scientific reports

Check for updates

OPEN

Universal image segmentation for optical identification of 2D materials

Randy M. Sterbentz, Kristine L. Haley & Joshua O. Island✉

Machine learning methods are changing the way data is analyzed. One of the most powerful and widespread applications of these techniques is in image segmentation wherein disparate objects of a digital image are partitioned and classified. Here we present an image segmentation program incorporating a series of unsupervised clustering algorithms for the automatic thickness identification of two-dimensional materials from digital optical microscopy images. The program identifies mono- and few-layer flakes of a variety of materials on both opaque and transparent substrates with a pixel accuracy of roughly 95%. Contrasting with previous attempts, application generality is achieved through preservation and analysis of all three digital color channels and Gaussian mixture model fits to arbitrarily shaped data clusters. Our results provide a facile implementation of data clustering for the universal, automatic identification of two-dimensional materials exfoliated onto any substrate.

Image segmentation has become an essential technique in fields from medical imaging[1–3] and autonomous driving[4] to robotic perception[5] and image compression[6–8]. Through unsupervised segmentation of large data sets, trained algorithms can recognize and predict elements of new images. An appealing application of image segmentation is in the thickness identification of two-dimensional (2D) materials from their digital optical microscopy images. Current flake detection methods rely heavily on identification by trained researchers, a human-learning process, in which flakes are identified by their contrast difference on a substrate after significant trial and error. Automatic thickness identification would relieve this tedious, time-consuming screening process and possibly improve identification accuracy.

The easiest implementation of image segmentation for 2D materials is by thresholding. This is performed by analyzing image contrast, from reflectance or transmittance for example, and partitioning regions of an image based on contrast level difference. This technique has been widely and successfully employed in the identification and characterization of exfoliated 2D materials[9–19]. Thresholding techniques, while easily implemented, suffer from inaccuracy when contrast differences become relatively small, for example a single layer of graphene on a silicon/silicon dioxide (Si/SiO_2) substrate, and can be highly dependent on precise experimental conditions hindering universal application.

Recently, a variety of more advanced machine learning techniques have emerged to automate and improve the process of identifying exfoliated 2D materials[20–30]. These include techniques based on neural networks and data clustering but have been primarily applied to opaque substrates and almost entirely to standard Si/SiO_2. Transparent substrates are commonly used for exfoliation or experiments on 2D materials[11,31–35]. A method that can be applied to identify the thickness of any material on any substrate is highly desirable.

Here we present an open source program[36] written in Python 3 to automatically identify the thickness of exfoliated 2D flakes which can be universally applied to different materials and substrates. We combine three well-established clustering techniques to form a training script to segment layers of a flake, manually label the layers, and then use that training to test thicknesses of other flakes. The program presents roughly a 95% pixel accuracy for graphene and transition metal dichalcogenides on silicon/silicon dioxide and polydimethylsiloxane (PDMS) substrates. Importantly, no change to the program's adjustable parameters are needed to identify different materials on different substrates, allowing simple and universal application to any material/substrate combination.

Overview of the program

An overview of the program applied to graphene on Si/SiO_2 is shown in Fig. 1. The training stage begins with a set of optical microscopy images cropped to few-layer flakes whose thicknesses are determined using optical contrast methods[9]. Figure 1a shows a cropped optical image of a few-layer graphene flake on a Si/SiO_2 (300 nm

Department of Physics and Astronomy, University of Nevada Las Vegas, Las Vegas, NV 89154, USA. ✉email: jisland@physics.unlv.edu
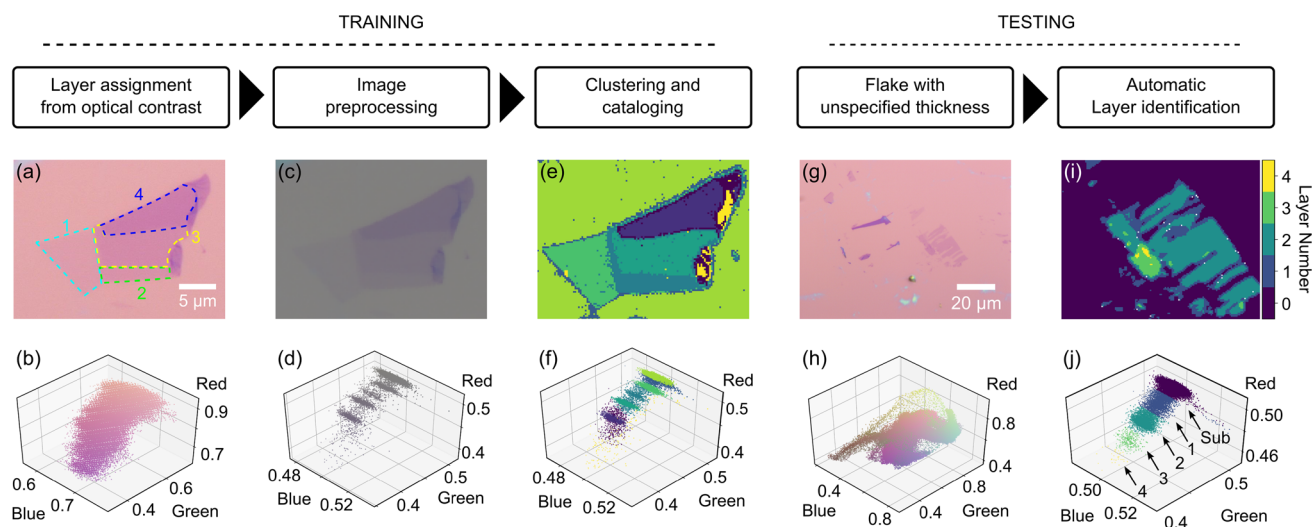
TRAINING

