model will be used to label the unlabeled data through self-learning. A CNN model will then be trained using both the labeled and pseudo-labeled data, and its performance will be evaluated using test data consisting real images for classification purposes. Therefore, in the following section, the designed CNN model will first be introduced, followed by an explanation of the teacher-student model. Finally, the methods used to define the success criteria of the approach will be discussed.

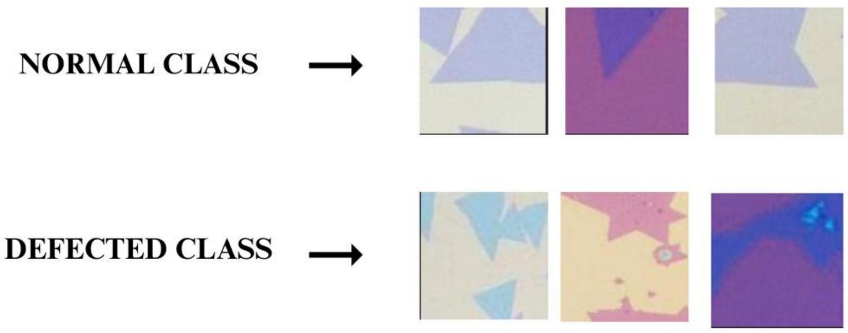


Figure 1. Normal and defected MoS_2 flakes

2.1. Convolutional Neural Networks

A Convolutional Neural Network (CNN) is a fundamental architecture in deep learning, designed specifically to process and extract information from images by progressively learning from low-level features to high-level patterns [31]. CNNs operate by applying multiple grids, known as kernels, over the input images. These kernels, which contain learnable parameters, capture different features and are optimized during training to enhance model performance. A non-linear activation function is employed to determine the activation of neurons based on the values obtained after linear computations [32].

Rather than using a single convolutional layer, CNNs consist of multiple layers, typically connected with pooling and fully connected layers. Pooling layers, such as max pooling or average pooling, reduce the spatial dimensions of the feature maps while retaining critical information, thus enhancing computational efficiency. Fully connected layers, often found in the final stages of the network, aggregate the information from the convolutional layers and serve as classifiers. Numerous state-of-the-art CNN architectures exist in the literature, including VGG [32], ResNet [33], and Inception [32], which have demonstrated strong performance in image classification tasks.

In this study, a CNN architecture with three convolutional layers is implemented, featuring 32, 64, and 128 filters of size 3x3 in each respective layer. Each convolutional layer is followed by a 2x2 max pooling layer. After the convolutional and pooling layers, a fully connected layer with 128 neurons is employed, and the final output layer utilizes a sigmoid activation function to support binary classification.

The objective of a machine learning model is to optimize its weights to achieve the best possible performance. During training, loss functions play a critical role in this optimization process by quantifying the discrepancy between the actual (ground truth) and predicted values of the model. The smaller the loss, the closer the model's predictions are to the ground truth, indicating higher model accuracy.

Several loss functions have been developed to address specific challenges, such as data imbalance. Data imbalance, a common issue in machine learning, arises when the distribution of classes is skewed, leading to biased model predictions favoring the majority class and poor performance in predicting the minority class. To mitigate this issue, specialized loss functions like Weighted Binary Cross Entropy (WBCE), Focal Loss, and Tversky Loss are often employed. However, in this study, the dataset is balanced, making the standard Binary Cross Entropy (BCE) loss function (Equation 1) sufficient for optimizing the model's performance.

$$L_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i log\left(\hat{y}_i\right) + \left(1 - y_i\right) log\left(1 - \hat{y}_i\right)\right) \tag{1}$$

where N is the number of samples, y_i is the real label/output, and $\hat{y}$ is the predicted label/output of the model. BCE effectively measures the difference between predicted probabilities and actual binary class labels, guiding the model towards better predictions.

An optimization algorithm is used to update a model's weights based on the error output from the loss function. This process, known as backpropagation, adjusts each weight by propagating errors backward through the layers. The learning rate, a hyperparameter of the optimizer, controls the step size of these updates.

Gradient descent is a fundamental optimization method that iteratively updates the model's parameters to minimize the loss. However, it is often impractical due to its use of the entire dataset at once, which can strain memory resources. For this reason, the stochastic gradient algorithm (SGD) that uses subsets of the dataset (mini-batch) is mostly preferred. Moreover, more effective algorithms such as momentum SGD, RMSprop and Adam have also been used based on SGD. In this study, the Adam optimizer was preferred due to its durability and widespread use in deep learning applications.

