



This is an electronic reprint of the original article. This reprint may differ from the original in pagination and typographic detail.

Cenev, Zoran; Venäläinen, Janne; Zhou, Quan

# Vision based event classification in robotic micromanipulation

Published in: Proceedings of the 2017 International Conference on Manipulation, Automation and Robotics at Small Scales, MARSS 2017

DOI: 10.1109/MARSS.2017.8001923

Published: 01/01/2017

Document Version Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:

Cenev, Z., Venäläinen, J., & Zhou, Q. (2017). Vision based event classification in robotic micromanipulation. In *Proceedings of the 2017 International Conference on Manipulation, Automation and Robotics at Small Scales, MARSS 2017* Article 8001923 IEEE. https://doi.org/10.1109/MARSS.2017.8001923

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

# Vision based event classification in robotic micromanipulation

Zoran Cenev, Janne Venäläinen and Quan Zhou Department of Electrical Engineering and Automation, School of Electrical Engineering, Aalto University. Espoo, Finland zoran.cenev@aalto.fi; janne.venalainen@aalto.fi; guan.zhou@aalto.fi

Abstract— Autonomous positioning of small objects to create heterogeneous structures has great potential to advance the current micromanipulation procedures. To achieve autonomous micromanipulation, it is required to recognize the manipulation events. In this work, different classification algorithms including five common supervised learning methods are assessed for identifying states of manipulation. The classifiers are trained with data that consists of 3056 video frames and validated on 2545 videos frames. The best machine learning classifiers classified the events with 92.9 % accuracy, higher than the result of logic-based classification (88.9 %).

Keywords — event classification; micromanipulation; machine vision; supervised learning.

### I. INTRODUCTION

Microhandling represents a set of operations such as positioning, dissection, injection, aspiration and assembly to name a few, for manipulating microscopic objects [1]. These methods have been applied in handling of single fibers [2]–[4], construction of complex micromechanical structures [5], automatic inspection of microelectronic components [6], manipulation of single microscopic biological organisms [7], etc. In the micromanipulation environment, the objects are often fragile and can be deformed easily. Autonomous intelligent micromanipulation is foreseen as one promising solution. To achieve autonomous intelligent manipulation, abstract representation of the state of the manipulation is critical.

However, there exists no research that exploits machine learning (ML) classification methods to infer abstract representation of the state of the robot or manipulated object in micromanipulation. In this work, a comparison of five different supervised learning algorithms for event classification in robotic microhandling of objects (SU-8 chips in this case) is presented. The hardware configuration of the microrobotic system (Fig. 1) and the software integration for the vision-based detection of the chips are elaborated in our previous work [8]. The event classification describes whether a manipulated chip has been correctly handled or not. The correct handling refers to successful release of the chip, while the incorrect handling refers to chip adherence to one of the gripping tips. Initially, relevant data is extracted from videos with image processing techniques. Then, the data is abstracted into features that are significant for the learning. Finally, different supervised classification learning algorithms are assessed with the features and labels (grasped/adhered event). The classifiers, therefore, can predict the outcome of the event within the novel data. The scope of this



Fig. 1. Microrobotic system for manipulating SU-8 chips. (a) Hardware configuration: top and side view microscopes, a gripper and SU-8 chips

work is limited to perception; the microrobot is not controlled based on the classification results.

The rest of the paper is organized in the following way: Section 2 gives a brief overview of the utilized learning algorithms; the designation of events, the data acquisition and feature engineering are presented in Section 3; Section 4 explains how each classification algorithm has been implemented; Section 5 presents the obtained results along with discussion; and Section 6 concludes the paper.

### II. BACKGROUND

The utilized learning classifiers are naive Bayes [9], tree learning [10], support vector machine [11], artificial neural networks (ANN) [12] and boosting ensemble classifier [13]. In Naive Bayes classifiers the most likely class is assigned to a given example described by its feature vector. Three learning classifiers use decision splits to create a model that predicts the value of a target variable. Support Vector Machine is nonprobabilistic binary linear classifier that generates separating hyperlines (with maximized width) between two or more separate classes. ANN is a classifier consisting of collection of neural units (artificial neurons) interconnected in layered architecture. Boosting ensemble classifier is a set of weak (computationally efficient) classifiers which classifies new data by weighting the predictions of each individual weak classifier.

