IOP Publishing Mach. Learn.: Sci. Technol. 3 (2022) 045029 https://doi.org/10.1088/2632-2153/aca744

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
25 August 2022

REVISED
28 November 2022

ACCEPTED FOR PUBLICATION
29 November 2022

PUBLISHED
22 December 2022

PAPER

Pixel-wise classification in graphene-detection with tree-based machine learning algorithms

Woon Hyung Cho[1,2], **Jiseon Shin**[1], **Young Duck Kim**[3] and **George J Jung**[1,2,*]

[1] Department of Physics, University of Seoul, Seoul 02504, Republic of Korea
[2] Department of Smart Cities, University of Seoul, Seoul 02504, Republic of Korea
[3] Department of Physics, Department of Information Display, KHU-KIST Department of Converging Science and Technology, Kyung Hee University, Seoul, 02447, Republic of Korea
[*] Author to whom any correspondence should be addressed.

E-mail: gjtbgf@g.uos.ac.kr

Keywords: tree-based machine learning, graphene detection, pixel-wise, without GPU, segmentation

Abstract

Mechanical exfoliation of graphene and its identification by optical inspection is one of the milestones in condensed matter physics that sparked the field of two-dimensional materials. Finding regions of interest from the entire sample space and identification of layer number is a routine task potentially amenable to automatization. We propose supervised pixel-wise classification methods showing a high performance even with a small number of training image datasets that require short computational time without GPU. We introduce four different tree-based machine learning (ML) algorithms—decision tree, random forest, extreme gradient boost, and light gradient boosting machine. We train them with five optical microscopy images of graphene, and evaluate their performances with multiple metrics and indices. We also discuss combinatorial ML models between the three single classifiers and assess their performances in identification and reliability. The code developed in this paper is open to the public and will be released at github.com/gjung-group/Graphene_segmentation.

1. Introduction

The fact that a few layers of graphene sheet can be prepared by simple mechanical exfoliation [1, 2] has facilitated a rapid growth of graphene and other two-dimensional (2D) van der Waals (vdW) materials research. In particular, graphene has been studied in a wide range of applications in recent years due to its unique electrical, mechanical, and optical properties [3–10]. The recent discovery of a robust unconventional superconductivity in twisted graphene systems [11–15] has reinvigorated research in graphene and other 2D vdW materials [16–21].

After the mechanical exfoliation of 2D materials, the number of layers of graphene or other vdW materials can be identified by various techniques including atomic force microscopy [22, 23], Raman spectroscopy [22, 24], or optical microscopy (OM) [25–28]. Among them, the most commonly used method is based on the optical contrast between single and multilayer graphene layers with different thicknesses in RGB color space of OM images of materials placed on a substrate of specific thickness [25–28]. However, it is a time consuming process to process more than $\sim 10^3$ scanned OM images to identify the interesting few layers exfoliated flakes regions deposited on the substrate. Here we propose a practical machine learning (ML) image recognition method that can be used to quickly identify specific target few layers graphene regions.

Traditionally, ML based techniques have been applied successfully in many different fields of industrial applications or services requiring repetitive human labor. Especially, image classification using deep learning (DL) has emerged as a game changer technique that has allowed drastic reduction of analysis time from hours to seconds. For example, the convolutional neural network (CNN) has been used in biomedical fields,

such as abdominal CT scan, cell, hippocampus, and pancreas segmentations [29–32], and in analyzing big image data obtained from satellites [33, 34] providing significant aid to error-prone human eyes [35–38].

Recent works have used ML techniques to identify the number of layers in a thin film of materials. These researches have employed supervised learning such as support vector machines [39, 40], deep neural network (DNN) [41–43], U-Net which belongs to CNN [44–46], or unsupervised learning such as clustering [41]. The image classifications in the earlier works, however, require too many images for the training dataset that need to be labeled accordingly with layer number, for example, $\sim 10^3$–10^5 labeled images for DNN [41–43], and $\sim 10^2$ [45, 46] or 10^3 labeled images for U-Net which are augmented from less than 50 labeled images [44], and more than a dozen GB of GPU memory to process a batch of image data.

In this paper, we suggest a GPU-free classification tool for detecting a specific number of graphene layers using supervised ML algorithms requiring only a few OM images for both training and testing thanks to its operation in units of a pixel. We compare the performance of four different tree-based ML algorithms such as decision tree (DT), random forest (RF), extreme gradient boosting (XGBoost), and light gradient boosting machine (LightGBM) in terms of accuracy, precision, recall, F1 score, and indices such as receiver operating characteristic (ROC) curve, area under curve (AUC), intersection over union (IOU). We take account of several combinations between RF, XGB, and LGBM entailing improvements in multiple metrics compared to using a single algorithm. Our source code was built with scikit-learn [47] ML library and is provided as open-source (See [48]).

This paper is organized as follows. In section 2 we first describe the sample preparation for OM images, feature extraction using several filters to pre-treat the images for ML algorithms. We elaborate on the four different tree-based ML algorithms that we employed in this work in section 3. In section 4, we evaluate and discuss the performances of a single tree-based ML algorithm and combinations for fusions of them using the several metrics. In section 5 we present the conclusions.

2. Data preprocessing

In the following we describe the data preprocessing techniques used for this work that consists in dataset preparation from the OM images obtained from the experimental system and feature extraction from the images by using different filters.

2.1. Dataset preparation

Many 2D materials can be produced in a large area $\sim cm^2$ using chemical vapor deposition (CVD) that also show contrasts in color depending on thickness [49–51] but the small area $\sim \mu m^2$ exfoliation method is still needed to obtain higher-quality 2D materials by avoiding contamination in the process of transferring it from the growth substrate [52, 53]. The graphene samples in the present work were exfoliated using scotch tapes and placed on top of 285 nm SiO_2/Si substrates as shown in figure 1(a). We use a motorized microscope as seen in figure 1(b) to scan the 20 × magnified images with 1832 × 1372 resolution. We scan the entire sample along the path indicated by the black arrow as shown in figure 1(c). Afterwards, we get scanned OM images like figure 1(d). When the scan is complete, we can get more than 1500 images per one substrate. Among these images, we first sorted out the candidates expected to have exfoliated graphene through optical contrast analysis, and confirmed the thickness of graphene using Raman spectral analysis.

To improve the speed of image preprocessing and classification, images were resized with 458 × 343 pixels which are moderately large for finding graphene. We train our model with five OM images pixel by pixel, leading to 785 470 (=458 × 343 × 5) pixels for the training, and test each pixel of 1 OM image from the test dataset, which is equivalent to 157 094 (=458 × 343 × 1) pixels. The mask for labeling was made manually using the APEER platform [54].

2.2. Feature extraction

Feature extraction is one of the most important key elements in ML. Here we use the HSV color space and several commonly used filters that help us extract relevant features to recognize and classify the monolayer graphene from the background as we describe in the following. There are several color bases for generating arbitrary colors that we are familiar with, such as RGB (red, green, blue) and CMYK (cyan, magenta, yellow, black). However, here we introduce the HSV (hue, saturation, value) color space, which is a familiar way for humans to perceive color. The HSV color space describes colors with their hue (H) together with saturation (S), namely, amount of gray, and value (V) of brightness or luminous intensity as illustrated in figures 2(b)–(d). Each pixel of the graphene images obtained from the motorized microscope which were originally expressed in RGB color space are transformed to HSV [55, 56].

