

CS229 Final Report:

Bismuth Vanadate (111) Facet Detection

Zixi Liu, Wanling Liu, Jiyao Yuan
{liu1322,liuwl,yuan999} @ stanford.edu

December, 13, 2018

Abstract— Solar energy conversion using Bismuth Vanadate ($BiVO_4$) has been a promising and prevalent method nowadays. To equip efficient energy conversion, existence of (111) facet is crucial. In our project, we utilized three different machine learning models to detect (111) facets in scanning electron microscope (SEM) images. These models include support vector machine (SVM) with histogram of oriented gradient (HOG), Residual Network (ResNet) and Shallow convolutional neural network (Shallow CNN). In this paper, we will talk about architectures, experiments and results of these models. Moreover, data preprocessing such as data augmentation, edge detection and feature extraction will also be included.

1 Introduction

Nowadays, solving the energy challenge by harvesting the energy directly from sunlight through photosynthesis has become an attractive way. Many efforts have been made to find materials that can output chemical fuels using solar energy. Recently, $BiVO_4$ came out as the most promising material for solar energy conversion and environmental protection. The chemical reaction of $BiVO_4$ can be used as a photoanode that oxidizes solar water to O_2 in the photoelectrochemical cells.¹ (see Fig.2)

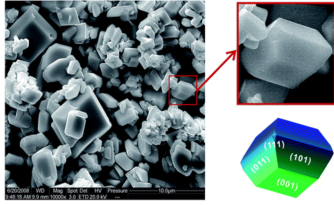


Figure 1: SEM image of $BiVO_4$ ²

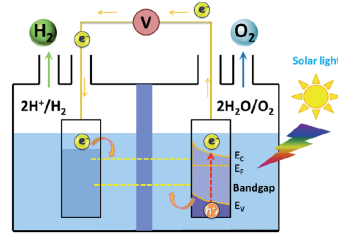


Figure 2: Theoretical progress in photoelectrochemical cell

Based on that purpose, the surface physicochemical properties of $BiVO_4$ has been carefully examined in order to improve the performances of $BiVO_4$ -based photoanodes. According to G.L. Li's study², only (110) and (111) surface has band-gap states which act as hole acceptors. When photogenerated holes and electrons migrate to the surfaces, only (110) and (111) could make the photogenerated holes localize at the surface. The localization of holes would reduce the possibility of charge recombination and consequently promote charge separation. Therefore, if the percentage of (111) or (110) can be enhanced by morphology design, the efficiency of charge separation would be further improved for $BiVO_4$.

Subsequently, identification of (111) facet of $BiVO_4$ becomes important. The traditional method would be using Electron Backscatter Diffraction (EBSD) to determine crystal facets, however, this method is restrictive and not quantifiable. In addition, there exists limited numbers of research on facet detection, especially for $BiVO_4$. Therefore, this project is focusing on the identification of (111) facet of $BiVO_4$ from SEM images. (see Fig.1)

2 Dataset and Features

2.1 Dataset

Our dataset contains around 3000 samples, including approximately 60% training data and 40% testing data. Each sample is a 160×215 -pixel grey-scale SEM image, which was generated from lab experiments.

These SEM images were labeled into either positive or negative. Positive label means the SEM image contains (111) facet and negative label means no (111) facet is detected. In order to prevent error from data labeling, the labeling process was guided by a PhD student who leads the experimental side of this project. Labeled SEM images were saved into either positive folder or negative folder and the algorithms could read the label of images based on the folder’s name.

2.2 Data Augmentation

Due to the high reaction tendency of (111) facet, exposure of this facet requires complicated chemical reaction which makes the SEM images difficult to obtain. Therefore, this project was facing a difficulty caused by this relatively small dataset. To solve this problem, data augmentation, which includes cropping, scaling, and flipping/mirroring, became an essential part of this project. As for cropping, different SEM images have different working distance which is the distance the beam is focused at; this leads to different image scale and resolution. Therefore, high scale images were cropped into 9 pieces and low scale image were cropped into 4 pieces (see Fig.3).