While deep learning methods try to increase their accuracy performance, they only work on the dataset on which the training was made. If the algorithm is not stopped at a certain point, memorization may occur on this dataset. As a result, while very high-performance values are obtained on the training set, the results would be worse on the test data. This situation is called overfitting and should be prevented. The strategies applied to prevent overfitting include increasing the training data, using data augmentation techniques, applying weight decay and adding dropout layers. On the contrary, in cases where the model is not complex enough or training is done with a small number of data, underfitting occurs and the performance of neural networks on the test data is again low. Increasing the model complexity can help address underfitting.

While CNNs perform effectively when trained on sufficiently large datasets, acquiring sufficiently labeled data in real-world applications can be challenging in some cases. For example, such challenges can be encountered when working with datasets that require expertise for labeling, such as microscopic images. To overcome this problem, semi-supervised approaches that utilize both labeled and unlabeled data, such as the self-training teacher-student model, have emerged as effective strategies. Improving performance in cases where limited labeled data is available can be achieved with the self-training teacher-student model.

2.2. Self-Training Model

Data labeling is one of the most challenging pre-processing steps for preparing data for model training, requiring significant expertise, time, and attention. In this study, a custom dataset was obtained under a microscope, with corresponding annotations provided by an expert. To address the challenges of labeling, a self-training method is employed. This technique, part of semi-supervised learning, utilizes a combination of a small number of labeled and a large number of unlabeled data to train the model. The training phase begins with only the labeled data, and the trained model is referred to as the teacher (Figure 2).

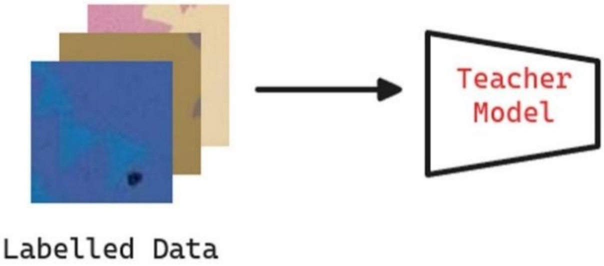


Figure 2. Teacher model trained with labelled data

After the teacher model is trained, it generates pseudo-labels by predicting on the unlabeled samples (Figure 3). As these labels are produced by the teacher model, some degree of noise is present in the predictions. Consequently, these pseudo-labels do not serve as an exact substitute for the true labeling process; however, the information they contain remains valuable for subsequent training phases. It is expected that the pseudo-labels generated by the teacher model may initially contain a certain amount of noise and may not be able to label all examples correctly. However, in the iterative teacher-student architecture, each new student model is updated, and the pseudo-labeling process continues in this loop. This gradually improves the accuracy of the pseudo-labels, improving the performance of the model.

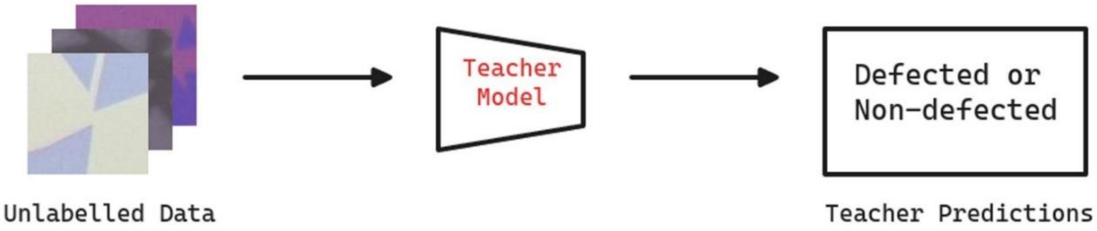


Figure 3. Teacher model predicting pseudo-labels

A new model is subsequently trained from scratch using both the real labeled and pseudo-labeled data (Figure 4). Since this model is trained on a combination of real and pseudo-labeled data, it is referred to as the student model. Although the student model may incorporate some noise, its performance is expected to improve due to training on a more diverse dataset. The noise level in pseudo-labels may affect the performance of the model at each iteration, and the low quality of pseudo-labels may lead the model to learn some incorrect patterns. This problem can be mitigated by using an early stopping point. Additionally, improving the quality of pseudo-labels using thresholding or filtering methods can significantly increase the accuracy of the model. This improvement can be further investigated in future work to optimize this method.

