

NORTHWESTERN UNIVERSITY

Utilizing Literature to Characterize Materials from Images

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Computer Science

By

Weixin Jiang

EVANSTON, ILLINOIS

June 2022

© Copyright by Weixin Jiang 2022

All Rights Reserved

ABSTRACT

Utilizing Literature to Characterize Materials from Images

Weixin Jiang

The past decade has seen the rapid progress of deep learning, which becomes a game-changing technique in different data-intensive domains, with the availability of large scale data, cost-effective computing hardware and more advanced learning theory and algorithms. Despite of the rapid progress of deep learning methods in daily-life applications, such as face recognition, video enhancement, image classification, there are some challenges that prevent the application of deep learning into more research fields, such as materials science.

Materials science have been developed through the empirical correlation of processing and properties for thousands of years. Recently, tons of experimental and simulated data are captured/produced everyday due to the fast image acquisition devices and super computing facilities. The success of deep learning techniques in other fields (e.g. computer vision) motivates researchers in materials science to develop more advanced algorithms to accelerate the process of discovering and designing new improved materials with desired

properties. Unfortunately, applying deep learning techniques in materials science still remains at its early stage and requires more efforts from researchers.

In this thesis, I will present my work in understanding the characterization of materials from images. One challenge in developing data-driven algorithms in materials science is the lack of well-labeled datasets (e.g. microscopy images). In fields that dealing with natural image classification or detection tasks, large amount of images are annotated by human annotators (e.g. ImageNet, MS-coco), however, it would be expensive and even not feasible in the field of materials science, due to its requirement of sufficient domain expertise. To this end, we present our work in construction of Materials dataset from scientific literature, in which we developed an effective tool to construct a self-labeled electron microscopy dataset of nanostructure images.

In the second part of the thesis, I will present our work on the interpretation of spectrographs. For the purpose of understanding the insights behind these measurements, data points are usually displayed in graphical form within scientific journal articles. However, it is not standard for materials researchers to release raw data along with their publications. As a result, other researchers have to use interactive plot data extraction tools to extract data points from the graph image, which makes it difficult for large scale data acquisition and analysis. Therefore, we propose the Plot2Spectra pipeline, which enables an efficient spectra data extraction from plot images in an fully automatic fashion.

As the last part of the thesis, I will present our work in deducing structure information from STEM (Scanning Transmission Electron Microscopy) measurements. Microscopic imaging providing the real-space information of matter in a large range of scale, which

plays an important role for understanding the correlations between structure (e.g. morphology, phase, atomic structure, surface facet, interfacial structure) and properties in the field of materials science. Thus, extracting the structural information (e.g. atomic positions) plays a very important role in exploration of the crystallographic phases, atomic configurations and the insights behind the structure related material-specific properties and performance. However, it is a challenging task to deduce the structure information from STEM measurements. To this end, we present a representation learning framework for HAADF-STEM image retrieval, named STEM2SIM, to deduce the structure information (e.g. crystalline structure) from the given STEM image by efficiently find the similar image (i.e. known structure) from a simulated dataset.

Acknowledgements

Eventually, I have reached the last stop of this PhD journey. Five years is quite a long way, even compared to human's life, it is still not a short period. Even though at the very beginning of this journey, I had strong confidence that I will make it to the end with a doctor's degree, I did not expect there are so many ups and downs, and of course, cry and laugh, sadness and happiness. In a word, the whole PhD experience is not only about doing research and gaining domain expertise, but also about how to be a decent person and how to be yourself. I cannot achieve these accomplishments and get to this point without the support of my advisors, my mentors, my friends and all the people around me. Thus, I would like to take the chance here, at the beginning of my thesis, to express my sincere thanks to all of them.

First I would like to thank my academic advisors Prof. Oliver Cossairt from Northwestern University and Dr. Maria Chan from Argonne National Lab. The connection between Ollie and I started almost six years ago, when I was pursuing my Master's degree at Tsinghua University. Back to that time, I decided to apply for PhD in the United States and was sending out cold emails around professors who had projects that match my background. Ollie replied and was quite interested in the research I had done. After a few back and forth emails or phone chatting, Ollie provided me the chance to join the Comp Photo Lab (CPL). It is my honor to work with all the people in the CPL and the past five years in the CPL was a great experience, I was always inspired by Ollie's

enthusiasm in exploring different research topics and open mind for broadly collaboration. Such freedom and flexibility allows me to have the chance to try different things and find out what I am really interested in. Ollie is always supportive, even though after trying to work on ptychography and structure light projects I made my mind to work more on computer vision and deep learning related research. I have learnt a lot from Ollie, not only the way to do good research or the way to tell a attracting story about your work, but also the attitude we should have for your life and the way to be a good person. Maria and I met in late 2018, when Ollie and I were looking for collaboration around. Maria was looking for students that have background of deep learning and computer vision. What a coincidence! After a few chats, I joined Maria's group in April 2019 as a visiting PhD student at Argonne National Lab. Maria's expertise in Materials Science and her passion in developing deep learning to accelerate the discovery and design of materials are really impressive. It was a great experience for me. From one side, material science is something new to me, such that I got the chance to walk out of my comfortable zone and learn something really different. From the other side, such experience to work on interdisciplinary research topic is quite extraordinary, walking me out of the "ivory tower" and get to know and think about something more practical or grounded. Besides, She also shows me a good example of what an excellent scientist is like.

Then I would like to thank Prof. Aggelos Katsaggelos and Prof. Jack Tumblin to be my committee members. Aggelos and I have been working on the ptychography project and the event-based vision project during my PhD study. He is rigorous about the detail of the algorithms and always shows his curiosity about what and why questions, which always impressed me. Jack is really a great expert in the field of imaging and graphics.

His critical thinking and insightful discussion always remind me to dig deeper into the research problem.

Also, I would like to thank all members in the CPL, IVPL and Argonne National Lab (ANL), I really enjoy working or chatting with them. I met Dr. Zihao Wang before I started my PhD study, he is a very nice person and gave me a brief overview about the lab. In the second year of my PhD, when I was looking for deep learning related projects, he brought me into the event-based vision project, in which I was able to explore the fusion between learning based methods and physics based methods. His domain expertise for computational photography and his hard-working show me what a good PhD should be like. Also, I would like to thank Dr. Fengqiang Li, Dr. Chia-kai Yeh, Prof. Florian Willomitzer, Dr. Kuan He, Prof. Ming Zhao, Prof. Jian Gao, Dr. Jason Holloway, Dr. Sushoban Ghosh, Dr. Yudong Yao, Florian Schiffers, Lionel Fiske, Sudarshan Nagesh, Fabian Wagner, Bingjie Xu and Jiazhang Wang. They have created a very nice working place in CPL, which means a lot to me. Also, working with colleagues in IVPL was a great experience, Srutarshi Banerjee and Henry Chopp are very nice and hardworking PhD candidates. During my visiting at ANL, I met a lot of friends and I enjoyed working with them. Dr. Eric Schwenker worked close to me ever since I joined ANL, he is really talent, not only in Materials Science, but also in Computer Science. His insightful analysis for algorithms/systems and capability for project management always impressed me. Also I would like to thank Dr. Arun Baskaran, Dr. Joydeep Munshi, Dr. Mingren Shen, Dr Luqing Wang, Trevor Spreadbury, Yiming Chen and Haili Jia, working and chatting with them is always enjoyable.

Outside of the campus, I would like to thank Dr. Jue Wang and Dr. Vishy Swaminathan for giving me the chance to have internships and experience how research is like in industry. I had my internship at Megvii Research Lab in the summer of 2019 at Redmond, WA. Dr. Wang is a very nice person and allows me to explore different directions of research during the internship. I had my second internship at Adobe Research in the summer of 2021 under the supervision of Vishy. During that time, I was working on something quite new to me, but Vishy and other mentors were very patient even when the project did not work out well. These two internship experiences are really precious to me, along with the nice people I met during such experiences such as Dr. Haibing Huang, Dr. Kai Li, Dr. Xue Bai, Dr. Jianchao Tan, Dr. Heming Zhang, Dr. Xuewen Yang, Jiayuan Shi from Megvii and Dr. Gang Wu, Dr. Haoliang Wang, Dr. Stefano Petrangeli, Yang Li, Kai Wang from Adobe.

Besides, staying in a different country far away from my hometown is lonely, I would like to thank the help and accompany from my friends. Mr. Libing Song (1st year), Mr. Yukun Ma (3rd year) and Mr. Weijian Li (4th year and after) are my roommates during my stay in Evanston, I would really like to express my appreciation to the kindness that they offer. I got to know Dr. Kai Li ever since my master's study, during which he was a senior PhD student in our lab. But we finally met each other when I was doing my internship at Megvii. He is really knowledgeable in the field of computer vision, and offers a lot of impressive suggestions and help ever since then, not only in research but also in daily life. Mr. Jiachen Li is one of my best friends. We got to know each other ever since my undergraduate study, during which we were roommates. He is a very nice person,

hardworking and mature. I learned a lot from him, such as, rationality, confidence and optimism.

Lastly, I would express my appreciation to my parents, who give me their unconditional love and support. It is their timely encouragement and endless patience that help me get through the up and down in my life.

It is the last stop of my Phd study, but also the beginning of my next chapter. What I take away from my research experience is that, never be afraid of something uncertainty, as long as you keep trying, as long as you keep updating, you will get closer and closer to the optimal solution. As the old Chinese saying goes, "I should establish myself in the society according to the manners when it comes to the my thirties." Future is unpredictable, but I am ready for it.

Table of Contents

ABSTRACT	3
Acknowledgements	6
Table of Contents	11
List of Tables	14
List of Figures	15
Chapter 1. Introduction	28
1.1. Statement of the Problem	30
1.2. Summary and Outline of the thesis	33
1.3. Disclaimer	35
Chapter 2. Related Work	37
2.1. Deep Learning in Computer Vision	37
2.2. Deep Learning in Materials Science	44
2.3. Representation Learning	46
Chapter 3. Construction of Materials datasets from scientific literature	52
3.1. Introduction	52
3.2. Design Overview	54

	12
3.3. Technical Details	62
3.4. Results	72
3.5. Conclusion	85
Chapter 4. Spectra Data Extraction from Spectrographs	86
4.1. Introduction	86
4.2. Method	88
4.3. Experiments	94
4.4. Ablation study	99
4.5. Conclusion and discussion	104
Chapter 5. Representation learning for STEM images	107
5.1. Introduction	107
5.2. Methods	110
5.3. Results	114
5.4. Conclusion	120
Chapter 6. Other Research Summary	121
6.1. Event-driven video frame synthesis	121
6.2. Task-oriented near-lossless burst image compression	129
Chapter 7. Conclusion	141
Appendix A. Supplementary Material for Compound Figure Separation	143
A.1. Study of the insights behind the proposed subfigure label detection	143
A.2. Guided detection for compound figure separation	144

	13
A.3. More compound figure separation results	152
References	156

List of Tables

3.1	Comparison of the performance of different compound figure separation methods.	69
3.2	Comparison of the performance ($AP_{0.5}$) of training subfigure label detector with/without module decoupling.	71
4.1	Axis misalignment with different anchor-free object detection models.	100
4.2	Quantitative comparison of plot data extraction with different line width or line style.	104
5.1	Results of applying different projection heads in the contrastive learning branch.	116
5.2	The comparison of different methods on average precision.	117
A.1	Results comparison of subfigure detection.	151
A.2	Results comparison of subfigure label detection.	152

List of Figures

- 3.1 An Overview of the EXSCLAIM! Pipeline. The schematic highlights the path from a user-defined query to the exsclaim output data structure (both JSON objects). After the query is submitted to the pipeline by calling the run method, the journal scraper extracts figure/caption pairs from articles in the specified journal family, the caption distributor divides the caption text into segments that are consistent with the images in the figure, and the figure separator computes bounding boxes that separate and classify the individual images from the full figure. The caption assignment method is effectively the “self-labeling”, as it takes all of the caption segments, reduces them to individual keywords if possible, and pairs them with their corresponding image within the full figure. The final output data structure, i.e. the exsclaim output file, contains all the descriptions and references necessary to construct the full labeled imaging dataset. 55
- 3.2 Master-Dependent-Inset (MDI) Model. The panel on the left shows the initial figure before annotation, and the panel on the right is a copy that has been properly annotated according to the Master-Dependent-Inset (MDI) model. The presence of a subfigure

label is necessary for all images given a “master image” designation. Notice the master image corresponding to subfigure label “b” contains two dependent images (illustration and graph). Since the master image corresponding to subfigure label “b” governs more than one distinct image (i.e. the caption text associated with “b” will refer to all the images in the series), this master image is classified as a parent. Additional image features such as the scale bar and scale bar label (when present) are also identified.

58

3.3 Mechanical Turk GUI for Figure Separation Annotation. A snapshot of the graphical user interface (GUI) that is accessible via the MTurk worker platform (<https://www.mturk.com/worker>) is included below. MTurk workers with an extensive record of positive work on the platform (both in number of tasks completed and overall accuracy when it comes to task acceptance) have the proper credentials to access and complete the “HITS” (Human Intelligence Tasks) when they’re posted to the site. Workers can select the various categories at the bottom that are used to describe the objects in the figure, and after drawing a bounding box around the object, they are asked to either further classify the image (i.e. choosing diffraction as the image type in the example below), or transcribe the text in the case of subfigure labels and scale bar labels. All workers must follow the directions closely and are required to acknowledge that they understand the task before submission. When one of the members of

the EXSCLAIM! project reviews the HIT submitted by the worker, if directions are not followed, or it is clear that the task was completed in a rushed or careless fashion, the HIT is rejected. This helps ensure the quality of the training dataset created for this task of figure separation. 61

3.4 An example of compound figure separation following authorial intent: (a) shows the original compound figure. (b) shows the results from the proposed compound figure separation method, in which each subfigure is associated with the descriptive information from the caption and its image type. Image source: (Wu et al. Nanoscale, 2012) 63

3.5 Overview of the proposed method. The compound figure is fed into the subfigure label detector to locate subfigure labels. The location of the subfigure labels is concatenated with the compound figure to generate a 4D image tensor, which is then fed into the master image detector to get the segmentation result. 64

3.6 Distribution of different subfigure labels. 65

3.7 A step-by-step illustration of the subfigure detection module. The binary mask containing the global layout information from the detected subfigure labels is concatenated with the compound figure to provide extra features. Then the location of subfigure labels is used to select anchor feature vectors. Latent feature vectors are then computed from the the anchor feature vectors for a more accurate prediction. 67

3.8 Comparison of different compound figure separation algorithms. (a). An example compound figure with its respective caption. Subfigure labels (the ones in red circles) appear in both the compound figure and the caption, bridging the connection between them. (b). The separation result by the proposed framework, which detects subfigures along with their respective subfigure labels. (c). The separation result by [145]. (d). The separation result by [218]. Image source: (Wu et al. Nanoscale, 2012)

70

3.9 Precision and Recall for Image Classification. The confusion matrices highlight the nature of the mistakes made in each classification scenario (i.e. looking at a column and row for the same image type, the intersection represents the true positive number while the associated false positives are the column entries and the associated false negatives are rows) at two confidence thresholds (a) no threshold, and (b) high threshold for $N=3555$ images. In both cases, the precision scores are adequate, particularly in the case of the microscopy, graph, and diffraction images. Recall suffers across the board as correctness is emphasized over completeness in the design of the pipeline. Images in (c) highlight some of the more easily rationalized examples of false positive microscopy classifications. — DOIs for articles containing the example images (left to right): [10.1038/ncomms14925](https://doi.org/10.1038/ncomms14925), [10.1038/srep08722](https://doi.org/10.1038/srep08722), [10.1038/ncomms4631](https://doi.org/10.1038/ncomms4631).

73

3.10 Distribution of Image Types and Keyword Labeling Accuracy.

The query in this example is used to extract electron microscopy images of general nanostructures from Nature journals. The bar plot in (a) shows the distribution of image types extracted at two different thresholds. The bar plots in (b) and (c) further subdivide the population of high confidence microscopy images. In (b), the distribution of label types are recorded. In this context, single and multi-label refer the existence of a keyword label. Label unassigned (uas) means that caption text has been distributed to the image, but no keyword label from the query exists. Caption unassigned (uas) refers to a scenario where the caption distributor was not able to confidently distribute a proper substring to caption text. The bar plot in (c) represents the distribution of labels in the top 10% of retrieved microscopy images and provides a measurement of the joint probability that an image classified as microscopy and given the corresponding subsequent label is a true microscopy image represented by the given label.

76

3.11 Examples of Images Extracted and Labeled with EXSCLAIM! The extracted images contain both a keyword label from a word family associated with the query, as well as grammatically sufficient sequence of distributed caption text. Some caption distribution examples are simple, as in (a) where all the words distributed are linearly connected, however, the current caption distributor extraction class

is also designed to adequately capture more complex structural dependence relationships (b-c) where the subject is separated from the text completing the full consistent description.

79

3.12 Accuracy of scale bar label detection. The plot in (a) shows how accuracy varies as a function of confidence threshold, including the number of images present at a given threshold. For all thresholds shown, there are at least 500 samples. When the confidence level associated with the scale bar label detection is 0.6, the overall accuracy (percentage of scale bar labels that are exactly right – both the number the unit) is ≈ 0.95 . The confusion matrices in (b) and (c) highlight the accuracy of the predicted number and unit components for the scale bar label recognition. Larger scales (i.e. mm and cm) were not adequately represented in the training set, so they are not part of the test set. Both number and scale recognition accuracy is high at the 0.2 threshold ($\sim 92\%$ and $\sim 99\%$ respectively for the labels shown). The examples in (d) show common instances of the low-resolution and low-contrast conditions that are responsible for a majority of the prediction errors.

81

3.13 Example Image Labels after Hierarchical Label Assignment. (a-g) Examples of hierarchical label assignment for images containing a properly distributed caption. For each image, the caption labels are limited to caption text only. Conversely, the abstract labels are free to draw additional relevant words from the abstract text,

and the topic labels come from the human-assigned topic names from the LDA topic summaries. — DOIs for articles containing the example images (natural reading order): 10.1038/s41598-019-40198-1, 10.1038/s41467-019-12142-4, 10.1038/s41467-019-12885-0, 10.1038/s41598-019-55803-6, 10.1038/srep17716, 10.1038/am2016167, 10.1038/srep42734.

83

- 4.1 An overview of the proposed Plot2Spectra pipeline. An example of XANES graph images is first fed into the axis alignment module, which outputs the position of axes, the values of the ticks along x axis and the plot region. Then the plot region is fed into the plot data extraction module which detects plot lines. Figure is from Ref. [173]. 87
- 4.2 Examples of axis misalignment. (a) There are noticeable gaps between the left/bottom edges of the bounding boxes (blue boxes) and the axes. (b) The left/bottom edges of the bounding boxes (red boxes) are perfectly aligned with the the axes. Figures are from Refs. [179, 211, 201] (left to right, top to down). 89
- 4.3 Results of plot line detection with different Δ_g . Too large or too small values fail to have a correct detection. Figures are from Refs. [57, 194] (top to down). 92
- 4.4 Comparison of the results of plot data extraction between instance segmentation algorithms and the proposed method. The first column is the input plot images and the rest columns are plot line detection

	results with different methods. All pixels belonging to a single line (data instance) should be assigned to the same color. Figures are from from Ref. [74, 1, 173] (top to down).	95
4.5	Plot line detection with imperfect semantic segmentation results. (a). Part of plot lines is missing in the probability map. Figures are from Refs. [69, 57] (top to down). (b). Background pixels are misclassified as foreground. Figures are from Refs. [164, 238] (top to down).	96
4.6	Plot line detection with hard examples. Figures are from Refs. [233, 43] (left to right).	97
4.7	Visual comparison of plot data extraction with different losses. Figure is from Ref. [244].	100
4.8	Results of plot line detection with different start position. Start positions are highlighted in the detection result images. Figures are from Refs. [234, 244] (top to down).	101
4.9	Quantitative comparison of plot data extraction with different start positions.	102
4.10	Plot line detection with simulated plot image with different line styles.	103
4.11	Failure cases. (a) Plot2Spectra fails when there is a significant peak. Figures are from Refs. [124, 11] (top to down). (b) Plot2Spectra fails when a large portion of background/foreground pixels are misclassified. Figures are from Refs. [47, 27] (top to down).	105
5.1	The proposed framework for HAADF-STEM image retrieval.	108

- 5.2 A typical pipeline for STEM image retrieval. The query image and all images in the database are first fed into the encoder to generate feature representations in the feature space, then the similarity measurement module computes the similarity between the feature representation of the query image and the feature representations of all images from the database. Then the retrieval system pops out the top similar candidates as the result. 109
- 5.3 Examples of similarity for STEM measurements. (a). A reference STEM image (ICSD id: 4, space group number: 165). (b) A visually similar but semantically different example (ICSD id: 9508, space group number: 173). (c) and (d) refer the the same structure prototype as (a), (c) is generated by adding the source size broadening effect with Gaussian blur and constant background bias while (d) is measured from a different viewing angle. 112
- 5.4 Distribution of average precision with different projection heads. (a). The distribution of average precision over the testing set which is trained with the contrastive learning branch only. (b). The distribution of average precision over the testing set which are trained with the contrastive learning branch and classification branch with space group number prediction. (c) The distribution of average precision over the testing set which are trained with the contrastive learning branch and classification branch with structure prototype prediction. 116

- 5.5 Manifold of learned feature representation with tSNE. (a) ResNet_{pt}. (b) ResNet_{sgn}. (c) ResNet_{icsd}. (d) ResNet_{cts}. (e) ResNet_{cts+sgn}. (f) ResNet_{cts+icsd}. (g) SSIM. 118
- 5.6 An example of STEM image retrieval with different methods (left to right the similarity score decreases), in which images in green are true positive proposals and images in red are false positive proposals. (a) ResNet_{pt}. (b) ResNet_{sgn}. (c) ResNet_{icsd}. (d) ResNet_{cts}. (e) ResNet_{cts+sgn}. (f) ResNet_{cts+icsd}. 119
- 6.1 We propose a fusion framework of intensity image(s) and events for high-speed video synthesis. Our synthesis process includes a differentiable model-based reconstruction and a residual “denoising” process. 123
- 6.2 Forward models considered in this section. Case 1: interpolation from two observed intensity frames and event frames. Case 2: prediction from one observed intensity frame at the beginning and event frames. Case 3: Motion video from a single observed intensity frame and event frames. 124
- 6.3 Frame prediction. Given a start frame and the future events (a), we predict the end frame (ground truth omitted). Our results using DMR alone outperforms existing algorithm, Complementary Filters (CF) [182]. 127

- 6.4 Comparison of denoising performance. Our learned Residual Denoiser (RD) reconstructs the intermediate frame (1-frame interpolation case) with fewer motion artifacts. 128
- 6.5 Overview of proposed near-lossless image compression system. 132
- 6.6 Lipschitz constant estimation when considering super-resolution as the burst downstream processing task. 138
- 6.7 Proposed Nearlossless Compression method’s BPSP over Tau on 10bit Burst images. 139
- A.1 Intuition of adding the classification constraint along with the high IoU constraint. (a) is a compound figure, where the letter "b" in the red box is picked out. (b) We randomly crop patches around the red box in the compound figure and sort the patches by their IoU score associated with the ground truth bounding box. Then a few patches are selected and fed into the subfigure label classifier. The number below each patch is its IoU score. The patches in green box are correctly classified while the ones in red box are misclassified as "h". 144
- A.2 Master image detection with different overall layout settings. (a-c) show that different CFS results of the same compound figure when given different overall layout information. Blue, green and orange bounding boxes mean that the image type of the subfigure is graph, microscopy, and parent respectively. Image source: (Zhao

		26
		145
A.3	Pipeline of label guided subfigure detection. (zoom in for better visualization.)	149
A.4	Pipeline of subfigure guided label detection. (zoom in for better visualization.)	150
A.5	Examples of similar compound figures with different overall layout information. The compound figure on the left panel is similar to the one on the right panel, in which each row consists of a microscopy image and a graph image. More subfigure labels imply a finer-level decomposition (right) while less subfigure labels lead to a much coarser level decomposition. Images come from [181, 73]	153
A.6	Examples of similar compound figures with different local layout information. These two compound figures are similar except the second row. The second row of the figure on the left panel contains only one subfigure label, making it a parent image. The second row of the figure on the right panel contains four subfigure labels, making it decomposed as four separated microscopy images. Images come from [30, 81]	154
A.7	Example of incorrect compound figure separation. The proposed compound figure separation approach fails due to the incorrect subfigure label detection. (a). Subfigure label "b" is missing during the subfigure label detection process, which causes a	

under-segmentation (b). Some internal feature of the image is being misidentified as the subfigure label "c", which causes an over-segmentation. (c). The actual subfigure label "b" is missing, while the letter "b" of the coordinate system is being misidentified as a subfigure label. Images come from [230, 240, 155]

CHAPTER 1

Introduction

Deep learning (DL) [111] is a special kind of machine learning (ML) [146, 16], which is an algorithm that is able to learn from data. A popular definition of learning [141] in the context of computer program is "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E ". Ever since 1980s, machine learning has evolved from a field of laboratory demonstrations to a field of significant commercial value. However, the limitation of conventional machine learning algorithms is their ability to process natural data in their raw form, which usually requires careful engineering or considerable domain expertise to transform the data from the raw form to some suitable feature representation. A notorious problem is known as the curse of the dimensionality, that is, machine learning problems become exceedingly difficult when the number of dimensions in the data is high. The insights behind this phenomenon is that with the increase of the dimensions, the number of possible distinct configurations of the variables increases exponentially, resulting in most configurations with no training examples associated. However, in deep learning, it requires very little engineering by hand, instead it takes advantage of increases in the amount of available computation and data to learn features from the raw-form data using a general-purpose learning procedure.

For the past decade, the availability of large scale datasets (e.g. ImageNet¹, MS-coco²), the widespread of powerful yet affordable computing resource (e.g. Graphics Processing Unit (GPU), 10 or 20 times faster regular Central Processing Unit (CPU)) and the rapid progress in developing advanced algorithms give rise to the emergence of deep learning techniques. Nowadays, deep learning has become a game-changing technique and dominant in daily-life applications (e.g. social media [13, 112], healthcare [165, 12]) and data-intensive research domains (e.g. image recognition [110, 55, 215, 207], speech recognition [140, 75, 178], material science [166, 167, 183, 2]).

The development of deep learning is not always that smooth. Back to the early days of this century, the mainstream computer-vision and machine-learning communities mainly focus on developing elegant hand-engineered feature extractors with domain expertise and mathematical knowledge. At that time, only relatively small models can be trained on relatively small dataset with CPUs in reasonable time, regardless the success of convolutional neural networks (ConvNets) in the field of object detection, segmentation and recognition. Also the concern that the non-convexity of deep learning models cannot be simply optimized by first-order optimization methods (e.g. gradient descent) prevents the spread of deep learning. Only since the ImageNet competition in 2012, when deep ConvNets achieved spectacular results with a dataset of about a million images from the web that contained 1,000 different classes [52], the communities started to pay significant attention to ConvNets. Now, ConvNets are the dominant approach for almost all recognition and detection tasks [214, 208, 188, 62, 200] and approach human performance on some tasks.

¹<https://www.image-net.org/>

²<https://cocodataset.org/home>

Very soon, the success of deep learning techniques becomes widespread, not only in computer science, but also in other research fields, such as Materials Science. Understanding the Processing–Structure–Property–Performance (PSPP) relationships [154] is key to discover and design new improved materials with desired properties. In fact, almost everything in materials science depends on these PSPP relationships, where the cause–effect relationships of science go from left to right, and the goals–means relationships of engineering go from right to left. For millennia, materials have been developed through the empirical correlation of processing and properties, in which the prevalent practice of empirical development involves minimal up-front theoretical analysis and a large amount of relatively superficial evaluation of prototypes that leads to empirical correlations that produce materials with limited predictability of behavior. The next paradigm of materials research comes with the increasing cost of experiment and decreasing cost and increasing power of computation-based theory, large and complex theoretical models (system of equations) became solvable via large-scale simulations of real-world phenomena. For the past decade, a new paradigm of science, known as the (big) data-driven science, has emerged to be the dominant approach with the explosive growth in the generation of data from the previous paradigms (i.e. both experimental data and simulation data).

1.1. Statement of the Problem

In this new paradigm of data-driven science, we would like to develop deep learning models to understand the characterization of materials. A big obstacle that prevents deep learning applications in the field of materials science is the lack of large scale dataset or

well labeled dataset. In a windfall of recent improvements in image resolution and acquisition speed, materials microscopy is experiencing an explosion of published imaging data. The standard publication format, while sufficient for traditional data ingestion scenarios where a select number of images can be critically examined and curated manually, is not conducive to large-scale data aggregation.

Journal articles have long been the standard for communicating important advances in scientific understanding. As sophisticated measurement and visualization tools render scientific communication more intricate and diverse, the visual presentation of scientific results as figures in these articles is noticeably more complex – especially within journals considered high-impact. With this complexity, which is often a byproduct of the “compound” layout of the figures (i.e. figures containing multiple embedded images, graphs, and illustrations, etc.), the meaning of each individual image as a standalone entity is not always apparent. The result is that individual images are not only unsearchable, but the effort required to extract them into a machine-readable format is significant. **With a mechanism for automatic dataset construction that includes both separating out individual images from compound figures, as well as providing concise annotations describing key aspects or classification of the image content, much more of the scientific imaging data in literature could be utilized for training and developing DL tasks.**

Spectroscopy, primarily in the electromagnetic spectrum, is a fundamental exploratory tool in the fields of physics, chemistry, and astronomy, allowing the composition, physical structure and electronic structure of matter to be investigated at the atomic, molecular and macro scale, and over astronomical distances. Different types of spectroscopies, such

as X-ray absorption near edge structure (XANES) and Raman spectroscopy, play a very important role in analyzing the characteristics of different materials. For the purpose of understanding the insights behind these measurements, data points are usually displayed in graphical form within scientific journal articles. However, it is not standard for materials researchers to release raw data along with their publications. As a result, other researchers have to use interactive plot data extraction tools to extract data points from the graph image, which makes it difficult for large scale data acquisition and analysis. In particular, high-quality experimental spectroscopy data is critical for the development of machine learning (ML) models, and the difficulty involved in extracting such data from the scientific literature hinders efforts in ML of materials properties. **It is therefore highly desirable to develop a tool for the digitization of spectroscopy graphical plots in a fully automatic fashion.**

Scanning transmission electron microscopy (STEM) uses focused electron beam to raster-scan over the sample to capture the structure of the materials in atomic resolution. Previous paradigm in materials science research allows simulations of complex real-world phenomena based on the theoretical models, e.g. density functional theory (DFT) and molecular dynamics (MD) simulations. From the engineering perspective of materials design and discovery, the inductive goal-means relations aims at predicting the structure and properties of the materials given the observation of the performance (i.e. inverse models). The construction of inverse models is typically formulated as an optimization problem wherein a property or performance metric of interest is intended to be maximized or minimized, subject to the various constraints on the representation of the material, which is typically in the form of a composition- and/or structure-based function.

The optimization process usually involves multiple invocations of the forward model, and these inverse relationships are one-to-many (i.e. multiple optimal solutions). **Therefore, solving such inverse models could be time-consuming and even infeasible in practice, developing a DL-based model to interpret STEM measurements in a efficient and effective manner is highly desirable.**

1.2. Summary and Outline of the thesis

In this thesis, we develop deep learning models to help understand the characterization of materials from images. The outline of this thesis is summarized as below.

In Chapter 2, we first review previous work on deep learning applications in the field of computer vision and materials science. We then dive deep to understand the insights of deep learning as representation learning. In Chapter 3, we develop the EXSCLAIM!, an automatic pipeline for construction of labeled materials imaging dataset from literature. In particular, I will get into more details about the design of the compound figure separation module. In Chapter 4, we develop the Plot2Spectra, which transforms plot lines from the graph image into sets of coordinates in an automatic fashion. In Chapter 5, we develop the STEM2SIM, which interpret the structure and properties of the STEM measurements in an efficient and effective manner. In Chapter 6, we talk some other deep learning related research in the field of computer vision. In Chapter 7, we summarize the whole thesis and talk about some future work that are worth exploring. We summarize major contributions of the thesis as below:

- We present a tool for the automatic EXtraction, Separation, and Caption-based natural Language Annotation of Images (EXSCLAIM!) for scientific figures and

demonstrate its effectiveness in constructing a self-labeled electron microscopy dataset of nanostructure images. In particular, we propose a two-stage framework to address the proposed compound figure separation problem. In the first stage, the subfigure label detection module detects all subfigure labels. Then, in the subfigure detection module, the detected subfigure labels help to detect the subfigures by optimizing the feature selection process and providing the global layout information as extra features. The EXSCLAIM! tool is developed around materials microscopy images, but the approach is applicable to other scientific domains that produce high-volumes of publications with image-based data as well as graphs and illustrations.

- We develop Plot2Spectra, which transforms plot lines from the graph image into sets of coordinates in an automatic fashion. Specifically, the plot digitizer is a two-stage framework. The first stage involves an axis alignment module. We first adopt an anchor-free object detection model to detect plot regions, and then refine the detected bounding boxes with the edge-based constraint to force the left/bottom edges to align with axes. Then we apply the scene text detection and recognition algorithms to recognize the ticks along the x axis. The second stage is the plot data extraction module. We first employ semantic segmentation to separate pixels belonging to plot lines from the background, and from there, incorporate optical flow constraints to the plot line pixels to assign them to the appropriate line (data instance) they encode.
- We develop STEM2Sim, a representation learning framework for STEM images, which deduces the structure information (e.g. crystalline structure) from the

given STEM image by efficiently find the similar image (i.e. known structure) from a simulated dataset. In particular, we introduce two different types of downstream task, the classification branch and the similarity measurement branch, to construct a proper encoding module, such that it is able to extract the visual and semantic information from the given image and encode the information into a compact feature representation.

1.3. Disclaimer

The mathematical derivations and formulations, results and a large part of the text in the thesis are adapted from [203, 94, 91, 93, 26, 185, 90, 186, 92, 231, 245, 246]:

- Weixin Jiang, Eric Schwenker, Trevor Spreadbury, Kai Li, Maria K.Y. Chan and Oliver Cossairt. "Plot2Spectra: an Automatic Spectra Extraction Tool." arXiv:2107.02827.
- Eric Schwenker, Weixin Jiang, Trevor Spreadbury, Nicola Ferrie, Oliver Cossairt and Maria Chan. "EXSCLAIM! – An automated pipeline for the construction of labeled materials imaging datasets from literature." arXiv:2103.10631.
- Weixin Jiang, Eric Schwenker, Trevor Spreadbury, Nicola Ferrie, Maria K.Y. Chan and Oliver Cossairt. "A Two-stage Framework for Compound Figure Separation." *In Image Processing (ICIP), IEEE International Conference on* , pp. 1204-1208, 2021.
- Weixin Jiang, Eric Schwenker, Trevor Spreadbury, Oliver Cossairt and Maria K.Y. Chan. "A Hybrid Image Retrieval System for Microscopy Images." *Microscopy and Microanalysis* 2021.

- Eric Schwenker, Weixin Jiang, Trevor Spreadbury, Sarah O'Brien, Nicola Ferrie, Oliver Cossairt and Maria Chan. "Constructing Self-Labeled Materials Imaging Datasets from Open Access Scientific Journals with EXSCLAIM!" *Microscopy and Microanalysis* 2020.
- Zihao Wang, Weixin Jiang, Kuan He, Boxin Shi, Aggelos Katsaggelos, and Oliver Cossairt. "Event-driven video frame synthesis." *In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops* , pp. 0-0. 2019.
- Weixin Jiang*, Gang Wu*, Vishy Swaminathan, Stefano Petrangeli, Haoliang Wang, Ryan Rossi and Nedim Lipka. "Task-Oriented Near-Lossless Burst Compression." *Manuscript submitted for publication*, 2022

CHAPTER 2

Related Work

2.1. Deep Learning in Computer Vision

Computer vision [67, 25, 226] is an interdisciplinary field of artificial intelligence that aims to guide computers and machines toward understanding the contents of digital data (i.e., images or video), such that computers may understand human actions, behaviors, and languages similarly to humans. A basic computer vision task is image classification, which aims at recognizing the presence of the object in the given image. The computer vision community shifted the attention from hand-crafted methods to deep learning since 2012, when the deep CNN model achieved spectacular results in the classification task, with a dataset of about a million images from the web that contained 1,000 different classes. Since then, more and more deep learning based computer vision techniques have been widely used in daily-life applications, such as pedestrian detection, autonomous driving, biometric systems, the movie industry, driver assistance systems, video surveillance, and robotics as well as medical diagnostics and other healthcare applications.

Decades ago, computer vision tasks were solved by hand-crafted methods. For example, given the image classification task, researchers tended to design some meaningful feature extractor with their domain expertise and trained a simple classifier (e.g. linear model) with the extracted features. It is a challenging task to extract features that are discriminative but insensitive to translation/illumination/orientation variation, even

though some elegant feature extractors were developed in the past (e.g. Shift Invariant Feature Transform (SIFT) [130] and Histogram of Oriented Gradients (HOG) [46]). The availability of large scale data and cost effective computational resource make it possible to learn such a feature extractor from the data. Moreover, given the task T and input data X , deep learning solves the problem in an end-to-end manner, that is, finding the proper function f such that $T \sim f(X)$. For example, in an image classification task, the input data is an 2d image, the task is to find a categorical label for the image, that is $f : \mathbb{R}^{N \times N} \rightarrow \mathbb{R}$. Earlier work (e.g. AlexNet [110], VGGNet [200], GoogleNet [207]) model the classification task with shallow network architecture (i.e. less than 20 layers), while the batch normalization [84] and skip connection [70] allows the possibility of convergence of very deep neural networks (e.g. ResNet [70]). Deeper networks usually result in better classification accuracy, at the cost of computational resource. Later work (e.g. MobileNets [79], EfficientNet [209]) aim to improve the efficiency of the model, making it possible to deploy the models for mobile vision applications.

Object Detection. Object detection task is one step further from the classification task, not only recognizing the presence of the target object, but also localizing the position of the object, that is, $f : \mathbb{R}^{N \times N} \rightarrow \{\mathbb{R}^5\}$, where $\mathbb{R}^5 = \{x, y, w, h, c\}$ denotes the position of the bounding box and the category. Unlike the classification task, in which only one label is needed, different number of objects may appear in the given image. However, for CNN models, it usually has a fix size of output. The common solution for this is to output a fix number of region proposals (i.e. \mathbb{R}^5) but use a threshold to suppress those less confident proposals.

