




ARTICLE OPEN



Deep-learning-based image segmentation integrated with optical microscopy for automatically searching for two-dimensional materials

Satoru Masubuchi¹ , Eisuke Watanabe¹, Yuta Seo¹, Shota Okazaki², Takao Sasagawa², Kenji Watanabe³ , Takashi Taniguchi^{1,3} and Tomoki Machida¹ 

Deep-learning algorithms enable precise image recognition based on high-dimensional hierarchical image features. Here, we report the development and implementation of a deep-learning-based image segmentation algorithm in an autonomous robotic system to search for two-dimensional (2D) materials. We trained the neural network based on Mask-RCNN on annotated optical microscope images of 2D materials (graphene, hBN, MoS₂, and WTe₂). The inference algorithm is run on a 1024 × 1024 px² optical microscope images for 200 ms, enabling the real-time detection of 2D materials. The detection process is robust against changes in the microscopy conditions, such as illumination and color balance, which obviates the parameter-tuning process required for conventional rule-based detection algorithms. Integrating the algorithm with a motorized optical microscope enables the automated searching and cataloging of 2D materials. This development will allow researchers to utilize a large number of 2D materials simply by exfoliating and running the automated searching process. To facilitate research, we make the training codes, dataset, and model weights publicly available.

npj 2D Materials and Applications (2020)4:3; <https://doi.org/10.1038/s41699-020-0137-z>

INTRODUCTION

The recent advances in deep-learning technologies based on neural networks have led to the emergence of high-performance algorithms for interpreting images, such as object detection^{1–5}, semantic segmentation^{4,6–10}, instance segmentation¹¹, and image generation¹². As neural networks can learn the high-dimensional hierarchical features of objects from large sets of training data¹³, deep-learning algorithms can acquire a high generalization ability to recognize images, i.e., they can interpret images that they have not been shown before, which is one of the traits of artificial intelligence¹⁴. Soon after the success of deep-learning algorithms in general scene recognition challenges¹⁵, attempts at automation began for imaging tasks that are conducted by human experts, such as medical diagnosis¹⁶ and biological image analysis^{17,18}. However, despite significant advances in image recognition algorithms, the implementation of these tools for practical applications remains challenging¹⁸ because of the unique requirements for developing deep-learning algorithms that necessitate the joint development of hardware, datasets, and software^{18,19}.

In the field of two-dimensional (2D) materials^{20–22}, the recent advent of autonomous robotic assembly systems has enabled high-throughput searching for exfoliated 2D materials and their subsequent assembly into van der Waals heterostructures²³. These developments were bolstered by an image recognition algorithm for detecting 2D materials on SiO₂/Si substrates^{23,24}; however, current implementations have been developed on the framework of conventional rule-based image processing^{25,26}, which uses traditional handcrafted image features, such as color contrast, edges, and entropy^{23,24}. Although these algorithms are computationally inexpensive, the detection parameters need to be

adjusted by experts, with retuning required when the microscopy conditions change. To perform the parameter tuning in conventional rule-based algorithms, one has to manually find at least one sample flake on SiO₂/Si substrate, every time one exfoliates 2D flakes. Since the exfoliated flakes are sparsely distributed on SiO₂/Si substrate, e.g., 3–10 thin flakes in 1 × 1 cm² SiO₂/Si substrate for MoS₂²³, manually finding a flake and tuning parameters requires at least 30 min. The time spent for parameter-tuning process causes degradation of some two-dimensional materials, such as Bi₂Sr₂CaCu₂O_{8+δ}²⁷, even in a glovebox enclosure.

In contrast, deep-learning algorithms for detecting 2D materials are expected to be robust against changes in optical microscopy conditions, and the development of such an algorithm would provide a generalized 2D material detector that does not require fine-tuning of the parameters. In general, deep-learning algorithms for interpreting images are grouped into two categories²⁸. Fully convolutional approaches employ an encoder–decoder architecture, such as SegNet⁷, U-Net⁸, and SharpMask²⁹. In contrast, region-based approaches employ feature extraction by a stack of convolutional neural networks (CNNs), such as Mask-RCNN¹¹, PSP Net³⁰, and Deeplab¹⁰. In general, the region-based approaches outperform the fully convolutional approaches for most image segmentation tasks when the networks are trained on a sufficiently large number of annotated datasets¹¹.

In this work, we implemented and integrated deep-learning algorithms with an automated optical microscope to search for 2D materials on SiO₂/Si substrates. The neural network architecture based on Mask-RCNN enabled the detection of exfoliated 2D materials while generating a segmentation mask for each object. Transfer learning from the network trained on the Microsoft common objects in context (COCO) dataset³¹ enabled the

¹Institute of Industrial Science, University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan. ²Laboratory for Materials and Structures, Tokyo Institute of Technology, 4259 Nagatsuta, Midori-ku, Yokohama 226-8503, Japan. ³National Institute for Materials Science, 1-1 Namiki, Tsukuba, Ibaraki 305-0044, Japan. [✉]email: msatoru@iis.u-tokyo.ac.jp; tmachida@iis.u-tokyo.ac.jp

development of a neural network from a relatively small (~2000 optical microscope images) dataset of 2D materials. Owing to the generalization ability of the neural network, the detection process is robust against changes in the microscopy conditions. These properties could not be realized using conventional rule-based image recognition algorithms. To facilitate further research, we make the source codes for network training, the model weights, the training dataset, and the optical microscope drivers publicly available. Our implementation can be deployed on optical microscopes other than the instrument utilized in this study.

RESULTS

System architectures and functionalities

A schematic diagram of our deep-learning-assisted optical microscopy system is shown in Fig. 1a, with photographs shown in Fig. 1b, c. The system comprises three components: (i) an autofocus microscope with a motorized XY scanning stage (Chuo Precision); (ii) a customized software pipeline to capture the optical microscope image, run deep-learning algorithms, display results, and record the results in a database; (iii) a set of trained deep-learning algorithms for detecting 2D materials (graphene, hBN, MoS₂, and WTe₂). By combining these components, the system can automatically search for 2D materials exfoliated on SiO₂/Si substrates (Supplementary Movie 1 and 2). When 2D flakes are detected, their positions and shapes are stored in a database (sample record is presented in supplementary information), which can be browsed and utilized to assemble van der Waals heterostructures with a robotic system²³. The key component developed in this study was the set of trained deep-learning algorithms for detecting 2D materials in the optical microscope images. Algorithm development required three steps, namely, preparation of a large dataset of annotated optical microscope images, training of the deep-learning algorithm on the dataset, and deploying the algorithm to run inference on optical microscope images.

