LEARNING-BASED DATA-DRIVEN AND VISION METHODOLOGY FOR OPTIMIZED PRINTED ELECTRONICS

FAHMIDA PERVIN BRISHTY

A THESIS SUBMITTED TO

THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

GRADUATE PROGRAM IN

ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

YORK UNIVERSITY

TORONTO, ONTARIO

November 2020

Abstract

Inkjet printing is an active domain of additive manufacturing and printed electronics due to its promising features, starting from low-cost, scalability, non-contact printing, and microscale on-demand pattern customization. Up until now, mainstream research has been making headway in the development of ink material and printing process optimization through traditional methods, with almost no work concentrated on machine learning and vision-based drop behavior prediction, pattern generation, and enhancement. In this work, we first carry out a systematic piezoelectric drop on demand inkjet drop generation and characterization study to structure our dataset, which is later used to develop a drop formulation prediction module for diverse materials. Machine learning enables us to predict the drop speed and radius for particular material and printer electrical signal configuration. We verify our prediction results with untested graphene oxide ink.

Thereafter, we study automated pattern generation and evaluation algorithms for inkjet printing via computer vision schema for several shapes, scales and finalize the best sequencing method in terms of comparative pattern quality, along with the underlying causes. In a nutshell, we develop and validate an automated vision methodology to optimize any given two-dimensional patterns. We show that traditional raster printing is inferior to other promising methods such as contour printing, segmented matrix printing, depending on the shape and dimension of the designed pattern. Our proposed vision-based printing algorithm eliminates manual printing configuration workload and is intelligent enough to decide on which segment of the pattern should be printed in which order and sequence. Besides, process defect monitoring and tracking has shown promising results equivalent to manual short circuit, open circuit, and sheet resistance testing for deciding over pattern acceptance or rejection with reduced device testing time. Drop behavior forecast, automatic pattern optimization, and defect quantization compared with the designed image allow dynamic adaptation of any materials properties with regards to any substrate and sophisticated design as established here with varying material properties; complex design features such as corners, edges, and miniature scale can be achieved.

Acknowledgments

I was lucky enough to meet a significant number of notable people during my Master's degree. All of their help, whether explicitly or implicitly, propelled me towards the end of my degree. I would have written here about all the help I received and expressed gratitude to each person individually if space was not limited.

In the first place, I would like to convey my gratitude and wholehearted thanks to my supervisor, Professor Gerd Grau, for his continued support in each aspect of my graduate life. The works done for this thesis is a testament to the kind guidance I received from him. I have experienced the freedom to explore many newer ideas while having an expert opinion when needed. I would convey my gratitude to Dr. Ruth Urner, a great mentor, and an excellent collaborator, the Chair of my thesis supervisory committee.

The support of lab technologists Md Amin Haque Talukder, Giancarlo Ayala-Charca from Electrical Engineering and Computer Science, and Shane Guo in the SPARC cleanroom facility the characterization tools is praiseworthy. Especial thanks to Md Saifur Rahman, Paria Naderi, Mohamad Kannan, Bobak Badrian, Milad Ghalamboran for lab instrument training, and thoughtful discussion regarding practical problems. Their combined support made the lab environment friendly and enjoyable.

Lastly, but most importantly, I want to express gratitude toward my husband Saif for his constant encouragement and family support.

Table of Content

List of Tables

List of Figures

1. Chapter 1: Introduction

1.1. Overview and Research Objectives

In this thesis, we focused on studying machine learning-based inkjet drop formulation prediction, automated pattern sequence generation, and evaluation. We break down the inkjet printing optimization problem into three parts: 1. developing a learning-based pipeline to predict jettable ink drop formation from the machine and material parameters, 2. generating a print pattern sequence with vision-based algorithms and printing the optimized pattern, 3. evaluating the printed pattern and quantizing defects in terms of graph and segmentation based region of interest determination against the ground truth design through vision processing. Once the prediction region is defined for ink and printer signal property set, an automated printing assembly line can be set up for industrial inkjet manufacturing with the vision pattern generation and evaluation pipeline. Defect quantity is an essential property of any inkjet-printed pattern. With the defect threshold knowledge beforehand, the whole printing optimization process can be computerized, repetitive, and becomes less time and material consuming.

Inkjet printing generates droplets of circuit material with a piezo-electrically actuated dispenser controlled through frequency, voltage pulse, and timing parameters. A significant challenge is the rapid optimization of stable jetting conditions while preventing common problems (no ejection, perturbation, satellite drop, multiple drops, drop breaking, nozzle clogging). This work aims to replace the material consuming trial and error experiments with an intelligent ML algorithm to forecast the jetting window based on machine and material properties. There is a small window of the combination of material and signal values where there is stable jetting with optimal drop velocity and volume. Drop quality defines the next step of manufacturing. If the drops are breaking or jetting with satellites, then the lines or patterns formed with it will have irregular shapes. Here, shapes can be the length, width, or thickness. For example, if the materials in a transistor are not jetting well, the gate length, width, and thickness can vary, so the transistor behavior may not be repeatable and sometimes can be damaged. With precise jetting, the formed layers and patterns follow exact design dimensions permitting less defective industrial production. Machine learning can forecast the drop ejection behavior with known material properties. A successful prediction model built on dataset consisting of all essential printing features can reduce the setup time and effort as well as manufacturing defects. For previously untested inks, learning-

based printing can easily identify the active region of the electrical signal or help to select inks' optimum material properties to achieve the desired drop volume and velocity. There is an absence of learning based studies on printing and material attributes during inkjet printing processes at the micron scale. Forecasting the jetting conditions will assist in bulk electronics manufacturing through the decisive material and electrical signal selection, optimizing the drop speed, experimental time, and the process.

The second part of this study demonstrates that sequencing the placement and order of droplets in contrast to the standard raster-scan approach delivers optimum electronics printing outcomes. The generation and evaluation of the printing sequence are turned into a computer-vision problem by taking the desired printed pattern as an input image and converting it into a printing sequence using contour, symmetric, matrix sequencing, and corner compensation with the precise details. After printing, pattern defects are detected by automated image processing to evaluate it against the designed ground truth image and to distinguish the best possible algorithm for the printing sequence generation. On the whole, a vision-based experimental approach is established to provide better solutions for solving the printing and defect optimization problem with regard to precision, recall, and accuracy. The vision-based patterning results are justified with sheet resistance and surface topology results.

1.2. Background and Literature Review

1.2.1. Additive Manufacturing

Additive manufacturing (AM) is a promising technology for digital fabrication of electronic devices. AM offers on demand flexibility, high-throughput, less material consumption and efficiency to large-area, shorter lead times, low-cost electronics manufacturing operations. This technology is a prospective competitor of traditional subtractive semiconductor processing. AM is an active domain of exploration, optimization of complex design geometries with a view to depositing material in precise geometric shapes. Diverse electronics materials from semiconductors to conductors and insulators including polymer, nanoparticles, metal, organic and bio-materials can be rapidly prototyped and high-quality electronic devices and systems are digitally manufactured with unique shapes and multifunctional compositions [1]. Artificial intelligence and machine learning innovations have started to enter the AM field to predict the

print outcome of any material ensuring the best possible quality [2]. AI driven AM production inline can automate the printing production lines without any human supervision. This exploration of the status and prospects for improvement of our AM study is targeted at material jetting technology, one out of seven types defined in the Standard F2792 produced by the American Society for Testing and Materials (ASTM) [3]. Binder jetting is a 3D printing technology that recoats the powder material layer by layer then prints liquid on top that bind the material. Direct energy deposition melts metal powder with a focused energy source and the feeding nozzle continuously adds the material on the building block. However, it doesn't bring a great resolution with its high setup cost. Extrusion is one of the simplest 3D printing process that melts the material filament in the printer head during dispensing. Among the seven different types, fourth process type- jetting is one of the most promising one due its low cost, customization and on demand printing. Drop on Demand (DOD) process optimization is the main focus of our study. In case of powder bed, the thickness of layer is controlled by the amount of powder spreading, material and process condition. Sheet lamination has the least AM resolution, although it is faster and comes with low cost. Printed 3D material also requires quite amount of post processing. Vat photopolymerization is comparatively expensive and only support limited number of photo-resin material. Yet, it provides higher printing accuracy with lower material waste. Table 1-1compares these seven different types of AM.

Table 1-1: Comparison of various additive manufacturing processes.

Process Type	Process Name	Process Description
1	Binder jetting	In this process, liquid bonding agent is selectively deposited to join powder materials [4].
2	Directed energy deposition	This method uses focused thermal energy and fuses materials. Focused laser, electron beam, or plasma arc are used to melt the deposition materials [5].
3	Extrusion	During the extrusion procedure, material is carefully dispensed through a nozzle or orifice[6][7].e.g. FDM(Fused Deposition Modeling)
4	Jetting	In the jetting process, droplets of build material are selectively deposited [8][9].e.g. DOD printing

5	Powder Bed Fusion	Throughout the process, material placed on powder bed is selectively heated. While melted, powder layer is immediately spread over and they are fused together [10].
6	Sheet Lamination	Deposition material sheets are bonded together to form a part [11].
7	Vat Photopolymerization	Liquid photopolymer resin in a vat (tank) is selectively cured by ultraviolet (UV) light and the hardened resin forms the object without any structural support [12].

1.2.2. Comparative Study of Printed Electronics and Applications

Printed electronics is an emerging manufacturing technology complementary to traditional silicon electronics due to its advantageous properties such as low-temperature processing, absence of vacuum processes, compatibility with numerous cheap, flexible substrates, and large-scale roll-to-roll process ability. Multilayered printing provides exciting possibilities for the on-demand fabrication of passive electronics such as inductors, capacitors, and resistors [13][14]. Low-cost radio frequency antennas for consumer package tracking RFID tags are industrially manufactured through printing contacts, interconnects, active, and passive components on flexible substrates with conductive inks[15][16][17]. The LCD industry has started adopting printed active-matrix color filter layers [18]. Hybrid printed structure improves surface irregularity and intrinsic pinhole through an absorbing polymer buffer layer deposition following with inkjet-printing on the top. It leads to cheap, large-area, and multicolor OLED production [19]. There have been several demonstrations of printed transistors avoiding complicated traditional fabrication steps such as sputtering [20][22][22]. Figure 1-1 illustrates some exciting applications developed through printed electronics.

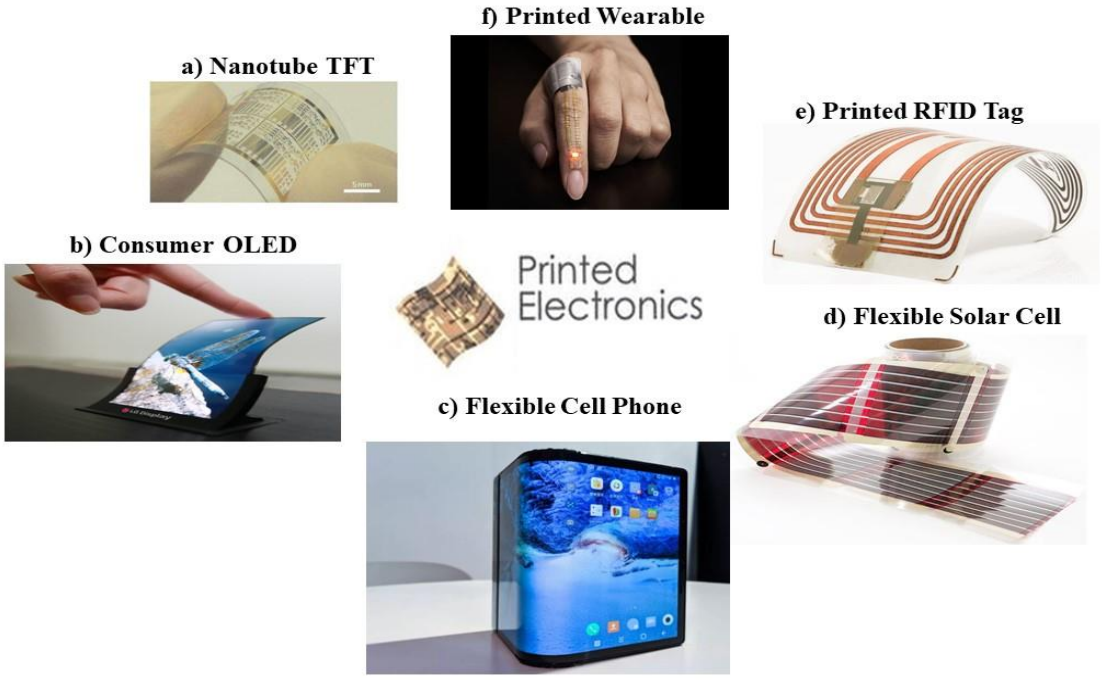


Figure 1-1: Some large-area applications that can benefit from the advantages of printed electronics. (a) Carbon nanotube transistors lead to inexpensive, flexible electronics (www.phys.org/news). (b) 5 inch plastic OLED Flexible display for consumer electronics (www.inhabitat.com). (c) Flexible cell phone (www.theverge.com). (d) Large scale flexible printed organic solar cell (www.heli-on.infinitypv.com). (e) Screen-printed RFID antenna and plastic IC tag (www.rfid-wiot-tomorrow.com). (f) Breathable, wearable electronics on skin for long term health monitoring (www.europeanpharmaceuticalreview.com)

The interest is growing towards digital printing (inkjet printing, extrusion printing, 3D printing), which is easier to configure through digital input files without any predefined mask or structure requirement. Digital printing techniques are more flexible and can be readily adapted to any required customization. Digital printing also enables on-demand production along with online or offline rapid pattern testing [24]. Other than the organic, inorganic ink materials (nanoparticles, nanowires, nanotubes, etc.) with improved electrical properties has made printed electronics more viable in the industry. However, digitally printed pattern quality needs to be measured in terms of morphology (uniformity, bulging, scalloping, beads, and holes), layer-to-layer registration, and so on. Various ink materials, including organic semiconductors, dielectrics, and conductive inks selection for the active or passive device requires extensive printing and material parameter

configuration. These printing configurations, printed design improvement, and defect identification can be resolved to employ machine learning and computer vision. The development of newer materials might augment the electronic fabrication industry in the form of readily printable inks.

Depending on the ink type, the printing conditions need to be changed significantly. The inks have their own electrical and material properties for different applications: high mass loading or concentration, variation in particle size, viscosity, density, surface tension, and acoustic wave speed. All these variant properties affect the jetting and the design patterning. The printer is driven by an electrical waveform, which has timing, voltage, and frequency parameters. Several theoretical and practical studies have been explored before understanding the jetting window [25]-[28]. These windows are often expressed in terms of non-dimensional fluid mechanical numbers such as Z-number, Weber number (We), capillary number (Ca), Ohnsorge number (Oh), or Reynolds number (Re) and are not very accurate defined by equation (1)-(3)(4).

$$\text{Reynolds number, Re}=\frac{\text{Inertial force}}{\text{Viscous force}}=\frac{\text{DensityXDrop VelocityXNozzle Diameter}}{\text{Viscosity}} \quad (1)$$

$$\text{Weber number, We}=\frac{\text{Inertial force}}{\text{surface tension}}=\frac{\text{DensityX DropVelocityX Nozzle Diameter}}{\text{Surface Tension}} \quad (2)$$

$$\text{Capillary number, Ca}=\frac{\text{Viscous force}}{\text{surface tension}}=\frac{\text{ViscosityXDrop Velocity}}{\text{Surface Tension}} \quad (3)$$

$$Z=\sqrt{\frac{\text{Surface Tension X Density X Nozzle Diameter}}{\text{Viscosity}}} \quad (4)$$

1.2.3. Inkjet Deposition of Functional Materials

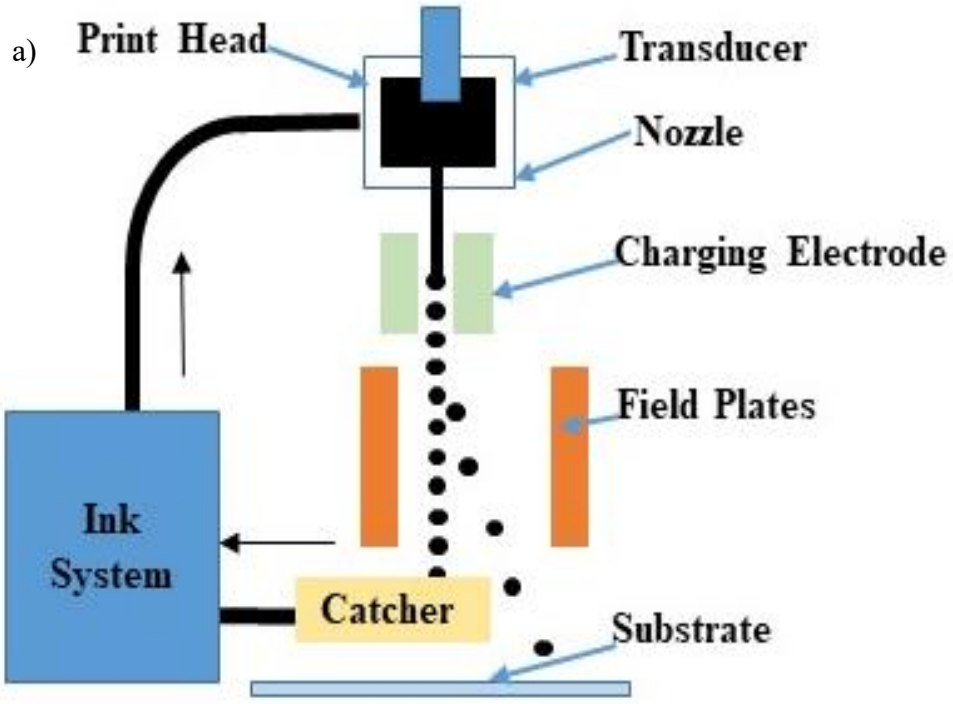
Inkjet patterning is a state of art technology for depositing functional materials on desired substrates. With its easy to load and print micro pattern from the digital configuration file, low temperature, low material waste and high-speed characteristics; it is a promising alternative for complex microlithography processes. This contactless drop by drop digital fabrication is challenging due to its large parameter configuration range and difficulty locating the optimum operation region. A number of research works have reported theoretical and simulation results on how the intrinsic characteristics of functional materials led to significant changes in the print configuration and printed pattern morphology. Consequently, taking it as our motivation, we

explored material and machine parameter effects on the printing process as well as how these finally manipulate electronics design and fabrication. We first explore inkjet printing technologies and their current applications before examining the data-driven context driving the forecasting of the inkjet printing process.

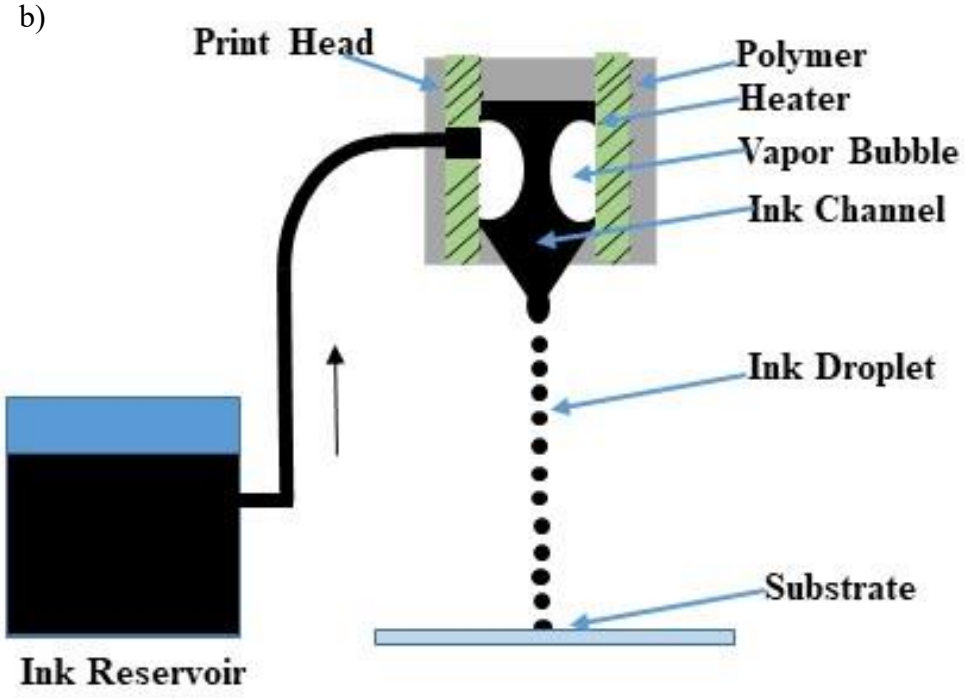
Inkjet can be classified by two major methods of drop formation and placement: continuous inkjet printing (CIJ) and drop on demand DOD (thermal inkjet printing (TIJ), piezoelectric printing). These processes surpass other notable drop generation technologies: electrospray printing, valve-jetting [29] in terms of cost, process control, complexity and timing. In CIJ, a jet emerges from the nozzle, which breaks into a stream of droplets, as shown in Figure 1-2(a). CIJ jets a continuous stream of electrostatically charged ink droplets into a gutter. The gutter, in other words, the catcher, returns the extra ink drops into the ink supply [30][33]. There are two field plates that deflect the stream momentarily and direct the drop sequence onto the substrate. This printing method utilizes a small proportion of the generated droplets, and the rest is recycled through the ink system. For preserving the correct viscosity throughout the cycle, the ink is suspended in a volatile organic solvent-based thinner. CIJ arrangements are relatively complex and composed of ink pumps, pressure regulators, filters, and sensitive print heads. It requires high maintenance, dedicated factory operators. There is no downtime in continuous printing.

CIJ's main competitors are DOD Thermal inkjet (TIJ) and DOD piezoelectric inkjet (PIJ) systems. They are displayed in Figure 1-2(b-c). The extent of ink loss throughout the CIJ printing process is a significant disadvantage that concerns most users. Adaptation of the DOD inkjet process sorts out this issue. In a DOD inkjet scheme, the printer head, termed as the nozzle, ejects material droplets under computer control [30][33]. The ejected drop falls under gravity until it hits the printing substrate. Then, surface tension drives the droplet spreading and solvent within drop vaporizes. Finally, pattern is developed through the overlying droplets.

a)


Print Head
Transducer
Nozzle
Charging Electrode
Field Plates
Ink System
Catcher
Substrate

b)


Print Head
Polymer
Heater
Vapor Bubble
Ink Channel
Ink Droplet
Substrate
Ink Reservoir

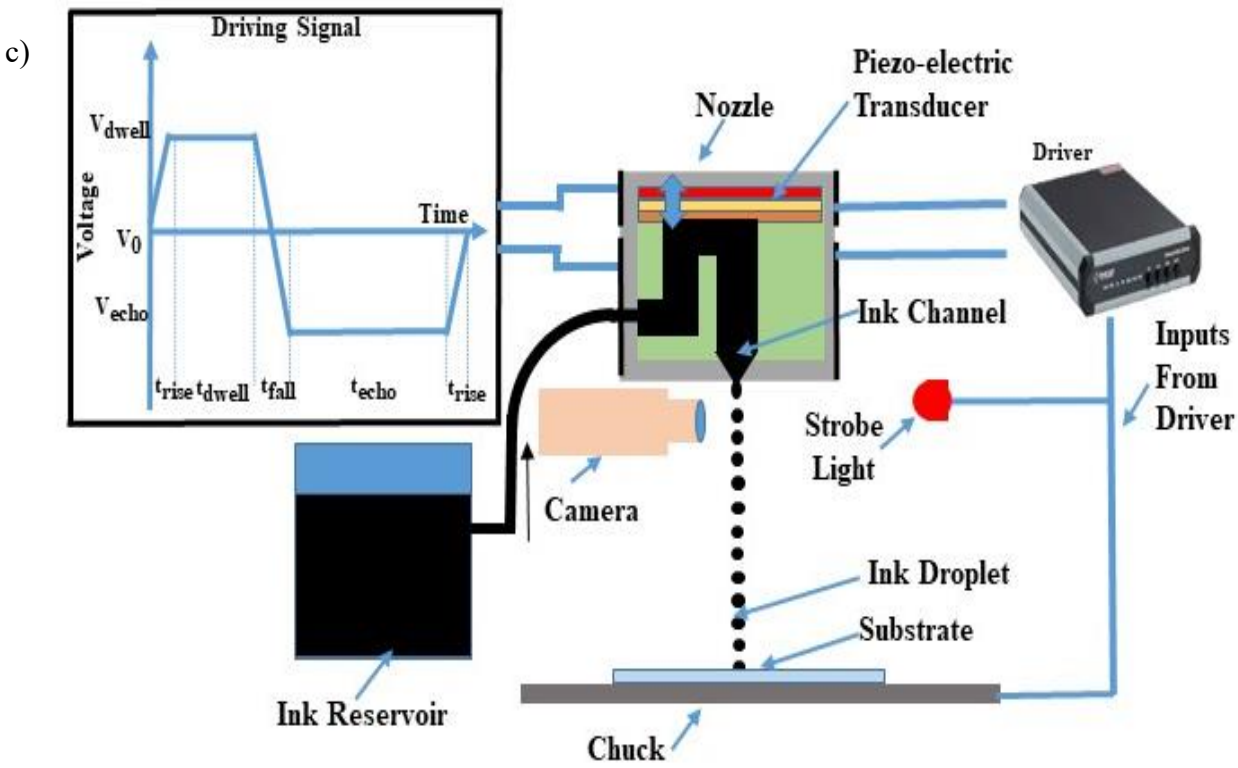


Figure 1-2: Schematic of Conventional inkjet printer setup. (a) Continuous inkjet printer (CIJ), (b) Thermal inkjet printer (TIJ), (c) Piezoelectric inkjet printer (PIJ).

DOD is cost-effective for small to medium production setup and can be configured for intermittent production runs. Different technologies generate droplets; the two main methods are thermal bubble jet and piezoelectric jet [30][32]. In TIJ or bubble jet, displayed in Figure 1-2(b), the current drive heats up a thin, polysilicon resistive heater surrounding the ink material. TIJ nozzle is disposable, cheaper than piezoelectric crystal nozzles. TIJ is integrated with a controller integrated that injects current to heat and evaporate ink material. Generated vapor stresses ink droplet out of the nozzle within a short time. The heater is then turned off momentarily to let the void space get filled with ink from the reservoir. Although TIJ can be operated with two-thirds of a CIJ's cost per 1000 prints [35], the ink delivery system needs to be changed with ink material. Again, the nozzle needs to be frequently changed whenever leftover ink material piles up on the heater surface. Besides, the delicate ink materials susceptible to temperature change (e.g., nanoparticle-based) can't be printed with TIJ.

PIJ, as shown in Figure 1-2 (c) [30][33][36], is the most popular inkjet printing system. Typical PIJ dispensing includes a DOD operation with a piezo-electrically actuated dispenser micro-dispenser, LED strobe, ink reservoir, drive electronics components. The dispenser is controlled through voltage pulse, frequency, and timing parameters to convey the necessary droplet generation energy. These parameters have noticeable consequences on printing consistency. A minimum reservoir pressure for purging solution through the nozzle is also imperative. The other parts, like the observation camera, video monitor, mechanical mounting, do not impact stable droplet generation. PIJ doesn't apply a thermal load to organic inks as the thermal bubble inkjet technique. Piezoelectric plate triggers in the microsecond range, and this feature makes PIJ faster than the thermal bubble type. PIJ, TIJ DOD printing resolution is limited by the nozzle make diameter. Usually, make size is between 20-120 μm. Higher feature resolutions come with a smaller nozzle diameter and face the challenge of frequent nozzle clogging. Moreover, as the diameter gets smaller, drop jetting, sequencing, printing gets unpredictable. The expedition for reaching higher resolutions, printing speeds, and greater accuracy has led to learning-driven automated inkjet printing methods.

Various inks are used for printing conductor, insulator, or semiconductor layers. The inks have different electrical and material properties for other applications: high mass loading or concentration, variation in particle size, viscosity, density, surface tension, acoustic wave speed. All of these ink properties affect the jetting and the pattern formation on the substrate. There exist a small window of material and signal (e.g., jetting voltage, frequency, or timing) parameter combinations where there is stable jetting with optimal drop velocity and volume. Outside of this window, either there is no ejection from the nozzle or the jetting is unstable, and the drop breaks up into multiple droplets or satellites. For multiple droplet cases, the biggest one is referred to as the primary drop, and the other one is termed as the secondary drop. Drop quality is critical for successful manufacturing using inkjet printing. If the drops break or jet with satellites, the printed lines or circuit patterns will have irregular shapes and defects. This can adversely affect the performance, yield, and variability of printed devices such as transistors. Therefore, methods need to be developed to achieve precise jetting. In this case, the print head uses the deformation of a piezoelectric ceramic element. Piezoelectric print heads can handle a broader range of liquids than thermal print heads with relatively smaller drop size. It also doesn’t require ink recycling system. Various drop diameters are achievable in DOD printing, starting from tens of μm to hundreds of

μm. DOD inkjet printing is illustrated in Figure 1-2 (c). It allows patterns to be customized on-the-fly in the form of droplets detached from the nozzle under surface tension. If the ejection conditions are not optimized, in most of the cases, the jet is unstable. Hence, the drop breaks up or gets accompanied by one or smaller satellite drops. We have discussed different jetting phenomena and how to realize a stable jetting scenario in chapter 2 with a machine learning approach.