TESTING

| Layer assignment from optical contrast | Image preprocessing | Clustering and cataloging | Flake with unspecified thickness | Automatic Layer identification |

Figure 1. Overview of the program composed of training and testing stages. (**a,b**) Raw optical image of a few-layer graphene flake on 300 nm of SiO$_2$ (**a**) and its corresponding scatter plot of each pixel in RGB color space (**b**). The scatter plot data points are colored to match their RGB value. (**c,d**) The result after preprocessing the image in (**a**). The preprocessing reveals well-defined clusters in the associated scatter plot (**d**). (**e,f**) A color plot of pixel-cluster association (**e**) and corresponding scatter plot (**f**). Each pixel is colored based on its most probable data cluster identity. (**g,h**) Raw optical image of few-layer graphene flake with unknown thickness (**g**) used for testing and its corresponding RGB scatter plot (**h**). (**i,j**) Crop of (**g**) around the flake of interest (**i**) and corresponding scatter plot after the testing stage (**j**).

SiO$_2$) substrate with each layer thickness labeled (1–4 layers). A scatter plot of the red, green, and blue (RGB) channel values (normalized to a range 0–1) for each pixel in the image is shown in Fig. 1b. The data points have been colored according to their RGB values. At this stage, the scatter plot shows only broad features that can be generally associated with the substrate (pink) and few-layer graphene (purple) but no clear correspondence to individual thicknesses can be made. The raw image is preprocessed using a bilateral filter to reduce noise and a background normalization using a planar fit. The result, after compression to roughly 10,000 total pixels, is shown in Fig. 1c. Preprocessing reveals the individual clusters of data in the scatter plot (Fig. 1d) associated with the substrate and flake layers.

The location and distribution of these clusters in RGB space are found using a series of unsupervised clustering techniques summarized here and detailed further below. The centers are first located using mean shift and density-based spatial clustering. Once the centers are identified, fit characteristics such as the weight, mean position, and distribution of each cluster are found using a Gaussian mixture model. An image of the result of the fitting algorithm is shown in Fig. 1e. The pixels in the color plot have been colored according to the fit results and the scatter plot (Fig. 1f) shows these fit assignments in more detail. Once the cluster characteristics are extrapolated for several training images, a master catalogue is created that ties the fit clusters to the predetermined flake thicknesses in our training images.

To determined accuracy of the training, we test the master catalogue on a set of images with identified thicknesses. An example of the testing results for graphene on Si/SiO$_2$ is shown in Fig. 1g–j. In the testing stage, untrained optical images (Fig. 1g,h) are first preprocessed using the same procedure as in training but without cropping. The preprocessed images are then checked against the master catalogue for flake thickness assignment given the pixel location in RGB space. Figure 1i shows the result (cropped to show detail of flake of interest) with layer thicknesses identified by the color bar. The associated scatter plot in Fig. 1j shows the corresponding clusters and layer assignments. In the following section we detail the implementation of the clustering algorithms before presenting results of our program performed on other materials and substrates.

Unsupervised clustering algorithms

Our training script incorporates three clustering algorithms to identify the center of the data clusters and fit their distributions. Without explicitly knowing the number of clusters (layers) in the image, the script begins with an unsupervised method of determining the seed number. We use mean shift[37–39] and density-based[40,41] algorithms to first find these cluster centers which are then fed to a Gaussian mixture model for fitting arbitrary ellipsoidal distributions.

