Automated Identification of Crystalline n-Layer TMD Flakes in Microscope Images for the Production of Van der Waals Heterostructures

Adam Robinson

April 30, 2020

Abstract

The characterization and study of Van der Waals heterostructures has been of significant interest in recent years. These structures exhibit novel and potentially useful properties that have driven efforts to improve the speed and efficiency with which they are constructed. One of the most laborious steps in the process of their construction is the identification of monolayers on a microscope slide. These slides often contain on the order of 10^4 (**check this**) crystalline flakes, the majority of which are not monolayers. Identification of these monolayers via an automated process, rather than manual inspection by a researcher, promises to reduce the number of man hours spent building them. We present a means of automatically inspecting microscope images and differentiating flakes from background images and noise. This method produces a dataset that is well-suited to unsupervised machine learning models that can be applied to classify these flakes by their thickness.

Introduction

Many of the most popular methods of building Van der Waals (vdW) heterostructures involve a probabilistic process by which thousands of microscopic flakes of varying sizes and shapes are deposited onto a microscope slide[1]. In order to build a heterostructure with the desired properties, a slide must be searched for crystals with a desired thickness, often a single layer of atoms. This search process can consume hours of a researchers time, because the ratio of polylayer to monolayer flakes produced by most current exfoliation processes is very high. Here, we present a means of automating this process using an unsupervised machine learning method and several common computer vision techniques. This process is designed to be integrated into an automated microscopy setup that images an entire microscope slide without human intervention.

This process is centered around differentiating crystal flakes from each other and from the background of an image. This is accomplished by using a Bayesian-Gaussian Mixture Model with a Dirichlet prior (BGMM-DP). This is an unsupervised machine learning algorithm that classifies members of an n-dimensional dataset into discrete clusters. Images are preprocessed using common computer vision techniques provided by the OpenCV library. A BGMM-DP model is fit to a subset of these images and is then used to classify each pixel in the image based on parameters that are determined during the fitting process. This process produces an image in which flakes and background are differentiated from each other. This method significantly improves that accuracy with which the boundaries of flakes can be determined. Once this segmentation process has completed, more traditional computer vision techniques are used to extract geometric properties from the differentiated image.

Theory

Visible light microscope images contain significant noise and color variation across flake surfaces. As a result, the use of classical computer vision algorithms often results in misidentification of noise and flake color variations as flake boundaries. (**Insert example image**). In order to accurately apply classical edge detection algorithms, it is necessary to produce an image that has distinct edges along the boundaries of flakes, without having boundaries inside of a flake. In the present work, this is accomplished by fitting a clustering

algorithm to the HSV color space of an image. A BGMM-DP model is fit and used to distinguish flakes from background and different flake thicknesses from each other.

Mixture Models

A mixture model is generally defined by a set of K components designed to represent the distribution of N observations. Here, N is a finite number of observations in M dimensions, where M is also finite. Each component (K) is a statistical distribution such as a Gaussian or Poisson distribution. In addition to the parameters that define the distribution for each component, each component is assigned a weight parameter, often denoted ϕ. A mixture model is designed to assign each observation to one of the K components. This is sometimes referred to as determining the class that each observation is a member of.

A Bayesian Gaussian Mixture Model uses K Gaussian distributions as components, but draws the parameters (mean and variance) for these distributions from another distribution. This distribution is referred to as a prior. In general, a Bayesian Gaussian Mixture model has one prior distribution for each of the parameters that define each of the K components of the model. These prior distributions may also be of different types. For example, the weight parameters ϕ, may be distributed according to Dirichlet distributions, while the means μ, may be distributed according to a Gaussian distribution.