Quite a few theoretical and experimental studies have explored the underlying physics and experimental conditions for droplet generation and jetting feature characterization [36][46]. Hoath et al. estimated drop speeds from a range of industrial drop-on-demand (DOD) inkjet print heads, namely Xaar, Spectra Dimatix, and MicroFab, through simple theoretical models with numerical simulations [38]. They concluded that drop speed depends on fluid properties, nozzle exit diameter, and printer voltage magnitude. All fluid types, either Newtonian, weakly elastic, or highly shear-thinning, demonstrate linear speed rise with voltage above a threshold. By dint of modeling and numerical simulations of fluids with varying fluid properties, surface tension, and viscosity, they reported that the final drop speed is a function of voltage, the threshold voltage (a function of viscosity) nozzle tip area. They maintained constant frequency, rise time, fall time, dwell time throughout the investigation. Nallan et al. explored fluid rheological parameters [39]. They found a jetting region with Capillary, Weber number derived from their optimal jetting conditions experiments for nanoparticle loaded ink design. However, this study didn't include effects from printer electrical parameters on the window. Liu et al. found that droplet formation is impacted by fluid properties such as viscosity and the driving waveform parameters [40]. The group moved further with an experimental inkjet behavior study highlighting a parameter space confined by Z, Weber number, and extracted thresholds from that space to specify the lower onset of drop ejection, besides the upper bound signifying satellite drop formation [41]. The Lai team conducted another research carried out computational fluid dynamics to inspect droplet ejection physical phenomena and compared experimental results with numerical simulations. It determined how nozzle channel curvature affects drop velocity, volume, and the number of satellite drops. Besides, they demonstrated that nozzle diameter, voltage amplitude, or frequency increment lead to volume rise accompanied by a drop in velocity [42]. Reis et al. experimented with a piezoelectric inkjet printer to explore the drop volume and velocity as a function of voltage, frequency, and summarized drop volume as a function of the Ohnesorge number of the orifice [43]. Yang eliminated satellite droplets via designing nozzles with super-ink-phobicity, ultralow adhesion to

enhance Rayleigh filament instability along with speeding up drop pinch-off from the nozzle [44]. Another remarkable work from the Wu group inspected the droplet formulation in terms of volume and velocity resulting from monopolar drive voltage, frequency, timing, and compared the results with the acoustic wave propagation theory. They mentioned that drop velocity and volume accelerate with the rise in voltage [45]. Alternatively, He et al. defined a binary fluid model for inkjet printing with a time-dependent actuation and reproduced single droplet ejection to investigate droplet formation in piezoelectric inkjet printing [46]. They determined that high nozzle wettability or low contact angle, low surface tension augment drop quality, and lessen drop speed. The stable jetting window is typically bounded by straight lines defined in non-dimensional quantities Ca, We, Z, Oh. There are several challenges with this approach. These window boundaries are not very accurate, and different studies in the literature report are jetting windows with somewhat different numerical values. This might be because different reports study different printable materials that may behave differently. Another difficulty is that most of these non-dimensional numbers, except Oh and Z, contain a velocity term. This is no problem when results are analyzed after performing printing experiments. However, for new ink, the drop velocity is generally not known a priori and needs to be determined experimentally. Hence, the demand for a technique for predicting jet ability, drop velocity, drop volume, and optimal printer parameters for new ink.

1.3. Machine Learning in the Field of Printed Electronics

Machine learning (ML) programs computers to optimize performance based on available data or past experience for predictions or to gain knowledge [47]. In the learning process, the parameters, cost function of the defined model are optimized using training data and then tested on unseen data. The data-driven ML combines fundamental concepts of computer science with statistics, probability, and optimization. Some examples of interpretable machine learning applications are classification, regression, ranking, and dimensionality reduction or manifold learning [48]. This new era of learning-driven problem solving can assist electronics engineers in design automation, evaluation and synchronization with the industrial production environment for the best possible circuit integration and customization. ML and computer vision approach can simplify and industrialize layer by layer, on-demand, complex, electronics printing system starting from devices to sensors. ML will help designers, layout engineers in device customization, and material selection will be readily aligned with the design requirement without retooling. Provided

that electronic materials, insulating dielectric, semiconductor, and conductive, are predictable concerning inkjet printing, forecasting will enable alternate use of new cheap novel materials with reduced design, lead-time, and cost. Furthermore, it will be possible to find and substitute materials with similar properties with the insights from learned models and operate printing machinery within its working zone without going for manual tuning. This paradigm of ML will bring out new expansion in the electronics manufacturing industry. Besides simplifying the manufacturing, this will drive exceptional feedback between printing equipment, and generated machine data and captured images can be used to improve the built model performance and give meaningful insights regarding the input features, predicted output. Eventually, it will get to the position of electrical engineers' most useful tool in determining and maximizing printing productivity, defects prevention, and anticipating material and machine operating region.

1.3.1. Machine Learning Definitions

Some standard definitions of machine learning used throughout our work have been described here. After data preprocessing, train test data has been selected through random sampling. Then ML data models are trained with optimized parameters selected through the grid search algorithm. Based on target data, the ML models' train and test performance are evaluated through the root mean square error (RMSE), accuracy, precision, recall, F1-score, and confusion matrix.

1.3.1.1. Train and Test Data-Set

Before applying the data model, the whole dataset needs to be segregated into training and testing data. For training purposes, most of the data is used, leaving a smaller proportion for testing. 80:20 ratio of train and test data has been used throughout the project. Through training data, ML data models are fitted to their optimized parameters. ML algorithms map the function given by equation (5) and (6).

$$Y = f(X) + \text{Error(Model)} \tag{5}$$

$$\text{Error(Model)} = \text{Variance(Model)} + \text{Bias(Model)} + \text{Variance(Irreducible Error)} \tag{6}$$

Here, X is observation feature inputs (usually column with headers), and Y is the target output vector (column going to be predicted). And the error of the model is the summation of bias, variance, and irreducible error dependent on variance. In this study, several output vectors (the drop velocity, drop radius, and jetting category) are used for different models, and f (X) is the

decision tree, random forest, boosting, neural network, and weighted averaging models. Testing data is separated from the training dataset, and this data does not alter the model training. Instead, it is used to evaluate the performance of the trained model. While trying to predict any kind of output using a model, there is a potential bias-variance trade-off issue while ensuring how much the model learns from training data. If it learns too much, the model is over fitted, and the training accuracy is way higher than the testing accuracy. This situation has a very high variance, as described in Figure 1-3(a). On the other hand, if the model learns too little, it would have a high bias, and the training data would be under fitted, as shown Figure 1-3(b).

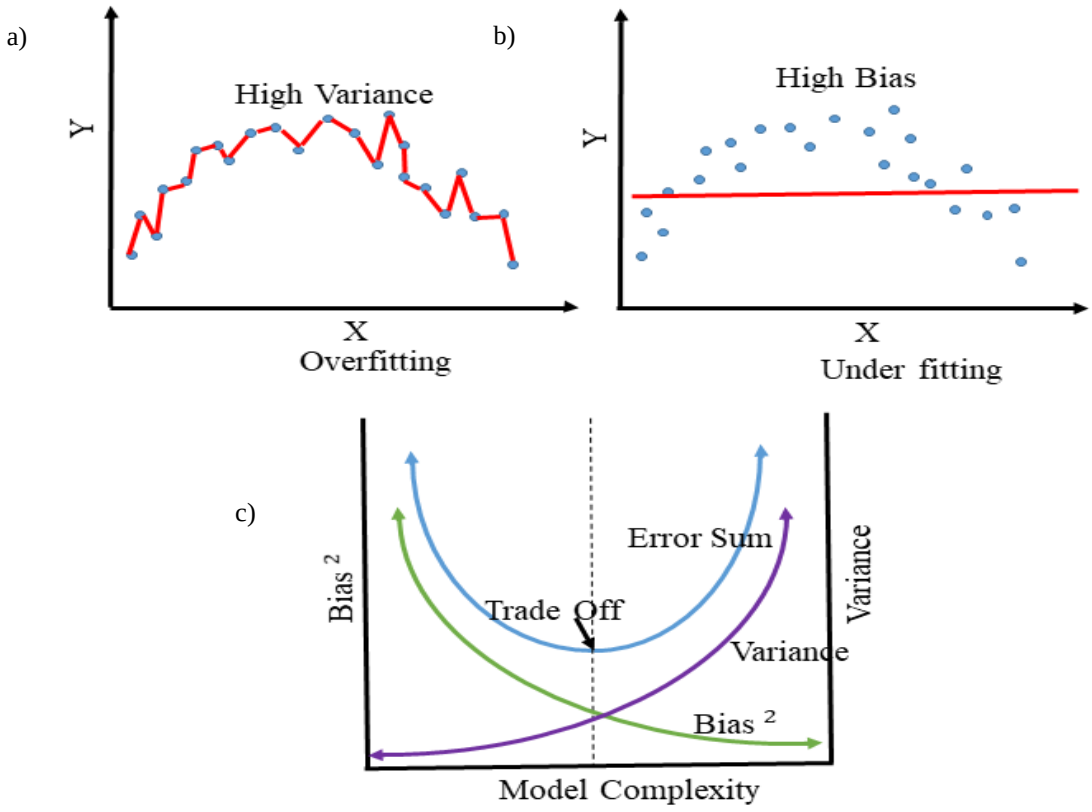


Figure 1-3: (a) Data model fits training data too well, resulting in high variance. (b) The data model can't fit training data adequately and displays high bias. (c) Bias, variance trade-off. Adapted from, "Understanding The Bias-Variance Tradeoff." [online] Medium. Available at: https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229 [Accessed 15 August 2020].

As an effective way to control how much our model learns from the data, data was split into training and testing sets to evaluate and ensure that it performs well on different elements. As another way of bias-variance trade-off Figure 1-3(c), we carried out cross-validation during training and evaluated model performance. We also performed random data shuffling before the train test split to avoid high bias and variance.

1.3.1.2. K-Fold Cross-Validation

K-Fold cross-validation helps to avoid overfitting and underfitting and yields a better model. The total sample space is divided into k approximately equal sets. Keeping the first fold as a validation set, the remaining (k-1) folds train the data model. The training and testing procedure is repeated k times. Instead of using the 1st fold every time, a different observation set is treated as a validation set, and accuracy is calculated. As a result, we get k test performance evaluation metrics, and final validation accuracy is computed by averaging the k performances. 10-fold cross-validation has been implemented for each of our ML models. Model parameters and evaluation metrics are calculated on the hold-out validation dataset to ensure the model performance [49]. The k-Fold cross-validation algorithm has been shown in.

Table 1-2: Algorithm for k-Fold Cross-Validation with grid search

For each fold of the k cross-validation, repeat steps one to three:
Step One: Perform random shuffling of the whole dataset.
Step Two: Split the dataset into k almost equal groups.
Step Three: For each of the k sets, perform the steps below: a)Hold out one group as validation data. b)Treat the remaining (k-1) sets as training data c)Fit the ML model on the training and evaluate it on the test data. d)Save the performance score.
Step Four: Summarize the k models' average performance.
Step Five: For grid searching of each of the hyperparameter range, perform steps one to four. Select the best hyperparameter from the range that links to the best cross-validation score. Use the hyperparameter for retraining and testing model with 80:20 ratio and concluding final score.

Our drop modulation forecasting data has been gone through grid search CV both for classification and regression best model parameter selection. We trained tree-based regressors and classifiers, mainly Decision Tree (DT), Random Forest (RF), Gradient Boosting (GB), to predict the inkjet drop modulation phenomenon. These ML models have several parameters to adjust, and there is no easy way to know which parameters work best other than trying out many different combinations. Scikit-learn provides GridSearchCV, a grid search cross-validation algorithm that explores many parameter settings automatically[51]. GridSearchCV is illustrated in Figure 1-4. Each parameter set produces one model, and the best-performing model is selected.

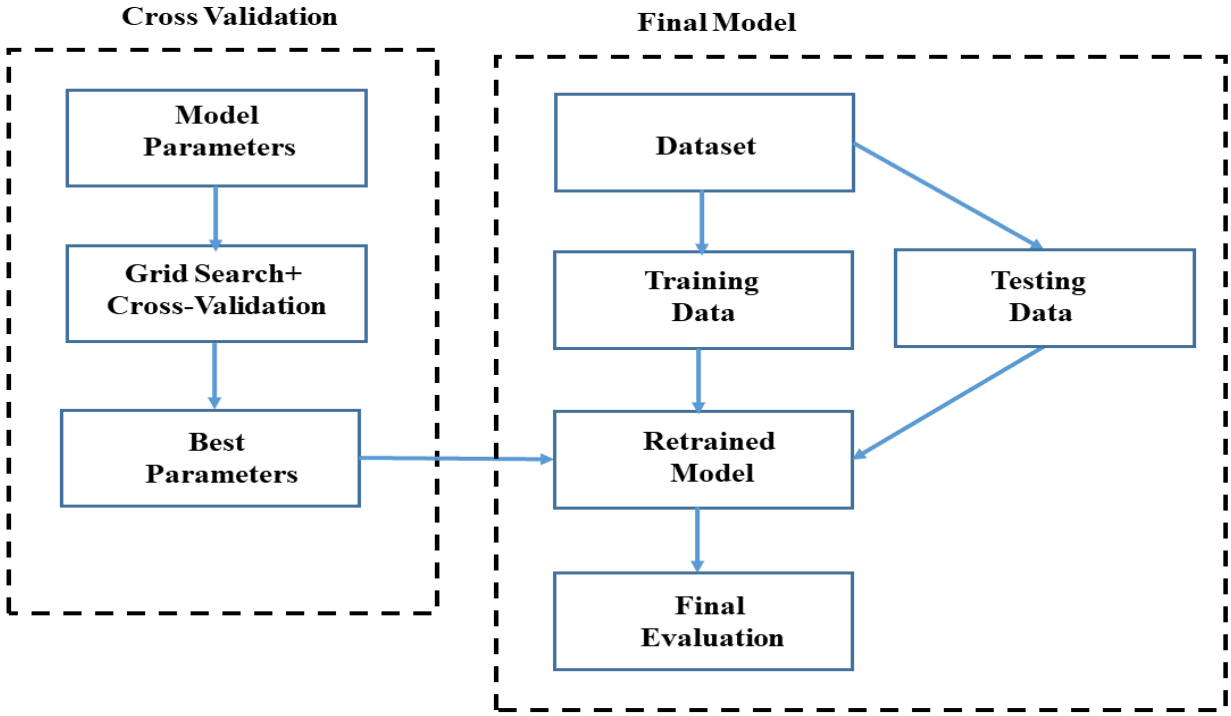


Figure 1-4: GridSearch cross-validation schema. Adapted with permission from (Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay; 12(85):2825−2830, 2011.). Copyright © JMLR 2011, Journal of Machine Learning Research.

1.3.1.3. Model Evaluation Error Metrics

Choosing the right evaluation metrics is crucial to assess the performance of any ML data model. For classification models, performance is best measured through test data prediction

accuracy. Nevertheless, precision, recall, F1-score, Kappa should also be studied for meaningful performance interpretation [52]. Table 1-3 describes confusion matrix terms for assessing classifier model performance.

Table 1-3: Confusion matrix

	Predicted: 0	Predicted: 1
Actual: 0	True Negative(TN)	False Positive(FP)
Actual: 1	False Negative(FN)	True Positive(TP)

As shown in Table 1-3, True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are the most basic terms. TP is the number of positive cases that are correctly predicted. TN is defined as the number of negative instances that are correctly predicted as negative. In the case of FP, the prediction label is positive, but the actual label negative. FP is known as a Type I error. In other words, FN, Type II error, is the count of cases where the prediction is false, but the actual tag is true.

In the case of continuous output regression problems, performance is instead evaluated through mean square error (MSE), root means square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE) [53]. Based on test data, MSE assesses the quality of any regression algorithm performance with the average squared distance between the test data and the predicted data. If N_p number of test prediction vector is sampled from N data points, y_i is the test data vector, y_{ip} is output prediction vector, then within-sample the MSE of the predictor is computed with the equation (7)

$$\text{MSE}=\frac{1}{N_p}\sum_{i=0}^{N_p}(y_i - y_{ip})^2 \tag{7}$$

MSE helps to distinguish between the larger and smaller errors with the drawback of squaring up data units. Consequently, the different unit makes it difficult to justify the estimation. RMSE is the square root of MSE shown by equation (8). This metric solves the problem of squared entities.

$$\text{RMSE}=\sqrt{\text{MSE}} \tag{8}$$

MAE is the most straightforward regression error metric to understand. It measures the average of the absolute model prediction residual given by equation (9). Absolute residual value ensures that negative and positive values do not cancel out. The average describes the average magnitude of the residuals. However, it fails to punish large errors in prediction, as each residual

contributes proportionally to the total amount of error, whether it is larger or smaller. A small MAE suggests that the model is great at prediction, while a large MAE suggests that model may have trouble predicting. An MAE of 0 means that the model is a perfect predictor of the outputs.

$$\text{MAE}= \frac{1}{N_p} \sum_{i=0}^{N_p} | y_i - y_{ip}| \tag{9}$$

Though MAE is easily interpretable, using the residual's absolute value is often not desirable while treating outliers or extreme data points. In such cases, squared metrics such as RMSE can focus more on the outliers or restrain them. MAPE estimates the percentage equivalent of MAE. Equation (10) looks just like equation (9) but with adjustments to convert everything into percentages.

$$\text{MAPE}= \frac{100\%}{N_p} \sum_{i=0}^{N_p} | \frac{y_i - y_{ip}}{y_i} | \tag{10}$$

However, MAE should be used more than MAPE for all of its advantages due to its inclination to predict outputs lower than the real one. MAPE fails to deal with the cases where the real data is 0. Also, the MAPE value rises surprisingly if the actual data is minimal. If we are sure that the dataset doesn't contain any zero values, MAPE can be used.

1.3.2. Data Processing

Drawing insights from data through ML has always been critical. It is ubiquitous not to reach the required research consequences after undergoing a lot of data training and applying the algorithm. Keeping in mind improper data processing and weak model selection as the major points of failure, we exerted most importance on data collection, preprocessing, and then vigorous ML model exploration. Data is the foundation of any ML algorithm and must be supplied in the form that the algorithm understands. The main function of ML algorithms is to unlock the concealed information/knowledge available in the data. The algorithms will provide incorrect, bogus insights if the data is available in a form not comprehended by the algorithm. Preprocessing the data includes both data engineering and feature engineering. Data engineering is the process of transforming raw data into an ML model appropriate form. Feature engineering then tunes the prepared data to create the features expected by the ML model. Feature processing is difficult, time-consuming, requires expert knowledge. We explored data along with features before going out for ML model deployment. Literature raw data was in the form of a table, graph, and image and converted into a common CSV source form for adopting ML. According to the 'Pyramid of

Needs,' we considered ML situated at the uppermost point throughout implementation [54]. The former necessary steps are collecting, cleaning, exploring, and transforming, as shown in Figure 1-5.

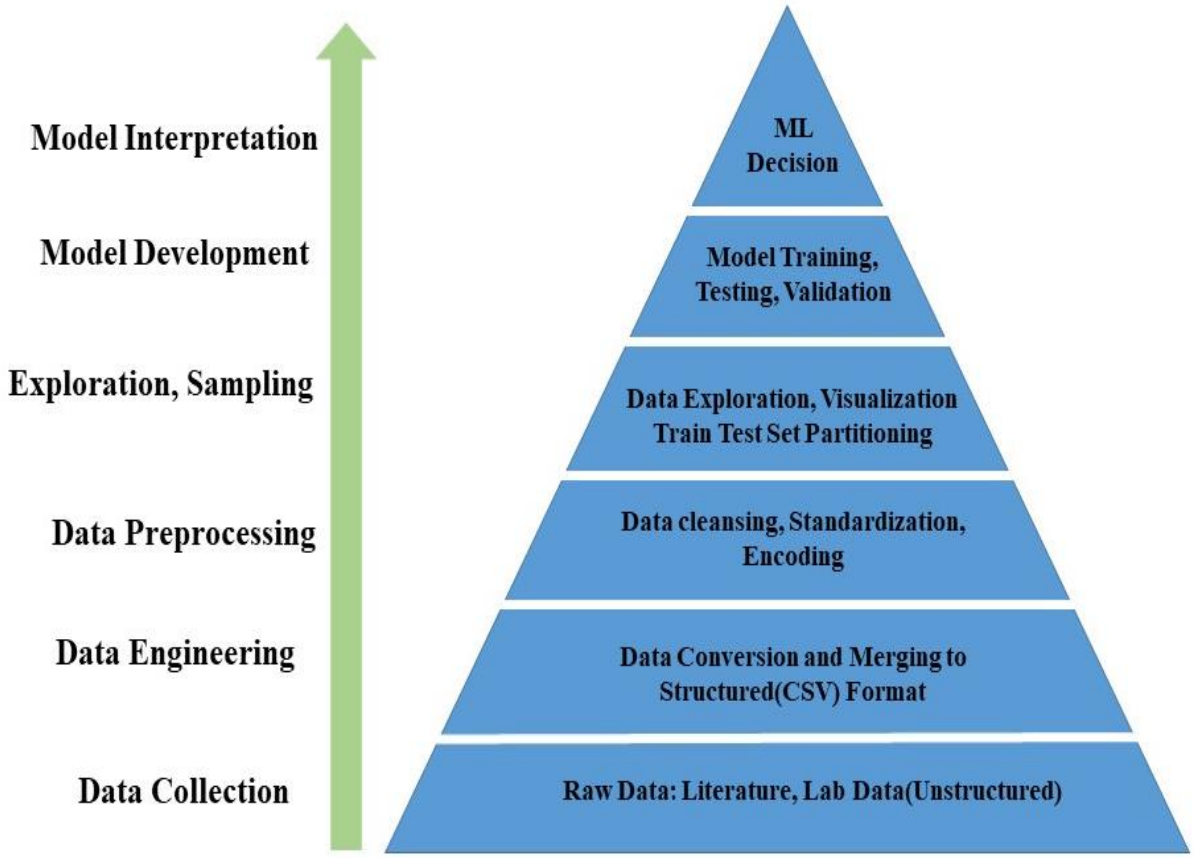


Figure 1-5: The flow of raw data to prepared data for deploying feature engineering to machine learning. Adapted from, "The AI Hierarchy of Needs | Hacker Noon", Hackernoon.com, 2020. [Online]. Available: https://hackernoon.com/the-ai-hierarchy-of-needs-18f111fcc007. [Accessed: 24- Aug- 2020].

After the dataset is transformed and processed, it is ready to be used format for ML task. Once all the data sources are parsed, joined, and put into a tabular form, they are aggregated with it and summarized uniquely. Each row in the dataset represents a unique sample case, and each column represents a distinct feature for the case. Irrelevant feature columns should be dropped, and invalid records (null values) must be filtered out. Preprocessing operations need to be carried on the merged dataset to tune the features expected by the model. Once data preprocessing

operations are performed, the dataset is clean with unique records. Each input numerical feature column's quality can be improved through data standardizing (scaling and normalizing), clipping outliers, and categorical target output transformation to numeric representation through one-hot encoding. Then the training and test evaluation dataset is selected through random sampling from input data points.

1.3.3. Machine Learning Data Models

All the ML algorithms can be categorized mainly using two grouping ways, as described in Figure 1-6. One way consists of combining algorithms with their learning style. Another is to group by the algorithm's functional similarity. An algorithm can have one of the three learning styles. With supervised learning, the ML data model is trained and tested on tagged data to predict. Supervised training continues until it reaches a desired evaluation metric on the training data, as defined in 1.3.1.3 section. Supervised problems are further divided into classification and regression [55][56]. Classification models predict a discrete set of categorical values. Binary classification forecasts one out of two classes, and multi-class classification predicts one out of multiple possible outcomes. In unsupervised learning, input data is not labeled and does not have a target either. A model is prepared by deducing structures present in the input data. This may be to extract general rules. It may be through a mathematical process to systematically reduce redundancy, or it may be to organize data by similarities, such as clustering, dimensionality reduction, and association rule learning. Semi-supervised learning learns through a mixture of labeled and unlabeled data. Rather than learning from labeled or unlabeled data, a semi-supervised learning agent learns about a dynamic environment by taking actions and receiving a positive or negative reward. The actions that lead to outcomes are the ones it tries to replicate. Examples are generative models, graph-based algorithms, multi-view algorithms, self-training [55].

Another useful but not perfect form of grouping is by means of functional similarity such as statistical learning, tree, and perceptron-based methods [55][57][58]. Regression such as Ordinary Least Squares (OLSR), Linear, Logistic, and Stepwise Regression are popular statistical methods for modeling the input variable relationship using an evaluation metric. There is a sophisticated version of regression referred to as regularization that penalizes the ML model based on the complexity. Such examples are Ridge Regression, Least Absolute Shrinkage, Selection Operator (LASSO), and Elastic Net. The instance-based learning model, for instance, k-Nearest Neighbor (kNN), Support Vector Machines (SVM) solve statistical decision problems by building

up a set of training data and compare test data using a similarity measure to find the best match and make a prediction [56][58]. Decision tree methods, namely Classification and Regression Tree (CART), ID3, C4.5, and C5.0, construct decision trees based on training data features until specific evaluation criteria are met. Cluster methods such as- k-Means, Hierarchical Clustering groups a set of training data so that data points in the same group are analogous and constitute a cluster.

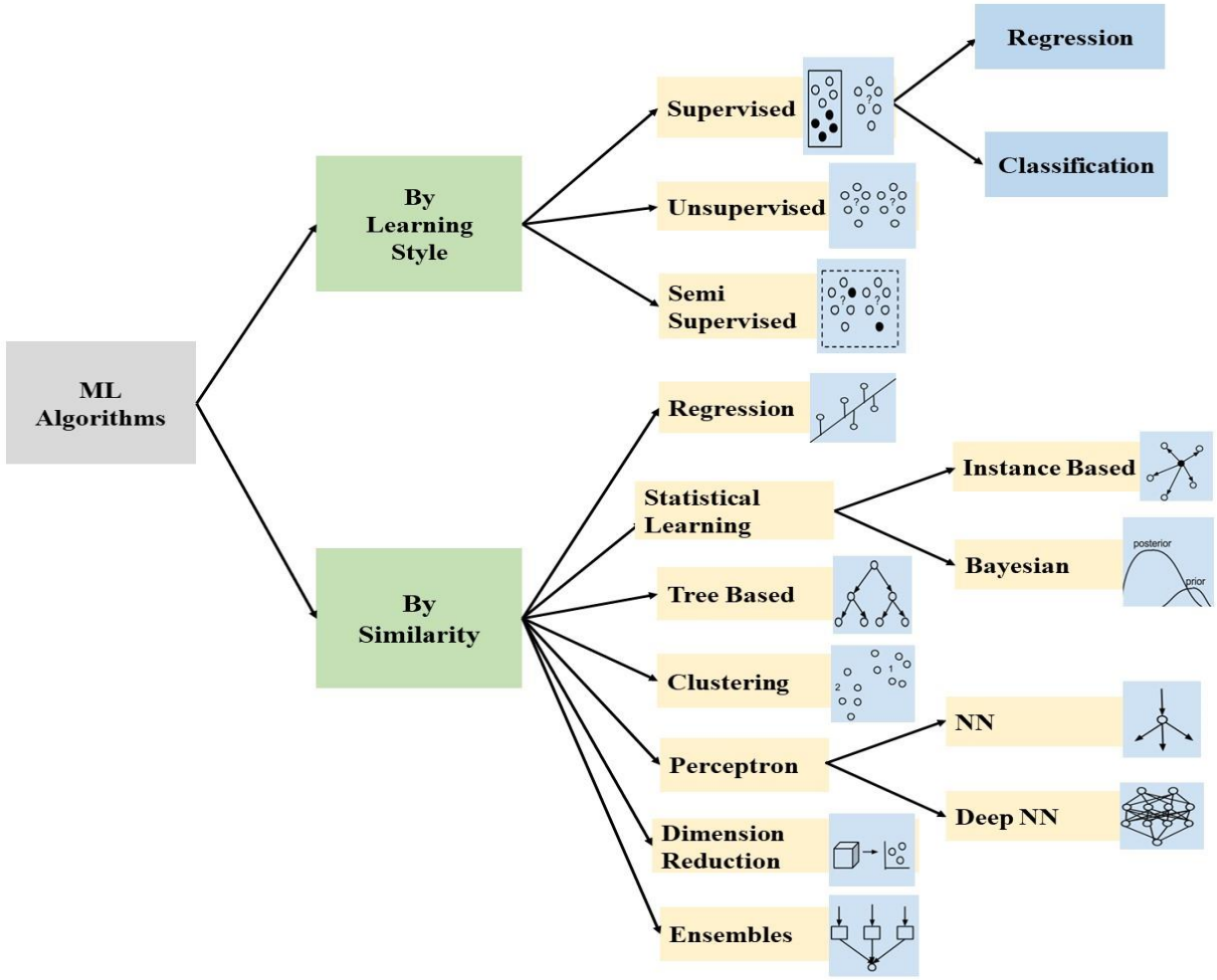


Figure 1-6: Grouping of machine learning algorithms.

