

# SVM Learning for GFIS Trimer Health Monitoring in Helium-Neon Ion Beam Microscopy

Andrew B. Kahng<sup>†‡</sup>, Hsin-Yu Liu<sup>‡</sup>, Chris Park<sup>+</sup>, Ramani Pichumani<sup>+</sup>, and Lawrence Saul<sup>†</sup>

<sup>†</sup>CSE and <sup>‡</sup>ECE Departments, UC San Diego, La Jolla, CA, USA

<sup>+</sup>Carl Zeiss SMT Inc., Pleasanton, CA, USA

{abk, hyl001, saul}@ucsd.edu, {chris.park, ramani.pichumani}@zeiss.com

## I. MOTIVATION

Helium-Neon ion beam microscopy (HIM) has shown significant promise in the areas of nanoscale imaging, editing, and patterning of semiconductor structures. The unique capabilities of this technology are enabling new applications in semiconductor process control by exploiting the ability to characterize materials in the sub-5 nanometer spatial scale with high speed. These instruments utilize a gas field ion source (GFIS) trimer with an atomically sharp metal tip to generate a stream of charged particles that are used for imaging and milling. As these applications move into high-throughput fabrication and measurement domains, it becomes increasingly important to optimize the stability and reliability of the imaging column to achieve the most useful results. One critical factor impacting repeatable tool operation is the ability to maintain the precise atomic structure of atoms comprising the source GFIS trimer. In this work, we propose the first machine-learning-based support vector machine model to monitor whether the current GFIS trimer is healthy or not for imaging. With an original dataset of 726 images our model achieves precisions of 97% for the “Healthy” class and 98% for the “Unhealthy” class with leave-one-out cross-validation. Even with a large amount of augmented data (21X of the original data) as input (total of 15246 images), we still reach precisions of 97% for the “Healthy” class and 95% for the “Unhealthy” class. The significance of this work is the ability to support in-line predictive maintenance, optimal performance, and equipment productivity in semiconductor fab environments.

## II. APPROACH

Scheduling-based preventative maintenance practices using historical reliability data or key equipment performance metrics are commonly used in the semiconductor equipment industry. Scheduling-based approaches, however, typically lead to over- or under-maintenance which reduce system availability and uptime, respectively. There is a growing trend for predictive and proactive approaches. Temporal dynamics of equipment parameters may be used to predict future events and leveraged as an indicator to spur proactive maintenance. In certain cases, temporal dynamics may not be sufficient and require image analytics supplementation.

One such use case is monitoring the health of a *gas field ion source* (GFIS). A GFIS can be used to generate focused ion beams (FIB) and has shown to reliably produce a focused probe with a spot size of 0.5nm for helium FIB as in Fig. 1 [1]. The condition for achieving best GFIS brightness and current stability is when the GFIS terminates with three atoms at its apex called a *trimer*. Traditional operation requires a human operator to examine the source to determine the state of the trimer. Characterization of the trimer state is not limited only to whether the trimer is intact, but also encompasses a search for indicators which pose a threat to trimer stability such as adsorption of contaminant oxygen molecules to the tip substrate. The latter discrimination requires expertise and may lead to mislabeling of the trimer state. Robust trimer state identification is required to maximize tool availability.

To this aim, we propose a fully automated methodology for analyzing the quality of the GFIS to eliminate human error. The tool provides a convenient method for a way of imaging the trimer configuration through an emitter reflection field which reconstructs atomic formation in the image sensor. To accurately identify the state of the GFIS trimer, Support Vector Machine (SVM) binary classifiers are trained to determine the state of a real-time captured image of GFIS trimer. SVM is a traditional machine learning model for binary classification [3], a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. SVM is a popular statistical model in the machine learning field.

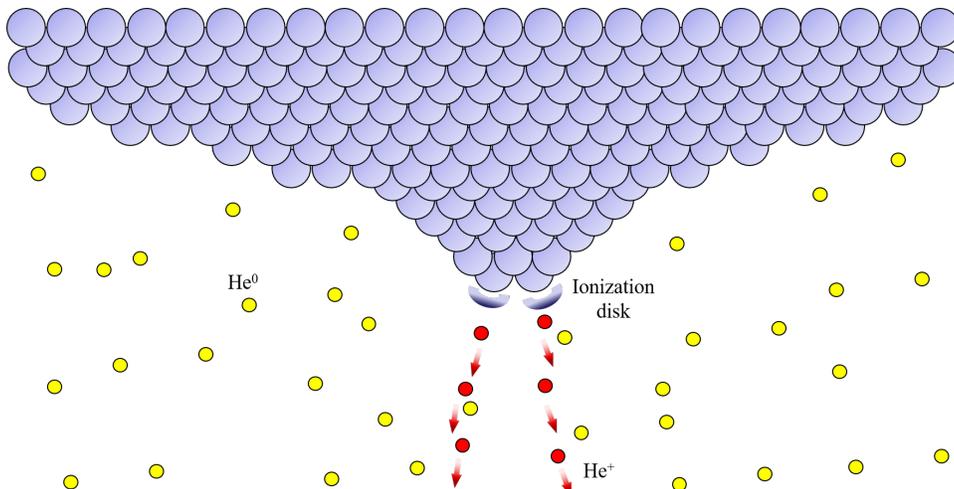


Fig. 1: GFIS schematic. Ionization occurs mostly at the most protruding corner and edge atoms [1].