We use several filters, the Prewitt, Scharr, and Gaussian filters in order to extract the relevant features. We use the grayscale images for this process as shown in figure 2(e) for the following two reasons. Firstly, the

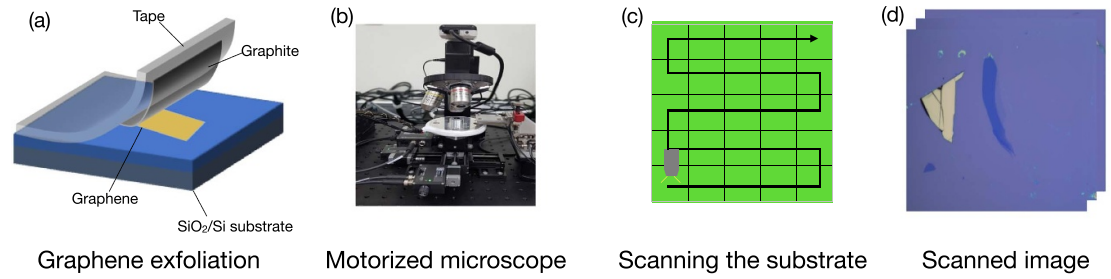

Figure 1. (a) Schematic diagram for exfoliation of graphene from graphite using the scotch tape on top of the substrate SiO_2/Si. (b) Motorized microscope for scanning the sample. (c) Schematic diagram for the entire sample is denoted by the green panel, the path indicated by the black arrow for scanning the sample to find the graphene, and the camera of the motorized microscope illustrated by the gray square. (d) Scanned images by the motorized microscope along with the black arrows in (c).

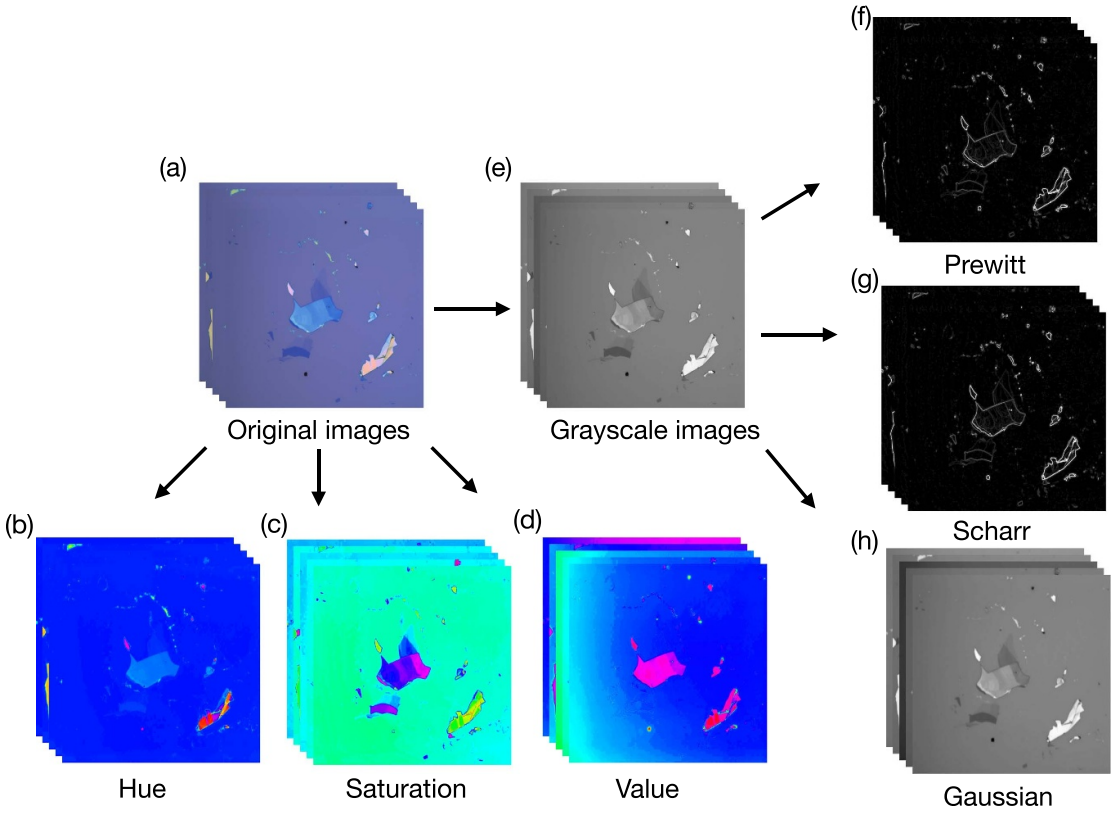

Figure 2. Flowchart for data preprocessing for ML feature extraction. (a) Original images from motorized spectroscopy. (b) Map for the hue (H) component of the original images. (c) Map for the saturation (S) component of the original images. (d) Map for the value (V) component of the original images. (e) Images in grayscale. (f) Resultant map of the Prewitt edge filter for the grayscale images. (g) Resultant map of the Scharr edge filter. (h) Resultant map of the Gaussian filter.

grayscale images preserve the essential information such as edge, shape, and texture information from their original RGB representation. Secondly, we can reduce complexity and unnecessary computational cost [57]. An edge of any object in images can be defined as a boundary where the value of brightness changes discontinuously, leading to a very sharp change in the associated intensity gradient. Hence, we can find abrupt changes in the intensity values for each pixel over the entire image given in HSV space, and find spots where its derivative is maximum. We utilize the 3×3 Prewitt, and Scharr operators for the x (vertical) and y (horizontal) direction as defined below [58, 59]

$$\text{Prewitt}_x = \frac{1}{3}\begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix},$$

$$\text{Prewitt}_y = \frac{1}{3}\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}, \tag{1}$$

and

$$\text{Scharr}_x = \frac{1}{32}\begin{pmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{pmatrix},$$

$$\text{Scharr}_y = \frac{1}{32}\begin{pmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{pmatrix}. \tag{2}$$

As an example, we present the resultant map for an image processed through the Prewitt and Scharr filters in figures 2(f) and (g), respectively. After the convolution of the two operators with our grayscale image, we get resultant F_x and F_y matrices of the same size as our source image. Hence, the magnitude for each pixel is given by

$$\text{Magnitude} = \sqrt{F_x^2 + F_y^2}. \tag{3}$$

Afterwards, we use the Gaussian smoothing kernel based on the 2D Gaussian function of equation (4).

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}. \tag{4}$$

The Gaussian kernel [60] raises the quality of our model by blurring redundant noises as illustrated in figure 2(h). For example, the 5×5 Gaussian kernel for $\sigma = 1$ is

$$G_{5\times5}^{\sigma=1} = \frac{1}{273}\begin{pmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{pmatrix}. \tag{5}$$

3. Theoretical frameworks

In the following, we briefly summarize the theoretical footing of the four tree-based ML classification algorithms employed in the current work. The simplicity and efficiency of these models are at the heart of the practical applicability our model to reduce the number of training images and shorten the classification time. This section is devoted mostly to assess the training and validation of the more advanced extreme gradient boost and LightGBM models, and is introduced mainly for the benefit of the readership specialized in materials science.

3.1. Decision tree

The DT is one of the basic algorithms in ML, that as its name indicates infers the predicted label following a tree-like decision framework. Firstly, we have a root (a single leaf) to begin with and we assign a label to this root according to a majority vote among all the labels over the training set. For example, we can imagine that the label could be the length of the petal in the famous classification problem for Iris flowers. Then, we can split the root into two groups depending on whether or not the dataset satisfies the label and evaluate the effect of splitting over the iterations by calculating a measure which is called gain. The gain quantifies the improvement of the performance of our model due to the splitting. Among the possible splits, we can either choose the one that maximizes the gain or choose not to split the leaf. The merit of this algorithm is that we can see the procedures of decision-making. However, the major disadvantage of DT is the high risk of over-fitting the training set and lack of tolerance to non-representative single-node outliers [61–63].