There are no commonly accepted rules on how to select a specific learning classifier for a given task before applying the methods to the actual data. Ali and Smith generated guidelines for the selection task based on some features of the data [14]. However, the learning algorithms are selected by trial-and-error based on the prediction accuracy for the data in question [15].

Before their evaluation, the classifiers in this work are optimized in order to find the most suitable one for the examined event.

# III. EVENTS, DATA AND FEATURES

# A. Events for classification

The detection of proper grasp is a relevant and common task in the control of pick-and-place based robotics, especially in the micrometer scale where the adherence of objects is less predictable compared to macro-scale. Fig. 2 illustrates the two example events (labels in the feature matrix) which were selected for evaluating the different classification methods: (1) an event where the chip is being grasped (grasped-event) and (2) an event where a chip is adhered to a gripper tip (adhered-event) during its release.

Description of grasped-event:

- In top view, the distance between the two gripping tips, (gripper opening) must be similar to the size of the chip.
- In top view, the distances between the gripper tips and the sides of the chip must be small.
- In side view, the y-positions of gripper tips and the chip must be similar.
- In side view, the y-velocities of the chip and the gripper must be similar (to ensure that the chip is not slipping).
   Description of adhered-event:
- In side view, the y-positions of gripper tips and chip must be similar.
- In top view, the gripper opening must not be similar to the size of the chip.
- In top view, the distance between the chip and one of the tips must be small.

In top view, the magnitude of the y-velocities of the chip and the gripper tips must be similar and pointing to a proper direction. The proper direction depends on which tip the chip is being adhered to: if the chip is adhered to the upper gripper tip, then the direction of velocities of both the tip and the chip must be upwards.

### B. Data

The data for training and validation consists of 44 videos with a total of 5601 frames. The videos include occurrences of both of the events. The videos were recorded during micromanipulation of a dummy SU-8 chip with dimensions of  $250 \times 250 \times 30 \ \mu\text{m}^3$ . The videos were split to separate training and validation sets containing 23 videos (3056 frames) and 21 videos (2545 frames), respectively. The division was done in such a



Fig. 2. Illustration of events for classification. (a) Grasped-event while chip is being manipulated above the platform. (b) Adhered-event while chip is being adhered to one of the gripping tips during release.

way that both sets contain approximately equal amount of instances of both events. In addition, both sets contain all the variations of the events that have occurred during the collection of the data.

Trajectories of the objects of interest (gripper tips and manipulated chip) are obtained with the detection algorithms that are described in [8]. The gripper tips were tracked with two vision detectors in the top view and one detector from the side view. The manipulated chip was tracked with one vision detector from each view. The gripper trackers were returning (x, y) image pixel coordinates. The chip tracker was returning the position and the yaw angle of the chip (rotation around the optical axis of the top view). The data also contained the time of each pose from the beginning of the videos. The data was skewed, especially with the adhered-event.

In the training data of the adhered-event, for each positive sample (the event is happening), there were approximately 11 negative samples. In the validation data, however, the ratio was approximately 1:4. Some classifiers, especially probabilistic ones such as naïve Bayes, are prone to skewed data. Therefore, the training data was balanced with direct oversampling that creates duplicates of the minor class until the data is balanced. The validation data was not resampled because the used performance metric for the classifiers is robust against skewed data. From the many existing resampling methods, direct oversampling was selected because it does not dispose data, unlike e.g. random under sampling.