The deep-learning model we employed was Mask-RCNN¹¹ (Fig. 1a), which predicts objects, bounding boxes, and segmentation masks in images. When an image is input into the network, the deep convolutional network ResNet101³² extracts the position-aware high-dimensional features. These features are passed to the region proposal network (RPN) and the region of interest alignment network (ROI Align), which propose candidate regions where the targeted objects are located. The full connection network performs classification (Class) and regression for the bounding box (BBox) of the detected objects. Finally, the convolutional network generates segmentation masks for the objects using the output of the ROI Align layer. This model was developed on the Keras/TensorFlow framework^{33–35}.

To train the Mask-RCNN model, we prepared annotated images and trained networks as follows. In general, the performance of a deep-learning network is known to scale with the size of the dataset³⁶. To collect a large set of optical microscope images containing 2D materials, we exfoliated graphene (covalent material), MoS₂ (2D semiconductors), WTe₂, and hBN crystals onto SiO₂/Si substrates. Using the automated optical microscope, we collected ~2100 optical microscope images containing graphene, MoS₂, WTe₂, and hBN flakes. The images were annotated manually using a web-based labeling tool³⁷. The training was performed by the stochastic gradient descent method described later in this paper.

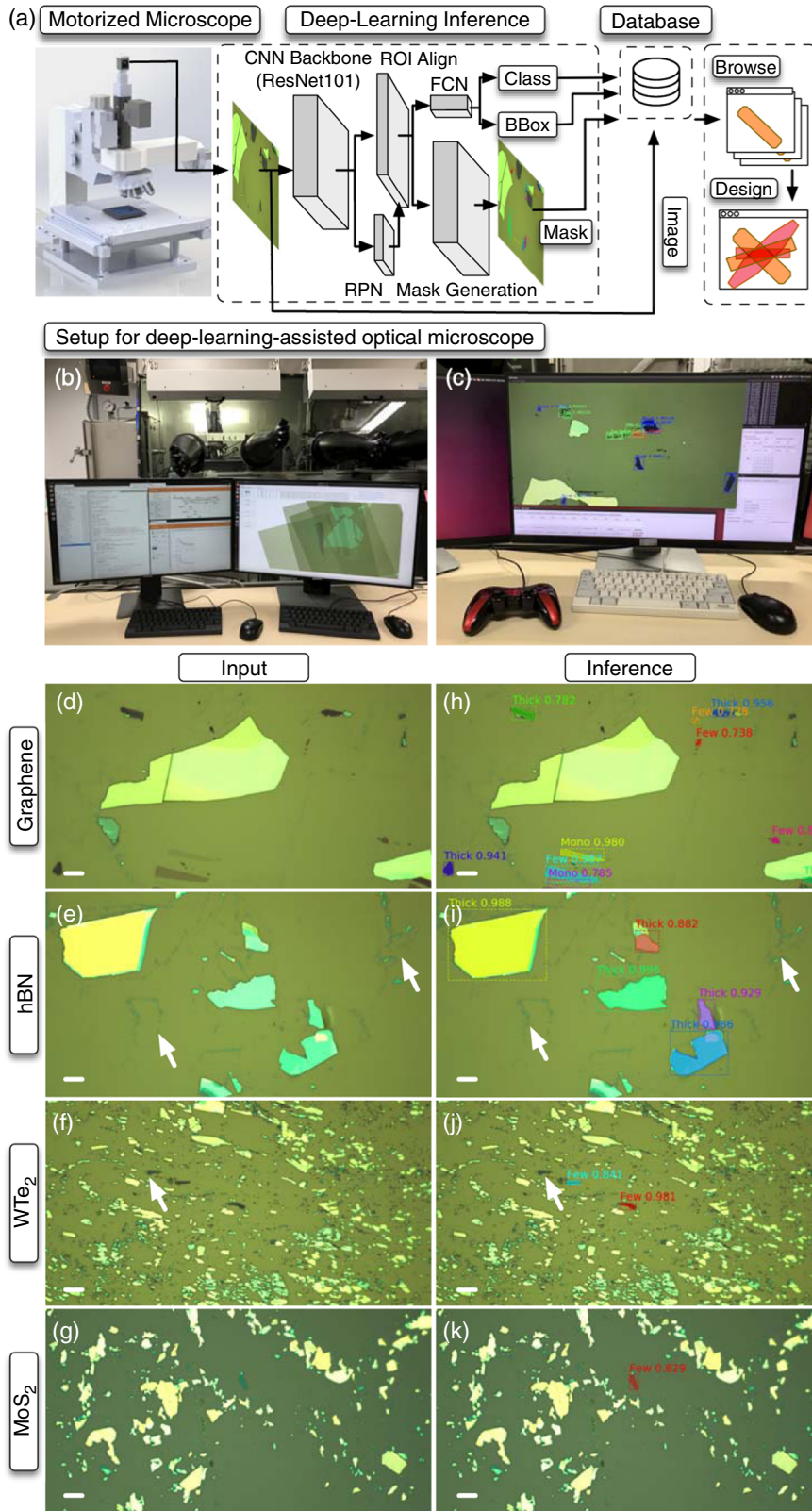
We show the inference results for optical microscope images containing 2D materials. Figure 1c–f shows optical microscope images of graphene, WTe₂, MoS₂, and hBN flakes, which were input into the neural network. The inference results shown in Fig. 1g–j consist of bounding boxes (colored squares), class labels (text), confidences (numbers), and masks (colored polygons). For

the layer thickness classification, we defined three categories: “mono” (1 layer), “few” (2–10 layers), and “thick” (10–40 layers). Note that this categorization was sufficient for practical use in the first screening process because final verification of the layer thickness can be conducted either by manual inspection or by using the computational post process, such as color contrast analysis^{24,38–42}, which would be interfaced with the deep-learning algorithms in the future works. As indicated in Fig. 1g–j, the 2D flakes are detected by the Mask-RCNN, and the segmentation mask exhibits good overlap with the 2D flakes. The layer thickness was also correctly classified, with monolayer graphene classified as “mono”. The detection process is robust against contaminating objects, such as scotch tape residue, particles, and corrugated 2D flakes (white arrows, Fig. 1e, f, i, j).

