

# A MACHINE LEARNING APPROACH TO DETECT AND CLASSIFY 3D TWO-PHOTON POLYMERIZATION MICROSTRUCTURES USING OPTICAL MICROSCOPY IMAGES

Israel Goytom and Gu Yinwei

Department of Microelectronics Science and Engineering,  
Faculty of Science, Ningbo University, Ningbo,315211,China

## **ABSTRACT**

*For 3D microstructures fabricated by two-photon polymerization, a practical approach of machine learning for detection and classification in their optical microscopic images is state and demonstrated in this paper. It is based on Faster R-CNN, Multi-label classification (MLC) and Residual learning framework Algorithms for reliable, automated detection and accurate labeling of Two Photo Polymerization (TPP) microstructures. From finding and detecting the microstructures from a different location in the microscope slide, matching different shapes of the microstructures classify them among their categories is fully automated. The results are compared with manual examination and SEM images of the microstructures for the accuracy test. Some modifications of ordinary optical Microscope so as to make it automated and by applying Deep learning and Image processing algorithms we can successfully detect, label and classify 3D microstructures, designing the neural network model for each phase and by training them using the datasets we have made, the dataset is a set of different images from different angles and their annotation we can achieve high accuracy. The accurate microstructure detection technique in the combination of image processing and computer vision help to simulate the values of each pixel and classify the Microstructures.*

## **KEYWORDS**

*Multi-label classification, Faster R-CNN Two-Photon Polymerization, computer vision, 3D Microstructures*

## **1. INTRODUCTION**

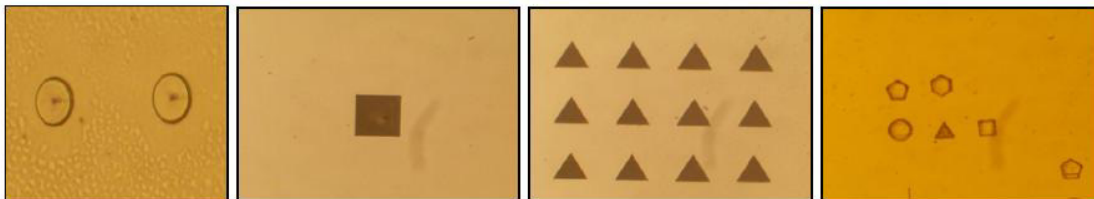
In the last two decades, two-photon polymerization (TPP) has been established as a versatile tool for the microfabrication of three-dimensional structures [1]. Applications include the fabrication of photonic crystals [2], directly printed lens systems [3], micro fluidic devices[3], biological scaffolds[4] and templates for metamaterials[5]. In general, TPP relies on similar principles as known for common optical lithography, namely, exposure of an often-negative photo resist and a wet development process followed by the drying of the structures. Most of the 3D conductive patterns in the literature are fabricated using a layer-by-layer strategy with planar lithography techniques. On the other side to see and study the 2D microstructures we can use the optical microscope while the 3D microstructures see by either 3D microscopes or scanning electron microscope (SEM). Microscopes are a widely used optical instrument at this stage optical microscopes can amplify the objects up-to Micron size. It amplifies the observed objects through the optical system and displays the microstructure to recognize and study the characteristics of the objects from the microform. Currently, it has been widely used in biology, pathology, cell

