ELPV dataset classification

Mo Li  
  
SECOND PERSON  
  
THRID PERSON   
  
FOURTH PERSON

LAST PERSON

# Introduction

In literature review section, we will briefly discuss the related work in this field, and method I choose or considered to complete the task. In method section, I will provide in-depth discussion.

# Literature Review

Defect detection for EL images is active topic in recent years. There are several different methods proposed to solve this problem.

## Traditional Machine Learning

Traditional machine learning method is used to solve this problem. Generally, it can be described as following steps: pre-processing image for better feature extraction, feature extraction and selection, and then use a classifier to classify defect and non-defect. Sergiu [1] follows the tradition method, tested with different feature extraction method, and use SVM and CNN as classifier. Another very interesting approach proposed by Tsai [2] uses ICA to extract the basis image, and then calculate similarity between testing image and basis image. This method can only detect defect/non-defect but cannot find percentage defect. I also included this method in the code, so I can give a taste of different approaches.

When dealing with datasets, class-imbalanced problem is significant. SVM on the other hand is very sensitive to noises. Chapter 5 of He [3] gives a very good overview of how to deal with class-imbalanced problem. Imbalance problem in SVM causing decision boundary biased toward the majority class, eg. more testing sample will be classified as majority class. To solve this problem, this book [3] provided several methods such as, uses different sampling method (up sampling, and down sampling), apply class ratio to parameter C (Different error cost), that is use different C to different classes. Rehan Akbani [4] also proposed several methods to solve this problem, such as use SMOTE method for up sampling, and use different error cost to enhance the performance. Other methods are also considered, for example, kernel modification methods. This method uses class boundary alignment (CBA) to enlarge more of the class boundary around the minority class, that is, the distance of “surrounded area” are increased for minority class, that will push the decision boundary towards majority class.

Features extraction is also very important for this problem. In this project, I examined several different features extraction method. SIFT, SURF, ORB and KAZE are most common descriptors used in computer vision. Shaharyar [5] compared all methods mentioned above in different aspects, and result shows that SIFT are found to be most accurate algorithm on average. Sergiu [1] further compared SIFT and SURF, and the result shows SIFT perform well on this dataset.

Once the feature descriptors are obtained, next step is to form feature vectors. One can simply uses descriptor as feature vector by concatenate all descriptor together, however, the constraints would be different images can generate different amount of feature vectors. Most common approach would be using bag of virtual words. Gabriella’s [6] Journal article provide comprehensive conclusion using this method, and in this paper SIFT method was also used. Other methods to form feature vectors include Feature Pyramid [7], Spatial Pyramid Matching [8] are considered but not used in this project.

## Deep Learning

# Methods

In this section, we will discuss different methods used for this project. Two machine learning methods are proposed, ICA method and SVM classifier.

## ICA methods

The first method I use is following Tsai's reconstruction method [2].

### Data Preprocessing

### From some basis data analysis, EL image has randomly shaped dark regions in the background, these regions are not considered as defect. Defects are dark line shape or bar shape regions. The purpose of preprocessing is these randomly shaped region can be removed, meanwhile, we also need to retain the line-shape or bar-shape defects.

### Tsai proposed to use morphological smoothing. Morphological are used for contrast-enhancement, texture description, edge detection and thresholding. We only use morphological smoothing; it contains two basic method erosion and dilation. Erosion assigns each pixel the minimum value found over the neighborhood of the structuring element, and dilation does exactly the opposite. Dilation will assign each pixel the maximum value found in structuring elements. In this project, three structuring elements are designed and compared. Since I want to remove the random noise in the background but keep the crack line or bar, the length of structuring elements should longer than most random shaped regions but shorter than the cracks. This number is proposed as 13 in original paper, but the image in original paper is shaped 206x206. Our project has image size of 300 x 300, so I linearly increase the length. Empirically, this number is given as 17 gives optimal results. Although cracks can be in all directions, but it’s not possible to cover all directions. Three directions are used 45-degree, 135 degree and 90 degrees. They are two diagonals in 17x17 matrix, and vertical vector. 0 degree is purposely not used. Since part of image contains 2 horizontal lines, but not others, it can be easily considered as cracks, so these two lines should be removed. If 0-degree structure elements are preserved, then it will be still as it is in the image. Below image shows effect after morph smoothing.