Similar to clustering, dimensionality reduction algorithms viz. Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Multi-linear Discriminant Analysis (MDA) mostly summarize data efficiently with fewer features. The perceptron-based techniques can be subdivided into a neural network (NN) and deep neural network (DNN) architectures based on model scale, complexity, and hidden layers. They can be adopted to solve any problem, including regression, classification, unsupervised or semi-supervised incorporating a large amount of

structured, semi-structured (text, JSON file, log file), unstructured (image, audio, video) dataset. Such popular algorithms include Multilayer Perceptions (MLP), Back-Propagation, Stochastic Gradient Descent, Convolutional Neural Network (CNN), Recurrent Neural Networks (RNNs), Long Short-Term Memory Networks (LSTMs), Auto-Encoders. The ensemble is an effective way of combining multiple independently trained ML prediction output and making an overall better prediction.

1.3.3.1. Decision Tree

The Decision Tree (DT) algorithm belongs to the family of supervised learning algorithms [51][55]. It has been used in Chapter 2 as a regression model for drop velocity, radius prediction, and later for classification of drop category. Starting from the root, any nodes except the leaves represents an attribute, and each leaf node denotes a class label or regression output. The constructed tree has been displayed graphically, and the feature relationship with the expected output is interpreted in Chapter 2. DT pseudocode is represented in Table 1-4.

Table 1-4: Algorithm for decision tree.

1.	Use the best attribute of the dataset at the root of the tree using the Gini index.
2.	Use recursive binary splitting to divide the training set into subsets.
3.	Redo step 1 and step 2 on each subset until the number of samples in the leaf nodes in all the tree branches meets the stopping threshold.
4.	Use K-fold cross-validation to choose the maximum tree depth. That is, divide the training observations into ten folds. For each K = 1, . . .,10: (a) Redo steps 1 to 3 on all except one of the K-fold training dataset. (b) Evaluate the error on the left-out one fold data out of the K-folds, as a function of depth.
5.	Average the K-fold results for each value of depth, and pick the depth that gives the best score.
6.	Return the tree from step 5 that corresponds to the depth's chosen value with the best model score.

DT can easily over-fit on training data and result in a too-complicated model with many sub-branches. However, using ensemble of trees (random forests, boosting); prediction results can be improved in terms of higher accuracy and lower RMSE.

1.3.3.2. Random Forest

Random forest is a modified bootstrap aggregating (bagging) ensemble learning technique that builds N number of base learners (trees, linear models) by bootstrapping train data into different subsets [59]. During each sampling, r (=√t generally, but may vary) arbitrary features are chosen out of all t features to trade-off the sampling bias, variance and reduce the learner error [60]. Afterward, the N base learner model fitted using bootstrap sampling, and outputs are aggregated through majority voting (for classification) or averaging (for regression). The random forest pseudocode is represented in Table 1-5.

Table 1-5: Algorithm for Random Forest

For each of the N learners follow step 1 to 5.
1. Take out m number of bootstrap sample subsets (training, target) from the original set of samples n (m<n).
2. During recursive training, select r arbitrary features from all the features for each of the N estimators.
3. Among r features, place the best feature at the root of the tree using the Gini index.
4. Use recursive binary splitting to divide the root data into two node subsets.
5. Redo step 1 to step 4 on each subset until the number of samples in the leaf nodes in all tree branches meets the stopping threshold.
6. Build RF by repeating step 1 to step 5, N times to create N number of tree estimators. For regression, average predictions from N estimators to make the final prediction. In the case of classification, take majority voting of N estimators' predictions to make the final prediction.
7. Use K-fold cross-validation to choose the number of trees N, the maximum number of features r, and the minimum number of samples in the leaf to set the stopping rule. That is, divide the training observations into ten folds. For each K= 1, . . .,10:

(a) Redo steps 1 to 6 on all except one of the K-fold training dataset.

(b) Evaluate the error on the left-out one fold data out of the K-folds as a function of the number of trees, the maximum number of features, and the minimum number of samples. Average the results for each hyper parameter and pick the value that gives the best score.

8. Return the RF model from Step 7 that corresponds to the chosen value of the number of trees, features, and samples.

In each of the RF bootstrap training set, about two-thirds of the samples are used to fit the tree, and the remaining one-third of the observations are referred to as the out-of-bag observations [60][61]. RF reduces variance while aggregating some uncorrelated trees through averaging and avoids over-fitting. Consequently, RF brings in noticeable improvements for models such as artificial neural networks, classification and regression trees, and subset selection in linear regression. However, it can mildly degrade the performance of stable methods such as K-nearest neighbors [55]. A single decision tree's predictions are highly sensitive to noise in its training set; the average of many trees is not. Again, bootstrap sampling enhances RF performance by generating de-correlated trees. An optimal number of trees N has been found using the grid search technique.

1.3.3.5. Gradient Boosting

Gradient Boosting (GB) is an ensemble model, which learns through the optimization of prediction error gradient. GB results in low bias and low variance compared to a single decision tree [62]-[64]. GB uses a gradient descent algorithm for loss minimization recursively [63]. There are several versions of GB algorithms, and the Gradient Boosting Machine (GBM) has been adopted for drop characteristics prediction in chapter 2. Tianqi Chen has implemented GBM, and the model has been termed as XGBoost [63]. XGBoost includes several improvements over time, including overfitting regularisation. XGBoost outpaces other models with its faster performance and data parallelization [64]. Therefore, the XGBoost Scikit-learn python package is used to build a boosting model and tune parameters [51]. The gradient boosting pseudocode is represented in Table 1-6.

Table 1-6: Algorithm for Gradient Boosting.

1. Given a sample data distribution n, take a subsample of the whole dataset m as the training sample distribution (m<n).

2. Train a tree estimator using the training sample using the following steps:
   (a) Among all features, place the best feature at the root of the tree using the Gini index.
   (b) Use recursive binary splitting to divide the root data into two node subsets.
   (c) Repeat step (b) and (c) to grow tree until tree has reached maximum depth.

3. From the estimator prediction, compute the model loss.

4. Calculate the negative gradient of the model loss.

5. Optimize the weight of each estimator based on the gradient of the model loss and the training features.

6. Build GB by repeating step 1 to step 5, N times to create N number of tree estimators. The final prediction from N estimators is the summation of (N-1) estimator prediction and optimized weight from step 5 multiplied by learning rate, λ.

7. Use K-fold cross-validation to choose the number of trees N, maximum tree depth, learning rate, sample ratio of columns when constructing each tree, subsampling ratio of the training set, and the minimum number of samples in the leaf to set the stopping rule. That is, divide the training observations into ten folds. For each K= 1, . . .,10:
   (a) Redo steps 1 to 6 on all except one of the K-fold training dataset.
   (b) Evaluate the error (RMSE for regression, accuracy for classification) on the left-out one fold data out of the K-folds, as a function of GB constraints- number of trees, maximum tree depth, learning rate, sample ratio of columns when constructing each tree, subsampling ratio of the training set through Grid Search CV. Average the results for each hyper parameter and pick the value that gives the best score.

8. Return the GB model from Step 7 that corresponds to the chosen constraints.

Each recursion update is scaled with the learning rate, λ, and finally, all the predictors are combined with different weights for each predictor. At the final prediction stage, each model's

error rates are kept track because better models are given larger weights. GB builds trees greedily as each of the estimators starts from a single leaf and iteratively adds.

1.4. Printed Pattern Control through Vision Approach

Electronic material (e.g., conductors, semiconductors) patterning at micrometer scales can suffer from several non-idealities due to unwanted fluid flow on the substrate. Pattern deviations may arise in the form of bulging, scalloping, and bead segregation at the beginning, junctions, and over the intended pattern boundaries [75] shown in Figure 1-7. The printed dimension may deviate significantly from the designed dimension due to substrate wettability and ink rheology. This has a severe influence on the optimization, consistency, and accuracy of electronics manufacturing. Moreover, electrical resistance depends on pattern length and width after printing. It changes significantly with the printing direction's orientation relative to the pattern orientation and the edge roughness of the pattern [76][77]. Depending on the standard deviation of the edge roughness, it can increase device resistance and capacitance by up to 20%-30%. Therefore, it is essential to print patterns without such defects and make it readily adjustable concerning the underlying large-area flexible substrates.

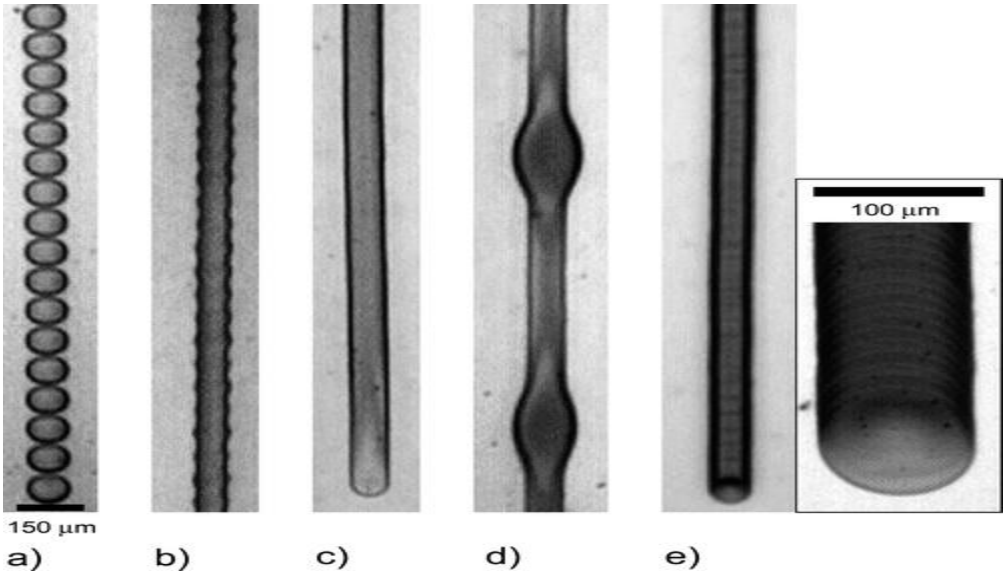


Figure 1-7: Examples of printed pattern behaviors: (a) individual drops, (b) scalloped, (c) uniform, (d) bulging, and (e) stacked coins. Drop spacing decreases from left to right. Reprinted with permission from (Soltman, D. and Subramanian, V., 2008. Inkjet-printed line morphologies and temperature control of the coffee ring effect. Langmuir, 24(5), pp.2224-2231). Copyright © 2008, American Chemical Society.

There has been significant work to understand the fluid mechanics of these non-idealities and avoid them for simple designs such as lines or squares by changing the location and order in which drops are printed. Sequential design and pattern splitting promising exhibit results in bulging inhibition and optimized output compared to traditional raster printing [78]. Inkjet pattern control has been thoroughly investigated for lines [75][79]-[82] and 2D shapes [83]-[86] by characterizing ink and substrate properties and manipulating drop, and line-to-line spacing [87][88] can potentially enhance the quality of printed morphologies. Careful selection of optimum drop spacing [75][80][81] and line-to-line spacing[80] can generate stable lines and films with less bulging instability. However, this is typically not implemented for automated industrial inkjet-manufacturing. Even line and film edges with consistent corners can be attained by managing coalescence speed, drop spacing, and viscosity-surface tension ratio [79][81][87]. Additional co-solvents [83], advancing receding contact angle [79][84], and substrate roughness[85][86] play dynamic roles for homogeneous raster-scanned line and rectangle pattern generation. Micro-scale liquid tracks with several distinct morphologies: cap, bulge, and ring have been investigated before using fluid dynamics methods [88][89]. Preprinting any feature's contour improves corner morphology by inducing or enhancing contact angle hysteresis, and additional anchoring in front of each segment can significantly subdue the bulging effect [85][88]. Other works [82][90] have demonstrated the circumvention of the multi-line intersection thickness irregularity and minimum line-to-line separation using compensation techniques in all-inkjet printed processes. Most of them describe pattern correction by adding or reducing pixels from particular areas to improve downscaled print resolution without the implementation of pixel ordering. Another practical problem is that compensation is not applicable for shapes consisting of less than five droplets width and length. The pixel compensation method is not automatic and intelligent for arbitrary patterns, and the pattern defect quantification is not versatile and online, i.e., real-time during printing. However, practical applications demand intricate patterns, for example, with multiple corners. Manual optimized drop sequence design using the above described methods for such complex patterns is challenging. Thus, most printers use simple raster printing where the nozzle follows subsequent rows without consideration for the desired pattern. Here, we propose to deploy machine vision algorithms for optimizing computerized printed electronics manufacturing. The goal is to automatically generate improved patterning steps based on automated detection and elimination of

printing inaccuracies. We demonstrate several algorithms to convert different design patterns into drop sequences. By ordering droplets' deposition and changing the coalescence forces between droplets, the functional material, underlying substrate, and evaporation do not need to be changed, which can be undesirable when printing functional materials for printed electronics. We make the optimization more versatile for any shapes and patterns as the machine vision algorithms implement design rules, and no theoretical modeling of the fluid mechanics is needed. We validate and compare the various methods by micro inkjet printing with a conducting ink. Our results show significant improvements in print quality over simple raster printing. The automated printing process is divided into two major parts: computer vision informed optimized printing pattern generation through pixel sequencing and evaluation of the printed pattern.

We propose that electronics engineers will be able to automate inkjet designing, defect localizing, and categorizing in terms of vision-based segmentation and skeletonization approaches to synchronize the printed electronics inspection environment. Image-based pattern recognition and visual inspection have been adopted before for inspection of printed circuits board (PCB) manufacturing processes [91][92]. There have been multiple uses of a divide and merge algorithms and graphical modeling for PCB pattern inspection [93]. Still, for additive manufacturing, this method of consecutive analysis of multiple defects has not been explored before. The best possible circuit integration and customization often get limited with standard PCB orthogonal interconnect architecture. Layer by layer circuit image to printing pattern translation is more flexible than PCB as it allows not only as many layers as required but also adaptable interconnections with lower manufacturing costs, lead time, and micron-scale feature size. Skeletonized defect routing of intricate printed patterns is more feasible without incorporating any design rules proposed by some researchers for PCB optical inspection [91]. We show here for industrial flexible electronics manufacturing facilities, vision-based defect routing yields high-resolution devices enabling any shape at any scale getting printed on the fly.

1.4.1. Edge and Contour Detection

Edges are the fundamental feature of image processing. Contour is built on top of edges most of the time. After detecting the intended edges from images, contours of objects can be obtained using fitting. Accurate and fast contour detection can aid in many other algorithm developments such as classical image segmenter, detectors, and shape analyzer [94]. The edge of an image illustrates the content object boundaries with a strong intensity gradient. Edge is one of

the essential features in computer vision [94]. This feature information can potentially be used to develop detection applications, measure object size, segregate, and count objects other than background. There are many ways to extract this feature and can be categorized mostly into two: Laplacian and gradient-based [95]. Filters are used to identify sharp changes. Most of the filters detect edges based on discontinuities in intensity based on three fundamental steps. At first, the image is smoothed to suppress noise. Next, using the thresholding technique on the filtered image, important edges are retained. Finally, edges are localized with thinning and linking. Various edge detection algorithms and their performance on patterns have been explored in Chapter 3 in detail.

Object contour is another critical attribute widely used for developing any image-based pattern generation, segmentation, and classification application. Contours outline the shape or form of an object in an image. When not closed, contour extraction gets quite challenging. Classical contour detection techniques fall into any of these three types: pixel-based, edge-based, and region-based [96]. Besides these traditional ways, deep convolutional neural network approaches are gaining popularity nowadays due to its better detection performance. We have adopted edge-based contour detection for automated pattern sequence generation. The detailed model and results are presented in Chapter 3.

1.4.2. Object Detection Techniques

For printing improvement, we can partition the printing design ground truth image into various segments. The circuit is depicted in Figure 1-8(a), contains two types of objects, mainly-rectangles and circles. A multi-label (circle-rectangle) object classification model will predict wherever there is a circle or rectangle. But for printing coordinate generation, we also need to know the location of an object in the designed image. This object localization helps identify a single object's location in an image, as in Figure 1-8(b). The designed circuit might contain multiple objects present, and so we have to implement object detection (OD) to tell the location and type of pattern the same as Figure 1-8(c). OD not only classifies the patterns but also puts bounding boxes around them. It involves two main challenges: one is to determine the bounding box size and location; another is to classify the pattern within the box. For computation efficiency, instead of working on the whole image, it can be segregated into smaller regions. All portions of the image don’t carry useful information, such regions are left out, and the rest are kept. Mostly, the similar image pixels are clustered together, and the region grows. This OD is essential because

depending on the type of pattern, the printing algorithm might be different, discussed elaborately in Chapters 3 and 4.

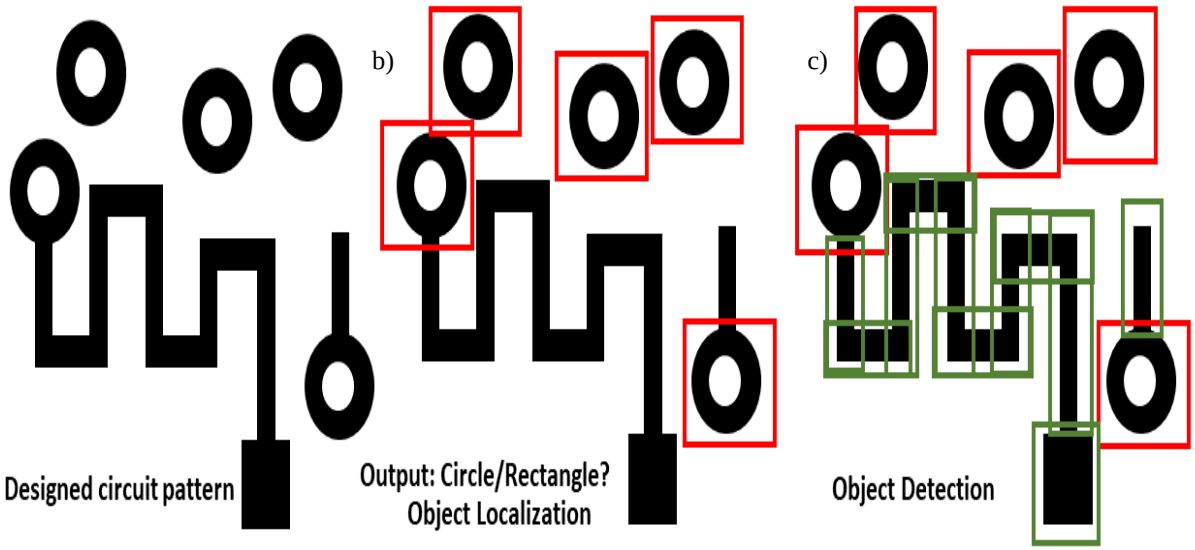


Figure 1-8: Object detection techniques. (a) Ground truth designed pattern. (b) Object localization. (c) Object detection.

Two major approaches have been explored by the researchers so far for solving the OD problem. The first one is the two-stage region-proposal-based methods. It works by proposing the "regions of interest" (areas which likely contain an object) and then refine the areas and categorizes the object within them. In contrast, single-stage methods treat OD as a regression problem, propose bounding boxes and classification in a single step. The region-proposal-based method can be traditional vision-based “Selective Search," such as graph-based segmentation. It can also be built from the CNN classifier, namely R-CNN, Faster R-CNN. Nevertheless, the region-based approach is comparatively inefficient in terms of time consumption though accuracy is better. Two-stage object detection methods first propose regions containing objects and then go through a refinement stage. Single-stage methods simultaneously predict bounding boxes, and categories such as YOLO (You Only Look Once) versions 1 (2016), YOLO 2 (2017), and YOLO 3 (2018), SSD (Single-Shot Detector, 2016), Retina Net (2018). These networks are much faster at processing images than the two-stage detectors since there is only one stage of processing, but lag inaccuracy.

For some of the simpler processing steps mentioned in Chapter 3, graph-cut based image segmentation has been adopted. This technique provides one of the best classical image segmentation results. It divides the image into multiple segments based on defined object features

such as edges, contours, lines, etc. Eriksson et al. introduced a novel graph cut techniques to segment coral reefs and holiday pictures [97]. Later, Uzkent et al. efficiently tuned graph cut for segmenting cardiac pictures [98]. Recently, there has been a tremendous advancement in deep-learning-based object detection applications employing convolutional neural networks (CNN). As a backbone, CNN has directed many high-performance applications starting from object detection [99]-[103], semantic segmentation [102][104]-[108], instance segmentation[109], and image generation[110]. The main advantage of the deep segmentation method comes from the fact that NN can very efficiently pick up features (edges, corners, contours, texture, etc.) in pixel scale from megabytes to terabytes of training objects [111]. As a supervised model, CNN outperforms conventional edge or contour-based image segmenters in terms of image understanding accuracy. With its deep layer-wise structure, it is proficient at learning complex and intricate high-dimensional data structures. It can also detect objects in any images even if it has not seen it before during training [112]. However, despite significant advances in image recognition algorithms, the implementation of these tools for practical applications remains challenging[116] because of the unique requirements for developing deep-learning algorithms that necessitate the joint development of hardware, datasets, and software [116][117]. In the field of materials science, lots of intended object detection advancements have been made [118]-[120] using image recognition algorithms. In this work, we implemented graph-based algorithm for pattern segmentation. Neural network based architecture can also be implemented for efficient detection of patterns.

1.5. Thesis Organization

This work presents how learned printing enables efficient material and printing parameter selection, consequently speeding up the development of novel materials and inks for printed electronics by eliminating money, time, and material intensive jetting experiments. A computer vision-based scheme is then developed and evaluated for the control of inkjet printing at the micrometer scale. Finally, printed results are analyzed using automated defect detection. The developed method improves print quality for complex shapes in an automated fashion, which is impossible or requires extensive manual intervention with traditional methods.

This thesis consists of five chapters. Chapter 1 includes the thesis introduction, research motivation, background and literature review, research outcome objectives, and the relevant references.

Chapter 2 describes the piezoelectric DOD jetting optimization based on machine learning algorithms. It studies the effect of printing and material parameters on drop features: velocity and radius. The jetting region with several solvents and inks with a DOD printer has been analyzed, and a dataset has been prepared for model development. Models are explored, validated, and interpreted for predicting drop features and jetting criteria. A version of this chapter has been published as a conference poster presentation: F. P. Brishty, R. Urner, G. Grau, "Machine Learning-Based Data-Driven Approach for Optimized Inkjet Printed Electronics," MRS Fall Meeting & Exhibit, Symposium MT02: Closing the Loop—Using Machine Learning in High-Throughput Discovery of New Materials; 2019 Dec 1–6; Boston, Massachusetts: MRS; 2019. MT02.12.02. A journal paper is in preparation.

In chapter 3, multiple vision algorithms and approaches for inkjet pattern generation are compared using python programming. Fabrication of flexible high-resolution silver and copper patterns with several shapes, patterns, and scales on glass and PET via inkjet printing are discussed along with theoretical analysis and experiments. Finally, it concludes which kind of patterning techniques are useful for shapes as determined from vision and graph-based defect quantification techniques. A version of this chapter has been submitted as a journal paper: F. P. Brishty and G. Grau, "Machine Vision Methodology for Inkjet-Printed Pattern Generation and Validation."

Chapter 4 presents and verifies electrical measurements of several patterning shapes and techniques. It analyses how the segmentation-based matrix vectorization was developed and adopted to improve pattern generation and evaluation methods. Finally, pattern manufacturing was carried out using segmentation techniques. After printing, printed results were analyzed using a non-contact based vision approach. Contact-based probing was performed for the pattern resistance, thickness measurement printed with various algorithms. A comparison of the methods demonstrates and verifies how segmentation can effectively improve printed patterns.

Chapter 5 summarizes the main findings of this thesis work and identifies directions for future work.

2. Chapter 2: Machine Learning-Based Data-Driven Approach for Optimized Inkjet Printed Electronics [1]

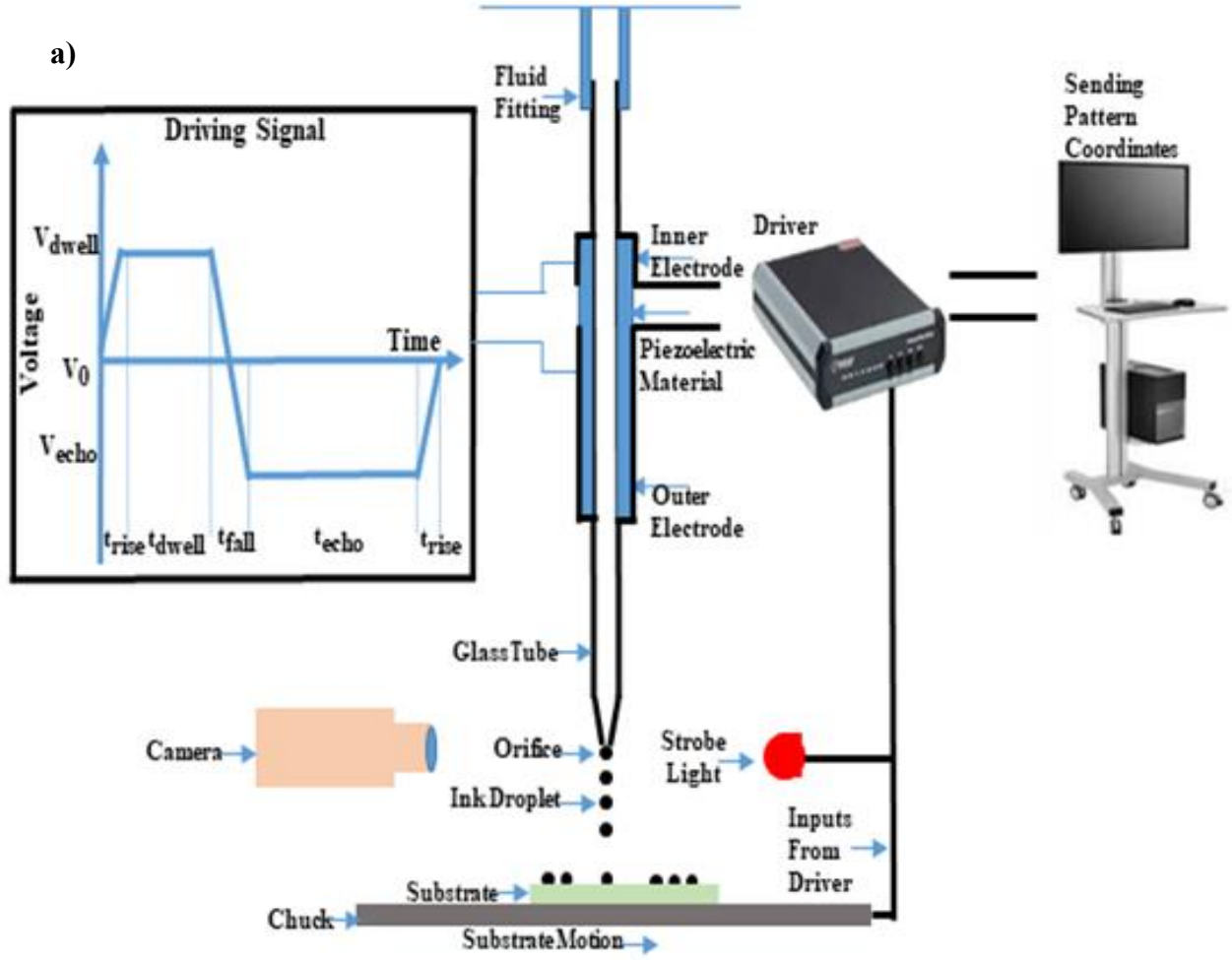
2.1. Introduction

Machine Learning (ML) predictive methodology can potentially minimize the inkjet printing configuration workload. This essential additive manufacturing technique brings in plentiful attractive attributes, including low-cost, scalability, noncontact printing, and microscale on-demand customizations. Inkjet generates circuit material droplets with a piezoelectric dispenser controlled through frequency, voltage pulse, and timing parameters. The significant challenge is rapid optimization of stable jetting conditions while preventing common problems (no ejection, perturbation, satellite drop, multiple drops, drop breaking, and nozzle clogging). Material-consuming trial error experiments are replaced with jetting learning windows based on machine and material properties through extracted literature data merged with experimentally collected data points. After the first stage exploratory data analysis and feature identification, detailed analysis is carried out to compare various (linear and non-linear) regression models to recognize models with high predictive capacities while at the same time allowing for interpretation of the underlying implied dependencies of the involved features. The models are trained on 80% of the data, and root means square errors are calculated on 20% test data. Simple linear relationship consideration between the input and output features yields coarse prediction. Instead, small ensembles of decision trees (boosted decision trees and random forests) are explored further to estimate drop velocity and radius. The models are validated with an experimentally collected graphene oxide (GO) data set not included in the training set. Consequently, several classification algorithms are utilized for drop categorization. Learned printing enables efficient material and printing parameter selection speeding up the development of novel ink materials for printed electronics by eliminating money, time, and material intensive jetting experiments.