# C. Features

The classifiers were trained with features that describe the relations of the objects. The processed image/video frame data (coordinates of objects) was used for defining the features. Before the features were calculated, the data was smoothed with a moving average filter with a window size of 5 data points. The smoothing was performed because it reduces the amount of noise in the trajectories, which is important especially when calculating the derivatives of the trajectories, i.e. the velocities of objects. Additionally, the trajectories were converted from pixel to meter units by using the pixels-per- $\mu$ m ratio obtained from the camera calibration procedure.

Features 1 & 2: the minimum distances between the gripper tips and the chip in the top view. Features 1 and 2 are the minimum distances between the manipulated chip and the tip of the upper gripper and the tip of the lower gripper, respectively, Fig. 3a. They are measured from the data that is extracted from the top view camera. The motivation for using these two features arises from the descriptions of the events: if the grasped-situation occurs, both of the gripper tips are in contact with the chip (*d* is small for both tips); if the adhered-situation occurs, at least one of the tips is in contact with the chip (*d* is small for one of the tips and large for the other one).

Feature 3: gripper opening in top view. The third feature is the distance between the gripper tips in the top view. First, the distance in  $\mu$ m is calculated with Euclidean norm. Then, the distance was normalized with respect to the manipulated chip size. The normalization was performed in such way that when the distance is equal to the chip size then the value of this feature is 1. When the gripper is fully closed, the value is 0. This feature was used in adhered-event identification because the event cannot be observed unless the gripper has been opened more than the size of the chip. **Feature 4: gripper-chip alignment in side view.** The fourth feature is the vertical distance *d* from the gripper tips to the chip in the side view, Fig. 3b. The distance is obtained by comparing the position of the gripper tips to the edges of the chip. The top and bottom edges of the chip are calculated by adding and subtracting, respectively, the half of the known chip height to the y-value of the detected center-coordinate.

The sign and value of d is determined by the three regions illustrated in Fig. 3b. If the gripper is above the chip, the sign of d is positive, and if the gripper is below the chip, the

sign is negative. If the gripper tips are in between the top and bottom edges of the chip, d is zero. This feature is useful in determining whether the grippers are in contact with the chip.

The contact cannot be inferred from the top view alone, even if the values of the features 1, 2, and 3 suggest a contact, because the gripper might be at a wrong height.

Feature 5-11: velocities of objects of interest. Velocities of objects of interest are required in the detection of slipping of the chip when is being grasped. Additionally, they are required in the adhered-event classification. For instance, the two events appear to be occurring in Fig. 4, where a-b depict the graspedevent and c-d the adhered-event. However, it cannot be inferred from the static position in the figure that the chip is not grasped, meaning the chip is slipping from a gripper tip while being lifted. In order to infer slipping, it is required to measure the velocity of the chip with respect to the gripper. Furthermore, from the static position images c-d, it cannot be observed whether the chip is adhered to the gripper. Although it is possible to observe that there is a contact between them, the velocities need to be examined in order to infer that they are adhered. Hence, images from at least two instances of time are required to infer a proper grip. Therefore, seven velocities presented in Table I were considered as features.

The velocities v were calculated with backward difference

$$v(i) = \frac{P(i) - P(i-1)}{t(i) - t(i-1)}, \qquad i = 1, \dots, n$$
(1)

where *n* is the length of the data and P(i) is the position  $(x, y \text{ or } \theta)$  at the time t(i). The time of each instance of position data is retrieved from the data file. The initial velocity v(0) is assumed to be zero.

**Feature matrix.** After all the features were generated, they were normalized. The normalization consisted of two phases: mean removal and scaling. The features were normalized because some learning algorithms are prone to fail with non-normalized data. For example, SVM finds the optimal



Fig. 3. Features 1 to 4. (a) The minimum distances  $d_{upper}$  and  $d_{lower}$  between the upper gripper and the lower gripper tips to the manipulated object, respectively;  $d_{opening}$  denotes gripper opening in top view. (b) The different regions that determine the sign and value of the distance *d* in feature



Fig. 4. (a) Grasped-event in top view. (b) Grasped-event in side view. (c) Adhered-event in top view. (d) Adhered-event in side view.