histology, genetics, clinical diagnosis, materials testing, aviation and space technology, geology and archeology, Electronic components performance testing and analysis and other fields. A traditional microscope looks at the sample from one specific direction and acquires a two-dimensional (2D) projection of the sample in that direction. Although the information collected by a conventional microscope is beneficial in understanding the microstructures of the sample, under many circumstances 2D information alone is not enough or even confusing. In recent decades deep learning techniques are being used for all different purposes with great success and are becoming more popular within various disciplines. Because of its generality, similar architectures put together through deep learning can be applied to many classification problems. Within the given amount of data, they are increasingly being used as a tool for multi-label classification [6]. Applying Deep learning algorithms for Detection of Microstructures results in efficiency, fast and accurate. Separating any structures in the Microscope slides is an outcome of image processing like thresholding and edge detection. On the machine learning side, there is a pre-training phase because there is no previous dataset available for this area we need a pre-training phase. In the pre-training phase, we need to collect images for the dataset and labeling them manually; this will help us to make datasets for Faster R-CNN and MLC. After we build the dataset let's review the Faster R-CNN architecture, along with its earlier variants, by Girshick et al. [7]– [9]. The R-CNN architecture has gone under a few iterations and improvements, but with the latest Faster R-CNN architecture, we can train end-to-end deep learning object detectors. The architecture itself includes four primary components. The first component is the base network (i.e., ResNet[10], VGGNet[11], etc.) which is used as a feature extractor. We then have the Region Proposal Network (RPN), which accepts a set of anchors, and outputs proposals as to where it thinks objects are in an image. Because RPN does not know what the object is in the image, rather a potential object exists at a given location. The region of Interest Pooling is used to extract feature maps from each proposal region. Finally, a Region-based Convolutional Neural Network is used to obtain the final class label predictions for the proposal and further refine the proposal locations for better accuracy. Given a large number of moving parts in the R-CNN architecture, we didn't want to implement the entire architecture by hand, Instead, it's we use the TensorFlow Object Detection API [12] the TensorFlow Object Detection API is an open source framework built on top of TensorFlow that makes it easy to construct, train and deploy object detection models. After training the detection model, train our model using multi labeling algorithm[6] which is useful for classifying different types of samples. By using our trained models, we can detect samples from the slide, matching the sample and classify the sample. Image processing and Deep learning integrated when it comes to counting the pixels with edges and later on make the 3D Model simulation or 3D construction. Making the dataset is collecting images and labeling them, to collect the images we use mini servo motor attached to microscope knobs (mechanically modified) and take sequenced images, labeling the collection and use them as a separate file in one compression. Preparing the dataset images for training our classifier network contains reshaping the layers to 2D array form, the use of reshaped layers is to change the shape of the data back into a 2D image for the convolutional layer. For convolutional layers, we need to select how many filters, and the size of the convolution area like too small or too large and we won't be able to obtain interesting features. The dropout layers, the dropout rate is also important to balance learning and overfitting. Each input is represented as 64×64 grayscale image; we will simply resize our 640×480 images to 64×64 using Open CV resize function or any other like Pillow's resize. These resized grayscale pixel intensities are unsigned integers, with the values of the pixels falling in the range [0, 255]. All digits are placed on a black background with a light foreground (i.e., the sample itself) being white and various shades of gray. Modification of the Microscope was done by mechanically attaching servo and stepper motors to the Microscope knobs. The servo controlling mechanism is, within a given range of the rotation  $\mathbf{R}$ . If the range of the servo motor is from 0 to 90 degrees. To get the pixel values of the sample in the Z direction by giving the height of the structure and range factor is calculated from the angle difference of the maximum angle. With each and given T (where T is the time used to rotate the servo motor by one angle) there will be X frames which will be saved as new frame arrays.

## 2. RESULTS

### 2.1. Our Dataset:

We have collected around ~10,000 positive images from each of five different types of microstructures. We added mini servo motor to the focus knobs (Fine Adjustment) of the microscope, by rotating the servo motor in given angle  $T$  we can get a unique image feature from the sample. Applying traditional image processing techniques thresholding, edge detection and Gaussian smoothing techniques used to binarize and augmentation. Our dataset contains images and label file which includes file name, height, type, and xMins, xMaxs, yMins, and yMaxs. The xMins, xMaxs, yMins, and yMaxs will store the (x; y)-coordinates for our bounding boxes, respectively. The Labels list is a list of human-readable class labels for each bounding box. Similarly, we have classes, a list of integer IDs for each class label. We make a constructor which merely performing a series of initializations from our dataset, then export the constructor values as output and make them annotation file for training.



**Figure 1.** Different types of Microstructures fabricated using TPP

### 2.2. Our Model:

We have prepared two models for detection and classification. We use Tensorflow object detection-API for detection and Multi-label classification for a class classifier. The F-RCNN we use in this paper has four stages; Input an Image, Extract region proposals (regions of the image that potentially contain objects) by using selective search, use transfer learning for feature extraction to compute features for each proposal (effectively an ROI) using pre-trained CNN and classify each proposal using the extracted features with a support vector machine (SVM). The architecture is end-to-end trainable, and the complete object detection pipeline takes place inside the network this includes Regional Proposal, Feature extraction, computing the bounding box coordinates of the box. For the training we separate our dataset 80 % for training and 20 % for testing, we use ResNet-101 as base network (feature extractor), then our RPN which accepts anchors and outputs proposals, here we know that our RPN doesn't know what the object is in the image, our RoI (region of interest) pooling is used to extract feature maps from each proposal region, and finally a Region based CNN is used to obtain the final class label predictions for the proposal and further refine the proposal location for better accuracy. We have trained using NVIDIA GTX 1080ti GPU, and by the time we reach step 20,000 the loss value is a ~ 0.03 with 97 % mAP. We test the model on NVIDIA Jetson TX2 for autonomous mode, and we have reached 9 FPS (frame per second) speed for real-time detection.