As the neural network locates 2D crystals using the high-dimensional hierarchical features of the image, the detection results were unchanged when the illumination conditions were varied (Supplementary Movie 3). Figure 2a–c shows the deep-learning detection of graphene flakes under differing illumination intensities (*I*). For comparison, the results obtained using conventional rule-based detection are presented in Fig. 2d–f. For the deep-learning case, the results were not affected by changing the illumination intensity from *I* = 220 (a) to 180 (b) or 90 (c) (red, blue, and green curves, Fig. 2). In contrast, with rule-based detection, a slight decrease in the light intensity from *I* = 220 (d) to 200 (e) affected the results significantly, and the graphene flakes became undetectable. Further decreasing the illumination intensity to *I* = 180 (f) resulted in no objects being detected. These results demonstrate the robustness of the deep-learning algorithms over conventional rule-based image processing for detecting 2D flakes.

The deep-learning model was integrated with a motorized optical microscope by developing a customized software pipeline using C++ and Python. We employed a server/client architecture to integrate the deep-learning inference algorithms with the conventional optical microscope (Supplementary Fig. 1). The image captured by the optical microscope is sent to the inference server, and the inference results are sent back to the client computer. The deep-learning model can run on a graphics-processing unit (NVIDIA Tesla V100) at 200 ms. Including the overheads for capturing images, transferring image data, and displaying inference results, frame rates of ~1 fps were achieved. To investigate the applicability of the deep-learning inference to searching for 2D crystals, we selected WTe₂ crystals as a testbed because the exfoliation yields of transition metal dichalcogenides are significantly smaller than graphene flakes. We exfoliated WTe₂ crystals onto 1 × 1 cm² SiO₂/Si substrates, and then conducted searching, which was completed in 1 h using a ×50 objective lens. Searching identified ~25 WTe₂ flakes on 1 × 1 cm² SiO₂/Si with various thicknesses (1–10 layers; Supplementary Fig. 2).

To quantify the performance of the Mask-RCNN detection process, we manually checked over 2300 optical microscope images, and the detection metrics are summarized in Supplementary Table 1. Here, we defined true- and false-positive detections (TP and FP) as whether the optical microscope image contained at least one correctly detected 2D crystal or not (examples are presented in Supplementary Figs 2–7). An image in which the 2D crystal was not correctly detected was considered a false negative (FN). Based on these definitions, the value of precision was TP/(TP + FP) ~0.53, which implies that over half of the optical microscope images with positive detection contained WTe₂ crystals. Notably, the recall (TP/(TP + FN) ~0.93) was significantly high. In addition, the examples of false-negative detection contain only small fractured WTe₂ crystals, which cannot be utilized for assembling van der Waals heterostructures. These results imply that the deep-learning-based detection process does not miss usable 2D crystals. This property is favorable for the practical application of deep-learning algorithms to searching for



2D crystals, as exfoliated 2D crystals are usually sparsely distributed over SiO₂/Si substrates. In this case, false-positive detection is less problematic than missing 2D crystals (false negative). The screening of the results can be performed by a human operator without significant intervention⁴³. In the case of

graphene (Supplementary Table 1), both the precision and recall were high (~0.95 and ~0.97, respectively), which implies excellent performance of the deep-learning algorithm for detecting 2D crystals. We speculate that there is a difference between the exfoliation yields of graphene and WTe₂ because the mean

Fig. 1 Deep-learning-assisted automated optical microscope for searching for two-dimensional (2D) crystals. **a** Schematic of the deep-learning-assisted optical microscope system. The optical microscope acquires an image of exfoliated 2D crystals on a SiO₂/Si substrate. The images are input into the deep-learning inference algorithm. The Mask-RCNN architecture generates a segmentation mask, bounding boxes, and class labels. The inference data and images are stored in a cloud database, which forms a searchable database. The customized computer-assisted-design (CAD) software enables browsing of 2D crystals, and designing of van der Waals heterostructures. **b, c** Photographs of (b) the optical microscope and (c) the computer screen for deep-learning-assisted automated searching. **d–k** Segmentation of 2D crystals. Optical microscope images of (d) graphene, (e) hBN, (f) WTe₂, and (g) MoS₂ on SiO₂ (290 nm)/Si. **h–k** Inference results for the optical microscope images in **d–g**, respectively. The segmentation masks and bounding boxes are indicated by polygons and dashed squares, respectively. In addition, the class labels and confidences are displayed. The contaminating objects, such as scotch tape residue, particles, and corrugated 2D flakes, are indicated by the white arrows in **e, f, i, and j**. The scale bars correspond to 10 μm.