hyperplane in the feature space that consists of all of the features. Hence, if some of the features have values that are significantly larger than the values of other features, they have more weight during the optimization. (Nevertheless, some algorithms, e.g. tree learning, do not require normalization because they do not combine the features into a single space, but rather, they examine the features separately). The mean was removed by first calculating the arithmetic mean  $\mu(f_i)$ , i = 1, ..., 11 of all the elements of a feature vector  $f_i$ , and then subtracting the mean from all the elements. The scaling is calculated by dividing all of the elements of the zero-mean feature vector  $f'_i$  is given by:

$$f'_{i} = \frac{f_{i} - \mu(f_{i})}{\sigma(f_{i})}, i = 1, \dots, 11$$
(2)

The normalization parameters  $\mu(f_i)$  and  $\sigma(f_i)$  were retrieved from the training data and stored to normalize novel data. The parameters could not be obtained from the novel data because they are observed one instance at a time. The features were combined into a matrix where each column corresponds to a feature vector and each row corresponds to an instance of the values of the features in a video frame. Additionally, the corresponding labels (event states: grasped, adhered, none) are added as a column to the end of the matrix. Thus, the form of the matrix is a follows

/feature 1	feature 2		feature 11	<i>labels</i> \	
value 1.1	value 1.2		value 1.11	label 1	
value 2.1	value 2.2		value 2.11	label 2	(3)
1 :	:	۰.	:	: ]	
∖ value n.1	value n.2		value n.1	label n/	

TABLE I. THE SEVEN VELOCITY-FEATURES

Feature number	Feature description
#5	x-velocity of the manipulated chip in top view
#6	y-velocity of the manipulated chip in top view
#7	Angular velocity of the manipulated chip in top view
#8	y-velocity of the tip of the upper gripper in top view
#9	y-velocity of the tip of the lower gripper in top view
#10	y-velocity of the chip in side view
#11	y-velocity of the tips of the gripper in side view

where n is the number of training examples. Two separate feature matrices were created for training and validation.

# IV. IMPLEMENTATION

# A. Logic based classification

Before applying machine learning algorithms to the classification problems, logic based algorithm has been assessed. Considering the descriptions of the events and the extracted features, it was feasible to write rules that classify the events. The pseudo-codes for the grasped-event and the adheredevent are given in Fig. 5 and 6, respectively. Both algorithms were implemented in Matlab. The parameters were optimized in three different ways: hand-picked only, hand-picked and then optimized, and randomly optimized parameters. The optimization was performed with *patternsearch* algorithm from the Global Optimization Toolbox in Matlab. This algorithm finds the minimum of a cost function with linear and non-linear inequalities around initial parameter setting. Classification errors were calculated by comparing the predictions with the labels. These errors were the outputs of the cost functions. The classification errors of the logic-based classification algorithms are presented in Table II. The lowest cumulative classification error was obtained for the randomly optimized parameters. The misclassification for the grasped-event and the adherent-event were 14.2% and 7.84%, respectively.

# B. Naïve Bayes

The implementation of the Naïve Bayes algorithm performed by using the *fitcnb* function in Matlab. The features were normalized before using them within the function. The error was 9.44 % for the grasped-event, and for the adhered event 15.9 %.

# C. Tree learning

The complexity of a tree was controlled by restricting the number of decision splits in the tree. To find the optimal complexity of the tree, different number of maximum splits were used for training. The optimal number of splits was selected by choosing the one that resulted in minimum error. The Matlab function *fitctree* was used for implementation and evaluation. The upper bound for maximum number of splits for optimization was set to a point where the trees do not grow further. Fig. 7a shows the classification errors of validation data of both events with respect to the maximum number of splits. In both events, the learner reached the minimum error with rather few splits, after which it started to overfit. The minimum error for grasped classifier is 8.17 % with 2 splits. The minimum error for adhered classifier is 6.74 % with 4 splits.