Mean shift is an unsupervised machine learning algorithm that locates centers of high data density. The algorithm begins by populating color space with an array of points, referred to as "mean points" ($\vec{\rho}_k$). Figure 2a shows the same data as in Fig. 1d but with red closed circles indicating the initial positions of the equally spaced mean points (an 8 × 8 × 8 array). The next step groups all data points within a defined radius (ϵ) of each mean point together. We define ϵ to be just large enough to overlap with its nearest neighbors. The average location of the data pixels within ϵ of a given mean point becomes the new position of that mean point ($\vec{\rho}_k'$) after one iteration of the algorithm. This is calculated by:
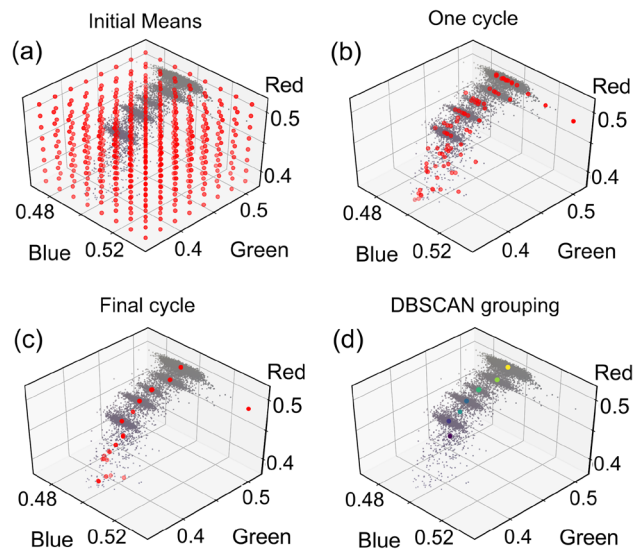
Figure 2. Mean shift and DBSCAN clustering for identification of cluster centers. (**a**) The same data as in Fig. 1d with red closed circles showing the initial positions of the mean points in the mean shift algorithm. (**b**) Scatter plot after one cycle of mean shift; outlier mean points have been deleted and others have moved towards their local density maxima. (**c**) The final state of the mean points after they have converged to their maxima. (**d**) The RGB pixels plotted with the identified cluster centers (colored closed circles) after the DBSCAN algorithm.

$$\vec{\rho}_k' = \frac{1}{M} \sum_i \vec{x}_i \text{ for } |\vec{x}_i - \vec{\rho}_k| < \epsilon,\tag{1}$$

where $\vec{x}_i$ is the position of each data point and M is the total number of data points within ϵ of $\vec{\rho}_k$. In this way, each mean point gradually shifts towards higher densities of data. Figure 2b shows one iteration of the algorithm. Several points have moved to their new mean positions according to the data within ϵ of each $\vec{\rho}_k$.

Mean shift is computationally slow, having to calculate the distance between every data point ($\approx$ 10,000 pixels) and every mean point (initially 512). To increase efficiency, mean points that have no data within ϵ after the first cycle, and thus make no contribution towards a data cluster, are deleted. Additionally, mean points may approach their local maximum at different rates. Per mean point, as soon as the number of data points within their radius starts to decrease, they are turned off and no longer involved in future calculations. Figure 2c shows the final state of the algorithm where all mean points have converged to their local density maxima. After this, outliers are removed before moving to the next algorithm.

Once mean shift is complete, several mean points will themselves be clustered in color space and some mean points will have converged to outliers. Due to the ellipsoidal shape of the clusters in RGB space after preprocessing, the mean points will tend to lie along lines. An efficient algorithm for grouping these lines is Density-Based Spatial Clustering of Applications with Noise (DBSCAN)[40,41]. DBSCAN groups data together by following the trajectory of nearby points. The algorithm starts by "visiting" a random mean point. A radius (we find $\epsilon/2$ works well) around it is checked for other mean points. If none are found, the starting point is labeled an outlier. If there are neighbors, they are grouped together. One of the other points in this group is visited next, checking the same radius around itself to find new points to add to the group. This repeats until no new points are added to the group and every point within the group has been visited. Once the group is finished, a new group starts at a randomly chosen mean point and the process repeats. The centers of each group are found by averaging their respective mean points. Figure 2d shows the result after running the DBSCAN algorithm on the mean points in Fig. 2c.

The combination of mean shift and DBSCAN presents an unsupervised method of determining how many clusters are in a given image and their centers. Following this step, this information can be used to seed a more powerful clustering technique for data with ellipsoidal distributions. The popular K-means clustering technique[42–44], for example, is undesirable here as it assumes spherical clusters. Instead, we use a multivariate Gaussian Mixture Model (GMM)[2,45–49] that allows fitting of data with arbitrary normal distributions. This expands application of the program by automatically handling new materials and substrate combinations that may have different cluster distributions in RGB space.