### 2.3. Train Our Model:

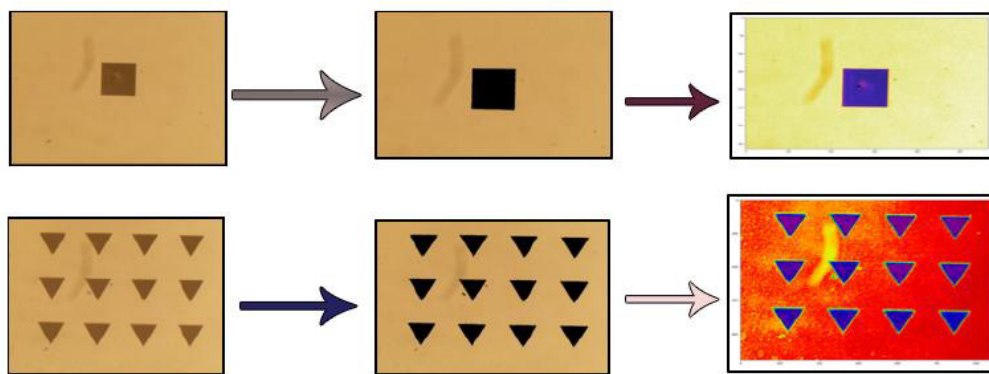
After Design the Network structure using Multi-label classification and Residual learning framework we trained our model using our dataset using Keras for classifier and TFOD API for detection. For the classifier, we used the images containing our Sample gathered using previous Technique and randomly 20000~ Images that do not include our samples. Reshaping each image to get the filtered data is the preprocessing of the Program. The model has a similar structure with

Le-net. We have classified our dataset as 80% training and 20% test data. For detection, we work on Tensorflow object detection API. When working with the TFOD API, we need to re-build a dataset consisting of both the images and their associated bounding boxes. However, before we can get to building the dataset, we need to consider

what makes up "data point" for object detection? According to the TFOD API, we need to supply some attributes, including:

- The TensorFlow-encoded image
- The width and height of the image
- The file encoding of the image (i.e., JPG, PNG, etc.)
- The filename
- A list of bounding box coordinates, normalized in the range [0;1], for the image
- A list of class labels for each bounding box

We make a constructor which merely performing a series of initializations. The xMins, xMaxs, yMins, and yMaxs will store the (x; y)-coordinates for our bounding boxes, respectively. The textLabels list is a list of human-readable class labels for each bounding box. Similarly, we have classes, a list of integer IDs for each class label. Then save export the constructor values as output and make them annotation file for training.



**Figure 2.** Making Roi and Define Each edge for Training. *Left* Input Images *Middle* Encoding Image's Edges *Right* Getting The xMins, xMaxs, yMins, and yMaxs from (x:y) coordinates

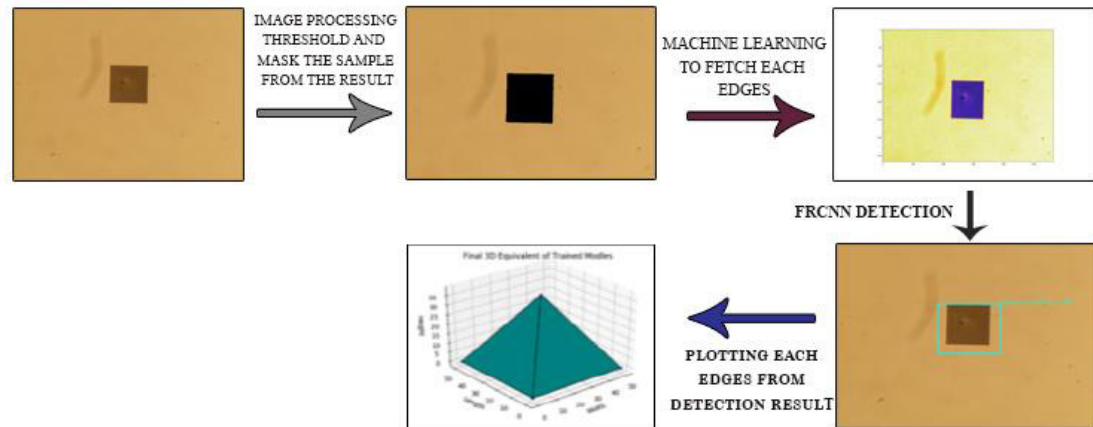
#### Training Network Architecture steps:

1. Input an image
2. Extract regions proposals (i.e., regions of the image that potentially contain objects) using an algorithm such as Selective Search [13].
3. Use transfer learning, specifically feature extraction, to compute features for each proposal (which is an effectively an ROI) using the pre-trained CNN.
4. Classify each proposal using the extracted features with a Support Vector Machine (SVM).