### III. PROBLEM STATEMENT

Given a dataset of GFIS images, our goal is to automatically monitor and identify whether the current source is healthy for imaging or needs to be reformed for better imaging quality of the Helium-Neon ion beam microscope.

**Input:** A dataset of captured GFIS images.

**Output:** A classifier that accurately identifies whether current trimer is “Healthy” or “Unhealthy”.

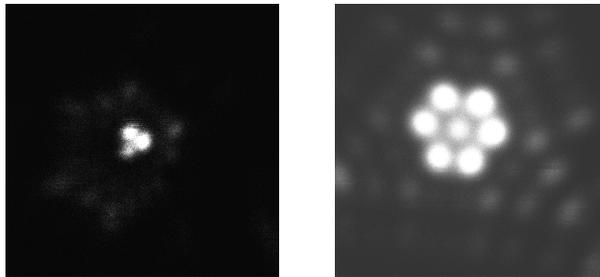
**Objective:** Avoid human intervention and operation of source monitoring.

**Methodology:** Supervised machine learning model training, validation, model improvement and optimization.

In this work we use a Support Vector Machine to classify two classes of images based on the labeling by experts.

### IV. CLASSIFICATION FRAMEWORK

In this section, we first present preliminary steps of data preparations which converts image data into meaningful and essential inputs into our Support Vector Machine for image classification.



(a)  $\langle 111 \rangle$ -oriented single-crystalline nanotip consists of three atoms. This is an example of a “Healthy” trimer [2].

(b) As the atoms of the topmost layer escape, the second atomic layer is exposed, which indicates the source needs to be repaired. This is an example of an “Unhealthy” trimer.

Fig. 2: Labeling “Healthy” and “Unhealthy” GFIS trimers.

#### A. Labeling

Labeling typically takes a set of unlabeled data and augments each piece of that unlabeled data with meaningful, informative tags. The original dataset is labeled by domain experts with deep understanding of GFIS atomic structures and its formation and transformation through time. Data are labeled into two classes: “Good Trimers” and “Indication Trimer Needs Reforming”, which we respectively classify as “Healthy” and “Unhealthy” in our problem for simplicity. 726 images are collected with 608 images labeled as “Healthy”, e.g., as shown in Fig. 2(a) and 118 images labeled “Unhealthy”, e.g., as shown in Fig. 2(b). Since there are only two classes in our problem, it is a classical binary image classification problem in the machine learning field.

#### B. Image Pre-Processing

Image pre-processing refers to operations with images at the lowest level of abstraction [13], where both input and output are the intensity of the images as represented by a matrix of image brightness values. The aim of pre-processing is to improve the image data by suppressing distortions and enhancing image features that are critical for further processing, namely, for machine learning model training, validation and testing. Common transformations used for image pre-processing are rotation, scaling, and translation. We use *OpenCV* [15] as the engine for pre-processing in this work. We also use cropping and downsampling to reduce the dimensionality of data.

1) **Cropping:** Most data are appended with machine parameters which are uninformative and add unnecessary dimensionality to our classifier. They are inserted to record the parameters applied at the moment the image is captured, e.g., working distance, GFIS aperture size, timestamp, etc. For image pre-processing, we crop out any appended information and convert the image into a square (see Fig. 3). Cropping reduces the dimensionality of the data needed for our model.

2) **Downsampling:** “The Curse of Dimensionality” [12] occurs when the sample density decreases exponentially with the increase of dimensionality. It likely leads to overfitting with comparatively few training samples. Therefore, to reduce the dimensionality of the feature space, we downsample the pixels from original images ( $512 \times 512$ ) to half along each axis ( $256 \times 256$ ) with Gaussian pyramid construction [15]. This process maintains the image analysis quality, and reduces computation cost and complexity [10].

#### C. Support Vector Machine

A Support Vector Machine (SVM) is a machine learning algorithm designed to solve binary classification problems [3] with very high dimensional covariates and relatively small amounts of data samples [5]. It determines an optimal hyperplane that separates the training data with a maximal margin. SVM is primarily used for pattern recognition and can handle smaller datasets with high dimensions.

This matches our problem, which has comparatively small amount of data (726 images) with high dimensionality ( $256 \times 256$ ). With a training and test set split ratio of 70% (508 images) : 30% (218 images), we could achieve precisions of 97% for both “Healthy” and “Unhealthy” classes.

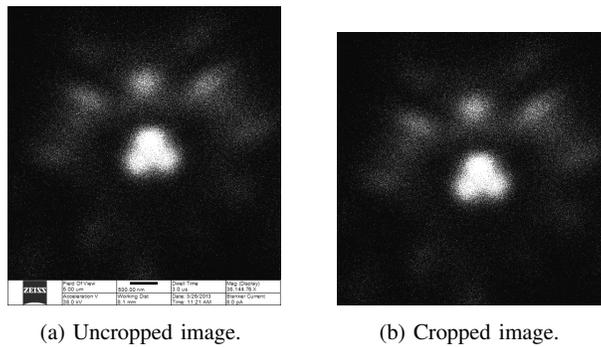


Fig. 3: Example of image cropping.