In the GMM, each fitting ellipsoid has three characteristics developing throughout the process: the weight (ϕ_k, defining the number of data points near ellipsoid k), the centroid ($\vec{\mu}_k$, defining the mean of the data points belonging to ellipsoid k), and the covariance matrix (Σ_k, defining the shape and orientation of ellipsoid k in RGB space). These characteristics are used to calculate the probability γ_{ik} of a data point $\vec{x}_i$ belonging to ellipsoid k. This probability is given by:
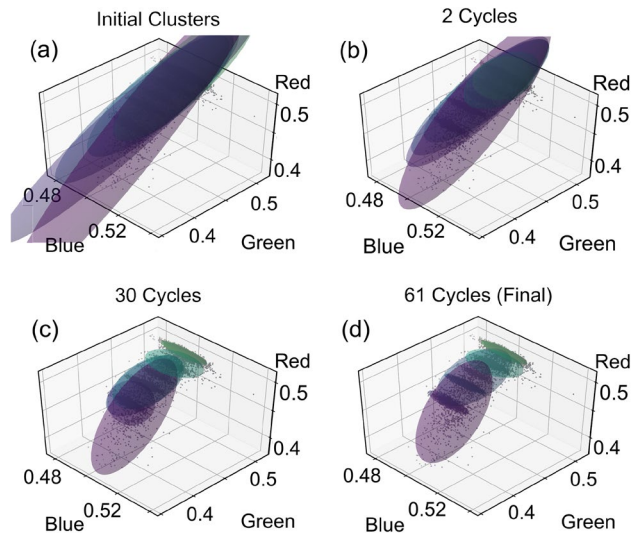
Figure 3. Cluster fitting with a Gaussian mixture model (GMM). The scatter plot from Fig. 1d is superimposed with 95% confidence ellipsoids based on the fit characteristics of the GMM-EM algorithm. (**a**) The initialized ellipsoids show little correspondence to the underlying data. (**b**) After two cycles of expectation-maximization, the ellipsoids better resemble the data clusters. (**c**) After 30 cycles, some ellipsoids have nearly converged on their data clusters. (**d**) The convergence condition is reached after 61 cycles.

$$\gamma_{ik} = \frac{\phi_k \mathcal{N}(\vec{x}_i, \vec{\mu}_k, \Sigma_k)}{\sum_{j=1}^{K} \phi_j \mathcal{N}(\vec{x}_i, \vec{\mu}_j, \Sigma_j)}, \tag{2}$$

where K is the total number of clusters and $\mathcal{N}$ is the three-variable (for 3-dimensional RGB space) Gaussian distribution given by:

$$\mathcal{N}(\vec{x}_i, \vec{\mu}_k, \Sigma_k) = \left[(2\pi)^3 |\Sigma_k| e^{(\vec{x}_i - \vec{\mu}_k)^T \Sigma_k^{-1}(\vec{x}_i - \vec{\mu}_k)} \right]^{-\frac{1}{2}}. \tag{3}$$

The weights, means, and covariance matrices used in these relations are calculated through:

$$\phi_k = \frac{1}{N} \sum_{i=1}^{N} \gamma_{ik}, \tag{4}$$

$$\vec{\mu}_k = \frac{\sum_{i=1}^{N} \gamma_{ik} \vec{x}_i}{\sum_{i=1}^{N} \gamma_{ik}}, \tag{5}$$

$$\Sigma_k = \frac{\sum_{i=1}^{N} \gamma_{ik} |\vec{x}_i - \vec{\mu}_k|^2}{\sum_{i=1}^{N} \gamma_{ik}}. \tag{6}$$

First we initialize each of the fitting ellipsoids by setting all initial weights to $1/K$. The centroids are taken directly from the results of DBSCAN $\vec{\mu} = \vec{\rho}$. The covariance matrices are initialized from the centroids using Equation 6 with $\gamma_{ik} = 1$. Figure 3a shows the initialization of the fitting ellipsoids for our example few-layer graphene data set from Figs. 1d and 2. The ellipsoids have been scaled to a 95% confidence level.