From the perspective of the framework, object detection methods are divided into one-stage models and two-stage models. One stage models aim to design an end-to-end model, in which the model outputs region proposals and categorical labels simultaneously, resulting in fast detection speed. YOLO (You Look Only Once) [171] framed object detection problem as a regression problem, which extracted a feature map from the target image and made predictions in a grid-by-grid manner. SSD [126] made prediction with multiscale feature maps to improve recall ratio. YOLOv2 [169] introduced high resolution training images, dimension cluster, and convolutional with anchor box on top of the YOLO model. YOLOv3 [170] further optimized the standard YOLO architecture by building the feature extractor with more convolutional layers and making predictions over different scaled feature maps. YOLOv4 [18] combined new features (e.g. Weighted-Residual-Connections (WRC), Cross-Stage-Partial-connections (CSP), Cross mini-Batch Normalization (CmBN), Self-adversarial-training (SAT) and Mish-activation) to achieve better detection performance. Two stage models first generated a set of object-independent object proposals in the form of regions in the input image, then extracted fixed-length deep features using CNN from the processed regions for category prediction and bounding box regression. R-CNN (regions with CNN features) [62] applied selective search [220] to generate ~ 2000 category-independent region proposals and extracted features from each proposal for prediction. Fast R-CNN [61] mapped the region proposals from the image space to the feature space, which reduced the training time and testing time significantly. Faster R-CNN [172] further replaced the selective search algorithm with a trainable region proposal network to detect objects in real-time.

A few more things we need to think about when we design an object detection system. 1). Scale. The scale of the target object may vary significantly from image to image. SSD [126] made prediction on different scaled feature map. FPN [121] combined low-resolution, semantically strong features with high-resolution, semantically weak features via a top-down pathway and lateral connections. PANet [125] added a bottom-up pathway and lateral connections on top of the FPN. 2). Anchor box. Anchor boxes are a set of predefined bounding boxes and turn the direct prediction problem into the residual learning problem between the pre-assigned anchors and the ground truth bounding boxes because it is not trivial to directly predict an order-less set of arbitrary cardinals. For natural objects, anchor boxes usually result in better performance [172, 169, 170]. However, anchor boxes are usually computed by clustering the size and aspect ratio of the bounding boxes in the training set, which is time consuming and likely to fail if the morphology of the object varies dramatically (e.g. as the example shown in Chapter 4). To address this problem, anchor-free methods either learn custom anchors along with the training process of the detector [243] or reformulate the detection in a per-pixel prediction fashion [213]. 3). Imbalance training. The actual number of objects in the given image is usually far fewer than the number of region proposals, which causes an imbalance of the positive and negative samples during the training. To address this, a typical method is to apply hard example mining trick [22], which did a re-sample for the negative samples to match the number of positive samples. RetinaNet [122] proposed the focal loss to reshaping the standard cross entropy loss such that it down-weights the loss assigned to well-classified examples.

Scene Text Detection and Recognition. Scene text detection and recognition is a special case in the domain object detection. It is quite challenging, since the orientation/font/style/color of the text information could vary from image to image despite the variation of the target language.

Scene text detection aims at locating the text information in a given image. Early text detectors use box regression adapted from popular object detectors [172, 129]. Unlike natural objects, texts are usually presented in irregular shapes with various aspect ratios. Deep Matching Prior Network (DMPNet) [128] first detect text with quadrilateral sliding window and then refine the bounding box with a shared Monte-Carlo method. Rotation-sensitive Regression Detector (RDD) [119] extracts rotation-sensitive features with active rotating filters [250], followed by a regression branch which predicts offsets from a default rectangular box to a quadrilateral. More recently, character-level text detectors are proposed to first predict a semantic map of characters and then predict the association between these detected characters. Seglink [195] starts with character segment detection and then predicts links between adjacent segments. Character Region Awareness For Text detector (CRAFT) [9] predicts the region score for each character along with the affinity score between adjacent characters.

Scene text recognition aims at recognizing the text information in a given image patch. A general scene text recognition framework usually contains four stages, normalizing the text orientation [86], extracting features from the text image [70, 200], capturing the contextual information within a sequence of characters [196, 197], and estimating the output character sequence [37]. In Chapter 4, we apply a pre-trained scene text detection model [9] and a pre-trained scene text recognition model [8] to detect and recognize text

labels along the x axis, respectively. We focus on the x-axis labels because in spectroscopy data, the y-axis labels are often arbitrary, and only relative intensities are important. However, the general framework presented can be extended in the future to include y-axis labels.

Image Segmentation. In object detection, the goal is to use bounding boxes to locate each target object, while in image segmentation, it aims at pixel-wise recognition. Semantic segmentation aims to assign different semantic labels to each pixel in the given image while instance segmentation takes more step to group pixels into different instances. In this case, $f : \mathbb{R}^{N \times N} \rightarrow \{\mathbb{R}^{N \times N}\}$, where the output is a map of labels.

As an early work in semantic segmentation, Long et al. [129] adapted contemporary classification networks into fully convolutional networks and defined a skip architecture that combines semantic information from a deep, coarse layer with appearance information from a shallow, fine layer to produce accurate and detailed segmentations. Noh et al. [152] introduced deep deconvolution networks (i.e. deconvolution, unpooling, and rectified linear unit (ReLU) layers) for semantic segmentation on top of the convolutional layers adopted from VGG 16-layer net. Hong et al. [78] framed the segmentation task as a combination of classification and segmentation tasks and learned a separate network for each task, which reduces search space for segmentation effectively by exploiting class-specific activation maps obtained from bridging layers. A common challenge for feedforward networks in segmentation task is that, lower layers in convolutional nets capture rich spatial information, while upper layers encode object-level knowledge but are invariant to factors such as pose and appearance. To this end, Pinheiro et al. [162] proposed to augment feedforward nets with a top-down refinement approach, which efficiently leveraged features

at all layers of the net to generate high-fidelity object masks. In [32, 33], the advantages of spatial pyramid pooling module and encode-decoder structure were combined to encode multi-scale contextual information and gradually recover the spatial information from coarse to fine. More recently, visual transformer is introduced into the semantic segmentation field, [249] treated semantic segmentation as a sequence-to-sequence prediction task by deploying a pure transformer (i.e., without convolution and resolution reduction) to encode an image as a sequence of patches, while [204] relied on the output embeddings corresponding to image patches and obtain class labels from these embeddings with a point-wise linear decoder or a mask transformer decoder to model global context already at the first layer and throughout the network.

Instance segmentation algorithms can usually be divided into two categories: proposal-based methods and proposal-free methods. Proposal-based methods [71, 125] address the instance segmentation by first predicting object proposals (i.e. bounding boxes) for each individual instance and then assigning labels to each pixel inside the proposals (i.e. semantic segmentation). The success of the proposal-based methods relies on the morphology of the target object, and is likely to fail if the object is acentric or if there is significant overlap between different instances. Proposal-free methods [149, 49, 109, 150, 148] first take segmentation networks to assign different semantic labels to each pixel, then map pixels in the image to feature vectors in a high dimensional embedding space. In this embedding space, feature vectors corresponding to pixels belonging to the same object instance are forced to be similar to each other, and feature vectors corresponding to pixels belonging to different object instances are forced to be sufficiently dissimilar. However, it is difficult to find such an embedding space if the objects do not have rich features, such as the plot

lines in graph images. In Chapter 4, we customize our plot data extraction module by replacing the pixel embedding process with an optical flow based method, which groups data points into plot lines with continuity and smoothness constraints.

2.2. Deep Learning in Materials Science

An important task in the field of materials science is to understand the structure-property relationship, e.g. how to achieve materials-specific information through experimental or simulated data. In case of complex problems that involve massive combinatorial spaces or nonlinear processes, analytical and numerical solution in the previous paradigm (i.e. experiments and simulations) can tackle only at great computational cost or even cannot solve. The emerging technique, deep learning, opens up a new paradigm of data-driven science with the availability of the enormous scale of data, the development of advanced computer algorithm and the breakthroughs in affordable computing hardware. In this section, we explore how deep learning techniques help to progress and reduce the barriers in materials design, synthesis, characterization and modeling [21, 39, 3, 83, 58, 191, 85, 192, 190].

Materials Synthesis. Decades ago, human experts were usually required to specify conditionals and contextual rules to limit the number of choices during the synthesis process. The combination of extremely complex systems and huge numbers of potential solutions, which arise from competing objective functions (such as cost, purity, time and toxicity), make synthetic materials ill-suited to the application of traditional algorithmic approaches. Segler et al. [187] applied Monte Carlo tree search and symbolic artificial intelligence (AI) to discover retrosynthetic routes, achieving thirty times faster than the

traditional computer-aided search method. As an alternative deep learning approach that eliminates the rule-based expert system component, Liu et al. [123] treated the retrosynthetic reaction prediction task as a sequence-to-sequence mapping problem and proposed a seq2seq model, ending up with comparable performance. The exploration of chemical space for new reactivity, reactions and molecules is limited by the need for separate work-up-separation steps searching for molecules rather than reactivity. Dragone et al. [54] presented a system that can autonomously evaluate chemical reactivity within a network of 64 possible reaction combinations and aims for new reactivity, rather than a predefined set of targets. Starting from initial data on failed and successful experiments, the machine-learning approach directed future experiments and was shown to be capable of covering six times as much crystallization space as a human researcher in the same number of experiments.

Materials Characterization. The structure of molecules and materials is typically deduced by a combination of experimental methods, such as X-ray and neutron diffraction, magnetic and spin resonance, and vibrational spectroscopy. Deep learning is able to establish a structure-property relationship between atomic structure and their properties with high accuracy [60]. Ryan et al. [177] presented a neural-network model trained to compare topologies of atomic sites in the known crystal structures and applied the knowledge gained from such comparison to predict possible compositions of unknown compounds that might be pursued by a synthetic chemist. In [142], the feature extraction module pre-trained with natural images was re-used for SEM image recognition. In condensed-matter physics, Carrasquilla et al. [24] used neural networks to encode phases

of matter and discriminate phase transitions in correlated many-body systems, either in case of conventional ordered phases or unconventional phases.

Materials Modeling. With precise modeling, the properties of a molecule or material can be calculated for any chemical composition and atomic structure via atomistic simulations. However, in practice, it would likely be so complicated and could take unreasonable amounts of computer time as the size of the system increases without any proper approximations. For example, using the exact functional in Density functional theory (DFT) would give little practical advantage relative to the best wave function theories. There are notable limitations of the current approximations, which often require a more sophisticated many-body Hamiltonian. In [135], a combinatorially optimized, range-separated hybrid DFT function was proposed to reduce the computational cost. Moreover, Brockherde et al. [20] learned density models via examples to construct accurate density functions for realistic molecular systems, bypassing the necessity to solve the Kohn–Sham equations [108]. In [202], a transferable deep learning potential that is applicable to complex and diverse molecular systems is proposed to be extensible to larger molecules than those included in the training set.

2.3. Representation Learning

Data representation matters. In general, data representation could be in any format, as long as it conveys information from the data. The goal of representation learning is to find out a good representation of the data such that useful information is extracted to improve the performance over one or a few downstream tasks. But it is not entirely clear what makes a good representation. For example, in case of word embedding [6], which

aims to find the vector representation for each word. One-hot vector could be a valid representation for each word, but it cannot be used to measure the semantic similarity between different words, so it is not a good representation. A popular analysis by Bengio et al. [14] about the definition of a good representation is: "a good representation has the properties of local smoothness of input and representation, is temporally and spatially coherent in a sequence of observations, has multiple, hierarchically-organised explanatory factors which are shared across tasks, has simple dependencies among factors and is sparsely activated for a specific input." To some extent, finding a good data representation is the key to the success of machine learning algorithms.

Representation learning has been accompanied and nourished by a remarkable string of empirical successes both in academia and in industry, such as speech recognition [44, 45], object recognition [76, 40], natural language processing [41] and so on. Empirically, a few core principals of good representation have been developed: 1). *Distributed*, a reasonably-sized learned representation can capture a huge number of possible input configurations. 2). *Abstraction and invariance*, contain abstract concepts that are generally invariant to most irrelevant changes of the input according to the downstream task. 3). *Disentangled*, different explanatory factors of the data tend to change independently of each other in the input distribution.

Formally, the goal of representation learning is to find out a proper mapping function f , such that $f : \mathbb{R}^N \rightarrow \mathbb{R}^K$. Usually, we are looking for a low-dimensional feature space, that is, $K \ll N$.

Contrastive Learning. Unlike a discriminative model (e.g. classifier) that learns a mapping to some labels, in contrastive learning a representation is learned by comparing

among the input samples. The goal of contrastive learning is to force the representation of "similar" samples to be mapped close together while that of "dissimilar" samples should be further away in the embedding space. Here, the notions of similarity and dissimilarity could be different depending on the end goals, e.g., different views of the same objects are regarded as similar counterparts, while images of different objects are regarded as dissimilar.

A typical framework [34, 100, 99] for contrastive learning usually contains four major components: 1). *Data augmentation module*, generating positive data pairs (e.g. stochastic augmentation transformation) 2). *Base encoder*, mapping input data to vector representations 3). *Projection head*, mapping vector representation to metric space for loss computing 4). *Contrastive loss function*, defining the metric space for similarity measurement.

Contrastive learning techniques have been widely applied in different vision-related tasks in the past few decades. Chopra et al. [38] proposed a method for training a similarity metric from data in terms of recognition or verification applications where the number of categories is very large and not known during training. Later, Chechik et al. [28] presented a scalable learning scheme that learns a bilinear similarity measure over sparse representations with a large margin criterion and an efficient hinge loss cost. Hoffer et al. [77] proposed the triplet network model to learn useful representations by distance comparisons. Sermanet et al. [189] introduced a self-supervised representation learning method (TCN) based on multi-view video by anchoring a temporally contrastive signal against co-occurring frames from other viewpoints. Oord et al. [156] proposed a universal unsupervised learning approach to extract useful representations from high-dimensional

data, achieving strong performance on four distinct domains: speech, images, text and reinforcement learning in 3D environments.

Self-supervised Learning. Deep learning techniques have shown outstanding performance on various supervised learning tasks, however, it relies heavily on expensive manual labeling and suffers from generalization error, spurious correlations and adversarial attacks [127]. The emerging technique, termed self-supervised learning (SSL), opens up a new paradigm in deep learning from architecture engineering to data engineering. The features of SSL can be summarized as: 1). Obtain "labels" from the data itself via different augmentation process 2). Predict data or part of the data from its context.

In general, pretext tasks play a very important role in SSL, which can be divided into two categories: generative and contrastive. In generative SSL, the pretext task is to recover whole, or parts of its original input. In NLP, the objective of auto-regressive language modeling is usually maximizing the likelihood under the forward autoregressive factorization [235]. In computer vision, auto-regressive methods are used to model images pixel by pixel, e.g. PixelRNN [223], PixelCNN [221]. However, generative objective is usually formulated as maximum likelihood function, which is likely to get low-level abstraction and leads to conservative distribution. To this end, contrastive SSL is recently found far more competitive than generative SSL, especially in classification scenarios. In contrastive SSL, context-instance contrast aims to predict relative positions of two patches from a sample [53], or to recover positions of shuffled segments of an image (solve jigsaw) [102, 153], or to infer the rotation angle's degree of an image [59], while instance-instance contrast achieves superior performance via instance-based discrimination [212, 72, 34, 63].

Image Retrieval. A direct application of representation learning is image retrieval [48, 35], or content-based image retrieval (CBIR). Widespread of portable imaging devices and significant image acquisition speed in different fields (e.g. materials microscopy) enable tens of millions of images are captured everyday, making it important to develop a proper image retrieval system for image organization, i.e., easy to browser/search. To this end, compact yet rich feature representations are at the core of CBIR. Therefore, two questions have been driving research in this domain: 1). How to find good feature representations for images? 2). How to perform efficient image retrieval for large-scale datasets?

Before the rise of deep learning techniques, Scale-Invariant Feature Transform (SIFT [130]) has been a milestone hand-crafted feature descriptor and dominated in the field. Later, pre-trained CNN models (e.g. AlexNet [110], VGG [200], ResNet [70]) haven been introduced as the off-the-shelf models. Due to the model-transfer or domain-shift issues between tasks [248], the features extracted by models trained for classification task do not necessarily well suited for the image retrieval task. The availability of annotated datasets (e.g. Holidays [88], Oxford-5k [161], Paris-6k [160]) allows to fine-tune the off-the-shelf models towards better representations for images. A common way of supervised fine-tuning is the verification-based fine-tuning, which minimizes or maximizes the distance of pairs to validate and maintain their similarity. For example, Wang et al. [228] proposed a multiscale network with triplet sampling strategy to learn similarity metric directly from images. Besides, some unsupervised fine-tuning methods are also proposed in case there is not enough supervisory information. Tzelepi et al. [219] amplified the primary retrieval presumption that the relevant images to a certain query are meant to be closer to the

query in the feature space and retrained the pre-trained CNN model by minimizing the squared distance between each image representation and its nearest representations. SSL methods are also employed in unsupervised fine-tuning, e.g., employing an AutoEncoder [193, 64] to reconstruct its output as closely as possible to its input.

CHAPTER 3

Construction of Materials datasets from scientific literature**3.1. Introduction**

In a windfall of recent improvements in image resolution and acquisition speed, materials microscopy is experiencing an explosion of published imaging data. Journal articles have long been the standard for communicating important advances in scientific understanding. As sophisticated measurement and visualization tools render scientific communication more intricate and diverse, the visual presentation of scientific results as figures in these articles is noticeably more complex – especially within journals considered high-impact [80]. With this complexity, which is often a byproduct of the “compound” layout of the figures (i.e. figures containing multiple embedded images, graphs, and illustrations, etc.), the meaning of each individual image as a standalone entity is not always apparent. The result is that individual images are not only unsearchable, but the effort required to extract them into a machine-readable format is significant. This plays a major factor in the relative scarcity of general materials characterization images in the development and testing of new algorithms in deep learning, an emerging field characterized by the use of deep neural networks – hierarchical, multi-layered structures of processing elements – to learn representations of input data (often images) that reveal important characteristics about its content or overall appearance. The current surge of interest in DL stems from recent success in applications such as facial recognition [208], self-driving cars [29],

and complex game playing [199]. Much of this success is the byproduct of having large labeled datasets available for training [205], and in order for current materials imaging classification and recognition tasks [65, 50, 51, 4, 7, 144, 96, 97, 89] to really reap the benefits afforded by DL pipelines, having access to substantial labeled data is crucial. Fortunately, the incentive to publish is nearly ubiquitous across all scientific disciplines, and with a mechanism for automatic dataset construction that includes both separating out individual images from compound figures, as well as providing concise annotations describing key aspects or classification of the image content, much more of the scientific imaging data in literature could be utilized for training and developing DL tasks.

The effort to automate the construction and labeling of datasets from general web data has garnered broad attention from the computer vision, language technologies, and even chemistry/materials informatics communities [184, 116, 82, 236]. From the chemistry and materials informatics perspective, most of the focus has been on the development of text-mining tools adapted for “chemistry-aware” natural language processing (NLP), and have been used to create datasets of material properties and synthesis parameters from journal article text [206, 103, 104, 105, 42, 217]. For imaging datasets, standard computer vision approaches take the top images retrieved from a keyword query of an image search engine (i.e. Google, Flickr, etc.) and train classifiers to further populate datasets based on a keyword [236]. Unfortunately, this approach is problematic for scientific figures because of their compound layout. Current works that address this layout problem (what we refer to as “figure separation”) rely on hand-crafted rule based approaches [145, 147, 118, 117, 210], or adapt neural-networks to interpret figure separation as an object detection problem [218, 198]. While handcrafted techniques generally work well for sharp

image boundaries, and neural network approaches capture irregular image arrangements, neither of these methods are designed such that explicit references to the caption text are considered as part of the separation. This is problematic for both figure separation and labeling because when the two are not consistent with each other (i.e. more/fewer images than subfigure labels) the intended description for an image is not always clear. In these instances, (NLP) with extra-grammatical constructs [82], can be extended to properly summarize pertinent scientific results consistent with the subdivision of the image. Despite the individual successes of a few previous studies to advance figure separation, and labeling related to automatic dataset creation [145, 147, 5], tools that do this in a seamless end-to-end (query-to-dataset) fashion, capable of extracting and labeling images from journal articles based on the specific user requirements, are currently lacking.

In this work, we present a tool for the automatic EXtraction, Separation, and Caption-based natural Language Annotation of Images (EXSCLAIM!) for scientific figures and demonstrate its effectiveness in constructing a self-labeled electron microscopy dataset of nanostructure images. The EXSCLAIM! tool is developed around materials microscopy images, but the approach is applicable to other scientific domains that produce high-volumes of publications with image-based data as well as graphs and illustrations.

3.2. Design Overview

The main goal of the EXSCLAIM! toolkit is to provide researchers with a collection of modules that, when executed as a sequential pipeline (1) enable comprehensive keyword searches for scientific figures within open-source journal articles, and (2) facilitate the extraction and pairing of images from within figures, to the appropriate descriptive

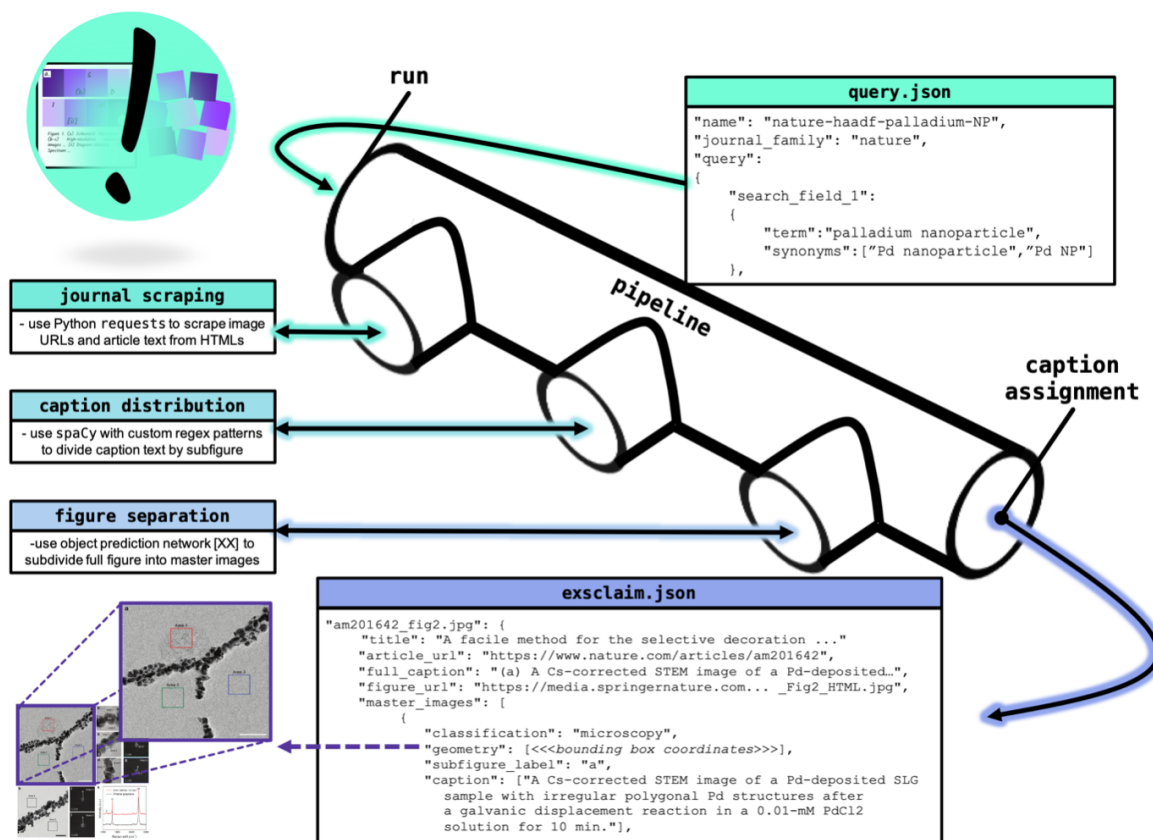


Figure 3.1. An Overview of the EXSCLAIM! Pipeline. The schematic highlights the path from a user-defined query to the exsclaim output data structure (both JSON objects). After the query is submitted to the pipeline by calling the run method, the journal scraper extracts figure/caption pairs from articles in the specified journal family, the caption distributor divides the caption text into segments that are consistent with the images in the figure, and the figure separator computes bounding boxes that separate and classify the individual images from the full figure. The caption assignment method is effectively the “self-labeling”, as it takes all of the caption segments, reduces them to individual keywords if possible, and pairs them with their corresponding image within the full figure. The final output data structure, i.e. the exsclaim output file, contains all the descriptions and references necessary to construct the full labeled imaging dataset.

keywords and phrases from caption text. To begin, users populate an input file (JSON object) with parameters that define the search (i.e. keywords, synonyms, how to order search results, journal family to search, etc.), as well as the number of total articles to consider in the process. This input file, referred to as the query, is the basic initialization structure that is run in the EXSCLAIM! pipeline. A run of the EXSCLAIM! pipeline involves sequential execution of the journal scraper, caption distributor, and figure separator modules before a final call to the caption assignment module which is responsible for the final pairing of the distributed caption text to the extracted figures. Fig. 3.1 provides an overview of the pipeline, highlighting the structure of both the query and final exclaim output file (another JSON object, summarized for better readability), as well as the role that the extraction modules serve in transforming the data. In Fig. 3.1, the query for palladium nanoparticles is run through the pipeline, and each extraction module incrementally populates the exclaim output file, which provides a final record of all the relevant figures with bounding box and label information for the associated images. This exclaim output file provides all the necessary file identifiers and pointers needed to construct a high-quality self-labeled imaging dataset based on the details of the overall search as defined by the query.

Journal Scraper. This module performs the first extraction step in the pipeline. It is responsible for retrieving figure/caption pairs from articles that fit the parameters of the search defined in the query, and uses the python requests library to handle all of the HTTP requests. First, article URLs are extracted from a collection of individual searches formed from all possible combinations of search field keywords and associated synonyms. With an ordered list of open-source article URLs, the scraper contains a method that

sends further GET requests to retrieve each article and its figure/caption pairs. Finally, the exsclain file is populated with all the general article information returned during the extraction steps, including the full caption data for each figure, and URLs to the articles and their associated figures – providing basic provenance to the data workflow. (Full article text can also be included if desired.) The journal scraper has the functionality to parse both open-source Nature family journals as well as journals that are part of the American Chemical Society (ACS) umbrella. All of the figure and caption extraction is performed directly from the HTML version of the article and does not require PDF downloads.

Caption Distributor. With all of the figures and caption information recorded, the next step is to distribute subsequences of the full caption text to the respective subfigure that they reference. This first involves sentence tokenization with Parts-of-Speech (POS) tagging. POS tagging deconstructs strings of sentence text into small units (tokens), which are given a tag that describes their part-of-speech in context of the sentence. For this, the robust natural language processing (NLP) tokenization tools from the spaCy library are extended to properly assign “subfigure identifier” tags, to patterns that indicate the presence of a subfigure description. With this custom tagging, the “(a)” in a phrase such as “(a) Nanoparticles deposited on . . .”, is properly interpreted as the subfigure identifier (denoted by ‘CAP’) instead of a determiner surrounded by parenthesis. From there, a regular expression style of pattern matching is performed on the list of the custom POS tags with a dictionary of reference sequences collectively representing a “standard syntax” for typical subfigure image descriptions in caption text.

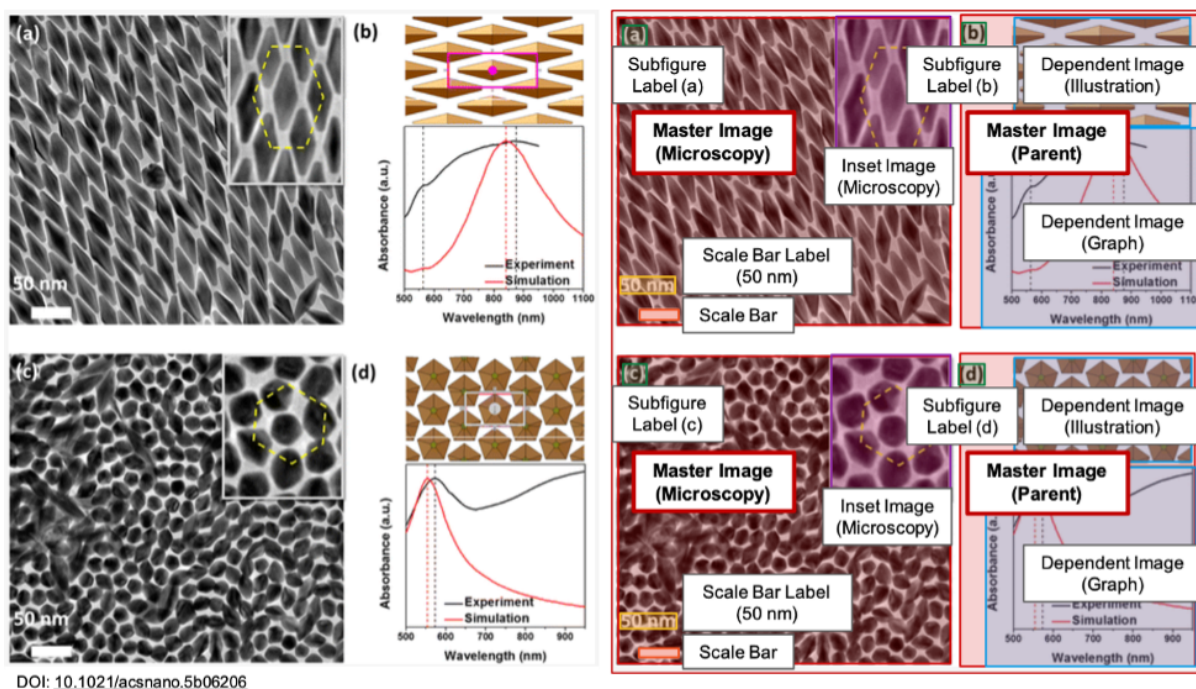


Figure 3.2. Master-Dependent-Inset (MDI) Model. The panel on the left shows the initial figure before annotation, and the panel on the right is a copy that has been properly annotated according to the Master-Dependent-Inset (MDI) model. The presence of a subfigure label is necessary for all images given a “master image” designation. Notice the master image corresponding to subfigure label “b” contains two dependent images (illustration and graph). Since the master image corresponding to subfigure label “b” governs more than one distinct image (i.e. the caption text associated with “b” will refer to all the images in the series), this master image is classified as a parent. Additional image features such as the scale bar and scale bar label (when present) are also identified.

Figure Separator. The final component of the EXSCLAIM! pipeline involves separating the extracted figures into “master images” according to what we establish as the Master-Dependent-Inset (MDI) modeling paradigm. In the MDI model, the master image is defined relative to a subfigure label (“(a)”, “(b)”, etc.). The subfigure label is the functional element bridging the visual image content to the text describing it. All

visual components (i.e. all individual images, drawing, clarifying annotations) belonging to the complete entity referenced by the subfigure label, regardless of the shape it forms when all entities are considered together, are collectively referred to as the “master image”. In addition to defining the master images in context of the full figure, the figure separator module both classifies the image according to the nature of the image content, i.e. microscopy, diffraction, graph, illustration image etc., and also extracts any scaling information that is present in the form of a scale bar on images within the figure. Fig. 3.2 provides a detailed view of the MDI model applied to a standard figure. Both insets and scaling info are shown in subfigure “(a)”, and subfigure “(b)” is useful for demonstrating the need for a “master image” classification, as it is clear that the subfigure label is referencing more than one distinct image. Moreover, when a master image contains multiple dependent images, it is classified as a parent. The detection and classification of master images is the primary task of the figure separator module, which follows a two-stage framework outlined in Sec. 3.3. In the first stage is subfigure label detection, where the goal is to detect all of the subfigure all subfigure labels presented in the compound figure. This is achieved using a combination of object localization (YOLOv3-style object detector [170])) and object recognition (ResNet-152 [70]) neural networks.. The second stage starts with encoding the global subfigure label layout information, which is extracted from the detected subfigure labels, into a binary mask. Then this binary mask, which provides visual anchors for the positions of the subfigure labels, is concatenated with the standard RGB input channels and fed into the master image detection module (YOLOv3-style object detector [170]). Taken together, these neural networks function to locate master images within the figure while preserving the association between master

images and their respective subfigure labels, and further classify each master image based on the appearance of its content.

Scale detection. Once the master images are detected, localized, and classified, the final step is to extract any scaling information that exists. Scale detection also proceeds through two networks. First an object detection neural network (Faster R-CNN [172]) is used to detect the bounding boxes of any scale bar labels and scale bar lines in a given figure. Next, the detected scale bar labels are fed into a Convolutional Recurrent Neural Network (CRNN) [115, 114] in order to perform text recognition, making the scale bar label text machine readable. The result of the CRNN is processed by a rule-based search to ensure the output is a valid scale bar label (i.e. a number followed by a unit). Multiple scale bar lines and scale bar labels in a single figure are matched by assigning the scale bar labels to scale bar lines greedily based on the distance between the center of their respective bounding boxes. Each matched scale bar pair is assigned to the subfigure in which it is contained. Using the length in pixels of the scale bar line and the subfigure and the scale bar label text, the real space size of the subfigure can be determined.

Crowdsourcing Figure Annotation with Mechanical Turk. Ultimately, DL models are needed for detecting the locations of the master images and classifying these images by type. In order to achieve sufficient accuracy, these models must be trained on images that are deemed as proper references, or have been verified as representing the correct way to locate and classify master images (i.e. ground truth). Since the demands of this task are unique in that figure separation does not fall explicitly within the canon of standard computer vision training tasks, we needed an approach to scale the annotation effort to ensure the best accuracy on the figure separation task. For this, we used the

amazon mturk
worker

Annotate images from... (HIT Details) Auto-accept next HIT Time Elapsed 5:32 of 20 Min

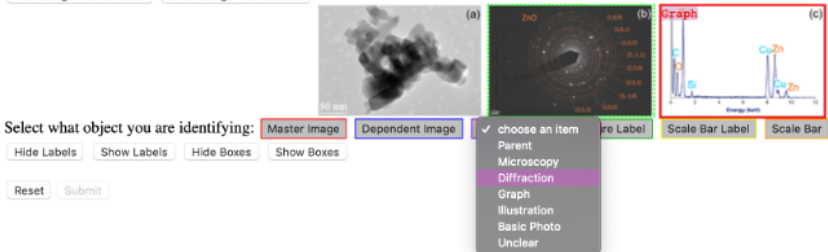
[Return](#)

Draw Bounding Boxes around all objects of interest in the image and then either label them or transcribe their text

- Primary Task: Draw bounding boxes around objects of interest (Master Image, Dependent Image, Inset Image, Subfigure Label, Scale Bar Label, Scale Bar Line)
- Secondary Task #1: Classify images as one of: Parent, Microscopy, Diffraction, Graph, Illustration, Basic Photo, Unclear
- Secondary Task #2: Enter in the text for Scale Bar Labels and Subfigure Labels **EXACTLY** as it appears except you may enter μm as um and \AA as A. (this means include spaces and parenthesis!)
- **Interacting with the interface:** To begin drawing a box, double click. The location of your double click will be one corner of the box. When you have moved your cursor to the other corner of the box, single click. To move a previously drawn box, press down on its border and move it to where you would like it to go and then release the mouse.
- **IMPORTANT: It is better to put too much in a bounding box than too little. If most of your boxes exclude parts of the object, we will reject you.**
- If the image is of terrible quality, draw one scalebar label box anywhere and enter text 'SKIP'
- **Most subfigures are master images.** A dependent image is typically when one subfigure label refers to multiple subfigures. The entire image will only be a master image if there is only one subfigure or if there are no labels.
- Full instructions with sample labeling session [here](#) Example labeled images available [here](#)

I have read [the user agreement](#) and I understand that if I do not follow the instructions, my assignment will be rejected.

[Hide Longer Instructions](#) [Show Longer Instructions](#)



Select what object you are identifying: Master Image Dependent Image choose an item Subfigure Label Scale Bar Label Scale Bar

Parent Microscopy Diffraction Graph Illustration Basic Photo Unclear

Hide Labels Show Labels Hide Boxes Show Boxes

Figure 3.3. Mechanical Turk GUI for Figure Separation Annotation. A snapshot of the graphical user interface (GUI) that is accessible via the MTurk worker platform (<https://www.mturk.com/worker>) is included below. MTurk workers with an extensive record of positive work on the platform (both in number of tasks completed and overall accuracy when it comes to task acceptance) have the proper credentials to access and complete the “HITS” (Human Intelligence Tasks) when they’re posted to the site. Workers can select the various categories at the bottom that are used to describe the objects in the figure, and after drawing a bounding box around the object, they are asked to either further classify the image (i.e. choosing diffraction as the image type in the example below), or transcribe the text in the case of subfigure labels and scale bar labels. All workers must follow the directions closely and are required to acknowledge that they understand the task before submission. When one of the members of the EXSCLAIM! project reviews the HIT submitted by the worker, if directions are not followed, or it is clear that the task was completed in a rushed or careless fashion, the HIT is rejected. This helps ensure the quality of the training dataset created for this task of figure separation.

crowdsourcing platform from Amazon called Mechanical Turk (MTurk). Though proper interpretation of a scientific image often requires an expert to understand the nuances of the image content, identifying where the master images are located, as well as their proper classification, can be formulated so that those without a rigorous science background can annotate the images with only a very modest amount of instruction. As such, we designed a custom figure annotation GUI (snapshots of the GUI are included in the Fig. 3.3) within the MTurk platform, and asked workers to draw bounding boxes around each master image in the dataset, and then classify them. This allowed us to quickly create a dataset of > 3000 MDI annotated figures ($\sim 18,000$ separate images). The basis for training the current version of the figure separator involves augmentation of a random sampling of 2000 of the annotated images from MTurk and is described in more detail in Sec. 3.3.