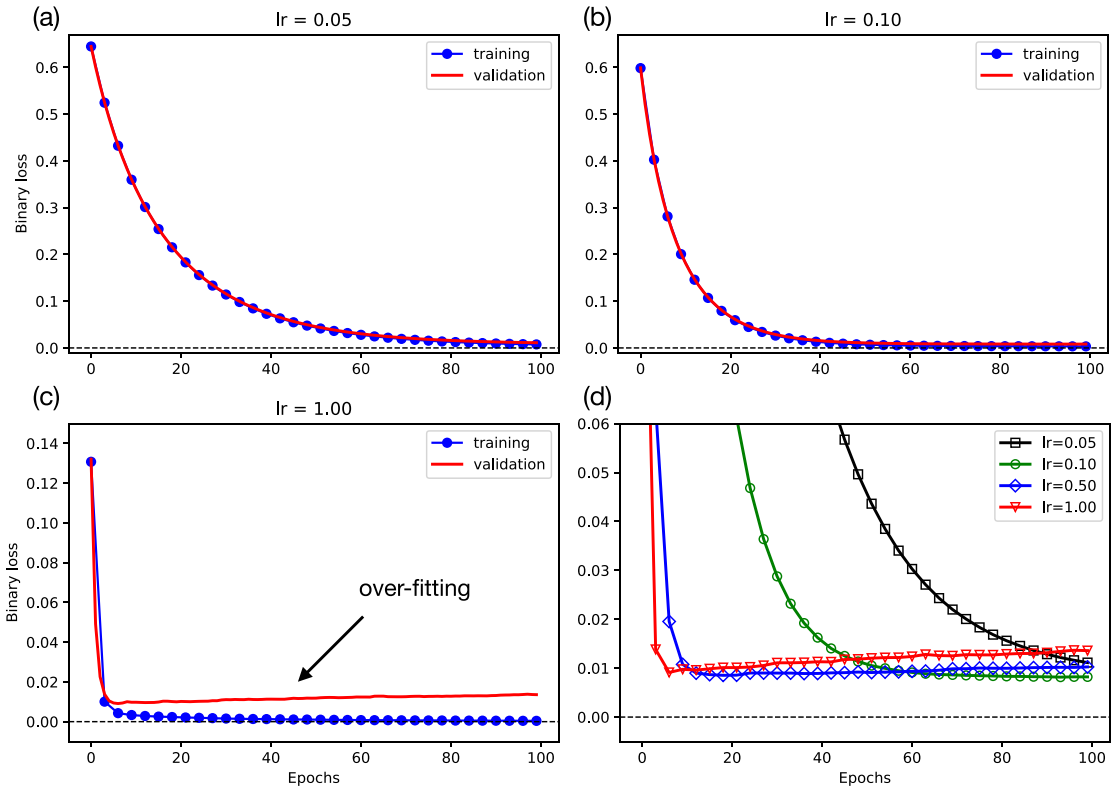

Figure 3. Training (blue filled circles) and validation (red solid line) loss in XGB algorithm using the binary cross-entropy loss function for the learning rates (a) lr = 0.05, (b) 0.1, and (c) 1.0. (d) Validation losses for four different learning rates, lr = 0.05 (black squares), 0.10 (green circles), 0.5 (blue diamonds), and 1.0 (red triangles). XGB is over-fitted with the training dataset when the learning rate is larger than lr ≈ 0.5.

3.2. Random forest

The RF is literally an ensemble of the DTs with randomness such that it generates multiple DTs that chooses a class by majority vote among the trees and aggregates them by using their average for a regression. Therefore, RF is less vulnerable to over-fitting than DT. However, the decision-making could slow down as the size of the dataset increases [64–66].

3.3. XGBoost

The extreme gradient boost (XGB or XGBoost) is a scalable end-to-end ML algorithm which was proposed by Chen and Guestrin in 2016 [67, 68]. XGB is based on a gradient boosting decision tree (GBDT). GBDT combines base-learners (e.g. DT) into a single strong learner over the many iterations by fitting the base-learners using a loss function (e.g. the mean squared error). The aim of gradient boosting is to train the model to minimize the loss function using the functional gradient descent which leads the next iteration toward the direction of negative gradient. While GBDT uses only the first-order derivatives of the loss function, XGB utilizes the Newton–Raphson method for the functional gradient. Namely, the second-order derivatives of the loss function are used in the fitting procedure. Consequently, XGB proposes a newly distributed algorithm for tree searching, outperforming RF in general, with the drawback that it is generally more time consuming [69].

Figure 3 shows the training and validation losses in XGB using the binary cross-entropy loss function

$$L_{\log} = \frac{1}{N}\sum_{i=0}^{N-1} -(y_i \log(p_i) + (1-y_i)\log(1-p_i)) \tag{6}$$

as implemented in scikit-learn [47] as a function of the epochs for the three different learning rates (lr) where $y \in \{0, 1\}$ is the true label and $p = \Pr(y = 1)$ is the probability and the i index runs over all sample points. The training loss decreases and approaches zero for all cases as the epoch increases. For lr = 0.05 and 0.10 the validation loss decreases monotonically, while the validation loss has a local minimum at the epoch = 16 and grows as the epoch increases for lr = 1.0, implying that the model is over-fitted with the training dataset. Our model starts over-fitting when lr is larger than ≈ 0.5 as shown in figure 3(d). Thus, in this paper, we use lr = 0.1 for XGB unless stated otherwise.

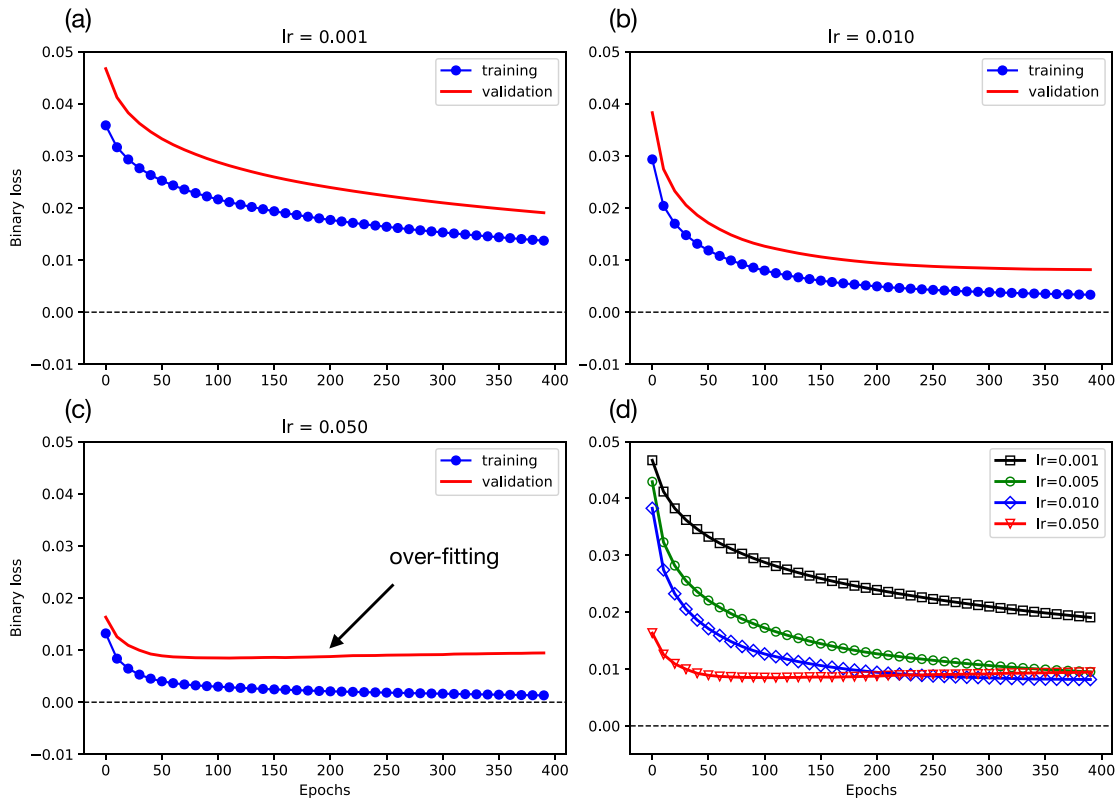

Figure 4. Training (blue filled circles) and validation (red solid line) loss in LGBM algorithm using the binary cross-entropy loss function for the learning rates (a) lr = 0.001, (b) 0.01, and (c) 0.05. (d) Validation losses for four different learning rates, lr = 0.001 (black squares), 0.005 (green circles), 0.01 (blue diamonds), and 0.05 (red triangles). LGBM is over-fitted with the training dataset as the learning rate increases more than lr ≈ 0.05.