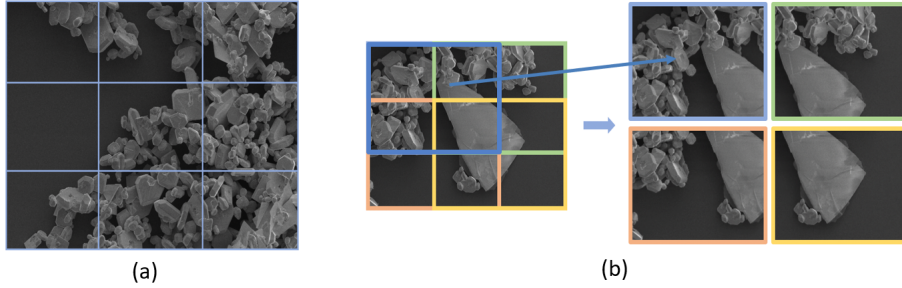


Figure 3: Data Augmentation for high (a) and low (b) working distance

2.3 Edge Detection

Another challenge came with this project was the shape variation of (111) facet. As the (111) facet grows, it could have triangular, rectangular and trapezoidal shapes. Therefore, to help the algorithms identify the key features of (111) facet, this project also tested the effect of edge detection techniques including canny edge detection, probabilistic Hough transform and adaptive thresholding. (see Fig.4)

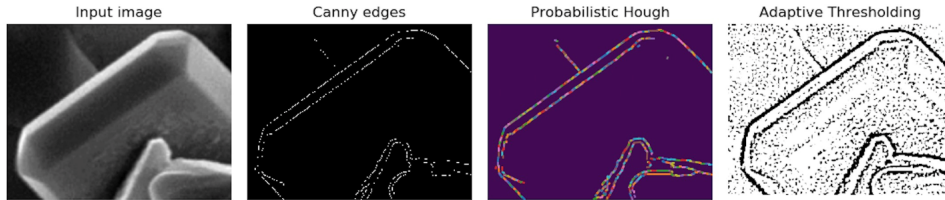


Figure 4: Edge Detection Methods

2.4 HOG

HOG is one of the most common techniques to extract features for object detection, especially for human detection. HOG represents each image by a feature vector by calculating distributions of intensity gradients, which preserves the information of local object appearances and shapes to a large extent. Since we aimed to determine the existence of (111) facet by detecting the shapes of $BiVO_4$ in our project, we chose HOG as one of the candidate methods to extract features.

3 Methods

3.1 SVM with HOG

We firstly used HOG to extract features from SEM images. After calculating HOG, a two-dimension image was converted into a one-dimension vector. We then input the vector into SVM and determined whether

it was positive or negative. We mainly considered two kinds of kernels of SVM, including linear kernel and Gaussian kernel, whose formula are shown in equations (1) and (2) respectively. The structure of this method is shown in Fig.5. The detailed information of model structures and parameters is presented in section 4.

$$k(x, y) = x^T y + c \quad (1)$$

$$k(x, y) = \exp\left(\frac{-||x - y||^2}{2\sigma^2}\right) \quad (2)$$



Figure 5: SVM with HOG

3.2 Shallow CNN

The most common method for image classification is CNN. Therefore, in this project, we used the architecture as shown in Fig.6, which contains two convolutions layers, two max-pooling layers, one flattening process, one fully connected layer, one feature dropout process, and one node output layer.

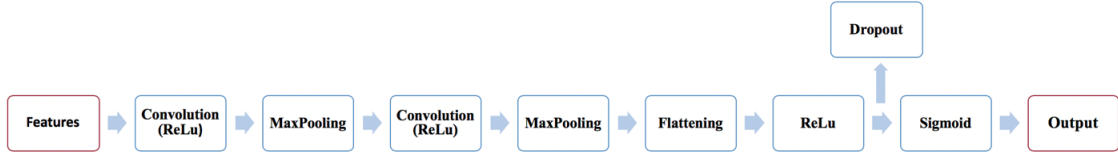


Figure 6: Shallow CNN architecture

Two activation function were used for the above shallow CNN:

ReLU

$$g(z) = \max(0, z) \quad (3)$$

Sigmoid

$$\sigma(x) = \frac{-1}{(1 + e^{-x})} \quad (4)$$

3.3 ResNet

Deep neural networks are of great importance for many image recognition tasks, and we also tried using the ResNet to train our data. Kaiming He, et al, proposed the ResNet framework to solve the degradation problem and ease the training process of deeper neural networks³. Compared with standard neural network architectures, ResNet has shortcut connections performing identity mapping and recasts the original underlying mapping $H(x)$ into $F(x) + x$ (see Fig.7).