Our SVM model is based on the *soft-margin* SVM [3], instead of *hard-margin* [11]. Soft-margin SVM is an extended version of hard-margin SVM. Hard-margin SVM only works well when data is completely linearly separable without any errors (noise or outliers). In case of errors, either the margin is smaller or hard-margin SVM fails. On the other hand, soft-margin SVM was proposed to solve this problem by introducing slack variables.

We have also validated several Convolutional Neural Network (CNN) models with different architectures and hyperparameter combinations. However, accuracy achieved in our training/test data with our CNN models are not as good as with SVM. There are a number of potential reasons for this, including insufficient amount of original data (CNN performs better when the dataset is relatively large), vulnerability to overfitting, and over-complicated network structures.

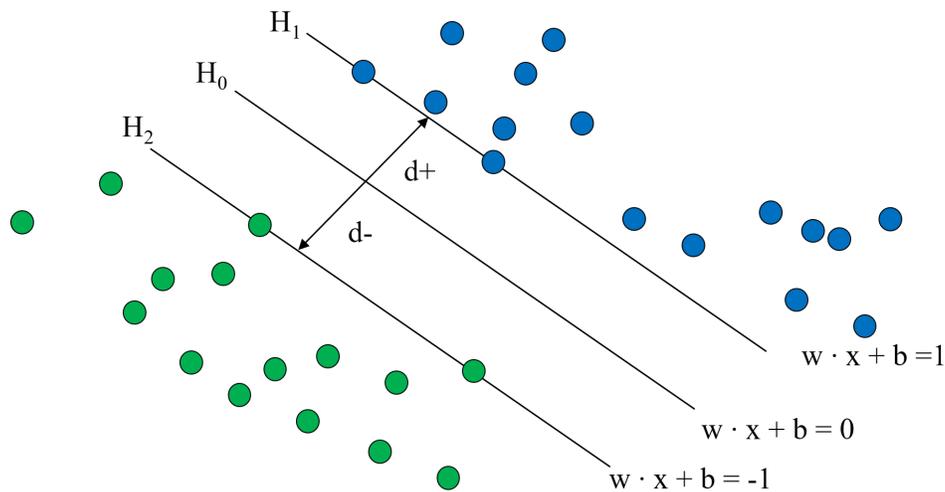


Fig. 4: Schematic of SVM hyperplane separating two classes with maximal margin [17].

In Fig. 4,  $w$  is the weight vector,  $x$  is the input vector, and  $b$  is bias. For  $H_1$ ,  $w \cdot x_i + b \geq +1$ , when  $y_i = +1$  and for  $H_2$ ,  $w \cdot x_i + b \leq -1$ , when  $y_i = -1$ .  $H_1$  and for  $H_2$  are the hyperplanes for separating data points apart and  $H_0$  is the median of  $H_1$  and  $H_2$  where  $w \cdot x_i + b = 0$ . The distance between  $H_0$  and  $H_1$  is  $|w \cdot x + b| / \|w\| = 1 / \|w\|$ . Total distance between  $H_1$  and  $H_2$  is  $2 / \|w\|$ , where  $d+$  is the distance between  $H_0$  and  $H_1$ , and  $d-$  is the distance between  $H_0$  and  $H_2$  [17]. In order to maximize the margin,  $\|w\|$  needs to be minimized.

To ensure the robustness and generalization of our models, we implement image pre-processing, data augmentation, model optimization and diagnosis for our models. The entire experimental flow is illustrated in Fig. 5.

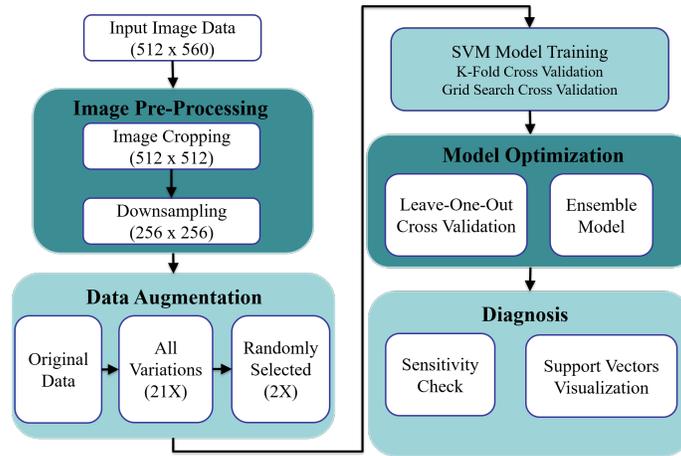


Fig. 5: Experimental flow chart.

## V. COMMON METRICS FOR CLASSIFICATION PROBLEM

We use two common metrics for quality of binary classification problems: classification report [4] and confusion matrix [4] [16]. Details are explained below.