# D. Ensemble learning

Different ensemble methods were evaluated with varying number of weak learners to find the most suitable configuration for the two events. The function *fitensemble* in Matlab was used for the assessment. The classifiers for both of the events were trained with the following ensemble learning methods: subspace, boosting and bagging. Fig. 7b shows the classification errors of the grasped-event with different ensemble and weak learning methods. The error is obtained by varying the number of the weak learners in the ensemble classifier that effectively changed its complexity. The best ensemble configuration to



Fig. 5. Logic based algorithm 1: Grasped-event detector

```
Input An instance of features
Output Prediction
Check if gripper and chip are at similar height in side view.
1: if feature 4 < parameter 1 then
Check that gripper opening is not equal to chip size.
      2: if feature 3 < parameter 2 or feature 3 > parameter 3 then
    Check adhesion between the upper gripper and the chip.
     3: if feature 1 < parameter 4 then
      Check that gripper and chip are moving upwards
        4: if feature 6 < parameter 5 then
         Check gripper position in side view.
           5: if feature 8 < parameter 6 then
            6: return adhered-event is occurring
           7: end if
        8: end if
            Check adhesion between the lower gripper and the chip.
        9: else if feature 2 < parameter 7 then
              Check that gripper and chip are moving downwards.
           10: if feature 6 > parameter 8 then
            Check gripper position in side view.
              11: if feature 8 > parameter 9 then
               12: return adhered-event is occurring
              13: end if
           14: end if
     15: end if
  16: end if
17: end if
Some of the features did not have an adequate value.
18: return Adhered-event is not occurring
```

Fig. 6. Logic based algorithm 2: Adhered-event detector.

TABLE II. CLASSIFICATION ERROR FOR LOGIC BASED CLASSIFICATION WITH THREE DIFFERENT WAYS TO OBTAIN THE PARAMETERS OF THE CLASSIFIER.

Hand-picked		Optimized hand picked	Randomly optimized	
Grasped-event	19.1	11.4	14.2	
Adhered-event	14.4	14.3	7.84	

classify grasped-event was boosting with trees with 500 weak learners and with an error of 7.74 %. The error was approximately constant after 500 weak learners. Therefore, 500 weak learners was considered optimal for the grasped-event ensemble. Fig. 7c shows the classification error for adheredevent with different hyper-parameters of the ensemble classifier. In this case, the optimal number of weak learners was between 10 and 100 trees with boosting, with smallest error being 7.10 % achieved with 10 weak learners.

After obtaining the optimal hyper-parameter configurations of the ensemble methods, the learning rate was optimized. Learning rate is a parameter related to the step size in the optimization of the ensemble methods. The smaller it is, the more accurate classifiers are usually obtained. However, the smaller learning rate also means more computational time spent for learning process and more prone to get trapped in local minima. Table III summarizes the optimal hyper-parameters obtained for the boosting ensemble method.

# E. Support Vector Machine (SVM)

Support vector machine (SVM) classification algorithm was implemented and evaluated by using the Matlab function fitcsvm. Three hyper-parameters: kernel function, kernel scale and box constraint were tuned to achieve the smallest classification error. First, two different kernel functions were used for the evaluation: linear and Gaussian. The classification errors for the linear kernel were 8.81 % and 18.9 %; and for the Gaussian kernel 8.14 % and 23.6 % for the grasped-event and the adhered-event, respectively. The kernels with the smallest errors were selected for each case. After the selection of the kernel function, the kernel scale and box constraint were optimized. Box constraint controls the penalty that is related to soft margins. Because kernel scale and box constraint are continuous hyper-parameters, they are optimized with the patternsearch algorithm. The initial box constraint was obtained by testing the performance of the initial model with the upper and lower boundaries. The parameters and results of the optimization are presented in Table IV. The lowest classification error for grasped and adhered situations were 6.11 % and 18.5 %, respectively.