Typical DOD inkjet printer ejects ink droplets from a piezoelectrically driven nozzle (see Figure 2-1(a) for an illustration) with micro-dispenser, LED strobe, ink reservoir, drive electronics

[1]. This chapter has been published as a conference poster presentation: F. P. Brishty, R. Urner, G. Grau, "Machine Learning Based Data Driven Approach for Optimized Inkjet Printed Electronics," MRS Fall Meeting & Exhibit, Symposium MT02: Closing the Loop—Using Machine Learning in High-Throughput Discovery of New Materials; 2019 Dec 1–6; Boston, Massachusetts: MRS; 2019. MT02.12.02. A journal paper is in preparation.

essential components. Various inks properties affect the jetting and the pattern formation on the substrate. There is only a small window of material and signal (e.g., jetting voltage, frequency, or timing) parameter combinations with stable jetting with optimal drop velocity and volume. Outside of this window, either there is no ejection from the nozzle or the jetting is unstable, and the drop breaks up into multiple droplets or satellites (see Figure 2-1(b)). For multiple droplet cases, the biggest one is referred to as the primary drop, and the other one is termed as the secondary drop. Drop quality is critical for successful manufacturing using inkjet printing. If the drops break or jets with satellites, the printed lines or circuit patterns will have irregular shapes and defects. This can adversely affect the performance, yield, and variability of printed devices such as transistors. Therefore, methods need to be developed to achieve precise jetting.



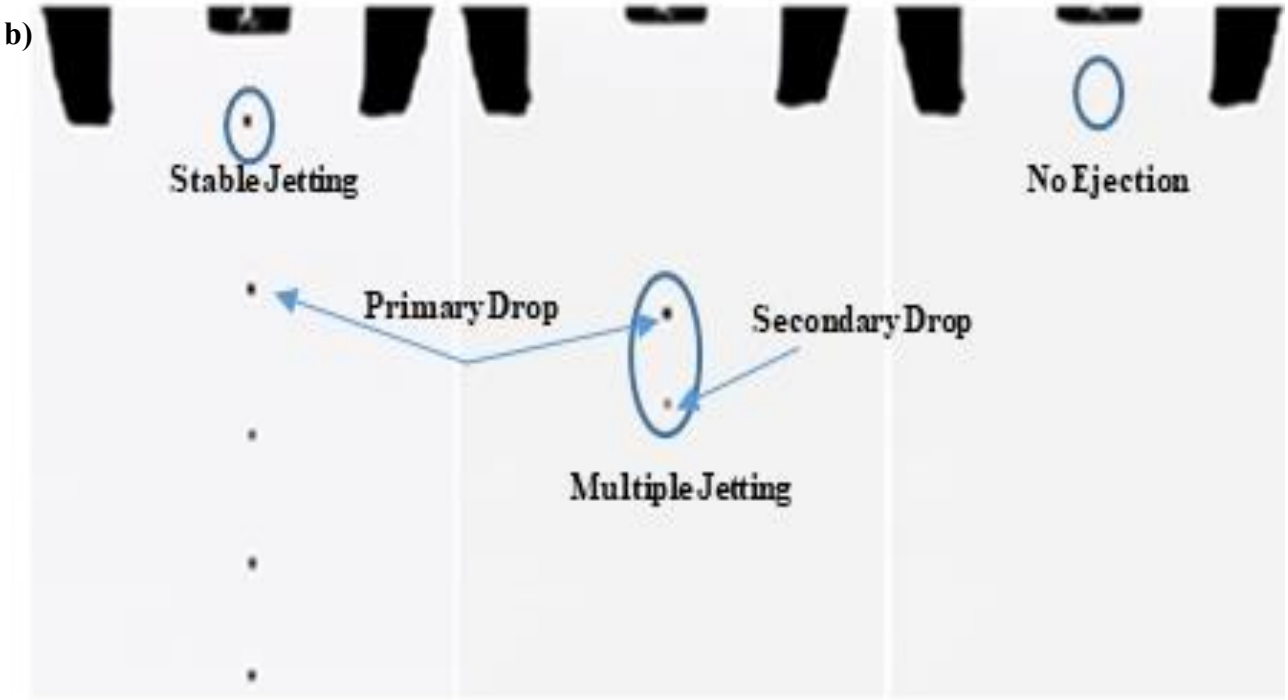


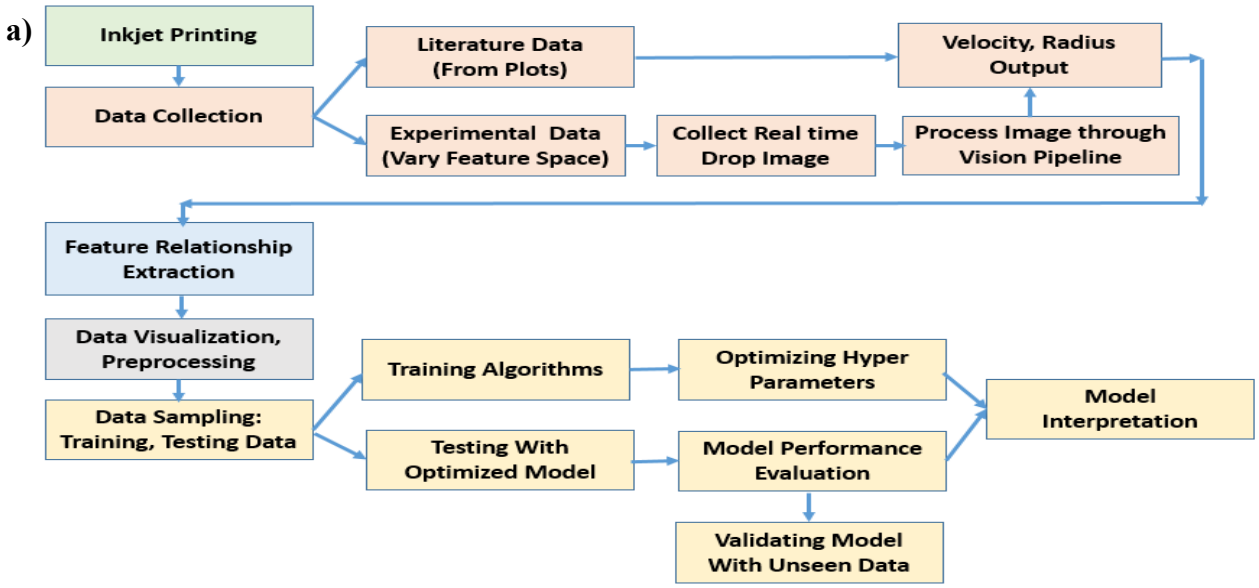
Figure 2-1: (a) Illustration of the inkjet printing process. A voltage waveform is applied to the piezoelectric nozzle, which ejects ink droplets and is scanned relative to the substrate to create printed patterns. (b) Stroboscopic images of different jetting regimes. Ideally, a stable stream of well-defined drops is created. Outside of this desirable regime, the ejected drops can break up into multiple droplets, or there can be no ejection.

Several theoretical and experimental studies have explored the underlying physics and experimental conditions for droplet generation and jetting feature characterization. For new inks, the drop characteristic such as velocity is generally not known a priori and need to be determined experimentally. Therefore, there is a need for a method that can predict jet ability, drop velocity, drop radius, and optimal printer parameters for new ink. Machine learning is a technique that can potentially forecast drop velocity and radius and categorize jetting type. It is motivated not only to derive a more accurate jetting window but also to save experimental time and cost. Only one other work uses a data-driven approach to predict an inkjet-printed polymer's drop speed and volume. This work predicted jetting from only three signal parameters: voltage, pulse duration, and rise time [134]. These features are not sufficient if the material is varied as they don’t include any fluid properties such as viscosity, density, surface tension. There is a shortage of learning-based studies

on printing and material attributes during inkjet printing at the micrometer scale. Here, we demonstrate a more comprehensive machine learning approach that takes various ink and printer parameters into account (11 features in total). The model is trained on a range of inks to learn general dependencies. This learning-based approach can help select optimum ink material properties for previously untested inks and identify the nozzle signals' active region to achieve the desired drop radius and velocity.

2.2. Methodology

The problem has been divided into three parts: 1. collect data experimentally with different printing conditions and dissimilar materials, as well as compile literature data, 2. process experimental image data and explore feature importance then relationship, 3. construct and validate predictive models for drop formulation and finally assemble results. The chart in Figure 2-2 (a) represents the workflow. Lab image datasets are processed through an image processing pipeline, as shown in Figure 2-2(b), to calculate the drop velocity and radius from images captured during the experiments. Experimental data is merged with literature data. The merged dataset is cleaned, and feature relationships are extracted. For drop velocity and radius prediction, the same input features are pre-processed, sampled, and different forecasting models are applied to the training set with hyperparameter optimization. The optimized trained models are saved and tested on the test data. Test results are interpreted with the 'Tree Interpreter' module. The respective interpretation, description of the models can be found in the result analysis section, and the relevant codes can be found at a GitHub repository [135].



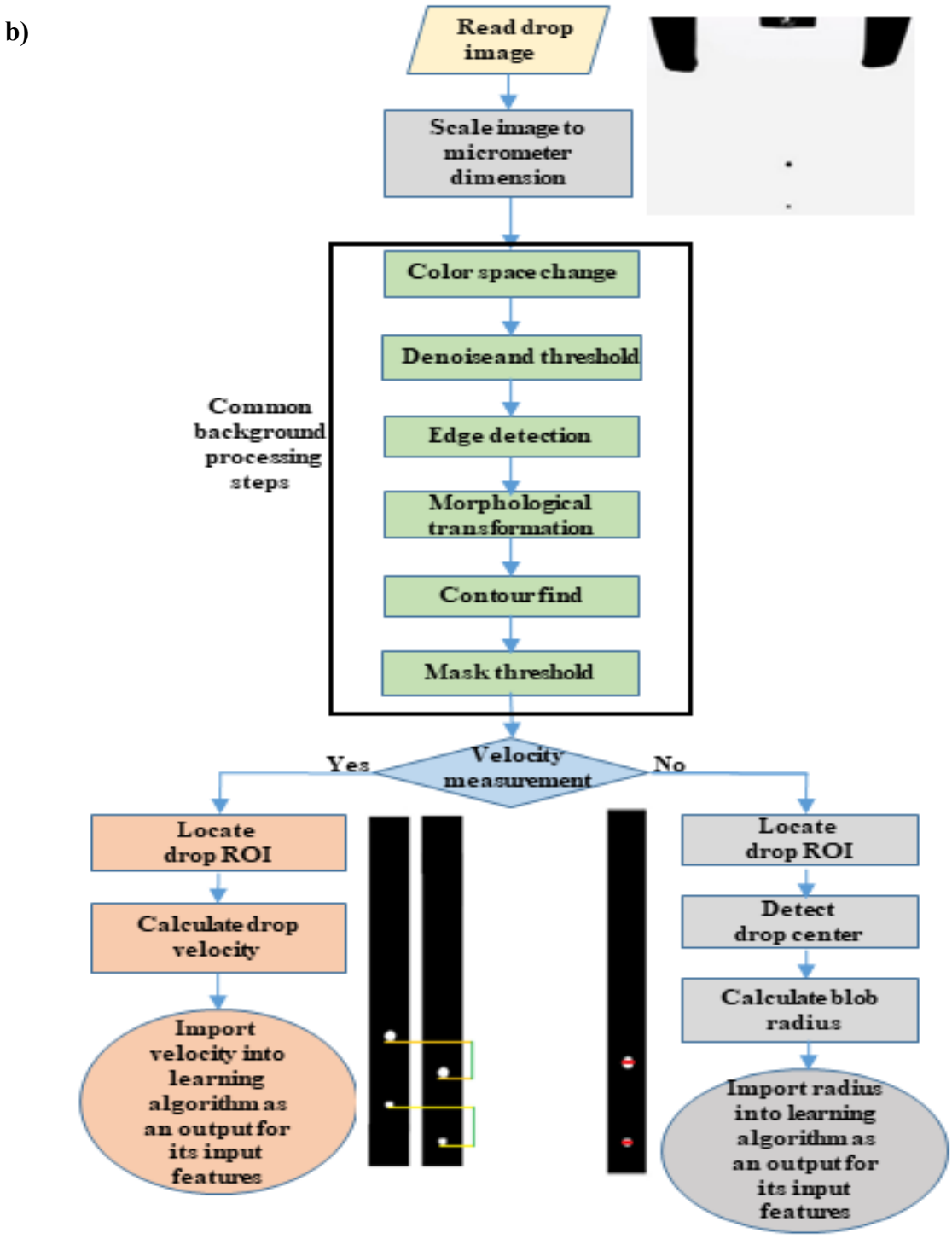


Figure 2-2: Methodological Workflow. (a) The workflow of the data-driven inkjet optimization scheme is presented here. (b) The workflow of the image processing process to extract drop radius and velocity from experimentally collected stroboscopic drop images for each set of input features studied.

2.2.1. Feature Identification

Data collection, identification of essential input features, and target output are the preliminary steps for implementing an ML model. Input data assessment is completed in two stages: varying material and machine parameters but not at the same time. As a function of these inputs, the model predicts two continuous outputs (drop velocity and drop radius) as well as one discrete output (jetting quality:single drop, no ejection, or multiple drops). Density, viscosity, surface tension was varied as material parameters, affecting material printability.

2.2.2. Machine and Material Parameter Variation

The critical machine parameters that are considered for successful inkjet printing are frequency, dwell voltage (V_{dwell}), echo voltage (V_{echo}), dwell time (t_{dwell}), echo time (t_{echo}), rise time (t_{rise}), fall time (t_{fall}) and nozzle diameter, as shown in Figure 2-1(a). Each of these parameters has its effects. The input waveform might consist of unipolar, bipolar, sinusoidal pulses; nonetheless, the "bipolar" trapezoidal signal creates more stable drops. Signal wave has a DC voltage offset level called "Idle voltage," and it is set at zero. Other than this DC voltage level, the positive and negative pulse amplitudes have essential roles to play. The positive voltage amplitude is called dwell voltage, and the time required to reach this amplitude from the bias voltage is called the rise time, providing an interval for the initial fluid expansion. The negative voltage amplitude is named echo voltage. The time duration of the dwell pulse's falling edge is labelled as fall time, which determines fluid compression and drop discharge time from the nozzle. Thus, echo voltage and its timing adjustment might potentially reduce unstable drop formation. Durations of the positive and negative voltage pulse plateaus are termed as dwell and echo time and should be sufficient enough to arrange for pressure wave propagation through the dispenser. Conventionally, V_{dwell}=-V_{echo} and t_{dwell}=2t_{echo} to allow pressure wave optimization within the nozzle. Collected literature data includes a wide variation of $V_{dwell}, V_{echo}, t_{dwell}, t_{echo}$. For the lab data collection, the feature space is divided within the following variation range:

1. Frequency: 500-2000 Hz in steps of 500 Hz
2. Rise Time: 1-35 µs (at two different voltages (30 V, 35 V)) in steps of 3 µs
3. Fall Time: 1-35 µs (at two different voltages (30 V, 35 V)) in steps of 3 µs
4. Dwell Time: 3-30 µs (at two different voltages (30 V, 35 V))
5. Echo Time: 3-70 µs (at two different voltages (30 V, 35 V))

6. Dwell Voltage: 0-80 V in steps of 5 V

7. Echo Voltage: -60 to 0 V in steps of 5 V

8. Viscosity: 0.59-15.26 cp

9. Surface Tension: 21.22-53.04 mN/m

10. Density: 786-1425 kg/m^3

11. Nozzle Diameter: 60 µm (Literature data varied from 25-120 µm)

There are some physical restrictions for choosing the feature space. According to the various piezoelectric inkjet printer manufacturers [46], there is a recommended range for creating a jettable fluid, although it varies from one company to another. Materials parameters have been chosen to lie within common inkjet nozzle manufacturers' specifications, MicroFab and Fujifilm Dimatrix. Besides, there is a range of feature values for features 1-7 beyond which the nozzle doesn't work. When the frequency goes beyond 2500 Hz or rise time, fall time, dwell time exceeds 40 µs, the jetting becomes unstable, and in most cases, the velocity and radius are not measurable. For unipolar pulse, dwell voltage can go up to 80 V. In the bipolar case beyond 45 V, mostly unstable jetting is produced with echo voltage below -80 V. Considering all these physical restrictions, the above range was selected for collecting the lab data.

2.2.3. Experimental

The experimental data points are collected, varying the identified attributes as discussed in 2.1 and merged with corresponding literature data points.

2.2.3.1. Data Collection

Five solvents triethylene glycol monoethyl ether (TGME), 2-propanol (IPA), 1-hexanol, toluene, methoxy ethanol are used. Silver nanoparticle ink (DGP 40LT-15C) with the major solvent TGME was bought from Advanced Nano–Products, Co., Sejong, Korea. To have a variation in material properties, three binary mixtures of TGME and silver ink are prepared with 70%-30%, 80%-20%, and 90%-10% concentrations, respectively. In total, data were collected for five different pure solvents and three different concentrations of silver ink. To collect jettability data, clean nozzle and solvents and inks are necessary. A customized inkjet printer with 60 µm diameter nozzle (MJ-ATP-01-60-8MX, MicroFab Technologies, Inc., Plano, TX) can handle only low viscous materials with a viscosity below 20 cP and surface tension between 20-50 mN/m. To ensure this, viscosity was measured by Brookfield viscometer, Kruss contact angle analyzer was

adopted for measuring surface tension using the pendant drop method, and density was measured through weight balance calculation. The inkjet setup was prepared by cleaning the nozzle meticulously with acetone, IPA, and de-ionized water in a sonicator. The led strobe and the camera are regulated to capture bright, clear, focused drop images without blurring. For each solvent, the same jetting setup is tested twice on two different dates. Additionally, published data for six materials (Alumina suspension in hydrocarbon media, DI Water, Ethylene Glycol, Acetonitrile, Methanol Monohydrate, and Butyl Carbitol were collected from previous literature [43][45][46] and MicroFab technote-03,04 [136] [139]. The collected literature data consists of different nozzle sizes and printers from different manufacturers for different materials but contains only velocity information, not drop radius. Literature data were collected through a web-based image extraction system called web plot digitizer. Finally, we have 769 lab data points and 2176 literature data points. Histograms of the collected lab data are displayed in Figure 2-3(a). These histograms here are immensely useful as they indicate each feature range, and the three-color labeling points out the jetting region at that time. For instance, the voltage histogram says that it is varied from 0 to 80V, and below 10V, we could not find any jetting. This data is easy to get an idea of the obtained velocity and radius range. The collected drop radius lies approximately within 25-55 µm with the nozzle radius 30 µm depending on the other features varied at that time. Figure 2-3 (b) shows that collected lab data over three different drop classes are almost equal, with no single class given utmost priority.

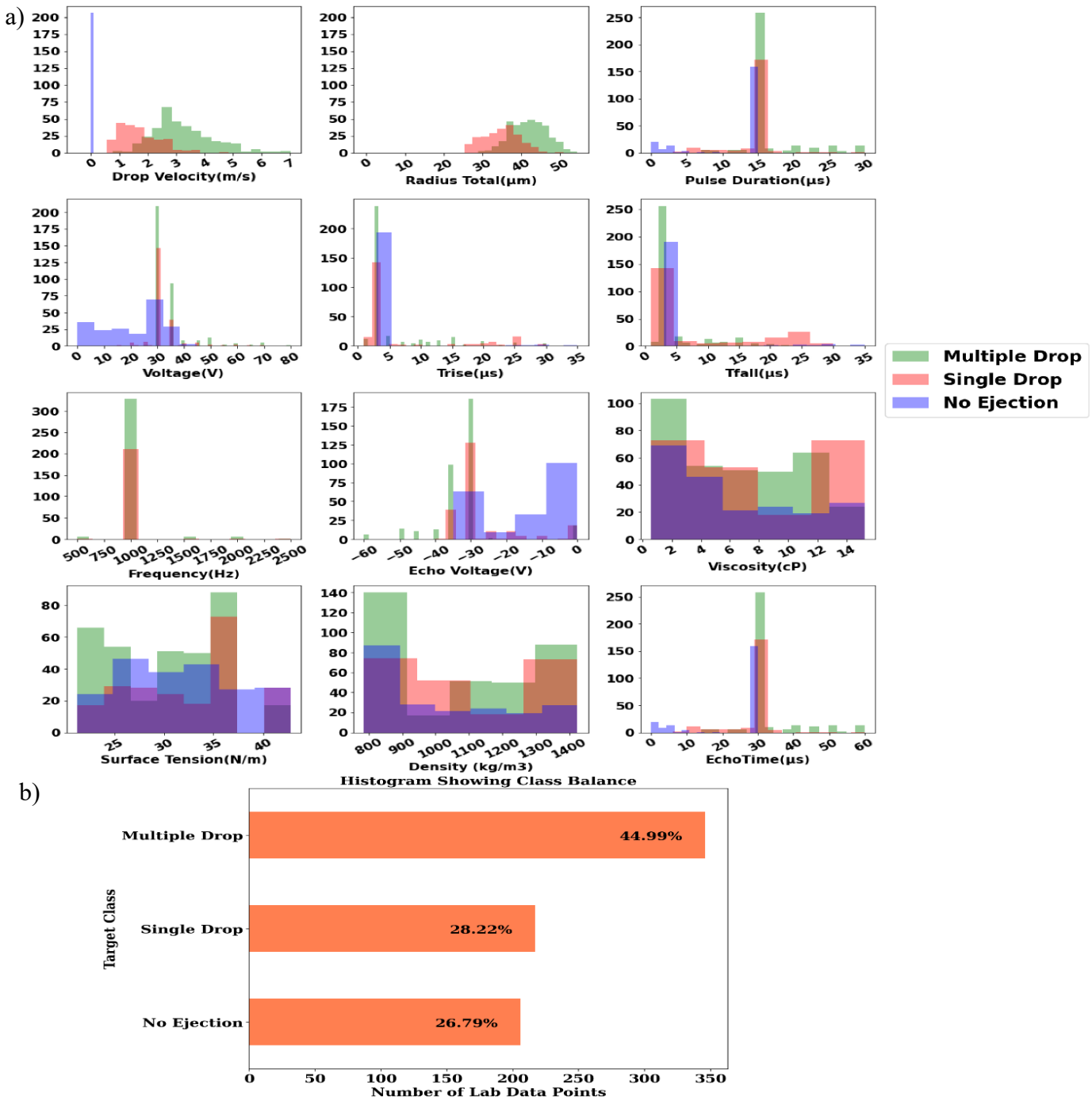


Figure 2-3: Input features and target distribution of the collected lab data. (a) Data model intake feature (dwell time, voltage, rise time, fall time, frequency, echo voltage, viscosity, density, surface tension, echo time) and measured yield (drop velocity, radius) distribution displayed as histograms with y-axis label proportional to the frequency of attribute occurrence and x-axis ticks as values with respect to observed jet ability region. (b) Target class balance of three categories of jet ability: multiple drops, single drop, and no ejection.

The percentage of data in each class is calculated by dividing the number of data points in a specific class by the total number of data points over the three classes. Shannon entropy is used here to measure class balance using equations (11) and (12). For the collected lab data with a set of 769 data points in three classes (multiple drops (346 points), single drops (217 points), and no ejection (206 points)), the entropy is computed with equation (11).

$$H=-\sum_{i=1}^{k}\frac{c_i}{n}\log\frac{c_i}{n} \quad (11)$$

Here, n is the total number of data points, k=3 is the number of classes, and c_i is the size or data count in each class. The data balance is computed with equation (12).

$$\text{Balance}=\frac{H}{\log k} \quad (12)$$

The balance value is 0 for a very unbalanced dataset, and for balanced data, the value should be close to 1. For our lab dataset with 769 data points, the computed balance is 1.069. Therefore, the dataset is considered to be overall balanced.

2.2.3.2. Image Processing

All of the mentioned features vary within their range, as described in section 2.2.1. The resulting drop image is captured with a camera using EZ-grabber version 3 with 720×480 pixels resolution. One pixel is equivalent to less than one μm. While capturing the image, a strobe LED illuminated the drop flight continuously. The generated drops are > 60 μm in diameter, and in the captured images, the drop radius ranges from 20-40 pixels. The measurement error is ± 5 % for the diameter and ± 15 % for the drop volume [141]. For each feature set, two consecutive drop images are taken using a 50 μs drop delay to measure the drop velocity. For drop radius measurement, the delay is adjusted to get a uniform round shaped drops without satellites. Figure 2-2(b) shows the image processing pipelines for the velocity and radius measurements. All the collected images are processed in OpenCV for Python performing RGB to grayscale conversion, noise elimination, binary thresholding, automatic cropping, and scaling to remove the nozzle area and keep only the generated drops. For radius measurement, SimpleBlobDetector is adopted from OpenCV, which works on thresholded binarized images (1-white background, 0-black drop foreground). In each binarized image, connected black drop pixels are grouped and form blobs. The blobs (drops) centers are computed, and blobs closer than the minimum threshold distance are merged. Finally, the centers and radius of the merged drops are computed and returned. This

algorithm performs better than other edge detection algorithms. The comparison result is displayed in Figure 2-4.

Method	Radius(um) Error(%)	Volume(PL) Error(%)
1. Canny	20.40%-22.22%	50.6%-535%
2. Laplace	6%-7.1%	18.7%-21.5%
3. Hough Transform	9.77%-11.11%	28.07%-301%
4. Sobel	100%-138%	1233.95%-85100%
5. Watershed (Matlab)	19.14%-24.76%	90.31%-700%
6. Blob Detection	**0.1%-0.4%**	**0.86%-3.8%**

Figure 2-4: Comparison of different Image Processing Algorithms for drop radius estimation.

For drop velocity calculation, every two consecutive images (taken as a drop at two sequential positions after 50 µs delay) from all the images went through edge detection. The lowest bottom point of each drop is measured. The distance traveled by each drop was calculated by the difference in the lowest bottom position in two subsequent images and divided by 50 µs to find its velocity.