Mixture models are considered unsupervised machine learning algorithms, because they do not require any knowledge about the classification of the observations they are built on. One of the most common methods for fitting a mixture model is called likelihood maximization. The likelihood function of a mixture component is generally defined as,

$$L\left(\Theta|\,\mathbf{X}\right) = \prod_{i=1}^{N} p\left(x_i|\,\Theta\right)$$

Here, Θ is the set of parameters that define the component distribution, $p\left(x_i|\,\Theta\right)$ is the probability of the observation x_i and $L\left(\Theta|\,\mathbf{X}\right)$ is the likelihood function. The Likelihood function is defined for each component of the mixture. Algorithms designed to maximize the likelihood often work with the log likelihood, $\log L\left(\Theta|\,\mathbf{X}\right)$. This is done because logarithms increase monotonically, but their derivatives increase with proportion to the reciprocal of their argument. This improves numerical accuracy by reducing the magnitude of values involved in the maximization process. The most common likelihood maximization algorithm is called the Expectation Maximization algorithm. This is designed to maximize the expectation value (mean) of the likelihood functions for all components. For Gaussian mixtures, this is often accomplished finding values of Θ, such that the gradient of the expectation with respect to Θ is zero.

Algorithm

In the present work, a Bayesian Gaussian Mixture Model with a Dirichlet Prior is used to differentiate crystal flakes from image background and from each other. The model is defined by K Gaussian distributions, each with a weight, denoted ϕ_i. Each Gaussian component is defined by a covariance matrix, $\boldsymbol{\sigma}_i$ and a mean vector, $\boldsymbol{\mu}_i$. As this is a Bayesian model, the parameters ϕ_i, $\boldsymbol{\sigma}_i$ and $\boldsymbol{\mu}_i$ do not hold definite values, but are distributed according to the following,

$$\begin{aligned}
\boldsymbol{\phi} &\sim \mathrm{Dir}\left(\boldsymbol{\alpha}\right) \\
\boldsymbol{\sigma}_i &\sim W_M\left(\mathbf{V}_i,\, n\right) \\
\boldsymbol{\mu}_i &\sim \mathcal{N}\left(\tilde{\boldsymbol{\mu}}_i,\, \tilde{\boldsymbol{\sigma}}_i^2\right) \\
i &\in [1,\, K]
\end{aligned}$$

Here, $A \sim B$ indicates that the variable A is distributed according to B. $\mathrm{Dir}\left(\boldsymbol{\alpha}\right)$ is the the Dirichlet distribution, $W_M\left(\mathbf{V}_i,\, n\right)$ is the Wishart distribution and $\mathcal{N}\left(\tilde{\boldsymbol{\mu}}_i,\, \tilde{\boldsymbol{\sigma}}_i^2\right)$ is the normal distribution. $\boldsymbol{\alpha}$ is a vector

of positive real values, $\mathbf{V}_i$ is an $M \times M$ positive definite matrix (often referred to as a scale matrix), n is the number of degrees of freedom for the Wishart distribution (M in the present work), $\tilde{\boldsymbol{\mu}}_i$ is the mean vector for the normal distribution and $\tilde{\boldsymbol{\sigma}}_i^2$ is a diagonal covariance matrix for the normal distribution. The Wishart distribution is used, because it generates a positive definite matrix, which is necessary to form a covariance matrix for a normal distribution. The mixture model is effectively defined by the hyper parameters $\boldsymbol{\alpha}$, $\mathbf{V}_i$, $\tilde{\boldsymbol{\mu}}_i$ and $\tilde{\boldsymbol{\sigma}}_i^2$, because they control the distribution of the variables that are used to calculate the likelihood function. The model is fit using the Expectation Maximization algorithm, which attempts to find

$$\text{argmax}\ \mathrm{E}\left[\log L\left(\Theta \middle| \mathbf{X}\right)\right]$$
$$\Theta = \{\boldsymbol{\alpha},\, \mathbf{V},\, \tilde{\boldsymbol{\mu}},\, \tilde{\boldsymbol{\sigma}}\}$$

where Θ collectively refers to the parameters that define the prior distributions and $\mathrm{E}\left[\log L\left(\Theta \middle| \mathbf{X}\right)\right]$ refers to the expectation of the logarithm of the likelihood function over the likelihood functions for all components.

Results and Analysis

Results Results

Discussion and Conclusion

Discussion Discussion

References

[1] s41699-018-0084-0

TODO

1. Implement k-means clustering as an option. This may be more performant than BGMM