3.3. Technical Details

Scientific literature contains large volumes of complex, unstructured figures that are compound in nature (i.e. composed of multiple images, graphs, and drawings). Separation of these compound figures is critical for information retrieval from these figures. Most previous work on compound figure separation (CFS) decomposes the figures into the smallest possible subfigures [239, 113, 210, 218, 145], which misses an important opportunity to incorporate association between subfigures and caption information. It is important to note that captions provide essential description to help users understand the content of the figures in scientific articles. Frequently, authors include separate descriptions for each subcaption in the compound figure caption. Thus, a method for parsing compound figures

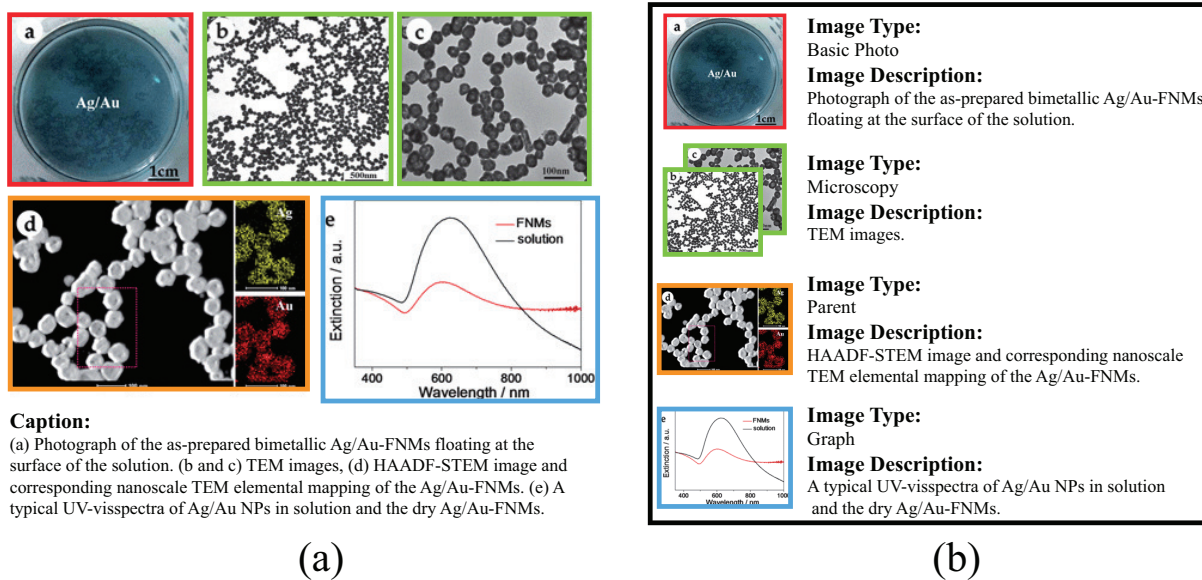


Figure 3.4. An example of compound figure separation following authorial intent: (a) shows the original compound figure. (b) shows the results from the proposed compound figure separation method, in which each subfigure is associated with the descriptive information from the caption and its image type. Image source: (Wu et al. Nanoscale, 2012)

into subfigures in such a way that is consistent with authorial intent has the promise to improve separation performance.

By convention, authors typically place indices (subfigure labels, i.e. "a", "b", "c", etc.) in front of the subfigures and their respective caption components¹, as shown in Fig. 3.4(a). In this paper, we propose a new compound figure separation strategy that decomposes compound figures under the guidance of subfigure labels. In Fig. 3.4(b), we provide an example where the compound figure is decomposed into several subfigures while each subfigure is assigned a subfigure label. The decomposition is accomplished using a two-stage framework, as shown in Fig. 3.5. In the first stage, subfigure labels

¹Our method also handles the case where sub-figure labels are placed above/below subfigures

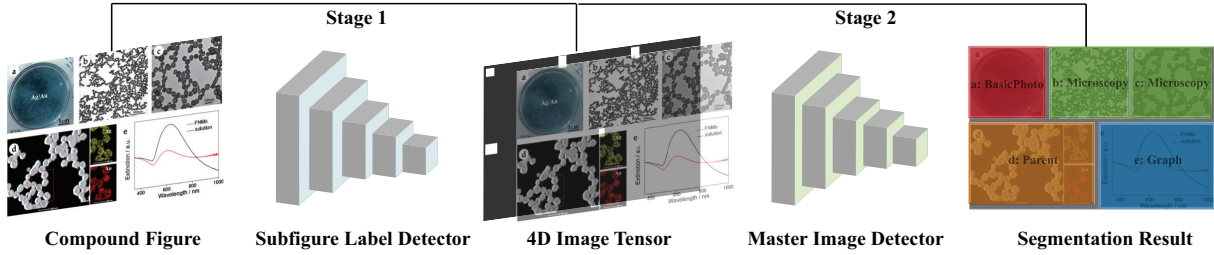


Figure 3.5. Overview of the proposed method. The compound figure is fed into the subfigure label detector to locate subfigure labels. The location of the subfigure labels is concatenated with the compound figure to generate a 4D image tensor, which is then fed into the master image detector to get the segmentation result.

are detected with the subfigure label detection module. Then in the second stage, the compound figure, along with layout information extracted from detected subfigure labels, are fed into the subfigure detection module for subfigure detection.

Before we describe in detail the proposed framework, it is important to note that there are two existing challenges in addressing the compound figure separation problem. One is the imbalanced distribution of different subfigure labels (i.e. data imbalance problem). As shown in Fig. 3.6, we count the number of different subfigure labels in about 1200 compound figures. Obviously, some subfigure labels (e.g. "g", "h") occur far less frequently than other subfigure labels (e.g. "a", "b"), which could easily lead to a subfigure label detector with a huge bias. The other challenge is the huge in-class variation of subfigures. Target subfigures can sometimes be a combination of several images (e.g. the subfigure labeled by "d" in Fig. 3.8(a)). Previous methods, such as the CFS algorithm (e.g. [218] shown in Fig. 3.8(d)), will likely fail to make a correct detection because intrinsic features of the subfigure are not enough to lead to a confident prediction.

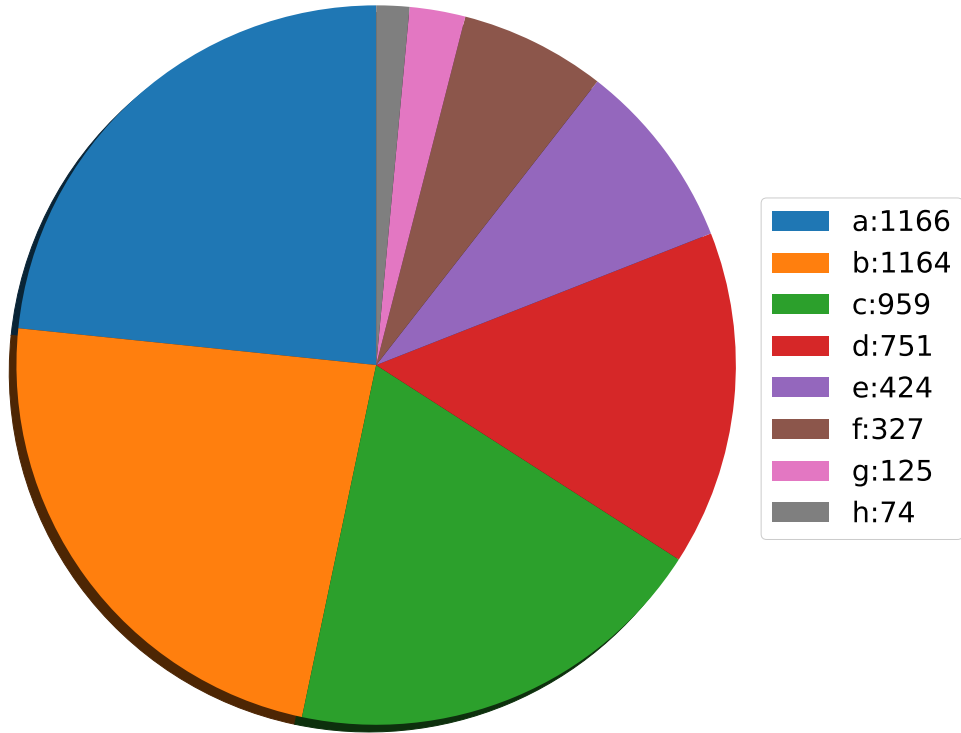


Figure 3.6. Distribution of different subfigure labels.

Subfigure label detection module. Unlike a conventional object detection framework, which trains the object localization module and recognition module simultaneously, we decouple these two modules to address the data imbalance problem. The subfigure label detection module is formulated as:

$$\begin{aligned}
 \{F_{i,j}^l\} &= \mathcal{F}^l(I_c) \\
 B_{i,j}^l, O_{i,j}^l &= \mathcal{G}^l(F_{i,j}^l) \\
 C_{i,j}^l &= \mathcal{H}^l(I_c, B_{i,j}^l)1[O_{i,j}^l > \epsilon]
 \end{aligned}
 \tag{3.1}$$

where \mathcal{F}^l is the feature extraction function, which encodes the input compound figure I_c into a feature map, composing of a set of feature vectors $F_{i,j}^l$ (i, j denote the coordinate

of the feature map, l denotes the subfigure label detection module). \mathcal{G}^l is the bounding box prediction function, which predicts the bounding box $B_{i,j}^l$ and confidence score $O_{i,j}^l$ for each feature vector. $B_{i,j}^l$ is represented by the xy-coordinates of the center of the bounding box and the width and height of the bounding box (i.e. $B_{i,j}^l = \{x, y, w, h\}$). $1[\cdot]$ is the indicator function. \mathcal{H}^l is the classification function, which classifies the patches cropped from the compound figure into different subfigure label categories $C_{i,j}^l$. During inference, low confidence bounding boxes are culled before being fed into the classification function.

Then the localization module (i.e. $\mathcal{F}^l, \mathcal{G}^l$) and classification module (i.e. \mathcal{H}^l) are optimized separately.

$$\begin{aligned} \mathcal{F}^l, \mathcal{G}^l &= \arg \min_{\mathcal{F}^l, \mathcal{G}^l} \sum_{i,j} \mathcal{L}_1(B_{i,j}^l) + \lambda \mathcal{L}_2(O_{i,j}^l) \\ \mathcal{H}^l &= \arg \min_{\mathcal{H}^l} \sum_{i,j} \mathcal{L}_3(C_{i,j}^l) \end{aligned} \tag{3.2}$$

where \mathcal{L}_1 is the regression loss measuring the difference between the predicted bounding box and the ground truth. We use anchor-based loss[169, 170] in this paper. \mathcal{L}_2 is the binary cross-entropy loss predicting the confidence scores and \mathcal{L}_3 is the cross-entropy loss which is widely used for classification problems. λ is a hyperparameter.

Subfigure detection module. A key component in our framework is the subfigure detection module, which incorporates layout information from the detected subfigure labels to improve the performance of detecting subfigures within a compound figure. A

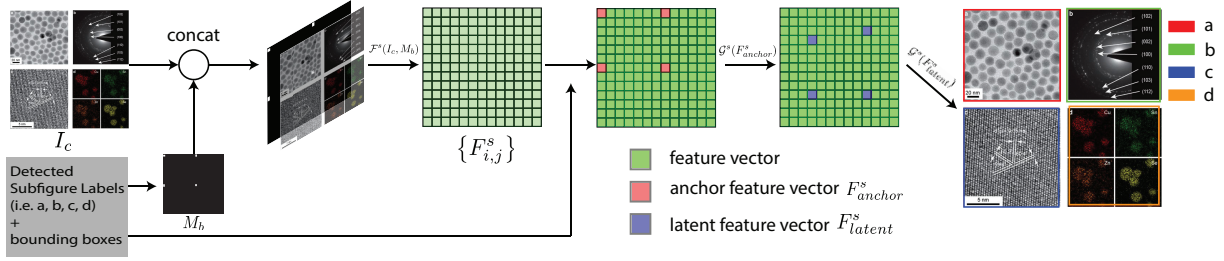


Figure 3.7. A step-by-step illustration of the subfigure detection module. The binary mask containing the global layout information from the detected subfigure labels is concatenated with the compound figure to provide extra features. Then the location of subfigure labels is used to select anchor feature vectors. Latent feature vectors are then computed from the the anchor feature vectors for a more accurate prediction.

step-by-step illustration of the subfigure detection module is provided in Fig. 3.7. In particular, the layout information of the detected subfigure labels is encoded into a binary mask.

$$M_b(u, v) = \begin{cases} 1, & (u, v) \in B^l \\ 0, & \text{otherwise} \end{cases} \quad (3.3)$$

where $B^l = \{B_{i,j}^l | C_{i,j}^l > 0\}$ is the set of bounding boxes of detected subfigure labels.

Then the binary mask is used as an extra feature, which is fed into the feature extraction function \mathcal{F}^s along with the compound figure (s denotes the subfigure detection module).

$$\{F_{i,j}^s\} = \mathcal{F}^s(I_c, M_b) \quad (3.4)$$

Unlike conventional object detection frameworks, in which feature vectors with high confidence scores are selected, we use the location of the detected subfigure labels to select proper feature vectors (i.e. anchor feature vector F_{anchor}^s). By doing so, the association

between subfigure labels and their respective subfigures are preserved.

$$F_{\text{anchor}}^s = \{F_{x,y}^s | (x, y) \in B_{i,j}^l, C_{i,j}^l > 0\} \quad (3.5)$$

According to the anchor-based bounding box regression loss[169, 170], the feature vectors located at the center of the target object are more likely to make a more accurate estimation of bounding boxes. However, subfigure labels usually appear at the corner of their respective subfigures, thus the selected anchor feature vectors usually appear far from the center of the target subfigure. As shown in Fig. 3.7, we introduce the latent feature vectors (i.e. F_{latent}^s) for a more accurate bounding box prediction, which are computed from the anchor feature vectors but closer to the center of the target subfigures (see ablation study in Sec. 3.3.2). Then bounding boxes of subfigures are predicted from these latent feature vectors.

$$\begin{aligned} B_a^s, O_a^s &= \mathcal{G}^s(F_{\text{anchor}}^s) \\ F_{\text{latent}}^s &= \{F_{x,y}^s | (x, y) \in B_a^s\} \\ B^s, O^s &= \mathcal{G}^s(F_{\text{latent}}^s) \end{aligned} \quad (3.6)$$

where B^s, O^s denote the bounding box and confidence score of each detected subfigure. B_a^s, O_a^s are auxiliary variables.

$$\mathcal{F}^s, \mathcal{G}^s = \arg \min_{\mathcal{F}^s, \mathcal{G}^s} \sum_{(x,y) \in B_a^s} \mathcal{L}_4(x, y) + \mathcal{L}_1(B^s) + \lambda \mathcal{L}_2(O^s) \quad (3.7)$$

As shown in Eq. 3.7, the optimization process is based on three different loss functions, the regression loss (i.e. \mathcal{L}_1) of bounding boxes, the binary cross-entropy loss (i.e. \mathcal{L}_2) of

Method	$AP_{0.5}$	$AP_{0.75}$	$AP_{0.5:0.95}$
Mukaddem [145]	24%	20%	19%
Tsutusi [218]	84%	76%	63%
Proposed	90%	82%	72%

Table 3.1. Comparison of the performance of different compound figure separation methods.

confidence scores, and mean square error loss (i.e. \mathcal{L}_4) measuring distance between the location of the latent feature vectors and the centers of subfigures.

3.3.1. Experiments

Approximately 2000 compound figures were crawled from the Royal Chemistry Society (RCS), Springer Nature, and American Chemical Society (ACS) journal families, and were uploaded to Amazon Mechanical Turk (MTurk), a platform for crowdsourced data labeling. Then the labeled figures were randomly divided into the training set (~ 800 figures) and the testing set (~ 1200 figures)².

For the feature extraction function ($\mathcal{F}^l, \mathcal{F}^s$), we use Darknet-53[170] to encode the input images into a stack of different scaled feature maps, followed by a feature pyramid layer[121]. We use 1x1 convolutional layer as the bounding box prediction function ($\mathcal{G}^l, \mathcal{G}^s$). We use the ResNet-152[70] as the classification function ((\mathcal{H}^l))³.

During the training process, we use the Adam[106] optimizer while decaying the learning rate every 10000 iterations. It is worth noting that the classification function is trained with a mixture of real MTurk-labeled images and synthetic images. Each synthetic image

²The code of this paper is released as part of the Github project: <https://github.com/MaterialEyes/exsclaim>

³A validation dataset is released on https://petreldata.net/mdf/detail/exclaim_validation_v1.1/

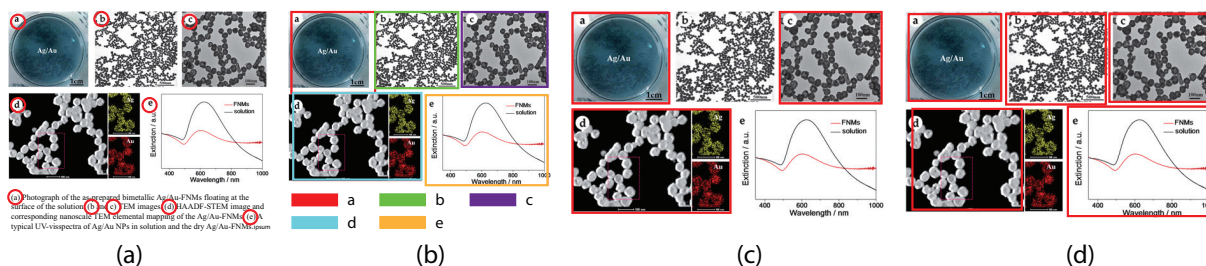


Figure 3.8. Comparison of different compound figure separation algorithms. (a). An example compound figure with its respective caption. Subfigure labels (the ones in red circles) appear in both the compound figure and the caption, bridging the connection between them. (b). The separation result by the proposed framework, which detects subfigures along with their respective subfigure labels. (c). The separation result by [145]. (d). The separation result by [218]. Image source: (Wu et al. Nanoscale, 2012)

is generated by cropping a random background patch from a random compound figure and pasting a random letter (e.g. "a", "b") onto it.

For the testing set containing 1164 images and 4982 subfigure labels, the detector detects 4777 true positive labels and 24 false positive labels, resulting in a precision equal to 99.5% and a recall equal to 95.9%.

As shown in Table. 3.1, the Average Precision (AP) is used to measure the performance of different compound figure separation. Here the notation AP_{σ} denotes the value of AP when setting the Intersect-over-Union (IoU) as σ . Mukaddem et al. [145] uses hand-crafted features for subfigure detection, an approach which works only for subfigure with sharp boundary and rich content (as shown in Fig. 3.8(c)). Tsutusi et al. [218] is a CNN-based method, which performs better than [145] but also fails to detect "compound subfigure" (as shown in Fig. 3.8(d)). As expected, with the help of subfigure labels, the proposed algorithm outperforms the other two methods, improving the mean Average Precision (mAP) metric by over 9% compared to existing approaches.

Methods	a	b	g	h	Avg.
Baseline	83%	80%	34%	40%	60%
Proposed	88%	89%	86%	81%	84%

Table 3.2. Comparison of the performance ($AP_{0.5}$) of training subfigure label detector with/without module decoupling.

3.3.2. Ablation study

We conduct two experiments here for ablation study.

One is to validate how the decoupling process addresses the data imbalance problem. As the baseline, we train the subfigure label detector directly with the training set. As shown in Table 3.2, we compare the $AP_{0.5}$ of the baseline and the proposed on the most frequent and least frequent subfigure labels. Without the decoupling process (Table 2), the performance drops significantly ($\sim 40\%$) for the least frequent subfigure labels. As expected, the decoupling process produces a detector with less bias, improving average performance by 24%.

In another experiment, we validate how the latent feature vectors improve the detection precision. As a baseline, we predict the bounding boxes directly from the anchor feature vectors, which results in $AP_{0.5} = 90\%$, $AP_{0.75} = 65\%$, $AP_{0.5:0.95} = 60\%$. As expected, introducing the latent feature vectors significantly improves the detection precision (e.g. 17%) with a large IoU threshold (e.g. 0.75).

3.3.3. Conclusion

The proposed two-stage framework, which decomposes the compound figure according to the authorial intent, is able to automatically label each separated subfigure by the information from its corresponding caption, with existing natural language processing tools

(described in more detail in [186]). The proposed approach is also useful for information retrieval tasks. More results and interesting findings can be found in Appendix A.

3.4. Results

The extraction modules, outlined individually in the Section 3.2, incrementally transform the query into a final exsclaim output structure, which contains all of the information necessary to create a dataset annotated from caption descriptions in the literature. There are several components of this pipeline that must be considered in evaluating overall performance. Here, we (1) validate classification accuracy for the figure separator tool using precision and recall metrics obtained on a reference data set, (2) examine the various scenarios for how caption text is assigned, quantifying accuracy for the case where a single keyword is used to describe the image, and finally (3) provide suggestions for how to create new labels or general topics to associate with images that are left un- or under-annotated. In total, the results emphasize the attention placed on accuracy and extensibility of the EXSCLAIM! pipeline design.

3.4.1. Validation of the figure separator tool on MTurk Dataset

In order to validate the classification and bounding box prediction accuracy of the figure separator tool, 784 figures (3555 separated images) from the crowdsourced MTurk dataset – withheld during training – were used as part of the validation set. The results are shown in Fig. 3.9. Images were scraped from both Nature and ACS publishing sources (the code for the scrapers is easily extendable to other journal sources), and positive predictive value

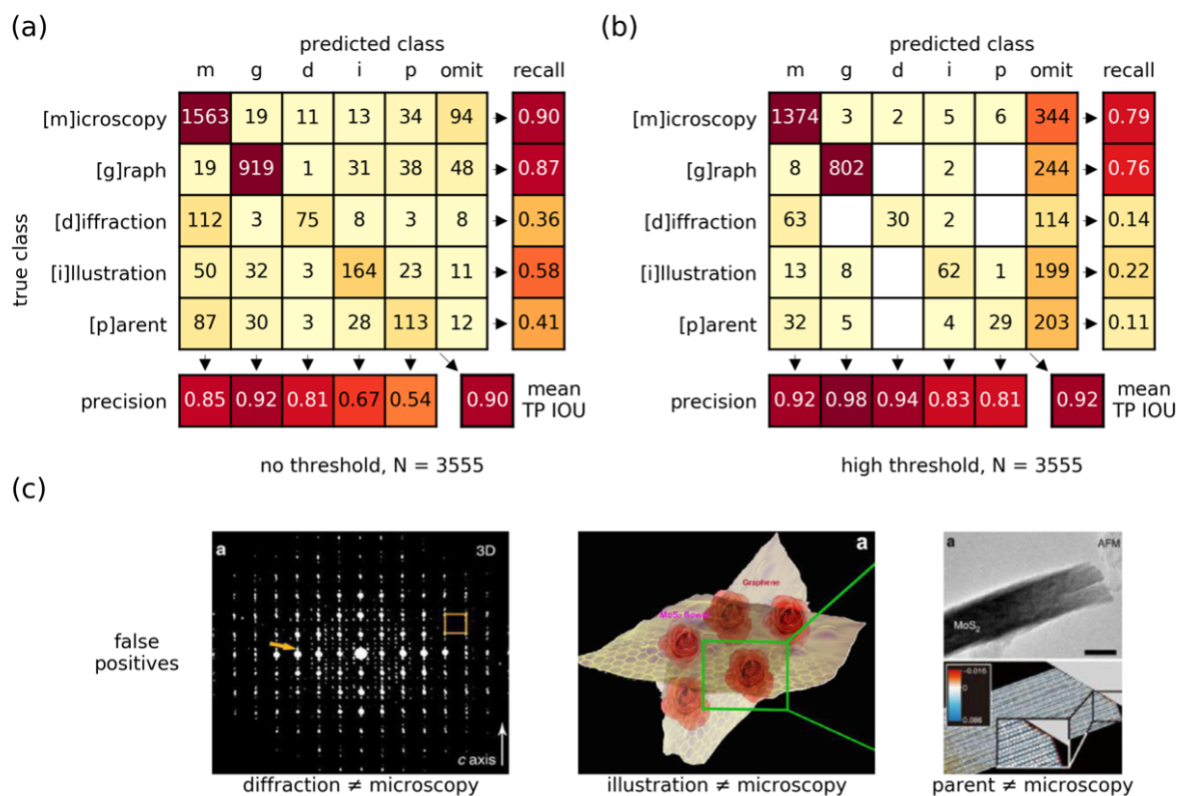


Figure 3.9. Precision and Recall for Image Classification. The confusion matrices highlight the nature of the mistakes made in each classification scenario (i.e. looking at a column and row for the same image type, the intersection represents the true positive number while the associated false positives are the column entries and the associated false negatives are rows) at two confidence thresholds (a) no threshold, and (b) high threshold for N=3555 images. In both cases, the precision scores are adequate, particularly in the case of the microscopy, graph, and diffraction images. Recall suffers across the board as correctness is emphasized over completeness in the design of the pipeline. Images in (c) highlight some of the more easily rationalized examples of false positive microscopy classifications. — DOIs for articles containing the example images (left to right): 10.1038/ncomms14925, 10.1038/srep08722, 10.1038/ncomms4631.

(precision), which is loosely the correctness of the positive classification, is always prioritized over a measure of completeness, such as recall. Confusion matrices summarizing

important aspects of the classification performance are given in both a “no threshold” (Fig. 3.9(a)) and “high threshold” (Fig. 3.9(b)) condition. In the “no threshold” condition, the most likely classification in the final output of the neural network is accepted, regardless of its magnitude. In the “high threshold” condition, only classifications with values of magnitude greater than or equal to 0.99 are accepted. In this context, the threshold is a proxy for classification confidence in the figure separator. Both microscopy and graph classification with “no threshold” and “high threshold” specification are favorable from a precision perspective with all scores in excess of 0.80 – and furthermore, the “high-threshold” precision for graphs is 0.98.

One of the primary use cases for the EXSCLAIM! toolkit is the construction of self-labeled datasets, and in the assumption of data abundance (i.e. the opportunity cost of passing up on an image is low), the low recall values from false negatives (particularly diffraction images identified as microscopy images) are not as detrimental to the integrity of the set as images that are incorrectly identified as an image type they are not (i.e. false positives). Fig. 3.9(c) highlights some of the interesting false positive trends where a sizable population of diffraction, illustration, and parent classes are being incorrectly assigned as microscopy images. In the case of the diffraction example (left), the diffraction patterns show periodicity reminiscent of atomic-resolution microscopy images, so the microscopy assignment seems logical. The false positive for an illustration (middle) maintains some features commonly associated with a microscopy image, such as a darker background, and even the coloring of the graphene sheet resembles a texture present in microscopy images, but globally is clearly not a microscopy image. Finally, the parent on the far right (Fig. 3.9(c)) actually does contain a microscopy image, but since the image

below it does not contain a subfigure label and is “semantically” tied to the microscopy image in “a”, the most appropriate classification for this image is “parent”.

3.4.2. Sample Query: Electron Microscopy Images of Nanostructures

We illustrate the utility of EXSCLAIM! for labeling materials imaging datasets with an example of electron microscopy images of nanostructures. Open source Nature articles were collected from a “Sort By Relevance” search related to the collection of queries formed from the following lists of keywords: (“electron microscope”, “electron microscopy”), and (“nanoparticle”, “nanosheet”, “nanoflake”, “nanorod”, “nanotube”, “nanoplate”, “nanocrystal”, “nanowire”, “nanosphere”, “nanocapsule”, “nanofiber”). This specific query mimics a wildcard-style search for nanostructures imaged in an electron microscopy modality and returned a total of 13,450 open-source articles with 83,504 figure-caption pairs. For the purpose of quantifying overall retrieval performance on microscopy images, which involves both an assessment of image classification and keyword labeling accuracy, we restrict the following quantitative measurements to articles in the top 10% of the relevancy ranked list, which is a reasonable simplification because the labels defined by the query (“nanoparticle”, “nanowire”, etc.) depends on the presence of the keyword in the caption, and the median keyword frequency decays exponentially across article rank. This collection of articles has a yield of 29,096 separate images.

Graphs and microscopy images are among the most popular image types for this specific query. The high prevalence of microscopy images is expected, as a result of including microscopy-relevant keywords as a separate word family, whereas the high frequency of graphs is most likely a result of how authors choose to format scientific results. Fig.

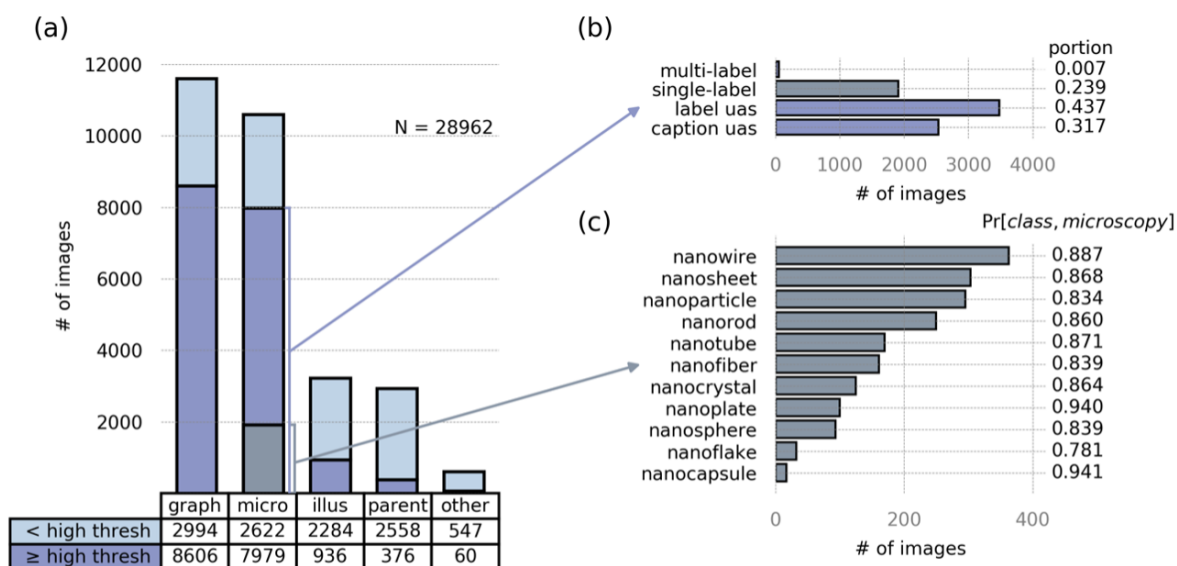


Figure 3.10. Distribution of Image Types and Keyword Labeling Accuracy. The query in this example is used to extract electron microscopy images of general nanostructures from Nature journals. The bar plot in (a) shows the distribution of image types extracted at two different thresholds. The bar plots in (b) and (c) further subdivide the population of high confidence microscopy images. In (b), the distribution of label types are recorded. In this context, single and multi-label refer the existence of a keyword label. Label unassigned (uas) means that caption text has been distributed to the image, but no keyword label from the query exists. Caption unassigned (uas) refers to a scenario where the caption distributor was not able to confidently distribute a proper substring to caption text. The bar plot in (c) represents the distribution of labels in the top 10% of retrieved microscopy images and provides a measurement of the joint probability that an image classified as microscopy and given the corresponding subsequent label is a true microscopy image represented by the given label.

3.10(a) highlights the distribution of predicted image types in the top 10% of retrieved articles, and indicates the threshold used to assign the classification with further color coded divisions of each bar. These thresholds (“no threshold” and “high threshold”) again act as proxy for classification confidence and are consistent with the definitions

given in the evaluation of the figure separator on the MTurk dataset. Within both graphs and microscopy images, 75% of the total population receives a high-level of confidence associated with the prediction of its image type (i.e. high threshold condition), which is likely a consequence of having a more precisely defined image type. This distribution of predicted image types in Fig. 3.10(a) is useful for quantifying the approximate number or percentage of high-confidence extractions (image classification) that one could expect to obtain for a given query. Moreover, taken with the results on the MTurk images from Fig. 3.9, it is likely that a large fraction of the high-confidence classifications, particularly in the case of microscopy images and graphs, are actual instances of microscopy images or graphs because of the high precision scores.

To give classification confidence some further meaning in the context of the construction of self-labeled microscopy image dataset, it is important to start examining both the frequency and quality of the processed caption text during caption assignment, because it is this text that is used to ultimately describe the image content (i.e. it is the difference between constructing a generic dataset of microscopy images vs. a highly specific dataset of atomic resolution microscopy images of Ag nanoparticles). First, we examine the frequency at which the processed text is assigned to images and look further at the nature of the text extracted (i.e. is it a single keyword? multiple keywords? sentence or phrase?) Fig. 3.10(b) identifies four categories of possible image labeling. The most common case, “label uas” (label unassigned) represents a population of the separated images that have received a portion of the caption text as a description, but do not contain any of the keywords from the query. The “caption uas” (caption unassigned) category, contains separated images that have not received a portion of the caption text at all.

These captions that should be assigned typically have sentence structure complexities, such as multiple compound subjects, or long intervening phrases, etc., that make caption distribution somewhat ambiguous. And while the percentage of “caption uas” images is somewhat high, this is to some degree intentional, as the data abundance assumption and “no information is better than bad information” design principles underscore many of the extraction steps in the pipeline for the purposes of keeping the data clean. Also, the current regular expression approach taken in caption distribution provides users with the ability to extend or fine tune the system to better suit individual use cases. This is one of the advantages of using a rule-based approach opposed to learning-based approach: the logic behind caption distribution is transparent, readable, and easily modifiable. The final conditions, identified in Fig. 3.10(b), are the single and multi-label conditions, which represent images containing assigned caption text with explicit reference to one or more of the query keywords.

The single-label condition is further broken down in Fig. 3.10(c), and its relation to the initial full set of extracted microscopy images is emphasized with the gray coloring in Fig. 3.10(a) (24% of the images in the high-threshold group have a single keyword label). The bar chart shows the distribution of images assigned to each keyword, as well as a measure of the joint probability of both the predicted microscopy classification and keyword as being correctly identified. The average joint probability of the positive microscopy/keyword prediction computed across all classes is 87%, which means that of the 1909 single-label microscopy images, 1660 are true microscopy images with appropriate keyword labels. Across the entirety of the extracted images (beyond the results for the top 10% recorded here), there are approximately 4300 microscopy images with a single keyword explicitly

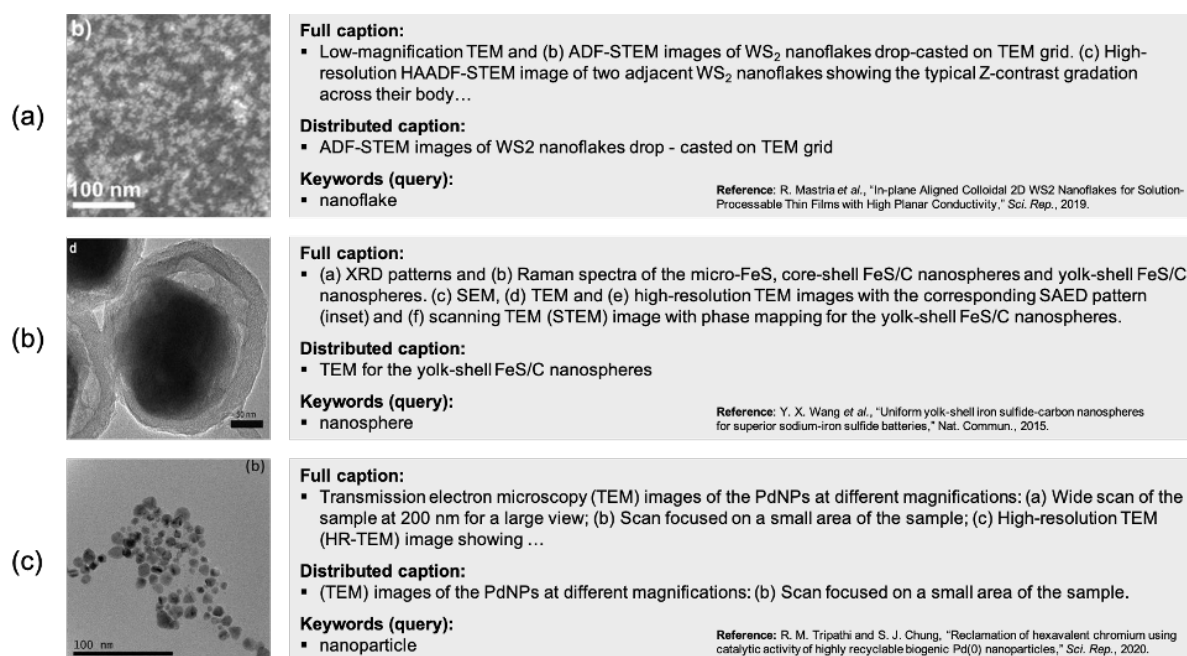


Figure 3.11. Examples of Images Extracted and Labeled with EXSCLAIM! The extracted images contain both a keyword label from a word family associated with the query, as well as grammatically sufficient sequence of distributed caption text. Some caption distribution examples are simple, as in (a) where all the words distributed are linearly connected, however, the current caption distributor extraction class is also designed to adequately capture more complex structural dependence relationships (b-c) where the subject is separated from the text completing the full consistent description.

related to the query. With the 87% joint probability of the positive microscopy/keyword prediction, we would expect a full dataset for this query to contain approximately 3725 high-confidence microscopy images with the appropriate keyword label.

Current caption distribution prioritizes identification of appropriate keywords over language correctness, so only keyword accuracy is factored into the ground truth comparisons in Fig. 3.10. There are, however, many examples where both the keywords are successfully extracted, and the distributed caption is grammatically sufficient, and this

even extends to scenarios where the caption text assignment contains non-contiguous segments. For example, in Fig. 3.11(a), the distributed caption uses all the text from the subfigure identifier “(b)” to the period signifying the end of the sentence, however Figs. 3.11(b-c) show that the current method is capable of parsing the sentence at a higher structural level and pick out the structurally closest sequence of text, even when it is not linearly the closest. For example, in the Fig. 3.11(c), the subject of the sentence “TEM images” is not matched to the linearly closest descriptions (adjectives) or even closest preposition, but in this case to the preposition at the end of the sentence, which structurally makes sense and is the proper division of this caption text.

3.4.3. Extraction of Scaling Information from Images

Magnification of objects is fundamental to microscopic imaging. With suitable estimation of object magnification, achieved by recognizing and interpreting the scale bar length and accompanying text (scale bar-scale bar label) in the separated subfigures images, researchers can begin quantifying image content based on size. In addition, searches through resulting databases can be refined to a particular size range, giving users greater control over results. Further modeling can combine scale and caption information to associate keyword terms with dimensions. To quantify the accuracy of the scale detection step, 440 figures containing 920 scale bar-scale bar label pairs from the MTurk dataset that were withheld during training were used. The predicted pixel length of the scale bar line differed from the ground truth value by a mean absolute error of 5.4%. This level of error is similar to that of humans performing the task and negligible when determining the approximate scale of the image. Scale bar label recognition overall is 92% accurate

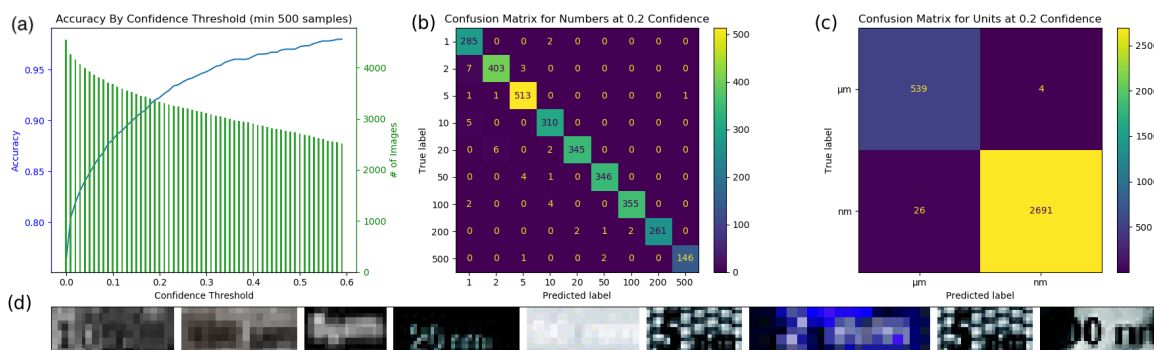


Figure 3.12. Accuracy of scale bar label detection. The plot in (a) shows how accuracy varies as a function of confidence threshold, including the number of images present at a given threshold. For all thresholds shown, there are at least 500 samples. When the confidence level associated with the scale bar label detection is 0.6, the overall accuracy (percentage of scale bar labels that are exactly right – both the number the unit) is ~ 0.95 . The confusion matrices in (b) and (c) highlight the accuracy of the predicted number and unit components for the scale bar label recognition. Larger scales (i.e. mm and cm) were not adequately represented in the training set, so they are not part of the test set. Both number and scale recognition accuracy is high at the 0.2 threshold ($\sim 92\%$ and $\sim 99\%$ respectively for the labels shown). The examples in (d) show common instances of the low-resolution and low-contrast conditions that are responsible for a majority of the prediction errors.