### A. Classification Report

A classification report is one of the commonly used metrics for binary classification problems (e.g., see Table I). Below is the explanation of each term:

**Precision:** The ratio  $TP / (TP + FP)$ , where TP is the number of true positives and FP the number of false positives. Precision measures the ability of the classifier to not mislabel a negative sample as being positive.

**Recall:** The ratio  $TP / (TP + FN)$ , where FN is the number of false negatives. The recall measures the ability of the classifier to find all the positive samples.

**F1-Score:** The harmonic means of precision and recall.

**Support:** The number of occurrences of each class.

**Micro average:** Average of the total true positives, false negatives and false positives.

**Macro average:** Average of the unweighted mean per label.

**Weighted average:** Average of the support-weighted mean per label.

### B. Confusion Matrix

In the field of statistical classification with machine learning, a *confusion matrix*, also known as an *error matrix* [7], is a visualization of the performance of the particular algorithm shown in a table layout. Each row in the matrix represents the instances in a predicted class while each column indicates the instances in an actual class (see Fig. 6).

		P	N
		P	True Positives (TP)
N	False Positives (FP)	True Negatives (TN)	
Actual Class		Predicted Class	

Fig. 6: Sample confusion matrix layout of a binary classification model (transposable).

## VI. RESULTS

The ability and effectiveness of the machine learning model can be evaluated with cross-validation (CV). This also can be seen as a re-sampling process to evaluate a model when there is limited data. The most widely used CV methodology is *train-test-split*, which randomly splits the data with a specified ratio of training and test data, typically 70 : 30 or 80 : 20. We choose 70 : 30 because too few test data can lead to overfitting. This is also a baseline of our model. We separate the entire dataset into 70% of training data and remaining 30% of test data.

1) **K-fold Cross-Validation:** K-fold cross-validation (K-fold CV) divides the training dataset into K subsets. One subset is used as a test set, while the remaining K-1 subsets are used as a training set. The average error of all K trials is calculated. The benefit of this methodology is that as the number of K is increased, the variance of the resulting estimate is reduced. In our experiments we use 10-fold cross-validation as the baseline. Results are shown in Figs. 7, 8, and 9.

$$CV(\hat{f}) = \frac{1}{N} \sum_{i=1}^N L(\hat{y}_i, \hat{f}^{-\kappa(i)}(x_i))$$

$CV(\hat{f})$  is the cross-validation estimate of prediction error,  $N$  is the number of samples, and  $\kappa: \{1, \dots, N\} \mapsto \{1, \dots, K\}$  is an indexing function that indicates the partition to which observation  $i$  is allocated by the randomization. Denoted by  $\hat{f}^{-\kappa(x)}$  is the fitted function, and  $L$  is the loss function, computed with the  $k^{th}$  part of the data removed. Typical choices of  $K$  are 5 or 10 [14].



Fig. 7: Diagram of the K-fold cross-validation.

TABLE I: CLASSIFICATION REPORT OF THE K-FOLD CV MODEL WHERE  $K = 10$

	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Unhealthy	0.97	0.85	0.90	39
Healthy	0.97	0.99	0.98	179
Micro avg	0.97	0.97	0.97	218
Macro avg	0.97	0.92	0.94	218
Weighted avg	0.97	0.97	0.97	218

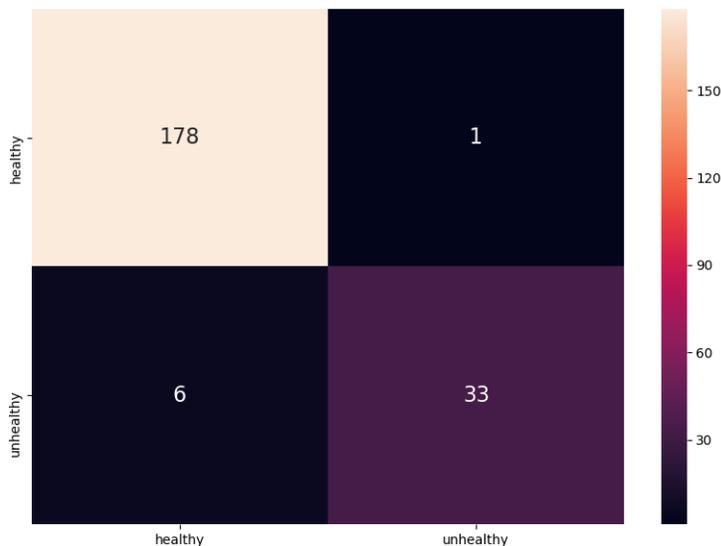


Fig. 8: Confusion matrix of the K-fold CV model where  $K = 10$ .

