Optical Signal Recognition — Characterization of Focused Laser Spots

Yiyang Liu, Taigao Ma, Xiangpeng Luo, Lingxiao Zhou
Application Track

Abstract

We study in this project the machine learning-based recognition of a laser beam measured and displayed in an intensity map. The characterization of the laser signal includes its spatially integrated intensity and its position on the map. We show firstly that unsupervised learning methods assist in predicting the intensity level, and then supervised k-means++ and CNN can determine the location of the laser spot. The percentage errors from various unsupervised learning methods are reported, and the relevant origins of such errors from either the model deficiency or the label pre-computation process are discussed. The excellent capability of k-means++ and CNN distinguishing real signals from impurities is also explained in this study.

1 Introduction

Pattern recognition is a popular field of research, as it allows automated machines to interpret and categorize what they "see" and "observe" in an image or a video. Combined with machine learning (ML), pattern recognition has been widely used in character recognition [1], automatic drive [2], computed tomography [3] and so on.

Apart from those common commercial applications, pattern recognition can also be applied in scientific domains. Laser-based optical research has long been a focus in various subjects such as Physics, Chemistry, Material Science, Electrical Engineering etc., greatly owing to the multivariate features of a laser beam and the capacity of such features coupling to other degrees of freedom. It is desired to characterize those features of a laser beam to study, for example, how the laser interacts with the probed sample. In an optical research lab in the University of Michigan, researchers find it drudgery to recognize and extract the optical signal from images collected during experiments. The current manual procedure of processing the images is further hindered by the enormous amount of images. The research group readily consents to applying ML methods in the situation.

Our task is to recognize if an optical signal exists from the noisy image, to identify the position as well as the effective intensity of the signal if there is one. Since the research lab is currently searching optical signals by brute-force, our project will potentially facilitate the process of data analyses, and free the scientists from the time-consuming manual inspection, so that efficient and productive research can be anticipated.

2 Dataset

We will use data from actual experiments from an optical research lab in the University of Michigan. The dataset has merely been analyzed based on visual inspection and conventional optimization by the lab and was made public and available when the relevant work was published.

2.1 Data description

In the signal collection end of an optical system, a focused laser beam is sent on a detection camera—a pixeled charge-coupled device (CCD). The CCD senses the spatial distribution of the beam intensity over its 512-pixel by 512-pixel area and outputs the image as a square matrix of positive integers

$$\boldsymbol{I}_i, \text{ where } I_{i,mn} \text{ is the intensity at pixel } (m, n)$$

Each image will be vectorized as an input feature vector $\boldsymbol{I}_i \mapsto \boldsymbol{x}_i$.

One may observe the following features from a typical image as Fig. 1:

- A prominent laser spot with some distribution of high intensity;
- An otherwise homogeneous background with certain level of fluctuation;
- Some random impurities from either Cosmic Rays or instrumental imperfections. The former occasionally appears in some images, manifested in the forms of an evanescent line (Fig. 1(a)) or a sharp spot (Fig. 1(b)). The latter may cause usually weaker and more extended features, as will be discussed in Fig. 4.

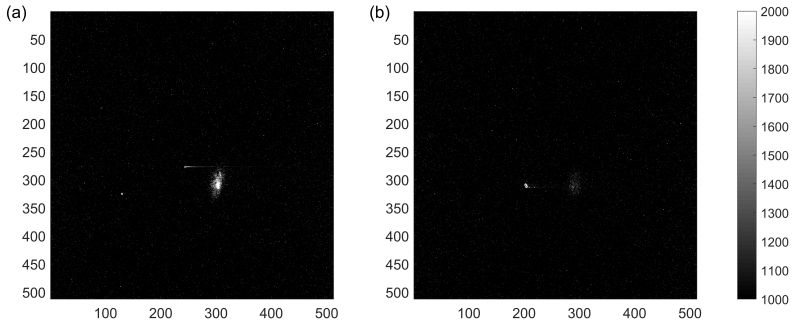

Figure 1: Two typical input images showing the typical signal and Cosmic Rays manifestations

The images were acquired under very different conditions and could be segregated into subsets. To formulate the dataset, consider the image $\boldsymbol{x}(\mathscr{C}; c)$, where the $\mathscr{C}$ denoting the overall experimental condition such as samples being studied, different optical components etc., is the Tag for subsets, and c represents a controlled parameter within each subset. Across different subsets, the positions of the laser spots as well as the intensity levels would vary in unpredictable fashions. However within an individual subset, controlling the parameter c will only modify the laser spot in a patterned way, such as the laser spot tracing a small ellipse and the intensity showing trigonometric dependence, as exampled in Fig. 2.

