Laser Beam Quality Classification

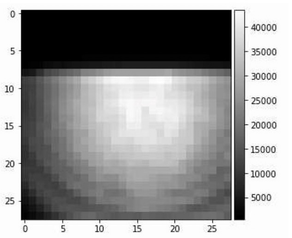
Mengzhu (Zoey) Sun
https://github.com/zoey1124/laser_beam_quality

Problem we try to solve & Background

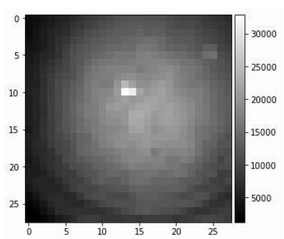
Laser beam scientists build lots of cameras around the laser beam injector to catch the image of laser beam and detect the quality of it. If there is aberration in the image, the laser beam team need to detect it and figure out what went wrong.

In this project, an algorithm was developed to automate the process of laser beam image aberration detection, and aberration category classification. First using threshold comparison to detect aberration image, then an ML algorithm will classify which aberration it is.

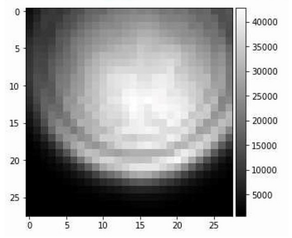
Image Visualization and Preprocess



Clipped edge

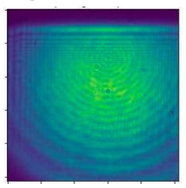


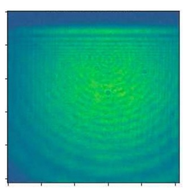
Hot spot



Airy ring

There are mainly 3 types of aberrations aberrations: clipped edge, hot spot, and airy ring.





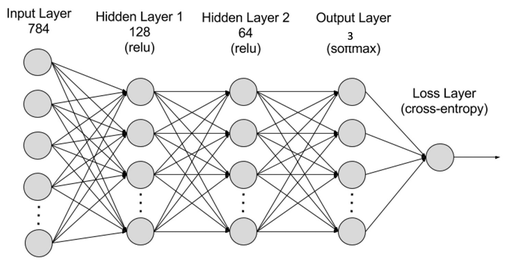
The data augmentation functions include clipping image to square, flipping, rotating, random cropping, and adding gaussian noise. Using the right images as an example, the bottom image is transformed via adding noise from the above one.

ML Model for Aberration Classification

In order to save computational resources, resize the training data resolution to 28 x 28 before feeding into the model.

Tried 2 different models: baseline_1 and baseline_2. The accuracies and training times of models can be referred to the right form.

During the training process, model is suffering from lack of enough training data, so *oversampling* strategy is used, which means sampling the minority category (aberration images) more frequently to balance the distribution of training dataset. Noticeably, oversampling can reach the same effect as adding weight on loss function to misclassifying the minority groups but has better stability.



The image above show the general structure for the CNN model (baseline_2). There are 2 hidden layers, each contains a linear forward layer, Dropout, and ReLU activation. The final layer use Cross Entropy Loss as criterion and stochastic gradient descent (SDG) as optimizer.

How to detect aberration? – Threshold Comparison

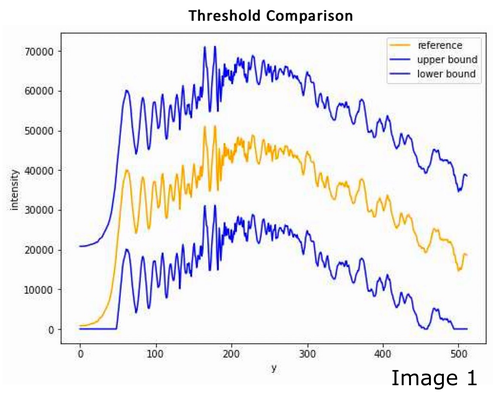


Image 1

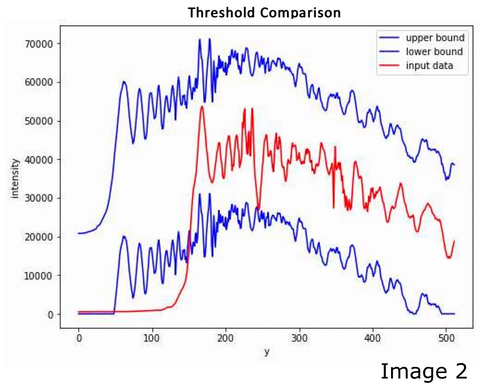


Image 2

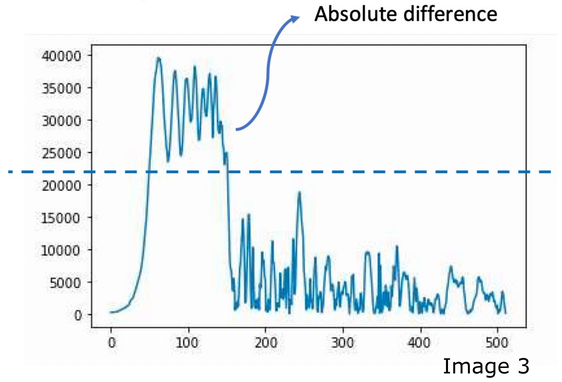


Image 3

We use threshold Comparison for detecting abnormal images. As image 1 shown, the average image of camera (yellow line) is used as a reference, setting threshold value range above and below the reference (blue lines), anything beyond this range will be reported as aberration.

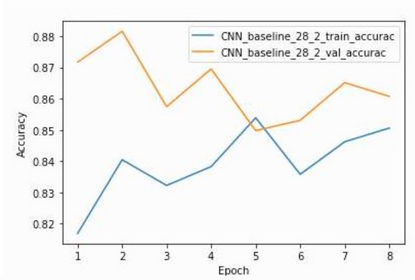
When running the algorithm, first get the absolute difference between testing input image and referenced image, then drawing a line using threshold value, if seeing any part surpass the threshold, then report the image as aberration (image 3).

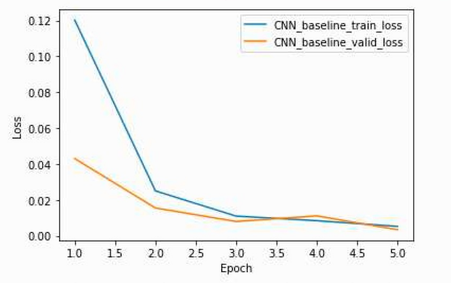
The threshold comparison saves computation resources compared with ML mode, so we use it to detect aberration first, then only the abnormal images will go to ML model.

Accuracy & Loss

model	Validation accuracy	Training time	Testing time (per image)
Baseline_1	0.88	1232s	0.85s
Baseline_2	0.86	1969s	0.96s

The general testing time for only one image is within 1 sec with 1 machine, which makes the model practical into use for cameras that captures images with ~ 1Hz frequency.





With enough training data, the loss and validation accuracy of the model is stabilized after ~ 5 epochs.

Acknowledgments

Special thanks for Tyler Johnson, Joe Robinson, and Eric Cunningham for many helps.