2) **Leave-One-Out Cross-Validation:** Leave-One-Out cross-validation (LOOCV) is K-fold CV pushed to the extreme, where  $K$  is equal to  $N$ , the number of data points of the entire dataset. The classifier is trained on all the data except for one data point and a prediction is made for that particular “left-out” data point. Evaluation given by LOOCV is promising, but the computation cost is expensive for problems with large amounts of data. Since our original dataset is comparatively small, we utilize LOOCV to observe the precision improvement made. However, the LOOCV computation is costly (even with our small dataset), so we train with 50% of the total amount of data (363 out of 726

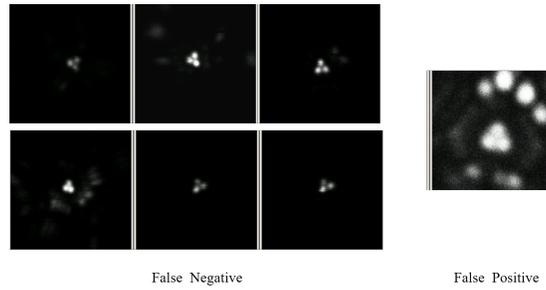


Fig. 9: False-Positive and False-Negative images of the K-fold CV model where K = 10.

images), reserve the remaining 50% as test data, and are able to achieve a classification precision greater than 97% (as shown in Table II and Figs. 10 and 11 below).

TABLE II: CLASSIFICATION REPORT OF THE LOOCV MODEL

	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Unhealthy	0.98	0.87	0.92	62
Healthy	0.97	1.00	0.99	301
Micro avg	0.98	0.98	0.98	363
Macro avg	0.98	0.93	0.95	363
Weighted avg	0.98	0.98	0.97	363

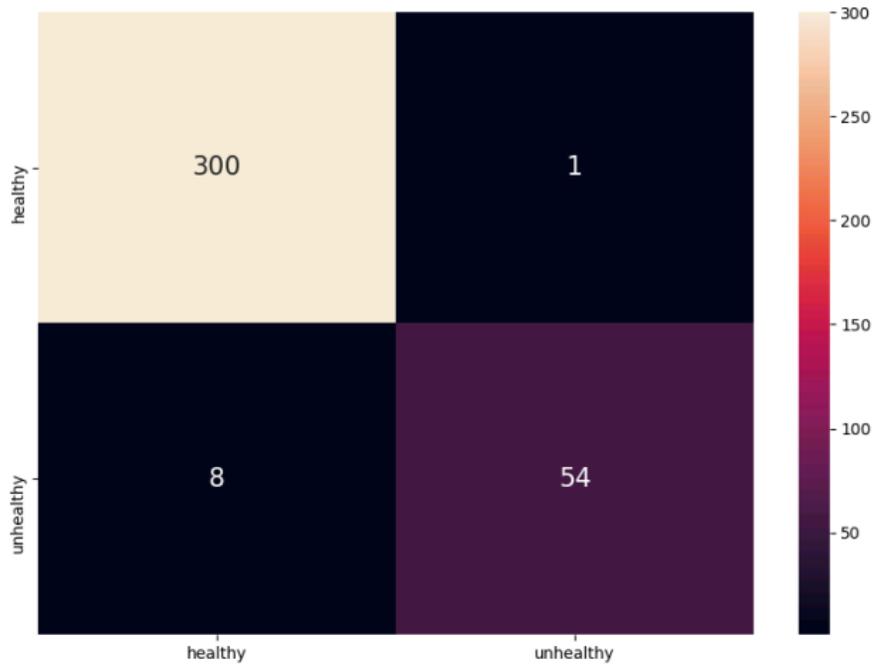


Fig. 10: Confusion matrix of the LOOCV model.

3) *Listing of Falsely Predicted Images*: For the purpose of listing all the false-positive and false-negative image data as reported in the previous section, we implement an algorithm to first record all the labels and corresponding image file paths as a matrix. Then after the train-test-split, we write another matrix with indices of the original matrix indicating the  $i^{th}$  element and corresponding labels. While evaluating the test data of our SVM models, we record the falsely predicted labels, actual labels and corresponding file paths. Details of the False-Prediction Listing algorithm are given below, and notations are explained in Table III.

## VII. MODEL ROBUSTNESS IMPROVEMENT AND DIAGNOSIS

In order to ensure that our model is robust enough to deal with unseen data, we use an ensemble model and data augmentation for model robustness improvement. We extract the number of support vectors and perform sensitivity checks for diagnosis. In classification problems, there are gray areas where data points are the most difficult to be distinguished. In SVM-based classification, the support vectors are those

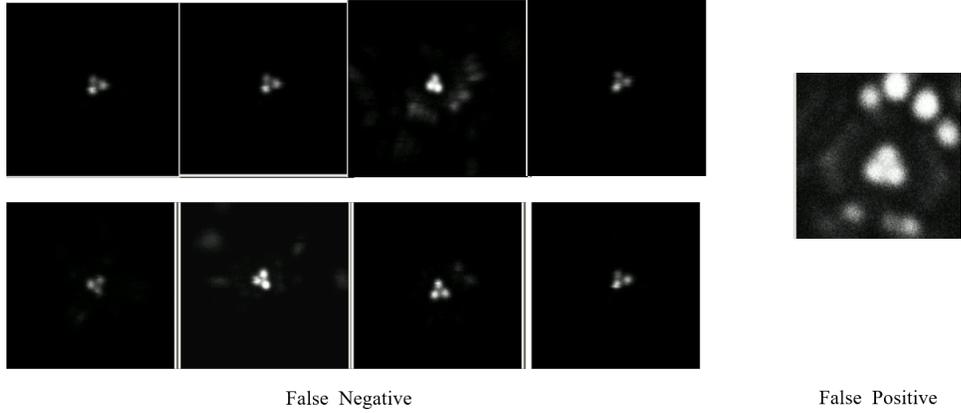


Fig. 11: False-positive and false-negative images of the LOOCV model.