2.2.3.3. Data Processing

If the data processing is not handled carefully and model selection is not appropriate, drawing insights from data through ML gets quite critical even after a lot of training. Keeping this in mind, we exerted most importance on data collection, pre-processing, and then vigorous ML model exploration. Data as the core of any ML algorithm should be supplied in the form that the algorithm understands and unlocks the meaningful patterns out of it. All in all, 'Crisp methodology' is followed for the whole data management. First, data engineering was carried on to convert raw literature data, lab data in the form of a common structured form (CSV source) for adopting ML. Literature data is unstructured. It is a combination of documents, graph image, table image, and the collected lab data is also unstructured as the target is collected as image format. Literature data sources have been parsed, joined, and put into a tabular form. The unstructured lab image dataset is passed through the image processing algorithm as defined in section 2.2.3.1 to

measure the target drop velocity, radius values. Two sets of data are merged into a final structured CSV format. Then a data processing operation is performed on the merged dataset. Data is cleaned, and unique records are kept so that each row is unique, representing a unique drop modulation test case, and each column represents a distinct feature for the case. Records with outlier drop velocity and radius were removed. Finally, we have 769 lab data points and 2176 literature data points. As a second step, feature engineering is then implemented to comprehend the features and tune them as expected by the ML model. At first, 17 input features (material name, waveform type, printer name, material mass loading, dwell time, echo time, rise time, fall time, dwell voltage, echo voltage, frequency, density, viscosity, surface tension, wave speed, nozzle orifice diameter) and three targets (drop velocity, radius, jetting category) are collected. Then irrelevant feature columns are dropped (material name, material mass loading effect can be replaced with density, viscosity, surface tension; waveform type effect come in terms of echo voltage value if zero unipolar, negative bipolar; printer name doesn't have any effect as printer attributes are well-identified with voltage, frequency and timing parameters). Next, as a third step, each input numerical feature column's quality is further improved through data standardizing (normalizing), clipping outliers using scikit-learn[142], and categorical target output (three jetting categories) are transformed to numeric representation through label encoding. The collected velocity and radius are formatted into the same scaled float values (two digits precision). Subsequently, the training and test evaluation subsets are selected as the fourth step through random sampling from the merged shuffled dataset. ML data models have been developed to have a meaningful insight and forecast drop velocity, radius, and drop type with ensemble learning. The fifth step was to select the best performing ensemble algorithm based on the RMSE for test data given by equation (13). Here finally, the algorithm performance is evaluated with untested data.

$$\text{RMSE} = \text{sqrt}\left(\frac{1}{N_p}\sum_{i=0}^{N_p}\left(y_i - y_p\right)^2\right) \tag{13}$$

Here, N_p = Total Number of Data Points, y_i = Target Value, y_p = Predicted Value

2.2.4. Model Architecture

2.2.4.1. Decision Tree

Decision tree (DT) is a white-box supervised learning procedure for discrete and continuous prediction tasks to forecast inkjet printing by learning simple decision rules from the

printer and material features. Because of simple if-then-else logic constructions, decision rules are interpretable, and prediction cost is logarithmically dependent on the number of training samples. Overfitting has been suppressed by optimizing the minimum sample data at each node and maximum tree depth. The scikit-learn [143] CART (Classification and Regression Trees) adaptation has been implemented with binary trees using the eleven features and the most considerable information gain thresholding at each node. Given the training features set x_i in R (training space), target value y in R, the data at node p is represented by Q. Each split θ=(j, t_p) contains a feature j and threshold t_p and partitions the data into two subsets $Q_{left}(\theta)$ and $Q_{right}(\theta)$ given by equation (14) and (15).

$$Q_{left}(\theta) = (x, y) | x_j \leq t_p \tag{14}$$

$$Q_{right}(\theta) = Q \backslash Q_{left}(\theta) \tag{15}$$

For velocity and radius regression tasks, the impurity at the node p is computed using the impurity function I, given by equation (16) with y_p value substituted by equation (17) . However, for drop classification, prediction output taking on values k=0 (no ejection), 1 (stable jetting), 2 (multiple drops) at node p, representing a region R_p with N_p observations, impurity is calculated with Gini index impurity is calculated with the equation (8). Here, prediction probability, $prob_{pk}$ is estimated by the equation (19) and the value is sited in equation (8). Finally, the split function given by equation (20) is estimated recursively at every node with the left(Q_{left}) and right(Q_{right}) samples impurity values until $N_p = 1 \; or \; N_p < defined \; \min_{samples}$. The minimum value of the equation (20) is the optimum number of split for CART, and the training stops after that.

$$I(X_p) = \frac{1}{N_p} \sum_{i \in N_p} (y_i - \overline{y_p})^2 \tag{16}$$

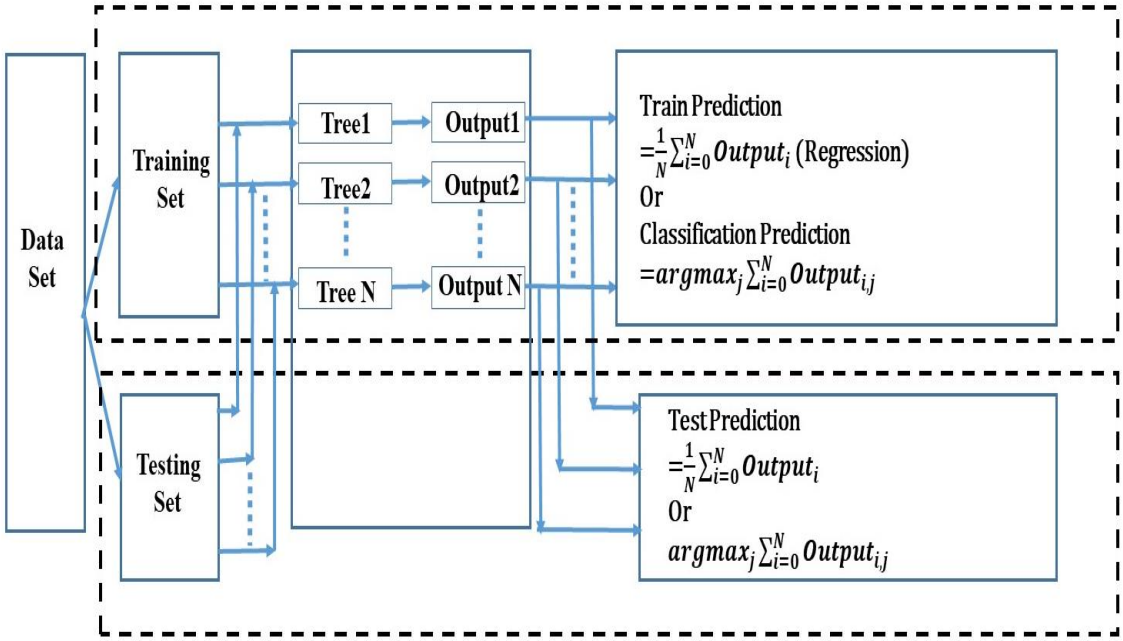
$$\overline{y_p} = \frac{1}{N_p} \sum_{i \in N_p} y_i \tag{17}$$

$$I(X_p) = \sum_k prob_{pk}(1 - prob_{pk}) \tag{18}$$

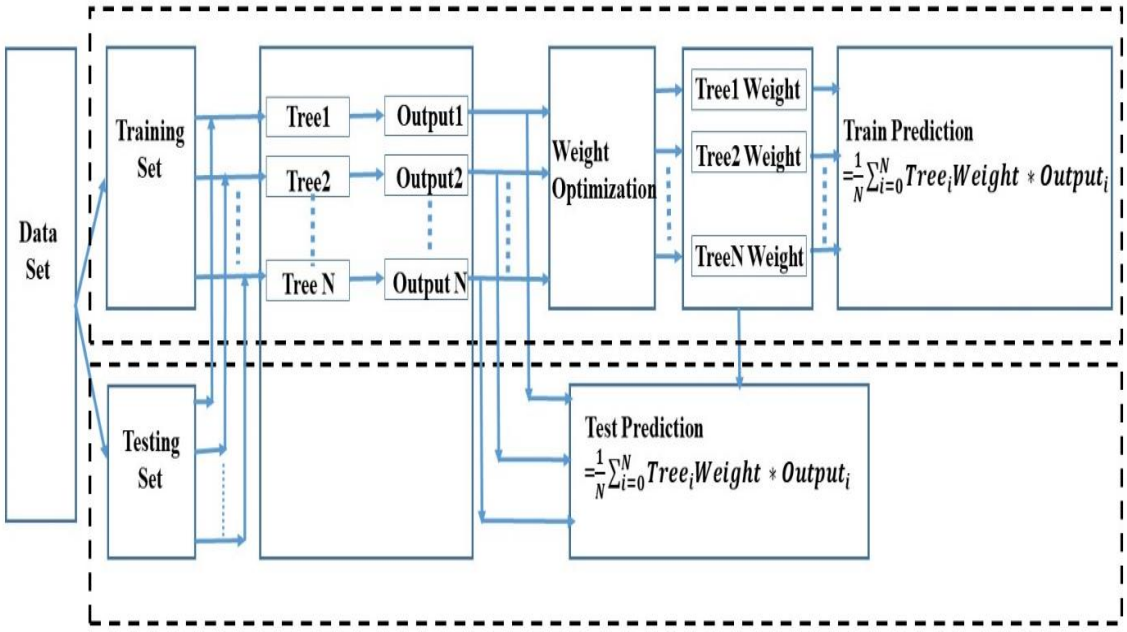
$$\text{prediction probability} = prob_{pk} = \frac{1}{N_p} \sum_{k=0}^{2} (y_i = k) \tag{19}$$

$$\theta^{min} = argmin_\theta \left[\frac{n_{left}}{N_p} I(Q_{left}(\theta)) + \frac{n_{right}}{N_p} I(Q_{right}(\theta)) \right] \tag{20}$$

a)


Data Set
Training Set
Testing Set
Tree1
Tree2
Tree N
Output1
Output2
Output N
Train Prediction
$=\frac{1}{N}\sum_{i=0}^{N} Output_i$ (Regression)
Or
Classification Prediction
$=argmax_j \sum_{i=0}^{N} Output_{i,j}$
Test Prediction
$=\frac{1}{N}\sum_{i=0}^{N} Output_i$
Or
$argmax_j \sum_{i=0}^{N} Output_{i,j}$

b)


Data Set
Training Set
Testing Set
Tree1
Tree2
Tree N
Output1
Output2
Output N
Weight Optimization
Tree1 Weight
Tree2 Weight
TreeN Weight
Train Prediction
$=\frac{1}{N}\sum_{i=0}^{N} Tree_i Weight * Output_i$
Test Prediction
$=\frac{1}{N}\sum_{i=0}^{N} Tree_i Weight * Output_i$

c)

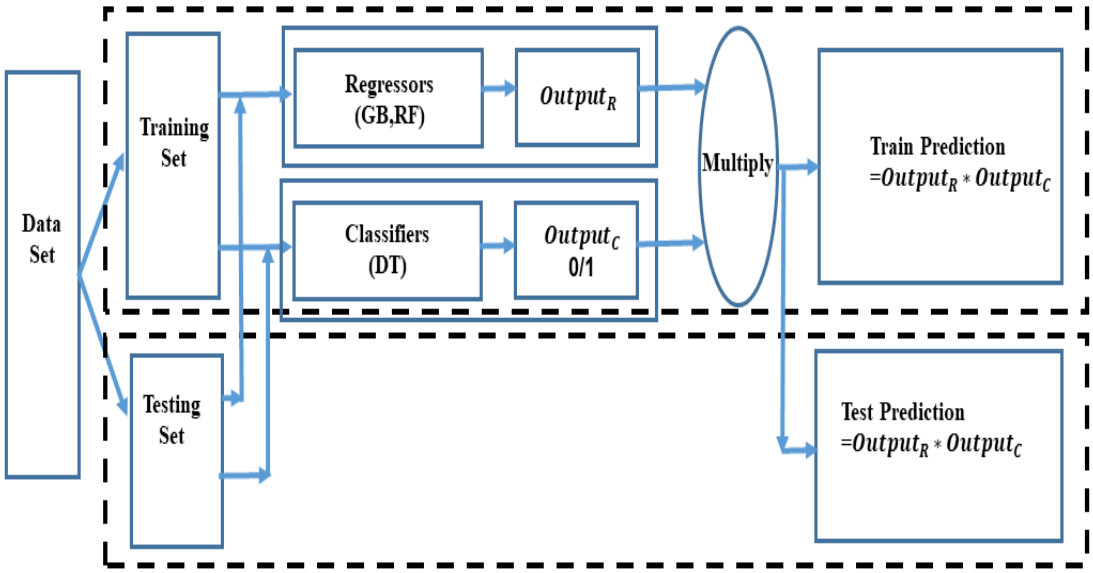


Figure 2-5: Model Descriptions. (a) Random Forest (RF). (b) Gradient Boosting (GB). (c) Merged model for velocity and radius prediction.

2.2.4.2. Random Forest

Random forest is a modified bootstrap aggregating (bagging) ensemble learning technique that builds N number of base learners (trees, linear models) by bootstrapping train data into different subset [143]. During each sampling, r (=√t) arbitrary features are chosen out of all t features to trade-off the sampling variance and reduce the learners [59]. As shown in Figure 2-5(a), random forest (RF) regressor fits N number of DTs individually on bootstrap sampled subsets of the data, and aggregates tree through majority voting (for classification) or averaging (for regression). We found the using cross-validation. It took 0.517 seconds to train the model with grid search CV cross-validation to determine the optimal number of trees N, the maximum number of features r, and the minimum number of samples in the leaf to set the stopping rule. Given a training set x_i (i=1,…,n) and targets y_i (i=1,…,n), RF performs bootstrap sampling N (number of trees) times. Sampling is done q (q = 1, ..., N) times with a random subset of the features. The sampled training and testing set are x_q, y_q are smaller than the original training set. Given a standard training set x_i (i=1,2,…,n) and targets y_i (i=1,2,…,n) of size n, RF takes out N number of training, target subsets x_{jq}, y_{jq} (j=1,2,…,N) (q=1,2,…,m) of size m, through random sampling with replacement from the original set (m<n). Then, each of the N base learners gets trained with

the sampled training and target sets x_{jq}, y_{jq} .This training data selection process is termed as bootstrap sampling. Due to sampling with replacement, some observations may be repeated in each x_{jq}, y_{jq}. If m=n, then for large n the set $x_{jq}$is expected to approximately have the 63.2% fraction of the unique samples of x_i during bootstrap sampling. Eventually, regression prediction output is calculated by averaging the predictions from all the individual regression trees, which is calculated by averaging the predictions from all the individual regression trees using equation (21).

$$y=\frac{1}{N}\sum_{i=0}^{N} y_{p_i} \text{ , } y_p = \text{prediction from each regression tree} \quad (21)$$

And for classification, the final prediction output from an ensemble of T_j (j=1,2,…,N) trees is calculated with majority voting using equation (22).

$$y=\text{argmax}_{k\in\{0,1,\ldots,C\}} \sum_{j=0}^{N} y_{p_{j,k}} \quad (22)$$

y_p=probability of prediction from each tree, k=number of class,

C=Total class

For a class k, the sum $\sum_{j=0}^{N} y_{p_{j,k}}$tabulates the number of votes for that class. Consequently, argmax function chooses the class k that maximizes the sum. RF is better than a single DT in terms of accuracy and overfitting prevention. The split criterion is based on RMSE, given by equation (13). RF improves variance while aggregating uncorrelated trees through averaging and avoids over-fitting. And, that's why RF generates "improvements for unstable procedures," such as artificial neural networks, classification and regression trees, and subset selection in linear regression. However, it lowers the K-nearest neighbor's performance [65]. Being an average of DT is immune to training noise as opposed to a single DT. Again, bootstrap sampling adds up to RF performance by generating non-correlated trees.

2.2.4.3. Gradient Boosting

Gradient Boosting (GB) [62] machine Scikit-learn [142] version XGBoost has been adopted here for regression and categorization tasks. Cross-validation training of GB model with constraint (number of trees, tree depth, learning rate) optimization took a minimum time consumption of 0.512 seconds. GB utilizes CART as base learners, as described in section 2.2.4.1 The maximum depth of each base learner tree has been chosen through the grid search CV. The number of leaf nodes can be maximum (the depth -1). Given a standard training set x_i (i=1,2,…,n), targets y_i (i=1,2,…,n) of size n, and N number of base learner trees h_m (j=1,2,…,N); GB model

prediction output, $\widehat{y_i}$ for a given input x_i is given by the following equation(23). As in Figure 2-5(b), boosting fits N number of DT simultaneously on the training set and builds a recursive model F_m in a greedy fashion according to (24) regularization strategy set with learning rate with λ. Individual base tree $h_m(x)$ is fitted with a view to minimizing loss function given by equation (25), which is finally reduced to equation (27). The minimization equation (28) is solved via steepest descent at the current model F_{m-1}. Loss function used in equation (25) and (27) is a regular RMSE for regressors as in equation (13) and accuracy for the classification task.

$$\widehat{y_i}=F_N(x_i) = \sum_{m=1}^{N} h_m(x_i) \tag{23}$$

h_m=individual base learner output

$$F_m(x) = F_{m-1}(x) + \lambda\, h_m(x) \tag{24}$$

$$h_m = argmin_h \sum_{i=1}^{n} Loss(\, y_i, F_{m-1}(\, x_i) + \lambda\, h_m(x_i)) \tag{25}$$

With a first-order Taylor approximation, equation (25) turns into (26).

$$h_m \approx argmin_h \sum_{i=1}^{n} (Loss(\, y_i, F_{m-1}(\, x_i)) + \lambda\, h_m(x_i)[\frac{\partial Loss(\, y_i, F(\, x_i))}{\partial F(\, x_i)}]_{F=F_{m-1}}) \tag{26}$$

The first term of equation (26) is constant and removed. The derivative of the loss with respect to x_i, evaluated at $F_{m-1}$and λ are represented together by ∇_i.

$$h_m \approx argmin_h \sum_{i=1}^{n} h(x_i)\, \nabla_i \tag{27}$$

Finally, replacing the value of equation (27) into equation (26), we get the final GB equation (28).

$$F_m(x) = F_{m-1}(x)+\lambda * \; argmin_h \sum_{i=1}^{n} h(x_i)\; \nabla_i \tag{28}$$

The loss function contains the prediction errors, and it is minimized so that the predicted values are sufficiently close to actual values. The optimization of the weight, as in Figure 2-5(b), is carried out with a shrinkage rate. Each recursion update is scaled with the learning rate, λ, and finally, all the predictors are combined with different weights for each predictor. During each recursion on model F, the most successful and unsuccessful data are tracked, and higher loss

outputs are given more weight and used multiple times to train the model. At the final prediction stage, each model's error rates are kept track because better models are given larger weights.

2.2.4.4. Merged Regressors and Classifiers

Drop velocity and radius regression tasks have been implemented by multiplying regressor and classifier results. A simple decision tree is used as a binary classifier to separate the two output regions: No ejection(0), ejection (1). A significant drawback of applying tree-based predictor came out in the case of actual zero values while it was mostly getting predicted as some small nonzero values. Thus, the model needed to get learned regarding the zero to nonzero transition region for better output. The same test and train set is passed through the classifier, and the regressor pipeline and their prediction output are multiplied to get a more accurate result with lower RMSE. Without the classifier integration, the regressor mispredicts some target values as non-zero even when there is no ejection. This problem is reduced with the multiplication of 0/1 values with the regression forecast values in Figure 2-5(c). We have applied two main regressors: GB, RF, and voting, weighted averaging separately on top of these two models. The results are discussed in the following sections. Finally, three jetting criteria have been suggested for stable jetting prediction, as shown in Figure 2-1(a). A three-class categorization model labels 'No Ejection' as 0 when there is no jetting. When there is a single drop but no secondary one, the model outputs 1 for the 'Single Drop' jetting. And the rest are categorized as 2 or 'Multiple Drop'. A three-layer DNN (Deep Neural Network) having 200 nodes in the 1^{st} and 2^{nd} hidden layer with 'relu' and 'tanh' activation function respectively and 40% dropout in each was deployed with 'Keras' python package. Along with this, a simple decision tree classifier with depth three and K-nearest neighbor classifier with two neighbors are constructed, and performances are compared.

2.3. Results and Discussion

Seven hundred sixty-nine experimental data points are merged with data from academic papers [43][45] and MicroFab technotes [136] [138]. The final dataset consists of 3033 rows pre-processed through a Scikit-learn pipeline normalization, scaling, categorical encoding, and improper and missing value elimination. Eleven important features (Pulse Duration, Echo Time, Rise Time, Fall Time, Frequency, Nozzle Orifice, Voltage, Echo Voltage, Density, Viscosity, and Surface Tension) are selected through applying two different feature selection classifiers, Gradient Boosting and Random Forest. We trained in total fourteen regressive models, eleven linear models

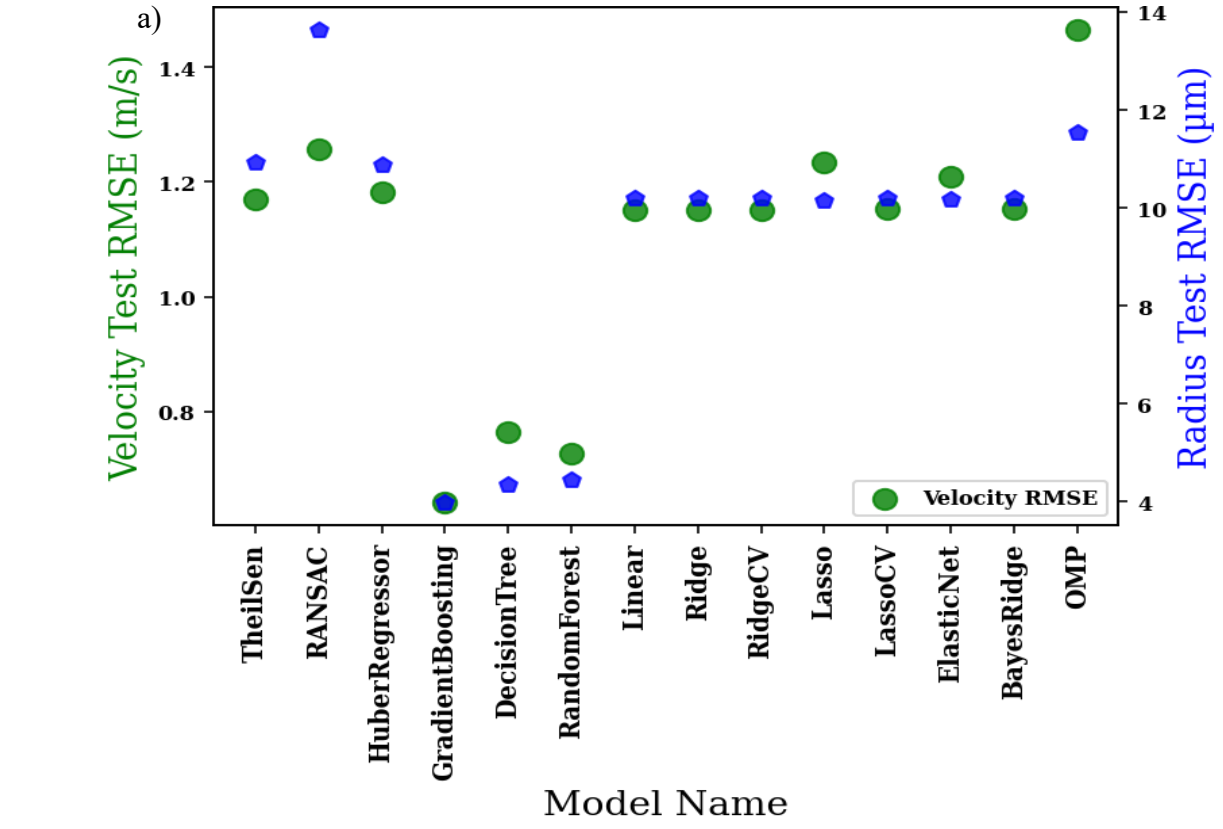
(Linear Regression, Ridge, RidgeCV, Lasso, LassoCV, ElasticNet, BayesRidge, Orthogonal Matching PursuitTheil-Sen, RANSAC, HuberRegressor) and three non-linear (Random Forest, Gradient Boosting, Decision Tree)[60]. With 80% of the whole data and the most important 11 features. Close observation on the models' root mean square error as in equation (13) on 20% test data revealed that simple linear relationships between the inputs and output, as in Figure 2-6a, do not give a good prediction. Rather, non-linear regressive models, particularly decision trees, random forest, and gradient boosting, model the underlying physics with less error. As depicted in Figure 2-6(a), lasso, ridge, support vectors do not perform well with higher test and training RMSE. Again, gradient boosting, random forest, and decision trees are doing well with lower RMSE. Therefore, after deciding to implement non-linear regressive models, we have focused on these three best models and also considered implementing averaging, weighted averaging, and majority voting with Scikit-learn[142] to minimize the RMSE of both radius and velocity prediction. The data were classified into three jetting regions: 'No Ejection', 'Single Drop', 'Multiple Drop'.

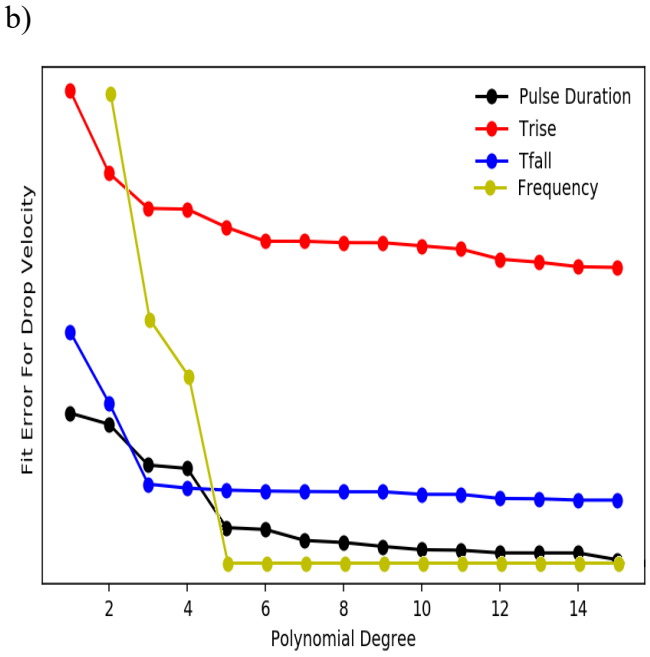
It was recognized that pulse duration, rise time, fall time, and frequency follow polynomial trends, while the others (nozzle diameter, viscosity, density, surface tension, voltage, echo voltage) exhibit linear trends. The polynomial degree that best fits the data without overfitting is determined by plotting the RSME as a knee curve in Figure 2-6(b). Around five degrees, there is a sharp decline in the error rate showing that five-degree polynomial fitting most accurately describes the feature pattern relationships. Some of the other mentionable relationships from the collected data are displayed in Figure 2-6(c-g), where dots represent experimental results, and solid lines are linear and polynomial fitting. Some vital information is obvious from these plots drop velocity and volume both show linear relationships with dwell and echo voltage while keeping other parameters unchanged, but maintain a polynomial relationship of 5th degree with the dwell time. Figure 2-6(c) shows different slope for unipolar pulses (approximately 65 pL/V, echo voltage set to zero), bipolar waveform (approximately 75 pL/V, echo voltage with a negative value), echo voltage variation at a fixed dwell voltage (approximately 55 pL/V). This means the slope of the volume change with dwell voltage is almost 1.5 times of the echo voltage for bipolar pulses. It means that the same change in volume or velocity can be achieved by applying 1.5 times echo voltage or 0.67 dwell voltage. However, the higher slope of the bipolar dwell voltage pulse results in high output volume and velocity with the same unipolar dwell voltage. Figure 2-6(d) is a histogram

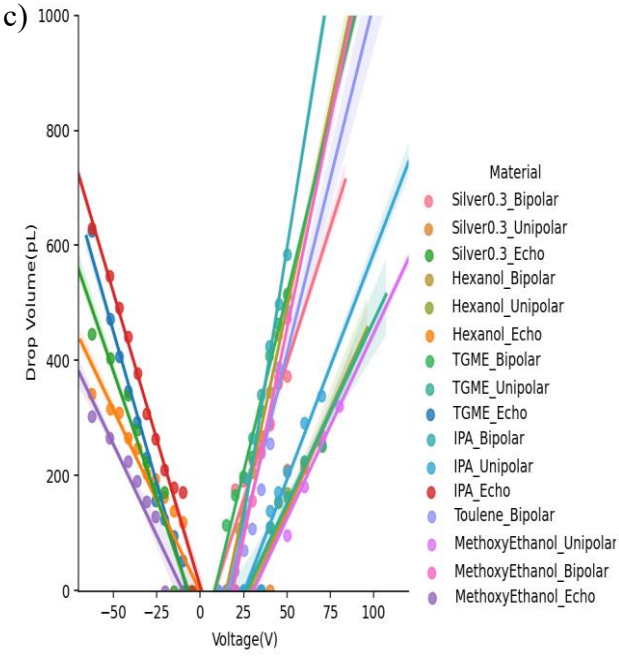
representation of minimum voltage value for creating drop ejection while other parameters are kept at a fixed value of dwell time 15, rise and fall time at 3, the frequency at 1000 Hz, and echo voltage at -30V. As shown by Duineveld et al. [161] the minimum velocity required for creating a drop is given by equation (29).

$$Velocity_{\ min} = \sqrt{\ \frac{(4 * Surface\ Tension)}{(Density * Nozzle\ Diameter)}} \tag{29}$$

The calculated minimum velocity of each of the material is plotted on the y-axis of the histogram, while the minimum required voltage value to create this minimum velocity was extrapolated from the linear fitting voltage vs. velocity curve for bipolar pulses as shown in Figure 2-6(d) . For TGME and silver, this velocity is found to be lower than the low viscosity materials. With the nozzle diameter of 60 μm (MicroFab), each of the materials with their measured density and surface tension, we calculated minimum velocity and plotted it as the bar in figure Figure 2-6(e). The measured minimum drop velocity (marked as the bar) leaving the nozzle in the lab setup is a little lower than the calculated result from equation (11) (marked as a circle) for all the materials. . Figure 2-6(f) displays that for increasing voltage (30V to 35V), there is a prominent peak shifting in optimum dwell time. The maximum ejected drop volume and velocity shift towards the right for each of the materials. For low viscosity material, two peaks can be seen with pulse dwell time. For example, IPA and hexanol show one small peak at 13, 11, and the other one at 27, 25 respectively at 30V. Optimum pulse width is the pulse width value for which the maximum velocity or volume is observed. But the value is somewhat different for every material. For the most viscous material of our experiments (silver ink), the velocity at the optimum pulse width is observed to be the lowest.