(both number and unit) (as shown in Fig. 3.12(a-c)) when the confidence threshold is 0.2. This is a reasonable accuracy given the variance in the quality of the scale bar text itself. It is not uncommon for authors to leave the default scale bar text untouched when a microscopy image is included as part of a compound figure – this is problematic because the native text size is often too small for high quality visualization, and there are many instances where the color of the text is similar to the image content it sits on top of (i.e., recognition suffers from very low contrast). Refer to some of the examples included in the Fig. 3.12(d) which support these general observations.

3.4.4. Hierarchical Label Assignment

We have demonstrated the effectiveness of the EXSCLAIM! tool in situations where keywords are extracted from the caption text, and we even show that in some situations when the structure of the caption is complex, caption assignment is still capable of extracting sentences and phrases in a way that preserves the intended meaning. However, currently, keywords are defined explicitly by the query, so any other related topics that are prominent in the returned set, but not explicitly specified within the query, are ignored. The advantage of treating the problem of dataset construction in this manner is that it ensures a high degree of relevance in the images that are assigned a keyword label, but it significantly limits the scope of the self-labeling task in general (i.e. Fig. 3.10(b). reveals that 43% of the images in the nanostructure example contained distributed caption text but were without explicit keyword labels). To this end, we explore how modern natural language processing (NLP) tools can be leveraged to transform the phrases or sentences of the distributed caption text into a series of relevant, hierarchical labels for each image they are referencing. More details of the outlined approach can refer to [186].

The task of resolving the sentence or phrases into a series of relevant image labels is handled using Word2Vec [139] and Latent Dirichlet Analysis (LDA) [17] topic modeling at multiple structural levels. First, the explicit text of the caption must be processed to extract the most important nouns and adjectives that are indicative of the specific content of the given image. Fig. 3.13 provides examples of how an image/caption pair is transformed into a series of hierarchical labels that describe and provide context for the image content. In all examples, the “caption” labels are determined using an iterative word dropout approach that removes words furthest away (measured by cosine similarity)








	extracted images	distributed captions	assigned labels
(a)		(POM) images of silica rods of average length 6.5 μm and diameter 0.75 μm (designated as micro-rods) dispersed in a planar cell without crossed polarisers.	caption: 'silica' 'rods' 'planar' 'crossed' 'micro' abstract: 'cell' 'elongated' 'cylindrical' topic: 'optics and plasmonics'
(b)		HRTEM image of Fe/FeO NCs. Inset: EDS-mapping of Fe/FeO NCs.	caption: 'hrtem' 'eds' 'ncs' 'fe' 'mapping' abstract: 'nanotherapeutics' 'icg' 'nanocarrier' topic: 'cancer treatment'
(c)		TEM and magnified (inset) images of RuIrZnOx-U nanoboxes.	caption: 'tem' 'nanoboxes' abstract: 'electrode' 'photocatalyst' 'air' 'synthetic' topic: 'catalysis materials'
(d)		The TEM bright field image is showing twinning blocky austenite grains after compression test at $\sim 10^{-1}$ s (a).	caption: 'austenite' 'grains' 'bright' 'tem' 'compression' abstract: 'twin' 'microstructure' topic: 'microstructure mechanics'
(e)		The SEM images of AgNW thin film before electron beam irradiation	caption: 'beam' 'irradiation' 'agnw' 'film' 'sem' abstract: 'substrate' 'nanowire' topic: 'flexible sensors'
(f)		TEM images of perovskite NCs.	caption: 'perovskite' 'ncs' 'tem' abstract: 'mapbbr' 'photovoltaics' topic: 'solar energy'
(g)		The relationship between the initial structures of Si under a capacity restriction of 1500 mAh g ⁻¹ . TEM images of aggregated lump (e).	caption: 'si' 'tem' 'capacity' 'structures' 'aggregated' abstract: 'anodes' topic: 'battery materials'

Figure 3.13. Example Image Labels after Hierarchical Label Assignment. (a-g) Examples of hierarchical label assignment for images containing a properly distributed caption. For each image, the caption labels are limited to caption text only. Conversely, the abstract labels are free to draw additional relevant words from the abstract text, and the topic labels come from the human-assigned topic names from the LDA topic summaries. — DOIs for articles containing the example images (natural reading order): 10.1038/s41598-019-40198-1, 10.1038/s41467-019-12142-4, 10.1038/s41467-019-12885-0, 10.1038/s41598-019-55803-6, 10.1038/srep17716, 10.1038/am2016167, 10.1038/srep42734.

from the center of the current group. The “abstract” labels are the 2-3 words closest to the center of the combination of abstract and caption words, and typically provide context for the image that is not found explicitly in the caption. Finally, the LDA model trained on the abstracts and introduction texts, assigns the best “topic” label to the document containing the image, if a confidence threshold of 0.80 is exceeded. Fig. 3.13 overall

provides several compelling examples of how this approach can provide useful contextual labels for the images from language outside that found explicitly in the distributed caption. For example, we learn things that we can confirm visually, such as the fact that nanorods are elongated and cylindrical (Fig. 3.13(a)). We also learn things that an expert might know that are useful for understanding the function of the image content, such as the facts that: Fe/FeO nanocrystals function as nanotherapeutics (Fig. 3.13(b)); RuIrZnOx-U nanoboxes are synthetic as opposed to biological catalysts (Fig. 3.13(c)); austenite grains describe the “microstructure” of the image (Fig. 3.13(d)); nanocrystals are MA lead halide perovskite (numerical characters get stripped in the text preprocessing, so ‘mapbbr’ refers to for MAPbBr₃) (Fig. 3.13(f)), and even in Fig. 3.13(e) where the abstract context is more redundant than unique or complimentary, the topic provides useful context for where the specific image content appears from an application perspective.

There are a few subtle issues with some of the assigned labels that are a result of some of the known shortcomings in Word2Vec model training. Most notably is that in some cases, similarity is more indicative of how interchangeable/related words are, as opposed to measuring their actual likeness. For example, the nanoboxes in Figure 8c are inaccurately labeled as photocatalysts and should be described as electrocatalysts. While these words are highly related and often found in interchangeable contexts (i.e. [photocatalysts/electrocatalysts] facilitate water-splitting ... etc.), they can present certain instances where the suggested labeling is problematic. Overall, the combination of Word2Vec modeling with LDA topic discovery provides a solid backbone for self-labeling imaging effort. Future work will involve finding ways to use language components and the imaging analysis jointly to describe image content

3.5. Conclusion

We present EXSCLAIM!, a software pipeline for the automatic EXtraction, Separation, and Caption-based natural Language Annotation of IMages from scientific figures, detailing the specific extraction tools and providing quantitative measures of performance for image type classification and keyword labeling accuracy on a crowdsourced-labeled dataset, and an extracted dataset of nanostructure figures from Nature family journals, respectively. In addition, we provided discussions and useful model implementations aimed at assigning image labels from complete sentence text. All tools in the main extraction pipeline are available on github (<https://github.com/MaterialEyes/exsclaim>). Successful consolidation of images and self-labeling of images from scientific literature sources will not only enhance the navigation and searchability of images spanning materials, medical, and biological domains, but is a vital first step towards introducing scientific imaging to the canon of training datasets for state-of-the art deep learning and computer vision algorithms.

CHAPTER 4

Spectra Data Extraction from Spectrographs

4.1. Introduction

Spectroscopy, primarily in the electromagnetic spectrum, is a fundamental exploratory tool in the fields of physics, chemistry, and astronomy, allowing the composition, physical structure and electronic structure of matter to be investigated at the atomic, molecular and macro scale, and over astronomical distances. In materials science, in particular, X-ray absorption near edge structure (XANES) and Raman spectroscopy play a very important role in analyzing the characteristics of materials at the atomic level. For the purpose of understanding the insights behind these measurements, data points are usually displayed in graphical form within scientific journal articles. However, it is not standard for materials researchers to release raw data along with their publications. As a result, other researchers have to use interactive plot data extraction tools to extract data points from the graph image, which makes it difficult for large scale data acquisition and analysis. In particular, high-quality experimental spectroscopy data is critical for the development of machine learning (ML) models, and the difficulty involved in extracting such data from the scientific literature hinders efforts in ML of materials properties. It is therefore highly desirable to develop a tool for the digitization of spectroscopy graphical plots. We use as prototypical examples XANES and Raman spectroscopy graphs, which often have a series

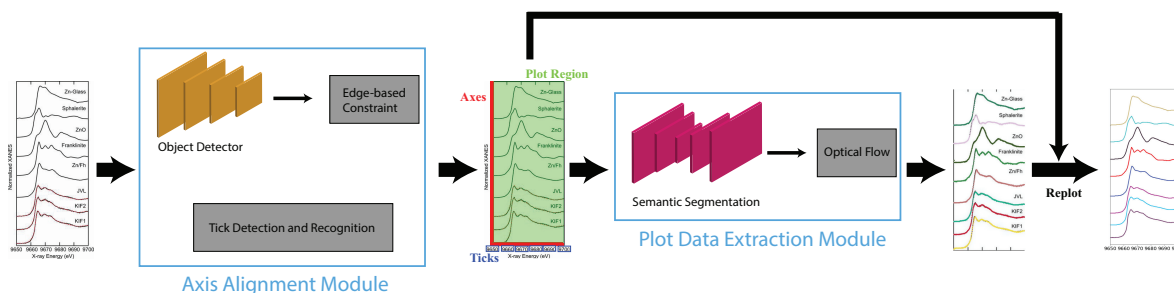


Figure 4.1. An overview of the proposed Plot2Spectra pipeline. An example of XANES graph images is first fed into the axis alignment module, which outputs the position of axes, the values of the ticks along x axis and the plot region. Then the plot region is fed into the plot data extraction module which detects plot lines. Figure is from Ref. [173].

of difficult-to-separate line plots within the same image. However, the approach and tool can be applied on other types of graph images.

Earlier work [98, 132, 131] on extracting plot lines from the graph images focus on dealing with plot lines with pivot points, which are likely to fail if the assumption does not hold. WebPlotDigitizer [175] is one of the most popular plot data extraction tools to date. However, the burden of having to manually align axes, input tick values, pick out the color of the target plot, and draw the region where the plot falls in is cumbersome and not conducive to automation.

In this chapter, we develop Plot2Spectra, which transforms plot lines from the graph image into sets of coordinates in an automatic fashion. As shown in Fig. 4.1, there are two stages in the plot digitizer. The first stage involves an axis alignment module. We first adopt an anchor-free object detection model to detect plot regions, and then refine the detected bounding boxes with the edge-based constraint to force the left/bottom edges to align with axes. Then we apply the scene text detection and recognition algorithms to

recognize the ticks along the x axis. The second stage is the plot data extraction module. We first employ semantic segmentation to separate pixels belonging to plot lines from the background, and from there, incorporate optical flow constraints to the plot line pixels to assign them to the appropriate line (data instance) they encode.

The contribution of this paper is summarized as follows.

- (1) To the best of our knowledge, we are the first to develop the plot digitizer which extracts spectra data from the graph image in a fully automatic fashion.
- (2) We suppress axis misalignment by introducing an edge-based constraint to refine the bounding boxes detected by the conventional CNN-based object detection model.
- (3) We propose an optical flow based method, by analyzing the statistical proprieties of plot lines, to address the problem of plot line detection (i.e. assign foreground pixels to the appropriate plot lines).

4.2. Method

In this section, we provide more details about the proposed Plot2Spectra tool. The general pipeline of the proposed method is shown in Fig. 4.1. The pipeline is made of two modules. The first module is the axis alignment module, which takes the graph image as the input and outputs the position of axes, the value and position of ticks along x axis as well as a suggested plot region. The second module is the plot data extraction module, which takes the plot region as the input and outputs each detected plot line as a set of (x, y) coordinates. With the detected plot lines, ticks, and axes, we are able to perform any

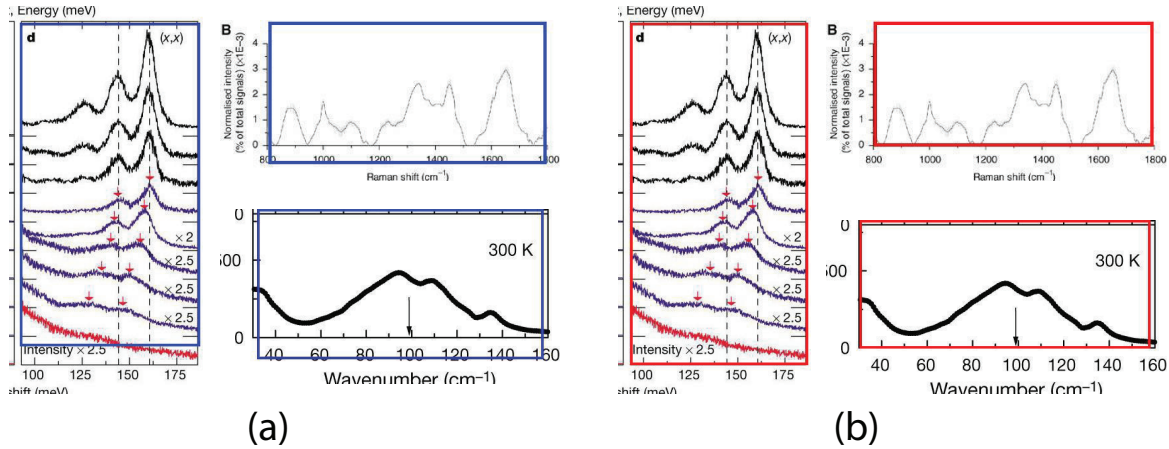


Figure 4.2. Examples of axis misalignment. (a) There are noticeable gaps between the left/bottom edges of the bounding boxes (blue boxes) and the axes. (b) The left/bottom edges of the bounding boxes (red boxes) are perfectly aligned with the axes. Figures are from Refs. [179, 211, 201] (left to right, top to down).

subsequent plot analysis (e.g. re-plot data into a new graph image, compute similarities, perform ML tasks, etc.).

4.2.1. Axis alignment

In the axis alignment module, we first adopt an anchor-free object detector [213] to detect plot regions and refine the predicted bounding boxes with the edge-based loss. We then apply the pre-trained scene text detector [9] and the pre-trained scene text recognizer [8] to extract and interpret all tick labels below the x-axis.

Given the graph image $I \in \mathbb{R}^{H \times W \times 3}$, let $F \in \mathbb{R}^{H_F \times W_F \times C}$ be the feature map computed by the backbone CNN. Assume the ground truth bounding boxes for the graph image are defined as $\{B_i\}$, where $B_i = (x_0^i, y_0^i, x_1^i, y_1^i) \in \mathbb{R}^4$. Here (x_0^i, y_0^i) and (x_1^i, y_1^i) denote the coordinates of the left-top and right-bottom corners of the bounding box, respectively.

For each location (x, y) on the feature map F , it can be mapped back to the graph image as $(\frac{W}{W_F}(x + \frac{1}{2}), \frac{H}{H_F}(y + \frac{1}{2}))$ (i.e. the center of the receptive field of the location (x, y)). For the feature vector at each location (x, y) , a 4D vector $t_{x,y} = (l, t, r, b)$ and a class label $c_{x,y}$ are predicted. The ground truth class label is denoted as $c_{x,y}^* = \{0, 1\}$ (i.e. 0, 1 denote the labels for background and foreground pixels, respectively) and the ground truth regression targets for each location is denoted as $t_{x,y}^* = \{l^*, t^*, r^*, b^* \mid l^* = x - x_0^i, t^* = y - y_0^i, r^* = x_1^i - x, b^* = y_1^i - y\}$. Then the loss function for the detector comprises a classification loss and a bounding box regression loss

$$\mathcal{L}^{det} = \frac{1}{N_{pos}} \sum_{x,y} \mathcal{L}_{cls}(c_{x,y}, c_{x,y}^*) + \mathbf{1}_{\{c_{x,y}^* > 0\}} \mathcal{L}_{reg}(t_{x,y}, t_{x,y}^*) \quad (4.1)$$

where \mathcal{L}_{cls} denotes the focal loss in [122] and \mathcal{L}_{reg} denotes the IoU (Intersection over Union) loss in [237]. N_{pos} denotes the number of locations that fall into any ground truth box. $\mathbf{1}_{\{\cdot\}}$ is the indicator function, being 1 if the condition is satisfied and 0 otherwise.

However, the left and bottom edges of the predicted bounding boxes by the detector may not align with the axes, as shown in Fig. 4.2(a). Therefore, we introduce an edge-based constraint to force the left/bottom edges of the detected bounding boxes to align with axes inspired by the observation that the values of pixels along axes usually stay constant.

$$\mathcal{L}^{edge}(x, y) = \sum_{u=x-l}^{x+r} I(u, y+b) + \sum_{v=y-t}^{y+b} I(x-l, v), \quad s.t. (l, t, r, b) \in t_{x,y} \quad (4.2)$$

The first term forces the left edge to have constant values and the second term forces the bottom edge to have constant values. Then, the axis alignment module is optimized with

both the detection loss and the edge-based loss:

$$\mathcal{L}^{AL} = \mathcal{L}^{det} + \mathcal{L}^{edge} \quad (4.3)$$

However, the edge-based loss term is not differentiable, which means the Eq. 4.3 cannot be optimized directly. In practice, we take the one-step Majorization-Minimization strategy to solve the problem.

$$\arg \min_{l,b} \{ \mathcal{L}^{edge} | \arg \min_{t_{x,y}} \mathcal{L}^{det} \} \quad (4.4)$$

As shown in Eq. 4.4, we first optimize the detection model with the gradient descent method to generate bounding boxes with high confidence scores, then we refine the left/bottom edges of the detected bounding boxes via a Nearest Neighbor search. In particular, we apply the probabilistic Hough transform [107] to detect lines (i.e. axes candidates) in the graph image and then search for the most confident candidates. Intuitively, the best candidates should be either horizontal or vertical, long enough and close to the edges of the detected bounding box.

$$L^* = \arg \min_{L_i \in \mathcal{H}(I)} D_{dist}(L_i, E) \quad (4.5)$$

$$s.t. \quad \|D_{angle}(L_i, E)\|_2^2 > \epsilon_1, \quad D_{length}(L_i, E) > \epsilon_2$$

where \mathcal{H} denotes the probabilistic Hough transform operator, $E \in \{E^{left}, E^{bottom}\}$ denotes the left or bottom edge of the bounding box. D_{angle} measures the cosine similarity between the given two lines. D_{length} measures the ratio between the length of the detected line and

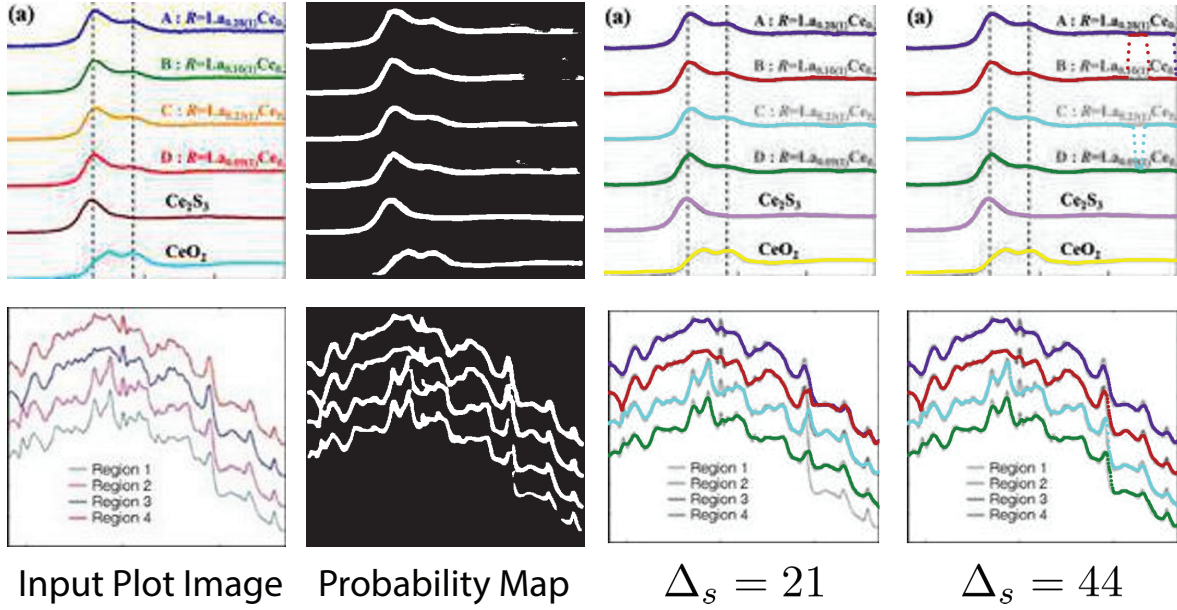


Figure 4.3. Results of plot line detection with different Δ_s . Too large or too small values fail to have a correct detection. Figures are from Refs. [57, 194] (top to down).

the edge, and D_{dist} measures the horizontal/vertical distance between the two parallel lines. Empirically, ϵ_1 and ϵ_2 are set to be 0.98 and 0.5, respectively.

4.2.2. Plot data extraction

In the plot data extraction module, we first perform semantic segmentation to separate pixels belonging to plot lines from the background, and from there, apply optical flow constraints to the plot line pixels to assign them to the appropriate line (data instance) they encode.

$$\mathcal{L}^{seg} = -C \log(\tilde{C}) - (1 - C)(1 - \log(\tilde{C})) \quad (4.6)$$

where $\tilde{C} = S(I^p) \in \mathbb{R}^{H^p \times W^p}$ denotes the probability map, which is computed by the semantic segmentation model S from the given plot image I^p . $C = \{c_i\} \in \mathbb{R}^{H^p \times W^p}$ denotes the ground truth semantic map, c_i is 1 if it is a foreground pixel and otherwise 0.

Pixel embedding in conventional instance segmentation is likely to fail because the plot lines often do not have a sufficient number of distinguishable features between different instances. As shown in Fig. 4.4, all pixels belonging to a single line (data instance) should be assigned to the same color. One common mode of failure in conventional instance segmentation models (i.e. LaneNet [149], SpatialEmbedding [148]) involves assigning multiple colors to pixels within a single line. Another common failure mode is a result of misclassifications of the pixels during segmentation (see second row in Fig. 4.4, LaneNet misclassifies background pixels as foreground and SpatialEmbedding misclassifies foreground pixels as background).

Intuitively, plot lines are made of a set of pixels of the same value and have some statistical properties, such as smoothness and continuity. Here, we formulate the plot line detection problem as tracking the trace of a single point moving towards the y axis direction as the value of x increases. In particular, we introduce an optical flow based method to solve this tracking problem.

$$I^p(x, y) = I^p(x + dx, y + dy) \quad (4.7)$$

Brightness constancy, a basic assumption of optical flow, requires the intensity of the object to remain constant while in motion, as shown in Eq. 4.7. Based on this property, we introduce the intensity constraint to force the intensity of pixels to be constant along

the line.

$$\mathcal{L}^{intensity} = \sum_{i=0}^{W^p-1} \|I^p(x_{i+1}, y_{i+1}) - I^p(x_i, y_i)\|_2^2 \quad (4.8)$$

Then we apply a first order Taylor expansion to Eq. 4.7, which estimates the velocity of the point towards y axis direction at different positions. Based on this, we introduce the smoothness constraint to force the plot line to be differentiable everywhere, i.e.

$$\begin{aligned} V(x, y) &= -\frac{I_x^p(x, y)}{I_y^p(x, y)} \\ \mathcal{L}^{smooth} &= \sum_{i=0}^{W^p-1} \|y_{i+1} - y_i - V(x_i, y_i)\|_2^2 \end{aligned} \quad (4.9)$$

where I_x^p, I_y^p denote the gradient map along x-direction and y-direction, respectively. $V(x, y)$ denotes the velocity of the point along y-direction.

Also, we introduce the semantic constraint to compensate the optical flow estimation and force more foreground pixels to fall into the plot line.

$$\mathcal{L}^{semantic} = \sum_{i=0}^{W^p} \|1 - \tilde{C}(x_i, y_i)\|_2^2 \quad (4.10)$$

Therefore, the total loss for plot line detection is

$$\mathcal{L}^{line} = \mathcal{L}^{smooth} + \mathcal{L}^{intensity} + \mathcal{L}^{semantic} \quad (4.11)$$

4.3. Experiments

In this section, we conduct extensive experiments to validate the effectiveness and superiority of the proposed method.

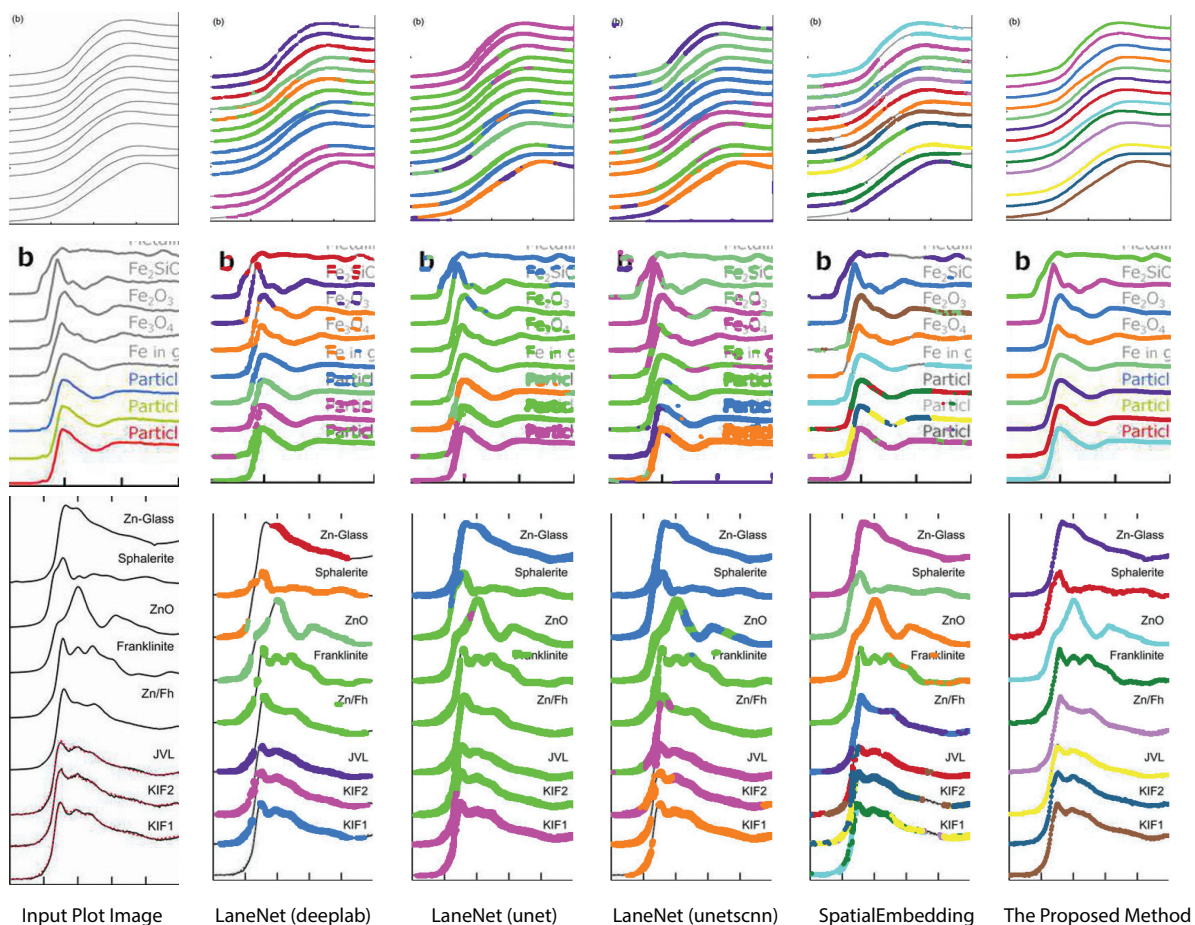


Figure 4.4. Comparison of the results of plot data extraction between instance segmentation algorithms and the proposed method. The first column is the input plot images and the rest columns are plot line detection results with different methods. All pixels belonging to a single line (data instance) should be assigned to the same color. Figures are from from Ref. [74, 1, 173] (top to down).

We collected a large number of graph images from literature using the EXSCLAIM! pipeline [92, 186] with the keyword "Raman" and "XANES". Then we randomly selected 1800 images for the axis alignment task, with 800 images for training and 1000 images for validation. For the plot data extraction task, we labeled 336/223 plot images as

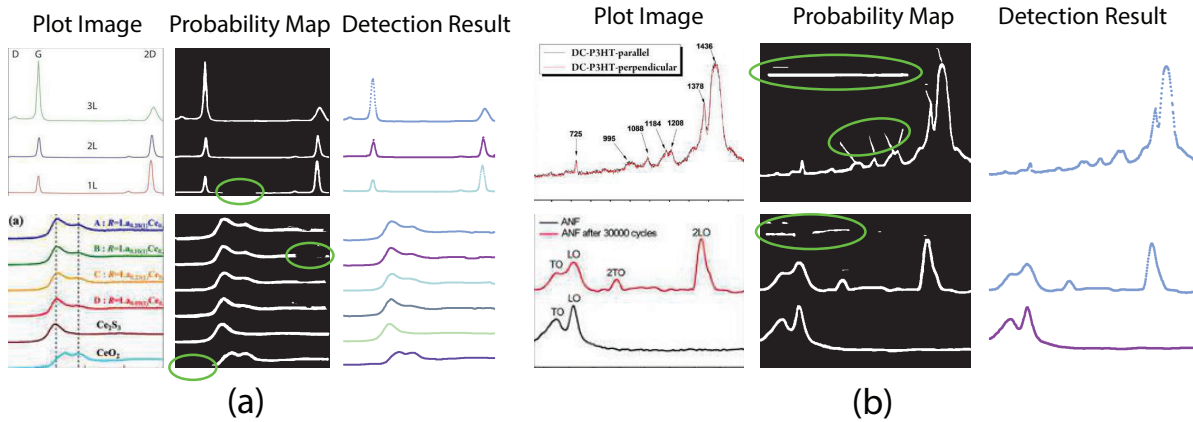


Figure 4.5. Plot line detection with imperfect semantic segmentation results. (a). Part of plot lines is missing in the probability map. Figures are from Refs. [69, 57] (top to down). (b). Background pixels are misclassified as foreground. Figures are from Refs. [164, 238] (top to down).

the training/testing set with LabelMe [227]. One thing to clarify here is that these real collected plot images may look in “low quality”, as shown in Figs. 4.4-4.6

During the training process, we implement all baseline object detection models [213, 229, 243] with the MMDetection codebase [31]. The re-implementations strictly follow the default settings of MMDetection. All models are initialized with pre-trained weights on the MS-coco dataset and then fine tuned with SGD optimizer with the labeled dataset for 1000 epochs in total, with initial learning rate as 0.005. Weight decay and momentum are set as 0.0001 and 0.9, respectively. We train the semantic segmentation module from [148] in a semi-supervised manner. In particular, we simulate plot images with variations on the number/shape/color/width of plot lines, with/without random noise/blur and then we train the model alternatively with the simulated data and real labeled data for 1000 epochs.

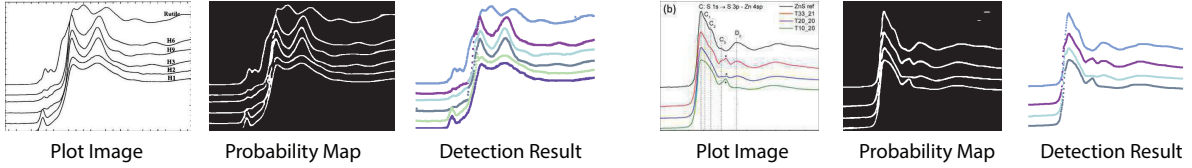


Figure 4.6. Plot line detection with hard examples. Figures are from Refs. [233, 43] (left to right).

The optical flow based method is implemented in Algorithm. 1.

$$\begin{aligned}
 \mathbf{SemanticMap}(\hat{y}^k, y_{cand}) &= \begin{cases} \hat{y}^k, & \text{if } \min_{y \in y_{cand}} \|y - \hat{y}^k\|_2^2 < \Delta_s \\ \arg \min_{y \in y_{cand}} \|y - \hat{y}^k\|_2^2, & \text{Otherwise} \end{cases} \\
 \mathbf{ColorMap}(x, \hat{y}^k) &= \begin{cases} \hat{y}^k, & \text{if } \min_{y \in \mathcal{N}(\hat{y}^k)} \|I^P(x, y) - \{I^P(x, \hat{y}^k)\}\|_2^2 < \Delta_c \\ \arg \min_{y \in \mathcal{N}(\hat{y}^k)} \|I^P(x, y) - \{I^P(x, \hat{y}^k)\}\|_2^2, & \text{Otherwise} \end{cases}
 \end{aligned} \tag{4.12}$$

where $\mathcal{N}(y)$ denotes the neighborhood of y , all values in the interval between $y - \delta$ and $y + \delta$. Empirically, δ is 10 in this paper. Δ_s, Δ_c are two thresholds, which help to suppress imperfection in the probability map (e.g. reject misclassified background pixels and inpaint missing foreground pixels). It is important to note that a proper value for Δ_s is important for doing a successful plot line detection. As shown in Fig. ??, if the Δ_s is too large (top row), the proposed method fails to inpaint correct misclassified foreground pixels, if the Δ_s is too small, the proposed method fails to compensate the error from optical flow estimation in case of sudden gradient change (e.g. peak).

Visual comparison on plot line detection between the proposed method and conventional instance segmentation algorithms is shown in Fig. 4.4. In particular, we have 4

Algorithm 1 Optical Flow based Algorithm for Plot Line Detection

Initialization: $V(x, y)$, \tilde{C} , $I^p(x, y)$
 Pick a point from each plot line as the start position $\{(x_t, y_t^k) \mid k = 1, 2, \dots, M\}$, where M is the number of plots
while $t < W^p$ **do**
 $\hat{y}_{t+1}^k \leftarrow y_t^k + V(x_t, y_t^k)$ \triangleright Estimate the next position of the point with optical flow
 $y_{cand} \leftarrow \{\tilde{C}(x_{t+1}) == 1\}$ \triangleright Select pixels of plot data in the semantic map
 $\{\hat{y}_{t+1}^k\} \leftarrow \mathbf{SemanticMap}(\{\hat{y}_{t+1}^k\}, y_{cand})$
 $\{y_{t+1}^k\} \leftarrow \mathbf{ColorMap}(x_{t+1}, \{\hat{y}_{t+1}^k\})$
 $t \leftarrow t + 1$
end while
while $t > 0$ **do**
 $\hat{y}_{t+1}^k \leftarrow y_t^k - V(x_t, y_t^k)$ \triangleright Estimate the next position of the point with optical flow
 $y_{cand} \leftarrow \{\tilde{C}(x_{t-1}) == 1\}$ \triangleright Select pixels of plot data in the semantic map
 $\{\hat{y}_{t-1}^k\} \leftarrow \mathbf{SemanticMap}(\{\hat{y}_{t-1}^k\}, y_{cand})$
 $\{y_{t-1}^k\} \leftarrow \mathbf{ColorMap}(x_{t-1}, \{\hat{y}_{t-1}^k\})$
 $t \leftarrow t - 1$
end while

baseline methods: LaneNet [149] with Deeplab [32] as the backbone, LaneNet with Unet [176] as the backbone, LaneNet with Unetscnn [159] as the backbone and SpatialEmbedding [148]. All the instance segmentation algorithms fail to distinguish pixels from different plot lines especially when the number of lines increases and the distance between adjacent lines decreases (e.g. first row in Fig. 4.4). As expected, the proposed optical flow based method correctly groups pixels into plot lines. Moreover, the proposed method still works even with imperfect semantic segmentation prediction. As shown in Fig. 4.5, the proposed method is able to inpaint the missing pixels and eliminate misclassified background pixels. The proposed plot line detection method also works for cases containing significant overlap between different plot lines - conditions that can even be challenging for humans to annotate or disambiguate. This is shown in Fig. 4.6.

Meanwhile, we introduce a quantitative metric to evaluate the performance of different methods. Here, $\{L^{pred}\}$ and $\{L^{gt}\}$ denote the set of detected plot lines and ground truth plot lines in the image, respectively. Then we define matched plot lines as $\{L^{match}|L \in \{L^{pred}\}, \min_{\tau \in \{L^{gt}\}} |L-\tau| < \epsilon_p\}$ while each ground truth plot line can have at most one matched plot line. Thus we have Precision = $\frac{\|\{L^{match}\}\|}{\|\{L^{pred}\}\|}$, Recall = $\frac{\|\{L^{match}\}\|}{\|\{L^{gt}\}\|}$, where $\|\cdot\|$ denotes the number of plot lines in the set and ϵ_p denotes the threshold of the mean absolute pixel distance between two plot lines. By setting different ϵ_p , we measure the performance of different algorithms, as shown in Fig. ???. As expected, the proposed algorithm achieves better precision and recall accuracy than the other methods. In particular, there are 935 plot lines in the testing set. Given $\epsilon_p = 1$, the proposed plot digitizer detects 831 matched plot lines and given $\epsilon_p = 2$, the proposed plot digitizer detects 890 matched plot lines.