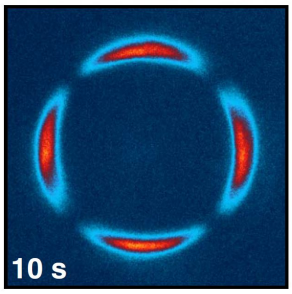

Figure 2: Summation of several images within a subset $\sum_i \boldsymbol{x}_i(\mathscr{C}; c_i)$. The laser spots in this subset trace a perfect circle and the integrated intensity varies sinusoidally. Color is added for contrast. The raw image is still in gray scale. Figure adapted from [4].

The function of the target ML algorithm is two-fold. Firstly, from each image it is desired to predict the integrated intensity of the true signal. This can be implemented through supervised learning, in which case the labels (the actual signal intensity y_i) for each image are pre-computed by the method described in the following section. Secondly, of more practical importance, one wishes to recognize the laser spot and extract its position on the image. This is required by the observation that as shown in Fig. 2, in a subset the positions of the signal trace a particular shape that contains much information for the researchers. This task is unsupervised in nature and accomplished by clustering methods and CNN.

For this second task, one may need to carefully fiddle with the impact of the impurities in recognizing the true signal. We take the advantage of the correlation of the signals in a subset. For example,

should more than one "signals" be recognized for a single image, one could refer to the other images in the same subset and judge that only the signal close to those in the other images is real. More details of this issue will be discussed later in this report.

2.2 Non-machine learning pre-processing

The evaluation of the integrated intensity y_i of the laser spot can be performed by a conventional two-dimensional searching. One may let a square window, the size of which is larger than that of the laser spot, slide through the whole area and search for the maximum intensity integrated within the window. This process can be expressed by

$$\text{I.I.}_i \text{ (Integrated Intensity)} = \max_{x_0,y_0} \sum_{m=x_0-a/2}^{x_0+a/2} \sum_{n=y_0-a/2}^{y_0+a/2} I_{i,mn} \tag{1}$$

where (x_0, y_0) is the center of the window and a is the length of square window side. Note the nominal intensity at pixel (m, n) contains both the signal intensity and the underlying background, thus this quantity must further be subtracted by an estimated background intensity

$$y_i = \text{I.I.}_i - N_{\text{box}} \frac{\sum_{(m,n)\text{outside the box}} I_{i,mn}}{N_{\text{outside the box}}} \tag{2}$$

Such y_i's have been pre-computed for each image and will be used for supervised learning.

3 Methodology[1]

Plenty of research has been done in this field. The k-mean++ we are going to use in this project was proposed by Arthur et al. in [5]. The article improves the k-means method first described by Lloyd [6] and achieves an expected error of order $O(\log K)$. The neural network based approach inspiring our work is devised and implemented by Badrinarayanan in [7], and the network architecture we use is proposed and implemented by [8]. Other common clustering and neural network based segmentation approaches include the Fuzzy C-Means clustering [9], Mask R-CNN [10], etc. A literature review of the existing image segmentation techniques is done by Nikhil R.Pal and Sankar K. Pal in [11].

In this project, we want to estimate the magnitude and recognize the position of the optical signal measured on a pixeled CCD camera. The expected output is the integrated intensity and the coordinates of the center of the signal. We use two types of algorithms - supervised and unsupervised learning for the two tasks respectively. In the supervised setting, we train the data with the manually evaluated intensity y_i, obtained from the method described in section 2.2 and aim to predict the position and signal intensity for the test data.