An unsupervised machine learning algorithm, referred to as expectation-maximization (EM), is used to further optimize the ellipsoid parameters and fit the data. The expectation step determines γ_{ik} based on the initialized weights, centroids, and covariance matrices calculated above. The maximization step uses these probabilities to re-calculate each cluster's weight, centroid, and covariance matrix. These two steps iterate and gradually the ellipsoid parameters converge. Figure 3b shows the algorithm results after 2 cycles and Fig. 3c shows the results after 30 cycles. After 30 cycles, the ellipsoids resemble the distributions of the data with several small tight ellipsoids corresponding to the substrate and 1–4 layers of graphene, and two larger ellipsoids (purple and blue) accounting for noise. The max change of all cluster's weights between maximization steps ($\Delta\phi_k < 0.0001$) is used to define convergence and end the algorithm. Figure 3(d) shows the results of the algorithm after convergence (total 61 cycles) for this data set. Note that the large purple and blue ellipsoids are a product of over-fitting the data (fitting 7 ellipsoids to 5 data clusters). These ellipsoids do not contribute to the master catalogue but are important for fitting data points associated with thicker layers (> 4) and outliers. Additionally, the over fitting allows the primary ellipsoids to confine themselves to the core of their data clusters. Once convergence has been reached, only ellipsoids that fit well to known layer thicknesses are added to a catalogue.
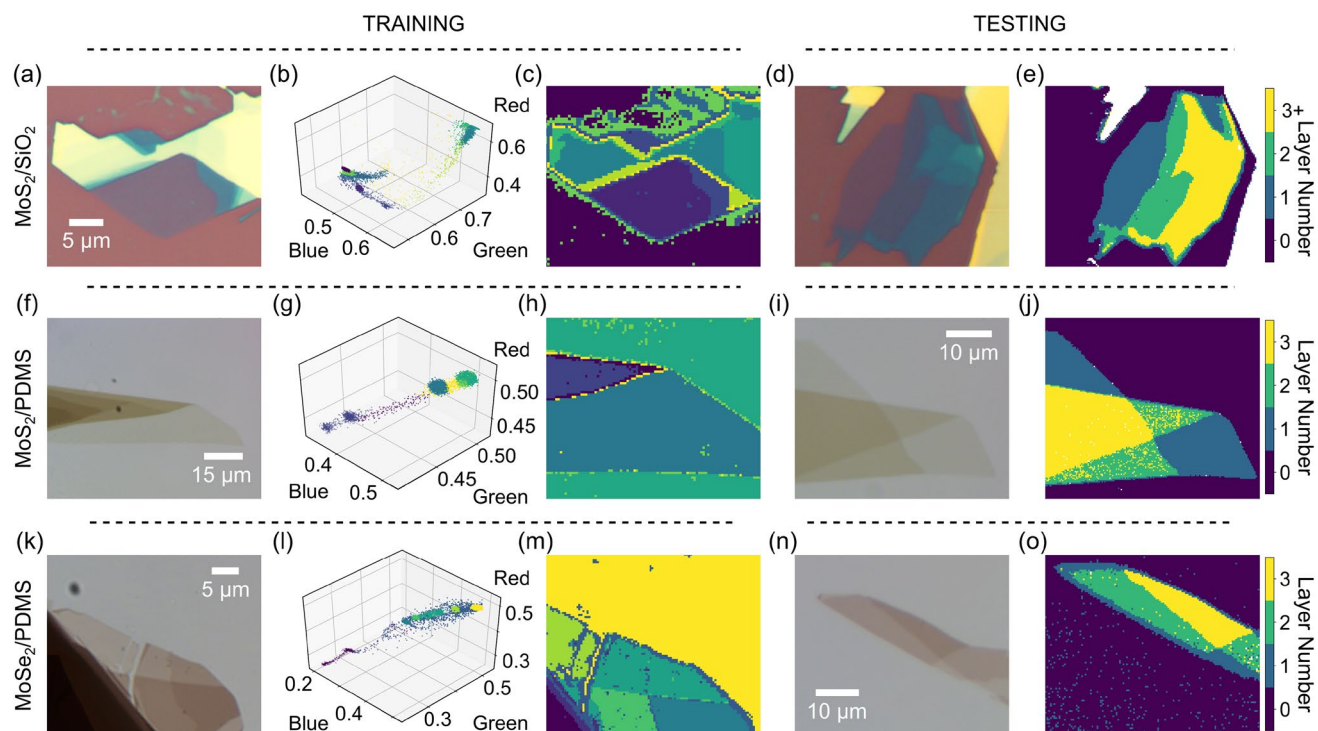
Figure 4. General thickness identification of 2D materials on opaque and transparent substrates. (**a**–**c**) An example training process for MoS_2 flakes exfoliated onto Si/SiO_2 substrates. (**a**) Raw optical image before preprocessing. (**b**) RGB scatter plot of the identified clusters with pixels colored according to the cluster they belong to. (**c**) Reconstructed image of the MoS_2 flake in (**a**) after the training step. (**d**,**e**) Testing process for MoS_2 on Si/SiO_2. (**d**) Raw optical image of an MoS_2 flake on Si/SiO_2. (**e**) Layer identification after testing the image in (**d**). (**f**–**h**) An example training process for $MoS_2/PDMS$. (**i**,**j**) An example testing process for $MoS_2/PDMS$. (**k**–**m**) An example training process for $MoSe_2/PDMS$. (**n**,**o**) An example testing process for $MoSe_2/PDMS$.

The training process is repeated for multiple flakes of the same material and substrate, saving their ellipsoid characteristics into the same catalogue. A master catalogue is then created by averaging together the characteristics of ellipsoids with like-thickness. This master catalogue is the tool with which we can test other images to determine their flake layer thicknesses (Fig. 1i,j).

General application to other materials and substrates

Our script can be universally applied to the identification of other 2D material thicknesses on opaque and transparent substrates. This generality is achieved by analyzing all three dimensions of the color-space data and fitting the resulting clusters of arbitrary shape with our GMM-EM algorithm. Importantly, no change in the adjustable parameters (ϵ or GMM convergence) are required for the following results.

Figure 4 displays the power of this generality by identifying the layer thickness of two additional materials, molybdenum disulfide (MoS_2) and molybdenum diselenide ($MoSe_2$), on opaque (Si/SiO_2) and transparent (polydimethylsiloxane (PDMS)) substrates. MoS_2 on Si/SiO_2 (Fig. 4a–e) presents clusters very similar to those of graphene on Si/SiO_2 but they are separated further in RGB space (Fig. 4b). Further training for this material/substrate combination would improve our testing results which only identify layer thicknesses of 1 and 2 (Fig. 4d,e). From the covariance matrices we note that while all the data clusters are technically triaxial ellipsoids (none of the semi-axes are equal), the clusters for materials on Si/SiO_2 are roughly prolate spheroids with one axis (blue) an order of magnitude larger than the other two semi-axes (red and green).