---

**Algorithm** False-Prediction Listing.

---

**Procedure:** *FPListing()*

**Inputs:** A look-up-table matrix  $LUT$  of all data index  $i$  with corresponding file path  $f_i$  and a vector  $y_{test}$  of test data.

**Outputs:** A matrix of actual labels, predicted labels of SVM model as well as corresponding file paths of falsely predicted data.

```

1:  $indices_{FP} \leftarrow \emptyset$ ;  $CR \leftarrow \emptyset$ 
2: read in  $LUT$  and  $y_{test}$ 
3: for all index  $i$  in the  $y_{test}$  do
4:   if  $y_{test} \neq y_{pred}$  then
5:      $CR.write(y_{test}[i], y_{pred}[i], y_{test}[i-1][0], LUT[y_{test}[i-1][0]])$ 
6:      $indices_{FP}.append(i)$ 
7:   end if
8: end for

```

---

TABLE III: NOTATIONS

Notation	Meaning
$indices_{FP}$	Index of falsely predicted data
$CR$	Matrix consists of predicted label, actual label, and data path to be written in classification report
$LUT$	Look-Up-Table matrix of all data path and its corresponding index
$y_{test}$	Vector of test data targets
$f_i$	File paths of the $i^{th}$ data
$y_{pred}$	Vector of predicted labels of test data
$y_{pred}[i], y_{test}[i]$	The $i^{th}$ element of test data target and predicted label of test data

particular points that lie closest to the decision surface (or hyperplane). These points have direct bearing on the optimum location of the decision surface. Furthermore, the number of support vectors, which equals the number of these difficult data points, can be considered to be a representative metric for evaluating this gray area.

#### A. Ensemble Model

The concept of an ensemble model is to take an average of the multiple model decisions as the final decision. Hence the ensemble model can obtain better predictive performance. In this work, we use a voting classifier of four SVM models with different kernel functions: Linear, RBF (Radial-Basis Function), Polynomial and Sigmoid. The kernel function defines inner products in the transformed space to find similarities in the transformed space [17]. In addition, we use a soft-voting mechanism which is also called “weighted average probabilities”. Specific weights can be assigned to each classifier via the weighting parameter. When weights are specified, the predicted class probabilities for each classifier are gathered, multiplied by the classifier weight, and then averaged. The final class label is then derived from the class label with the highest average probability [17]. Experimental results are shown in Fig. 12 and Table IV.

TABLE IV: CLASSIFICATION REPORT OF ENSEMBLE MODEL

	Precision	Recall	F1-score	Support
Unhealthy	0.96	0.71	0.82	66
Healthy	0.95	0.99	0.97	370
Micro Avg.	0.95	0.95	0.95	436
Macro Avg.	0.96	0.85	0.89	436
Weighted Avg.	0.95	0.95	0.95	436

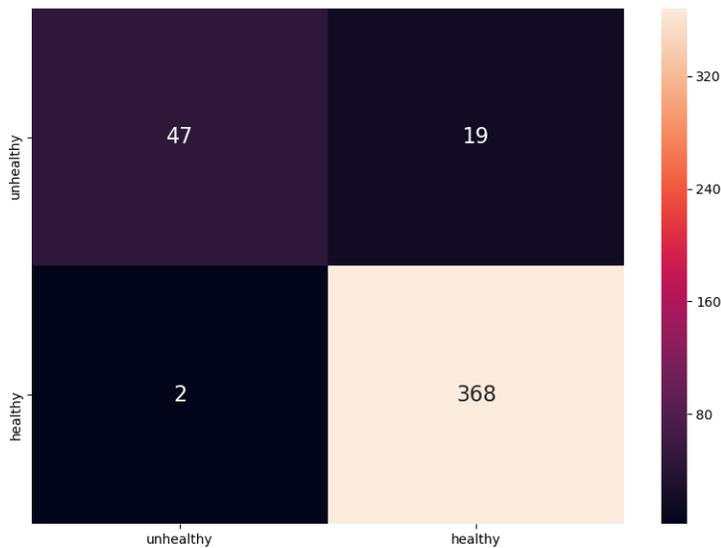


Fig. 12: Confusion matrix of the ensemble model.