In the unsupervised setting, we want to find the position of the signal by segmenting the signal from the background. That is, each image pixel (m, n) contains a one dimensional feature (the intensity I_{mn}), and we wish to assign a label $l_{mn} = f(I_{mn})$ to each pixel via clustering based on the similarity of the 1D feature, where $f : \mathbb{R} \to \{1, \cdots, K\}$. With our knowledge of the dataset, we know that the optimal number of clusters is $K = 2$. This problem is a slightly simplified version of unsupervised segmentation, where the algorithm will have to first optimize the number of labels. The desired results of the clustering should satisfy:

- There are two unique clusters, one for the signals C_{signal} and the other for the background C_{bkg}.
- The signal labels and background labels are correctly assigned to the pixels with correspondent features.
- The result of clustering should enable us to locate the signals and find their intensities, since these are the motivations of the project.

[1] In the supervised setting, we apply classical methods that are covered in class, and we mainly use the lecture notes as the reference. As for the unsupervised setting, we refer to literature in clustering and image segmentation.

3.1 Supervised learning algorithms

3.1.1 Data pre-processing

Considering the high dimension (512×512 pixels), it is presumed hard to perform regression or similar learning process based on this large number of degrees of freedom. However, downsizing each image into a smaller size (e.g., 64×64 pixels), which will potentially submerge our signal into noise, is not ideal as well. This is due to the experimental observation that the typical size of optical signal is very small compared to the total sensation area. Therefore, we choose the principal component analysis (PCA) to pre-process our data and reduce the dimensionality.

The idea behind PCA is to approximate $x_i \approx \mu + A\theta_i$, where $\mu \in \mathbb{R}^d, A \in \mathcal{A}_k, \theta_i \in \mathbb{R}^k$, $\mathcal{A}_k$ is the set of all $d \times k$ matrices with orthonormal columns, $d = 512 \times 512 = 262144$. In our pre-processing, we select the smallest k such that the optimal rank-k approximation explains 95% of the variation in the data, meaning that $\frac{\lambda_1+\cdots+\lambda_k}{\lambda_1+\cdots+\lambda_d} > 0.95$. After the PCA transformation, the dimension of images are reduced to $k = 900$, which greatly simplifies further learning process. We also point out that a followed robust-scale has been tested attempting to remove the median and scale the data according to the quantile range. However, it turns out this combined pre-processing produces similar results with only using PCA.

3.1.2 Comparison of some common methods

The dataset is divided into two parts by a given ratio 30%. The first 70% are used for training, while the remaining 30% are used for testing. After learning parameters from training samples and predicting the intensities of test samples, the percentage difference between predicted intensities and the actually values is calculated. The averaged percentage difference serves as the performance measure.

For unsupervised learning, multiple methods have been tested, including linear regression [12], random forest [13], gradient boosting [14] and multilayer perceptron (MLP) [15]. The results from those methods are described later in this report.

3.2 Unsupervised learning algorithms

3.2.1 Data pre-processing

For both k-means and CNN, we apply the non-local means denoising filter which replaces the intensity I_{mn} of a pixel with an average of the intensity of similar pixels $\tilde{I}_{mn}$ [16] to reduce the noise in the background. In the case of k-means, the denoising filter is applied before and after the clustering for different purposes (see Section 4.2). Additionally, since the inputs of the CNN are assumed to be RGB images (see Section 4.3), the images are converted from gray-scale to three-dimensional RGB images by stacking the one-dimensional features $\tilde{I_{mn}}$ of the filtered data. That is, we obtain new pixel-wise features for the image $s_{mn} = (\tilde{I}_{mn}, \tilde{I}_{mn}, \tilde{I}_{mn}) \in \mathbb{R}^3$. For images in which the signal is faint or merely visible, we set a threshold for the maximum intensity (i.e., the maximal pixel-wise feature) in an image. If $\tilde{I}_{mn} < 1$, the intensity of the entire image is scaled by a factor of 3. To avoid the bright spots around the edges of the image and to reduce the running time, we perform a center crop of size 300×300 to all the tested images.

3.2.2 K-means based algorithm

The k-means method is applied as our first unsupervised ML approach. The basic idea of this algorithm is to partition the input data into k clusters by minimizing the variance within each cluster. For our signal segmentation problem, we use k-means++ which is modified from traditional k-means method by incorporating the randomized seeding. With the input data being a matrix representing a gray scale image, by k-means++, the ideal output image (a matrix) should contain clear segmentation regions as clusters, from which we can easily identify where the signal spot locates. Especially for our dataset, we have a natural choice of $K = 2$, namely, we wish to separate each image into two regions: one for the optical signal, another for the background.