3.4. LightGBM

The LightGBM (LGBM or LightGBM) also originates from GBDT and inherits many strengths from XGB. However, LGBM has two significant technical differences from XGB in structuring trees which are called gradient-based one side sampling (GOSS) and exclusive feature bundling (EFB) [70].

The GOSS is a sampling tool that keeps the instances with large gradients but randomly drops out some portion of the ones with small gradients. In EFB two sparse features which are nearly exclusive integrate into one to reduce the number of instances. In short, GOSS and EFB reduce the number and the size of data instances remarkably while at the same time keeping high accuracy. Consequently, LGBM is faster and requires less memory, as its name suggests 'light'. However, LGBM is prone to over-fitting when the size of the dataset is small [71].

We show the training and validation losses for different learning rates lr = 0.001, 0.01, and 0.05 in figures 4(a)–(c) using the binary cross-entropy loss function. One can see that LGBM model for our graphene OM image datasets requires to be stopped early at the epoch = 20 to avoid over-fitting where the validation loss starts increasing when lr = 0.05. Compared to other learning rates, the loss starts increasing upon the epochs when lr $\gtrsim$ 0.05 as shown in figure 4(d). Therefore, we choose lr = 0.005 for LGBM throughout this paper.

4. Results and discussions

4.1. Single classifiers

We train the ML models for each pixel in an OM image with each one of the four classification models and use the five-fold stratified cross-validation to maintain objectivity [72]. Thus, we obtain the probability of having a monolayer graphene sheet at each pixel, and round off the probability from the first decimal place so that we assign 0 to pixels predicted as background, and 1 to the ones predicted as monolayer graphene at each pixel. In figure 5 we visualize our results for the four tree-based ML algorithms together with the original OM images and the target images labeled for the monolayer graphene. The graphene monolayers in the OM images in figure 5(a) are enclosed by the red dashed line. As shown in figure 5(b), the pixels for monolayers are labeled as 1 (white) and the others are labeled as 0 (black). Figure 5(c) shows the

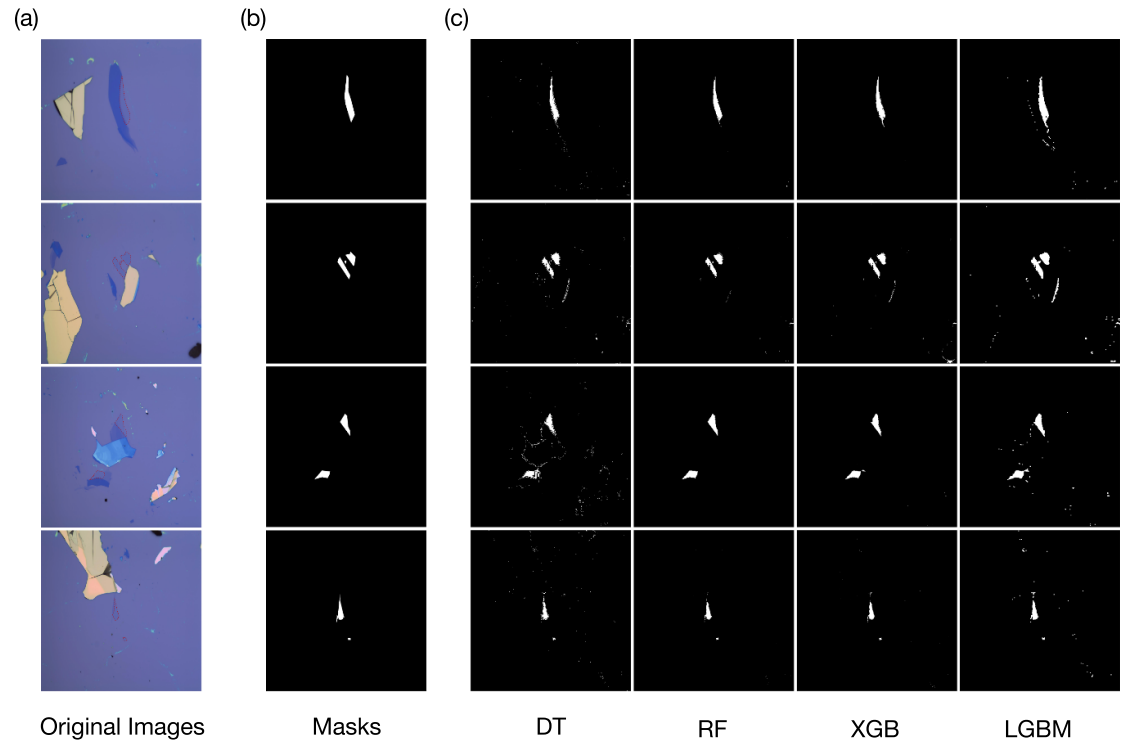

Figure 5. (a) Source images of the domain of interest. The flakes of monolayer graphene are enclosed by red dashed lines. (b) The flakes of monolayer graphene are labeled as 1 (white) and the rest is labeled as 0 (black). (c) Resultant images of pixel-wise classifications for the four algorithms, DT, RF, XGBoost (XGB), and LightGBM (LGBM).

classification results from DT, RF, XGB, and LGBM algorithms, respectively. Each algorithm appears to well differentiate the monolayer graphene from the OM image. Since background pixels are dominant, it leads to an highly imbalanced pool of pixels. Therefore we introduce several metrics and indices to evaluate performance of the model

First, the accuracy as defined in equation (7) is the most basic metric to evaluate the performance of a classification ML model.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{TN} + \text{FP}}, \tag{7}$$

where TP (TN) stands for the case where our model correctly predicts a graphene (background) pixel, while FP (FN) represents the opposite case where our model incorrectly predicts a background (graphene) pixel as a graphene (background) pixel. The precision which is defined as equation (8) is obtained by evaluating the ratio of correctly predicted graphene pixels. The higher precision score means that there are more actual graphene pixels among the pixels predicted as graphene by our model. If the precision is lower, the portion of background pixels classified as graphene is higher.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{8}$$

Similarly, we define the recall in equation (9). If the recall value is high, a graphene pixel is less likely to be classified as a background pixel.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{9}$$

Lastly, the F1 score in the equation (10) is nothing more than the harmonic mean between the precision and the recall.

$$\text{F1 score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{10}$$

The evaluation of the four tree-based classification algorithms using the four metrics are shown in table 1. The accuracies for the four algorithms are more or less the same but DT has the smallest accuracy

Table 1. Evaluations of the four different tree-based classification algorithms, DT, RF, XGBoost (XGB), and lightGBM (LGBM) using the four metrics such as accuracy, precision, recall, and F1 score defined in equations (7)–(10).

Algorithm \ metric	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
DT	99.68	73.70	77.92	75.36
RF	99.79	89.61	76.54	82.09
XGB	99.72	88.01	75.26	80.80
LGBM	99.77	91.23	73.37	80.96

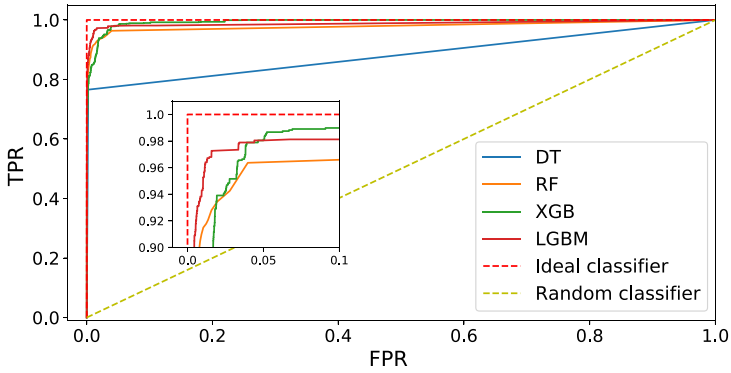

Figure 6. The ROC curves for random classifier (yellow dashed), ideal classifier (red dashed), DT (blue solid), RF (orange solid), XGB (green solid), and LGBM (pink solid). The inset shows the same ROC curve for the finer range of false positive range (FPR) in equation (11) near 0.0 and the true positive range (TPR) near 1.0.

compared to others. DT also has far less precision around ∼74% in contrast with the other methods which are above ∼88%. LGBM has the highest precision ∼91%. Regarding the recall values, on the other hand, DT scores above ∼78% which is superior to others. The RF, XGB, and LGBM record around ∼81%–82% in F1 score, while DT is lower by around 6% with respect to the other methods. The color contrast between the graphene flakes and the background is low because a graphene sheet is nearly transparent. This inherent limitation can lead to suppressed F1 score by wrongly identifying the background pixels around the edges of graphene as graphene itself. However, it has been noted that ML models work well for finding graphene flakes even when the F1 score is less than 80% [46]. In figure 5, our DT model with the lowest F1 score of 75.36% among our ML models also finds the graphene flakes well. The DT model misclassifies some pixels, but can clearly recognize the presence of monolayer graphene. Other models with an F1 score of 80.8 or higher have smaller errors and detect the graphene flakes even better.