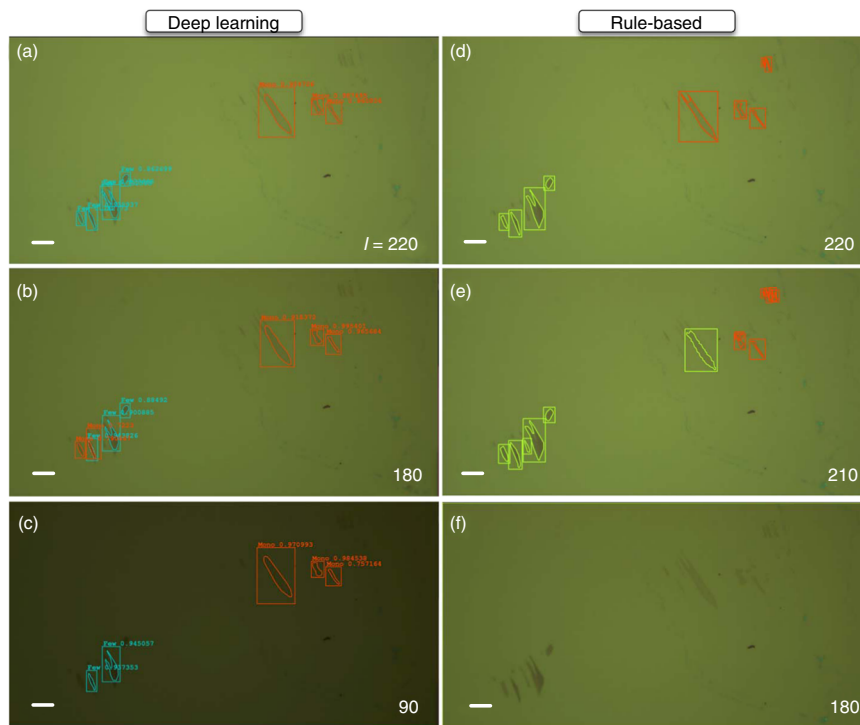


Fig. 2 Comparison between deep-learning and rule-based detection. Input image and inference results under illumination intensities of $I =$ **a** 220, **b** 180, and **c** 90 (arb. unit) for deep-learning detection, and $I =$ **d** 220, **e** 210, and **f** 180 (arb. unit) for rule-based detection. The scale bars correspond to 10 μm.

average precision (mAP) at the intersection of union (IOU) over 50% mAP@IoU_{50%} with respect to the annotated dataset (see preparation methods below) for each material does not differ significantly (0.49 for graphene and 0.52 for WTe₂). As demonstrated above, these values are sufficiently high and can be successfully applied to searches for 2D crystals. These results indicate that the deep-learning inference can be practically utilized to search for 2D crystals.

Model training

The Mask-RCNN model was trained on a dataset, where Fig. 3a shows representative annotated images, and Fig. 3b shows the annotation metrics. The dataset comprises 353 (hBN), 862 (graphene), 569 (MoS₂), and 318 (WTe₂) images. The numbers of annotated objects were 456 (hBN), 4805 (graphene), 839 (MoS₂), and 1053 (WTe₂). The annotations were converted to the JSON format compatible with the Microsoft COCO dataset using our customized scripts written in Python. Finally, the annotated dataset was randomly divided into training and test datasets in a 8:2 ratio. To train the model on the annotated dataset, we utilized the multitask loss function defined in refs^{11,33}

$$L = \alpha L_{\text{cls}} + \beta L_{\text{box}} + \gamma L_{\text{mask}} \quad (1)$$

where L_{cls} , L_{box} , and L_{mask} are the classification, localization, and segmentation mask losses, respectively; $\alpha - \gamma$ is the control parameter for tuning the balance between the loss sets as $(\alpha, \beta, \gamma) = (0.6, 1.0, 1.0)$. The class loss was

$$L_{\text{cls}} = -\log p_u \quad (2)$$

where $p = (p_0, \dots, p_k)$ is the probability distribution for each region of interest in which the result of classification is u . The bounding box loss L_{box} is defined as

$$L_{\text{box}}(t^u, v) = \sum_{i \in \{x, y, w, h\}} \text{smooth}_{L_1}(t_i^u - v_i) \quad (3)$$

where $\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2, & |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases}$ is an L_1 loss. The mask loss L_{mask} was defined as the average binary cross-entropy loss:

$$L_{\text{mask}} = -\frac{1}{m^2} \sum_{1 \leq i, j \leq m} [y_{ij} \cdot \log \hat{y}_{ij}^k + (1 - y_{ij}) \log (1 - \hat{y}_{ij}^k)] \quad (4)$$

where y_{ij} is the binary mask at (i, j) from an ROI of $(m \times m)$ size on the ground truth mask of class k , and \hat{y}_{ij}^k is the predicted class label of the same cell.