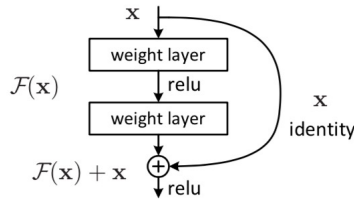


Figure 7: Residual learning: a building block³

4 Experiments

4.1 SVM with HOG

In this experiment, we did not explicitly follow the HOG steps. We divided each 160×215 -pixel one-channel image evenly into 24 40×35 -pixel blocks and calculated the total magnitude of gradient in each direction in each block, where direction was quantified into 64×64 bins. After that, we concatenated the results of 24 blocks. During the training process, we mainly modified the block number and bin number to obtain a higher accuracy.

After obtaining feature vector, SVM with linear kernel and Gaussian kernel were investigated separately. We judged the performance of each SVM model by accuracy, recall, and precision. It is shown in the results that Gaussian kernel has the highest accuracy but remarkably lower recall due to biased number of positive and negative training data. Thus, we chose linear kernel as the optimum, which has slightly lower accuracy but much higher precision and recall.

Aside from SVM kernel type, all the other parameters were chosen based on accuracy. The result showed that when block number, bin number, C are tuned to 4×6 , 80×80 and 3 respectively, the accuracy reaches a local maximum.

4.2 Shallow CNN

All parameters of CNN were chosen based on test accuracy. This project tested dropout rates, filter sizes, number of connected units, activation functions, and loss functions. For instance, among dropout rates 0.3, 0.4, 0.5 and 0.6, result showed that 0.5 dropout rate led to the highest test accuracy. Same comparison method was used on the other parameters which led to our final model. The convolutional layer contains 32 filters and each filter has a shape of 3×3 . The max-pooling layer uses a 2×2 filter. The fully connected layer contains 512 nodes and has ReLu as its activation function. The dropout process will drop out 50% of the total features. The last step for this network outputs one node through a sigmoid function. During the compiling process, it uses cross-entropy loss function.

To further understand the performance of our final CNN model, we calculated the precision, recall, and F1-score, which are 0.64, 1.00, 0.78 respectively. This result indicates that the algorithm predicted all negative image correctly, however, some of the positive prediction were wrong.

4.3 ResNet

In our project, we chose cross entropy loss as our loss function and stochastic gradient descent as our optimization algorithm. We tested our data on models with different numbers of layers (see Fig.8). Due to the relatively small dataset, we tried using transfer learning on pretrained models. As a result, the test accuracy was improved around 10% compared with non-pretrained models (see Table 1).

We found that ResNet with 50 layers yielded the best test accuracy and also was relatively computationally efficient compared with other numbers of layers. We further tuned the learning rate for ResNet50 (see Fig.9) and chose 0.0005. In addition, we used image transformation (resizing and cropping) to reduce overfitting.

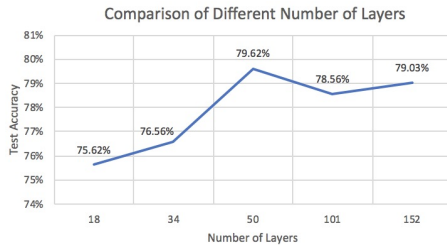


Figure 8: Comparison of different numbers of layers

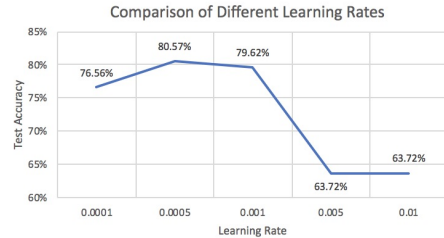


Figure 9: Comparison of different learning rates for ResNet50

5 Results

As illustrated from the result, Shallow CNN with dropout achieves the highest test accuracy. It could also be observed that both SVM and shallow CNN overfit slightly. As for the performance of ResNet, transfer learning on pretrained ResNet results in better performance. L2 regularization might be helpful for controlling the overfitting of the CNN training process, however, it also leads to a drop of test accuracy. Therefore, using only feature dropout is the better choice for shallow CNN.