We further employ several additional performance measures such as the ROC curves, their AUC, and the IOU. We show the ROC curve in figure 6 as a function of the true positive rate (TPR) which stands for the recall, and the false positive rate (FPR) which is defined as

$$\text{False Positive Rate (FPR)} = \frac{\text{FP}}{\text{FP} + \text{FN}}. \tag{11}$$

We get these ROC curves by changing the criteria of rounding off the probability between 0 (background) and 1 (graphene) for each pixel obtained from each ML model, leading to different values of TPR and FPR. The AUC for the ideal classifier whose ROC is denoted by the red dashed line is 1 using normalized units, whereas the AUC for the random classifier denoted by the yellow dashed line is $1/2$ as shown in figure 6. The AUCs for the four different classification algorithms which range between $1/2$ and 1 are presented in table 2. Their AUCs have an order such that DT $\ll$ RF $<$ LGBM $<$ XGB. One can see in the inset of figure 6 in details that XGB has a smaller TPR than RF and LGBM at nearly zero FPR but it starts to surpass the others when FPR $\gtrsim 0.05$.

To evaluate the performance of the pixel-wise segmentation method we use the IOU index defined as

$$\text{IOU} = \frac{\text{Target Pixel} \cap \text{Predicted Pixel}}{\text{Target Pixel} \cup \text{Predicted Pixel}}. \tag{12}$$

This index indicates how many pixels overlap between the target pixels and the predicted pixels and returns a value in the range between 0 and 1. The IOU values for the four algorithms are presented in table 3 and they have an order of DT $\ll$ RF $\ll$ LGBM $<$ XGB which indicates that the XGB is the most suitable algorithm for

Table 2. Area under the curve (AUC) score for the four tree-based ML algorithms such as DT, RF, XGBoost (XGB), and LightGBM (LGBM).

Algorithm	AUC
DT	0.8731
RF	0.9792
XGB	0.9942
LGBM	0.9884

Table 3. Evaluation of the IOU score for the four tree-based ML algorithms such as DT, RF, XGBoost (XGB), and LightGBM (LGBM).

Algorithm	IOU Score (%)
DT	54.61
RF	63.86
XGB	71.18
LGBM	70.73

Table 4. The inference time per image for the four tree-based ML algorithms such as DT, RF, XGBoost (XGB), and LightGBM (LGBM).

Algorithm	Inference time (s)
DT	0.1185
RF	0.1413
XGB	0.2822
LGBM	0.2006

pixel-wise segmentation of our datasets. When the IOU score exceeds 0.5 an object is regarded as detected [73]. In this sense all the models we have proposed are able to successfully detect graphene flakes.

The RF classifier has the highest accuracy and F1 score, but its pixel-resolved segmentation shows a relatively low IOU score that is slightly higher than the DT. On the other hand, the XGB has the highest IOU score but less accuracy, precision, recall, and F1 score than RF. The LGBM has higher precision and IOU score than RF. In practice we can choose the classifier depending on which metric or index is more relevant for the classification task at hand or mix more than one classifier as we will explain in the next section.

Regarding the inference time, as shown in table 4, the DT algorithm is the fastest model with a computation time of ∼0.1 s per image, while the XGB is the slowest one at ∼0.3 s per image using Intel(R) Core(TM) i5-10 400 CPU. In practice, this means that we can analyze a thousand images in a matter of minutes regardless of the method used.

4.2. Fusions between classifiers

In this section, we combine the different single classifiers RF, XGB, and LGBM in an attempt to improve the performance of our model as proposed in [74] that calculates different weights for each classifier depending on metrics accuracy. We will show that it is possible to improve the overall performance of different metrics of our choice at the expense of slightly reducing the IOU. We exclude the DT as it scores far fewer points particularly in the precision, F1 score, AUC and IOU. We predict to find a monolayer graphene pixel with the probability P defined in equation (13) for each pixel in an OM image when we utilize the combined probability consisting of N single classifiers through

$$P = \sum_{i}^{N} w_i P_i, \tag{13}$$

where P_i stands for the probability of finding a monolayer graphene at that pixel using the ith single classifier, and w_i is the ith classifier's weight which is given as [74]

$$w_i = \frac{\text{metric}(i)^{10}}{\sum_{j=1}^{N} \text{metric}(j)^{10}}, \tag{14}$$

where metric(i) represents the ith single classifier's metric. In other words, the probability P of a joint classification model is defined such that the probabilities of finding a monolayer graphene at that pixel using single classifiers are expressed as a linear combination of each classifier's probability multiplied by the respective weight coefficients. Like in the single classifiers, we round off the final probability to either 0 (background) and 1 (graphene). Then we use P at each pixel to calculate the metrics as presented in table 5.

Table 5. Algorithm choice versus metric table for five different combinations of the three single classification algorithms RF, XGBoost (XGB), and lightGBM (LGBM) using the five metrics accuracy, precision, recall, F1 score, and IOU score. In parentheses below each one of the fused-classifiers, we specify the metrics used to optimize the weights of the classifiers. We listed several metrics when their performances were the same down to two decimal places.

Algorithm \ metric	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)	IOU score (%)
RF + XGB (precision, recall, F1 score)	99.77	89.36	72.95	79.96	66.82
RF + LGBM (precision)	99.78	91.23	73.37	80.96	66.82
RF + LGBM (recall, F1 score)	99.77	89.36	72.95	79.96	66.82
XGB + LGBM (precision, F1 score)	99.78	91.23	73.37	80.96	68.22
XGB + LGBM (recall)	99.77	88.01	75.26	80.80	67.98
RF + XGB + LGBM (precision, recall, F1 score)	99.78	90.64	73.82	81.03	68.28

We take into account four combinations, RF + XGB, RF + LGBM, XGB + LGBM, RF + XGB + LGBM. We calculate the weights w_i for the ith single classifier as defined in equation (14) taking one of the four metrics—accuracy, precision, recall, and F1 score. We indicated within parentheses the reference metric used to calculate the weights of the classifiers, and listed more than one when they gave the same results down to two decimal places. We exclude the accuracy as a weight because the accuracies of RF, XGB, and LGBM are all on the order of ∼99.7%.

The single classifier RF has a higher accuracy of 99.79%, recall of 76.54%, and F1 score of 82.09%, but it has a smaller IOU score of 63.86% than XGB and LGBM. If we join XGB with RF, regardless of which metric we use, we get an improved IOU score by ∼3%, but sacrificing accuracy by 0.01%, precision by 0.25%, recall by ∼3.6%, and the F1 score by ∼2.1% compared to using the single RF classifier. On the other hand, we get improved accuracy by 0.05%, precision by ∼1.4%, sacrificing recall by 2.3%, F1 score by ∼0.8%, and the IOU score by ∼4.4% compared to using the single XGB classifier.