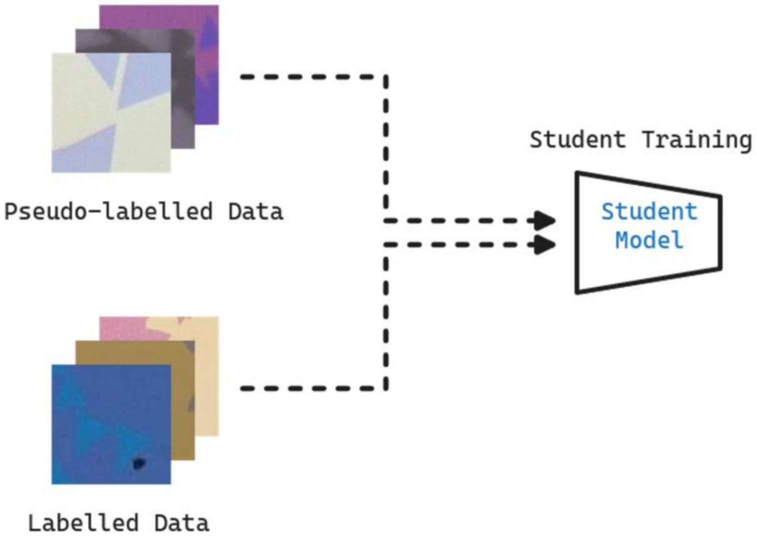


Figure 4. Student model trained with labelled and pseudo-labelled data

Upon completion of the training phase, the student model is promoted to serve as the new teacher model (Figure 5). It then generates a new set of pseudo-labels, which are expected to be of higher quality than the previous ones.

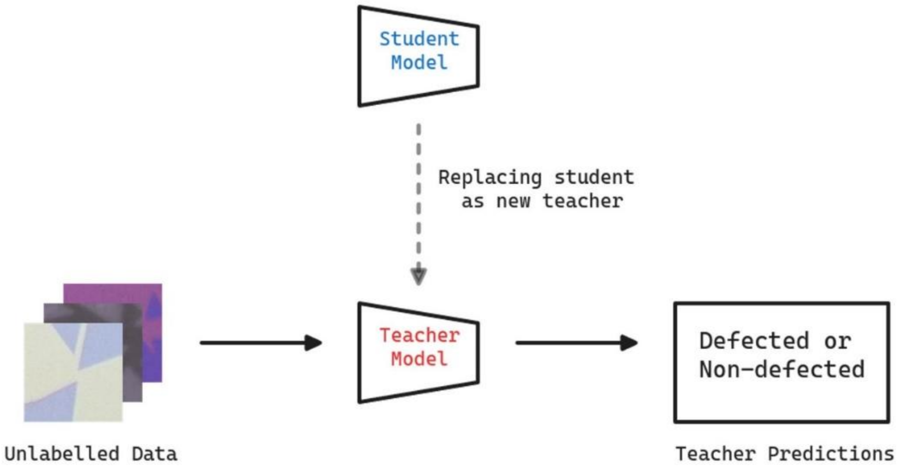


Figure 5. New teacher model copied from student model

There is no fixed limit to the number of student models that can be created or the number of pseudo-labels that can be generated. The process is iterative, as shown in Figure 6, with each new student model serving as the basis for generating updated pseudo-labels. This cycle continues until there is no significant improvement in the performance of the model. The challenges associated with training multiple student models iteratively are minimal, as the primary additional requirement is rerunning the software for the number of cycles in the teacher-student framework. This results in a linear increase in overall training time proportional to the number of iterations in each cycle. However, apart from this increase in computation time, no other significant challenges or complexities are introduced by the method. The process remains straightforward and scalable within the limits of available computational resources. At this point, the iterative process is typically stopped to avoid overfitting. The overall steps are illustrated in Figure 6.