#### 2.4. Test Our Model:

To test our models, we have to handle scaling our image to the range [0, 1], converting it to an array and adding an extra dimension. As we train/, classify images in batches with CNNs. Adding an extra dimension to the array using NumPy arrays which allows our image to have the shape that we want. All our models succeeded with detecting Samples and Non-Samples and

classification of the sample. Detecting the microstructures by accepting the thresholding and masked images, export the detected microstructures then train the classifier to classify under desired Samples label category. Identifying (classifying) type of the sample from the category list and focusing to each unique edge to draw their equivalent 3D model all required training using labeling uniquely from single input has been done under testing the model.



**Figure 3.** The image analysis protocol with tensor-based analysis and edge estimation for detection and classification of the Microstructures.

## 2.5. Finding the Sample Using CNN and CV (The Automated)

After we put our slide on the microscope, the program will start to find the sample by controlling the stepper motors using Arduino when the Arduino is connected and managed by the Machine learning based scripts running on the Nvidia Jetson TX. We use computer vision to differ the plane and any blobs. When there is a blob on the slide, the motion is paused to know the blobs are either dirty or real sample, to analyze this we use our trained model if the detected object is our sample we will save the location into NumPy arrays for later analysis. This process is continuous from the slice  $X_0, Y_0, Z_0$  to  $X_n, Y_n, Z_n$ .

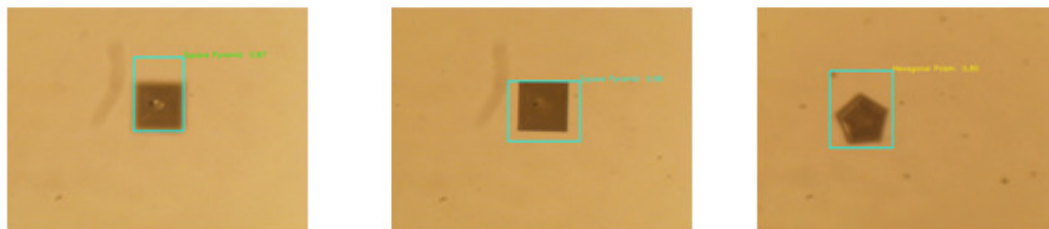
## 2.6. Detecting the structures From the Slice:

By using one of the Image Processing technique thresholding [14], we can apart the image into structures and background. Image thresholding [14] is yet effective way of partitioning an image into a foreground and background. This image analysis technique is a type of image segmentation that isolates objects by converting grayscale images into binary images. By thresholding, we can differentiate any structures in the slide. Thresholding image will make our detection more accurate; many Deep Learning techniques use the specific location of the object they want to detect to Train their Neural Network [15] in Fig (5 (b)) We applied different types of thresholding methods [16] according to their effectiveness on different edges. By thresholding, we can speed up out detector network because our detector network won't check every object in given frame.



**Figure 4.** (a) Input Original Image Adaptive

(b) Thresholding Input image using Global, Mean, and Adaptive Gaussian thresholding.



**Figure 5.** Detection Result using Faster R-CNN

### Augmentation

As we have a limited number of Positive data and our Classifier and detector network big data for better accuracy, we need to augment our input images for the Augmentation of our simple augmented data generator. The augmentation code state as follows.

```
aug=ImageDataGenerator(rotation_range=30,width_shift_range=0.1,height_shift_range=0.1,shear_range=0.2,zoom_range=0.2,horizontal_flip=True,fill_mode="nearest")
```

### 3. CONCLUSION AND FUTURE WORKS:

We have performed an experiment of different types of Two-Photon Polymerization (TPP) Microstructures and detected them from their Optical Microscopy real-time Image and classify their 3D structure type. The main aspects that influence the speed and accuracy of the object detectors methods that we have been used are the complexity of the designed structures like many dimensional images, after all, we compared the results with SEM images, and manual examination the results are almost similar with 25 % faster detection speed in dense slides. We hope this will help practitioners and Researchers choose an appropriate AI method for detection. We have also identified some new techniques for classification without sacrificing much accuracy. Auto focusing to the Widest Edge which will help to see the exact structure at a given time and 3D plotting from their 2D input which will help to get the 3D image in matplotlib 3D axis. This work is available at <https://github.com/isrugeek/2dto3d/>

### Acknowledgment:

Ningbo University Faculty of Science supported this work, and we would like to acknowledge Professor Zhou for Nvidia GPU and Jetson TX2 support, Dr. Lin Dong Dong and Professor Weidong Tao for reviewing our Paper before Submission.