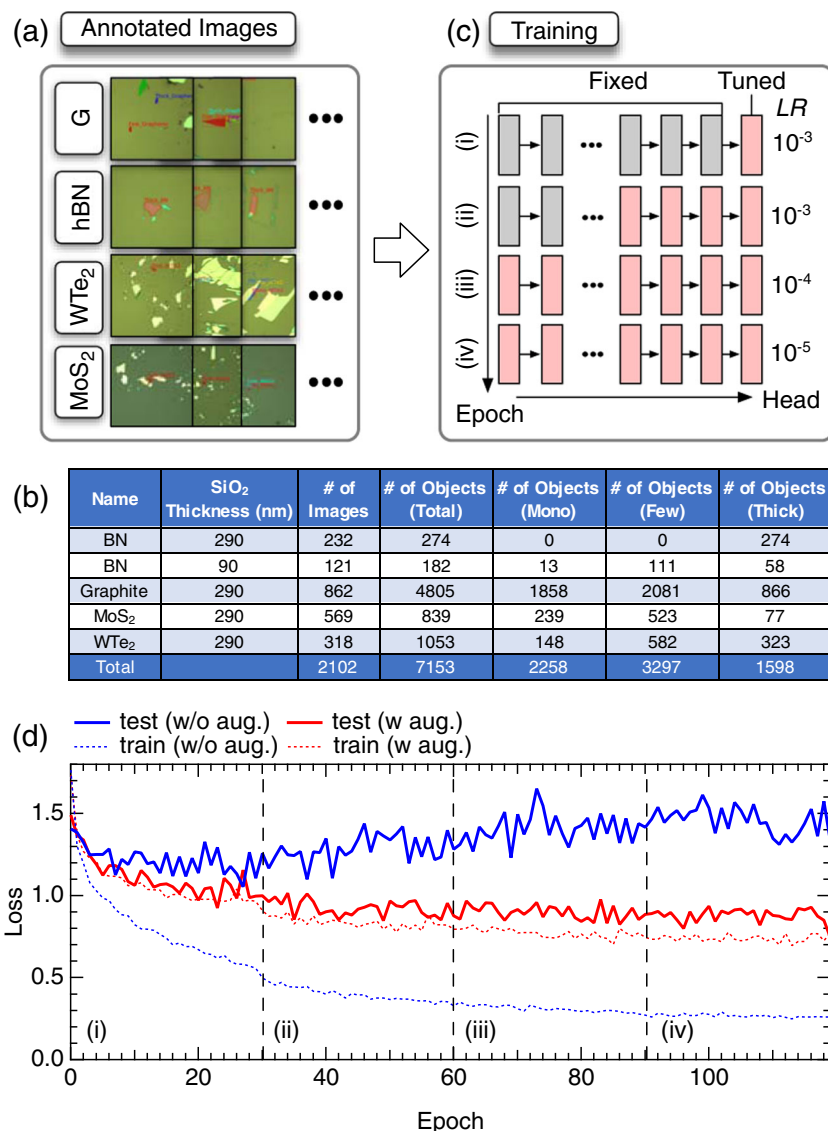


Fig. 3 Overview of training. **a** Examples of annotated datasets for graphene (G), hBN, WTe₂, and MoS₂. **b** Training data metrics. **c** Schematic of the training procedure. **d** Learning curves for training on the dataset. The network weights were initialized by the model weights pretrained on the MS-COCO dataset. Solid (dotted) curves are test (train) losses. Training was performed either with (red curve) or without (blue curve) augmentation.

Instead of training the model from scratch, the model weights, except for the network heads, were initialized using those obtained by pretraining on a large-scale object segmentation dataset in general scenes, i.e., the MS-COCO dataset³¹. The remaining parts of the network weights were initialized using random values. The optimization was conducted using a stochastic gradient descent with a momentum of 0.9 and a weight decay of 0.0001. Each training epoch consisted of 500 iterations. The training comprised four stages, each lasting for 30 epochs (Fig. 3c). For the first two training stages, the learning rate was set to 10⁻³. The learning rate was decreased to 10⁻⁴ and 10⁻⁵ for the last two stages. In the first stage, only the network heads were trained (top row, Fig. 3c). Next, the parts of the backbone starting at layer 4 were optimized (second row, Fig. 3c). In the third and fourth stages, the entire model (backbone and heads) was trained (third and fourth rows, Fig. 3c). The training took 12 h using four GPUs (NVIDIA Tesla V100 with 32-GB memory). To increase the number of training datasets, we used data augmentation techniques, including color channel multiplication, rotation, horizontal/vertical flips, and horizontal/vertical shifts. These

operations were applied to the training data with a random probability online to reduce disk usage (examples of the augmented data are presented in Supplementary Figs 8 and 9). Before being fed to the Mask-RCNN model, each image was resized to 1024 × 1024 px² while preserving the aspect ratio, with any remaining space zero padded.

To improve the generalization ability of the network, we organized the training of the Mask-RCNN model into two steps. First, the model was trained on mixed datasets consisting of multiple 2D materials (graphene, hBN, MoS₂, and WTe₂). At this stage, the model was trained to perform segmentation and classification, both on material identity and layer thickness. Then, we use the trained weights as a source, and performed transfer learning on each material subset to achieve layer thickness classification. By employing this strategy, the feature values that are common to 2D materials behind the network heads were optimized and shared between the different materials. As shown below, the sharing of the backbone network contributed to faster convergence of the network weights and a smaller test loss.

Training curve

Figure 3d shows the value of the loss function as a function of the epoch count. The solid (dotted) curves represent the test (training) loss. The training was conducted either with (red curves) or without (blue curves) data augmentation. Without augmentation, the training loss decreased to zero, while the test loss was increased. The difference between the test and training losses was significantly increased with training, which indicates that the generalization error increased, and the model overfits the training data¹³. When data augmentation was applied, both the training and validation losses decreased monotonically with training, and the difference between the training and validation losses was small. These results indicate that when 2000 optical microscope images are prepared, the Mask-RCNN model can be trained on 2D materials without overfitting.

Transfer learning

After training on multiple material categories, we applied transfer learning to the model using each sub-dataset. Figure 4a–d shows the learning curves for training the networks on the graphene, hBN, MoS₂, and WTe₂ subsets of the annotated data, respectively. The solid (dotted) curves represent the test (training) loss. The network weights were initialized using those at epoch 120 obtained by training on multiple material classes (Fig. 3d) (red curves, Fig. 4a–d). For reference, we also trained the dataset by initializing the network weights using those obtained by pretraining only on the MS-COCO dataset (blue curves, Fig. 4a–d). Notably, in all cases, the test loss decreased faster for those pretrained on the 2D crystals and MS-COCO than for those pretrained on MS-COCO only. The loss value after 30 epochs of training on 2D crystals and MS-COCO was of almost the same order as that obtained after 80 epochs of training on MS-COCO only. In addition, the minimum loss value achieved in the case of pretraining on 2D crystals and MS-COCO was smaller than that achieved with MS-COCO only. These results indicate that the

feature values that are common to 2D materials are learnt in the backbone network. In particular, the trained backbone network weights contribute to improving the model performance on each material.

To investigate the improvement of the model accuracy, we compared the inference results for the optical microscope images using the network weights from each training set. Figure 4e–h shows the optical microscope images of graphene and WTe₂, respectively, input into the network. We employed the model weights where the loss value was minimum (indicated by the red/blue arrows). The inference results in the cases of transferring only from MS-COCO, and from both MS-COCO and 2D materials, are shown in Fig. 4f, g for graphene, and Fig. 4i, j for WTe₂. For graphene, the model transferred from MS-COCO only failed in detecting some thick graphite flakes, as indicated by the white arrows in Fig. 4f, whereas the model trained on MS-COCO and 2D crystals detected the graphene flakes, as indicated by the white arrows in Fig. 4g. Similarly, for WTe₂, when the inference process was performed using the model transferred from MS-COCO only, the surface of the SiO₂/Si substrate surrounded by thick WTe₂ crystals was misclassified as WTe₂, as indicated by the white arrow in Fig. 4d. In contrast, when learning was transferred from the model pretrained on MS-COCO and 2D materials (red arrow, Fig. 4b), this region was not recognized as WTe₂. These results indicate that pretraining on multiple material classes contributes to improving model accuracy because the common properties of 2D crystals are learnt in the backbone network. The inference results presented in Fig. 1 were obtained by utilizing the model weights at epoch 120 for each material.