If we join LGBM with RF, the results are different depending on the metrics we use for a weight. In the case of precision as a weight, it has ∼1.6% higher precision, ∼3% higher IOU score than a single RF classifier. If we use either recall or F1 score as a metric, it has benefit in IOU score by ∼3% compared to the single RF. On the other hand, combining LGBM with RF has no merit in comparison with the single LGBM.

If we combine XGB and LGBM, in the case where either precision or the F1 score is chosen as a weight, the accuracy is improved to 99.78%, and the IOU hits 68.22% which is less than both XGB (71.18%) and LGBM (70.73%) separately. The precision (91.23%), recall (73.37%), and F1 score (80.96%) are the same as those of a single classifier of LGBM. When we use the recall as a weight, the IOU hits 67.98% which is again smaller than that of single XGB and LGBM classifiers. The accuracy is the same as that of LGBM (99.77%), but the precision (88.01%), recall (75.26%), and F1 score (80.80%) have the same value as those of XGB.

Lastly, if we combine all three classifiers, all metrics and indices are overall averaged regardless of the metrics used for the weights. The accuracy and F1 score are improved in comparison with single XGB and LGBM classifiers, and the precision becomes higher than single RF and XGB. The recall becomes higher than a single LGBM, and the IOU is improved compared to a single RF.

The resulting fused-classifier metrics presented in table 5 indicates that we can indeed improve the overall performance of our models if we target a particular metric to optimize, and at times it is possible to achieve an overall improvement in several metrics at the expense of a small decrease in the IOU by a few percents with respect to the XGB and LGBM single classifier scores.

5. Conclusions

The preparation of graphene electronic devices during almost the last two decades has relied on human inspection during many hours of routine scanning in the entire space of samples to identify the number of graphene layers deposited on the substrate. Recently several research teams have attempted to introduce ML in this process to reduce such time-consuming and error-prone human labor. However, the existing proposals are mostly focused on developing DNN or CNN based models which require quite a large number

of labeled training dataset, and this threshold makes practical application difficult. In this paper, we have proposed a pixel-wise tree-based ML classification tool that can be used for samples prepared under consistent scanning conditions such as illumination or the thickness of the substrate. Our method can distinguish a specific number of graphene layers from the background and performs precisely even with a small number of training images, in contrast to existing works utilizing DL tools that require hundreds to thousands of image data to train a model.

We have trained our ML model with four different tree-based algorithms such as DT, RF, XGB, and LGBM, and examined their outcomes using several metrics and indices—accuracy, precision, recall, F1 score, ROC, AUC, and IOU. There is no absolute best model among the single classifiers as they have different strengths and weaknesses in metrics and indices, and those metrics vary depending on the data that we deal with. We have shown that it is possible to improve the performance by combine more than one classifier. Considering overall performance, we propose that the LGBM or XGB are good choices when using a single classifier, and for the fused classifiers model, the RF + XGB + LGBM shows satisfactory performance and can be chosen in routine applications.

In this work, we have sorted out the monolayer graphene from the OM images, but the same process can be performed to find any other number of multilayer graphene or for any other 2D vdW materials such as hexagonal boron nitride, transition metal chalcogenides or black phosphorus. We expect our method to be applicable to distinguish different layer thickness also in CVD grown samples provided that their colors are distinguishable. Furthermore, our model requires only a few images and costs less computational time than a few minutes in total to train a ML model even without using a GPU. Therefore, we expect our work will be of immediate utility for researches on 2D materials, and will greatly ease the repetitive and time-consuming tasks in experiments.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: github.com/gjung-group/Graphene_segmentation.

Acknowledgments

This work was supported by Korean NRF through the Grant Nos. 2020R1A5A1016518 (W H C), 2021R1A6A3A01087281 (J S), 2020R1A2C3009142 (J J). Y D K was supported by the KIST Institutional Program (Project No. 2E31781-22-108). We acknowledge computational support from KISTI Grant No. KSC-2021-CRE-0389 and by the computing resources of Urban Big data and AI Institute (UBAI) at UOS and the network support from KREONET. We also acknowledge partial support for W H C from the Korean Ministry of Land, Infrastructure and Transport (MOLIT) from the Innovative Talent Education Program for Smart Cities.

Appendix A. Flowchart for data preprocessing in detail

In this appendix, we provide complimentary figures for figure 2 that shows the data preprocessing flow diagram of our ML model in detail. We unfold the case by case to explicitly show the OM images and their resulting figures at each step in figure 7.

Appendix B. Image processing for a better segmenation using marching squares algorithm

In this appendix, we explain how to resample the data in order to reduce the likelihood of false graphene flakes detection. During pixel-wise segmentation we can detect both large graphene flakes as well as small graphene fragments. The metrics drop mainly due to predition errors in small graphene fragments in areas not set as a mask. In order to reduce the errors in graphene flakes detection we created contours using the marching squares algorithm [75, 76] as shown in figure 8 and have sampled the regions within contours that exceed a given threshold size for final segmentation. We show in table 6 the improved evaluation metrics when the threshold sizes in the x- and y-axes are set as 10 μm where all metrics have increased on average as follows: Accuracy increased ∼0.09%, Precision increased ∼2.59%, Recall increased ∼8.37%, and F1 score increased ∼6.14%.

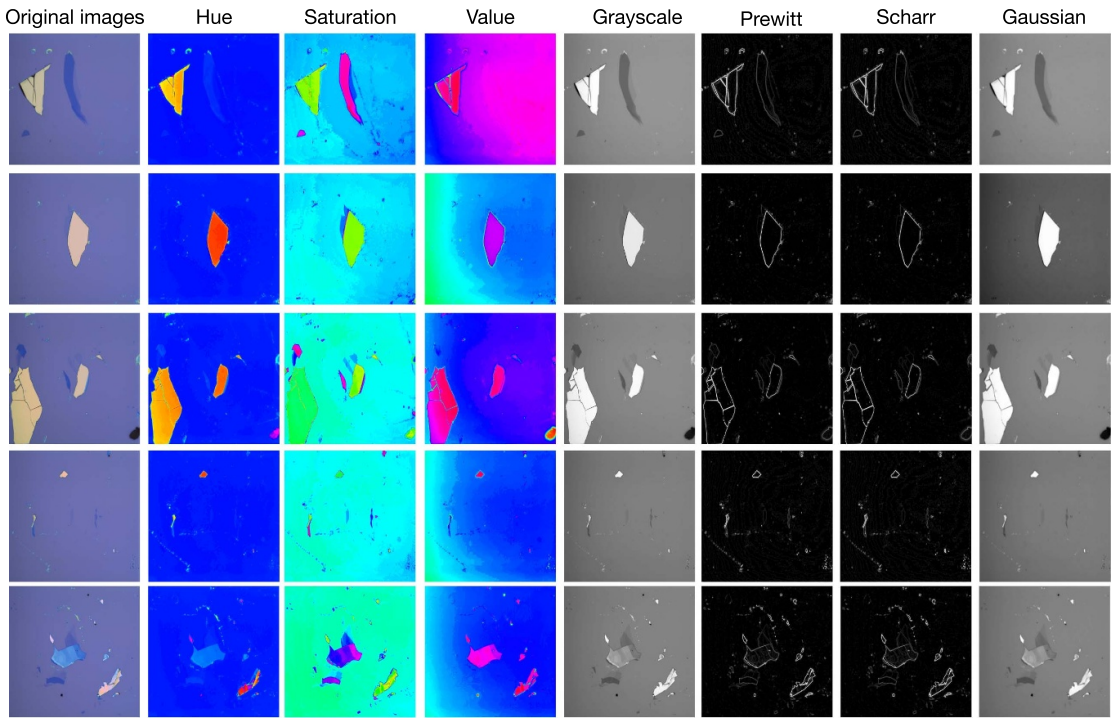

Figure 7. Spread flowchart for data preprocessing for ML feature extraction. Here we show five OM images (original), each component in HSV color space, in grayscale, resultant figures from Prewitt edge, Scharr edge, and Gaussian filters.