The goal is to identify the bright signal spot, characterizing its location and intensity. We firstly implement k-means++ on the original images. Since the contrast between the signal and the back-

ground is strong, only the region close to the brightest spot is contained in the cluster. Note the nominal intensity level at the signal region is actually the summation of true signal intensity and the underlying background, the latter presumed to be of the same level as those outside the signal region. A denoising process is therefore applied after clustering to remove the extra hidden background. The position and the intensity of the signal are thus extracted unbiased.

Noticeably, k-means with denoising achieves another special goal–to identify the impurities or fake signals. As mentioned in Section 2.1, some images come with random impurities that are due to either the Cosmic Rays or instrumental imperfections. For intensive and centralized Cosmic Rays as shown in Fig. 1, k-means++ tends to mis-recognize them as the signal cluster. To address this issue, we can quickly refer to the other images from the same subset $\mathscr{C}$, the signal positions of which must be spatially close and usually trace a circle, then judge that only the signal cluster in the current image close to that in other images is the true one. However such judgement may become invalid in the case of a weaker and broader impurity, such as the example shown later in Section 4.2. We find that, by swapping the order of denoising and applying k-means++, the weak impurity, the darker background and the brightest true signal can all be distinguished. This surprisingly good performance is further discussed in Section 4.2.

3.2.3 Convolutional Neural Networks in unsupervised learning

CNN has been widely used in image segmentation and object recognition in supervised settings, but its great ability to extract and process pixel-level features allows us to use it in an unsupervised setting. The model we use for this project is modified from the unsupervised image segmentation algorithm proposed in [8]. The architecture consists of M convolutional components and an argmax classifier. Each convolutional component includes a two-dimensional convolution layer, the ReLU activation function, and a batch normalization layer. The choices of parameters and dimensions of the data we use in the project are listed in Table. 1

Parameter	Value
Number of convolutional components	3
Max iteration	200
Learning rate	0.1
Number of output channels	2
Size of input image	300×300 (Center-croped)
Activation function	ReLU
Optimizer	SGD

Table 1: Choices of parameters for unsupervised CNN

In the CNN, the input feature $\{\mathbf{s}_{mn}\}$ are passed onto the next layer as $\{\hat{\mathbf{s}}_{mn}\}$ by applying the convolutional filters and the ReLU activation function. Then, the distribution of $\{\hat{\mathbf{s}}_{mn}\}$ will be normalized to that of zero mean and unit variance in the batch normalization layer [8]. Eventually, the cluster labels will be obtained with the argmax classification function which chooses the best label for each pixel. The key to correctly classify each pixel in unsupervised setting is a backpropagation process based on the stochastic gradient descent (SGD). The algorithm will calculate the cross-entropy loss between the normalized $\{\hat{\mathbf{s}}_{mn}\}$ and the defined labels. The parameters in the convolution layers are then updated according to the calculated loss. To retrieve the position and intensity of the original signal, we calculate the center of the correspondent cluster, and find the correspondent intensity from the raw data. The structure of the CNN is illustrated in Fig. 3.

4 Results and discussion

4.1 The results for supervised learning

For the unsupervised learning, we employ multiple models including preliminary linear regression, random forest, gradient boosting, and multilayer perceptron. Their performances, measured by the averaged percentage error, are shown in Table 2. Random forest achieves the best performance – 24.53% error. Overall, supervised learning models except the simple linear regression predict

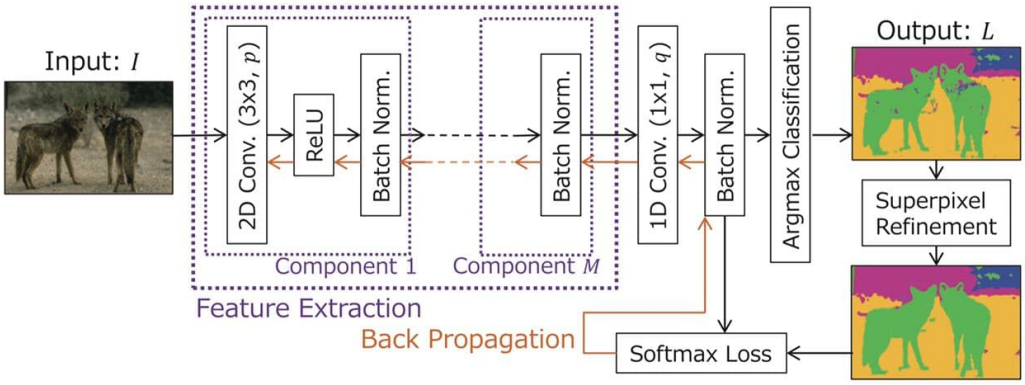

Figure 3: Illustration of the unsupervised CNN [8]. Since we do not worry about the spatial continuity of the image segments due to the simplicity of the images, we omit superpixel refinement to accelerate the process.