### B. Data Augmentation

Data augmentation is a technique widely used in image classification problems and has shown success in increasing the accuracy of classification tasks [8]. In this work we use image transformation for generating augmented data with *OpenCV* by creating various categories of transformed images. For details of augmented data, please see Tables V and VI. We randomly select two times the amount of all augmented data as a subset to avoid memory allocation errors. Furthermore, we utilize the *LIBLINEAR* [6] SVM solver library for large linear classification and are able to manage all augmented data (15246 images). The cross-validation time is significantly reduced by using the *LIBLINEAR* solver. With 10672 training images, 10-fold cross-validation and grid search cross-validation, total training time is 7 hours and 25 minutes in our case (see Fig. 13 and Table VII).

TABLE V: DATA AUGMENTATION

Augmented Attributes	Variations	Number of Data
Augmentation parameter 1	Variation consists of two coefficient expansion: $\alpha: \{1, 1.25, 1.5\}$ , and $\beta: \{-25, 0, 25\}$ .	6534
Augmentation parameter 2	$\{45, 90, 135, 180\}$	2904
Augmentation parameter 3	$\{+32, -32\}$ on x-axis, and $\{+32, -32\}$ on y-axis.	2904
Augmentation parameter 4	$\{0.8, 0.9, 1.1, 1.2\}$	2904

TABLE VI: DATASETS FOR SVM MODELS

Dataset	"Healthy" Images	"Unhealthy" Images	Training Set	Test Set	Total Images
Original	608	118	508	218	726
2X Original	1216	236	1016	436	1452
All Data Augmentation	12768	2478	10672	4574	15246

TABLE VII: CLASSIFICATION REPORT OF ALL AUGMENTED DATA

	Precision	Recall	F1-score	Support
Unhealthy	0.95	0.85	0.90	755
Healthy	0.97	0.99	0.98	3819
Micro Avg.	0.97	0.97	0.97	4574
Macro Avg.	0.96	0.92	0.94	4574
Weighted Avg.	0.97	0.97	0.97	4574



Fig. 13: Confusion matrix of the SVM model trained with all augmented data.

### C. Diagnosis

1) **Sensitivity Check:** In order to diagnose which attribute variation most affects the decisions of our model, we categorize falsely predicted data by their augmented attribute. The sensitivities of these attributes are then ranked: parameter 3, parameter 2, parameter 1, and parameter 4 (see Fig. 14 and Table VIII).

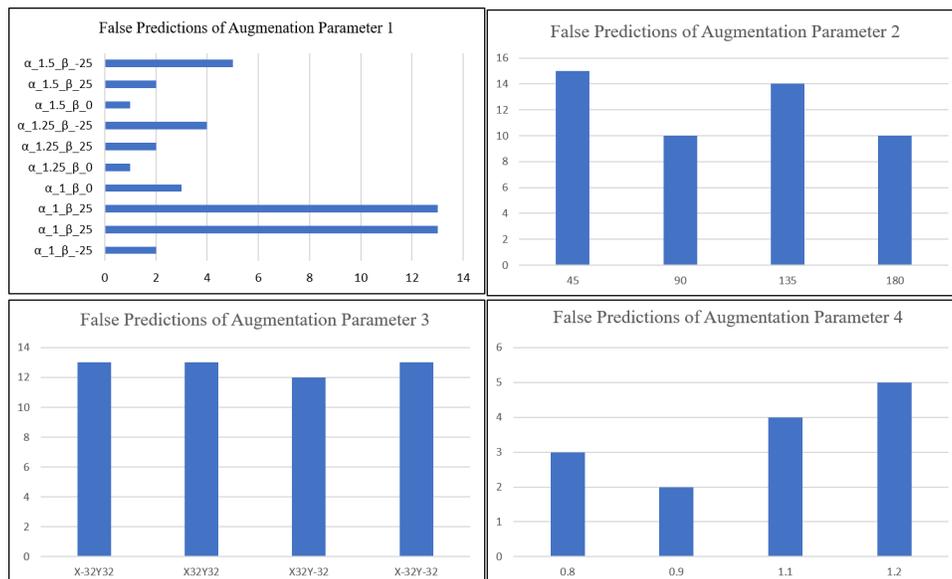


Fig. 14: Sensitivity of model prediction to varied augmentation methods.

TABLE VIII: SENSITIVITY OF AUGMENTED DATA

Category	Number of falsely predicted data	Number of augmented data	Ratio of false prediction data
Augmentation parameter 1	34	6534	0.52%
Augmentation parameter 2	49	2904	1.69%
Augmentation parameter 3	51	2904	1.76%
Augmentation parameter 4	14	2904	0.48%

2) **Number of Support Vectors:** With SVM classifiers, support vectors are the data points which lie closest to the hyperplane. They are the most difficult data points to be classified, a decision function of SVM is fully specified by a subset of training samples, along with the support vectors [17].

In our investigations, a total of 107 support vectors out of 1016 training data points, with 54 images from “Unhealthy” and 53 images from “Healthy”, are seen with the 2X data. Also, a total of 73 support vectors out of 508 training data points, with 31 images from “Unhealthy” and 42 images from “Healthy”, are seen with the original data. These results indicate that the performance of our model is acceptable, that the labeling is accurate, and that we have a clean dataset with few anomalies [9].

### VIII. CONCLUSIONS

In this work, we have proposed the first Support Vector Machine for helium-neon ion beam microscope GFIS trimer health monitoring, achieving high accuracy (over 95% for both “Healthy” and “Unhealthy” classes.)