a)
Velocity Test RMSE (m/s)
Radius Test RMSE (µm)
1.4
1.2
1.0
0.8
14
12
10
8
6
4
Velocity RMSE
TheilSen
RANSAC
HuberRegressor
GradientBoosting
DecisionTree
RandomForest
Linear
Ridge
RidgeCV
Lasso
LassoCV
ElasticNet
BayesRidge
OMP
Model Name


b)
Pulse Duration
Trise
Tfall
Frequency
Fit Error For Drop Velocity
2
4
6
8
10
12
14
Polynomial Degree


c)
1000
800
600
400
200
0
Drop Volume(pL)
−50
−25
0
25
50
75
100
Voltage(V)
Material
Silver0.3_Bipolar
Silver0.3_Unipolar
Silver0.3_Echo
Hexanol_Bipolar
Hexanol_Unipolar
Hexanol_Echo
TGME_Bipolar
TGME_Unipolar
TGME_Echo
IPA_Bipolar
IPA_Unipolar
IPA_Echo
Toulene_Bipolar
MethoxyEthanol_Unipolar
MethoxyEthanol_Bipolar
MethoxyEthanol_Echo

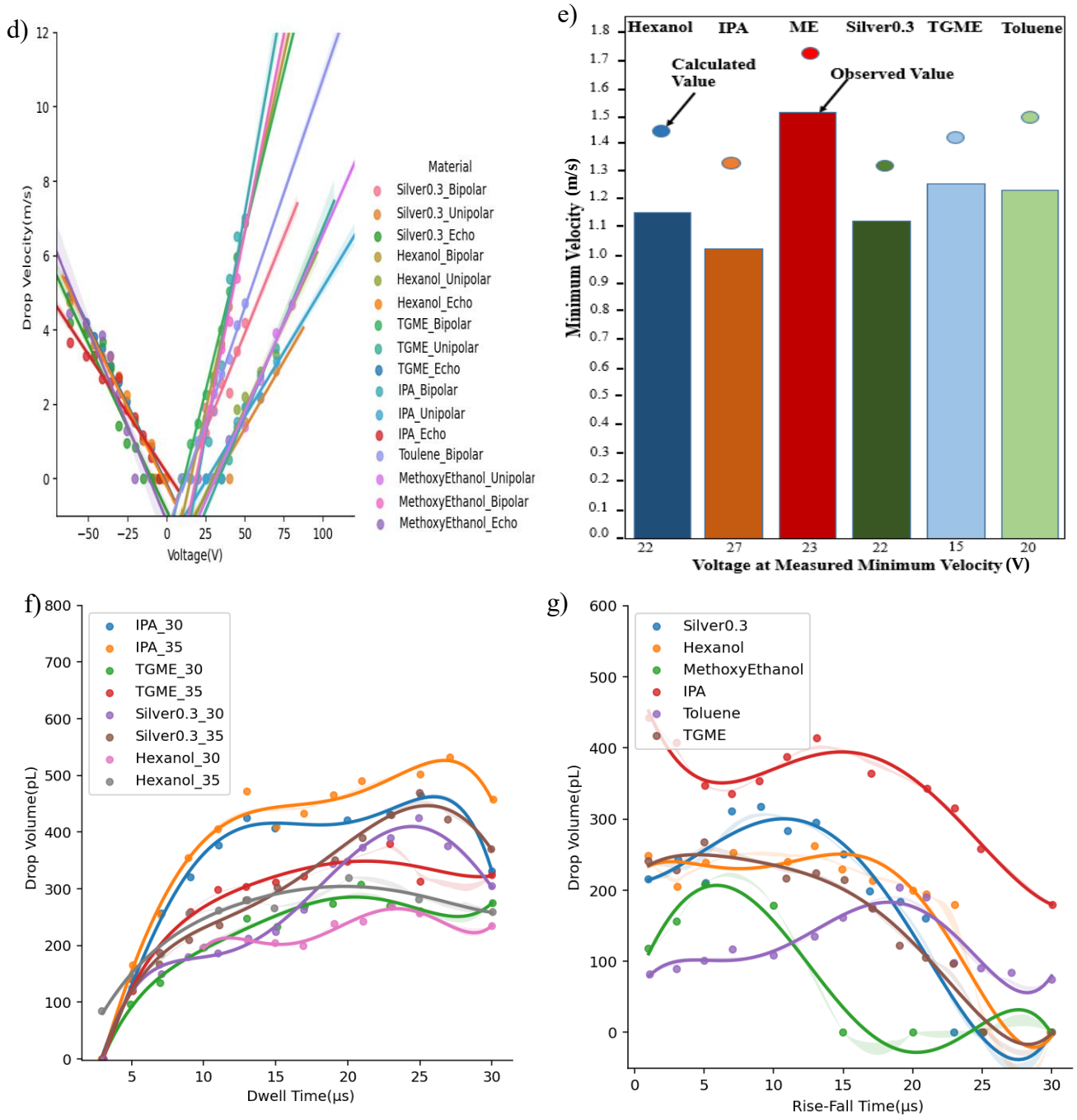


Figure 2-6: (a) Comparison of test results among different algorithms for drop velocity and radius prediction. (b) Polynomial order choosing for predicting output velocity. (c) Unipolar, bipolar, echo voltage relation with drop volume (d) Unipolar, bipolar, echo voltage relation with drop velocity. (e) The minimum velocity of ejection for different materials, (f) Dwell time drop volume relation. (g) Rise/ fall time effect on drop volume.

When rise and fall times were varied together by the same amount, the output velocity and radius maintain polynomial shapes (see Figure 2-6(g)). Drop velocity and volume behaviour of different materials are different when subject to the same signal parameters due to their differences in viscosity, surface tension, and density. But it is not straightforward to predict drop velocity and volume from these material properties only. From the above results, it is evident that the prediction of jetting is a multi-dimensional problem. The underlying behaviour cannot be easily captured by simple linear or polynomial fitting, especially without a huge dataset. In the following sections, more sophisticated predictive methods are applied to the problem.

2.3.1. Drop Velocity Prediction

To predict the drop velocity from the machine and material parameters, the three most promising models Random Forest, Decision Tree, and Gradient Boosting are deployed. Based on individual performance, ensembles of RF, GB, DT models are arranged through majority voting and weighted averaging to achieve the best performance. Tree-based ML models have a number of parameters to fine-tune, and there is no easy way to know which parameters work best, other than trying out many different combinations through Scikit-learn GridSearchCV searching algorithm [149]. Grid Search CV made use of k-Fold cross-validation [150] while exploring the best model parameter values in terms of minimum velocity prediction errors. The hyperparameter values are saved and used later on to create the best estimator representative of each model. Decision trees, as the most elementary model, have fewer parameters to optimize. DT performance is optimized with maximum depth selection through GridSearchCV taking 0.171 seconds. A pruned depth value of ten made the tree explainable and understandable, as shown in Figure 2-7(a). If it has not been optimized, the tree nodes would have been are expanded until all the leaves contain less than the minimum amount of samples (model defined) and caused overfitting. Twelve estimators of depth fourteen have been selected through GridSearchCV for Random forest's best score with an average time utilization of 0.692 seconds. In this case, RF arbitrarily chooses a subset of the eleven features for final prediction. However, the Gradient Boosting constraints, number of tree or estimators (10), maximum tree depth (14), learning rate (0.50), column sample by tree (0.8, sample ratio of columns when constructing each tree), subsample (0.80, subsampling ratio of the training set to prevent overfitting), minimum child weight (2, how big each group in the tree has to be), are selected through the Grid Search CV. Cross-validation training of the optimized GB

model combines inputs from all the ten estimators for the final velocity decision through a voting process.

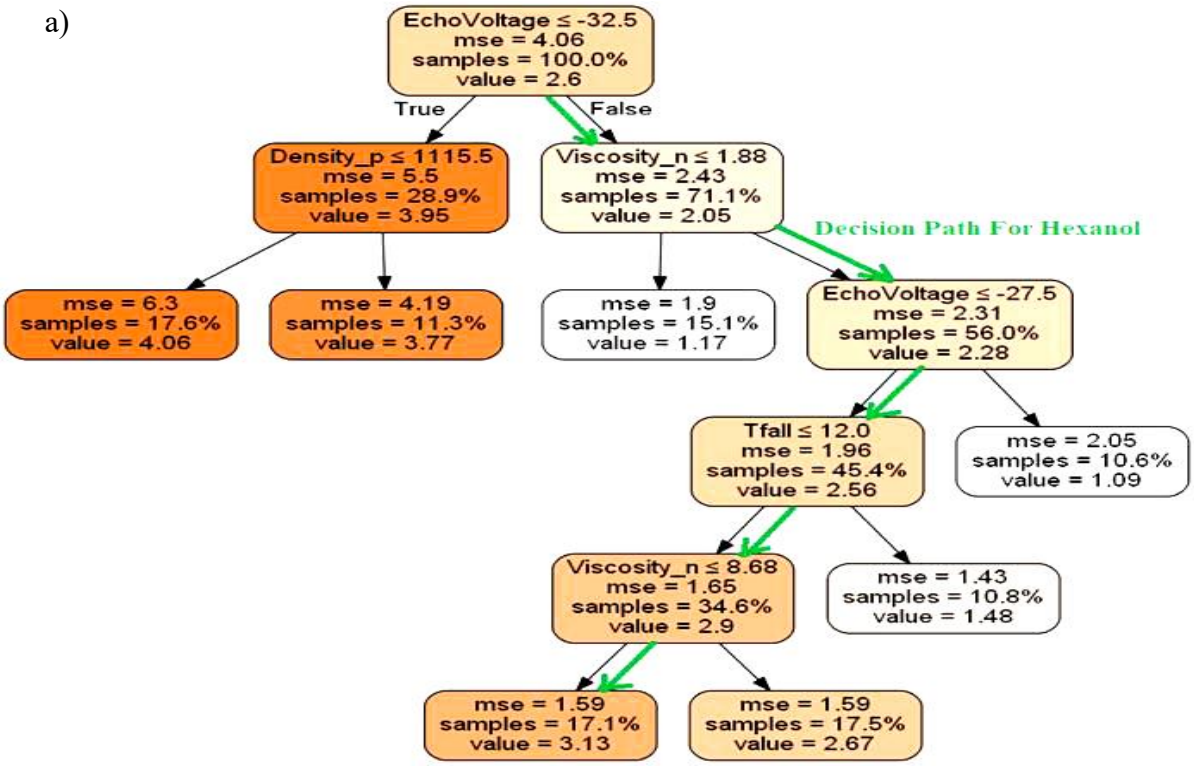
In Figure 2-7(a), we interpreted the decision tree model output with maximum depth set to five for better visualization. It shows the 'Echo Voltage' root node as the initial point for forecasting. The next split adds or subtracts a term to this sum, depending on the next node in the path. For each test data point, the path that matches the conditions is tracked, and an ultimate regression outcome is obtained. The output can be written as equation (30).

$$\text{Test}_{\text{Prediction}} = \text{Bias of Trainset Target+ Root to decision node path contributions} \quad (30)$$

It is evident in Figure 2-7(a), some features (Echo Voltage, Viscosity) are utilized in multiple splitting stages, and so they are added as contributions several times. The 'value' indicates the predicted velocity in each node. For instance, if the tree is used to predict the velocity for a test set of hexanol with Echo Voltage -30V, Density 815kg/m^3, Pulse Duration 21us, Voltage 30V, Frequency 1000 Hz, Viscosity 4.59cp, Surface Tension 25.73mN/m, and nozzle orifice 60µm; it will follow the marked green path and will result in a velocity prediction of 3.13 m/s with a residual of (4.0816-3.13) = 0.9516 m/s, which is close to the RMSE of this tree model. The contribution of Viscosity = (2.28-2.05) + (3.13-2.9) = 0.47. The bias is 2.598, the contribution from fall time is 0.339, and the contribution of Echo Voltage is -0.272. So, the overall prediction = bias+ all feature contributions = 2.598+ 0.471 + 0.339- 0.272 = 3.136. The gradient boosting constructed with ten weighted trees has a much better overall RMSE of 0.398 m/s than a single decision tree RMSE of 1.445 m/s. Weights are set on each tree output prediction, and an average is taken on them, which is the final predicted velocity. Each of the booster trees has a maximum depth of 14, and in order to calculate a prediction, gradient boosting sums predictions of all its trees. Each of the ten trees is not a great predictor on its own. The aggregated prediction from the ten estimators gives a better RMSE of 0.398 m/s. Individual predictions of each booster tree can be explained by decomposing the prediction into the bias and contribution, as shown in Figure 2-7(b). The best booster has been used here to predict the drop velocity of new data from the literature. Figure 9 of that paper [152] depicts the drop velocity change of silver ink suspension with pulse amplitude. Two pulse amplitude values (36 V and 44 V) were tested with this model. The predicted velocity is very close to the experimental velocity. Each prediction can be expressed as a sum of feature contributions and bias from all the trees. With the help of the elif5 package code of Python, the decision path of

the best tree for two test data points is broken down in Figure 2-7(b). It describes that all the boosters predict velocity 2.195 m/s while the experimental velocity was 2.0 m/s, and the accuracy is 90.25%. Notably, this particular test result's most considerable contributions are from echo voltage, voltage, viscosity, and surface tension. Bias is the mean velocity value of the training dataset. Gradient boosting trees make dissimilar contributions for different datasets, although the bias remains the same for all. The table's right side shows aggregated boosting estimation for voltage 44V as 4.468 m/s with an accuracy of 91.12%. The feature contributions are different for two different test data points as they are arranged based on their overall impact. For both tests, there is a noticeable impact of Voltage, Echo Voltage, Pulse Duration, and Nozzle Orifice. The first test result is less than the bias as the positive impact of less significant material features is not able to counterweight the voltage, dwell time, and nozzle effect. For test 02, Echo Voltage plays a more critical role than voltage, pulse duration, and nozzle diameter. So the bias is not pulled down much by other features. Figure 2-6(c) depicts the top 20 rules for velocity prediction extracted from the ten boosting trees by the Molnar rule fit algorithm [153]. These rules multiplied with their coefficients and summed to get the final prediction result out of the features. Notably, the Voltage, Nozzle Orifice, Pulse Duration constitute the most essential prediction rules. The importance column shows the percentage of data being affected by the corresponding rule.

a)



b) **Drop Velocity** (score **2.195**) top features

Contribution?	Feature	Value
+4.854	<BIAS>	1.000
+0.929	EchoVoltage	-36.000
+0.423	Density_p	1216.000
+0.078	Viscosity_n	4.830
+0.003	Trise	2.000
-0.002	Tfall	2.000
-0.051	EchoTime	10.000
-0.083	SurfaceTension	24.500
-0.104	Frequency	1000.000
-0.727	NozzleOrificed_m	30.000
-1.397	PulseDuration	5.000
-1.728	Voltage	36.000

Drop Velocity (score **4.468**) top features

Contribution?	Feature	Value
+4.854	<BIAS>	1.000
+1.996	EchoVoltage	-44.000
+0.798	SurfaceTension	24.500
+0.570	Density_p	1216.000
+0.143	Trise	2.000
+0.018	Viscosity_n	4.830
-0.033	EchoTime	10.000
-0.105	Frequency	1000.000
-0.724	NozzleOrificed_m	30.000
-1.447	Voltage	44.000
-1.602	PulseDuration	5.000

Most Important Features

Test 01: Original Velocity-2m/s Test 02: Original Velocity- 4.9 m/s

c)

Rule	coef	importan
Voltage <= 59.83 & Voltage <= 47.5	-1.815738	0.851
Voltage	0.0204548	0.644
PulseDuration	0.0424258	0.574
Voltage > 59.83 & NozzleOrificed_m > 74.67 & Voltage > 109.78 & Frequency > 13100.87 & Frequency <= 22083.01	3.4106927	0.562
Frequency > 16834.33 & Voltage > 59.83 & NozzleOrificed_m > 74.96 & Frequency <= 21914.80	2.356405	0.496
Frequency	7.60E-05	0.446
Voltage <= 49.33 & Voltage <= 59.83 & Trise > 1.25 & Voltage <= 27.5	-1.6051573	0.427
Voltage > 59.83 & NozzleOrificed_m <= 85.5 & NozzleOrificed_m > 74.96 & Frequency <= 16834.33	0.9629144	0.421
Voltage > 59.83 & NozzleOrificed_m > 74.67 & PulseDuration <= 54.33 & Voltage <= 109.78 & PulseDuration <= 41.53 & PulseDuration <= 43.93	1.6524072	0.372
Voltage > 59.83 & NozzleOrificed_m > 85.5 & NozzleOrificed_m > 74.96 & Frequency <= 16834.33	1.2735849	0.366
Frequency <= 13100.87 & Voltage > 59.83 & Voltage > 109.78 & NozzleOrificed_m > 74.67	1.2460917	0.322
PulseDuration > 54.33 & Voltage > 59.83 & Voltage <= 109.78 & NozzleOrificed_m > 74.67	-2.7946283	0.300
Frequency <= 2012.37 & Voltage <= 67.90 & Voltage > 59.83 & NozzleOrificed_m <= 74.96	-1.8330304	0.262
Tfall	-0.0447137	0.237
EchoTime	0.0171286	0.230
Frequency > 2012.37 & Frequency > 17147.45 & Voltage > 59.83 & NozzleOrificed_m <= 74.67	1.3945776	0.210
Voltage <= 59.83 & Voltage > 47.5	-0.6877347	0.202
Voltage > 59.83 & NozzleOrificed_m > 74.67 & PulseDuration <= 54.33 & Voltage <= 109.78 & PulseDuration > 43.93	0.6875505	0.190
Voltage <= 49.33 & Voltage <= 59.83 & Voltage > 27.5 & PulseDuration <= 5.0	-1.2498125	0.169
Frequency <=2012.37 & Voltage > 59.83 & Voltage > 67.90 & NozzleOrificed_m <= 74.96	-0.9699072	0.139
SurfaceTension	0.0145887	0.138
Voltage <= 49.33 & Voltage <= 59.83 & Trise <= 1.25 & Voltage <= 27.5	0.7042802	0.095
Voltage <= 70.87 & Voltage >59.83 & NozzleOrificed_m <= 74.96	-0.5090087	0.073
Tfall > 17.5 & Voltage > 27.5 & Voltage <= 49.33 & PulseDuration > 5.0 & Voltage <= 59.83	-0.4037598	0.069
Trise	0.0195856	0.067
Density_p	0.0003063	0.054
NozzleOrificed_m	-0.0008302	0.011

Figure 2-7: Drop velocity prediction interpretation. (a) Example decision tree for velocity prediction. (b) Bias and features contribution interpretation of predicted result on test data from [8] for gradient boosting. (c) Top 20 Rules extracted from 10 gradient boosting trees.

The three tree-based regressive models exhibit having difficulty in predicting zero values. For some attribute values, there is a distinct region where no drop ejection occurs, and velocity is zero, and the model performance deteriorates. This ‘No Ejection’ region has been separated from the jetting region (‘Single Drop’, ‘Multiple Drop’) with a simple decision tree classifier. The intermediate regression values are multiplied with the classifier output, as in the algorithm shown in Figure 4d. The final multiplied predicted velocity exhibits lower RMSE. To confirm that the predicted output agrees well with the real experimental results, the GB model's predicted output was plotted against experimental data for the whole dataset. Predicted velocity and test velocity agree well (see Figure 2-8(a)), although there is some residual prediction error (0.398 m/s). GB is proved to be more dynamic and better for handling the new GO ink test dataset having a viscosity 8.7 cp, surface tension 57.96 N/m, and density 1232 kg/m3. In Figure 2-8(b), the lowest RMSE velocity model (GB) is validated with GO ink. The predicted velocity displays a linear trend with voltage, and the difference between the measured and predicted velocity is within the mentioned RMSE (0.398 m/s) of the gradient boosting model.

The test results from the tuned simple decision tree, gradient boosting, and random forest models RMSE are deviated by 1.55 m/s, 0.398 m/s, and 0.45 m/s from the measured velocity, respectively, as shown in see Figure 2-8(c). DT prediction result is the worst as the difference between test and train RMSE is large. This is also a sign of some overfitting, which might arise as the DT doesn’t have regularization parameters such as the learning rate. RF is much better than DT as it selects features randomly during prediction, so the training and testing errors are pretty close. It means that it is less likely to under or overestimate the output for new untrained datasets. GB's additional regularization term and weight updates help avoid over-fitting and result in the lowest test RMSE among these three models. The best two models (RF, GB) efficiencies are enhanced by further implementing simple ensemble methods: voting, averaging and weighted averaging. In the case of voting, the test predictions from the random forest and gradient boosting are regarded as ‘votes,’ and majority voting has been adopted on top of them to get the final prediction output. The Voting Regressor module of Scikit-learn [142] is used as the voting model. For averaging, two repressors' training and test results are averaged separately to calculate the training and test RMSE of the averaging model. And for weighted averaging, three weight optimization techniques are deployed to minimize the final prediction RMSE. Three used weight optimization methods are Neural Network, RMSE Minimization, and Random Forest. Weights are

allocated on the two best base models depending on the algorithm used for weight assignment. The predicted final output is calculated using the equation (31) with the RF and GB test prediction results and their optimized weights.

Weighted Averaging= weight1 * Gradient Boosting Test Prediction+ weight2 * Random Forest Test Prediction (31)

For neural network weight optimization, the number of hidden nodes in the input layer was set to three, and the output node was set to one. The test prediction results from the two base models are fed through the input layer, and the prediction from the output layer is compared against the test dataset to calculate RMSE. For RMSE minimization, the RMSE of the output from equation (13) output minimized through function SLQP (Sequential linear-quadratic programming, an iterative optimization method for nonlinear problems) Nelder-Mead functions against the test dataset. The weighted result from different weight optimization techniques is shown in Table 2-1. From the weights optimization of different algorithms, the gradient boosting has more weight among the two base models and has the lowest test RMSE. Averaging and voting have a similar RMSE of around 0.33 m/s. The most exciting part is that weighted averaging is the best of all predictive models with the lowest RMSE and smaller residuals.

Table 2-1: Optimized weights on different models for averaging according to equation (31).

Model	RF Weight (w1) (Test RMSE 0.455)	GB Weight(w2) (Test RMSE 0.398)	Best Score
RF Weight Selection	0.444	0.556	0.354
NN Weight Selection	-0.528	0.794	3.8609
SLQP Minimized Weight Selection	0.433	0.567	0.3380
Nelder Minimized Weight Selection	0.527	0.475	0.3163
Simple Averaging	1/2	1/2	0.3267

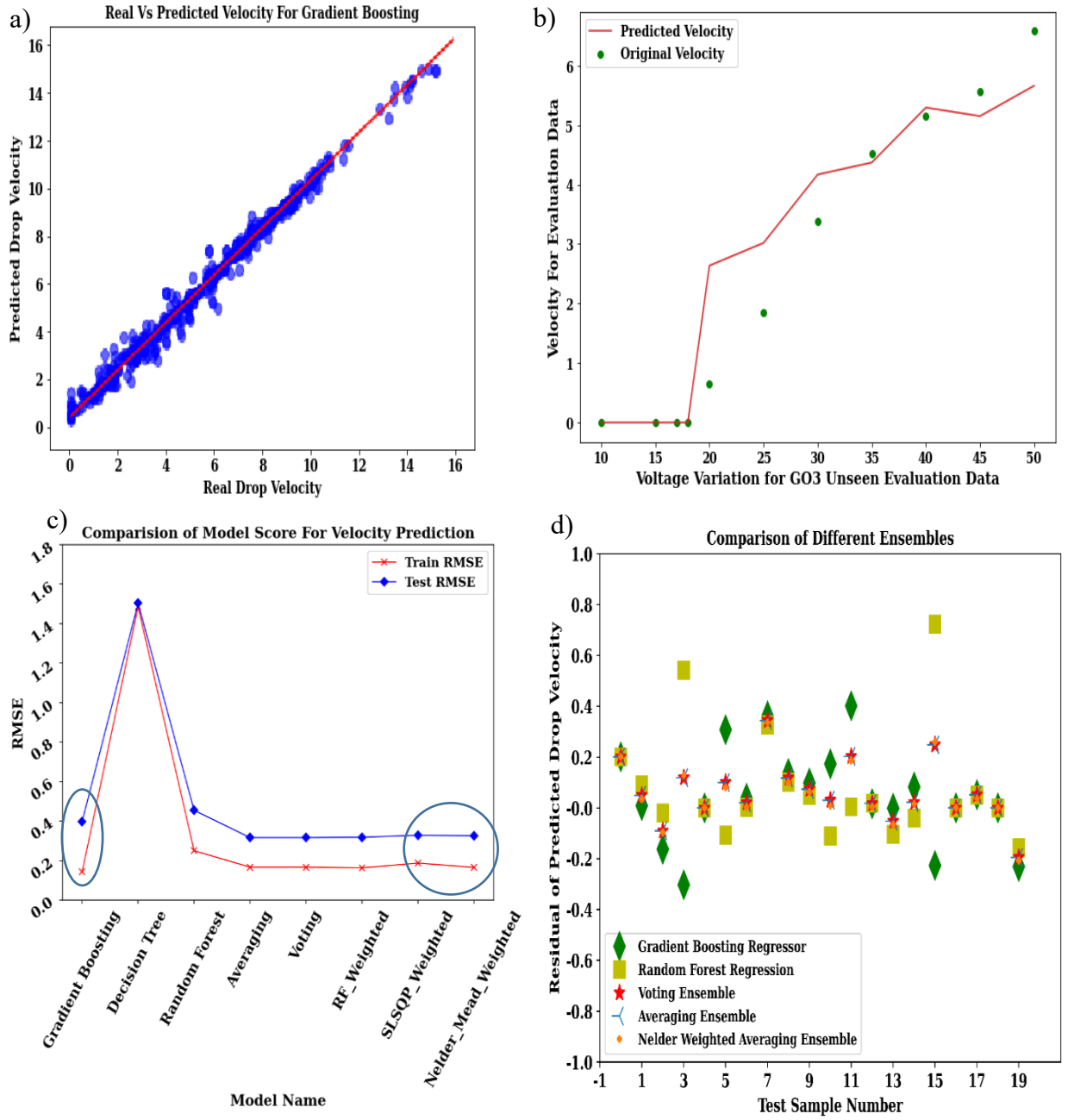


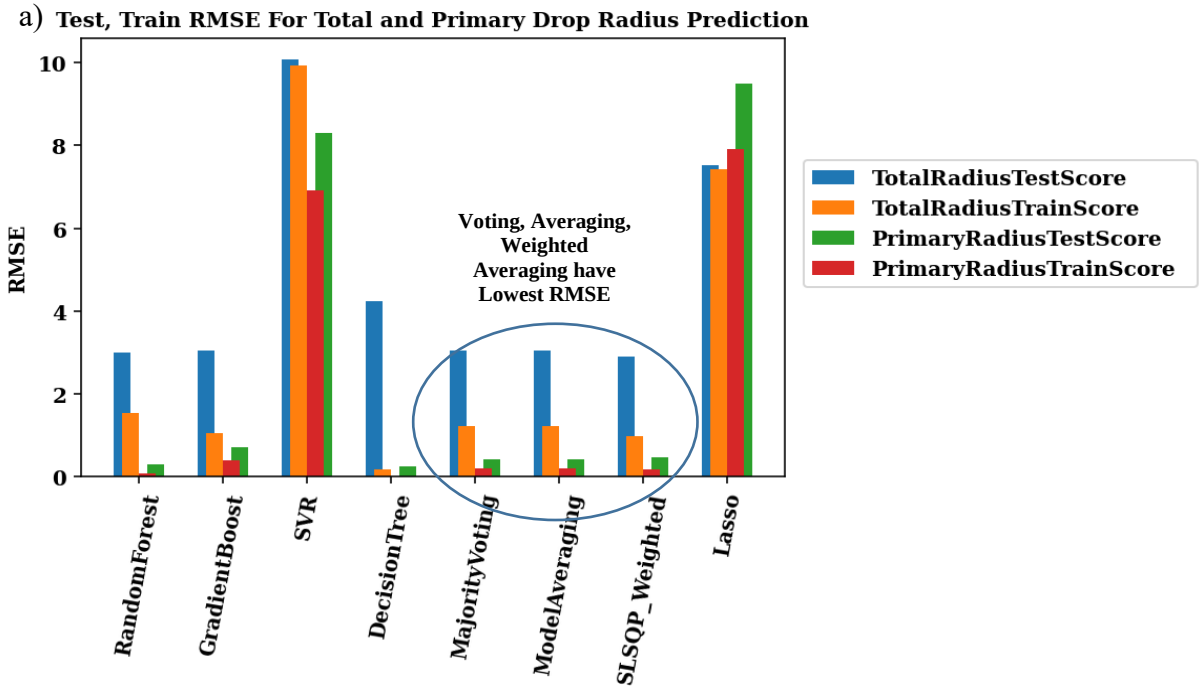
Figure 2-8: Velocity prediction models comparison. (a) Predicted velocity and real drop velocity relation justification for DT drop classification (0/1) result multiplied with GB velocity regression (Figure 4b model) from lab data. (b) Validating gradient boosting with untested GO ink (c) Test result comparison of different weight optimization models. (d) The four most promising models predicted residual (real velocity-predicted velocity) of a random 20 test sample.