Model	Training Accuracy	Test Accuracy
SVM with HOG	100.00%	61.60%
Shallow CNN (with DropOut)	99.79%	87.16%
Shallow CNN (L2 Regularization)	98.76%	85.17%
ResNet18 (non-pretrained)	63.49%	66.20%
ResNet18	70.34%	75.62%
ResNet50	74.82%	80.57%

Table 1: Results of different models

6 Discussion and Conclusion

- **SVM** Image preprocessing, such as image binarization and edge detection, has negligible effects on the results of SVM, because after extracting edge information, features of (111) facets are still less obvious compared with noise features.
- **ResNet** We tried several numbers of layers for ResNet and we found that prediction accuracy for our data did not benefit much from deeper network, might because we only have small dataset or this network structure is not suitable for our data.
- **Shallow CNN** After tuning parameters, shallow CNN obtained the test accuracy of 87%, which is the best accuracy among all models. Thus, this structure might be more suitable for our data.
- **Comparison** SVM method has fewer parameters and simpler structure compared with neural networks. Thus, it computed much faster than neural networks, but the accuracy was much lower. Compared with ResNet, shallow CNN had better control on the entire learning process. Therefore by tuning the parameters, it can get higher accuracy.
- **Data** During our labeling process, even though it was guided by experienced people in this area, there are still some error sources, such as low-quality of SEM images, different viewpoints and sizes of particles, and various shapes of particles. These led to ambiguous determination of (111) facet, which might influence the accuracy of the models.

7 Future Work

In the future, we intend to improve our project mainly in application aspect and algorithm aspect. In terms of application, we will expand (111) facet detection to other kinds of facet detection. And we will transfer classification problem into regression problem, which is to determine the proportion of target facet. In terms of algorithm, principal component analysis (PCA) will be added between HOG and SVM in order to avoid overfitting. Additionally, more feature descriptors, such as SIFT, SURF, and ORB, will be considered. For the two CNN models (shallow CNN and ResNet), we will use heatmap to detect the data concentration of the algorithms and see why shallow CNN performs better than ResNet. Based on the result of heatmap, parameters and architectures of models mentioned above will be further modified to improve the performance.

Contributions

This project is completed with the conscientious and cooperative effort of three group members Wanling Liu, Zixi Liu and Jiyao Yuan evenly in terms of idea generation, model establishment, code deployment as well as report writing. We would like to express deepest appreciation to Thomas Mark Gill for his data and guidance. Code at: https://drive.google.com/drive/folders/1F7LfFuMvE_tvuwQBP7JA1kabG1UpoK8F?usp=sharing

References

- [1] Yiseul, P., Kenneth, J., McDonald, B., & Kyoung, S.C. (2013). Progress in bismuth vanadate photoanodes for use in solar water oxidation. *Chemical Society Reviews*, Issue 6. <https://pubs.rsc.org/en/results?searchtext=Author%3AKenneth%20J.%20McDonald>
- [2] Li, G.L. (2017). First-principles investigation of the surface properties of fergusonite-type monoclinic BiVO₄ photocatalyst. *RSC Advances*. Issue 15. <https://pubs.rsc.org/en/Content/ArticleLanding/2017/RA/C6RA28006D#!divAbstract>
- [3] He, K.M., Zhang, X.Y., Ren, S.Q., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 770–778.
- [4] Transfer learning tutorial: https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html
- [5] Pytorch: <https://pytorch.org>
- [6] Tensorflow: <https://www.tensorflow.org/>
- [7] Keras: <https://keras.io/>
- [8] Hough transform: http://scikit-image.org/docs/dev/auto_examples/edges/plot_line_hough_transform.html