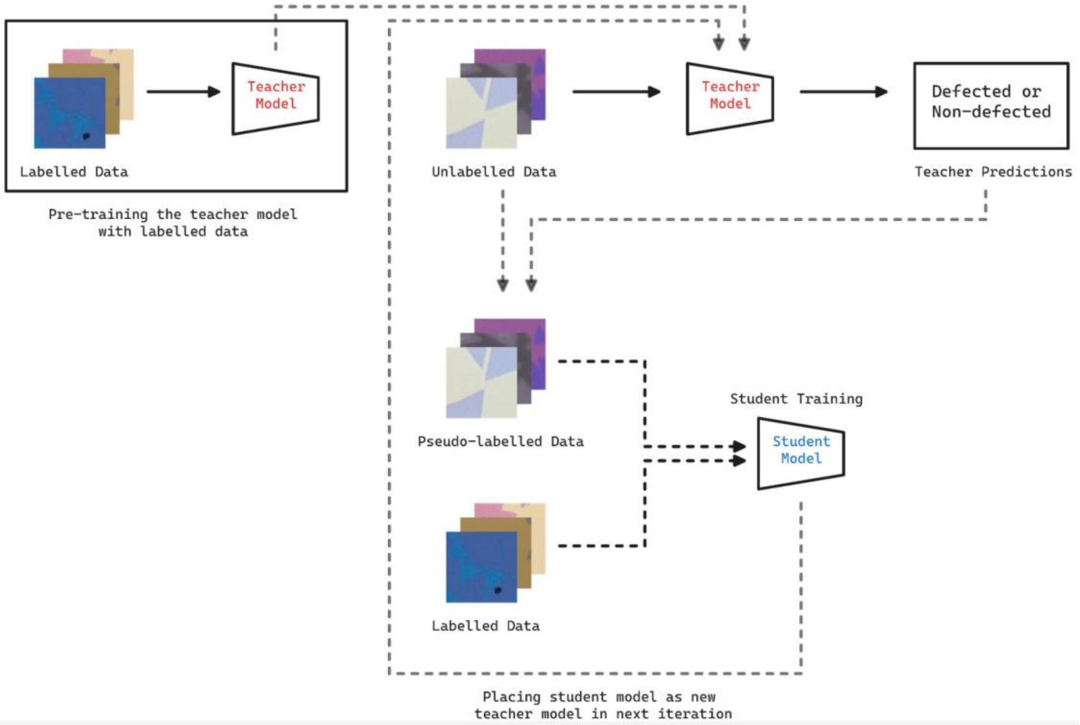
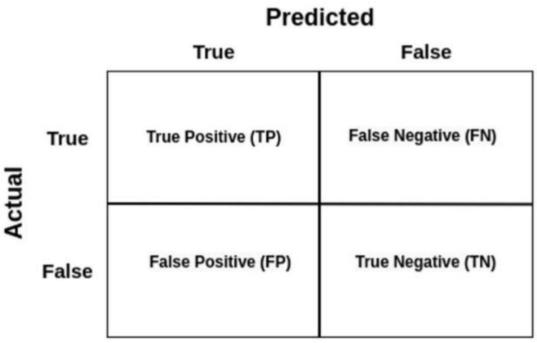


Figure 6. New teacher model copied from student model

2.3. Evaluation Metrics

The evaluation step assesses the true performance of the model. Since the model is trained on the training data and hyperparameters are optimized using the validation data, neither can be used for final evaluation, as the model is explicitly (training) or implicitly (validation) influenced by these datasets. Instead, the performance of the model on unseen samples is evaluated using a test set. Performance can be measured using various metrics, including accuracy, precision, recall, and F1 score (Equations 2-5). These metrics are computed based on a confusion matrix (Table 1), which provides a detailed breakdown of the performance of the model across specific criteria.

Table 1 Confusion matrix



		Predicted	
		True	False
Actual	True	True Positive (TP)	False Negative (FN)
	False	False Positive (FP)	True Negative (TN)

The confusion matrix displays the relationship between the predicted values and the actual true values in a tabular format. It is utilized to analyze the model's errors and to identify imbalances between classes. Given that the dataset for this study was specifically designed to address data imbalance issues, only the accuracy metric was employed to evaluate the initial teacher and student models.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F1\ score = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall} \tag{5}$$

Since the goal is to improve performance following the teacher model, the success criterion is defined as the increase in accuracy from the initial model to the final model.

3. RESULTS AND DISCUSSIONS

The model was conducted using Python programming with the Keras API of the TensorFlow framework and trained on an Nvidia GeForce GTX 1660 TI system with 6 GB of memory. Due to the relatively small image sizes (100x100 pixels), a CNN network was designed instead of resizing images to fit larger, state-of-the-art networks to mitigate overfitting. Data augmentation techniques, including Random Rotation and Random Vertical and Horizontal Flip, were applied. Non-linear activation functions in the hidden layers were selected as ReLU for its reliability, while a sigmoid function was used for the classifier at the end of the network. The networks were trained with a batch size of 4 and 10 epochs each. The choice of a low epoch size was made to prevent overfitting, given the limited amount of data. The hyperparameters and implementation details are provided in Table 2.

Table 2. Implementation details of the proposed model

Hyperparameter	Value
Input Image Size	100 x 100
Size of Labeled Training Dataset	246 (24%)
Size of Unlabeled Training Dataset	580 (56%)
Size of Validation Dataset	104 (10%)
Size of Test Dataset	104 (10%)
Epochs	10
Batch Size	4
Learning Rate	10^{-5}
Optimizer	Adam
Activation Function	ReLU
Loss Function	Binary Cross Entropy
Teacher Student Iteration	5
Data Augmentation	Random Rotation and Flip

A total of 6 networks were trained, consisting of one initial teacher network and five student networks. All results of the models were evaluated using the accuracy metric, which indicated that each subsequent student model demonstrated improved performance. The final results show an increase in accuracy from 77% to 82% with the use of self-training (Figure 7). Along with accuracy, other performance metric values (precision, recall, and F1-score) are provided in Table 3.