The training and test RMSE of the single and weighted averaging models are gathered in Figure 2-8(c). When the weights are chosen by applying the RMSE minimization method, the prediction output improves. The small difference between the train and test results indicate compliant over and underfitting tendencies. The averaging and voting techniques do not outperform them. The prediction residuals of the two individual models (Random Forest and Gradient Boosting) are more significant than the three weighted averaged models (weighted by SLSQP, Nelder, and Random Forest). Neural regression is the worst to find out the specific weight for base models. Weighted averaging techniques RMSE is lower than the GB, RF model; their main advantages are evident in Figure 2-8 (d). Randomly 20 data points are taken out from the test samples, and the difference between the real drop velocity and the predicted velocity of the five best models are plotted. Plotting all test prediction residual is avoided as makes it challenging to realize the significant difference. Sample data point 2,3,5 reveals an exciting fact. While any of the RF, GB model is making a bad prediction, any of the weighted averaging, voting, or simple averaging produces a lower RMSE. Weighted averaging can bring in a better prediction in case any or both of the GB, RF models predict with high residuals.

2.3.2. Drop Radius Prediction

Drop radius is a better quantitative estimation of drop size than volume. It was measured through a graph-based blob edge detection algorithm from the processed drop images. The estimated error of radius from the image was around ±5% in micrometers, so the converted volume includes an approximate error of 15% in pL. For the regressive estimation, the same 11 features (Voltage, Echo Voltage, Echo Time, Rise Time, Fall Time, Pulse Duration, Nozzle Diameter, Frequency, Viscosity, Density, and Surface Tension) were used. The main challenge was that our lab data consists of primary and secondary drop volume. Here, primary denotes the main droplet. The secondary drop volume occurs due to drop breaking or multiple drop phenomena such as satellites, as shown in Figure 2-1(b). This aspect is missing in the literature data; it contains the total volume (Primary drop volume + Secondary drop volume). Therefore, the volume was converted to total radius. It was assumed that there are no secondary drops in the literature data, and they were merged with the lab dataset. Like velocity estimation, ten general models were fitted and tested on the dataset shown to identify which model is best for total and primary drop radius prediction. The total result for radius prediction is found to improve by multiplying the results from classification and regression trees, as shown in Figure 2-5(c), similar to the velocity

prediction model. After this exploration, it was evident that the secondary drop radius is erratic and does not maintain an interpretable relationship with the signal and the material parameters. Although total and primary drop radius exhibit great train, test prediction results with three tree-based regressors (Decision Tree, Gradient Boosting, Random Forest) presented by Figure 2-9Figure 2-9: (a). Weighted averaging was also adopted on the two best models (Gradient Boosting, Random Forest) described by the equation (31). The weight-optimized by SLSQP is the best with the lowest total radius prediction RMSE of 2.91 µm. Unlike velocity, most of the weight is given on GB (0.85) and rest on RF (0.16). This weighted averaging is effective for total drop radius estimation. However, majority voting among GB, RF tends to give better results for primary drop radius prediction. The collected test and train RMSE of each of the models are shown in Figure 2-9: (a).



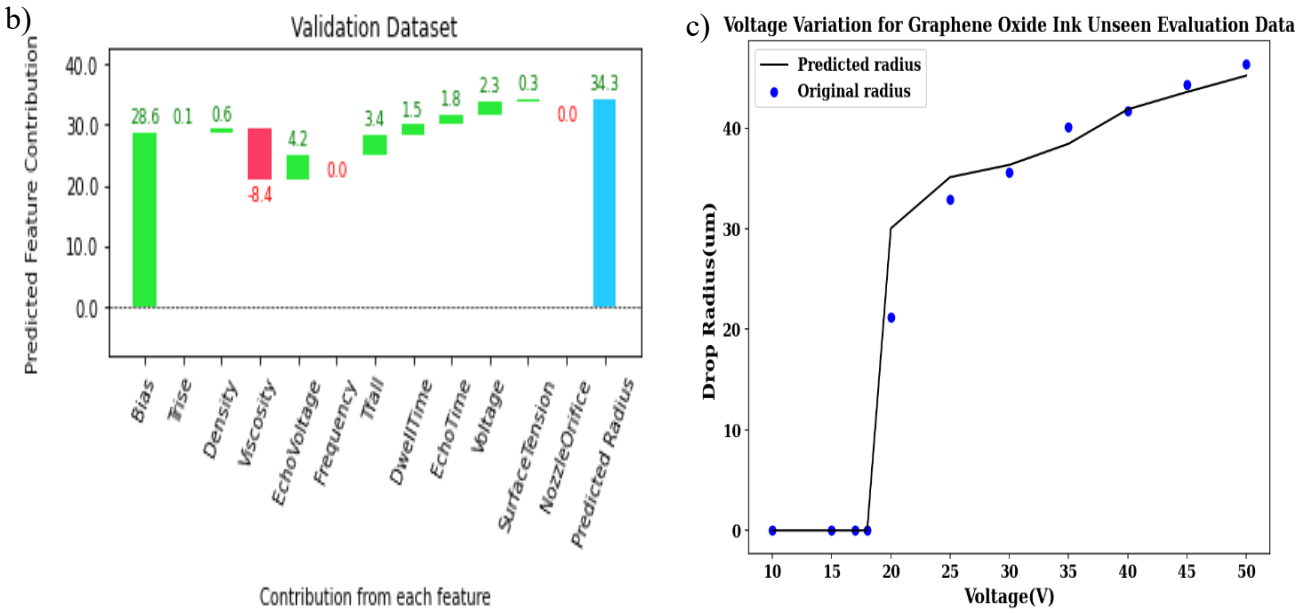


Figure 2-9: Drop radius prediction result. (a)Training and test RMSE for total and primary radius predictors. (b) Random forest tree interpretation for primary radius prediction. (c)Validating random forest drop radius model with untested GO ink data.

After the RF has made the radius prediction, we would like to know how the model came to this decision. RF is composed of a number of parallel decision trees, and it is quite impossible to comprehend the regression output by examining each tree. Each radius prediction is decomposed into contributions from each feature and bias given by equation(32). A sample primary drop radius prediction is shown in Figure 2-9: (b).

Predicted Total Drop Radius = Bias (Mean Total Drop radius of Training Dataset) + (32)
Summation of 11 feature contribution

Python package ‘tree interpreter’ is used from Github that adopts the same equation (32) for feature contribution explanation. For every test data point, the contribution of each feature is not fixed; rather, it changes according to the features while being traversed along each of the tree decision paths. How much each feature contributed to the total or primary radius summarizes that we need to tweak viscosity, echo voltage, fall time, and surface tension more than the other features to obtain a noticeable impact. In the case of a new dataset, this analysis reveals which features contribute most to any expected or unexpected behavior. This is particularly important to check the differences between two test data samples by comparing their bias (mean predictions) and corresponding average feature contributions. For a sample test data, the bias and each feature

contributions are broken down in Table 2-2. There is a clear difference between velocity and radius prediction in terms of feature contribution. For velocity, the important contributions are from voltage, echo voltage, pulse duration, and nozzle orifice for the test sample. RF primary radius prediction gives relatively higher importance to viscosity, echo voltage, fall time, and surface tension values. The radius prediction model was evaluated with untested GO ink data, displayed in Figure 2-9: (c). GO ink predicted radius maintains a linear relationship with the test voltage with a minimal residual error from the experimentally measured radius.

Table 2-2: Bias and contribution for radius prediction.

Features	**Value**	**Contribution**	**Decision Path**
Bias	28.6	0	28.6
Trise	3	0.07	28.6+0.07
Density	696	0.58	28.6+0.07+0.58
Viscosity	1.72	-8.41	28.6+0.07+0.58-8.41
Echo Voltage	-30	4.18	28.6+0.07+0.58-8.41+4.18
Frequency	1000	0	28.6+0.07+0.58-8.41+4.18+0
Tfall	3	3.44	28.6+0.07+0.58-8.41+4.18+0+3.44
Pulse Duration	15	1.53	28.6+0.07+0.58-8.41+4.18+0+3.44+1.53
Voltage	30	1.75	28.6+0.07+0.58-8.41+4.18+0+3.44+1.53+1.75
Surface Tension	42.8	2.27	28.6+0.07+0.58-8.41+4.18+0+3.44+1.53+1.75 +2.27
Nozzle Orifice	60	0.31	28.6+0.07+0.58-8.41+4.18+0+3.44+1.53+1.75+2.27+0.31
Echo Time	30	0	28.6+0.07+0.58-8.41+4.18+0+3.44+1.53+1.75+2.27+0.31+0.0
Primary Radius	**33.41 µm**	$\sum_{k=0}^{11}$	**Predicted Result = 34.31 micro-meter**

2.3.3. Jettability Prediction

Moving forward with the exploration of a stable jetting configuration, preprocessed lab collected data has been categorized into three classes-'No Ejection', 'Single Drop', and 'Multiple Drop.' Literature data does not have a clear indication of the jetting type, so they were not incorporated in building this model. A three-layer DNN (Deep Neural Network), a decision tree classifier and a K-nearest neighbor classifier are trained on 80% of the data and validated by the remaining untested data (20%). The classification accuracy of the test dataset of these models are reported in Table 2-3. DNN outperforms the rest data models with its higher test accuracy. The confusion matrix for the best classifier (Neural Network) is plotted in Figure 2-10(a) to demonstrate the classification performance parameter. The first type (Multiple Drop) is tested with 66 actual data points, and the model accurately predicts 63. However, 3 (2.42%) of them are incorrectly labeled as 'Single Drop'. The next group, 'No Ejection' consists of 18 traces, of which 2 (1.61%) are classified erroneously. Out of 40 'Single Drop' instances, the model forecast 1 (0.81%) as 'No ejection' and 4 (3.23%) as 'Multiple Drop'. This results in a total misclassification error of 8.06%. A classification report is considered to be the best way to monitor the categorizing performance of the final prediction algorithm. The report is shown in Table 2-3. Precision, recall, f1-score, and accuracy as in equations (33)-(36) are stated for each group concerning four outcomes: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

$$\text{Precision}=\frac{TP}{TP+FP} \tag{33}$$

$$\text{Recall}=\frac{TP}{TP+FN} \tag{34}$$

$$\text{F1-score}=\frac{2*Precision*Recall}{Precision+Recall} \tag{35}$$

$$\text{Accuracy}=\frac{TP+TN}{TP+TN+FP+FN} \tag{36}$$

Table 2-3: Comparison among several jetting classifiers.

Model Name	Accuracy
K Neighbors Classifier	78.23%
Decision Tree Classifier	83.87%
Neural Network	91.94%

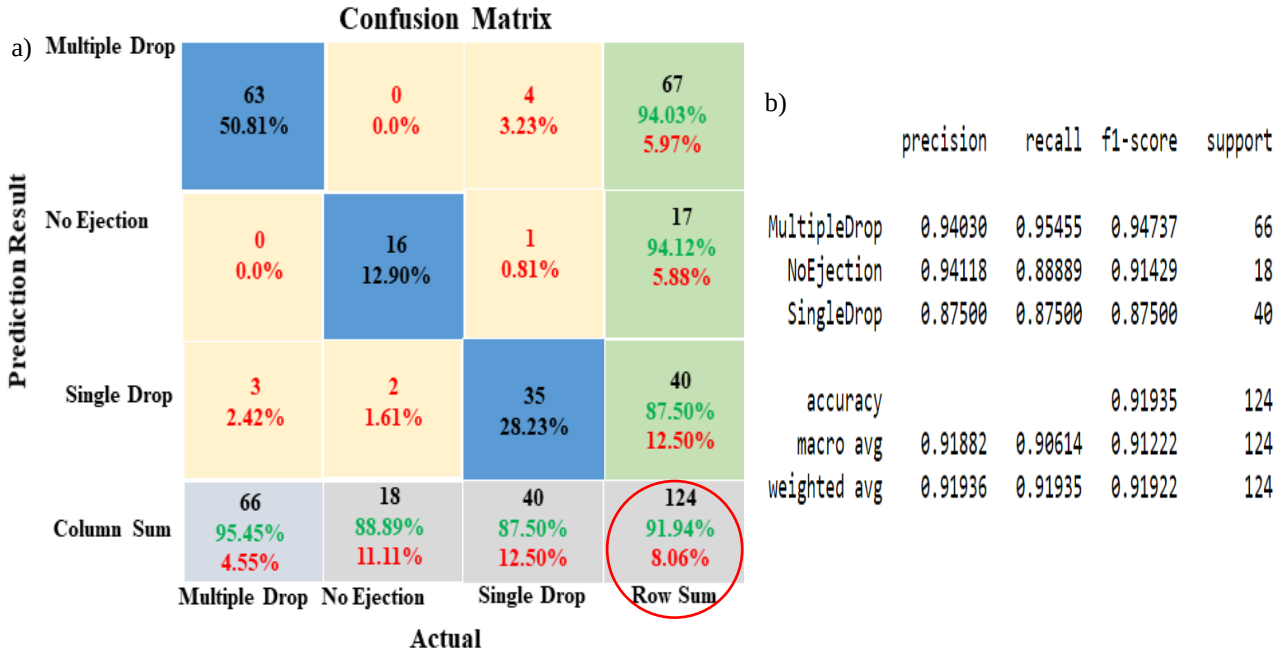
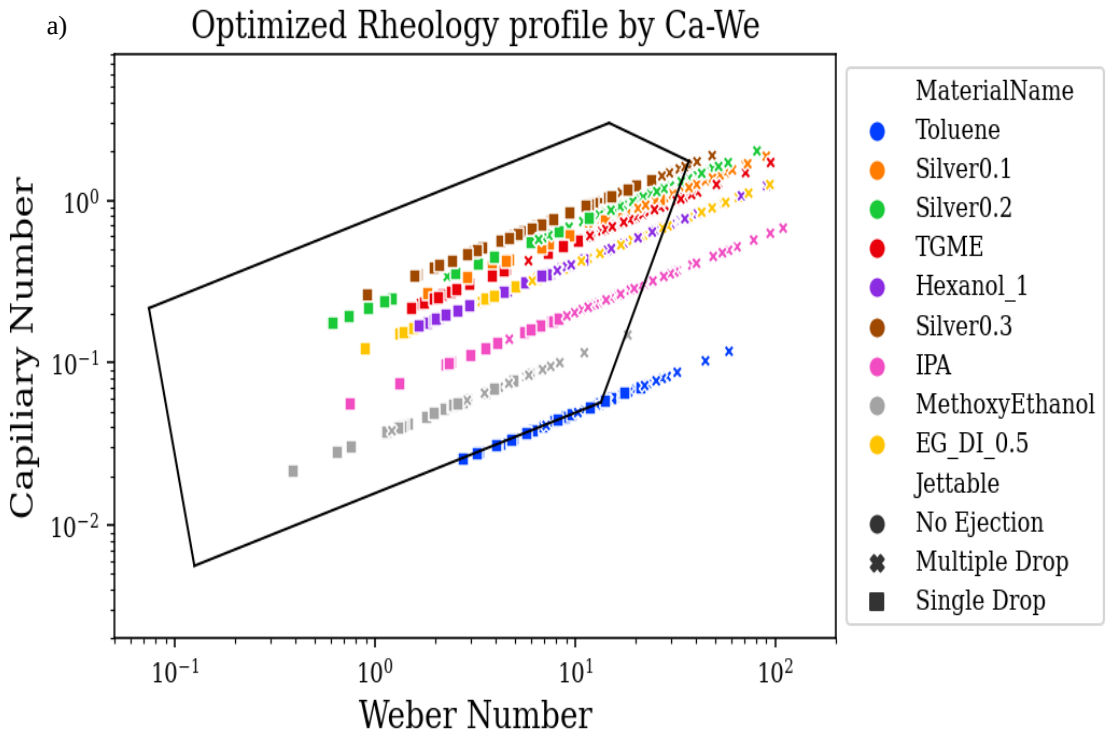


Figure 2-10: Jetting Classification Result for the neural network. (a) Jetting Classification Confusion Matrix. (b) Classification Report.

All of these metrics are measured with the Scikit-learn metrics package [142]. Among these three classes, 'Multiple Drop' prediction result is the best as it has the highest precision, recall, and f1-score. This means that it has higher TP and lower FP, FN. Therefore, the rate of mispredicting not multiple drops to multiple drops or multiple drops to not multiple drops is very low. F1 Score being the weighted average of precision, recall is higher because of the low FP, FN. From Figure 8(a), it is observed that among the 67 points in the first type 'Multiple Drop,' 63 elements are appropriately labeled, and that's why the recall is 63/66 = 95.455% in Figure 8b classification report. And the first row of the confusion matrix, the predicted class labels show 63 correct forecasts and four misinterpretations of 'Single Drop,' and it makes the precision 63/67= 94.03% in the Figure 8b report. 'No Ejection' category is the second-best correctly recognized positive instances. DNN predicts non-multiple jetting class into multiple jetting, with a higher FN value. The F1 score for this class gets lowered with the low recall value. 'Single Drop' has a lower precision, recall, and cuts down the F1 score with high FP, FN rate. It is because the DNN is misclassifying non-single drops to single drops and single drops to non-single drops. F1 is giving more insight than overall accuracy here.

All the output results are gathered to forecast a jetting window. In the inkjet printing literature, several types of windows have been proposed defined by pairs of non-dimensional numbers combining material parameters as well as drop velocity, mainly Reynolds number (Re), Weber number (We), Capillary number (Ca). Jettable conditions are enclosed in a window, as shown in Figure 2-11(a) for Ca-We from the literature [125]. Each material follows a straight line as expected. It can be observed that our lab data generally falls into the window as expected. However, there are some data points with multiple drops within the pentagon-shaped jetting window, which should only contain single drops. No ejection type with zero velocity can't be plotted because of the log scaling of the capillary and Weber axis, which both depend on velocity. The challenge with using such jettability windows is that data points can only be plotted retroactively after drop velocity was measured experimentally. With the predictive model proposed here, drop velocity, and jettable conditions can be predicted before conducting costly experiments, as shown in Figure 2-11(b).



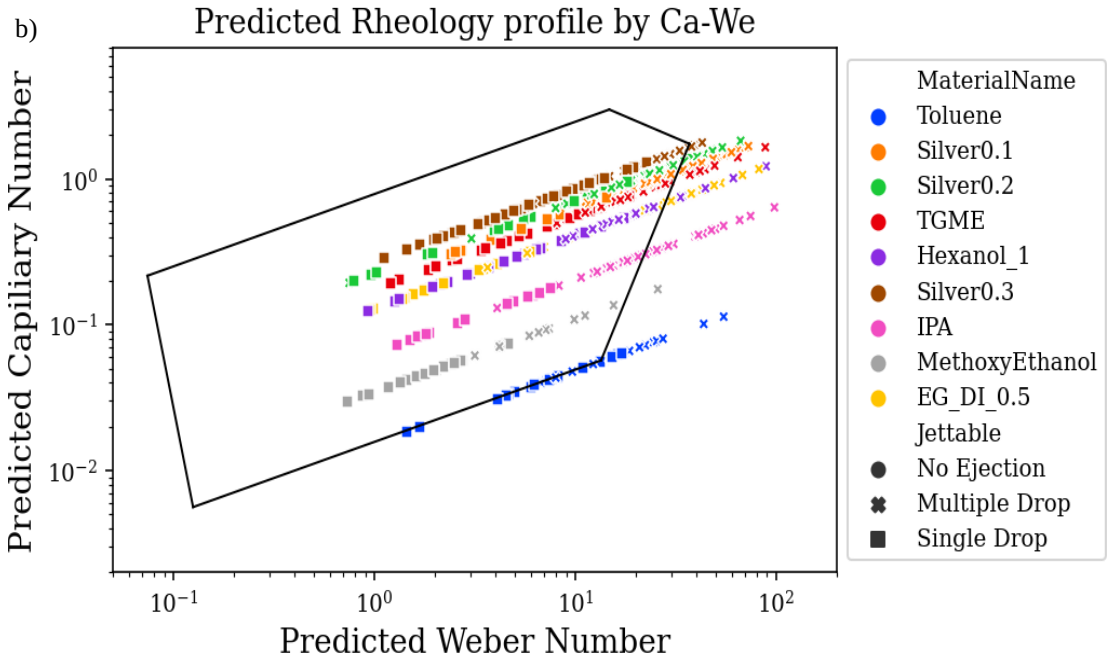


Figure 2-11: (a) Actual jetting window constructed with collected lab data over nine different material settings. (b) Predicted jetting window created with the calculated We and Ca using predicted drop velocity from the nelder-mead weighted averaging model over the same lab dataset. Linear boundaries are adapted with permission from (Nallan, H.C., Sadie, J.A., Kitsomboonloha, R., Volkman, S.K., and Subramanian, V., 2014. Systematic design of jettable nanoparticle-based inkjet inks: Rheology, acoustics, and jettability. Langmuir, 30(44), pp.13470-13477.) Copyright ©2014, American Chemical Society.

The predicted jettability agrees well with experimental results. A few no ejection cases (circular points) are plotted, which is a misclassification. This generally occurs for minimal velocity in the low We-Ca region. Multiple drop prediction generally occurs for large velocity values beyond the single drop regime. However, for toluene (on the lower boundary line) the single and the multiple drop prediction regions are mixed.

2.4. Conclusion

With a goal of investigating electrical signal and material intrinsic features' impact on DOD drop velocity, volume, and jetting type, several machine learning models have been deployed successfully. The theoretical jetting window is not very accurate for classifying the jetting type, and it also requires measurement of drop velocity experimentally. Our model can predict velocity

very closely, and so it is possible to omit that experimental part and jetting classification can be obtained from the signal and material features only. The predicted jetting window can be obtained beforehand to yield significant insights for the optimization of the pattering conditions and ink material designing. For example, a neural network model built on material and inkjet signal parameters can help classifying drop behavior as stable, satellite drop, drop breaking, no ejection, and so on with untested materials and signal values beforehand. We have applied ensembles of models to predict the drop velocity and total radius of 12 materials and observed root mean square error of 0.3163 m/s and 2.91um, which is very close to the experimental and literature value and predicted the jetting category with 91.94% accuracy. The models have been validated with an untested GO ink dataset. Collecting data with some more diverse material properties will help us infer the model output results and implement them at the industrial level. At this stage, we can predict the DOD droplet generation mode. This work can be implemented to continuous droplet generation methods like dripping, jetting tip streaming, and so on with enhanced features. Then we can switch our focus on other types of printing like electrohydrodynamic printing, aerosol jetting, and so on.

3. Chapter 3: Machine Vision Methodology for Inkjet-Printed Pattern Generation and Validation

3.1. Introduction

Inkjet printing technology for printed microelectronics suffers from a number of non-idealities due to unwanted ink flow on the substrate. This can be mitigated, and pattern fidelity can be optimized by using an optimized drop placement sequence in contrast to the standard raster-scanning approach. However, it is challenging to auto-generate such printing sequences for complex printed patterns. Here, the generation and evaluation of the printing sequence are turned into a computer-vision problem. Inkjet printing, as illustrated in Figure 2-1(a), allows patterns to be customized on-the-fly. The desired printed pattern is taken as an input image and converted into a printing sequence using contour, symmetric, and matrix sequencing, and corner compensation. After printing, pattern defects are detected by automated image processing to evaluate it against the designed ground truth image and to determine the best possible algorithm for printing sequence generation. The machine vision-based experimental approach identifies the best solutions for solving the printing and defect optimization problem in terms of precision, recall, and accuracy. We show here for industrial flexible electronics manufacturing facilities, vision-based defect routing yields high-resolution devices enabling any shape at any scale getting printed on the fly. The vision-based electronics printing workflow is shown as a flowchart in Figure 3-1. The first step in the pattern generation process is to design the desired patterns, i.e., the ground truth patterns using familiar drawing tools (Microsoft Paint, Adobe Illustrator, Electrical Computer-Aided Design (ECAD)) with specific print dimensions in the pixel scale. Each pixel in the image represents a drop from the printer. The drop spacing (DSP) is user-defined. DSP varies with the printed material (polymer, nanomaterial, conductor, semiconductor, insulator, etc.) and substrate on which the material is printed (glass, polyimide (PI), polyethylene naphthalate (PEN), polyethylene terephthalate (PET), etc.). The generated images are saved as a black and white digital image for the next sequencing step. The goal is to determine the order in which the printer prints the drops (i.e., pixels) in the pattern. As shown in Figure 3-1, in contrast to traditional raster scanning, four different vector sequencing models are implemented using computer vision processes. The final output of the algorithm is relative coordinates between subsequent drops, which defines the drop order or sequence for the practical printing on the substrate. The success is evaluated during and after printing, on and offline, using image segmentation, and the number of

printing defects is quantified using precision, recall, and accuracy against the designed ground-truth pattern. This evaluation scheme could be employed in manufacturing to determine if a print has acceptable quality or if there are defects. Depending on the type of shape (filled, non-filled), scale (10X10 pixel, 12000X12000 pixel), and the used substrate and inks, an acceptable inaccuracy threshold can be set after initial printing experiments and depending on the application.

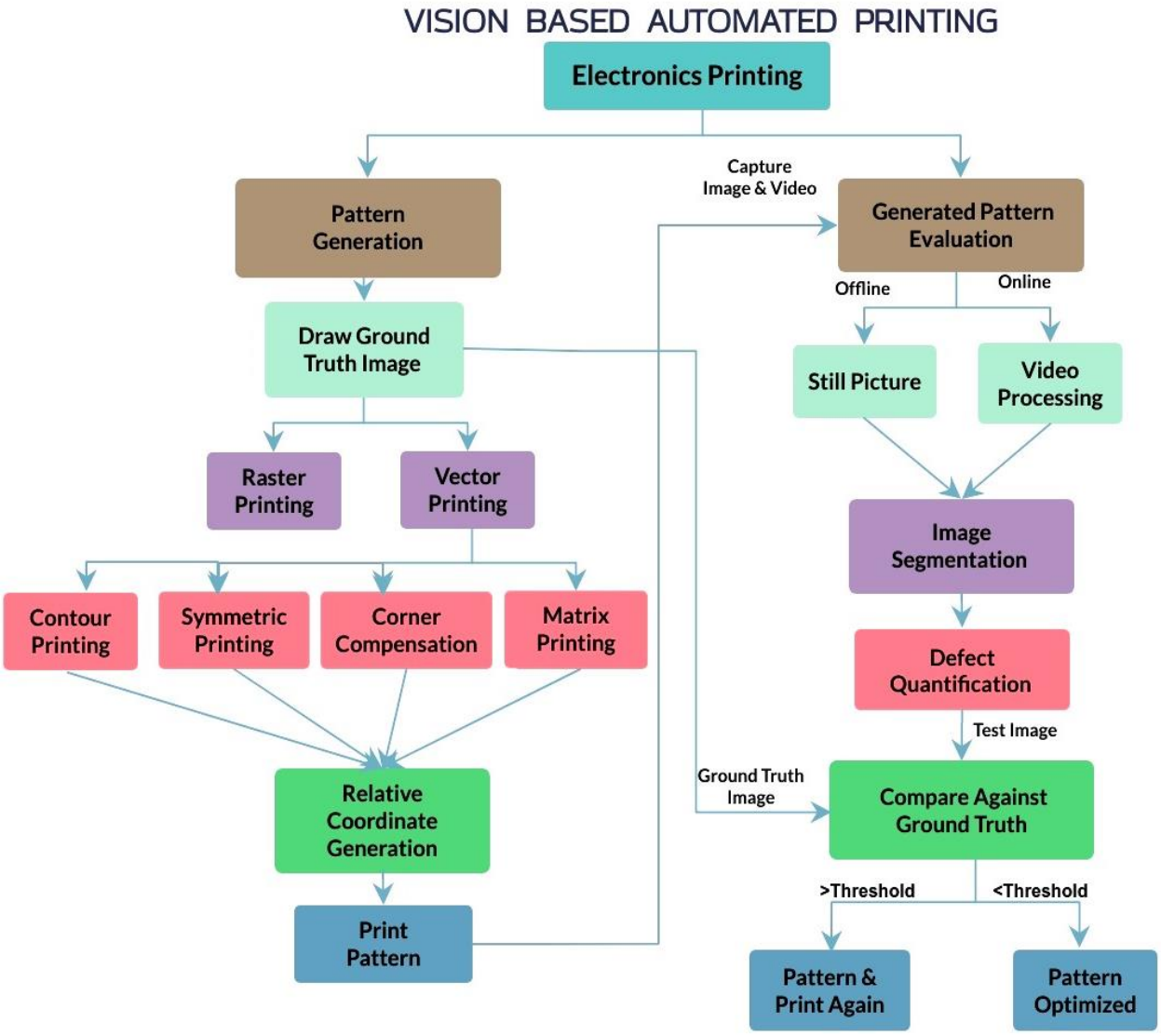


Figure 3-1: Workflow for automated electronics printing using machine vision. Drop sequences are generated with different vector printing methodologies and used for printing. Different methodologies are evaluated by processing images of the printed patterns.