MoS_2 on PDMS (Fig. 4f–j) presents clusters again extending along the blue axis, though not as strongly as materials on Si/SiO_2. The clusters are similarly well-separated in RGB space as they are for MoS_2 on Si/SiO_2. Testing for this set identifies monolayer, bilayer and trilayer thicknesses (Fig. 4j). Finally, $MoSe_2$ on PDMS presents the most spherical ellipsoids of our investigation still slightly extending along blue, (Fig. 4k–o) and mono- through trilayer thicknesses are easily identified (Fig. 4o).

Discussion

Our investigation focuses on the development of a program that can be universally applied to different 2D materials and substrates. This requirement invariably introduces computation time when compared with other recent segmentation methods[22,50]. The training time, for example, reported in Ref.[50] for the entire program is roughly 31 h. Computation times for the training stage of our program depend on the image composition. A single layer image can take about 10 min but images with multilayer flakes (more clusters) take as long as 5 h. Our program results here are from training sets of roughly 10 images corresponding to about 10 h computation time. However,

this is a single event time cost because once the master catalogue is trained for a particular material and substrate combination, it can then be used repeatedly in the testing step, which is more efficient.

Image testing requires roughly one minute to identify layer thicknesses of new images. Computation time is sufficiently short for testing because image pixels are simply compared with the master catalogue. This time would allow in-situ identification of flakes from images taken by human inspection of a substrate. The time spent scanning between images can take several minutes. The time may also be sufficient for an automated scanning system such as that presented in Ref.[51]. Improvements in computation time may be sought through further image compression or possibly reducing the testing step to two dimensions of the three-dimensional RGB space, possibly blue and either green or red, similar to algorithms presented in ref.[23]. Although, the dropped color dimension would have to be identified for a particular material/substrate combination.

For each material/substrate combination investigated in this study, the pixel accuracy was determined by creating a ground truth image and comparing it, pixel-by-pixel, with the testing images (see Fig. S1 in the Supplemental Materials for details). Pixel accuracy was slightly better for materials on PDMS but overall, the program achieves an average accuracy of 95% for the materials and substrates investigated in this study. This pixel accuracy is comparable to that achieved in studies based on much larger training sets. Reference[50] reports pixel accuracy of 97% from a training set of 917 images. Based on these results, normalized confusion matrices for each combination were calculated showing the individual layer accuracy as well. Finally, we note that a clear advantage to our approach is the simplicity of our program which relies on well-known and proven clustering techniques with relatively high pixel accuracy from small training sets.

Conclusion

Summarizing, we have presented a code for the automatic identification of flake thicknesses that can be universally applied to a variety of 2D materials and substrates. The algorithm analyzes data clusters in RGB space of preprocessed optical microscopy images. It can accurately identify mono- and few-layer thicknesses with a pixel accuracy of 95%. We anticipate the program will be of use for a wide variety of materials and substrates for the continued interest and investigation into the properties and characteristics of 2D materials.