The iterative process in the teacher-student framework follows certain criteria to ensure effective training and prevent overfitting. In each iteration, pseudo-labels are generated based on predictions with confidence scores greater than 0.5, which are then assigned as new labels. The difference in loss values between successive student models is monitored to determine when to stop the iterative process. If this difference falls below 0.01, the training process is stopped to prevent overfitting while maintaining optimal model performance. These criteria increase the robustness of the self-training approach by providing a balance between comprehensive training and computational efficiency.

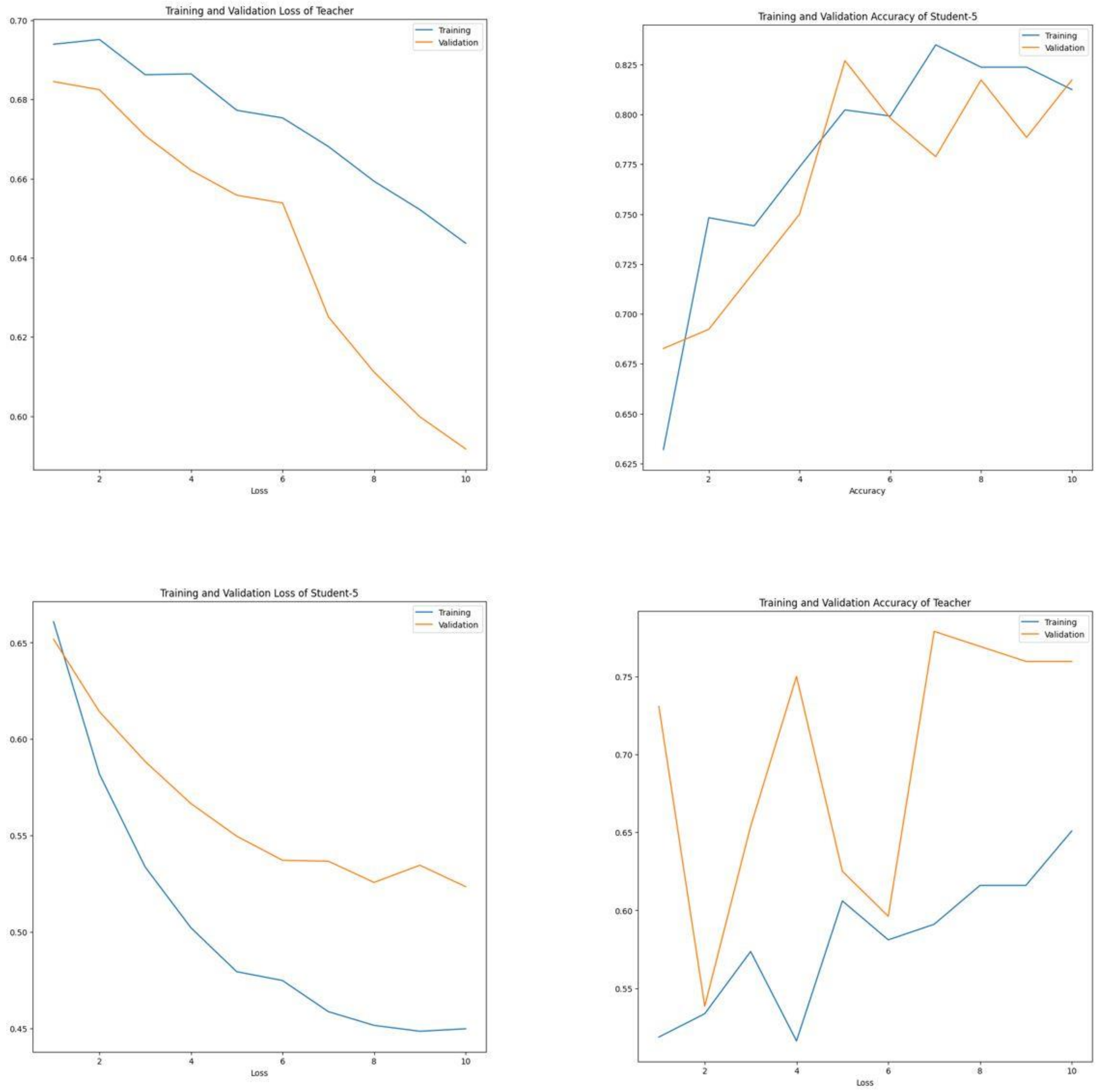


Figure 7. Loss and accuracy of teacher and student models

Table 3. Performance metric values

	Accuracy	**Precision**	**Recall**	**F1 Score**
Teacher	0.7788	0.7925	0.8077	0.8002
Last Student	0.8269	0.8864	0.7500	0.8125

Early stopping and best checkpoint saving techniques were also employed to utilize the optimal weights if the model exhibited high loss or poor accuracy at the end of the training phase. Consequently, pseudo-labels were generated using the best weights of the teacher model rather than the final weights.