Generalization ability

Finally, we investigated the generalization ability of the neural network for detecting graphene flakes in images obtained using different optical microscope setups (Asahikogaku AZ10-T/E, Keyence VHX-900, and Keyence VHX-5000 as shown in Fig. 5a–c,

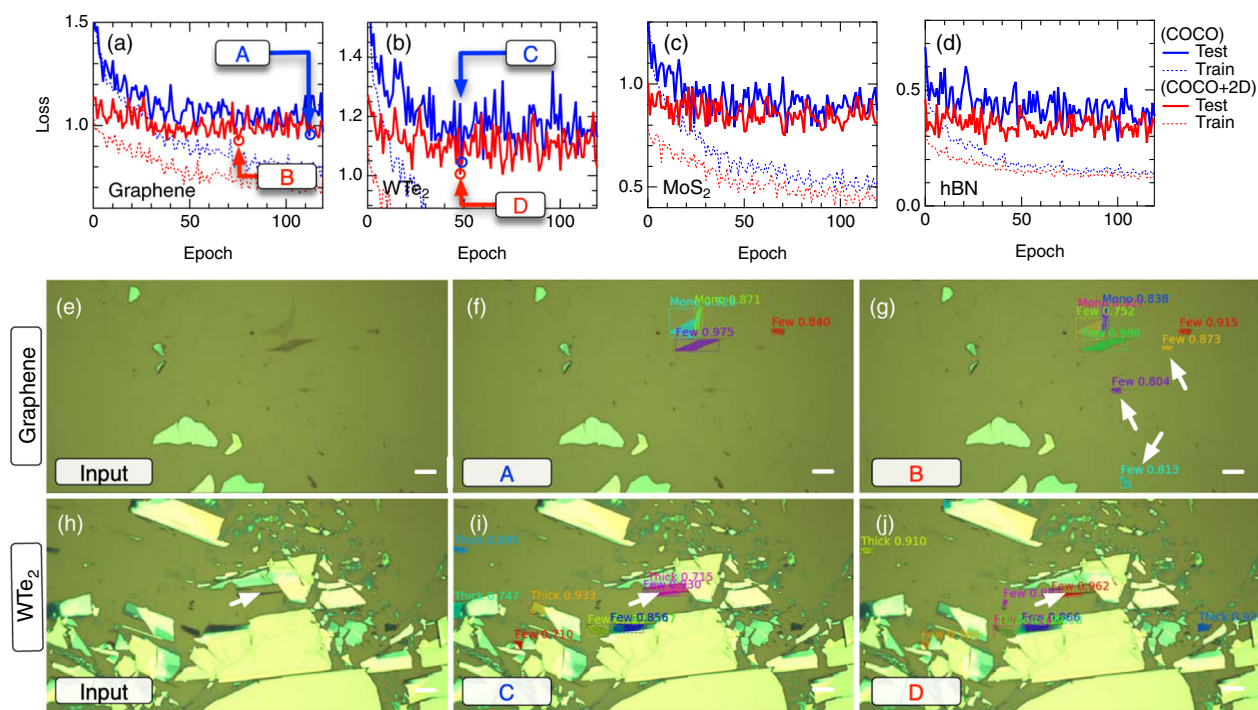


Fig. 4 Transfer learning from model weights pretrained on MS-COCO and MS-COCO+2D datasets. **a–d** Test (solid curves) and training (dotted curves) losses as a function of epoch count for training on **a** graphene, **b** WTe₂, **c** MoS₂, and **d** hBN. Each epoch consists of 500 training steps. The model weights were initialized using those pretrained on (blue) MS-COCO and (red) MS-COCO and 2D material datasets. The optical microscope image of graphene (WTe₂) and the inference results for these images are shown in **e–g** (**h–j**). The scale bars correspond to 10 μm.