In the image-based predictive maintenance context, storage capacity may limit the amount of data used. When a relatively small amount of data is available, data augmentation may be useful to assure the robustness and predictability of training models. This work has demonstrated the use and the viability of data augmentation to enhance model generalization. We have verified our model capability with various methodologies: (i) examination of individual false-positive and false-negative data points, (ii) extracting the number of support vectors, (iii) data augmentation, (iv) leave-one-out cross-validation, and (v) ensemble modeling. In addition, sensitivity diagnosis is used to determine how each imaging attribute variation impacts the prediction results. The sensitivity diagnosis furthermore suggests how the process of monitoring can lead to a more accurate prediction.

In future work, we could explore categorization of data which is “Marginally Unhealthy”, i.e., as a third class. However, additional effort for data collection and labeling would be required. We believe that our modeling approach can be portable to other instruments and contexts with similar binary image classification, e.g., maintenance and atomic contamination monitoring. However, development of relevant data labeling, image preprocessing, and data augmentation methodologies (if not enough data is collected) would be necessary. After implementing the model on a different dataset or for a different application, if the prediction performance is not as expected, hyperparameters, model ensembling, bootstrapping, bagging, and kernel functions could be considered to improve model accuracy. And lastly, Real-Time Machine Learning methods could be very useful for integrating newly collected data into the model dynamically, so as to enhance model performance over time.

### REFERENCES

- [1] R. V. Gastel, G. Hlawacek, B. Poelsema, and V. Veligura, “Helium Ion Microscopy”, *Journal of Vacuum Science & Technology B, Nanotechnology and Microelectronics: Materials, Processing, Measurement, and Phenomena*, 32.2 (2014): 020801.
- [2] T. Asai, K. Hata, T. Iwata, K. Kajiwara, K. Kajiwara, S. Nagai, and Y. Suzuki, “Stability Measurement of Neon Ion Beam Current Emitted from Supertip Gas Field Ion Source”, *25th International Vacuum Nanoelectronics Conference. IEEE*, 2012, pp. 1-2.
- [3] C. Corinna and V. Vapnik, “Support-Vector Networks”, *Machine Learning*, 20(3) (1995), pp. 273-297.
- [4] M. Blondel, M. Brucher, D. Cournapeau, V. Dubourg, E. Duchesnay, A. Gramfort, O. Grisel, V. Michel, A. Passos, F. Pedregosa, P. Prettenhofer, M. Perrot, B. Thirion, J. Vanderplas, G. Varoquaux, and R. Weiss, “Scikit-learn: Machine Learning in Python”, *Journal of Machine Learning Research*, 12(Oct) (2011), pp. 2825-2830.
- [5] G. M. Foody and A. Mathur, “The Use of Small Training Sets Containing Mixed Pixels for Accurate Hard Image Classification: Training on Mixed Spectral Responses for Classification by A SVM”, *Remote Sensing of Environment*, 103.2 (2006), pp. 179-189.
- [6] K.-W. Chang, R.-E. Fan, C.-J. Hsieh, C.-J. Lin, and X.-R. Wang, “LIBLINEAR: A Library for Large Linear Classification”, *Journal of Machine Learning Research*, 9.Aug (2008), pp. 1871-1874.
- [7] S. V. Stehman, “Selecting and Interpreting Measures of Thematic Classification Accuracy”, *Remote Sensing of Environment*, 62.1 (1997), pp. 77-89.
- [8] L. Perez, and J. Wang, “The Effectiveness of Data Augmentation in Image Classification Using Deep Learning”, *Convolutional Neural Networks Vis. Cognit.* (2017).
- [9] C. J. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition”, *Data Mining and Knowledge Discovery*, January 1998 Vol 2, pp. 121-167.
- [10] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden, “Pyramid Methods in Image Processing”, *RCA Engineer*, 29(6) (1984), pp. 33-41.
- [11] B. E. Boser, I. M. Cuyon, and V. N. Vapnik “A Training Algorithm for Optimal Margin Classifiers.” *Proceedings of The Fifth Annual Workshop on Computational Learning Theory*. ACM, 1992., p. 144-152.
- [12] C. M. Bishop and S. Markus, *Pattern Recognition and Machine Learning*, New York City, Springer, 2007.
- [13] R. Boyle, V. Hlavac, and M. Sonka, *Image Pre-Processing. In: Image Processing, Analysis, and Machine Vision*, Boston, Springer, 1993.
- [14] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*, New York, Springer Series in Statistics, 2001.
- [15] K. Adrian and B. Gary, “OpenCV”, *Learning OpenCV*, 2008, pp. 1
- [16] J. D. Hunter, “Matplotlib: A 2D Graphics Environment”, *IEEE Computing in Science & Engineering*, 9(3) (2007), pp. 90-95.
- [17] R. Berwick, “An Idiot’s Guide to Support Vector Machines (SVMs)” - MIT, <http://web.mit.edu/6.034/wwwbob/svm-notes-long-08.pdf>