图片包含 图表

描述已自动生成

### For each pixel in the image, we select the SE with minimum accumulated gray levels (sum of intensity within each SEs). The smallest direction will be selected as morphological dilation smoothing. Quote from original paper: "If -SE contains all defect points, the dilated value will be still small for a dark region. Otherwise, the dilated value will be large for a bright region in the background [2].” Then, dilation will select the highest value in the SE, and replace original pixel. This preprocessing method is very efficiency to remove the background noise and retain the defect.

### Image reconstruction

Tsai believe that the test image can be reconstructed by linear combination of basis images. This idea came from ICA decomposition. ICA can separate noise from observed signal, where is observed signal, is the de-mixing matrix, and is the estimation of source signal. maximize the negentropy for each independent component. If the images are “perfect” images plus random noise, then we can derive defects from basis images. Here, we make assumption that “random shape dark region” are independent and random. Once we obtain the original “perfect” image (basis image) without any random shade, then we can either use this to extract feature vector or reconstruct the image by representing test image as a linear combination of the learned basis image. Original paper tried both method, reconstruction method has better accuracy, so we only use reconstruction method.

Each test image can be represented by a coefficient vector and basis image, eg: , b can be found by b , where is pseudo-inverse of U. Coefficient vector can also use as feature vector for other classifier. Reconstruction method uses this coefficient vector to reconstruct test image y. Once images are reconstructed, simply calculate the different between reconstructed image and test image, If the test sample contains defects, then it is expected that the error should be large. Below is calculation for the error.

Here c is constraint parameter used as regularization. When this error is greater than a predefined threshold, then it considered as defect otherwise its non-defect. I further extend this idea to set three different thresholds, and each one corresponding to a different percentage of defects.

Tsai also proposed an easier way to calculate as below:

Use this way, we don’t need to calculate ICA, and can simply use original matrix X calculate reconstructed test image.

Threshold are calculated as follow. We first select all defect-free data in train set and we split this subset to 5-folds. We use 4 folds as X, and remaining fold as y, since all images are defect free, we can calculate the mean error and standard deviation of this error. We take mean of mean error from 5 folds and mean standard deviation as error standard deviation. The threshold is then mean error plus one, two and three standard deviations for each 33%, 66% and 100 % defect.

We can use same method on test images and compare the error with threshold values.

## SVM method

Second method we use is feature extractions and SVM classifier.

### Augumentations

Purpose of augmentation is to increase variety of sample and increase model’s ability to generalization. Several methods were considered in this project, include random noises, rotation, and flip. Here rotation and flip are only applied to method that is variant to rotations, and flip. SIFT on the other hand produce same feature vector use rotation and flip.

### Preprocessing

Image preprocessing can remove noises and can help feature extraction method to extract more accurate feature. I have tried a several methods: CLACHE contract, Gaussian blur prior to Laplacian for edge detection, local binary pattern, morphological opening, and morphological smoothing. Since there are no standard for what a good feature after preprocessing is, I can only try each combination. Each combination are tested, and only the preprocessing method yields best result will be used for feature extractions.

For parameters, I must visualize its effect to determine which one “looks like” gives better result, however this method is not rigorous.

### Feature extraction

Only SIFT and HOG descriptors are used in this project.

SIFT descriptor can be extracted from majority of images, however, for some images that is too dark, sift enable to extract any descriptors.

We later built a bag of visual words use extracted feature descriptor. All descriptors are first using different K value to find clusters. Each cluster represent how many different unique descriptors we can find in the image. Since we have 8 different classes, we can make assumption that each class can be represent by 10 different clusters of descriptors. For example, line-shaped scare can be detected as a descriptor, and amount of line shaped crack can then use as feature vector in classifiers. K value is set from 80, 160, 320, and 640. That is, we assume that there are n \* 10 unique features in each image. Then we run Kmean algorithm to predict all descriptors, if it close to any clusters, the corresponding bin will increase one. Some images with 0 descriptors will obtain all 0 in the histogram. The resulting feature is length of K for each image. Then we can use this histogram run classification use SVM

HOG feature also considered in this project. HOG is efficient in object detection task; it can capture the appearance and shape of an object in image by finding distribution of intensity gradients and edge directions. HOG run as following steps: it first create small cells, in our case 10x10 cell in the image, and for each cell, it will compute a histogram of gradient directions to capture local features. For global features, it will further combine these cells to blocks. HOG unlike SIFT, it will combine all these descriptors into bins, I set it to be 10 (HOG original proposed 9 bins, but I empirically tested that 10 would be better number bins). A window will be slide on entire image, and each window would be a block. I can then obtain a feature vector for each sliding. I set the number of cells per block to 1, again this is we tested with different number, and we find that 1 gives better classification result.