4.4. Ablation study

4.4.1. Edge-based constraint

We introduce a mean absolute distance between the estimated axes and the real axes to quantitatively measure the axis misalignment. Here we use (x_{pred}, y_{pred}) and (x_{gt}, y_{gt}) to denote the estimated and ground truth point of origin of the coordinates, respectively. Then the axis misalignment is computed as

$$D^{misalign} = \frac{1}{N} \sum_i |x_{pred}^i - x_{gt}^i| + |y_{pred}^i - y_{gt}^i| \quad (4.13)$$

where N denotes the number of graph images in the testing set.

We measure the axis misalignment of the detection results using three different anchor-free object detection models (i.e. FCOS [213], FreeAnchor [243] and GuidedAnchor [229]),

Method	Refined	Axis Misalignment
FCOS [213]	No	1.49
	Yes	1.33
FreeAnchor [243]	No	4.65
	Yes	2.47
GuidedAnchor [229]	No	3.03
	Yes	1.60

Table 4.1. Axis misalignment with different anchor-free object detection models.

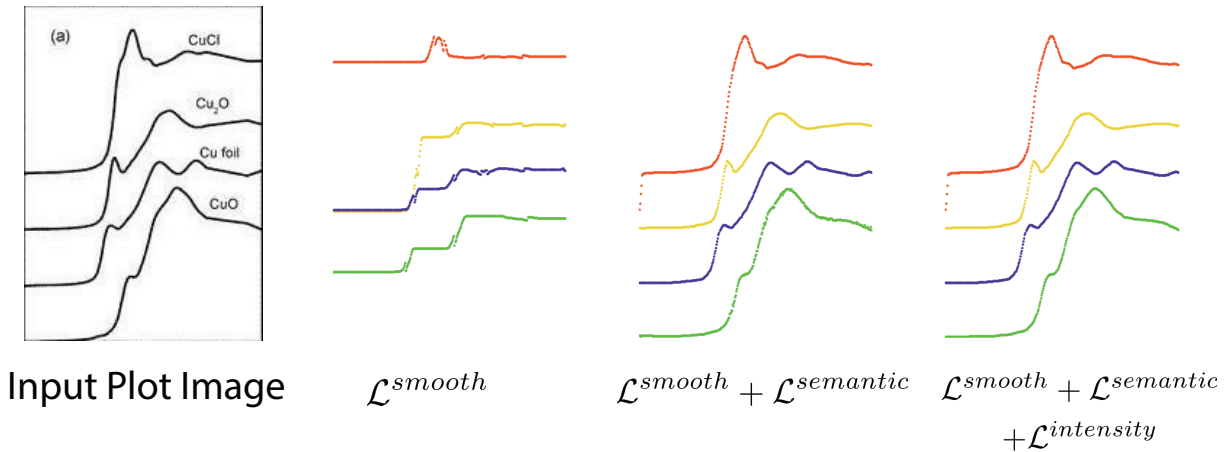


Figure 4.7. Visual comparison of plot data extraction with different losses. Figure is from Ref. [244].

with and without the edge-based constraint. A more detailed quantitative comparison between axis alignment with/without the edge-based constraint is shown in Table. 4.1. The refinement with the edge-based constraint suppresses the axis misalignment, with $\sim 10\%$ improvement for FCOS and $\sim 47\%$ improvement for the other detectors.

4.4.2. Different losses for plot line detection

We conduct experiments to study how each loss term affects performance of the plot data extraction module. A visual comparison between plot data extraction with different

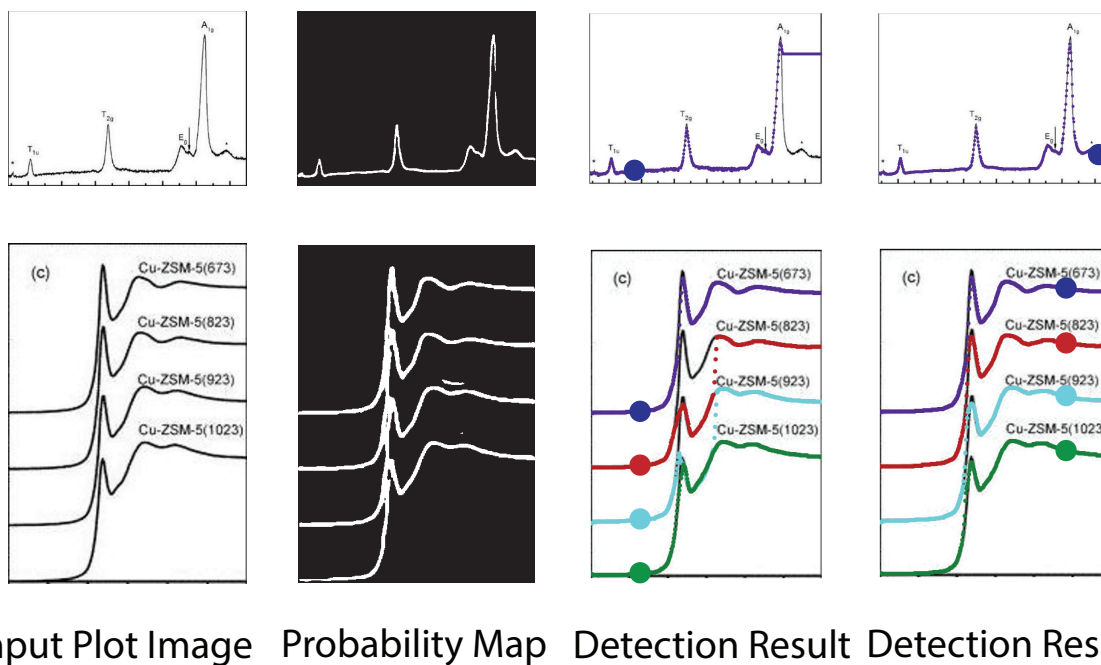


Figure 4.8. Results of plot line detection with different start position. Start positions are highlighted in the detection result images. Figures are from Refs. [234, 244] (top to down).

loss function is shown in Fig. 4.7. We take an example plot image from the testing set. With only the smoothness loss (i.e. middle left image), the extraction process is significantly affected by the imperfection of the plot image (e.g. noise), which produces an inaccurate gradient map. Adding the semantic loss term helps to compensate for the errors in the optical flow estimation, which results in significant improvements (i.e. middle right image). However, some glitches are still noticeable in the detection result (i.e. the green plot in middle right image). Finally, the intensity constraint term helps refine the detection results (i.e. right most image) by searching for the best intensity match in the neighborhood.

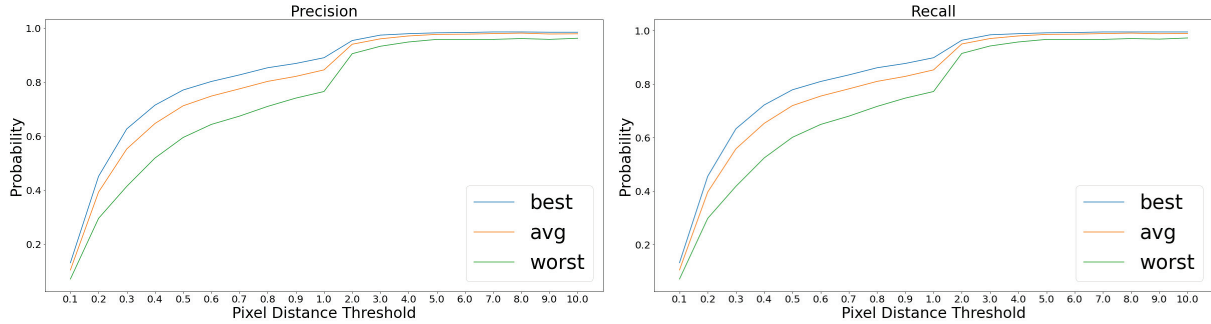


Figure 4.9. Quantitative comparison of plot data extraction with different start positions.

4.4.3. Different start positions for plot line detection

A good start position matters in the proposed optical flow method, especially in case that there are sharp peaks in the plot image or misclassified foreground/background pixels in the probability map. Intuitively, we select the start position at places where the gradients are small. As shown in Fig. 4.8, the top row shows that misclassified foreground pixels break the continuity of the plot line, which hinders the ability of tracking the motion of the point from one side. In the bottom row, there are sharp peaks in the plot image and significant overlap between different plot lines in the probability map, making it difficult to apply the optical flow method from the left side of the peak. Also, we conduct experiments to quantitatively measure the how the selection of start positions affect the performance. In particular, we randomly select ~ 20 start positions in each plot image and apply the proposed method to detect plot lines. We measure the best/average/worst performance of plot line detection with these start positions, as shown in Fig. 4.9. Clearly, selecting a proper start position is very important to the success of the algorithm.

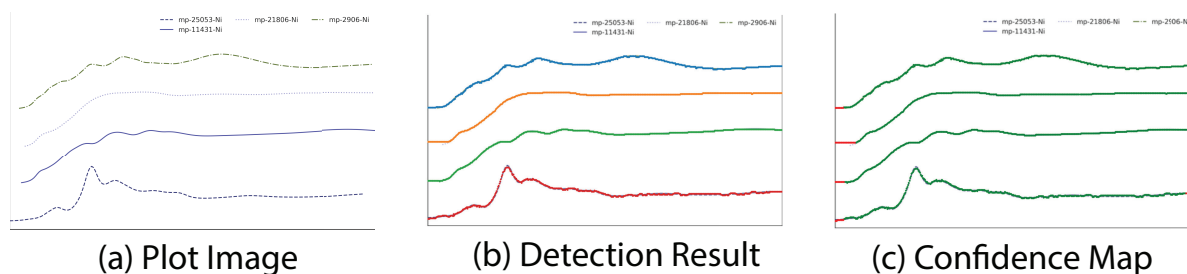


Figure 4.10. Plot line detection with simulated plot image with different line styles.

4.4.4. Different line width and line style for plot line detection

Different authors may prefer different line width or line styles when they generate the plot images. We conduct experiments to study how the variation of line width and line styles in the plot images would affect the proposed plot line detection model. For quantitative evaluation, we take 200 spectra data from the benchmark database¹ to generate a number of plot images with different settings of line width and line styles. In particular, solid line, dashed line, dashed-dot line and dotted line are commonly used and the line width is randomly selected from $\{2.0, 2.5, 3.0, 4.0\}$. Fig. 4.10(a) shows an example of the synthetic plot images, in which different plot lines have different line styles and colors. Since the semantic segmentation module is trained with both simulated data and real labeled data, we directly apply the trained model onto those synthetic plot images for plot line detection. As expected, the model works pretty well on the synthetic data. As shown in Fig. 4.10(b), each plot line is assigned with one unique color. Moreover, we noticed that the data range of different plot lines could be different, then we generate the confidence map for each

¹The XASdb[136, 36] that is hosted on the Materials Project[87] website is one of the world’s largest database for the computed X-ray absorption spectroscopy (XAS). This database stores more than 500,000 site-wise spectra and has been extensively used to accelerate the spectra interpretation through machine learning approaches[247, 216, 23]

LineWidth	RMSE	LineStyle	RMSE
2.0	0.015	solid	0.009
2.5	0.023	dashed	0.022
3.0	0.016	dashed-dot	0.016
4.0	0.018	dotted	0.026

Table 4.2. Quantitative comparison of plot data extraction with different line width or line style.

predicted plot line, as shown in Fig. 4.10(c), where data points in red are likely to be false positive prediction and data points in green are likely to be true positive prediction.

In this experiment, 195 out of 200 plot lines are detected, resulting a recall ratio of 97.5%. Then we compute the root mean square error (RMSE) between the predicted spectra data and the ground truth data with different line width or line styles. In particular, the predicted spectra data and the ground truth data of each plot line are normalized to $[0, 1]$. Also, since the coordinates of the predicted spectra data are integer while the ground truth data can be floating numbers, a bilinear interpolation is applied to align the prediction and the ground truth data. In total, the average RMSE of all detected plot lines is 0.018. We also compute the average RMSE for different settings of line width or line styles, as shown in Table. 4.2. As expected, the RMSE of solid plot lines is better than that of plot lines in other line styles since solid lines are the most widely used line style. The proposed model also performs quite robust in case of different line width and line styles.

4.5. Conclusion and discussion

In this chapter, we propose the Plot2Spectra to extract data points from XANES/Raman graph spectroscopy images and transform them into coordinates, which enables large scale

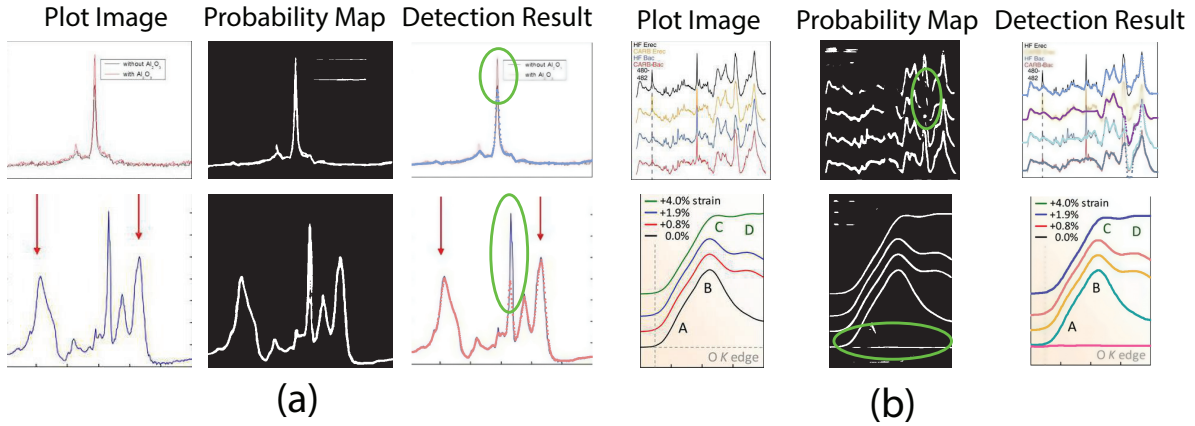


Figure 4.11. Failure cases. (a) Plot2Spectra fails when there is a significant peak. Figures are from Refs. [124, 11] (top to down). (b) Plot2Spectra fails when a large portion of background/foreground pixels are misclassified. Figures are from Refs. [47, 27] (top to down).

data collection, analysis, and machine learning of these types of spectroscopy data. The novelty of the technique is that we propose a hybrid system to address the problem. Due to the insufficient feature of plot lines, it is difficult for conventional instance segmentation methods to find a proper embedding space. To this end, we decouple the problem into an easy-to-model part (i.e. optical flow for data point tracking) and a hard-to-model part (i.e. CNN for plot data segmentation). Extensive experiments validate the effectiveness and superiority of the proposed method, even for very challenging examples.

Unfortunately, there are some cases that the proposed model is likely to fail. As shown in Fig. 4.11(a), the current model fails to detect sharp peaks. This is because the first order Taylor approximation in Eq. 4.9 does not hold for large displacement. A possible way to address this issue is to stretch the plot image along x-direction, which reduces the slope of the peaks. Another kind of failure cases is shown in Fig. 4.11(b). Since the

proposed plot line detection algorithm detects plot line in a sequential manner, the error from the previous stage (i.e. semantic segmentation) would affect the performance of the subsequent stage (i.e. optical flow based method). Even though the optical flow based method is robust if some pixels are misclassified (as shown in Fig. 4.5), significant error in the probability map would still result in a failure. A possible way to address this issue could be training a more advanced semantic segmentation model with more labeled data.

CHAPTER 5

Representation learning for STEM images**5.1. Introduction**

Microscopic imaging providing the real-space information of matter in a large range of scale, which plays an important role for understanding the correlations between structure (e.g. morphology, phase, atomic structure, surface facet, interfacial structure) and properties in the field of materials science. The structure-property linkage allows researchers to optimize new materials for applications by controlling the structure of materials. Scanning transmission electron microscopy (STEM) imaging technique has been intensively used to collect real-space information of crystallographic structure and defects [174, 133, 224, 224, 58, 134, 137, 252, 251]. STEM images usually have two unique capabilities: 1). provide an atomic number-contrast imaging where one can distinguish the elements directly from the contrast (e.g. high angle annular dark field (HAADF)) 2). have much better spatial resolution, which is determined by the electron probe size and can be down to sub-Armstrong.

Researchers in the field of STEM imaging analysis are usually interested in the characterization of the atomic structure, the nature of defects, as well as the morphology of samples. To this end, extracting the structural information (e.g. atomic positions [224], dumbbells [252]), column heights of atoms [133], lattice types [224]) plays a very important role in exploration of the crystallographic phases, atomic configurations and

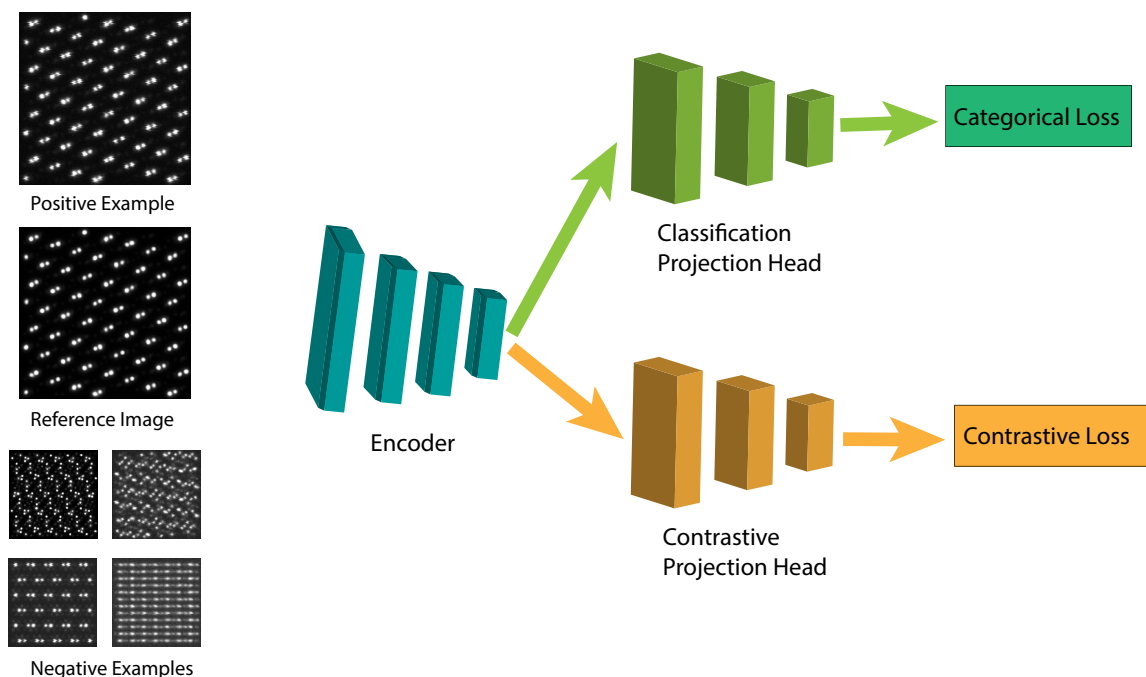


Figure 5.1. The proposed framework for HAADF-STEM image retrieval.

the insights behind the structure related material-specific properties and performance. However, it is a challenging task to deduce the structure information from STEM measurements. Currently, a typical way to do this requires domain expertise to come up with a few possible candidates structure given the STEM image and running an exhaustive search over all possible simulations (e.g. projection from different orientation, distortion) with hand-crafted metric (e.g. structural similarity index measure (SSIM)) to find the best match, which could be really time-consuming and not practical to scale up.

In this chapter, we present a representation learning framework for HAADF-STEM image retrieval (as shown in Fig. 5.1), named STEM2SIM, aiming to deduce the structure information (e.g. crystalline structure) from the given STEM image by efficiently find the

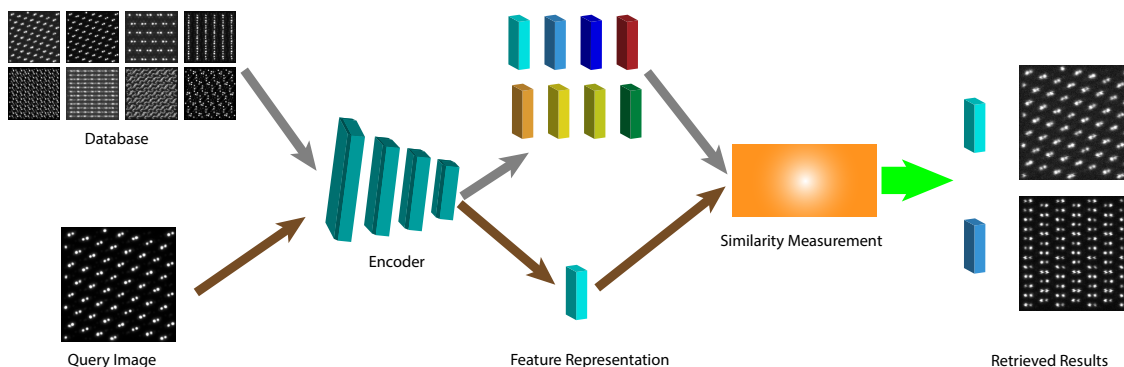


Figure 5.2. A typical pipeline for STEM image retrieval. The query image and all images in the database are first fed into the encoder to generate feature representations in the feature space, then the similarity measurement module computes the similarity between the feature representation of the query image and the feature representations of all images from the database. Then the retrieval system pops out the top similar candidates as the result.

similar image (i.e. known structure) from a simulated dataset. In particular, the key component of the proposed framework is the encoding module (also known as "encoder" or "feature extractor", we may use these terms interchangeably in the rest of the chapter), which extracts the visual and semantic information from the given image and encodes the information into a compact feature representation. In order to optimize the parameters in the encoder, we introduce two different types of downstream task, the classification branch and the similarity measurement branch. In this chapter, we also conduct extensive experiments to validate the effectiveness of the proposed framework and ablation study to understand the insights behind each key component.

5.2. Methods

In this section, we talk about the methodology behind the design of our STEM2Sim system. In particular, we treat the task as an image retrieval task, in which we extract the structure information from the STEM measurement by finding its similar counterpart in a known materials simulation database and propagate the labels. As shown in Fig. 5.2, a typical image retrieval pipeline for STEM images is comprised of the query image, database of images with known labels, an encoder to extract visual and semantic information from the query image and all images in the database, a similarity measurement module. Designing a proper encoder is the key to the success of the retrieval task.

The first thing that worth mention here is how to define and measure the similarity between STEM images. In general, visual similarity and semantic similarity are commonly used to in image retrieval. Visual similarity measures the difference between two images with respect to human visual system. Structural similarity index measure (SSIM) and peak signal-to-noise ratio (PSNR) are two popular hand-crafted metrics used to measure the similarity, in which SSIM is focused on the structure information (e.g. luminance, contrast and structure) while PSNR pays more attention to the relative strength between the distorting noise and the signal. More recently, deep learning based perceptual loss [95] have been widely used as a regularization term to minimize the visual similarity, which uses a few convolutional layers of a pre-trained network (e.g. VGG [200]) and measures the Euclidean distance between the representations in the feature space. Semantic similarity measure the categorical similarity between two images, which relies heavily on human’s preference. For example, in a coarse-grained scenario, images of different cats (e.g. white cats, black cats) are considered to be semantically similar while images of cats and images

of dogs are considered to be semantically different. Conversely, in a fine-grained scenario, types of cats or cat breeds may also need to be taken into consideration, then images of Siamese cats are similar counterparts, while image of Siamese cats and images of British Shorthair are considered to be different. Usually, visual similarity values more on the low-level information about the image (e.g. edges, color) while semantic similarity values more on the high-level abstraction of the image, both metrics are important for design a real-world image retrieval application. As shown in Fig. 5.3, given the reference image in (a), which has an ICSD id of 4 and a space group number of 165, we may have a few similar images from the database¹. (b) shows a visual similar example, which has a visually similar pattern as that of the reference image, however, this is actually a measurement from a totally different material, which has an ICSD id of 9508 and a space group number of 173. (c) is generated by adding the source size broadening effect with Gaussian blur and constant background bias, which is both visually similar and semantically similar to the reference image. (d) looks quite different from the reference image, but in fact, it refers to the same structure prototype as the reference image, but is measured from a different viewing angle.

In this section, we propose a generic framework for STEM image retrieval, as shown in Fig. 5.1. This framework is designed on top of the typical self-supervised contrastive learning framework [34], which is comprised of the following components:

- Input: We take one reference image (i.e. query image, we may use these two terms interchangeably), one or a few of its counterparts and a few negative counterparts

¹<https://github.com/MaterialEyes/atomagined>

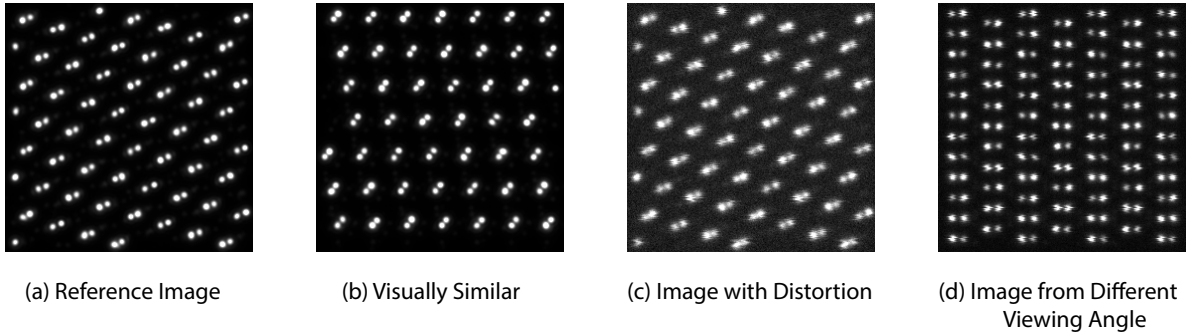


Figure 5.3. Examples of similarity for STEM measurements. (a). A reference STEM image (ICSD id: 4, space group number: 165). (b) A visually similar but semantically different example (ICSD id: 9508, space group number: 173). (c) and (d) refer to the same structure prototype as (a), (c) is generated by adding the source size broadening effect with Gaussian blur and constant background bias while (d) is measured from a different viewing angle.

as the input of the framework. The negative samples are images that have different semantic labels, and the positive samples are measurements generated from the same structure prototype by adding distortion or changing projection/viewing angles.

- Encoder: The encoder maps the given image into a vector representation, not only the reference image, but also its positive/negative counterparts. In our experiments, we used the feature extraction module of ResNet-18 [70] and the dimensionality of the feature representation is 512.
- Projection head: With the finding of [34], using a nonlinear projection head (multi-layer perceptron [68]) to map the feature representation to a lower dimensional metric space for contrastive loss computing results in a performance boost. In our experiments, we use nonlinear projection heads and identity projection for

the contrastive learning branch and multi-layer perceptron for the classification learning branch.

- **Loss:** Different loss functions are introduced to measure the difference between the prediction and the ground truth. There are two different types of task in our experiments. In classification task (i.e. space group number prediction), we use the cross entropy loss. In contrastive learning, we use the NT-Xent (the normalized temperature-scaled cross entropy loss) [34].

Classification learning branch. Classification task aims at assigning different labels to different images, which is quite commonly used as the downstream task for representation learning. Moreover, classification loss usually tends to generate feature representations with a centralized manifold while the center is the prototype representation for the category. Given the query image x_q , positive samples $\{x_p\}$ and negative samples $\{x_n\}$, we may have the loss for the classification branch:

$$\mathcal{L}^{cls} = CE(Proj^{cls}(Enc(x_q)), y_q) + \sum_{t \in \{\{x_p\}, \{x_n\}\}} CE(Proj^{cls}(Enc(t)), y_t) \quad (5.1)$$

where $CE(\cdot)$ denotes the cross entropy loss, $Proj^{cls}(\cdot)$ denotes the projection head for the classification branch, in which we use a multi-layer perceptron with a single hidden layer of size 512 and output vector size equals to the number of categories, $Enc(\cdot)$ denotes the encoding module. y denotes the ground truth classification label.

Contrastive learning branch. Contrastive learning aims to force the representations of similar images to be close and that of different images to be distance. Pair-wise and triplet loss are quite commonly years in image retrieval task, in which the loss force the distance to be 0 for positive image pair and 1 for negative pair (i.e. pair-wise loss) or

the distance between positive pair should be less than that of negative pair (i.e. triplet loss). However, since the number of positive samples is far fewer than that of negative samples, triplet loss could be really inefficient. In this work, we introduce a more efficient contrastive loss [34, 100]:

$$\mathcal{L}^{cts} = - \sum_{p \in \{x_p\}} \log \frac{\exp(\text{sim}(\text{Proj}^{cts}(\text{Enc}(x_q)), \text{Proj}^{cts}(\text{Enc}(p)))/\tau)}{\sum_{t \in \{x_n\}} \exp(\text{sim}(\text{Proj}^{cts}(x_q), \text{Proj}^{cts}(\text{Enc}(t)))/\tau)} \quad (5.2)$$

where Proj^{cts} denotes the projection head for the contrastive learning branch. $\text{sim}(\cdot)$ denotes the similarity measure function between feature vectors, in our experiments, we use the cosine similarity. τ denotes a temperature parameter, we use $\tau = 0.1$ in our experiments.

5.3. Results

In this section, we have conducted extensive experiments to validate the effectiveness of the proposed method. In particular, we use the Atomaged dataset² [157, 163] which is a collection of atomic-resolution images of unique ICSD³ (Inorganic Crystal Structure Database) structure prototypes, simulated in the high angle annular dark field (HAADF) STEM modality. All images are calculated using PRISM imaging software, and post-processed to emulate noise and distortion conditions common to the HAADF STEM imaging mode. In particular, each image from the database has two labels, that is the space group number⁴ and the structure prototype. We randomly select 672 different ICSD

²<https://github.com/MaterialEyes/atomaged>

³<https://icsd.products.fiz-karlsruhe.de/>

⁴A space group is the symmetry group of an object in space. In three dimensions, space groups are classified into 219 distinct types, or 230 types if chiral copies are considered distinct. (https://en.wikipedia.org/wiki/Space_group)

structure prototypes from the dataset for testing and use the rest images for training (\sim 6000 different ICSD structure prototypes).

Average precision. Average precision is widely used to measure the performance of a image retrieval system, which measures the area under the precision-recall curve.

$$AveragePrecision = \frac{1}{GTP} \sum_k Prec(k)Rec(k) \quad (5.3)$$

where GTP denotes the total number of ground truth positive samples, $Prec(k)$ and $Rec(k)$ denote the precision and recall ratio in the top k retrieved candidates.

First, we would like to conduct an ablation study for the projection head of the contrastive learning branch. The finding of [34] shows that a nonlinear projection head which maps the feature representation to a lower dimensional space is beneficial for unsupervised classification task. However, in our task, our ultimate goal is about designing a retrieval system other than a classifier. As shown in Table. 5.1, we apply two different projection heads in the contrastive learning branch. One is the identical projection, the other is a nonlinear projection, mapping the feature representation to a 128-d dimensional space. In particular, we compare the average precision over the testing set between 1). $ResNet_{cts}$, which is trained with the contrastive learning branch only. 2). $ResNet_{cts+sgn}$, which is trained with the contrastive learning branch and classification branch with space group number prediction. 3). $ResNet_{cts+icsd}$, which is trained with the contrastive learning branch and classification branch with structure prototype prediction. Unlike the findings in [34], the identical projection head shows a significant performance boost over the ones with the nonlinear projection head. Fig. 5.4 shows the distribution of average precision over the testing set for the three methods.

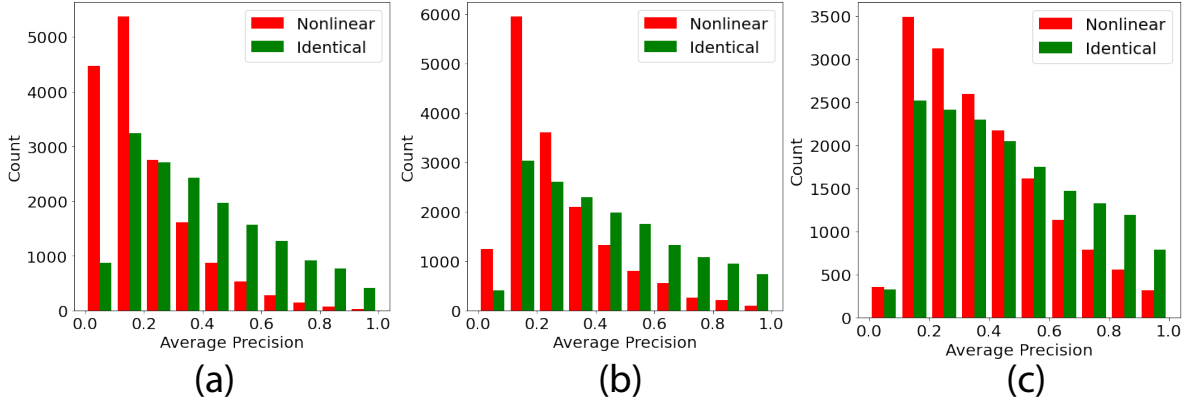


Figure 5.4. Distribution of average precision with different projection heads. (a). The distribution of average precision over the testing set which is trained with the contrastive learning branch only. (b). The distribution of average precision over the testing set which are trained with the contrastive learning branch and classification branch with space group number prediction. (c) The distribution of average precision over the testing set which are trained with the contrastive learning branch and classification branch with structure prototype prediction.

Methods	Projection Head	Average Precision
ResNet _{cts}	Identical	0.3936
ResNet _{cts}	Nonlinear	0.2076
ResNet _{cts+sgn}	Identical	0.4302
ResNet _{cts+sgn}	Nonlinear	0.2738
ResNet _{cts+icsd}	Identical	0.4586
ResNet _{cts+icsd}	Nonlinear	0.3826

Table 5.1. Results of applying different projection heads in the contrastive learning branch.

Then, we conduct experiments to study how the classification branch or the contrastive branch could affect the performance. As shown in Table. 5.2, We take the ImageNet pre-trained ResNet-18 as the baseline (i.e. ResNet_{pt}). ResNet_{sgn} and ResNet_{icsd} denote the

Methods	ResNet _{pt}	ResNet _{sgn}	ResNet _{icsd}
Average Precision	0.0125	0.1142	0.2616
Methods	ResNet _{cts}	ResNet _{cts+sgn}	ResNet _{cts+icsd}
Average Precision	0.3936	0.4302	0.4586

Table 5.2. The comparison of different methods on average precision.

ResNet-18 fine-tuned with the classification branch over the space group number prediction and the structure prototype prediction, respectively. Since the pretrained encoder is trained on natural image datasets, which have quite different distribution from the STEM measurements, it performs poorly over the testing set. After the proper fine-tuning, either with the classification task or with the contrastive learning task, the performance improves significantly. Since the task of predicting space group number is a rough-grained materials identification task, the performance is worse than the one trained directed on structure prototype prediction. Also, the encoder fine-tuned with the contrastive loss performs better than that fine-tuned with classification, showing contrastive loss is more close to the retrieval task than the classification loss. As expected, making use of both the classification branch and the contrastive learning branch results in the best performance.