As shown in Figure 8, accuracy increases with each iteration of training the new student model. Although a peak in loss is observed at student 2, it did not impact the overall improvement in model performance.

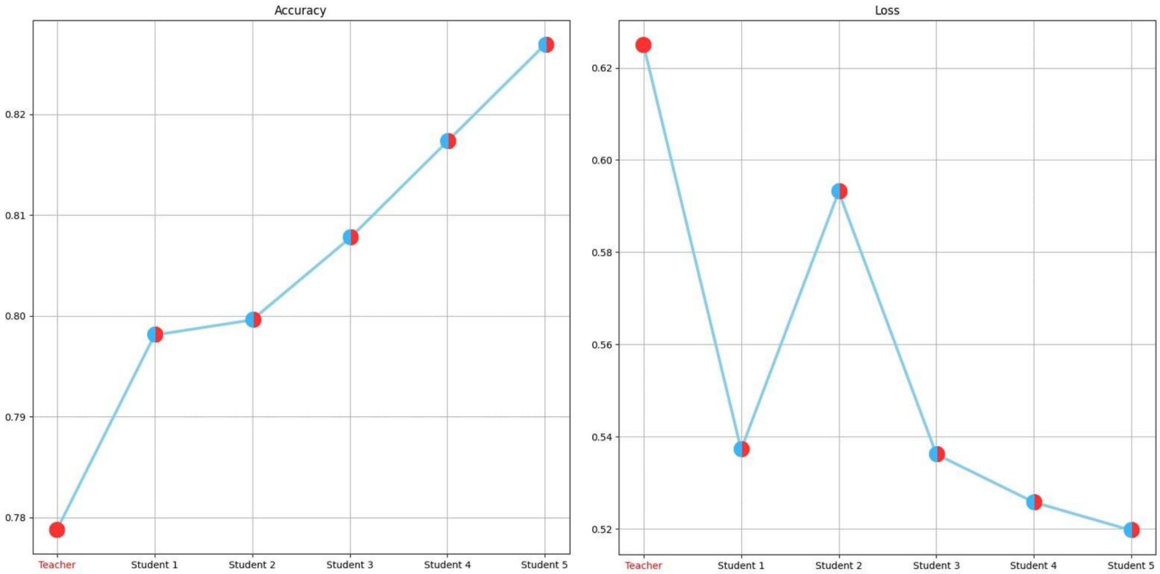


Figure 8. Accuracy and loss at each iteration

The model was tested using both the proposed Convolutional Neural Network and traditional machine learning approaches (Figure 9). The results indicate that the Convolutional Neural Network performs significantly better in classifying this problem compared to other models.

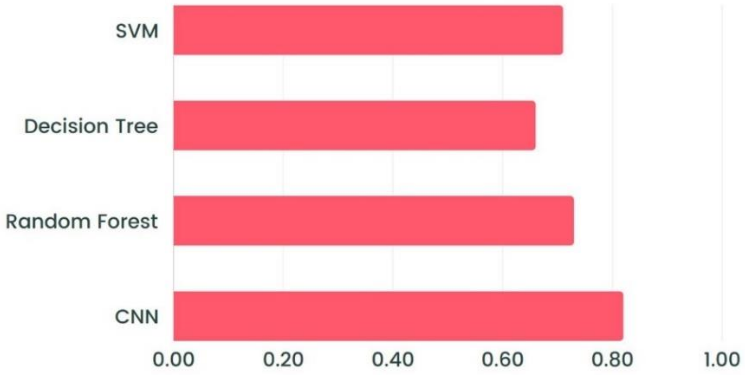


Figure 9. Comparison of machine learning algorithms with the proposed CNN model

A self-learning strategy was developed, given that only a small portion of the available data was labeled. The unlabeled data was labeled through this self-learning method. Initially, the teacher model achieved a relatively lower accuracy rate due to being trained on a limited amount of labeled data. However, as the previously unlabeled data were labeled and incorporated into the training set, an increase in the accuracy trend was observed, as expected. The model requires the training of multiple neural networks rather than a single one, which necessitates more time, but the results demonstrated improved performance compared to single-network training. Furthermore, this approach proved advantageous when compared to other machine learning models.