### Model selections

Different classification model was considered. I cross-validated KNN, SVM, Forest, Regression, adaboost, and gradient boost. Comparing each of its result, I found that SVM would be best in this project. Cross-validation on models is only used default parameters without fine-tuning. After I have decided to use SVM, we rigidly test different parameter for SVM.

As I mentioned in earlier sections, there are lots of application using SVM for EL images, and in this specific dataset, class-imbalance is causing a big issue.

SMOTE method is used along with DEC method to fix imbalanced class. SMOTE find the training data x and its closet neighbor z. A new data u will be generated between x and z.

Where w is a random number between 0 to 1. Using 2-D example,图表, 散点图

描述已自动生成

Above figure demonstrated synthetic data (x mark) are generated for class 0.

Another approach I used is to train classifier for both one vs one and one vs others. For each class, we can train a SVM classifier against all other classes, and when testing phase, distance to each boundary are calculated, and we predict test instance as the class that is far away from boundary.

## VGG Method

To fit the model, all images were converted into tensor and normalized. Normalization is done by subtracting the mean of the image and dividing it by its standard deviation, which helps to speed up the convergence of the model training and improves its performance.

We implemented the VGG11 model using the PyTorch framework. During training, the cross-entropy loss function and Adam optimizer were used. The split ratio between the training and test sets is 75% and 25%. To address the problem of category imbalance, we adopted an oversampling technique (SMOTE) to increase the number of samples from a few categories.

During model training, we tuned several hyper-parameters such as learning rate and batch size to obtain optimal training results. We also implemented an early-stop strategy to prevent overfitting, stopping training when the performance on the validation set no longer improves.

# Experimental Results

## Machine learning Method

There is total 656 data in the dataset, and 377 data (0.57) are 0 percent defect. 178 out of 656 are class 100 defect (0.27). So the base line is 57%.

Machine learning model achieves 70% overall accuracy use HOG, specifically, for mono, poly and combined dataset, 75%, 69% and 70% accuracy was obtained. Hog classifier can not accurately predict class 1 and 2 (33% defect and 66% defect), most of accuracy are gained from class 0 (0% defect). Weighted f1 score for best HOG model are 0.72, 0.64, and 0.66, since weighted f1 will consider imbalanced class and assign higher weight to majority class, if average f1 score is calculated for each class, then the result is terrible.

图形用户界面, 应用程序

描述已自动生成

SIFT also achieved similar result with 72%, 67%^ and 71% accuracy for mono, poly and combined data. 68%, 64% and 67% weighed f1 score are obtained use grid search.

图形用户界面, 应用程序

描述已自动生成

Re-construction method are the worst, it received same accuracy with baseline, 57% accuracy.

图表, 折线图

描述已自动生成

图表, 折线图

描述已自动生成

Above plots show SVM classifier under different parameter settings. K value is also considered as a parameter. From first figure, we can see that increasing in k value will slightly increase model performance, and rbf kernel can overfit the model yet also results best validation result.

## Deep Learning

Our model performs well on the test set with a high level of accuracy (The average accuracy can be as high as 87%). Specifically, the model has a balanced performance in classifying solar panels with different damage levels, proving its effectiveness in handling such image tasks.

The results of the confusion matrix show that the model is particularly effective in distinguishing between undamaged and fully damaged solar panels. However, the model's performance in recognizing moderately damaged (e.g., 33% and 67% damaged) solar panels needs to be improved. This suggests that the model may need further optimization when dealing with more subtle damage features.

We also utilized graphs to demonstrate the performance of the model, including the curves of the accuracy and loss function changes. These visualization tools help us better understand performance of the model.

# Discussion

## Machine Learning part

The result from machine learning part does not met expectation. Here might be the reason of it.

Feature extraction and preprocessing did not enhance classification result or increase model’s generalization ability.