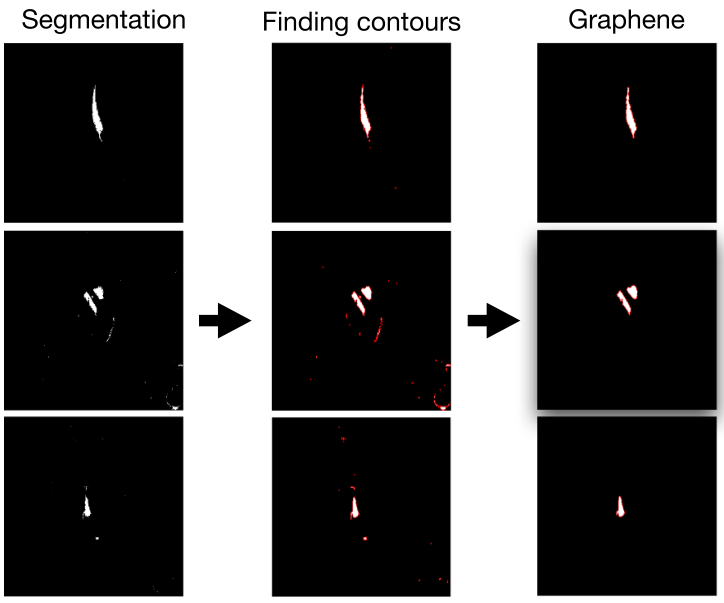

Figure 8. Flow chart for better segmentation using marching squares algorithm. After performing a segmentation, we found contour lines of prediction dots, and we selected the contour lines that are larger than the threshold size.

Table 6. After completing image processing using the marching squares algorithm, we evaluated four different tree-based classification algorithms using four metrics. All evaluation metrics are better than before performing image processing.

Algorithm \ metric	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
DT	99.79	82.72	85.43	83.84
RF	99.83	90.22	82.44	86.01
XGB	99.83	88.79	84.41	86.46
LGBM	99.85	91.16	84.30	87.46

ORCID iDs

Woon Hyung Cho https://orcid.org/0000-0003-2629-8370
Jiseon Shin https://orcid.org/0000-0002-1756-0536
Young Duck Kim https://orcid.org/0000-0003-2593-9826
George J Jung https://orcid.org/0000-0003-2523-0905

References

[1] Novoselov K S, Geim A K, Morozov S V, Jiang D-eng, Zhang Y, Dubonos S V, Grigorieva I V and Firsov A A 2004 Electric field effect in atomically thin carbon films *Science* **306** 666–9
[2] Yi M and Shen Z 2015 A review on mechanical exfoliation for the scalable production of graphene *J. Mater. Chem.* A **3** 11700–15
[3] Castro Neto A H, Guinea F, Peres N M R, Novoselov K S and Geim A K 2009 The electronic properties of graphene *Rev. Mod. Phys.* **81** 109
[4] Koppens F H L, Mueller T, Avouris P, Ferrari A C, Vitiello M S and Polini M 2014 Photodetectors based on graphene, other two-dimensional materials and hybrid systems *Nat. Nanotechnol.* **9** 780–93
[5] Geim A K and Novoselov K S 2007 The rise of graphene *Nat. Mater.* **6** 183–91
[6] Lee C, Wei X, Kysar J W and Hone J 2008 Measurement of the elastic properties and intrinsic strength of monolayer graphene *Science* **321** 385–8
[7] Cao K, Feng S, Han Y, Gao L, Ly T H, Xu Z and Lu Y 2020 Elastic straining of free-standing monolayer graphene *Nat. Commun.* **11** 284
[8] Bonaccorso F, Sun Z, Hasan T and Ferrari A C 2010 Graphene photonics and optoelectronics *Nat. Photon.* **4** 611–22
[9] Geim A K and Grigorieva I V 2013 Van der Waals heterostructures *Nature* **499** 419–25
[10] Ajayan P, Kim P and Banerjee K 2016 Van der Waals materials *Phys. Today* **69** 38
[11] Bistritzer R and MacDonald A H 2011 Moiré bands in twisted double-layer graphene *Proc. Natl Acad. Sci.* **108** 12233–7
[12] Cao Y, Fatemi V, Fang S, Watanabe K, Taniguchi T, Kaxiras E and Jarillo-Herrero P 2018 Unconventional superconductivity in magic-angle graphene superlattices *Nature* **556** 43–50
[13] Cao Y *et al* 2018 Correlated insulator behaviour at half-filling in magic-angle graphene superlattices *Nature* **556** 80–84
[14] Park J M, Cao Y, Watanabe K, Taniguchi T and Jarillo-Herrero P 2021 Tunable strongly coupled superconductivity in magic-angle twisted trilayer graphene *Nature* **590** 249–55
[15] Hao Z, Zimmerman A M, Ledwith P, Khalaf E, Najafabadi D H, Watanabe K, Taniguchi T, Vishwanath A and Kim P 2021 Electric field–tunable superconductivity in alternating-twist magic-angle trilayer graphene *Science* **371** 1133–8
[16] Chhowalla M, Shin H S, Eda G, Li L-J, Loh K P and Zhang H 2013 The chemistry of two-dimensional layered transition metal dichalcogenide nanosheets *Nat. Chem.* **5** 263–75
[17] Novoselov K S, Jiang D, Schedin F, Booth T J, Khotkevich V V, Morozov S V and Geim A K 2005 Two-dimensional atomic crystals *Proc. Natl Acad. Sci.* **102** 10451–3
[18] Zeng H, Zhi C, Zhang Z, Wei X, Wang X, Guo W, Bando Y and Golberg D 2010 White graphenes": boron nitride nanoribbons via boron nitride nanotube unwrapping *Nano Lett.* **10** 5049–55
[19] Watanabe K, Taniguchi T and Kanda H 2004 Direct-bandgap properties and evidence for ultraviolet lasing of hexagonal boron nitride single crystal *Nat. Mater.* **3** 404–9
[20] Samuel Reich E *et al* 2014 Phosphorene excites materials scientists *Nature* **506** 19
[21] Chen P, Li N, Chen X, Ong W-J and Zhao X 2017 The rising star of 2D black phosphorus beyond graphene: synthesis, properties and electronic applications *2D Mater.* **5** 014002
[22] Huang Y, Sutter E, Shi N N, Zheng J, Yang T, Englund D, Gao H-J and Sutter P 2015 Reliable exfoliation of large-area high-quality flakes of graphene and other two-dimensional materials *ACS Nano* **9** 10612–20
[23] Shearer C J, Slattery A D, Stapleton A J, Shapter J G and Gibson C T 2016 Accurate thickness measurement of graphene *Nanotechnology* **27** 125704
[24] Saito R, Hofmann M, Dresselhaus G, Jorio A and Dresselhaus M S 2011 Raman spectroscopy of graphene and carbon nanotubes *Adv. Phys.* **60** 413–550
[25] Blake P, Hill E W, Castro Neto A H, Novoselov K S, Jiang D, Yang R, Booth T J and Geim A K 2007 Making graphene visible *Appl. Phys. Lett.* **91** 063124
[26] Li H, Wu J, Huang X, Lu G, Yang J, Lu X, Xiong Q and Zhang H 2013 Rapid and reliable thickness identification of two-dimensional nanosheets using optical microscopy *ACS Nano* **7** 10344–53
[27] Jessen B S, Whelan P R, Mackenzie D M A, Luo B, Thomsen J D, Gammelgaard L, Booth T J and Bøggild P 2018 Quantitative optical mapping of two-dimensional materials *Sci. Rep.* **8** 6381
[28] Huang F 2019 Optical contrast of atomically thin films *J. Phys. Chem.* C **123** 7440–6
[29] Ronneberger O, Fischer P and Brox T 2015 U-net: convolutional networks for biomedical image segmentation *MICCAI* vol 9351 pp 234–41
[30] Oktay O *et al* 2018 Attention u-net: learning where to look for the pancreas (arXiv:1804.03999)
[31] Vizcaíno A, Sánchez-Cruz H, Sossa H and Quintanar J L 2021 Pixel-wise classification in hippocampus histological images *Comput. Math. Methods Med.* **2021** 6663977
[32] Gao S, Han L, Luo D, Liu G, Xiao Z, Shan G, Zhang Y and Zhou W 2021 Modeling drug mechanism of action with large scale gene-expression profiles using GPAR, an artificial intelligence platform *BMC Bioinform.* **22** 17
[33] Zhao K, Kang J, Jung J and Sohn G 2018 Building extraction from satellite images using mask R-CNN with building boundary regularization *In 2018 IEEE in CVF Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)* pp 247–51
[34] Leon-Garza H, Hagras H, Peña-Rios A, Conway A and Owusu G 2020 A big bang-big crunch type-2 fuzzy logic system for explainable semantic segmentation of trees in satellite images using HSV color space (IEEE) pp 1–7
[35] Li X, Shan G and Shek C H 2022 Machine learning prediction of magnetic properties of fe-based metallic glasses considering glass forming ability *J. Mater. Sci. Technol.* **103** 113
[36] Masubuchi S, Morimoto M, Morikawa S, Onodera M, Asakawa Y, Watanabe K, Taniguchi T and Machida T 2018 Autonomous robotic searching and assembly of two-dimensional crystals to build van der Waals superlattices *Nat. Commun.* **9** 1413