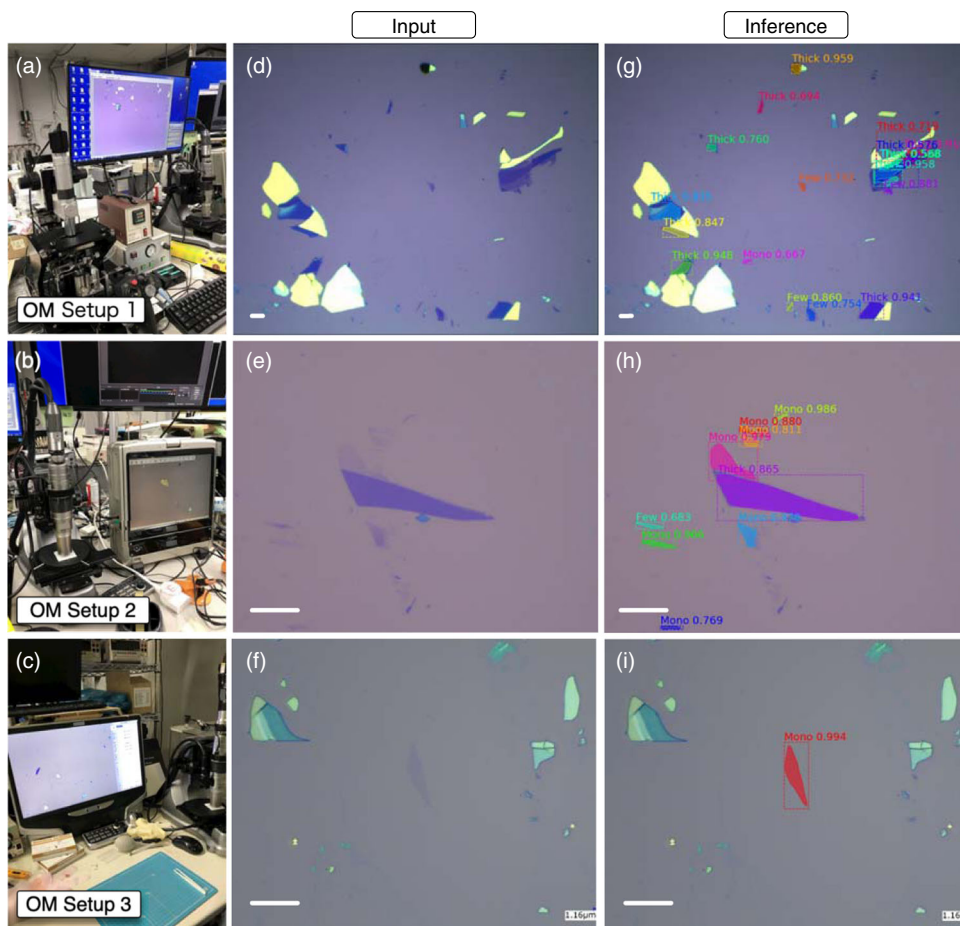


Fig. 5 Generalization ability of the neural network. **a–c** Optical microscope setups used for capturing images of exfoliated graphene (Asahikogaku AZ10-T/E, Keyence VHX-900, and Keyence VHX-5000, respectively). **d–f** Optical microscope images recorded using instruments (**a–c**), respectively. **g–i** Inference results for the optical microscope images in **d–f**, respectively. The segmentation masks are shown in color, and the category and confidences are also indicated. The scale bars correspond to 10 μm .

respectively). Figure 5d–f shows the optical microscope images of exfoliated graphene captured by each instrument. Across these instruments, there are significant variations in the white balance, magnification, resolution, illumination intensity, and illumination inhomogeneity (Fig. 5d–f). The model weights from training epoch 120 on the graphene dataset were employed (red arrow, Fig. 4d). Even though no optical microscope images recorded by these instruments were utilized for training, as shown by the inference results in Fig. 5g–i, the deep-learning model successfully detected the regions of exfoliated graphene. These results indicate that our trained neural network captured the latent general features of graphene flakes, and thus constitutes a general-purpose graphene detector that works irrespective of the optical microscope setup. These properties cannot be realized by utilizing the conventional rule-based detection algorithms for 2D crystals, where the detection parameters must be retuned when the optical conditions were altered.

DISCUSSION

In order to train the neural network for the 2D crystals that have different appearance, such as ZrSe_3 , the model weights trained on both MS-COCO and 2D crystals obtained in this study can be used as source weights to start training. In our experience, the MaskRCNN trained on a small dataset of ~ 80 images from the MS-COCO pretrained model can produce rough segmentation masks on graphene. Therefore, providing < 80 annotated images would be

sufficient for developing a classification algorithm that works for detecting other 2D materials when we use our trained weights as a source. Our work can be utilized as a starting point for developing neural network models that work for various 2D materials.

Moreover, the trained neural networks can be utilized for searching the materials other than those used for training. For demonstration, we exfoliated WSe_2 and MoSe_2 flakes on SiO_2/Si substrate, and conducted searching with the model trained on WTe_2 . As shown in Supplementary Figs 10 and 11 in supplementary information, thin WSe_2 and MoSe_2 flakes are correctly detected even without training on these materials. This result indicates that the difference of the appearances of WSe_2 and MoSe_2 from WTe_2 are covered by the generalization ability of neural networks.

Finally, our deep-learning inference process can run on the remote server/client architecture. This architecture is suitable for researchers with an occasional need for deep learning, as it provides a cloud-based setup that does not require a local GPU. The conventional optical microscope instruments that were not covered in this study can also be modified to support deep-learning inference by implementing the client software to capture an image, send an image to the server, receive, and display inference results. The distribution of the deep-learning inference system will benefit the research community by saving the time needed for optical microscopy-based searching of 2D materials.

In this work, we developed a deep-learning-assisted automated optical microscope to search for 2D crystals on SiO₂/Si substrates. A neural network with Mask-RCNN architecture trained on 2D materials enabled the efficient detection of various exfoliated 2D crystals, including graphene, hBN, and transition metal dichalcogenides (WTe₂ and MoS₂), while simultaneously generating a segmentation mask for each object. This work, along with the recent other attempts for utilizing the deep-learning algorithms^{44–46}, should free researchers from the repetitive tasks of optical microscopy, and comprises a fundamental step toward realizing fully automated fabrication systems for van der Waals heterostructures. To facilitate such research, we make the source codes for training, the model weights, the training dataset, and the optical microscope drivers publicly available.

METHODS

Optical microscope drivers

The automated optical microscope drivers were written in C++ and Python. The software stack was developed on the stacks of a robotic operating system⁴⁷ and the HALCON image-processing library (MVTec Software GmbH).

Preparation of the training dataset

To obtain the Mask-RCNN model to segment 2D crystals, we employed a semiautomatic annotation workflow. First, we trained the Mask-RCNN with a small dataset consisting of ~80 images of graphene. Then, we conducted predictions on optical microscope images of graphene. The prediction labels generated using the Mask-RCNN were stored in LabelBox using API. These labels were manually corrected by a human annotator. This procedure greatly enhanced the annotation efficiency, allowing each image to be labeled in 20–30 s.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

CODE AVAILABILITY

The source code, the trained network weights, and the training data are available at <https://github.com/tdmms/>.

Received: 20 October 2019; Accepted: 24 February 2020;