# F. Neural Networks

The last supervised ML classification algorithm to evaluate was artificial neural network (ANN). Different network topologies were assessed, i.e. ANNs with variable number of neurons and variable number of hidden layers. The Neural Network toolbox in Matlab was used for implementation and evaluation. The training data was split into training and validation sets with 70 % and 30 % portions, respectively. Fig. 7d shows the classification errors for ANNs with different topologies. The size of the hidden layers affects the classification error, especially for adhered-event. The number of hidden layers has only a small effect on the error. Nevertheless, the smallest error of  $8.12 \pm 0.87$  % (confidence interval: 68 %) for the grasped-event is achieved with a network consisting of two layers and approximately 200 neurons per layer. The optimal network topology for adhered-event is obtained with a single hidden layer of 5 neurons yielding  $10.2 \pm 2.8$  % error (confidence interval: 68 %).

TABLE III. OPTIMAL HYPER-PARAMETERS FOR THE IMPLEMENTED ENSEMBLE-CLASSIFIERS

	Grasped-event	Adhered-event
Ensemble method	AdaBoost	AdaBoost
Weak lerner	Tree (1 split)	Tree (1 split)
Number of weak lerners	500	10
Learning rate	0.1	1
Error [%]	7.06	7.10

TABLE IV. OPTIMIZED HYPER-PARAMETERS FOR THE IMPLEMENTED SVM

Hyper-parameters	Grasped-event	Adhered-event
Kernel function	Gaussian	Linear
Initial KernelScale	10.0	1.92
Search range for KernelScale	[1, 100]	[1, 10]
Optimal KernerScale	86.0	1.42
Initial BoxConstraint	1.77	0.01
Search range for BoxConstraint	[1, 10]	[0.001, 0.1]
Optimal BoxConstraint	4.77	0.00707
Error with optimal parameters [%]	6.11	18.5

# V. RESULTS AND DISCUSSION

Table V summarizes the results of the evaluation of different classification methods. The most suitable classifiers for graspedand adhered-events are SVM with an error of 6.11 % and tree learner with an error of 6.74 %, respectively. However, the best cumulative error for both events together was obtained for the ensemble learning classifier. Table V also shows that the classification errors of the grasped-event are significantly smaller for all of the learning based methods compared to the purely logic based classification. This is true also for 3 out of 5 ML-based methods in the adhered-event classification. Some classifiers seem not to be suitable for some events, such as SVM for adhered-event. Although the choice of the classifier affects the error, tuning the hyper-parameters has certain significance as well. For example, the error varies vastly when the number of splits in a tree changes or the number of weak learners in an ensemble varies. Fig. 7e-f show the errors of the different ML classifiers with respect to the number of training samples. From the figure, one can see that the error decreases when the number of training data samples increases. The error continues to decrease until the upper bound of the number of data is reached, which indicates that the accuracies can be increased with more data. Naive Bayes classifier is left out from the analysis because it cannot be trained with partial training datasets due to the fact that some features have zero variance in the subsampled datasets.

### VI. SUMMARY

To achieve autonomous micromanipulation, higher level of intelligence is required to identify events in the manipulation scene. In this work, machine learning algorithms are assessed for classification of the state of SU-8 chips while being manipulated. The work consists of feature definition; application and evaluation of machine learning classification algorithms. Definition of features was applied to generate simplified data representation which is efficient for machine learning. As no commonly accepted guidelines exist for the selection of a supervised machine learning classifier, five of the most common supervised machine learning classifiers were assessed. The results of the classifier evaluation suggest that the selection of a

TABLE V. MINIMUM CLASSIFICATION ERRORS [%] OF EACH TUNED CLASSIFIER.