the magnitude of signal deviating 20-30% from the labeled magnitude. Despite the seemingly not-as-pleasant percentage error, several remarks are made on the supervised learning methods.

Method	**Error (%)**
Linear Regression	48.56
Random Forest	24.35
Gradient Boosting	28.41
MLP	30.76

Table 2: Performance of supervised learning

We note the desired signal intensity can be naturally computed through

$$\text{signal} = \sum_{i \in C_{\text{signal}}} (x_i - \text{bkg}) \tag{3}$$

where x_i is the nominal intensity at pixel i, and bkg denotes the estimated per-pixel background intensity level which is presumed constant throughout the whole image. From the linear regression perspective, this formula can be expressed in terms of

$$\text{signal} = \boldsymbol{\omega}^T \boldsymbol{x} + b \tag{4}$$

where $\omega_i = \mathbf{1}_{\{i \in C_{\text{signal}}\}}$ and $b = -|C_{\text{signal}}| \times \text{bkg}$. Without the information of cluster assignment, namely the label of each pixel, a naive linear regression model is unable to determine which dimensions of $\boldsymbol{\omega}$ should be assigned 1. In certain circumstances where the training images and test images have coincidentally similar signal position and distribution, the linear regression shows relatively good performance, as the learned $\boldsymbol{\omega}$'s for such images have the very similar structure. However this "luck" does not extend to general data points where the signal positions could be quite random.

The ensemble methods and MLP, in general, better accommodate diversity of data and therefore could be less impacted by the complications mentioned above. We point out the 30% error could very much be intrinsic from the experiment itself, which can result from a sequence of non-ML facts. Firstly, the labeled intensities of both training data and test data are obtained from the preprocessing described in Section 2.2. This process brings error by itself that can be viewed from $\Delta(\sum_{\text{box}} I_{ij} - N_{\text{box}} \times \text{bkg}) = 30\% \times \text{Signal}$. With our signal intensity level of 10^5 and the number of pixels inside the box $N_{\text{box}} = 60^2$, this yields the background intensity variation $\Delta\text{bkg} = 8$, which is quite possible in our estimation of the background intensity. Meanwhile, the actual laser power from the research lab exhibits about 10% fluctuation that could also hinder the accuracy of our ML algorithms.

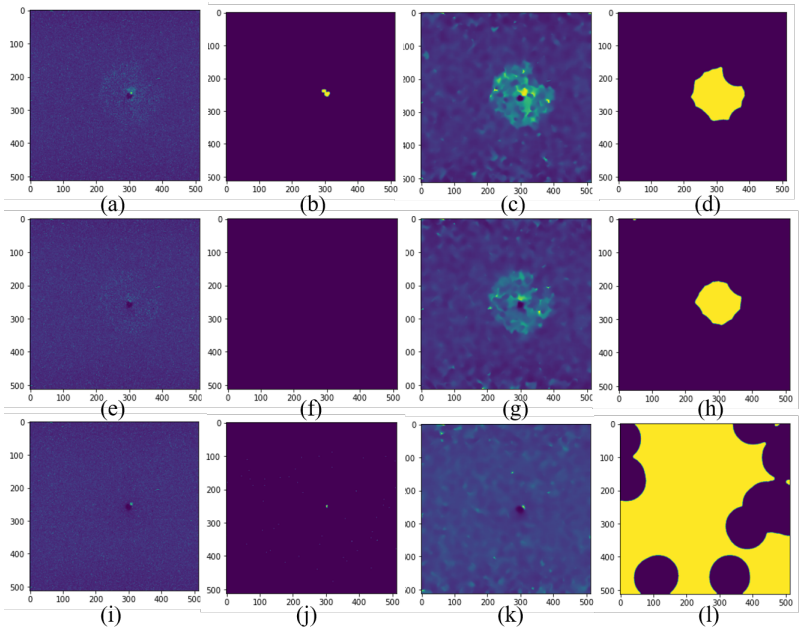

Figure 4: Typical results for unsupervised k-means. First column is the original images, second column shows the recognized bright cener, while the third columb shows the denoised images, and the last column shows the segmentation, while yellow and black region represent the region of fake signal and background.