Also, in order to study the insights behind the learned encoder to see how different losses could affect the performance of the retrieval system, we apply the tSNE (t-Distributed Stochastic Neighbor Embedding [222]) to visualize the manifold of the learned feature representation in 2d dimension. As shown in Fig. 5.5, the learned representations of the pretrained encoder and the one trained on the classification task with space group number prediction are not well separated. The encoder fine-tuned with both the classification task and the contrastive learning task leads to a quite satisfying separation in manifold. Fig. 5.6 shows the image retrieval results with different methods. The top row

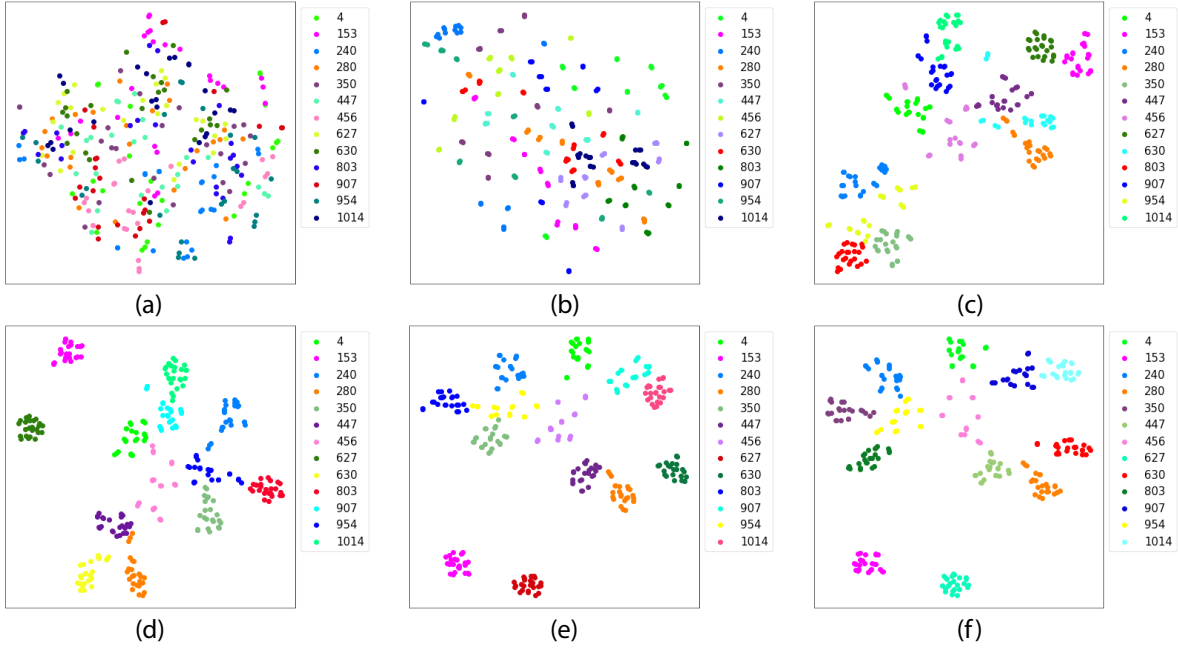


Figure 5.5. Manifold of learned feature representation with tSNE. (a) ResNet_{pt}. (b) ResNet_{sgn}. (c) ResNet_{icsd}. (d) ResNet_{cts}. (e) ResNet_{cts+sgn}. (f) ResNet_{cts+icsd}. (g) SSIM.

shows the query image, the rest rows show retrieved results with different methods, in which the similarity decreases from left to right. Obviously, the encoder trained with the contrastive loss performs better than the ones with classification loss only, while the ones trained with both losses make successful retrieval. Also, SSIM has been widely used before for similarity measurement, which is considered to be time-consuming because we have to compute the SSIM value between the query image and each image from the database every time we do retrieval. For example, given the STEM image with a size of $N \times N$, assume there are M images in the database, the time complexity of computing the SSIM values is $\mathcal{O}(MN^2)$. While in case of computing the similarity between feature representation with a size of D , the time complexity of computing the cosine similarity is $\mathcal{O}(MD)$,

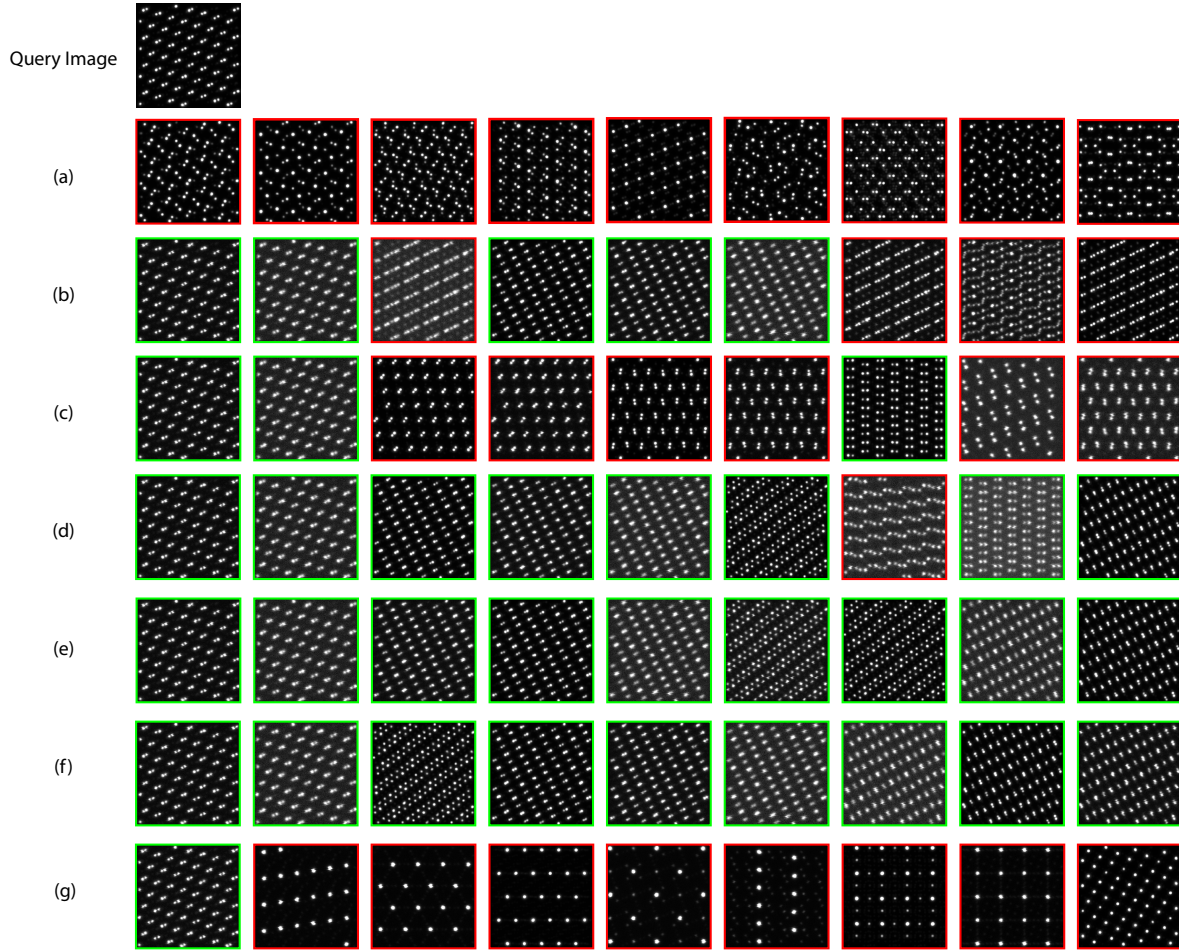


Figure 5.6. An example of STEM image retrieval with different methods (left to right the similarity score decreases), in which images in green are true positive proposals and images in red are false positive proposals. (a) ResNet_{pt} . (b) ResNet_{sgn} . (c) ResNet_{icsd} . (d) ResNet_{cts} . (e) $\text{ResNet}_{cts+sgn}$. (f) $\text{ResNet}_{cts+icsd}$.

in which $D \ll N^2$. We also show the SSIM-based retrieval in Fig. 5.6(g), which shows that SSIM is able to handle distortion-related variation but failed to recognize variation caused by different viewing/projection angles.

5.4. Conclusion

In this chapter, we present our STEM2Sim work in representation learning for STEM images, which makes use of the simulated data to deduce the structure of the material efficiently and accurately. The proposed method enables fast deducing of the crystalline structure and atomic position, which could potentially help the real-time processing of materials or set up the correct configurations for experiments.

There are some concerns related to the presented work and could be possibly improved as a future work. One is that, the model is mainly trained on simulated data only. As we know, the gap between simulated data and experimental measurements could potentially cause a performance drop when we apply the STEM2Sim to deduce the structure information from experimental measurements. Another work we could do to improve the current model is using approximate nearest neighbor to accelerate the inference process.

CHAPTER 6

Other Research Summary

A well-known notorious problem with deep learning models is its weak interpretability. In most tasks, deep learning models are used as a "black box" (very complex nonlinear function). A concern about deploying deep learning models in real-world applications is that you never when and why the model may go wrong. A common way to this issue is to decouple the task into different components, only train deep learning models in case you have (i.e. have no idea about how to model the process). By doing this, we may embed the deep learning models as one or several components in the system, bypassing the issue of interpretability to some extent. In this chapter, we present our work on designing systems to fuse the deep learning models and conventional non-learning based methods in different applications (i.e. video frame synthesis and burst image compression).

6.1. Event-driven video frame synthesis

6.1.1. Introduction

Conventional video cameras capture intensity signals at fixed speed and output signals frame by frame. However, this capture convention is motion agnostic. When the motion in the scene is significantly faster than the capturing speed, the motion is usually under-sampled, resulting in motion blur or large discrepancies between consecutive frames, depending on the shutter speed (exposure time). One direct solution to capture fast motion is to use high speed cameras, in exchange with increased hardware complexity, degraded

spatial resolution and/or reduced signal-to-noise ratio. Moreover, high speed moments usually happen instantaneously between regular motion. As a consequence, either we end up collecting long sequences of frames with a great amount of redundancy, or the high-speed moment is missed before we realize to turn on the “slow-motion” mode.

We argue that high speed motion can be acquired and synthesized effectively by augmenting a regular-speed camera with a bio-inspired event camera [19, 120]. Compared to conventional frame-based sensors, event pixels independently detect logarithmic brightness variation over time and output “events” with four attributes: 2D spatial location, polarity (e.g., “1”: brightness increases; “0”: brightness decreases) and timestamp ($\sim 1 \mu\text{s}$ latency). This new sensing modality has salient advantages over frame-based cameras: 1) the asynchronism of event pixels results in sub-millisecond temporal resolution, much higher than regular-speed cameras ($\sim 30 \text{ FPS}$); 2) since each pixel responds only to intensity changes, the temporal redundancy and power consumption can be significantly reduced; 3) sensing intensity changes in logarithmic scale enlarges dynamic range to over 120 dB¹. However, event-based cameras have increased noise-level over low framerate cameras. And the bipolar form of output does not represent the exact temporal gradients, introducing challenges for high framerate video reconstruction from event-based cameras alone.

In this section, we propose a high framerate video synthesis framework using a combination of regular-speed intensity frame(s) and neighboring event streams, as shown in Fig. 6.1. Compared to intensity-only or event-only Temporal Video Frame Synthesis (TVFS)

¹Typical dynamic range of a conventional camera is 90 dB.

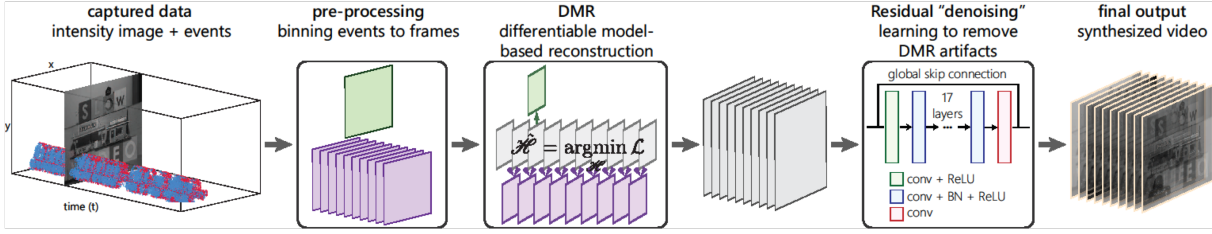


Figure 6.1. We propose a fusion framework of intensity image(s) and events for high-speed video synthesis. Our synthesis process includes a differentiable model-based reconstruction and a residual “denoising” process.

algorithms, our work takes advantages from both ends, i.e., high-speed information from events and high contrast spatial features from intensity frame(s).

6.1.2. Methods

Assume there exists a high framerate video denoted by tensor $\mathcal{H} \in \mathbb{R}^{h \times w \times d}$, $d > 1^2$. The forward sensing process results in two observational tensors, i.e., the intensity frame tensor F and event frame tensor E . Our goal is to recover tensor H based on the observation of intensity and event data.

Intensity frame tensor. We consider three sensing cases, i.e. 1) interpolation from the first and last frames of \mathcal{H} ; 2) prediction based on the first frame of \mathcal{H} and 3) motion deblur, in which case the intensity tensor is the summation over time. This can be visualized in Fig. 6.2.

Event frame tensor. As previously introduced, a pixel fires a binary output/event if the log-intensity changes beyond a threshold (positive or negative). Mathematically, the event firing process can be expressed as,

² \mathcal{H} is indexed on time axis starting from 1. Color channel is omitted here.

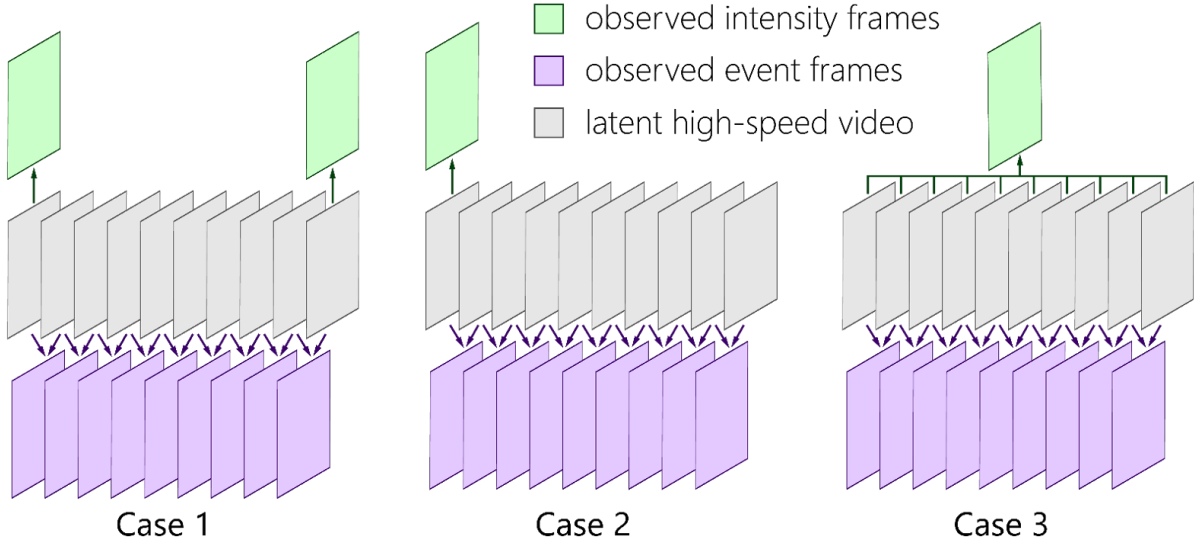


Figure 6.2. Forward models considered in this section. Case 1: interpolation from two observed intensity frames and event frames. Case 2: prediction from one observed intensity frame at the beginning and event frames. Case 3: Motion video from a single observed intensity frame and event frames.

$$e_t = \begin{cases} 1, & \theta > \epsilon_p \\ -1, & \theta < -\epsilon_p \\ 0, & \textit{otherwise} \end{cases} \quad (6.1)$$

where $\theta = \log(I_t + b) - \log(I_0 + b)$. If $\epsilon_t = 0$, no events are generated. In order to approximate this event firing process, we model each event frame as a function of the adjacent frames from the high framerate tensor \mathcal{H} , i.e.,

$$\mathcal{E}_t = \tanh \{ \alpha [\mathcal{H}_{t+1} - \mathcal{H}_t] \} \quad (6.2)$$

where α is a tuning parameter to adjust the slope of the activation curve. Based on this formulation, a video tensor with d temporal frames correspond to $d - 1$ event frames.

Differentiable model-based reconstruction. The differentiable model-based reconstruction (DMR) is performed by minimizing a weighted combination of several loss functions. The objective function is consist of a pixel-wise loss and a sparsity loss, more details are described in [231].

Residual denoiser. Although our proposed DMR can handle a variety of fusion settings, we observe that the DMR results may have visual artifacts. This is due to the ill-posedness of the fusion problem and different noise levels between the two sensing modalities. In order to address these issues, we model the artifacts outcome of DMR as additive “noise” and propose a “denoising” process to remove the artifacts. Inspired by ResNet [70] and DnCNN [242], we employ the residual learning scheme and train a residual denoiser (RD). Rather than training the denoiser from various levels of artificial noise, we design to train the network from the outcome of DMR. Mathematically, the residual \mathcal{R} is expressed as,

$$\mathcal{R} = \hat{\mathcal{H}} - \mathcal{H}_g \quad (6.3)$$

where $\hat{\mathcal{H}}$ represents the reconstructed frame from DMR, and \mathcal{H}_g represents the ground truth frame. We use a residual block similar to [241], which has a conv + ReLU and a conv layer at the beginning and end, with 17 intermediate layers of conv + BN + ReLU. The kernel size is 3×3 with stride of 1. The loss function for our denoiser is the mean squared error of $\hat{\mathcal{H}}$ and \mathcal{R} . During training, we augment data by randomizing the configuration parameters (including the running epochs) in DMR, summarized in Table 1. The goal of this augmentation is 1) to prevent overfitting; 2) to enforce learning of our DMR process;

3) to alleviate effects due to non-optimal parameter tuning. Our denoiser is single-frame, as we seek to enhance each DMR output frame iteratively without compromising the variety of DMR fusion settings.

6.1.3. Results

We design several experiments to show the effectiveness of our algorithm. For DMR, we evaluate the three cases described in Fig. 6.2 on the DAVIS dataset [143], and compare against state-of-the-art event-based algorithms, i.e., Complementary Filter [182] and Event-based Double Integral [158]. For RD, we evaluate the effectiveness of our learning strategy by comparing with Gaussian denoisers, e.g., DnCNN [242] and FFDNet [241]. We finally compare our results with a non-event-based frame interpolation algorithm, Sep-Conv [151]. Part of the results (i.e. frame prediction with DMR and RD) are displayed below, readers may refer more results can be found in [231].

Frame prediction with DMR. We show frame prediction results. We withhold the end frame of two consecutive frames and seek to predict it using the start frame and “future” events. The results are shown in Fig. 6.3. Compared to CF [182], our results are less noisy and closer to the ground truth.

Residual denoiser. We use publicly available high-speed (240 FPS) video dataset, the Need for Speed dataset [101]. The reason we choose this dataset is because it has rich motion categories and content (100 videos with 380K frames) which involves both camera and scene/object motion. We randomly split the dataset into 89 training classes and 11 testing classes. We select several video clips from the testing classes and compare our

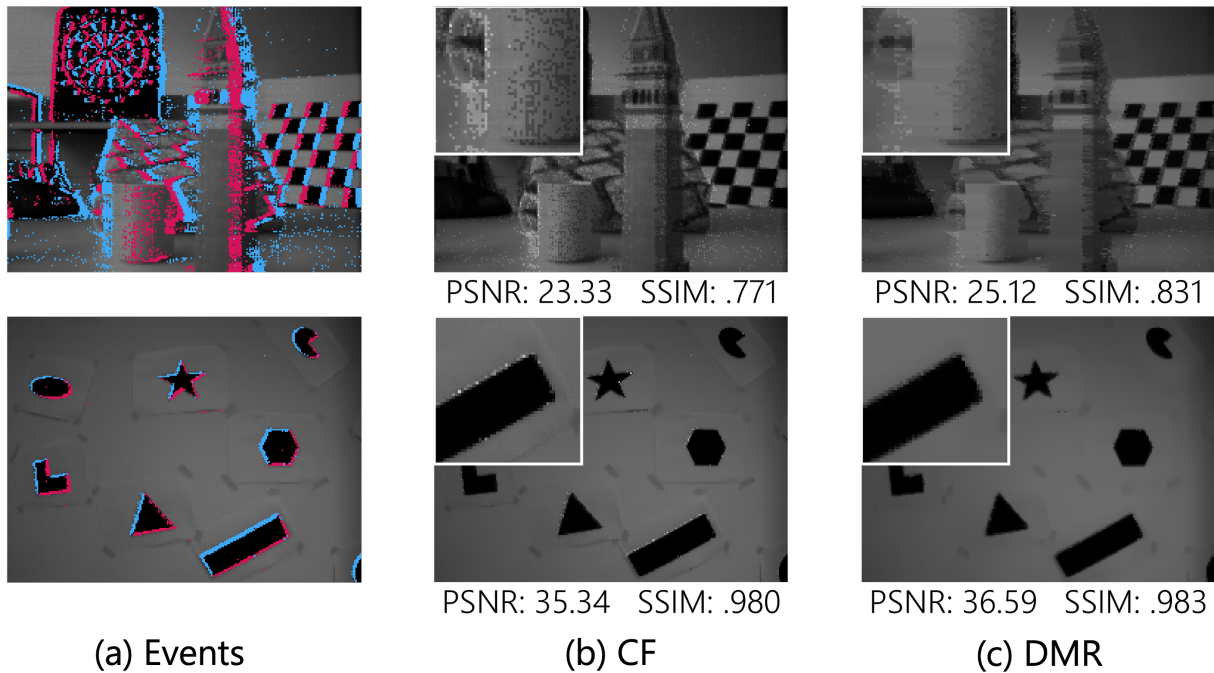


Figure 6.3. Frame prediction. Given a start frame and the future events (a), we predict the end frame (ground truth omitted). Our results using DMR alone outperforms existing algorithm, Complementary Filters (CF) [182].

results with two other denoisers, DnCNN [242] and FFDNet [241]. DnCNN is an end-to-end trainable deep CNN for image denoising with different Gaussian noise levels, e.g., [0, 55]. During our testing of DnCNN we found that the pre-trained weights do not perform well. We retrained the network using the Need for Speed dataset with Gaussian noise. The FFDNet is a later variant of DnCNN with the inclusion of pre- and post-processing. During our tuning of the FFDNet, we found that smaller noise levels (a tunable parameter for using the model) result in better denoising performance in terms of PSNR and SSIM metrics. For each testing image, we present the best tuned FFDNet result (noise level less than 10) and compare with our proposed denoiser. Partial results with zoom-in figures are presented in Fig. 6.4.

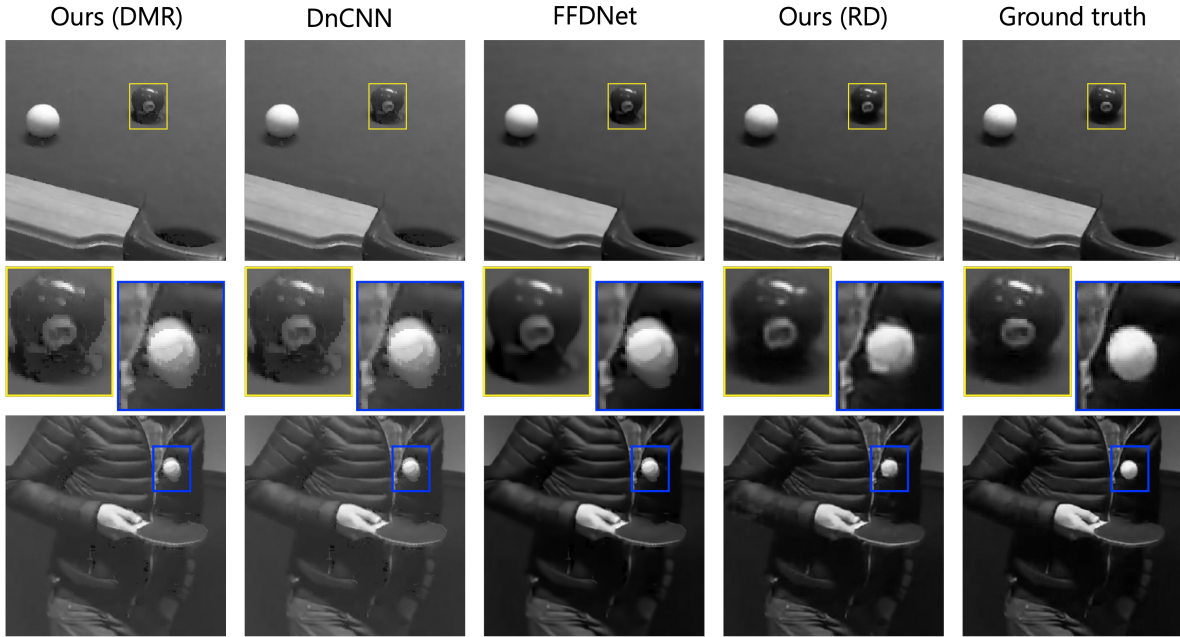


Figure 6.4. Comparison of denoising performance. Our learned Residual Denoiser (RD) reconstructs the intermediate frame (1-frame interpolation case) with fewer motion artifacts.

6.1.4. Conclusion

In this section, we have introduced a novel high framerate video synthesis framework by fusing intensity frames with event streams, taking advantages from both ends. Our framework includes two key steps, i.e., DMR and RD. Our DMR is free of training and is capable to unify different fusion settings between the two sensing modalities, which was not considered in previous work such as [158, 182]. We show in real data that our DMR performs better than existing algorithms. We show in simulation that a RD can be trained to effectively remove artifacts from DMR. Currently we train an RD from single-frame prediction case. It is interesting to further augment the training samples with all the cases, which we will investigate in the future. Applying our RD to real data faces a

domain gap due to the resolution (both spatial and temporal) and noise level mismatch. Currently, none of the existing DAVIS datasets contains enough sharp intensity images captured at high speed for training/fine-tuning. We will investigate event simulation using event simulator [168] in our future work

6.2. Task-oriented near-lossless burst image compression

6.2.1. Introduction

Burst-based imaging is one of the key techniques in modern computational photography to overcome the limitations of mobile devices' cameras. Capturing a group of images temporally close to each other, a so-called *burst*, enables very complex processing like superresolution and low-light photography with visual results unreachable with a single input image. This happens because burst frames present sub-pixel shifts with respect to each other, e.g., due to camera motion, thus providing different samplings of the captured scene.

However, the increased number of captured images represents a challenge in terms of storage. For this reason, most mobile devices today simply discard the original burst once it is processed at capture time. However, storing the original burst could be useful to enable further downstream tasks at a later stage, like super-resolution, focus switch, lighting adjustment, denoising, or other image enhancement tasks.

In this section, we therefore propose a burst-specific compression method that is able to consistently reduce the overall burst size while preserving the necessary information to enable multiple high-quality downstream image enhancement tasks. Traditional compression techniques focus on either lossless compression or lossy compression optimized

for human perception. In our problem instead, the compressed output is to be consumed by a downstream image processing algorithm, which requires us to selectively preserve information in the burst that is relevant for the task. We exploit this aspect to design a compression scheme that results in better performance compared to state-of-the-art methods both in terms of compression and distortion in the downstream task space. Particularly, we want the output of a downstream image enhancement task starting from our compressed burst to be as close as possible to the output starting from the uncompressed burst. Our proposed approach is based on a two-bitstream technique that allows us to reach near-lossless performance without compromising on compression. Moreover, we design the Lipschitz condition to predict the distortion of the compressed burst in the downstream task by simply considering the distortion of the burst frames compared to their uncompressed version, i.e., without actually processing the burst.

We summarize the contributions of our work as follows:

- We introduce the Lipschitz condition for our problem to bound the distortion in the downstream task space based on the distortion in the burst images space;
- We propose a two-bitstream near-lossless burst compression pipeline, which contains a lossy compression module and a residual coding module. This design allows us to trade-off between compression ratio and distortion of the lossy reconstruction via a hyper-parameter and reach near-lossless performance without compromising on compression;
- Our approach is the first solution for near-lossless compression of Bayer RAW images, the most common uncompressed image format produced by digital cameras;

- We evaluate our network considering superresolution as a downstream task, and show the effectiveness of our model in both image space and the task space.

6.2.2. Methods

In this section, we propose a two-stage framework for the task-oriented near-lossless compression of bursts, as shown in Fig. 6.5. Assume I_j denotes a set of burst frames, where j denotes the id of the frame. The image compression model is then denoted as (E, D) , where E, D denote the encoding and decoding modules, respectively. The encoding module transforms the image data into a feature vector, while the decoding module transforms the feature vector back to the image space. For conventional image compression algorithms, distortion is measured directly in the image space, i.e., $d_{img} = \|D(E(\{I_j\})) - \{I_j\}\|_p$, where $\|\cdot\|_p$ denotes the L^p norm. In this case, it is easy to bound the distortion by pixel-wise manipulation. In the task-oriented case instead, a downstream task T is introduced to transform the burst images from the image space to the task space, which entails that the distortion is measured in the task space, i.e., $d_{task} = \|T(D(E(\{I_j\}))) - T(\{I_j\})\|_p$. Bounding the distortion in the task space by operating on the pixels in the image space is a challenging problem given that the downstream task is often complex and highly non-linear (e.g., a neural network). The goal of our approach is to overcome this limitation and be able to control the task space distortion from the image space. We first show how to bound the distortion in the task space, and later on describe our near-lossless pipeline.

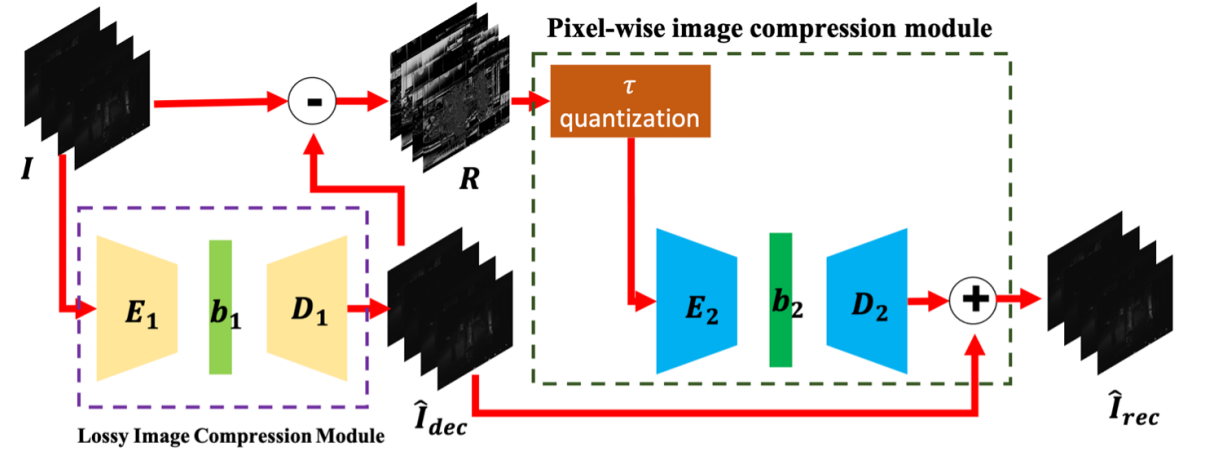


Figure 6.5. Overview of proposed near-lossless image compression system.

Image-space distortion to task-space distortion. In this section, we show how to estimate the error bound of the task-space distortion based on the error bound of the image-space distortion.

If the downstream task T is a linear transformation, it is possible to find the closed form of the correlation between the per-pixel variation in the image space with that in the task space. Without loss of generality, we first assume the downstream task is the bilinear interpolation, in which each pixel in the task space is interpolated by its adjacent pixels in the image space. We denote by (x, y) and (u, v) the coordinates of the pixels of the image in the image space and in the task space, respectively. In bilinear interpolation, the pixel (u_{2t}, v_{2t}) in the task space, of which the pixel value is $P_{2t, 2t}$, is interpolated by the pixels $\{(x_t, y_t), (x_{t+1}, y_t), (x_t, y_{t+1}), (x_{t+1}, y_{t+1})\}$ in the image space, of which the pixel values are $\{Q_{t,t}, Q_{t+1,t}, Q_{t,t+1},$

$Q_{t+1,t+1}$ }. More formally:

$$\frac{\partial P_{2t,2t}}{\partial Q_{t+k,t+l}} = \frac{(-1)^{k+l}(x_{t+1-k}-u_{2t})(y_{t+1-l}-v_{2t})}{(x_{t+1}-x_t)(y_{t+1}-y_t)} \quad (6.4)$$

where $k, l \in \{0, 1\}$. From Eq. 6.4, it is possible to see that the variation of the pixel value in the task space is linearly related to the variations of the pixel values in the image space. With $u_{2t} \in [x_t, x_{t+1}]$, $v_{2t} \in [y_t, y_{t+1}]$, we have $\left| \frac{\partial P_{2t,2t}}{\partial Q_{t+k,t+l}} \right| \leq 1$. If the distortion in the image space is bounded by ϵ , i.e., $d_{img} \leq \epsilon$, then the distortion in the task space is bounded by 4ϵ , i.e., $d_{task} = \sum_{k,l} \frac{\partial P_{2t,2t}}{\partial Q_{t+k,t+l}} \epsilon \leq 4\epsilon$. Therefore, if the downstream task is a linear transformation, it is possible to bound the per-pixel variation in the task space by bounding the per-pixel variation in the image space.

In the burst processing domain though, the downstream task is often a non-linear transformation (e.g., a neural network), which entails that $d_{task} = \max_{u,v} \sum_{x,y} \frac{\partial P_{u,v}}{\partial Q_{x,y}} \epsilon$ (where $P_{u,v}$ and $Q_{x,y}$ denote the pixel value in the task and the image space, respectively). Given the complexity of the downstream task, it might not be possible to compute this partial derivative, which indicates the pixel-wise correlation between the image and task space. Inspired by previous works on the robustness of neural networks [225, 253, 56], we introduce the Lipschitz continuity of a neural network, and bound the per-pixel variation in the task space with the estimated tight bound of the Lipschitz constant. Particularly, for a given function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, if there exists a non-negative constant $L \geq 0$ such that:

$$\|f(X) - f(Y)\|_p \leq L \|X - Y\|_p, \quad \forall X, Y \in \mathbb{R}^n \quad (6.5)$$

where the function f is Lipschitz continuous on \mathbb{R}^n , and the smallest such L is called the Lipschitz Constant (LP) of f . The Lipschitz constant is the maximum ratio between

variations in the output space and the variations in the input space and thus is a measure of sensitivity of the function with respect to input perturbations. In linear transformations, the Lipschitz constant is easy to estimate. Neural networks can be divided into linear operators (e.g., convolutions) and nonlinear operators (e.g., activation functions). The difficulty of estimating the Lipschitz constant of a neural network lies therefore in the nonlinear activation functions. Previous works found that even though activation functions are nonlinear in nature, they are usually slope restricted:

$$\alpha \leq \frac{\varphi(X) - \varphi(Y)}{X - Y} \leq \beta, \forall X, Y \in \mathbb{R}, \quad (6.6)$$

where $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ can be a nonlinear function, and $0 \leq \alpha < \beta < \infty$. In particular, the activation functions ReLU, tanh and sigmoid are all slope restricted with $\alpha = 0$, $\beta = 1$. Such slope-restricted non-linearities enable us to estimate a tight bound on the Lipschitz constant [56]. However, computing the Lipschitz constant of a neural network would require quadratic time, according to the number of neurons in the network, which is infeasible in practice. For this reason, in this work, we estimate the Lipschitz constant of the downstream task in a numerical manner. In particular, we add uniform noise to the input of the network with different means and measure the corresponding variations in the task space. Therefore, if the Lipschitz continuity holds for the downstream task, with corresponding Lipschitz constant L , we can bound the task-space distortion by manipulating image-space pixel-wise distortion, i.e., $d_{task} \leq Ld_{img}$.

A Near-lossless Image Compression Framework. We now introduce our near-lossless image compression framework, which targets the following objective function:

$$\begin{aligned} \min_E H(E(\{I_j\})) \\ \text{s.t. } \|T(D(E(\{I_j\}))) - T(\{I_j\})\|_p < \varepsilon \end{aligned} \tag{6.7}$$

where H denotes encoding costs. Assuming the Lipschitz continuity holds for the downstream task, we can rewrite the objective function as:

$$\begin{aligned} \min_E H(E(\{I_j\})) \\ \text{s.t. } \|D(E(\{I_j\})) - \{I_j\}\|_p < \frac{\varepsilon}{L} \end{aligned} \tag{6.8}$$

Inspired by previous works on "lossy plus residual" coding schemes [138, 10], we propose a two-stage framework for our compression pipeline. As shown in Fig. 6.5, the first stage utilizes a lossy image compression module to reduce the redundancy in the source image, in which we may apply any flexible lossy compression algorithms (e.g. JPEG, lossy FLIF etc.). The second stage is a pixel-wise image compression that focuses on encoding the residuals between the source image and the decoded image from the first stage. Particularly, the residual map is first fed into the τ quantization module, in which a binning process is implemented for error control, and then the quantized residual map is fed into a lossless image compression module. Thus, in the end, we use two bitstreams to encode the source image. The advantages of using this two-stage framework for near lossless image compression are two-fold. First, any lossy compression scheme (either traditional or CNN-based) can be employed in the first stage, which enables the framework to be applied in multiple compression scenarios. Second, unlike existing

learned lossy compression methods, our framework introduces invertibility, allowing for controllable distortion without need for retraining.

In particular, the τ quantization module is formulated as:

$$\hat{R} = \text{sgn}(R) (2\tau + 1) \left\lfloor \frac{|R| + \tau}{2\tau + 1} \right\rfloor, \quad (6.9)$$

where R and \hat{R} denote the residual map before and after quantization, respectively, $\text{sgn}(\cdot)$ denotes the sign function and $\lfloor \cdot \rfloor$ the maximum integer that is less than or equal to the given variable. With the τ quantization module, the per-pixel difference between R and \hat{R} is bounded by τ . To ensure the compression ratio of the second stage increases with increased tolerance on the distortion error, we apply a PMF quantization along with the τ quantization module. Assume $r_{x,y}$ and $\hat{r}_{x,y}$ denote the pixel values of the residual map R and \hat{R} at position (x, y) , respectively. According to the quantization scheme at Eq. 6.9, we have $\|\hat{r}_{x,y} - r_{x,y}\| \leq \tau$.

Similar to [180], the discretized logistic mixture model is introduced to estimate the distribution of each residual.

Note that each burst frame is in the RAW format instead of standard RGB format, which contains four RGGB channels. Thus, we may factorize the distribution of each pixel (i.e. $Prob(\hat{r}_{x,y})$) as the product of the distribution of each sub-pixel.

$$\begin{aligned} Prob(\hat{r}_{x,y}) &= Prob(\hat{r}_{x,y}^{c_r} | \mu^{c_r}(C_{x,y}), s^{c_r}(C_{x,y})) \\ &\times Prob(\hat{r}_{x,y}^{c_{g1}} | \mu^{c_{g1}}(C_{x,y}, \mu^{c_r}), s^{c_{g1}}(C_{x,y})) \times \dots \end{aligned} \quad (6.10)$$

where $c_r, c_{g_1}, c_{g_2}, c_b$ denote the four RGGB channels. $C_{x,y}$ denotes the context information.

Then we adapt the cross-channel auto-regression model for parameters estimation,

$$\begin{aligned}
\mu^{c_{g_1}} &= \mu^{c_{g_1}}(C_{x,y}) + \alpha_1(C_{x,y})\hat{r}_{x,y}^{c_r} \\
\mu^{c_{g_2}} &= \mu^{c_{g_2}}(C_{x,y}) + \alpha_2(C_{x,y})\hat{r}_{x,y}^{c_r} + \beta_1(C_{x,y})\hat{r}_{x,y}^{c_{g_1}} \\
\mu^{c_b} &= \mu^{c_b}(C_{x,y}) + \alpha_3(C_{x,y})\hat{r}_{x,y}^{c_r} + \beta_2(C_{x,y})\hat{r}_{x,y}^{c_{g_1}} \\
&\quad + \gamma(C_{x,y})\hat{r}_{x,y}^{c_{g_2}}.
\end{aligned} \tag{6.11}$$

6.2.3. Experimental Results

To evaluate the benefits of the proposed burst compression approach, we choose super-resolution as a downstream task, given its popularity. Specifically, we choose the Deep Burst SR network from Bhat et al. [15], which takes a set of burst frames and generates a single super-resolved frame.

First, we aim to investigate the impact of distortion on the individual burst frames on the final distortion of the super-resolute image generated by Deep Burst SR. Following similar convention in the near-lossless compression domain, we compute distortion τ as the largest pixel-level difference between the original and the compressed image (i.e., the H-infinity norm). As shown in Fig. 6.6(a), given the same level of distortion in the image space, the distortion in the task space may vary. Next, in Fig. 6.6(b), we compute the maximum ratio between the variations in the task space and those in the image space at different distortion levels. Notably, the sensitivity of the given neural network decreases when the distortion level increases as shown in Fig. 6.6(b).

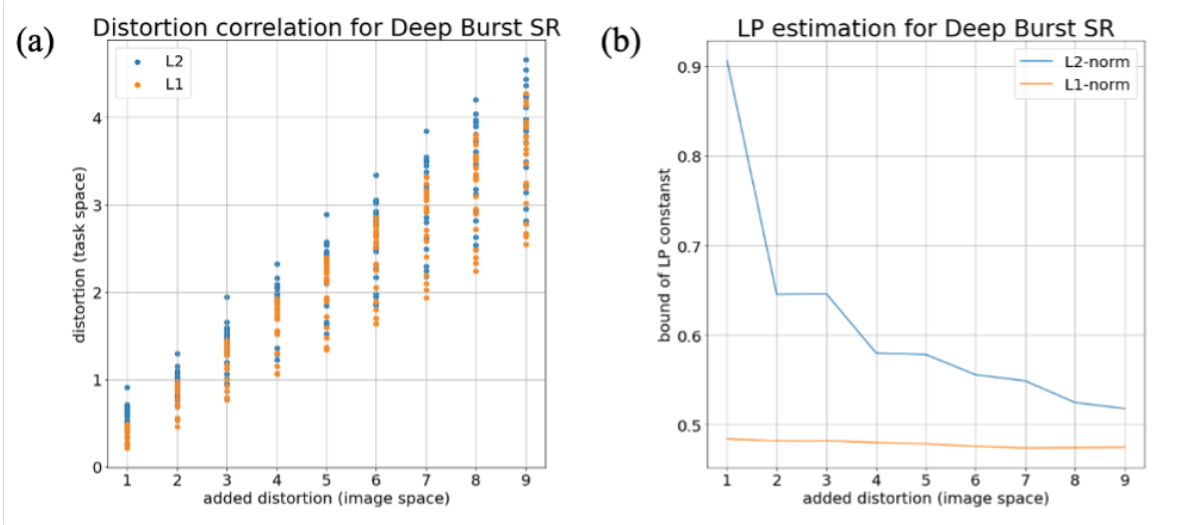


Figure 6.6. Lipschitz constant estimation when considering super-resolution as the burst downstream processing task.