References

1. Clarke, L. P. *et al.* MRI segmentation: Methods and applications. *Magn. Reson. Imaging* **13**, 343–368. https://doi.org/10.1016/0730-725X(94)00124-L (1995).
2. Ji, Z. *et al.* Fuzzy local gaussian mixture model for brain MR image segmentation. *IEEE Trans. Inf. Technol. Biomed.* **16**, 339–347. https://doi.org/10.1109/TITB.2012.2185852 (2012).
3. Forouzanfar, M., Forghani, N. & Teshnehlab, M. Parameter optimization of improved fuzzy c-means clustering algorithm for brain MR image segmentation. *Eng. Appl. Artif. Intell.* **23**, 160–168. https://doi.org/10.1016/j.engappai.2009.10.002 (2010).
4. Feng, D. *et al.* Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Trans. Intell. Transp. Syst.* https://doi.org/10.1109/TITS.2020.2972974 (2020).
5. Schwarz, M., Milan, A., Periyasamy, A. S. & Behnke, S. RGB-D object detection and semantic segmentation for autonomous manipulation in clutter. *The Int. J. Robot. Res.* **37**, 437–451. https://doi.org/10.1177/0278364917713117 (2018).
6. Ratakonda, K. & Ahuja, N. Lossless image compression with multiscale segmentation. *IEEE Trans. Image Process.* **11**, 1228–1237. https://doi.org/10.1109/TIP.2002.804528 (2002).
7. Sanderson, H. & Crebbin, G. Image segmentation for compression of images and image sequences. *IEE Troc Vis. Image Signal Process.* **142**, 15–21. https://doi.org/10.1049/ip-vis:19951681 (1995).
8. Shen, L. & Rangayyan, R. A segmentation-based lossless image coding method for high-resolution medical image compression. *IEEE Trans. Med. Imaging* **16**, 301–307. https://doi.org/10.1109/42.585764 (1997).
9. Li, H. *et al.* Rapid and reliable thickness identification of two-dimensional nanosheets using optical microscopy. *ACS Nano* **7**, 10344–10353. https://doi.org/10.1021/nn4047474 (2013).
10. Bing, D. *et al.* Optical contrast for identifying the thickness of two-dimensional materials. *Opt. Commun.* **406**, 128–138. https://doi.org/10.1016/j.optcom.2017.06.012 (2018).
11. 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. https://doi.org/10.1007/s12274-019-2424-6 (2019).
12. Ni, Z. H. *et al.* Graphene thickness determination using reflection and contrast spectroscopy. *Nano Lett.* **7**, 2758–2763. https://doi.org/10.1021/nl071254m (2007).
13. Jung, I. *et al.* Simple approach for high-contrast optical imaging and characterization of graphene-based sheets. *Nano Lett.* **7**, 3569–3575. https://doi.org/10.1021/nl0714177 (2007).
14. Wang, X., Zhao, M. & Nolte, D. D. Optical contrast and clarity of graphene on an arbitrary substrate. *Appl. Phys. Lett.* **95**, 081102. https://doi.org/10.1063/1.3212735 (2009).
15. Zhang, H. *et al.* Optical thickness identification of transition metal dichalcogenide nanosheets on transparent substrates. *Nanotechnology* **28**, 164001. https://doi.org/10.1088/1361-6528/aa6133 (2017).
16. Yu, Y. *et al.* Investigation of multilayer domains in large-scale CVD monolayer graphene by optical imaging. *J. Semicond.* **38**, 033003. https://doi.org/10.1088/1674-4926/38/3/033003 (2017).
17. Wang, Y. Y. *et al.* Thickness identification of two-dimensional materials by optical imaging. *Nanotechnology* **23**, 495713. https://doi.org/10.1088/0957-4484/23/49/495713 (2012).
18. Rubio-Bollinger, G. *et al.* Enhanced visibility of MoS2, MoSe2, WSe2 and black phosphorus: Making optical identification of 2D semiconductors easier. *Electronics* **4**, 847–856. https://doi.org/10.3390/electronics4040847 (2015).
19. Jessen, B. S. *et al.* Quantitative optical mapping of two-dimensional materials. *Sci. Rep.* **8**, 6381. https://doi.org/10.1038/s41598-018-23922-1 (2018).
20. Yang, J. & Yao, H. Automated identification and characterization of two-dimensional materials via machine learning-based processing of optical microscope images. *Extrem. Mech. Lett.* **39**, 100771. https://doi.org/10.1016/j.eml.2020.100771 (2020).