[37] Blum T, Graves J, Zachman M J, Polo-Garzon F, Wu Z, Kannan R, Pan X and Chi M 2021 Machine learning method reveals hidden strong metal-support interaction in microscopy datasets *Small Methods* **5** 2100035
[38] Cheng Z, Wang C, Wu X and Chu J 2022 Review in situ transmission electron microscope with machine learning *J. Semicond.* **43** 081001
[39] Lin X *et al* 2018 Intelligent identification of two-dimensional nanostructures by machine-learning optical microscopy *Nano Res.* **11** 6316–24
[40] Yang J and Yao H 2020 Automated identification and characterization of two-dimensional materials via machine learning-based processing of optical microscope images *Extreme Mech. Lett.* **39** 100771
[41] Masubuchi S and Machida T 2019 Classifying optical microscope images of exfoliated graphene flakes by data-driven machine learning *npj 2D Mater. Appl.* **3** 4
[42] Greplova E, Gold C, Kratochwil B, Davatz T, Pisoni R, Kurzmann A, Rickhaus P, Fischer M H, Ihn T and Huber S D 2020 Fully automated identification of two-dimensional material samples *Phys. Rev. Appl.* **13** 064017
[43] Shin Y J, Shin W, Taniguchi T, Watanabe K, Kim P and Bae S-H 2021 Fast and accurate robotic optical detection of exfoliated graphene and hexagonal boron nitride by deep neural networks *2D Mater.* **8** 035017
[44] Saito Y, Shin K, Terayama K, Desai S, Onga M, Nakagawa Y, Itahashi Y M, Iwasa Y, Yamada M and Tsuda K 2019 Deep-learning-based quality filtering of mechanically exfoliated 2D crystals *npj Comput. Mater.* **5** 124
[45] Dong X *et al* 2021 3D deep learning enables accurate layer mapping of 2D materials *ACS Nano* **15** 3139–51
[46] Siao H-Y, Qi S, Ding Z, Lin C-Y, Hsieh Y-C and Chen T-M 2021 Machine learning-based automatic graphene detection with color correction for optical microscope images (arXiv:2103.13495)
[47] Pedregosa F *et al* 2011 Scikit-learn: machine learning in python *J. Mach. Learn. Res.* **12** 2825–30
[48] We provide the readers the source code in this repository github.com/gjung-group/Graphene_segmentation
[49] Cai S, Liu X, Huang J and Liu Z 2017 Feasibility of polyethylene film as both supporting material for transfer and target substrate for flexible strain sensor of CVD graphene grown on CU foil *RSC Adv.* **7** 48333–40
[50] No Y-S, Choi H K, Kim J-S, Kim H, Yu Y-J, Choi C-G and Choi J S 2018 Layer number identification of CVD-grown multilayer graphene using Si peak analysis *Sci. Rep.* **8** 1–9
[51] Kato M, Guan S and Zhao X 2021 In-situ observation of graphene using an optical microscope *Appl. Surf. Sci. Adv.* **6** 100138
[52] Li X *et al* 2009 Large-area synthesis of high-quality and uniform graphene films on copper foils *Science* **324** 1312–4
[53] Cui X *et al* 2015 Multi-terminal transport measurements of MOS_2 using a van der Waals heterostructure device platform *Nat. Nanotechnol.* **10** 534–40
[54] We made the mask using this platform (available at: www.apeer.com)
[55] Wen C-Y and Chou C-M 2004 Color image models and its applications to document examination *Forensic Sci. J.* **3** 23–32
[56] Chen W, Shi Y Q and Xuan G 2007 Identifying computer graphics using HSV color model and statistical moments of characteristic functions (IEEE) pp 1123–6
[57] Bui H M, Lech M, Cheng E, Neville K and Burnett I S 2016 Using grayscale images for object recognition with convolutional-recursive neural network (IEEE) pp 321–5
[58] Prewitt J M S 1970 Object enhancement and extraction *Picture Processing and Psychopictorics* vol 10 (New York: Academic) pp 15–19
[59] Jähne B, Haussecker H and Geissler P 1999 *Handbook of Computer Vision and Applications* vol 2 (New York: Academic)
[60] Marr D and Hildreth E 1980 Theory of edge detection *Phil. Trans. R. Soc.* B **207** 187–217
[61] Ross Quinlan J 1996 Learning decision tree classifiers *ACM Comput. Surv.* **28** 71–72
[62] Myles A J, Feudale R N, Liu Y, Woody N A and Brown S D 2004 An introduction to decision tree modeling *J. Chemom.* **18** 275–85
[63] Loh W-Y 2011 Classification and regression trees *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **1** 14–23
[64] Breiman L 2001 Random forests *Mach. Learn.* **45** 5–32
[65] Liu Y, Wang Y and Zhang J 2012 New machine learning algorithm: random forest *Int. Conf. on Information Computing and Applications* (Springer) pp 246–52
[66] Akar Ozlem and Güngör O 2012 Classification of multispectral images using random forest algorithm *J. geodesy geoinformation sci.* **1** 105–12
[67] Chen T *et al* 2015 Xgboost: extreme gradient boosting *R package version 0.4-2* **1** 1–4
[68] Chen T and Guestrin C 2016 Xgboost: a scalable tree boosting system *Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (Association for Computing Machinery) (New York, USA)* pp 785–94
[69] Hastie T, Tibshirani R, Friedman J H and Friedman J H 2009 *The Elements of Statistical Learning: Data Mining, Inference and Prediction* vol 2 (Berlin: springer)
[70] Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q and Liu T-Y 2017 Lightgbm: a highly efficient gradient boosting decision tree *Adv. Neural Inf. Process. Syst.* **30** 3146–54
[71] Zhao Q, Ye Z, Su Y and Ouyang D 2019 Predicting complexation performance between cyclodextrins and guest molecules by integrated machine learning and molecular modeling techniques *Acta Pharm. Sin.* B **9** 1241–52
[72] Kohavi R *et al* 1995 A study of cross-validation and bootstrap for accuracy estimation and model selection *Ijcai* vol 14 pp 1137–45
[73] Everingham M, Van Gool L, Williams C K I, Winn J and Zisserman A 2010 The pascal visual object classes (voc) challenge *Int. J. Comput. Vis.* **88** 303–38
[74] Li W, Hou J and Yin L 2014 A classifier fusion method based on classifier accuracy (IEEE) pp 2119–22
[75] Lorensen W E and Cline H E 1987 Marching cubes: a high resolution 3D surface construction algorithm *ACM siggraph comput. graph.* **21** 163–9
[76] Van der Walt S, Schönberger J L, Nunez-Iglesias J, Boulogne Fçois, Warner J D, Yager N, Gouillart E and Yu T 2014 scikit-image: image processing in python *PeerJ* **2** e453