To implement our method, we use the lossy FLIF algorithm (with quality setting = 25) as the lossy compression module in the first stage and adapt the PixelCNN++ algorithm as the lossless image compression model for the generation of the residual stream in the second stage. Particularly, we extended the original Pixel CNN++, which is designed to work on 8-bit RGB images, to 10-bit Bayer RAW RGGB images³, the standard uncompressed output of modern digital cameras. We use the public HDR+ dataset [66] for our experiments, which contains 3640 10-bit RGGB bursts. 80% of the bursts are used for training and 20% for validation and testing.

For evaluation, we choose JPEG LS [232] as our baseline method, where we use its near-lossless mode to control the maximum pixel-level distortion, τ . We evaluate the two methods on a set of bursts consists of 10bit Bayer raw images. The results of this

³https://en.wikipedia.org/wiki/Bayer_filter

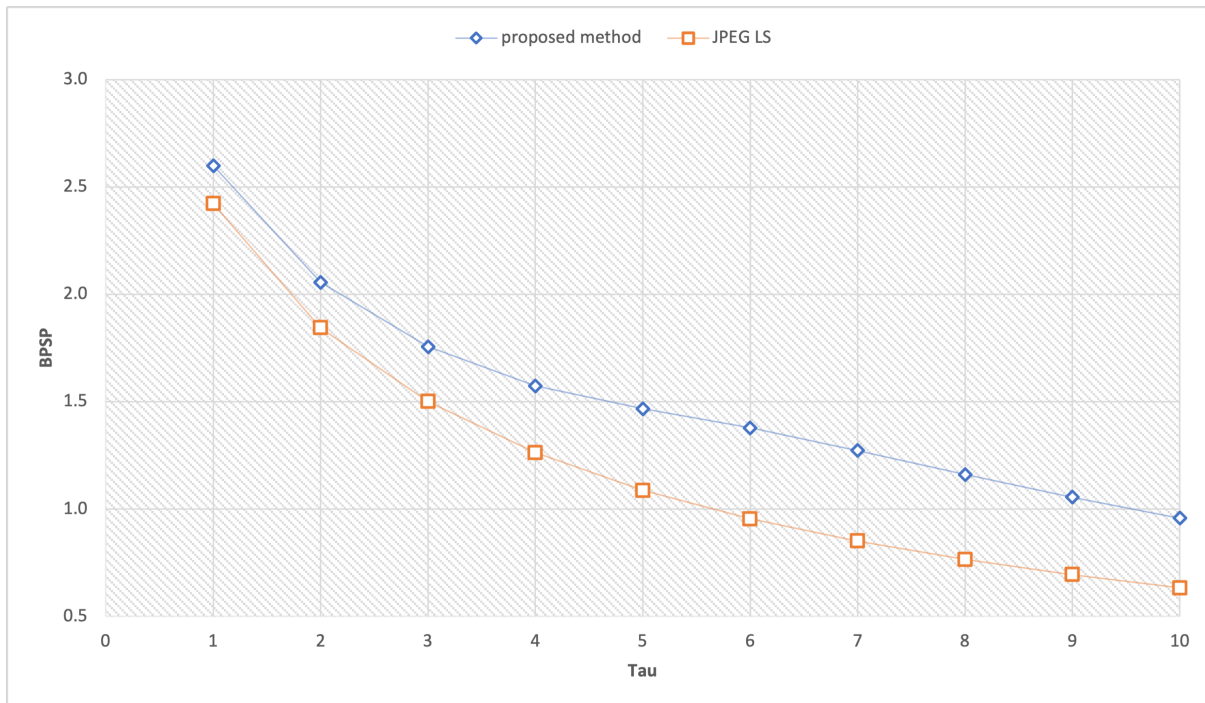


Figure 6.7. Proposed Nearlossless Compression method’s BPSP over Tau on 10bit Burst images.

analysis are as shown in Fig. 6.7, where compression rate is measured with bis per sub-pixel (BPSP) and distortion is measured in terms of Tau. As we can see, our proposed method still under-performs the highly-optimized hand-crafted codec JPEG LS. However, our proposed solution is the first learning-based solution for near-lossless compression of Bayer raw images.

6.2.4. Conclusion

In this work, we presented a novel approach for the near-lossless compression of bursts. Since bursts are meant to be processed by a downstream image processing algorithm, we design a two-stage pipeline that controls the image-space distortion of the individual

burst frames while guaranteeing a specific level of distortion in the task space. This is obtained by introducing the Lipschitz condition for our problem that relates task space distortion to image space distortion. Moreover, our approach represents the first attempt for the near-lossless compression of Bayer RAW images, the most common uncompressed output format of digital cameras. Experiments on the HDR+ burst dataset confirms the effectiveness of our scheme.

CHAPTER 7

Conclusion

In conclusion, we present our working in characterization of materials with images in this thesis. Recall the statement of problems in the Introduction Chapter, we present our work towards those issues. 1). We developed the EXSCLAIM toolkit, which enables automatic dataset construction that includes both separating out individual images from compound figures, as well as providing concise annotations describing key aspects or classification of the image content. The EXSCLAIM tool is developed around materials microscopy images, but the approach is applicable to other scientific domains that produce high-volumes of publications with image-based data as well as graphs and illustrations. 2). We developed the Plot2Spectra toolkit for the digitization of spectroscopy graphical plots in a fully automatic fashion, which enables large scale data collection, analysis, and machine learning of these types of spectroscopy data. 3). We developed the STEM2Sim framework to deduce the structure information (e.g. crystalline structure) from the given STEM image by efficiently find the similar image (i.e. known structure) from a simulated dataset, which could potentially help the real-time processing of materials or set up the correct configurations for experiments.

At this stage, the work presented in this thesis is still quite preliminary, demonstrating some possibilities to addressing the existing issues. In a more broad perspective, improvements over the proposed methods or more advanced models/algorithms/frameworks are needed for a more intelligent system that could solve all materials science problems at or

even beyond human-level. But that is exactly how research is done in all fields, we hope our work could somehow, in someday, inspire the talent researchers in this field to develop fantastic applications or systems.

As an outlook for possible future work, we may try something like this. One challenge in developing dataset construction tool or data extraction tool is the lack of proper evaluation, which is usually done by human. For example, in the developed EXSCLAIM pipeline, it is hard to quantitatively evaluate the quality of the constructed dataset precisely, especially when there are hundreds of millions of images in the dataset; also, in the proposed Plot2Spectra framework, we can only evaluate the performance of method in a limited-size of human annotated dataset or large scale simulated dataset, which may not be sufficient enough. The performance of data-driven algorithms relies heavily on the quality of the dataset, thus a self-evaluation or evaluation-in-the-loop need to be developed. Also, developing a highly intelligent deep learning model also means to make use of different modalities. In this thesis, our main focus is about the images (e.g. a little bit text/sentence is involved in the proposed EXSCLAIM), however, text information from captions/abstract/paragraphs and even texture annotation inside the images also provides essential information to understand the characterization of the material.

APPENDIX A

Supplementary Material for Compound Figure Separation**A.1. Study of the insights behind the proposed subfigure label detection**

In this section, we talk about our intuition of using the subfigure label classifier as an additional constraint for a performance boost. For general object detection targets, such as dog, cat, human, car, they contain rich features as well as high information redundancy. A full measurement of the object is not necessary for a successful recognition. For example, slight occlusion usually does not affect the recognition accuracy significantly. However, for objects like letters, which have much simpler structure and less information redundancy, even slight occlusion could make them indistinguishable.

For example, a compound figure is selected from our dataset, as shown in Fig. A.1(a). Intuitively, the letter "b" is similar in structure with the letter "h", so the letter "b" is selected here. Then we randomly crop patches around the subfigure label "b" (the red box shown in Fig. A.1(a)) to mimic the process of generating region proposals, ending up with about 200 region proposals with IoU scores over 0.7. All the region proposals are then fed into our subfigure label classifier, resulting in about 7 misclassification cases. In Fig. A.1(b), a few region proposals are selected and sorted by their IoU scores. Letters in green boxes are correctly classified as "b", while those in red boxes are misclassified as "h". Clearly, inaccuracies toward the bottom of the letter "b" can cause misclassification, even when the IoU score is as high as 0.8. Thus, adding the classification constraint along

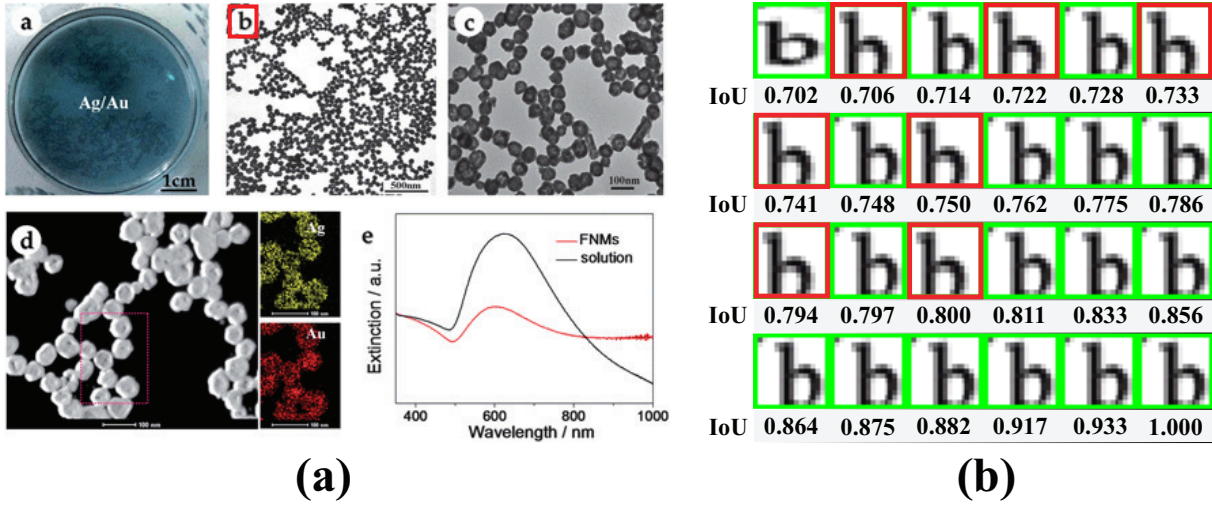


Figure A.1. Intuition of adding the classification constraint along with the high IoU constraint. (a) is a compound figure, where the letter "b" in the red box is picked out. (b) We randomly crop patches around the red box in the compound figure and sort the patches by their IoU score associated with the ground truth bounding box. Then a few patches are selected and fed into the subfigure label classifier. The number below each patch is its IoU score. The patches in green box are correctly classified while the ones in red box are misclassified as "h".

with the IoU constraint could help reject those proposals and boost the performance of the detector.

A.2. Guided detection for compound figure separation

The idea of guided detection is quite interesting, which tries to learn the implicit dependencies between two different objects (i.e. subfigure label and subfigure) and make use this information to improve the performance of the detection system for the target object. We also conduct some fun experiments on the current two-stage model, in which we intentionally manipulate the overall layout of the subfigure labels and use the different layout settings to guide the second stage detection. As shown in Fig.A.2, plugging in

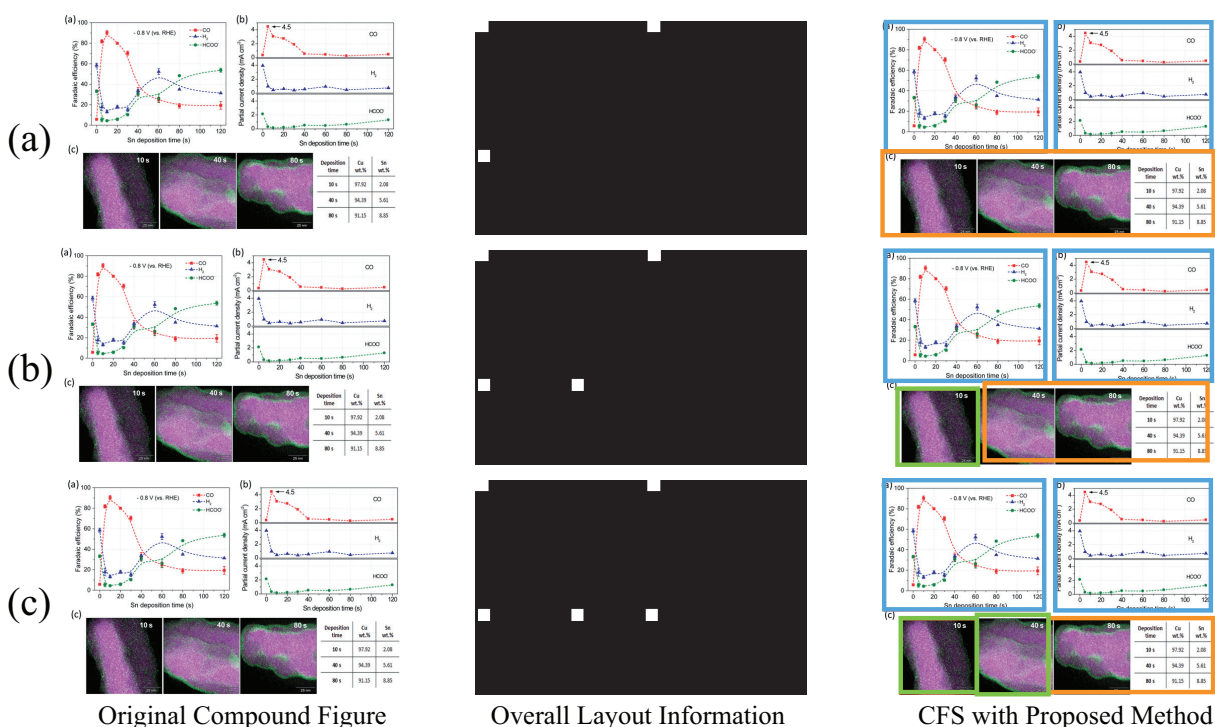


Figure A.2. Master image detection with different overall layout settings. (a-c) show that different CFS results of the same compound figure when given different overall layout information. Blue, green and orange bounding boxes mean that the image type of the subfigure is graph, microscopy, and parent respectively. Image source: (Zhao JMCA, 2016).

different overall layout information leads to different separation results. The interesting results inspired us to study the insights behind the guided detection method.

Label guided subfigure detection. By convention, authors typically place indices (subfigure labels) in front of the subfigures, indicating strong correlation between labels and subfigures. Thus, the Label Guide Subfigure Detection (LGSD) can be formulated

as:

$$\begin{aligned} \{B_i^s, S_i^s\}_{i=1}^m &= \mathcal{D}_\theta^s(I) \\ S_i^l &= S_i^s + \sum_{j=1}^n \mathcal{P}(B_j^l) \mathcal{OPN}(B_j^l, B_i^s) \end{aligned} \quad (\text{A.1})$$

where \mathcal{OPN} denotes the OPN module of the proposed framework. \mathcal{D}_θ^s denotes the subfigure detection module which outputs region proposals (B_i^s, S_i^s) . B_j^l denotes the bounding boxes of subfigure labels, which are known in this case. Since the subfigure label is known, $\mathcal{P}(B_j^l) = 1$.

The correlation between "label" and "subfigures" is different from that of "car" and "license plate", the content of the labels does not help predicting the distribution of subfigures. Instead, the layout information of labels matters. In particular, a binary mask is generate to represent the layout of the subfigure labels.

$$I_{bm}^l(x, y) = \begin{cases} 1, & (x, y) \in \{B_j^l\}_j \\ 0, & otherwise \end{cases} \quad (\text{A.2})$$

Then we construct the distribution estimation function \mathcal{M} to predict the distribution of subfigures from the binary mask

$$\{B_i^{s,l}, S_i^{s,l}\}_{i=1}^{m'} = \mathcal{M}(I_{bm}^l) \quad (\text{A.3})$$

where the estimation function \mathcal{M} outputs m' region proposals (i.e. $B_i^{s,l}, S_i^{s,l}$) with the layout information.

Then the consistency measure function is constructed as:

$$\mathcal{C}(\mathcal{M}(I_{bm}^l), B_i^s) = w_{i,j} \mathcal{IOU}(B_i^s, B_j^{s,l}) \cdot S_j^{s,l} \quad (\text{A.4})$$

where \mathcal{IOU} is measuring the intersection-over-union (IOU) between region proposals from object detection module and the estimation function. Since object detection module tends to encode intrinsic feature of the subfigure (e.g. edge, gradient) and the estimation function encodes the overall layout information, subfigures that fit both two modules are more likely to be a correct proposal. $w_{i,j}$ is an adaptive coefficient, suppressing inconsistent region proposals.

$$w_{i,j} = \begin{cases} 1, & j = \arg \max_j \mathcal{IOU}(B_i^s, B_j^{s,l}) \\ -1, & otherwise \end{cases} \quad (\text{A.5})$$

Therefore, the updated confidence score for each region proposal is:

$$S'_i = S_i^s + \sum_j^{m'} w_{i,j} \mathcal{IOU}(B_i^s, B_j^{s,l}) \cdot S_j^{s,l} \quad (\text{A.6})$$

Subfigure guided label detection. In this section, subfigure label is the target object while subfigures are correlated partners. The Subfigure Guided Label Detection (SGLD) can be formulated as:

$$\begin{aligned} \{B_i^l, S_i^l\}_{i=1}^m &= \mathcal{D}_\theta^l(I) \\ S'_i &= S_i^l + \sum_{j=1}^n \mathcal{P}(B_j^s) \mathcal{OPN}(B_j^s, B_i^l) \end{aligned} \quad (\text{A.7})$$

where \mathcal{D}_θ^l is the subfigure label detection module, which outputs region proposals (i.e. B_i^l, S_i^l). B_j^s denotes the knowledge of subfigures, which is provided. Thus, $\mathcal{P}(B_j^s) = 1$.

By observation, we notice that subfigure labels always appear near their corresponding subfigures. Also, authors tend to place pairs of labels and subfigures in a consistent way. For example, if there is a label "a" appears at the top-left corner of its corresponding subfigure, then for other subfigures in this image, their corresponding labels are more likely to appear at their top-left corners. Follow such observation, we construct the distribution function $\mathcal{M} : \mathcal{R}^{N \times N \times 3} \rightarrow \mathcal{R}^{N \times N}$. In particular, since the content of the subfigures does not affect the distribution of labels, we generate binary masks as input of the estimation function.

$$I_j^s(x, y) = \begin{cases} 1, & (x, y) \in B_j^s \\ 0, & otherwise \end{cases} \quad (\text{A.8})$$

The consistency measure function is consist of two parts, one part measures the consistency between the estimated distribution (from \mathcal{M}) and the region proposals (from \mathcal{D}_θ^l):

$$\mathcal{C}_1 = \sum_{(x,y) \in B_i^l} \mathcal{M}(I_j^s) \quad (\text{A.9})$$

The other part measure the consistency between different subfigure and subfigure label pairs. Here, we use the direction of the feature vector which connects the center of the subfigure and its corresponding subfigure label to measure such consistency. Assume (x_c^s, y_c^s) and (x_c^l, y_c^l) denote the coordinate of the center of the bounding box of the subfigure (i.e. B_j^s) and the subfigure label (i.e. B_i^l), respectively. Then the direction of the feature vector is $\mathcal{A}(j, i) = \text{angle}(B_j^s, B_i^l) = \arctan \frac{y_c^l - y_c^s}{x_c^l - x_c^s}$. The consistency between different pairs means the direction of feature vectors of different pairs should be the same. In particular,

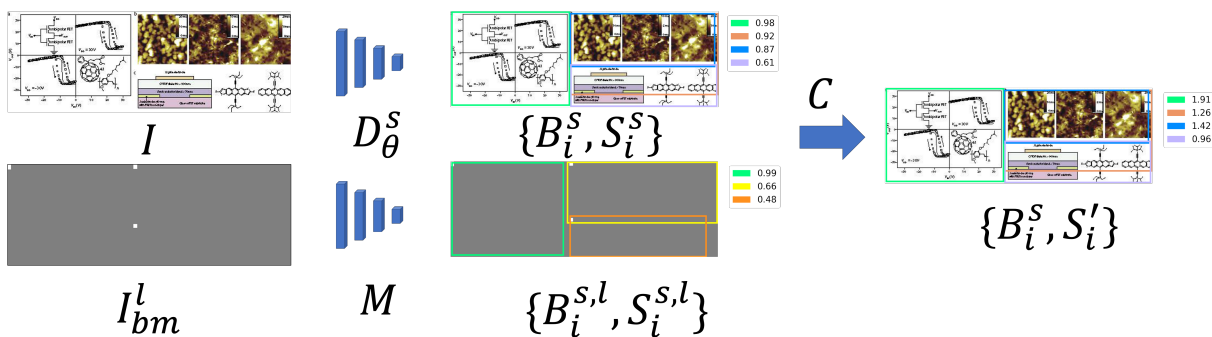


Figure A.3. Pipeline of label guided subfigure detection. (zoom in for better visualization.)

we count the number of vectors of each direction,

$$f(\mathcal{A}(j_0, i_0)) = \sum_{j,i} \mathbb{1}|\mathcal{A}(j_0, i_0) = \mathcal{A}(j, i)| \quad (\text{A.10})$$

Thus

$$C_2 = \max_j \frac{f(\mathcal{A}(j, i))}{\sum_{\mathcal{A}(j,i)} f(\mathcal{A}(j, i))} \quad (\text{A.11})$$

Therefore, the updated confidence score for each region proposal is:

$$S'_i = S_i^l + \max_j \frac{f(\mathcal{A}(j, i))}{\sum_{\mathcal{A}(j,i)} f(\mathcal{A}(j, i))} + \sum_j \sum_{(x,y) \in B_i^l} \mathcal{M}(I_j^s) \quad (\text{A.12})$$

Experiments. We conduct extensive experiments in this section. More than 2000 compound figures are crawled from the Royal Chemistry Society (RCS), Springer Nature, and American Chemical Society (ACS) journal families, and are uploaded to Amazon Mechanical Turk (MTurk), a platform for crowdsourced data labeling. Then these labeled data is firstly passed through a manually quality check process, and then randomly divided into training/validation set, with 1455 images and 620 images, respectively.

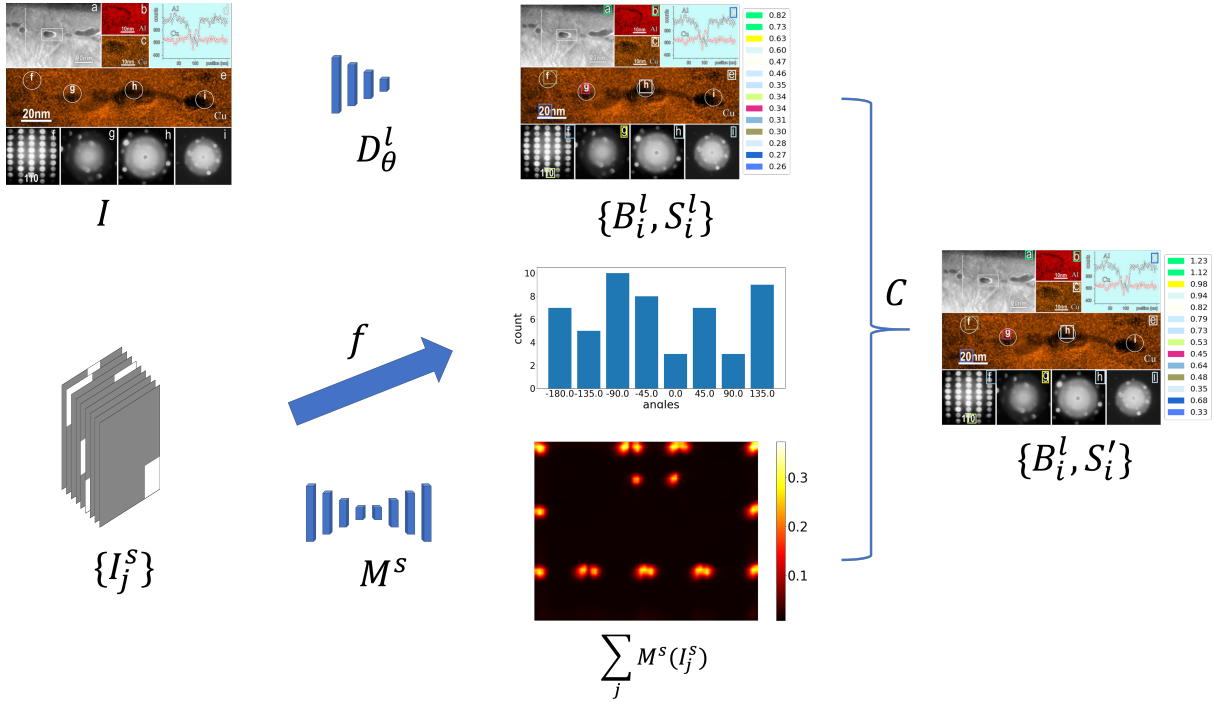


Figure A.4. Pipeline of subfigure guided label detection. (zoom in for better visualization.)

Baseline: We train an EfficientDet model with the training set for subfigure and subfigure label detection.

Label Guided Subfigure Detection (LGSD): We use the baseline to build the object detection module. The distribution estimation function is constructed with an additional EfficientDet model, which is trained to detect subfigures with a binary mask. The pipeline has been shown in Fig. A.3.

Subfigure Guided Label Detection (SGLD): We use the baseline to build the object detection module. The distribution estimation function is constructed with a typical U-Net, which is trained to estimate the distribution of labels with binary masks. The pipeline is shown in Fig. A.4.

AP	0.5	0.75	0.5:0.95
Baseline	0.981	0.949	0.896
LGSD	0.985	0.966	0.917

Table A.1. Results comparison of subfigure detection.

As shown in Fig. A.3, subfigure detection module outputs a set of region proposals based the intrinsic feature the subfigure. However, as we can see, the bounding box in orange which is supposed to be a false positive detection actually gets a larger confidence score than the true positive region proposals (i.e. bounding boxes in blue and purple). If we apply non-maximum suppression to eliminate duplicates, true positive proposals will be suppressed. The distribution estimation function in the proposed framework could fix this problem. The distribution estimation function takes the binary mask as input and outputs region proposals that match the overall layout of the compound figure. Then the consistency constraint is applied to reward region proposals (from the subfigure detection module) that are consistent with the distribution estimation. As a result, the bounding box in blue gets a higher confidence score than the blue one.

As shown in Table. A.1, the proposed LGSD model, which makes predictions with not only the intrinsic features of the subfigures but also global layout information from the distribution of labels, achieves higher accuracy than the conventional object detection model. It is worth noting that, the overall layout information helps suppress region proposals with inaccurate bounding box estimation but high confidence score, which achieves large performance gain at higher IOU threshold. Also as shown in Table. A.2, the proposed SGLD model performs superior than the conventional object detection model. Unlike LGSD model, distribution estimation of subfigure labels from subfigures is much more difficult.

AP	0.5	0.75	0.5:0.95
Baseline	0.899	0.451	0.483
SGLD	0.913	0.453	0.489

Table A.2. Results comparison of subfigure label detection.

Thus, SGLD model tends to improve the accuracy at low IOU threshold, which suppress false positive region proposal (e.g. as shown in Fig. A.4)

A.3. More compound figure separation results

In this section, more separation results with the proposed method are presented. The proposed method is able to distinguish figures with similar content but different distributed subfigure labels. As shown in Fig. A.5 and Fig. A.6, the model makes correct separation with the help of accurate detection of subfigure labels. Fig. A.7 shows incorrect separation results made by the proposed method. Due to error accumulation effect of the sequential framework, the method fails immediately when the subfigure labels are incorrectly detected. In particular, it happens when the subfigure label detector misses one or more subfigure labels, or misidentifies some non subfigure label objects as subfigure labels or a mixture of the above two.

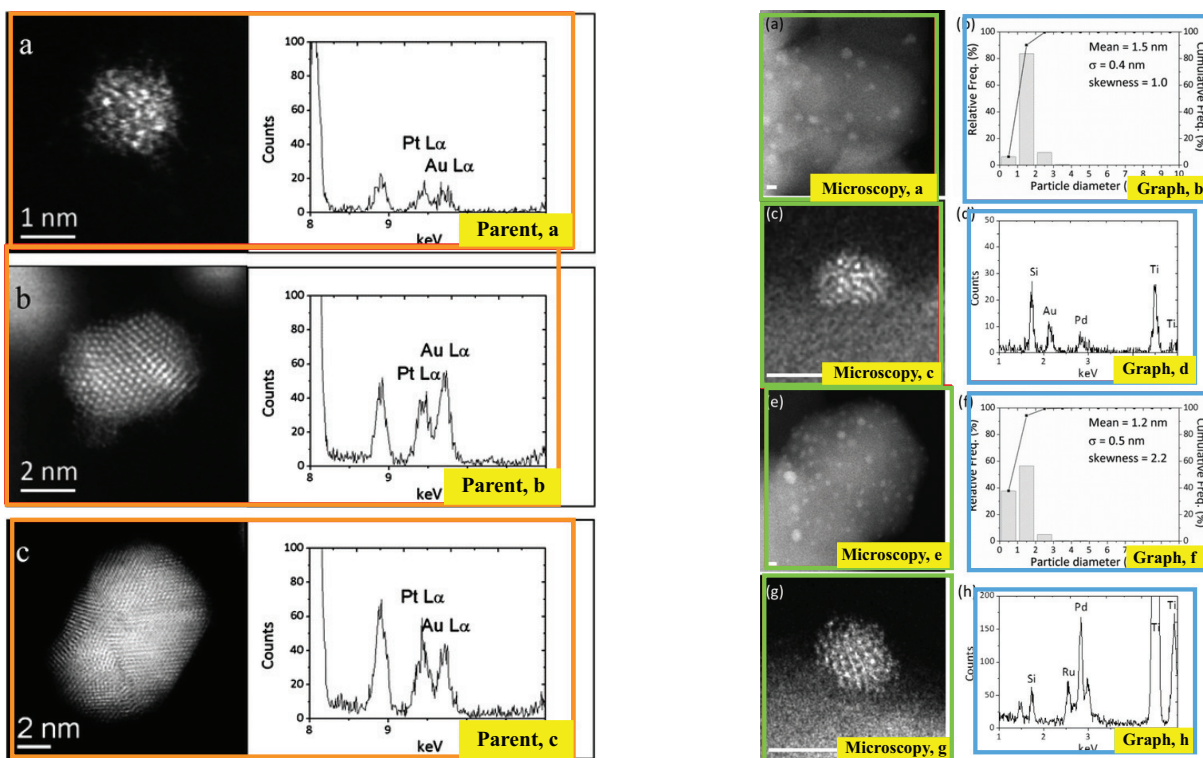


Figure A.5. Examples of similar compound figures with different overall layout information. The compound figure on the left panel is similar to the one on the right panel, in which each row consists of a microscopy image and a graph image. More subfigure labels imply a finer-level decomposition (right) while less subfigure labels lead to a much coarser level decomposition. Images come from [181, 73]

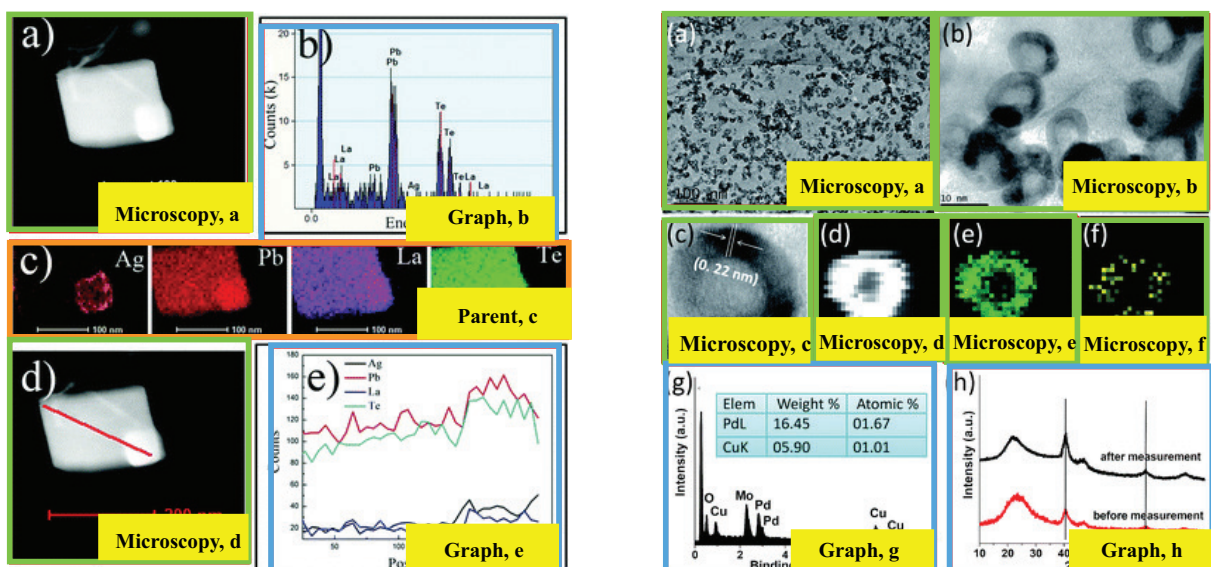


Figure A.6. Examples of similar compound figures with different local layout information. These two compound figures are similar except the second row. The second row of the figure on the left panel contains only one sub-figure label, making it a parent image. The second row of the figure on the right panel contains four sub-figure labels, making it decomposed as four separated microscopy images. Images come from [30, 81]

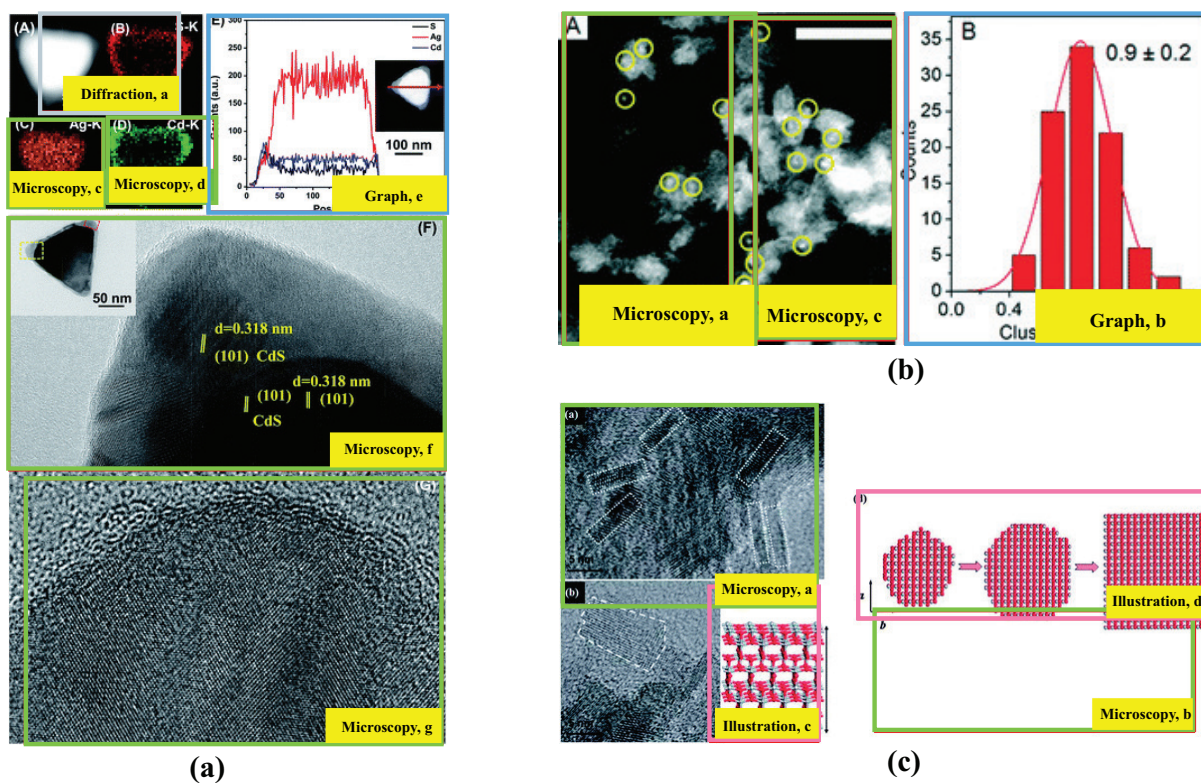


Figure A.7. Example of incorrect compound figure separation. The proposed compound figure separation approach fails due to the incorrect subfigure label detection. (a). Subfigure label "b" is missing during the subfigure label detection process, which causes a under-segmentation (b). Some internal feature of the image is being misidentified as the subfigure label "c", which causes an over-segmentation. (c). The actual subfigure label "b" is missing, while the letter "b" of the coordinate system is being misidentified as a subfigure label. Images come from [230, 240, 155]

References

- [1] Yoshinari Abe et al. “Detection of uranium and chemical state analysis of individual radioactive microparticles emitted from the Fukushima nuclear accident using multiple synchrotron radiation X-ray analyses”. In: *Analytical chemistry* 86.17 (2014), pp. 8521–8525.
- [2] Ankit Agrawal and Alok Choudhary. “Deep materials informatics: Applications of deep learning in materials science”. In: *MRS Communications* 9.3 (2019), pp. 779–792.
- [3] Ankit Agrawal and Alok Choudhary. “Perspective: Materials informatics and big data: Realization of the “fourth paradigm” of science in materials science”. In: *Applied Materials* 4.5 (2016), p. 053208.
- [4] JA Aguiar et al. “Decoding crystallography from high-resolution electron imaging and diffraction datasets with deep learning”. In: *Science advances* 5.10 (2019), eaaw1949.
- [5] Amr Ahmed et al. “Structured literature image finder: Parsing text and figures in biomedical literature”. In: *Web Semantics: Science, Services and Agents on the World Wide Web* 8.2-3 (2010), pp. 151–154.
- [6] Felipe Almeida and Geraldo Xexéo. “Word embeddings: A survey”. In: *arXiv preprint arXiv:1901.09069* (2019).