4.2 The results for unsupervised k-means based algorithm

Fig. 4 shows typical learning results for k-means unsupervised algorithms. The images are shown with color since they are converted purposely to the RGB space. We are trying to find the position of the optical signal, as well as the fake signal or impurities. The first column in Fig. 4 shows the original images to be processed, each corresponding to a specific situation. Fig. 4(a) possesses both the real signal and the fake signal, while only fake signal in Fig. 4(e), and only real signal in Fig. 4(i). The fake signal in this case is located around the real signal if there is one, and its brightness is much weaker than that of the center, although bright enough for us to distinguish from the background.

K-means shows great ability to distinguish the bright real signal from the fake signal or the even darker background. On one hand, after firstly k-means++ clustering and then denoising, the second column identifies the location and intensity of the real signal. Our algorithm does not recognize any signal for the second row, which is exactly what we expect. On the other hand, if applying denoising first, we see in the third column a rough appearance of the fake signal. Then applying k-means++, we notice that the last column extracts the region of fake signal, where yellow regions represents the fake signal, and the black regions refers to the background. We can see our algorithm works well to segment the fake signal from background for the first and the second rows. The shapes of recognized fake signal regions also relates well with the original images. However, For the last row, our algorithm recognize the signal, but it gives weird clustering result when we try to identify the fake signals. The average time this algorithm takes to process each image is 45.96 seconds. The long time is due to the randomness of k-means clustering, and to compensate for that we run the algorithm five times on each image.

4.3 The results for unsupervised CNN

We tested the algorithm on the first 200 images. We compare the positions of the signals identified by the algorithm with the human-labeled ones. If the identified signal is within a circle of radius 5 (pixels) centered at the position of the hand-labeled signal, we will call the identification a success. Out of the 200 images with a signal present in each image, we have identified that in 84% of the test images the signal is present, but only 56% of the calculated positions are close to the manually labeled signals by our standard.

Fig. 5 illustrates the four types of outputs we see when we run the unsupervised CNN algorithm on the first 200 images. The first 40 images contains impurities a.k.a. fake signals around the signal laser spot. The algorithm can successfully identify the signals in Tag 6 and Tag 7. These two input images

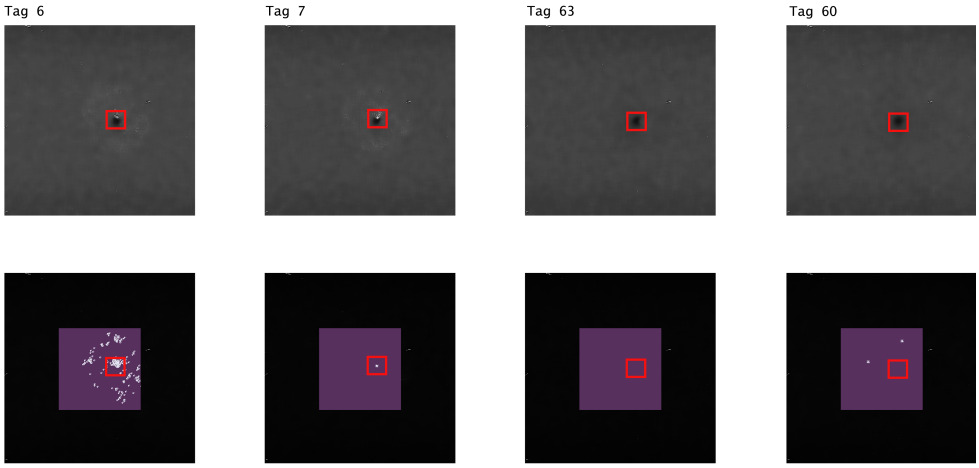

Figure 5: Some results for unsupervised CNN. The first row demonstrates the input image 6, 7, 60, and 63, and the second row demonstrates the result of the clustering. The cluster of the signal is in bright color, and the red frame indicates the position of the signal.