21. Wu, B., Wang, L. & Gao, Z. A two-dimensional material recognition image algorithm based on deep learning. In *2019 International Conference on Information Technology and Computer Application (ITCA)*, 247–252 (2019). https://doi.org/10.1109/ITCA49981.2019.00061.
22. Masubuchi, S. *et al.* Deep-learning-based image segmentation integrated with optical microscopy for automatically searching for two-dimensional materials. *NPJ 2D Mater. Appl.* **4**, 1–9. https://doi.org/10.1038/s41699-020-0137-z (2020).
23. Masubuchi, S. & Machida, T. Classifying optical microscope images of exfoliated graphene flakes by data-driven machine learning. *NPJ 2D Mater. Appl.* **3**, 1–7. https://doi.org/10.1038/s41699-018-0084-0 (2019).
24. Lin, X. *et al.* Intelligent identification of two-dimensional nanostructures by machine-learning optical microscopy. *Nano Res.* **11**, 6316–6324. https://doi.org/10.1007/s12274-018-2155-0 (2018).
25. Li, Y. *et al.* Rapid identification of two-dimensional materials via machine learning assisted optic microscopy. *J. Materiomics* **5**, 413–421. https://doi.org/10.1016/j.jmat.2019.03.003 (2019).
26. Greplova, E. *et al.* Fully automated identification of two-dimensional material samples. *Phys. Rev. Appl.* **13**, 064017. https://doi.org/10.1103/PhysRevApplied.13.064017 (2020).
27. Cellini, F., Lavini, F., Berger, C., de Heer, W. & Riedo, E. Layer dependence of graphene-diamene phase transition in epitaxial and exfoliated few-layer graphene using machine learning. *2D Mater.* **6**, 035043. https://doi.org/10.1088/2053-1583/ab1b9f (2019).
28. Hu, X., Qiu, C. & Liu, D. Rapid thin-layer WS2 detection based on monochromatic illumination photographs. *Nano Res.* **14**, 1–6 (2020).
29. Dong, X. *et al.* 3d deep learning enables accurate layer mapping of 2d materials. *ACS Nano* **15**, 3139 (2021).
30. Aleithan, S. H. & Mahmoud-Ghoneim, D. Toward automated classification of monolayer versus few-layer nanomaterials using texture analysis and neural networks. *Sci. Rep.* **10**, 1–8 (2020).
31. Castellanos-Gomez, A. *et al.* Deterministic transfer of two-dimensional materials by all-dry viscoelastic stamping. *2D Mater.* **1**, 011002. https://doi.org/10.1088/2053-1583/1/1/011002 (2014).
32. Island, J. O. *et al.* Precise and reversible band gap tuning in single-layer MoSe2 by uniaxial strain. *Nanoscale* **8**, 2589–2593. https://doi.org/10.1039/C5NR08219F (2016).
33. Lippert, S. *et al.* Influence of the substrate material on the optical properties of tungsten diselenide monolayers. *2D Mater.* **4**, 025045. https://doi.org/10.1088/2053-1583/aa5b21 (2017).
34. Nguyen, D. C. *et al.* Visibility of hexagonal boron nitride on transparent substrates. *Nanotechnology* **31**, 195701. https://doi.org/10.1088/1361-6528/ab6bf4 (2020).
35. Chu, H. *et al.* Linear magnetoelectric phase in ultrathin MnPS$_3$ probed by optical second harmonic generation. *Phys. Rev. Lett.* **124**, 027601. https://doi.org/10.1103/PhysRevLett.124.027601 (2020).
36. Sterbentz, R. M. *Universal Image Segmentation* (2020). https://github.com/islandlab-unlv/Universal-image-segementation.
37. Comaniciu, D. & Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**, 603–619. https://doi.org/10.1109/34.1000236 (2002).
38. Zhou, Y.-M., Jiang, S.-Y. & Yin, M.-l. A Region-Based Image Segmentation Method with Mean-Shift Clustering Algorithm. In *2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, Vol. 2, 366–370 (2008). https://doi.org/10.1109/FSKD.2008.363.
39. Zhou, H., Wang, X. & Schaefer, G. Mean shift and its application in image segmentation. In *Innovations in Intelligent Image Analysis, Studies in Computational Intelligence* (eds Kwasnicka, H. & Jain, L. C.) 291–312 (Springer, 2011).
40. Ester, M., Kriegel, H.-P., Sander, J. & Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96*, 226–231 (AAAI Press, 1996).
41. Xu, R. & Wunsch, D. Survey of clustering algorithms. *IEEE Trans. Neural Netw.* **16**, 645–678. https://doi.org/10.1109/TNN.2005.845141 (2005).
42. Celebi, M. E., Kingravi, H. A. & Vela, P. A. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert. Syst. with Appl.* **40**, 200–210. https://doi.org/10.1016/j.eswa.2012.07.021 (2013).
43. Dhanachandra, N., Manglem, K. & Chanu, Y. J. Image segmentation using K-means clustering algorithm and subtractive clustering algorithm. *Procedia Comput. Sci.* **54**, 764–771. https://doi.org/10.1016/j.procs.2015.06.090 (2015).
44. Dhanachandra, N. & Chanu, Y. J. A survey on image segmentation methods using clustering techniques. *Eur. J. Eng. Res. Sci.* **2**, 15–20. https://doi.org/10.24018/ejers.2017.2.1.237 (2017).
45. Gupta, L. & Sortrakul, T. A gaussian-mixture-based image segmentation algorithm. *Pattern Recogn.* **31**, 315–325. https://doi.org/10.1016/S0031-3203(97)00045-9 (1998).
46. Nguyen, T. M. & Wu, Q. M. J. Dirichlet Gaussian mixture model: Application to image segmentation. *Image Vis. Comput.* **29**, 818–828. https://doi.org/10.1016/j.imavis.2011.09.001 (2011).
47. Permuter, H., Francos, J. & Jermyn, I. A study of Gaussian mixture models of color and texture features for image classification and segmentation. *Pattern Recogn.* **39**, 695–706. https://doi.org/10.1016/j.patcog.2005.10.028 (2006).
48. Ribeiro, H. L. & Gonzaga, A. Hand image segmentation in video sequence by GMM: A comparative analysis. In *2006 19th Brazilian Symposium on Computer Graphics and Image Processing*, 357–364. https://doi.org/10.1109/SIBGRAPI.2006.23 (2006).
49. Santosh, D., Venkatesh, P. N., Poornesh, P., Rao, L. N. & Kumar, N. *Tracking Multiple Moving Objects Using Gaussian Mixture Model* (2013).
50. Han, B. *et al.* Deep-learning-enabled fast optical identification and characterization of 2D materials. *Adv. Mater.* https://doi.org/10.1002/adma.202000953 (2020).
51. Masubuchi, S. *et al.* Autonomous robotic searching and assembly of two-dimensional crystals to build van der Waals superlattices. *Nat. Commun.* **9**, 1413. https://doi.org/10.1038/s41467-018-03723-w (2018).

Acknowledgements

Author contributions
J.O.I. conceived the project. R.M.S. coded the program with guidance from J.O.I. R.M.S. and K.L.H. provided optical images of 2D materials. All authors wrote and reviewed the manuscript.

Competing interests
The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41598-021-85159-9.

Correspondence and requests for materials should be addressed to J.O.I.

Reprints and permissions information is available at www.nature.com/reprints.

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