Use sift as example, below two plot shows sift detection result on original image(left), CLACHE enhanced image(middle), from the visualization result, we can find enhancement help us (human) easier capture all the cracks and changes of color, but it SIFT algorithm can not capture these details (Dark blobs for the first image, crack from second image). From this plot, we can clearly spot few new interest point detected by SIFT, however, with all parameter unchanged, SIFT’s performance did not increase at all, SVM performance was reduced for 5%. I can only conclude that SIFT’s interest point can not reflect on this dataset feature.

图形用户界面, 文本, 应用程序

描述已自动生成

图片包含 图形用户界面

描述已自动生成

图形用户界面

描述已自动生成

图形用户界面

描述已自动生成

Another example of using Laplacian edge detection and adaptive thresholding to find features. Laplacian is very good at detect changes in the image, however, Laplacian is very sensitive to noises. Above images shows original image (left one), gaussian blur and Laplacian feature (left 2), morphological opening smooth (left 3), Adaptive thresholding on left 3 (fourth), sift feature on Laplacian image(5th), and HOG feature on Laplacian image.

Another question I have encountered is SVM’s class imbalanced issues. Although lots of methods are tested, but the effect is very small. SMOTE only increase accuracy by 1%. When I use ensemble learning bagging, each individual classifier is doing very well, all classifiers can obtain around 99% accuracy, however, when I combine the result together, the accuracy drop dramatically. A wild guess is due to the extreme imbalanced class.

# conclusion

In this project, we found that the performance of the VGG11 model is limited by the number of samples in the dataset. Although we used oversampling techniques such as Synthetic Minority Oversampling Technique (SMOTE) to increase the number of samples in a few categories, the overall sample size is still low, especially for solar panels with some specific damage levels such as slight or moderate damage. This leads to limitations in the model's ability to learn and generalize over these specific classes.

Since deep learning models, especially complex models like VGG11, rely on a large amount of data to learn rich and complex features, the lack of sample size may result in the model not being adequately trained, which in turn affects its performance in real-world applications. In our case, this may be the main reason for the model's poor recognition accuracy on certain damage categories.

In addition, due to the limitation of the amount of data, the model may be overfitted to the training data during the training process and perform poorly on unseen new data. This overfitting is more common in deep learning projects, especially when the available data is limited. However, the use of Convolutional Neural Networks (CNNs) such as VGG11 still demonstrate superior performance compared to traditional image processing and machine learning methods. The advantage of CNNs is their ability to automatically learn complex features from the data, which often requires manual design and tuning in traditional methods.

Despite the challenge of insufficient sample size, the VGG11 model is still able to identify different levels of solar panel damage. This demonstrates that even under limited data conditions, the deep learning capability of CNNs can provide more accurate classification performance than traditional methods. CNN shows its unique advantage in capturing and processing subtle features in images.

##### References

1. Deitsch, S., Christlein, V., Berger, S., Buerhop-Lutz, C., Maier, A., Gallwitz, F., &amp; Riess, C. (2019). Automatic classification of Defective Photovoltaic module cells in electroluminescence images. Solar Energy, 185, 455–468. https://doi.org/10.1016/j.solener.2019.02.067 J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
2. Tsai, D., Wu, S., & Chiu, W. (2013). Defect detection in solar modules using ICA basis images. IEEE Transactions on Industrial Informatics, 9(1), 122–131. https://doi.org/10.1109/tii.2012.2209663
3. He, H., & Ma, Y. (2013). Imbalanced learning. In Wiley eBooks. https://doi.org/10.1002/9781118646106
4. Akbani, R., Kwek, S., & Japkowicz, N. (2004). Applying support vector machines to imbalanced datasets. In Lecture Notes in Computer Science (pp. 39–50). https://doi.org/10.1007/978-3-540-30115-8\_7
5. Tareen, S. a. K., & Saleem, Z. (2018). A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET). <https://doi.org/10.1109/icomet.2018.8346440>
6. Csurka, G. (2004). Visual categorization with bags of keypoints. *European Conference on Computer Vision*, *1*, 22. https://www.cse.unr.edu/~bebis/CS773C/ObjectRecognition/Papers/Dance04.pdf
7. Lin, T., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2016). Feature pyramid networks for object detection. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1612.03144
8. S. Lazebnik, C. Schmid and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 2006, pp. 2169-2178, doi: 10.1109/CVPR.2006.68.