Classifier	Grasped	Adhered	Cumulative Error
Optimized logic-based	14.20	7.84	11.02
naive Bayes	9.44	15.9	12.67
Tree	8.17	6.74	7.45
Ensemble	7.06	7.10	7.08
SVM	6.11	18.50	12.30
Neural Networks	8.80	10.00	9.4



Fig. 7. Classification error rate as a function of different tuning parameters in the studied classifiers or amount of data. (a) Classification errors for tree learning with different maximum number of splits. Classification error rate as a function of number of weak learners for different ensemble configurations for the (b) grasped-event and (c) adhered-event classifiers. (d) Classification errors of the events with ANNs with different topologies. The learning rate of grasped-event (e) and adhered-event (f) with different ML classifiers with respect to the amount of training data.

certain classifier is not enough. Deliberate tuning of its hyperparameters have as much effect on the classification accuracy. Two example events were used in the evaluation task and the best cumulative (mean) ML classifier yielded 92.9 % accuracy. This cumulative accuracy is higher than the one of the logicbased classification (88.9 %). The accuracy of the ML classifiers can be further improved by introducing more data. Therefore, the machine learning based classification approach presented in this work is an adequate method to identify events during micromanipulation.

### REFERENCES

- Q. Zhou and V. Sariola, "Unified View of Robotic Microhandling and Self-Assembly," Robot. Microassembly, pp. 107–143.
- [2] P. Saketi, M. Von Essen, M. Mikczinski, S. Heinemann, S. Fatikow, and P. Kallio, "A flexible microrobotic platform for handling microscale specimens of fibrous materials for microscopic studies," J. Microsc., vol. 248, no. 2, pp. 163–171, 2012.
- [3] V. Liimatainen, J. Venalainen, M. Koverola, Q. Zhou, J. Lyytinen, J. Koskinen, V. Rontu, M. Berdova, and S. Franssila, "Characterization of nano-coated micro- and nanostructures by pushing," 2014 Int. Conf. Manip. Manuf. Meas. Nanoscale, 3M-NANO 2014 Conf. Proc., no. October, pp. 23–27, 2014.
- [4] J. Hirvonen, M. Myllys, and P. Kallio, "Method for 3D fibre reconstruction on a microrobotic platform," J. Microsc., vol. 263, no. 1,

pp. 20–33, 2016.

- [5] B. Tamadazte, E. Marchand, S. Dembele, and N. Le Fort-Piat, "CAD Model-based Tracking and 3D Visual-based Control for MEMS Microassembly," Int. J. Rob. Res., vol. 29, no. 11, pp. 1416–1434, 2010.
- [6] Q. Zhou, P. Korhonen, J. Laitinen, and S. Sjövall, "Automatic dextrous microhandling based on a 6-DOF microgripper," J. Micromechatronics, vol. 3, no. 3, pp. 359–387, 2006.
- [7] M. Ammi and A. Ferreira, "Biological cell injection visual and haptic interface," Adv. Robot., vol. 20, no. January 2015, pp. 283–304, 2006.
- [8] Z. Cenev, J. Venäläinen, V. Sariola, and Q. Zhou, "Object tracking in robotic micromanipulation by supervised ensemble learning classifier," 2016 Int. Conf. Manip. Autom. Robot. Small Scales, MARSS 2016, 2016.
- [9] I. Rish, "An empirical study of the naive Bayes classifier," Proc. IJCAI 2001 Work. Empir. Methods AI, Int. Jt. Conf. Artif. Intell., no. January 2001, pp. 41–42, 2001.
- [10] J. R. Quinlan, "Decision Trees and Decisionmaking," IEEE Trans. Syst. Man Cybern., vol. 20, no. 2, pp. 339–346, 1990.
   [11] T. Evgeniou and M. Pontil, "Support Vector Machines: Theory and
- [11] T. Evgeniou and M. Pontil, "Support Vector Machines: Theory and Applications," Acai ' 99, Lnai 2049, pp. 249–257, 2001.
- [12] S. J. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, vol. 9, no. 2. 1995.
- [13] D. Opitz and R. Maclin, "Popular Ensemble Methods: An Empirical Study," J. Artif. Intell. Res., vol. 11, pp. 169–198, 1999.
- [14] S. Ali and K. A. Smith, "On learning algorithm selection for classification," Appl. Soft Comput., vol. 6, no. 2, pp. 119–138, 2006.
- [15] D. Wolpert, "No free lunch theorems for search," Most, pp. 1–38, 1995.