are similar to the figure on the left in Fig.1, which contain visible laser spots. In the result for Tag 6, the signal cluster is not spatially continuous, and it classifies part of the fake signals into the signal cluster. However, since the fake signals are centers around the laser spot, the center of the cluster is still close to the true value. The result for Tag 7 is very neat. The signal cluster is one single point that locates at the exactly same position as the signal in the raw data. As we calculate the position by finding the center of the cluster, both 6 and 7 can give positions that satisfy our evaluation criterion. While in Tag 60 and 63, we encounter the situation on the right in Fig.1, where there is barely visible laser spot. The algorithm mistakes the noises as the signal (in Tag 63) or simply fails to identify a signal (Tag 60). However, as is explained in Section 4.2, such mis-label can be readily corrected by referring the other images in the same group.

In general, the unsupervised CNN algorithm demonstrates decent ability to identify and local the signal when a clear laser spot is present, and despite that the clustering result may not be ideal as in the case of Tag 7, it can still predict the position of the signal within a certain level of precision. However, it's unable to identify the signal when a bright spot is not present, and that's the main reason that the overall accuracy is not ideal. The total time for processing 200 images is 2628.38 seconds on a CPU @ 2.70 GHz, and that's about 13 seconds per image on average. Although the total has a linear relation with the number of the data, and that the running time can potentially be shortened by reducing the number of maximal iteration, the algorithm is still fairly time-consuming for a large dataset.

5 Conclusion

Optical signal recognition has been studied in this project. By applying basic ML algorithms to research lab data, we successfully introduce ML methods into conventional optical research, which potentially broadens the library of data analysis methods in this lab and sets new direction for promoting the research work efficiency. Our unsupervised learning practices exhibit moderate error level of laser signal intensity that is partly due to the intrinsic label pre-computation process. It is still under exploration if any unsupervised learning framework can directly produce the desired intensity, thus one can avoid the pre-computed labels that are more or less inaccurate. The unsupervised learning for locating laser spots works well, especially in terms of distinguished various forms of impurities. However our current algorithms still require some manual work when it comes to inspecting multiple images in a subset. Further coding refinement is required to realize a fully automated workflow.

References

[1] Geoff Dougherty. *Pattern recognition and classification: an introduction*. Springer Science & Business Media, 2012.

[2] EKIM YURTSEVER. A survey of autonomous driving: Common practices and emerging technologies. *IEEE ACCESS*, 8, 2020.

[3] Sasank Chilamkurthy, Rohit Ghosh, Swetha Tanamala, Mustafa Biviji, and Prashant Warier. Deep learning algorithms for detection of critical findings in head ct scans: a retrospective study. *The Lancet*, 392:2388–2396, 2018.

[4] J. W. Harter, L. Niu, A. J. Woss, and D. Hsieh. High-speed measurement of rotational anisotropy nonlinear optical harmonic generation using position-sensitive detection. *Opt. Lett.*, 40(20):4671–4674, Oct 2015.

[5] D. Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *SODA '07*, 2007.

[6] S. P. Lloyd. Least squares quantization in pcm. *IEEE Trans. Inf. Theory*, 28:129–136, 1982.

[7] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[8] Asako Kanezaki. Unsupervised image segmentation by backpropagation. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018.

[9] J. Bezdek, R. Ehrlich, and W. E. Full. Fcm: The fuzzy c-means clustering algorithm. *Computers Geosciences*, 10:191–203, 1984.

[10] Kaiming He, Georgia Gkioxari, P. Dollár, and Ross B. Girshick. Mask r-cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:386–397, 2020.

[11] N. Pal and S. Pal. A review on image segmentation techniques. *Pattern Recognit.*, 26:1277–1294, 1993.

[12] Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to linear regression analysis*, volume 821. John Wiley & Sons, 2012.

[13] Mark R Segal. Machine learning benchmarks and random forest regression. 2004.

[14] Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

[15] Dennis W Ruck, Steven K Rogers, and Matthew Kabrisky. Feature selection using a multilayer perceptron. *Journal of Neural Network Computing*, 2(2):40–48, 1990.

[16] A. Buades, B. Coll, and J. Morel. Non-local means denoising. *Image Process. Line*, 1, 2011.