Published online: 23 March 2020

REFERENCES

- Zhao, Z.-Q., Zheng, P., Xu, S.-t. & Wu, X. Object detection with deep learning: a review. *IEEE Transactions on Neural Networks and Learning Systems* **30**, 3212–3232 (2019).
- Ren, S., He, K., Girshick, R. & Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 91–99 (Neural Information Processing Systems Foundation, 2015).
- Girshick, R. Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, 1440–1448 (IEEE, 2015).
- Girshick, R., Donahue, J., Darrell, T. & Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 580–587 (IEEE, 2014).
- Liu, W. et al. SSD: Single shot multibox detector. *European Conference on Computer Vision*, 21–37 (Springer, 2016).
- García-García, A., Orts-Escobedo, S., Oprea, S. O., Villena-Martínez, V. & García-Rodríguez, J. A review on deep learning techniques applied to semantic segmentation. Preprint at <https://arxiv.org/abs/1704.06857> (2017).
- Badrinarayanan, V., Kendall, A. & Cipolla, R. SegNet: a deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**, 2481–2495 (2017).
- Ronneberger, O., Fischer, P. & Brox, T. U-Net: convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-assisted Intervention*, 234–241 (Springer, 2015).
- Long, J., Shelhamer, E. & Darrell, T. Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3431–3440 (IEEE, 2015).
- Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. & Yuille, A. L. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**, 834–848 (2017).
- He, K., Gkioxari, G., Dollár, P. & Girshick, R. Mask R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, 2961–2969 (IEEE, 2017).
- Goodfellow, I. et al. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 2672–2680 (Neural Information Processing Systems Foundation, 2014).
- Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning*. (MIT Press, 2016).
- LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
- Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 1097–1105 (Neural Information Processing Systems Foundation, 2012).
- Litjens, G. et al. A survey on deep learning in medical image analysis. *Med. Image Anal.* **42**, 60–88 (2017).
- Falk, T. et al. U-Net: deep learning for cell counting, detection, and morphometry. *Nat. Methods* **16**, 67–70 (2019).
- Moen, E. et al. Deep learning for cellular image analysis. *Nat. Methods* <https://doi.org/10.1038/s41592-019-0403-1> (2019).
- Karpathy, A. *Software 2.0*. <https://medium.com/@karpathy/software-2-0-a64152b37c35> (2017).
- Novoselov, K. S., Mishchenko, A., Carvalho, A. & Castro Neto, A. H. 2D materials and van der Waals heterostructures. *Science* **353**, aac9439 (2016).
- Novoselov, K. S. et al. Two-dimensional atomic crystals. *Proc. Natl Acad. Sci. USA* **102**, 10451–10453 (2005).
- Novoselov, K. S. et al. Electric field effect in atomically thin carbon films. *Science* **306**, 666–669 (2004).
- Masubuchi, S. et al. Autonomous robotic searching and assembly of two-dimensional crystals to build van der Waals superlattices. *Nat. Commun.* **9**, 1413 (2018).
- Masubuchi, S. & Machida, T. Classifying optical microscope images of exfoliated graphene flakes by data-driven machine learning. *npj 2D Mater. Appl.* **3**, 4 (2019).
- Nixon, M. S. & Aguado, A. S. *Feature Extraction & Image Processing for Computer Vision* (Academic Press, 2012).
- Szeliski, R. *Computer Vision: Algorithms and Applications*. (Springer Science & Business Media, 2010).
- Yu, Y. et al. High-temperature superconductivity in monolayer Bi₂Sr₂CaCu₂O_{8-x}. *Nature* **575**, 156–163 (2019).
- Ghosh, S., Das, N., Das, I. & Maulik, U. Understanding deep learning techniques for image segmentation. Preprint at <https://arxiv.org/abs/1907.06119> (2019).
- Pinheiro, P. O., Lin, T.-Y., Collobert, R. & Dollár, P. Learning to refine object segments. *European Conference on Computer Vision*, 75–91 (Springer, 2016).
- Zhao, H., Shi, J., Qi, X., Wang, X. & Jia, J. Pyramid scene parsing network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2881–2890 (IEEE, 2017).
- Lin, T.-Y. et al. Microsoft COCO: common objects in context. *European Conference on Computer Vision*, 740–755 (Springer, 2014).
- 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*, 770–778 (IEEE, 2016).
- Abdulla, W. *Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow* https://github.com/matterport/Mask_RCNN (2017).
- Chollet, F. Keras: Deep learning for humans <https://github.com/keras-team/keras> (2015).
- Abadi, M. et al. Tensorflow: a system for large-scale machine learning. *12th USENIX Symposium on Operating Systems Design and Implementation*, 265–283 (USENIX Association, 2016).
- Hestness, J. et al. Deep learning scaling is predictable, empirically. Preprint at <https://arxiv.org/abs/1712.00409> (2017).
- Labelbox, “Labelbox,” Online, [Online]. <https://labelbox.com> (2019).
- Lin, X. et al. Intelligent identification of two-dimensional nanostructures by machine-learning optical microscopy. *Nano Res.* **11**, 6316–6324 (2018).
- Li, H. et al. Rapid and reliable thickness identification of two-dimensional nanosheets using optical microscopy. *ACS Nano* **7**, 10344–10353 (2013).
- Ni, Z. H. et al. Graphene thickness determination using reflection and contrast spectroscopy. *Nano Lett.* **7**, 2758–2763 (2007).
- Nolen, C. M., Denina, G., Teweldebrhan, D., Bhanu, B. & Balandin, A. A. High-throughput large-area automated identification and quality control of graphene and few-layer graphene films. *ACS Nano* **5**, 914–922 (2011).
- Taghavi, N. S. et al. Thickness determination of MoS₂, MoSe₂, WS₂ and WSe₂ on transparent stamps used for deterministic transfer of 2D materials. *Nano Res.* **12**, 1691–1695 (2019).

43. Zhang, P., Zhong, Y., Deng, Y., Tang, X. & Li, X. A survey on deep learning of small sample in biomedical image analysis. Preprint at <https://arxiv.org/abs/1908.00473> (2019).
44. Saito, Y. et al. Deep-learning-based quality filtering of mechanically exfoliated 2D crystals. *npj Computational Materials* **5**, 1–6 (2019).
45. Han, B. et al. Deep learning enabled fast optical characterization of two-dimensional materials. Preprint at <https://arxiv.org/abs/1906.11220> (2019).
46. Greplova, E. et al. Fully automated identification of 2D material samples. Preprint at <https://arxiv.org/abs/1911.00066> (2019).
47. Quigley, M. et al. ROS: an open-source Robot Operating System. *ICRA Workshop on Open Source Software* (Open Robotics, 2009).

ACKNOWLEDGEMENTS

This work was supported by CREST, Japan Science and Technology Agency Grant Numbers JPMJCR15F3 and JPMJCR16F2, and by JSPS KAKENHI under Grant No. JP19H01820.

AUTHOR CONTRIBUTIONS

S.M. conceived the scheme, implemented the software, trained the neural network, and wrote the paper. E.W. and Y.S. exfoliated the 2D materials and tested the system. S.O. and T.S. synthesized the WTe₂ and WSe₂ crystals. K.W. and T.T. synthesized the hBN crystals. T.M. supervised the research program.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information is available for this paper at <https://doi.org/10.1038/s41699-020-0137-z>.

Correspondence and requests for materials should be addressed to S.M. or T.M.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2020