4. CONCLUSIONS

In this study, a self-learning method for the classification problem involving partially labeled datasets was investigated and tested on microscope images of two-dimensional nanomaterial Molybdenum Disulfide (MoS_2). Data labeling is a time-consuming and often expert-requiring preprocessing step. A dataset obtained from MoS_2 was collected and a small subset of the data was labeled under a microscope by experts. In addition to this data, unlabeled data was also provided. Semi-supervised self-training method, which allows both labeled and unlabeled data to be used in model training, was successfully implemented. The iterative training process significantly increased the performance of the algorithm as more pseudo-labeled data was used in each training cycle. All models were evaluated using the accuracy metric and it was observed that each subsequent learner model outperformed the previous model. The accuracy of the model increased from 77% to 82% with the self-training method compared to the CNN model trained with only manually labeled data. These results demonstrated the effectiveness of the method for partially labeled datasets. It has been shown to be a feasible solution to reduce the manual labeling burden in similar tasks.

In addition to the self-training method, the model has been tested using both the proposed Convolutional Neural Network (CNN) and traditional machine learning methods. The results show that proposed CNN significantly outperforms traditional models in classifying MoS_2 images. Combining labeled and pseudo-labeled data allow the model to achieve significant accuracy gains without requiring a fully labeled dataset. Collecting higher quality, larger, and more detailed images can further improve the performance of the model by providing a richer dataset for training. Furthermore, developing more advanced neural network architectures, optimizing hyperparameters, and experimenting with different learning strategies can also lead to better results. Integrating this approach into real-time platforms may offer practical applications for defect detection in industries such as semiconductor manufacturing or materials science.

In conclusion, the self-training method demonstrated in this study offers a promising solution to automate the labeling process and improve the classification accuracy of MoS_2 defects. By reducing the reliance on expert-labeled data, this method could significantly improve workflow in nanomaterial research and enable more efficient and accurate defect detection across a wider range of applications.

ACKNOWLEDGEMENTS

This work was supported by TÜBİTAK 2209-A Project no: 1919B012334338. We would like to express our gratitude to the Micro Nano Devices and Systems (MIDAS) laboratory at Eskişehir Technical University for generously providing the optical microscopy images used in this research. Special thanks to Prof. Feridun Ay and Prof. Nihan Kosku Perkgoz for the acquisition of the optical microscopy images.

CONFLICT OF INTEREST

The authors stated that there are no conflicts of interest regarding the publication of this article.

CRediT AUTHOR STATEMENT

Umut Kaan Kavaklı: Methodology, Software, Formal Analysis, Data Curation, Investigation, Visualization, Validation, Writing – Original Draft.
Bahar Görgün: Conceptualization, Methodology, Software, Data Curation, Investigation, Resources, Writing – Original Draft.
Ayşegül Terzi: Methodology, Software, Validation, Data Curation, Visualization, Investigation, Writing – Original Draft.
Cahit Perkgöz: Conceptualization, Methodology, Resources, Writing – Review & Editing, Supervision.

REFERENCES

[1] Alpaydin E. Introduction to machine learning: MIT press, 2020.

[2] Ahmed SF, Alam MSB, Hassan M, Rozbu MR, Ishtiak T, Rafa N, et al. Deep learning modelling techniques: current progress, applications, advantages, and challenges. Artificial Intelligence Review, 2023;56:13521-617.

[3] Liu Y, Yang Z, Yu Z, Liu Z, Liu D, Lin H, et al. Generative artificial intelligence and its applications in materials science: Current situation and future perspectives. Journal of Materiomics, 2023;9:798-816.

[4] Vasoya N. Revolutionizing nano materials processing through IoT-AI integration: opportunities and challenges. Journal of Materials Science Research and Reviews, 2023;6:294-328.

[5] Sobral JA, Obernauer S, Turkel S, Pasupathy AN, Scheurer MS. Machine learning the microscopic form of nematic order in twisted double-bilayer graphene. Nature Communications, 2023;14:5012.

[6] Nandipati M, Fatoki O, Desai S. Bridging Nanomanufacturing and Artificial Intelligence—A Comprehensive Review. Materials, 2024;17:1621.

[7] Geim AK, Novoselov KS. The rise of graphene. Nature materials, 2007;6:183-91.

[8] Zhang Y, Yao Y, Sendeku MG, Yin L, Zhan X, Wang F, et al. Recent progress in CVD growth of 2D transition metal dichalcogenides and related heterostructures. Advanced materials, 2019;31:1901694.

[9] Zhang H. Introduction: 2D materials chemistry. ACS Publications, 2018. p. 6089-90.