- [7] Rossella Aversa et al. “The first annotated set of scanning electron microscopy images for nanoscience”. In: *Scientific data* 5.1 (2018), pp. 1–10.
- [8] Jeonghun Baek et al. “What is wrong with scene text recognition model comparisons? dataset and model analysis”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 4715–4723.
- [9] Youngmin Baek et al. “Character region awareness for text detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9365–9374.
- [10] Yuanchao Bai et al. “Learning scalable ly=-constrained near-lossless image compression via joint lossy image and residual compression”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 11946–11955.
- [11] R Baker et al. “New relationships between breast microcalcifications and cancer”. In: *British journal of cancer* 103.7 (2010), pp. 1034–1039.
- [12] Ashwin Belle et al. “Big data analytics in healthcare”. In: *BioMed research international* 2015 (2015).
- [13] Gema Bello-Orgaz, Jason J Jung, and David Camacho. “Social big data: Recent achievements and new challenges”. In: *Information Fusion* 28 (2016), pp. 45–59.
- [14] Yoshua Bengio, Aaron Courville, and Pascal Vincent. “Representation learning: A review and new perspectives”. In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.
- [15] Goutam Bhat et al. “Deep burst super-resolution”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 9209–9218.

- [16] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.
- [17] David M Blei, Andrew Y Ng, and Michael I Jordan. “Latent dirichlet allocation”. In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022.
- [18] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. “Yolov4: Optimal speed and accuracy of object detection”. In: *arXiv preprint arXiv:2004.10934* (2020).
- [19] Christian Brandli et al. “A $240 \times 180 \times 130$ db $3 \mu\text{s}$ latency global shutter spatiotemporal vision sensor”. In: *IEEE Journal of Solid-State Circuits* 49.10 (2014), pp. 2333–2341.
- [20] Felix Brockherde et al. “Bypassing the Kohn-Sham equations with machine learning”. In: *Nature communications* 8.1 (2017), pp. 1–10.
- [21] Keith T Butler et al. “Machine learning for molecular and materials science”. In: *Nature* 559.7715 (2018), pp. 547–555.
- [22] Olivier Canévet and François Fleuret. “Efficient sample mining for object detection”. In: *Asian Conference on Machine Learning*. PMLR. 2015, pp. 48–63.
- [23] Matthew R Carbone et al. “Classification of local chemical environments from x-ray absorption spectra using supervised machine learning”. In: *Physical Review Materials* 3.3 (2019), p. 033604.
- [24] Juan Carrasquilla and Roger G Melko. “Machine learning phases of matter”. In: *Nature Physics* 13.5 (2017), pp. 431–434.

- [25] Junyi Chai et al. “Deep learning in computer vision: A critical review of emerging techniques and application scenarios”. In: *Machine Learning with Applications 6* (2021), p. 100134.
- [26] MARIA CHAN et al. *EXSCLAIM!* Tech. rep. Argonne National Lab.(ANL), Argonne, IL (United States), 2022.
- [27] Ravini U Chandrasena et al. “Strain-engineered oxygen vacancies in CaMnO₃ thin films”. In: *Nano letters 17.2* (2017), pp. 794–799.
- [28] Gal Chechik et al. “Large Scale Online Learning of Image Similarity Through Ranking.” In: *Journal of Machine Learning Research 11.3* (2010).
- [29] Chenyi Chen et al. “Deepdriving: Learning affordance for direct perception in autonomous driving”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2722–2730.
- [30] Dahong Chen et al. “Facile hydrothermal synthesis of AgPb₁₀LaTe₁₂ materials: controlled synthesis, growth mechanism and shape-dependent electrical transportation properties”. In: *Crytengcomm 14.22* (2012), pp. 7771–7779.
- [31] Kai Chen et al. *MMDetection: Open MMLab Detection Toolbox and Benchmark*. 2019. arXiv: 1906.07155 [cs.CV].
- [32] Liang-Chieh Chen et al. *DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*. 2017. arXiv: 1606.00915 [cs.CV].
- [33] Liang-Chieh Chen et al. “Encoder-decoder with atrous separable convolution for semantic image segmentation”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 801–818.

- [34] Ting Chen et al. “A simple framework for contrastive learning of visual representations”. In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.
- [35] Wei Chen et al. “Deep image retrieval: A survey”. In: *ArXiv* (2021).
- [36] Yiming Chen et al. “Database of ab initio L-edge X-ray absorption near edge structure”. In: *Scientific data* 8.1 (2021), pp. 1–8.
- [37] Zhanzhan Cheng et al. “Focusing attention: Towards accurate text recognition in natural images”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 5076–5084.
- [38] Sumit Chopra, Raia Hadsell, and Yann LeCun. “Learning a similarity metric discriminatively, with application to face verification”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 1. IEEE. 2005, pp. 539–546.
- [39] Kamal Choudhary et al. “Recent advances and applications of deep learning methods in materials science”. In: *npj Computational Materials* 8.1 (2022), pp. 1–26.
- [40] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. “Multi-column deep neural networks for image classification”. In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, pp. 3642–3649.
- [41] Ronan Collobert et al. “Natural language processing (almost) from scratch”. In: *Journal of machine learning research* 12.ARTICLE (2011), pp. 2493–2537.
- [42] Callum J Court and Jacqueline M Cole. “Auto-generated materials database of Curie and Néel temperatures via semi-supervised relationship extraction”. In: *Scientific data* 5.1 (2018), pp. 1–12.

- [43] Anup L Dadlani et al. “ALD Zn (O, S) thin films’ interfacial chemical and structural configuration probed by XAS”. In: *ACS applied materials & interfaces* 8.23 (2016), pp. 14323–14327.
- [44] George Dahl et al. “Phone recognition with the mean-covariance restricted Boltzmann machine”. In: *Advances in neural information processing systems* 23 (2010).
- [45] George E Dahl et al. “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition”. In: *IEEE Transactions on audio, speech, and language processing* 20.1 (2011), pp. 30–42.
- [46] Navneet Dalal and Bill Triggs. “Histograms of oriented gradients for human detection”. In: 2005.
- [47] Hannelore Daniel et al. “High-fat diet alters gut microbiota physiology in mice”. In: *The ISME journal* 8.2 (2014), pp. 295–308.
- [48] Ritendra Datta et al. “Image retrieval: Ideas, influences, and trends of the new age”. In: *ACM Computing Surveys (Csur)* 40.2 (2008), pp. 1–60.
- [49] Bert De Brabandere, Davy Neven, and Luc Van Gool. “Semantic instance segmentation with a discriminative loss function”. In: *arXiv preprint arXiv:1708.02551* (2017).
- [50] Brian L DeCost and Elizabeth A Holm. “A large dataset of synthetic SEM images of powder materials and their ground truth 3D structures”. In: *Data in brief* 9 (2016), pp. 727–731.
- [51] Brian L DeCost et al. “UHCSDB: ultrahigh carbon steel micrograph database”. In: *Integrating Materials and Manufacturing Innovation* 6.2 (2017), pp. 197–205.

- [52] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.
- [53] Carl Doersch, Abhinav Gupta, and Alexei A Efros. “Unsupervised visual representation learning by context prediction”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1422–1430.
- [54] Vincenza Dragone et al. “An autonomous organic reaction search engine for chemical reactivity”. In: *Nature communications* 8.1 (2017), pp. 1–8.
- [55] Clement Farabet et al. “Learning hierarchical features for scene labeling”. In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2012), pp. 1915–1929.
- [56] Mahyar Fazlyab et al. “Efficient and accurate estimation of lipschitz constants for deep neural networks”. In: *Advances in Neural Information Processing Systems* 32 (2019).
- [57] Yuma Fujita et al. “Growth and characterization of ROBiS2 high-entropy superconducting single crystals”. In: *ACS omega* 5.27 (2020), pp. 16819–16825.
- [58] Mengshu Ge et al. “Deep learning analysis on microscopic imaging in materials science”. In: *Materials Today Nano* 11 (2020), p. 100087.
- [59] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. “Unsupervised representation learning by predicting image rotations”. In: *arXiv preprint arXiv:1803.07728* (2018).
- [60] Justin Gilmer et al. “Neural message passing for quantum chemistry”. In: *International conference on machine learning*. PMLR. 2017, pp. 1263–1272.

- [61] Ross Girshick. “Fast r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.
- [62] Ross Girshick et al. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.
- [63] Jean-Bastien Grill et al. “Bootstrap your own latent-a new approach to self-supervised learning”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 21271–21284.
- [64] Yifan Gu et al. “Clustering-driven unsupervised deep hashing for image retrieval”. In: *Neurocomputing* 368 (2019), pp. 114–123.
- [65] Jordan A Hachtel, Juan Carlos Idrobo, and Miaofang Chi. “Sub-Ångstrom electric field measurements on a universal detector in a scanning transmission electron microscope”. In: *Advanced structural and chemical imaging* 4.1 (2018), pp. 1–10.
- [66] Samuel W. Hasinoff et al. “Burst photography for high dynamic range and low-light imaging on mobile cameras”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 35.6 (2016).
- [67] Mahmoud Hassaballah and Ali Ismail Awad. *Deep learning in computer vision: principles and applications*. CRC Press, 2020.
- [68] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. “The elements of statistical learning. Springer series in statistics”. In: *New York, NY, USA* (2001).
- [69] Robin W Havener et al. “Laser-based imaging of individual carbon nanostructures”. In: *NPG Asia Materials* 3.10 (2011), pp. 91–99.

- [70] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.
- [71] Kaiming He et al. “Mask r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.
- [72] Kaiming He et al. “Momentum contrast for unsupervised visual representation learning”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 9729–9738.
- [73] Qian He et al. “Switching-off toluene formation in the solvent-free oxidation of benzyl alcohol using supported trimetallic Au–Pd–Pt nanoparticles”. In: *Faraday discussions* 162 (2013), pp. 365–378.
- [74] Satoshi Hinokuma et al. “Local Structures and Catalytic Ammonia Combustion Properties of Copper Oxides and Silver Supported on Aluminum Oxides”. In: *The Journal of Physical Chemistry C* 121.8 (2017), pp. 4188–4196.
- [75] Geoffrey Hinton et al. “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups”. In: *IEEE Signal processing magazine* 29.6 (2012), pp. 82–97.
- [76] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. “A fast learning algorithm for deep belief nets”. In: *Neural computation* 18.7 (2006), pp. 1527–1554.
- [77] Elad Hoffer and Nir Ailon. “Deep metric learning using triplet network”. In: *International workshop on similarity-based pattern recognition*. Springer. 2015, pp. 84–92.

- [78] Seunghoon Hong, Hyeonwoo Noh, and Bohyung Han. “Decoupled deep neural network for semi-supervised semantic segmentation”. In: *Advances in neural information processing systems* 28 (2015).
- [79] Andrew G Howard et al. “Mobilenets: Efficient convolutional neural networks for mobile vision applications”. In: *arXiv preprint arXiv:1704.04861* (2017).
- [80] Bill Howe et al. “Deep mapping of the visual literature”. In: *Proceedings of the 26th international conference on world wide web companion*. 2017, pp. 1273–1277.
- [81] Chuangang Hu et al. “Small-sized PdCu nanocapsules on 3D graphene for high-performance ethanol oxidation”. In: *Nanoscale* 6.5 (2014), pp. 2768–2775.
- [82] Xian-Sheng Hua and Jin Li. “Prajna: Towards recognizing whatever you want from images without image labeling”. In: *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015.
- [83] Liyuan Huang and Chen Ling. “Practicing deep learning in materials science: An evaluation for predicting the formation energies”. In: *Journal of Applied Physics* 128.12 (2020), p. 124901.
- [84] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *International conference on machine learning*. PMLR. 2015, pp. 448–456.
- [85] Ryan Jacobs et al. “Performance, Successes and Limitations of Deep Learning Semantic Segmentation of Multiple Defects in Transmission Electron Micrographs”. In: *arXiv preprint arXiv:2110.08244* (2021).
- [86] Max Jaderberg et al. “Spatial transformer networks”. In: *arXiv preprint arXiv:1506.02025* (2015).

- [87] Anubhav Jain et al. “Commentary: The Materials Project: A materials genome approach to accelerating materials innovation”. In: *APL materials* 1.1 (2013), p. 011002.
- [88] Herve Jegou, Matthijs Douze, and Cordelia Schmid. “Hamming embedding and weak geometric consistency for large scale image search”. In: *European conference on computer vision*. Springer. 2008, pp. 304–317.
- [89] S Jesse et al. “Big data analytics for scanning transmission electron microscopy ptychography”. In: *Scientific reports* 6.1 (2016), pp. 1–8.
- [90] Weixin Jiang, Yongbing Zhang, and Qionghai Dai. “Parameterized reconstruction based Fourier Ptychography”. In: *2016 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE. 2016, pp. 1–6.
- [91] Weixin Jiang et al. “A hybrid image retrieval system for microscopy images”. In: *Microscopy and Microanalysis* 27.S1 (2021), pp. 474–476.
- [92] Weixin Jiang et al. “A two-stage framework for compound figure separation”. In: *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2021, pp. 1204–1208.
- [93] Weixin Jiang et al. “MaterialEyes: Utilizing literature to characterize materials from images”. In: *Bulletin of the American Physical Society* (2022).
- [94] Weixin Jiang et al. “Plot2Spectra: an Automatic Spectra Extraction Tool”. In: *arXiv preprint arXiv:2107.02827* (2021).
- [95] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. “Perceptual losses for real-time style transfer and super-resolution”. In: *European conference on computer vision*. Springer. 2016, pp. 694–711.

- [96] Surya R Kalidindi and Marc De Graef. “Materials data science: current status and future outlook”. In: *Annual Review of Materials Research* 45 (2015), pp. 171–193.
- [97] Sergei V Kalinin, Bobby G Sumpter, and Richard K Archibald. “Big–deep–smart data in imaging for guiding materials design”. In: *Nature materials* 14.10 (2015), pp. 973–980.
- [98] Saurabh Kataria et al. “Automatic extraction of data points and text blocks from 2-dimensional plots in digital documents.” In: *AAAI*. Vol. 8. 2008, pp. 1169–1174.
- [99] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. “Contrastive representation learning: A framework and review”. In: *IEEE Access* 8 (2020), pp. 193907–193934.
- [100] Prannay Khosla et al. “Supervised contrastive learning”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 18661–18673.
- [101] Hamed Kiani Galoogahi et al. “Need for speed: A benchmark for higher frame rate object tracking”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 1125–1134.
- [102] Dahun Kim et al. “Learning image representations by completing damaged jigsaw puzzles”. In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2018, pp. 793–802.
- [103] Edward Kim et al. “Machine-learned and codified synthesis parameters of oxide materials”. In: *Scientific data* 4.1 (2017), pp. 1–9.
- [104] Edward Kim et al. “Materials synthesis insights from scientific literature via text extraction and machine learning”. In: *Chemistry of Materials* 29.21 (2017), pp. 9436–9444.

- [105] Edward Kim et al. “Virtual screening of inorganic materials synthesis parameters with deep learning”. In: *npj Computational Materials* 3.1 (2017), pp. 1–9.
- [106] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).
- [107] Nahum Kiryati, Yuval Eldar, and Alfred M Bruckstein. “A probabilistic Hough transform”. In: *Pattern recognition* 24.4 (1991), pp. 303–316.
- [108] Walter Kohn and Lu Jeu Sham. “Self-consistent equations including exchange and correlation effects”. In: *Physical review* 140.4A (1965), A1133.
- [109] Shu Kong and Charless C Fowlkes. “Recurrent pixel embedding for instance grouping”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 9018–9028.
- [110] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012).
- [111] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), pp. 436–444.
- [112] Kathy Lee, Ankit Agrawal, and Alok Choudhary. “Forecasting influenza levels using real-time social media streams”. In: *2017 IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE. 2017, pp. 409–414.
- [113] Po-Shen Lee and Bill Howe. “Detecting and dismantling composite visualizations in the scientific literature”. In: *International Conference on Pattern Recognition Applications and Methods*. Springer. 2015, pp. 247–266.

- [114] Zhengchao Lei et al. “Scene text recognition using residual convolutional recurrent neural network”. In: *Machine Vision and Applications* 29.5 (2018), pp. 861–871.
- [115] Hui Li, Peng Wang, and Chunhua Shen. “Towards end-to-end text spotting with convolutional recurrent neural networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 5238–5246.
- [116] Li-Jia Li and Li Fei-Fei. “Optimol: automatic online picture collection via incremental model learning”. In: *International journal of computer vision* 88.2 (2010), pp. 147–168.
- [117] Pengyuan Li et al. “Compound image segmentation of published biomedical figures”. In: *Bioinformatics* 34.7 (2018), pp. 1192–1199.
- [118] Zhenwei Li, James R Kermode, and Alessandro De Vita. “Molecular dynamics with on-the-fly machine learning of quantum-mechanical forces”. In: *Physical review letters* 114.9 (2015), p. 096405.
- [119] Minghui Liao et al. “Rotation-sensitive regression for oriented scene text detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 5909–5918.
- [120] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. “A 128 x 128 120db 30mw asynchronous vision sensor that responds to relative intensity change”. In: *2006 IEEE International Solid State Circuits Conference-Digest of Technical Papers*. IEEE. 2006, pp. 2060–2069.
- [121] Tsung-Yi Lin et al. “Feature pyramid networks for object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125.

- [122] Tsung-Yi Lin et al. “Focal loss for dense object detection”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.
- [123] Bowen Liu et al. “Retrosynthetic reaction prediction using neural sequence-to-sequence models”. In: *ACS central science* 3.10 (2017), pp. 1103–1113.
- [124] Fangyang Liu et al. “Beyond 8% ultrathin kesterite $\text{Cu}_2\text{ZnSnS}_4$ solar cells by interface reaction route controlling and self-organized nanopattern at the back contact”. In: *NPG Asia Materials* 9.7 (2017), e401–e401.
- [125] Shu Liu et al. “Path aggregation network for instance segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8759–8768.
- [126] Wei Liu et al. “Ssd: Single shot multibox detector”. In: *European conference on computer vision*. Springer. 2016, pp. 21–37.
- [127] Xiao Liu et al. “Self-supervised learning: Generative or contrastive”. In: *IEEE Transactions on Knowledge and Data Engineering* (2021).
- [128] Yuliang Liu and Lianwen Jin. “Deep matching prior network: Toward tighter multi-oriented text detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 1962–1969.
- [129] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.
- [130] David G Lowe. “Distinctive image features from scale-invariant keypoints”. In: *International journal of computer vision* 60.2 (2004), pp. 91–110.

- [131] Xiaonan Lu et al. “Automated analysis of images in documents for intelligent document search”. In: *International Journal on Document Analysis and Recognition (IJDAR)* 12.2 (2009), pp. 65–81.
- [132] Junyu Luo et al. “ChartOCR: Data Extraction from Charts Images via a Deep Hybrid Framework”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021, pp. 1917–1925.
- [133] Jacob Madsen et al. “A deep learning approach to identify local structures in atomic-resolution transmission electron microscopy images”. In: *Advanced Theory and Simulations* 1.8 (2018), p. 1800037.
- [134] Artem Maksov et al. “Deep learning analysis of defect and phase evolution during electron beam-induced transformations in WS₂”. In: *npj Computational Materials* 5.1 (2019), pp. 1–8.
- [135] Narbe Mardirossian and Martin Head-Gordon. “ ω B97M-V: A combinatorially optimized, range-separated hybrid, meta-GGA density functional with VV10 nonlocal correlation”. In: *The Journal of chemical physics* 144.21 (2016), p. 214110.
- [136] Kiran Mathew et al. “High-throughput computational X-ray absorption spectroscopy”. In: *Scientific data* 5.1 (2018), pp. 1–8.
- [137] Ziatdinov Maxim et al. “Tracking atomic structure evolution during directed electron beam induced Si-atom motion in graphene via deep machine learning”. In: *Nanotechnology* 32.3 (2020), p. 035703.
- [138] Paul W Melnychuk and Majid Rabbani. “Survey of lossless image coding techniques”. In: *Digital Image Processing Applications*. Vol. 1075. SPIE. 1989, pp. 92–100.

- [139] Tomas Mikolov et al. “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781* (2013).
- [140] Tomáš Mikolov et al. “Strategies for training large scale neural network language models”. In: *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE. 2011, pp. 196–201.
- [141] Tom M Mitchell et al. *Machine learning*. 1997.
- [142] Mohammad Hadi Modarres et al. “Neural network for nanoscience scanning electron microscope image recognition”. In: *Scientific reports* 7.1 (2017), pp. 1–12.
- [143] Elias Mueggler et al. “The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM”. In: *The International Journal of Robotics Research* 36.2 (2017), pp. 142–149.
- [144] Tim Mueller, Aaron Gilad Kusne, and Rampi Ramprasad. “Machine learning in materials science: Recent progress and emerging applications”. In: *Reviews in computational chemistry* 29 (2016), pp. 186–273.
- [145] Karim T Mukaddem et al. “ImageDataExtractor: A Tool to extract and quantify data from microscopy images”. In: *Journal of Chemical Information and Modeling* (2019).
- [146] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [147] Robert F Murphy et al. “Searching online journals for fluorescence microscope images depicting protein subcellular location patterns”. In: *Proceedings 2nd Annual IEEE International Symposium on Bioinformatics and Bioengineering (BIBE 2001)*. IEEE. 2001, pp. 119–128.

- [148] Davy Neven et al. “Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8837–8845.
- [149] Davy Neven et al. “Towards end-to-end lane detection: an instance segmentation approach”. In: *2018 IEEE intelligent vehicles symposium (IV)*. IEEE. 2018, pp. 286–291.
- [150] Alejandro Newell, Zhiao Huang, and Jia Deng. “Associative embedding: End-to-end learning for joint detection and grouping”. In: *arXiv preprint arXiv:1611.05424* (2016).
- [151] Simon Niklaus, Long Mai, and Feng Liu. “Video frame interpolation via adaptive separable convolution”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 261–270.
- [152] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. “Learning deconvolution network for semantic segmentation”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1520–1528.
- [153] Mehdi Noroozi and Paolo Favaro. “Unsupervised learning of visual representations by solving jigsaw puzzles”. In: *European conference on computer vision*. Springer. 2016, pp. 69–84.
- [154] Gregory B Olson. “Computational design of hierarchically structured materials”. In: *Science* 277.5330 (1997), pp. 1237–1242.
- [155] Wee-Jun Ong et al. “Highly reactive {001} facets of TiO₂-based composites: synthesis, formation mechanism and characterization”. In: *Nanoscale* 6.4 (2014), pp. 1946–2008.

- [156] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. “Representation learning with contrastive predictive coding”. In: *arXiv preprint arXiv:1807.03748* (2018).
- [157] Colin Ophus. “A fast image simulation algorithm for scanning transmission electron microscopy”. In: *Advanced structural and chemical imaging* 3.1 (2017), pp. 1–11.
- [158] Liyuan Pan et al. “Bringing a blurry frame alive at high frame-rate with an event camera”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6820–6829.
- [159] Xingang Pan et al. *Spatial As Deep: Spatial CNN for Traffic Scene Understanding*. 2017. arXiv: 1712.06080 [cs.CV].
- [160] James Philbin et al. “Lost in quantization: Improving particular object retrieval in large scale image databases”. In: *2008 IEEE conference on computer vision and pattern recognition*. IEEE. 2008, pp. 1–8.
- [161] James Philbin et al. “Object retrieval with large vocabularies and fast spatial matching”. In: *2007 IEEE conference on computer vision and pattern recognition*. IEEE. 2007, pp. 1–8.
- [162] Pedro O Pinheiro et al. “Learning to refine object segments”. In: *European conference on computer vision*. Springer. 2016, pp. 75–91.
- [163] Alan Pryor, Colin Ophus, and Jianwei Miao. “A streaming multi-GPU implementation of image simulation algorithms for scanning transmission electron microscopy”. In: *Advanced structural and chemical imaging* 3.1 (2017), pp. 1–14.
- [164] Sanyin Qu et al. “Highly anisotropic P3HT films with enhanced thermoelectric performance via organic small molecule epitaxy”. In: *NPG Asia Materials* 8.7 (2016), e292–e292.

- [165] Wullianallur Raghupathi and Viju Raghupathi. “Big data analytics in healthcare: promise and potential”. In: *Health information science and systems* 2.1 (2014), pp. 1–10.
- [166] Krishna Rajan. “Materials informatics: The materials “gene” and big data”. In: *Annual Review of Materials Research* 45 (2015), pp. 153–169.
- [167] Rampi Ramprasad et al. “Machine learning in materials informatics: recent applications and prospects”. In: *npj Computational Materials* 3.1 (2017), pp. 1–13.
- [168] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. “ESIM: an open event camera simulator”. In: *Conference on Robot Learning*. PMLR. 2018, pp. 969–982.
- [169] Joseph Redmon and Ali Farhadi. “YOLO9000: better, faster, stronger”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7263–7271.
- [170] Joseph Redmon and Ali Farhadi. “Yolov3: An incremental improvement”. In: *arXiv preprint arXiv:1804.02767* (2018).
- [171] Joseph Redmon et al. “You only look once: Unified, real-time object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.
- [172] Shaoqing Ren et al. “Faster r-cnn: Towards real-time object detection with region proposal networks”. In: *Advances in neural information processing systems*. 2015, pp. 91–99.
- [173] Nelson Rivera et al. “Chemical speciation of potentially toxic trace metals in coal fly ash associated with the Kingston fly ash spill”. In: *Energy & Fuels* 31.9 (2017), pp. 9652–9659.

- [174] Graham Roberts et al. “Deep learning for semantic segmentation of defects in advanced STEM images of steels”. In: *Scientific reports* 9.1 (2019), pp. 1–12.
- [175] Ankit Rohatgi. *Webplotdigitizer: Version 4.4*. 2020. URL: <https://automeris.io/WebPlotDigitizer>.
- [176] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV].
- [177] Kevin Ryan, Jeff Lengyel, and Michael Shatruk. “Crystal structure prediction via deep learning”. In: *Journal of the American Chemical Society* 140.32 (2018), pp. 10158–10168.
- [178] Tara N Sainath et al. “Improvements to deep convolutional neural networks for LVCSR”. In: *2013 IEEE workshop on automatic speech recognition and understanding*. IEEE. 2013, pp. 315–320.
- [179] E Saitoh et al. “Observation of orbital waves as elementary excitations in a solid”. In: *Nature* 410.6825 (2001), pp. 180–183.
- [180] Tim Salimans et al. “Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications”. In: *arXiv preprint arXiv:1701.05517* (2017).
- [181] Meenakshisundaram Sankar et al. “Supported bimetallic nano-alloys as highly active catalysts for the one-pot tandem synthesis of imines and secondary amines from nitrobenzene and alcohols”. In: *Catalysis Science & Technology* 6.14 (2016), pp. 5473–5482.

- [182] Cedric Scheerlinck, Nick Barnes, and Robert Mahony. “Continuous-time intensity estimation using event cameras”. In: *Asian Conference on Computer Vision*. Springer. 2018, pp. 308–324.
- [183] Jonathan Schmidt et al. “Recent advances and applications of machine learning in solid-state materials science”. In: *npj Computational Materials* 5.1 (2019), pp. 1–36.
- [184] Florian Schroff, Antonio Criminisi, and Andrew Zisserman. “Harvesting image databases from the web”. In: *IEEE transactions on pattern analysis and machine intelligence* 33.4 (2010), pp. 754–766.
- [185] Eric Schwenker et al. “Constructing Self-Labeled Materials Imaging Datasets from Open Access Scientific Journals with EXSCLAIM!” In: *Microscopy and Microanalysis* 26.S2 (2020), pp. 3096–3097.
- [186] Eric Schwenker et al. “EXSCLAIM!—An automated pipeline for the construction of labeled materials imaging datasets from literature”. In: *arXiv preprint arXiv:2103.10631* (2021).
- [187] Marwin HS Segler, Mike Preuss, and Mark P Waller. “Planning chemical syntheses with deep neural networks and symbolic AI”. In: *Nature* 555.7698 (2018), pp. 604–610.
- [188] Pierre Sermanet et al. “Overfeat: Integrated recognition, localization and detection using convolutional networks”. In: *arXiv preprint arXiv:1312.6229* (2013).
- [189] Pierre Sermanet et al. “Time-contrastive networks: Self-supervised learning from video”. In: *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, pp. 1134–1141.

- [190] Mingren Shen. “Machine Learning Applications in Material Science Problems”. PhD thesis. The University of Wisconsin-Madison, 2021.
- [191] Mingren Shen et al. “A deep learning based automatic defect analysis framework for In-situ TEM ion irradiations”. In: *Computational Materials Science* 197 (2021), p. 110560.
- [192] Mingren Shen et al. “Multi defect detection and analysis of electron microscopy images with deep learning”. In: *Computational Materials Science* 199 (2021), p. 110576.
- [193] Yuming Shen et al. “Auto-encoding twin-bottleneck hashing”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 2818–2827.
- [194] Geeta Shetty et al. “Raman spectroscopy: elucidation of biochemical changes in carcinogenesis of oesophagus”. In: *British journal of cancer* 94.10 (2006), pp. 1460–1464.
- [195] Baoguang Shi, Xiang Bai, and Serge Belongie. “Detecting oriented text in natural images by linking segments”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2550–2558.
- [196] Baoguang Shi, Xiang Bai, and Cong Yao. “An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition”. In: *IEEE transactions on pattern analysis and machine intelligence* 39.11 (2016), pp. 2298–2304.
- [197] Baoguang Shi et al. “Robust scene text recognition with automatic rectification”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 4168–4176.

- [198] Xiangyang Shi et al. “Layout-aware Subfigure Decomposition for Complex Figures in the Biomedical Literature”. In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 1343–1347.
- [199] David Silver et al. “Mastering the game of Go with deep neural networks and tree search”. In: *nature* 529.7587 (2016), pp. 484–489.
- [200] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).
- [201] AA Sirenko et al. “Soft-mode hardening in SrTiO₃ thin films”. In: *Nature* 404.6776 (2000), pp. 373–376.
- [202] Justin S Smith, Olexandr Isayev, and Adrian E Roitberg. “ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost”. In: *Chemical science* 8.4 (2017), pp. 3192–3203.
- [203] Pengming Song et al. “Fourier ptychographic reconstruction using weighted replacement in the fourier domain”. In: *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2016, pp. 3171–3175.
- [204] Robin Strudel et al. “Segmenter: Transformer for semantic segmentation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 7262–7272.
- [205] Chen Sun et al. “Revisiting unreasonable effectiveness of data in deep learning era”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 843–852.

- [206] Matthew C Swain and Jacqueline M Cole. “ChemDataExtractor: a toolkit for automated extraction of chemical information from the scientific literature”. In: *Journal of chemical information and modeling* 56.10 (2016), pp. 1894–1904.
- [207] Christian Szegedy et al. “Going deeper with convolutions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.
- [208] Yaniv Taigman et al. “Deepface: Closing the gap to human-level performance in face verification”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 1701–1708.
- [209] Mingxing Tan and Quoc Le. “Efficientnet: Rethinking model scaling for convolutional neural networks”. In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.
- [210] Mario Taschwer and Oge Marques. “Automatic separation of compound figures in scientific articles”. In: *Multimedia Tools and Applications* 77.1 (2018), pp. 519–548.
- [211] SK Teh et al. “Diagnostic potential of near-infrared Raman spectroscopy in the stomach: differentiating dysplasia from normal tissue”. In: *British journal of cancer* 98.2 (2008), pp. 457–465.
- [212] Yonglong Tian, Dilip Krishnan, and Phillip Isola. “Contrastive multiview coding”. In: *European conference on computer vision*. Springer. 2020, pp. 776–794.
- [213] Zhi Tian et al. “Fcos: Fully convolutional one-stage object detection”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9627–9636.

- [214] Jonathan Tompson et al. “Efficient object localization using convolutional networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 648–656.
- [215] Jonathan J Tompson et al. “Joint training of a convolutional network and a graphical model for human pose estimation”. In: *Advances in neural information processing systems 27* (2014).
- [216] Steven B Torrisi et al. “Random forest machine learning models for interpretable X-ray absorption near-edge structure spectrum-property relationships”. In: *npj Computational Materials* 6.1 (2020), pp. 1–11.
- [217] Vahe Tshitoyan et al. “Unsupervised word embeddings capture latent knowledge from materials science literature”. In: *Nature* 571.7763 (2019), pp. 95–98.
- [218] Satoshi Tsutsui and David J Crandall. “A data driven approach for compound figure separation using convolutional neural networks”. In: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. Vol. 1. IEEE. 2017, pp. 533–540.
- [219] Maria Tzelepi and Anastasios Tefas. “Deep convolutional image retrieval: A general framework”. In: *Signal Processing: Image Communication* 63 (2018), pp. 30–43.
- [220] Jasper RR Uijlings et al. “Selective search for object recognition”. In: *International journal of computer vision* 104.2 (2013), pp. 154–171.
- [221] Aaron Van den Oord et al. “Conditional image generation with pixelcnn decoders”. In: *Advances in neural information processing systems 29* (2016).
- [222] Laurens Van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE.” In: *Journal of machine learning research* 9.11 (2008).

- [223] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. “Pixel recurrent neural networks”. In: *International conference on machine learning*. PMLR. 2016, pp. 1747–1756.
- [224] Rama K Vasudevan et al. “Mapping mesoscopic phase evolution during E-beam induced transformations via deep learning of atomically resolved images”. In: *npj Computational Materials* 4.1 (2018), pp. 1–9.
- [225] Aladin Virmaux and Kevin Scaman. “Lipschitz regularity of deep neural networks: analysis and efficient estimation”. In: *Advances in Neural Information Processing Systems* 31 (2018).
- [226] Athanasios Voulodimos et al. “Deep learning for computer vision: A brief review”. In: *Computational intelligence and neuroscience 2018* (2018).
- [227] Kentaro Wada. *labelme: Image Polygonal Annotation with Python*. <https://github.com/wkentaro/labelme>. 2016.
- [228] Jiang Wang et al. “Learning fine-grained image similarity with deep ranking”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 1386–1393.
- [229] Jiaqi Wang et al. “Region proposal by guided anchoring”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2965–2974.
- [230] Shuai Wang et al. “Nearly atomic precise gold nanoclusters on nickel-based layered double hydroxides for extraordinarily efficient aerobic oxidation of alcohols”. In: *Catalysis Science & Technology* 6.12 (2016), pp. 4090–4104.

- [231] Zihao W Wang et al. “Event-driven video frame synthesis”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 2019, pp. 0–0.
- [232] Marcelo J Weinberger, Gadiel Seroussi, and Guillermo Sapiro. “The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS”. In: *IEEE Transactions on Image processing* 9.8 (2000), pp. 1309–1324.
- [233] Jie Xu, Angus P Wilkinson, and Sidhartha Pattanaik. “Solution Processing of Calcium Zirconate Titanates, Ca (Zr x Ti_{1-x}) O₃: An X-ray Absorption Spectroscopy and Powder Diffraction Study”. In: *Chemistry of materials* 12.11 (2000), pp. 3321–3330.
- [234] Junqi Xu et al. “Fabrication of vertically aligned single-crystalline lanthanum hexaboride nanowire arrays and investigation of their field emission”. In: *NPG Asia Materials* 5.7 (2013), e53–e53.
- [235] Zhilin Yang et al. “Xlnet: Generalized autoregressive pretraining for language understanding”. In: *Advances in neural information processing systems* 32 (2019).
- [236] Yazhou Yao et al. “Towards automatic construction of diverse, high-quality image datasets”. In: *IEEE Transactions on Knowledge and Data Engineering* 32.6 (2019), pp. 1199–1211.
- [237] Jiahui Yu et al. “Unitbox: An advanced object detection network”. In: *Proceedings of the 24th ACM international conference on Multimedia*. 2016, pp. 516–520.
- [238] Minghao Yu et al. “Scalable self-growth of Ni@ NiO core-shell electrode with ultrahigh capacitance and super-long cyclic stability for supercapacitors”. In: *NPG Asia materials* 6.9 (2014), e129–e129.

- [239] Xiaohui Yuan and Dongyu Ang. “A novel figure panel classification and extraction method for document image understanding”. In: *International journal of data mining and bioinformatics* 9.1 (2014), pp. 22–36.
- [240] Erhuan Zhang et al. “Hollow anisotropic semiconductor nanoprisms with highly crystalline frameworks for high-efficiency photoelectrochemical water splitting”. In: *Journal of materials chemistry A* 7.14 (2019), pp. 8061–8072.
- [241] Kai Zhang, Wangmeng Zuo, and Lei Zhang. “FFDNet: Toward a fast and flexible solution for CNN-based image denoising”. In: *IEEE Transactions on Image Processing* 27.9 (2018), pp. 4608–4622.
- [242] Kai Zhang et al. “Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising”. In: *IEEE transactions on image processing* 26.7 (2017), pp. 3142–3155.
- [243] Xiaosong Zhang et al. “Learning to match anchors for visual object detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).
- [244] Yihua Zhang, Ian J Drake, and Alexis T Bell. “Characterization of Cu-ZSM-5 prepared by solid-state ion exchange of H-ZSM-5 with CuCl”. In: *Chemistry of materials* 18.9 (2006), pp. 2347–2356.
- [245] Yongbing Zhang, Weixin Jiang, and Qionghai Dai. “Nonlinear optimization approach for Fourier ptychographic microscopy”. In: *Optics Express* 23.26 (2015), pp. 33822–33835.
- [246] Yongbing Zhang et al. “Self-learning based Fourier ptychographic microscopy”. In: *Optics express* 23.14 (2015), pp. 18471–18486.

- [247] Chen Zheng et al. “Automated generation and ensemble-learned matching of X-ray absorption spectra”. In: *npj Computational Materials* 4.1 (2018), pp. 1–9.
- [248] Liang Zheng, Yi Yang, and Qi Tian. “SIFT meets CNN: A decade survey of instance retrieval”. In: *IEEE transactions on pattern analysis and machine intelligence* 40.5 (2017), pp. 1224–1244.
- [249] Sixiao Zheng et al. “Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 6881–6890.
- [250] Yanzhao Zhou et al. “Oriented response networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 519–528.
- [251] Maxim Ziatdinov et al. “Atomic mechanisms for the Si atom dynamics in graphene: chemical transformations at the edge and in the bulk”. In: *Advanced Functional Materials* 29.52 (2019), p. 1904480.
- [252] Maxim Ziatdinov et al. “Deep analytics of atomically-resolved images: manifest and latent features”. In: *arXiv preprint arXiv:1801.05133* (2018).
- [253] Dongmian Zou, Radu Balan, and Maneesh Singh. “On lipschitz bounds of general convolutional neural networks”. In: *IEEE Transactions on Information Theory* 66.3 (2019), pp. 1738–1759.