## Author Contributions

Israel Goytom developed the machine learning algorithms, performed the image analyses, and wrote the manuscript. Gu Yinwei, did the TPP experiments, preparing the 3D structure, observed the microstructures, performed the SEM image analysis, making up the device in machine mechanical part and wrote the manuscript. The manuscript was written up through contributions of all authors. All authors have approved the final version of the manuscript.

## Competing Interests

The author(s) declare no competing interests

## REFERENCES

- [1] K. S. Lee, R. H. Kim, D. Y. Yang, and S. H. Park, "Advances in 3D nano/microfabrication using two-photon initiated polymerization," *Progress in Polymer Science (Oxford)*, vol. 33, no. 6. pp. 631–681, 2008.
- [2] M. Rybin et al., "Band Structure of Photonic Crystals Fabricated by Two-Photon Polymerization," *Crystals*, vol. 5, no. 1, pp. 61–73, 2015.
- [3] S. Thiele, K. Arzenbacher, T. Gissibl, H. Giessen, and A. M. Herkommer, "3D-printed eagle eye: Compound microlens system for foveated imaging," *Sci. Adv.*, vol. 3, no. 2, 2017.
- [4] L. Valdevit and J. Bauer, "Fabrication of 3D Micro-Architected/Nano-Architected Materials," in *Three-Dimensional Microfabrication Using Two-Photon Polymerization: Fundamentals, Technology, and Applications*, 2015, pp. 345–373.
- [5] X. Zheng et al., "Multiscale metallic metamaterials," *Nat. Mater.*, vol. 15, no. 10, pp. 1100–1106, Oct. 2016.
- [6] J. Read and F. Perez-Cruz, "Deep Learning for Multi-label Classification," *arXiv Prepr. arXiv1502.05988*, pp. 1–8, 2014.
- [7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 580–587, 2014.
- [8] K. Buhler et al., "Fast R-CNN," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2015 Inter, no. Voc 2012, pp. 580–587, 2015.
- [9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [10] S. Wu, S. Zhong, and Y. Liu, "Deep residual learning for image steganalysis," *Multimed. Tools Appl.*, pp. 1–17, 2017.
- [11] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," pp. 1–14, 2014.
- [12] J. Huang et al., "Speed/accuracy trade-offs for modern convolutional object detectors," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017–Janua, pp. 3296–3305, 2017.
- [13] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, "Selective Search for Object Recognition," 2012.

- [14] H. K. A. Devi, "Thresholding: A Pixel-Level Image Processing Methodology Preprocessing Technique for an OCR System for the Brahmi Script," *Anc. Asia*, vol. 1, no. 0, p. 161, Dec. 2006.
- [15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks."
- [16] P. K. Sahoo, S. Soltani, and A. K. C. Wong, "A survey of thresholding techniques," *Computer Vision, Graphics and Image Processing*, vol. 41, no. 2, pp. 233–260, 1988.

### **Additional Resources**

1. [https://github.com/tensorflow/models/blob/master/research/object\\_detection](https://github.com/tensorflow/models/blob/master/research/object_detection)
2. [https://docs.opencv.org/2.4/modules/imgproc/doc/geometric\\_transformations.html?highlight=resize#cv2.resize](https://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html?highlight=resize#cv2.resize)
3. <https://pillow.readthedocs.io/en/5.2.x/releasenotes/2.7.0.html?highlight=resize>
4. <https://www.nvidia.com/zh-cn/autonomous-machines/embedded-systems-dev-kits-modules/>

### **Authors**

#### **Israel Goytom**

Israel Goytom is Degree Student with Microelectronics Science and Engineering Major in Ningbo University. My research areas are an application of Deep learning and Computer Vision in Physics, Optics and Robotics.



#### **Gu Yinwei**

I am a Master's Degree student of the Faculty of Science in Ningbo University. My research areas are Two-photon microstructure processing.