[10] Hua Q, Gao G, Jiang C, Yu J, Sun J, Zhang T, et al. Atomic threshold-switching enabled MoS_2 transistors towards ultralow-power electronics. Nature Communications, 2020;11:6207.

[11] Zhang Y, Wan Q, Yang N. Recent advances of porous graphene: synthesis, functionalization, and electrochemical applications. Small, 2019;15:1903780.

[12] Fortin E, Sears W. Photovoltaic effect and optical absorption in MoS_2. Journal of Physics and Chemistry of Solids, 1982;43:881-4.

[13] Yi M, Shen Z. A review on mechanical exfoliation for the scalable production of graphene. Journal of Materials Chemistry A, 2015;3:11700-15.

[14] Bonaccorso F, Lombardo A, Hasan T, Sun Z, Colombo L, Ferrari AC. Production and processing of graphene and 2d crystals. Materials today, 2012;15:564-89.

[15] Liu H, Wong SL, Chi D. CVD growth of MoS_2-based two-dimensional materials. Chemical Vapor Deposition, 2015;21:241-59.

[16] Liu D, Chen X, Yan Y, Zhang Z, Jin Z, Yi K, et al. Conformal hexagonal-boron nitride dielectric interface for tungsten diselenide devices with improved mobility and thermal dissipation. Nature Communications, 2019;10:1188.

[17] Özden A, Şar H, Yeltik A, Madenoğlu B, Sevik C, Ay F, et al. CVD grown 2D MoS_2 layers: A photoluminescence and fluorescence lifetime imaging study. physica status solidi (RRL)–Rapid Research Letters, 2016;10:792-6.

[18] Zhang J, Yu Y, Wang P, Luo C, Wu X, Sun Z, et al. Characterization of atomic defects on the photoluminescence in two-dimensional materials using transmission electron microscope. InfoMat, 2019;1:85-97.

[19] Yorulmaz B, Özden A, Şar H, Ay F, Sevik C, Perkgöz NK. CVD growth of monolayer WS_2 through controlled seed formation and vapor density. Materials Science in Semiconductor Processing, 2019;93:158-63.

[20] Lin X, Si Z, Fu W, Yang J, Guo S, Cao Y, et al. Intelligent identification of two-dimensional nanostructures by machine-learning optical microscopy. Nano Research, 2018;11:6316-24.

[21] Ngome Okello OF, Yang D-H, Chu Y-S, Yang S, Choi S-Y. Atomic-level defect modulation and characterization methods in 2D materials. APL Materials, 2021;9.

[22] Al-Waisy AS, Ibrahim DA, Zebari DA, Hammadi S, Mohammed H, Mohammed MA, et al. Identifying defective solar cells in electroluminescence images using deep feature representations. PeerJ Computer Science, 2022;8:e992.

[23] Yao G, Lei T, Zhong J. A review of convolutional-neural-network-based action recognition. Pattern Recognition Letters, 2019;118:14-22.

[24] Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of big Data, 2021;8:1-74.

[25] Bhuvaneswari V, Priyadharshini M, Deepa C, Balaji D, Rajeshkumar L, Ramesh M. Deep learning for material synthesis and manufacturing systems: A review. Materials Today: Proceedings, 2021;46:3263-9.

[26] Perkgoz C. Identifying optical microscope images of CVD-grown two-dimensional MoS_2 by convolutional neural networks and transfer learning. PeerJ Computer Science, 2024;10:e1885.

[27] Xie Q, Luong M-T, Hovy E, Le QV. Self-training with noisy student improves imagenet classification. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020. p. 10687-98.

[28] Zou Y, Yu Z, Liu X, Kumar B, Wang J. Confidence regularized self-training. Proceedings of the IEEE/CVF international conference on computer vision, 2019. p. 5982-91.

[29] Yu L, Liu X, Van de Weijer J. Self-training for class-incremental semantic segmentation. IEEE Transactions on Neural Networks and Learning Systems, 2022;34:9116-27.

[30] Ke R, Aviles-Rivero AI, Pandey S, Reddy S, Schönlieb C-B. A three-stage self-training framework for semi-supervised semantic segmentation. IEEE Transactions on Image Processing, 2022;31:1805-15.

[31] Yamashita R, Nishio M, Do RKG, Togashi K. Convolutional neural networks: an overview and application in radiology. Insights into imaging, 2018;9:611-29.

[32] Soffer S, Ben-Cohen A, Shimon O, Amitai MM, Greenspan H, Klang E. Convolutional neural networks for radiologic images: a radiologist's guide. Radiology, 2019;290:590-606.

[33] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016. p. 770-8.