3.2. Methods

3.2.1. Detection of Important Input Features

From the fluid mechanics perspective during printing, the most essential features of a printed shape are corners, orientation relative to the print direction, edges, and contours [157][158][159]. Pattern optimization strategies are explored here to reduce printing defects by engineering the printing sequence based on these features. The features need to be identified from the drawn ground truth image in terms of the best image processing descriptors. The truth image may be binary, colored, or grayscale and need to be processed through a standard pre-processing pipeline, as presented in Figure 3-2. Each processed layer is converted to a one channel grayscale image. This step ensures that different image formats (single or multilayer, RGB or grayscale) become appropriate for the next common processing steps. The binary image is then used as the input for raster or vector sequencing.

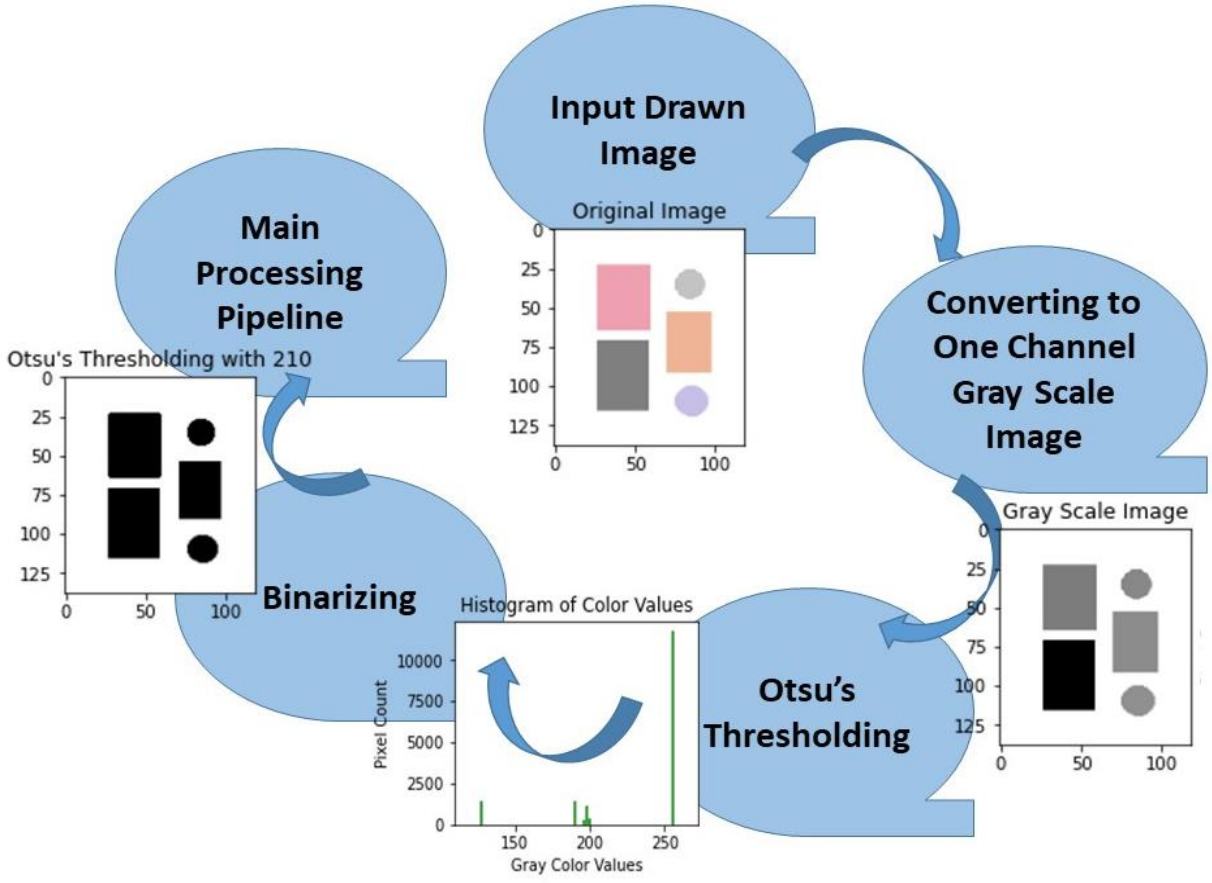


Figure 3-2: Common pre-processing pipeline.

Neighborhood-based operations are performed on each input pixel and its surrounding area to calculate and combine edge, contour, and corner features during vector sequencing. This includes different filtering operations with specific kernel sizes and weights. Edges are continuous or discontinuous pixels defining the physical extent of an object representing the maxima of intensity gradient obtained from edge detection techniques. Contours are a fit through the adjacent pixels of the detected edge map to locate the coordinates of the meaningful boundary of an object [158]. The performance of contour detection is solely dependent on the previously detected edge map. For a discontinuous edge map, detected contours will be interrupted as well. The perfect edge detector was found after running several of them on test patterns to ensure that the detected contour is perfect with continuous x,y coordinates. Patterns exhibit better printing results when the outline is printed before printing the drops inside. The contour creates a boundary for the internal fluid flow. Printing it first makes the edges smoother. The background is always white and the foreground shape to be printed is black. A detector pipeline has been established to execute each of the distinct algorithms. Several detectors have been implemented to identify the best-suited edge detector, as presented in Figure 3-3. For filled shapes (lines or rectangles that are more than one drop wide) and line patterns (lines that are one drop wide), the findings are meaningfully different. Many edge detectors have been developed in the past, some based on the first derivative of the image (Sobel, Canny), some on the second derivative (Laplacian of Gaussian (LOG)) [157][158]. We studied filled and line patterns that are small-scale (only a few drops) and large-scale. The Canny edge detector performs poorly for all pattern types, as shown in Figure 3-3(a-c)(ii). Following calculation of the derivative, it detects edges by applying non-maximum suppression (discard pixels with a gradient less than its neighbors), thresholding, and filtering out weak edges that are not related to strong edges by hysteresis. For improved Canny edge outputs, the thresholding requires intense optimization. For all of the three pattern types, this results in irregular edges, and the intersections are lost. The Canny edge map loses corners, junctions. For the large-scale spiral-shaped case, it misses numerous points on the edge map. Sobel edge detection uses the first derivative of the image with linear filtering. It combines horizontal and vertical filtering to locate the edges at the maximum and minimum values of the first derivative [157]. It behaves more like Canny, except it doesn't apply non-maximum suppression to differentiate between thick and thin edges. The obtained edge map is not reduced to a single pixel boundary and not capable of identifying a distinct contour. Instead, it results in multiple segregated contours, as shown in

Figure 3-3 (a-c)(iii). LOG detects edges from zero crossings of the second derivative of the image as in equation (37). Figure 3-3(a-c)(iv) shows it localizes edges, including corners well for all image scales, both filled and line patterns. LOG performs better than Canny and Sobel to recognize horizontal, vertical, and rotated edges.

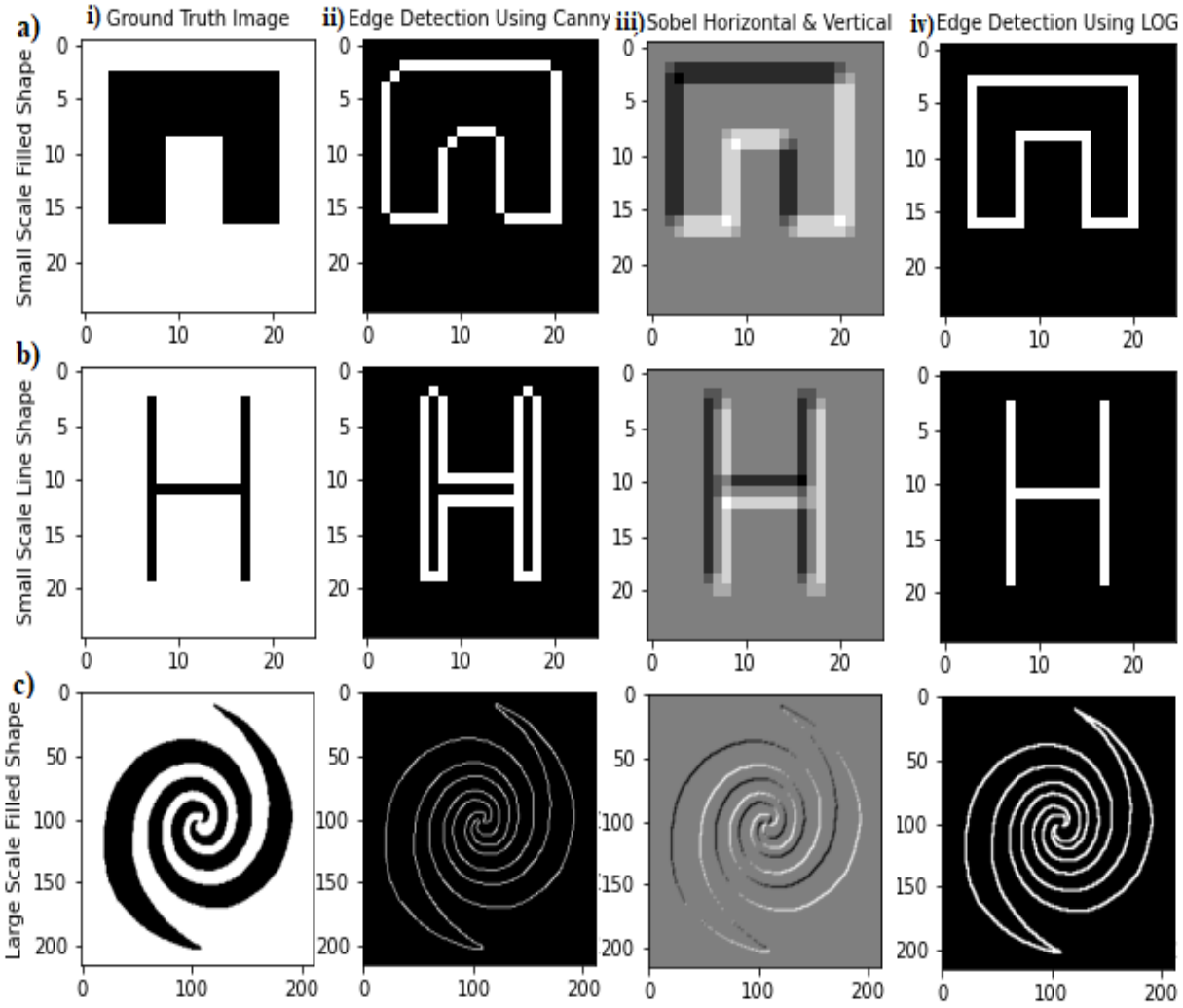


Figure 3-3: As a basis for contour fitting, edges detection of a variety of possible patterns a) small-scale filled shape, b) small-scale single-line pattern, c) large-scale filled structure, employing several edge detectors ii) Canny, iii) Sobel, iii) Laplace of Gaussian.

Therefore, LOG is chosen for contour fitting going forward. To eliminate LOG's inclination to noise, the image is smoothed with a Gaussian filter (equation (38)) before applying the LOG. Blurring removes noise from the image and creates a defined transition from low to high

pixels around the edges. Then, these edges are efficiently recognized by the Laplacian of Gaussian operation.

$$LoG(x,y) = \frac{1}{\pi\sigma^4}[1 - \frac{x^2+y^2}{2\sigma^2}]e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (37)$$

$$G(x,y) = \frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (38)$$

Next, for picking up corner information from the whole pattern, Harris corner detection is found to work best. Two widely used corner detection schemes, contour-based and intensity-based, have proven consistent for detecting corner coordinates precisely. Our application requires reliable corner detectors with low parameter dependency and without a priori knowing the exact number of corners present in the ground truth image. Despite having the contour information from our feature detection steps, corner extraction from the contour fitting is not ideal as it requires knowledge of the exact number of candidate corners along with prior knowledge of contour type (closed or open) to localize them correctly. The intensity-based Harris corner detector does not require knowledge of the number of corners, contour pattern, and is also computationally much faster. It is independent of local features and noise, and it can be improved by sub-corner grouping and thresholding. All of the sequence generation algorithms are developed based on these feature information set.

3.2.2. Drop Sequence Generation

Distinct vision algorithms are developed for each unique drop sequencing method with recognized and extracted features from the ground truth image. The designed pattern is represented in the form of a pixel matrix where every pixel represents the location of a drop. For generating a printable drop sequence, the original pixel representation is manipulated using point and neighborhood operations. The drop sequence is a set of relative coordinates (x; y) between successive drops. The most conventional technique for inkjet printing is raster-scan-based printing, where the pattern is printed line by line (see Figure 3-4 (b)); however, it can lead to defects for printing in different orientations (rotated patterns at the micrometer scale). The raster approach to electronics printing does not work well for intricate patterns with rapid edge and corner transitions. Raster sequencing is a point transformation as the output pixel coordinate sequence is determined solely as a function of the input pixel value (0 or 1) corresponding to that location. Conversely,

vector sequencing incorporates both point and neighborhood operations for identifying edges, corners, and contours feature vectors to generate sequence output, as shown in Figure 3-4(c-f).

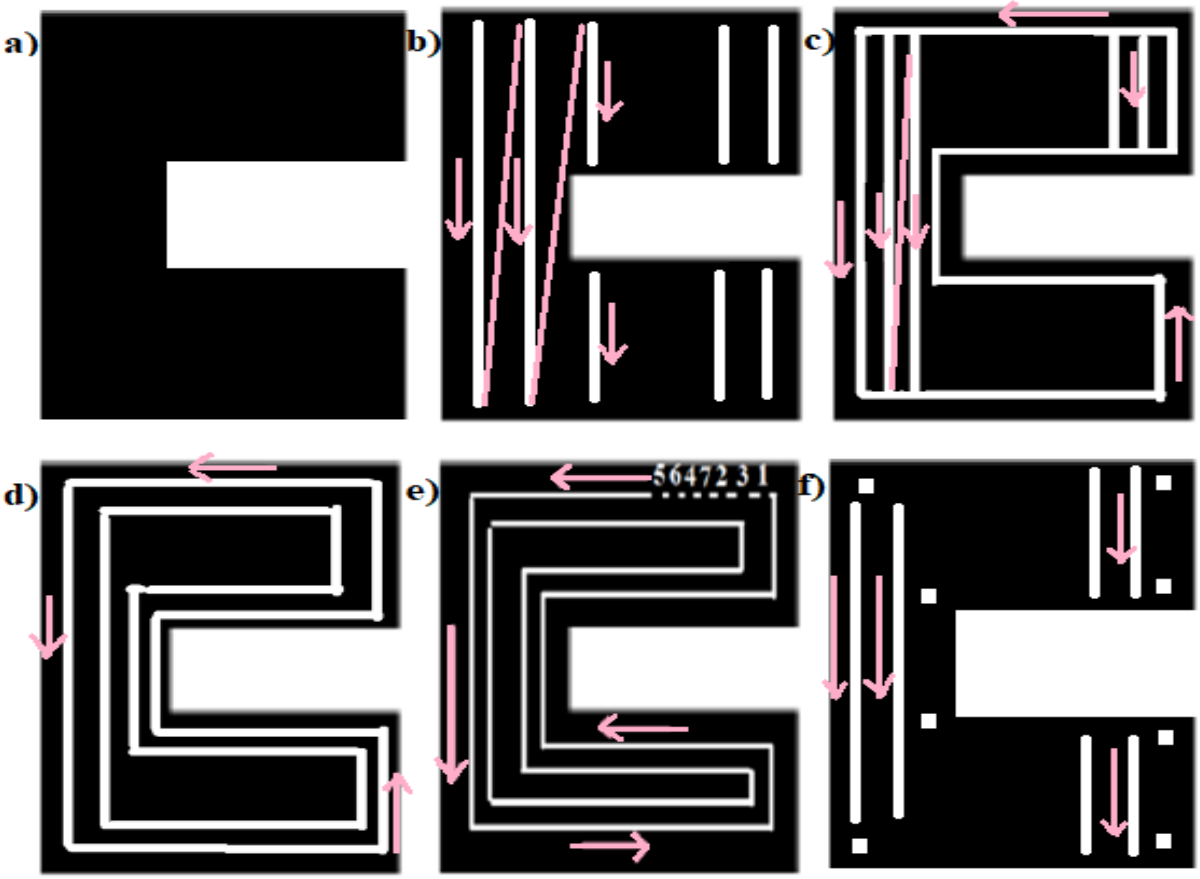


Figure 3-4: a) Input image that is printed with different drop sequencing methods: b) Raster pixel sequencing. The printer always prints one entire column before moving on to the next column. c) Parallel filling vectorization. The printer first prints a boundary and then fills drop inside with a raster pattern. d) Radial filling vectorization. The printer first prints an edge and then fills in the center in a spiral path. e) Vectorising using radial filling but with symmetric drop sequence. f) Corner compensation deleting corner pixels and filling with raster sequencing.

3.2.2.1.Parallel Filling Vector Sequence Generation

With the parallel filling method, the border of the pattern is printed first before filling the inside to reduce bulging and irregularities in the outline of the printed pattern. The pattern outline (in computer vision terms contour) is significant since it defines the shape and encloses the

boundary for the ink fluid flow. It exploits contact line pinning as the outline starts to dry before the inside of the pattern. The pre-processed image edges need to be detected as the first step to implement boundary filling. As described in Figure 3-3, LOG captures the complete edge information well and is used as a preliminary step for both parallel and radial contour filling vector sequencing. The detected edges are fitted to form complete contours. Contour coordinates are extracted in a clockwise sequence and saved in a text file, as shown in Figure 3-5 (a-e). The border contour is detached from the original image, and the remaining pixel coordinates of the subtracted image are collected with the raster conversion method. Finally, the boundary and inside coordinates are merged and converted to relative coordinates and stored as a text-based command file for the printer.

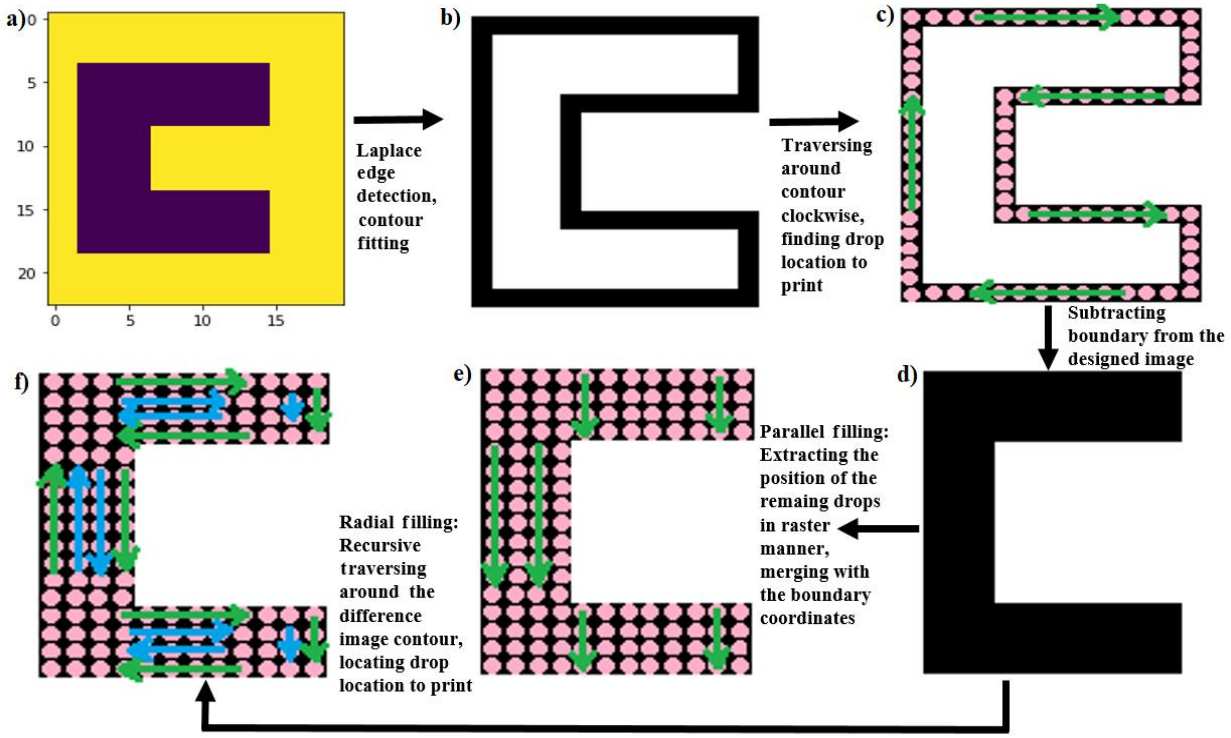


Figure 3-5: Parallel and radial filling pixel sequencing steps. a) Designed pattern. b) Detecting edge and fitting contour through the border pixels. c) Extracting the boundary pixel coordinates (x,y), collecting them clockwise, and calculating the relative coordinates using drop spacing. d) Subtracting the edge pixels from the ground truth image to get the difference image. e) For parallel vectorization, the difference image pixel coordinates are extracted through column-major order

based raster sequencing. f) For radial vectorization, border pixel coordinates of the difference image are removed recursively and merged together to generate the final relative coordinates.

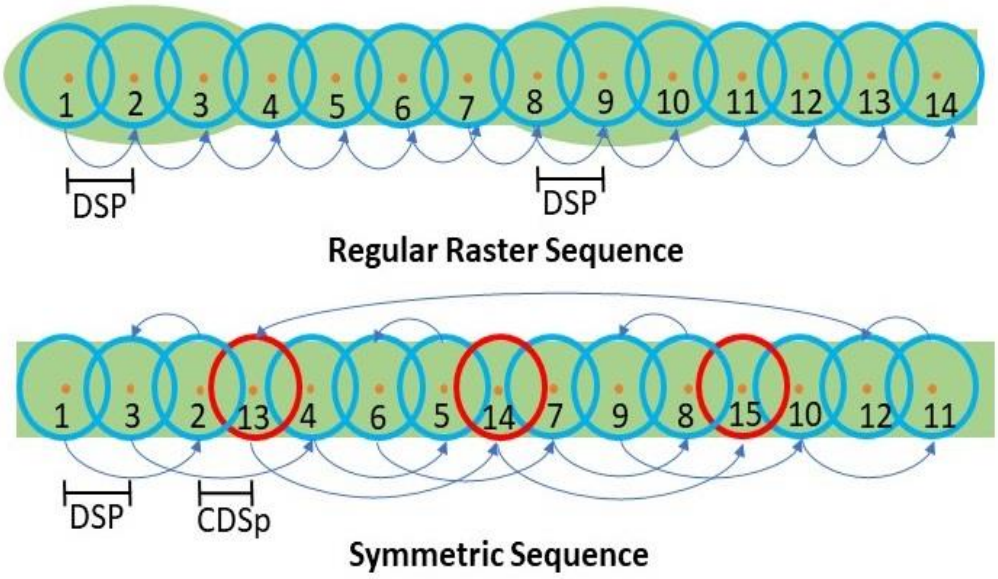
3.2.2.2. Radial or Spiral Filling Vector Sequence Generation

Radial filling sequencing undergoes the same pre-processing, edge localization, contour finding, and border subtracting as parallel sequencing. The main difference between the algorithms is that the border contour is detected and removed continuously from the original image in a loop until a blank image is formed. Each time, contour coordinates are merged with the previous ones (see Figure 3-5 (a-d,f)). In the end, relative coordinates are calculated and saved as a printer intake file. For filled shapes (i.e., more than one drop wide patterns), radial and parallel sequencing result in different printing sequences, and the print result can be dissimilar; however, non-filled forms (i.e., one drop-thick line) don't have multiple contours, so the generated sequence is the same for both algorithms.

3.2.2.3. Symmetric Vector Sequence Generation

Inkjet-printed patterns often bulge at the beginning of line and intersections between lines. This arises from the Laplace pressure difference between the already printed track and the newly added droplets. Consequently, linewidth can be very dissimilar in those regions. Bulging can be avoided with a segmented and symmetric printing methodology. In conventional raster printing, drops are positioned one after another along the pattern line without controlling for pressure imbalances, which can distort the printed pattern with irregular bulging. The imbalance can be considerably improved by fragmenting single-drop wide line patterns into three-drop long segments. At the start, the outer two drops of each segment are printed, then the central drop so that it does not experience a pressure gradient due to the symmetry of the segment. Subsequently, the three-drop segments are joined with a connecting drop. The pattern segments get linked, maintaining pressure equilibrium on either side of the pattern.

a)



b)

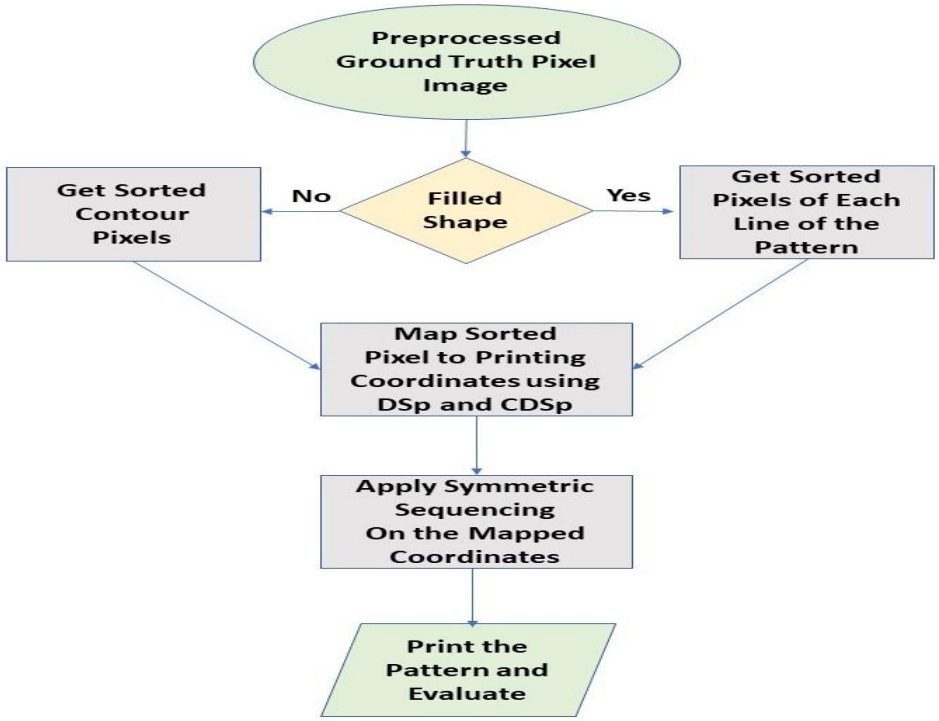


Figure 3-6: a) The top row shows bulging at regular intervals when a line of fourteen drops is printed with a traditional raster printing. The bottom row displays symmetric sequencing of fifteen drops with three segments and three connecting drops. b) Symmetric pixel sequencing algorithm.

Figure 3-6(a) displays raster and symmetric drop ordering. Raster follows constant drop spacing (DSP) in all directions, while symmetric has two distinct parameters (DSP and connecting drop spacing (CDSP)) and a different droplet print order, as shown in Figure 7(a). DSP is used within each segment of three drops, and CDSP is used between two consecutive segments. The second drop is two DSP away from the first drop, and the third drop is placed in the middle between the first two. Then, the next segment starts with the fourth drop placed at (DSP+2CDSP) distance relative to the center of the first segment (the third drop), leaving 2CDSP vacant space between two segments for a connecting drop to be filled after all the segments have been printed. The connecting drops, controls the fluid flow and evaporation rate between segments. Previous work has experimentally shown that CDSP should be smaller than DSP, and the optimum CDSP varies between 0.6 and 0.95 times DSP for different substrate and ink combinations [78].

The entire symmetric printing process is implemented using a vision pipeline shown in Figure 3-6(b). After pre-processing through the regular pipeline of Figure 3-2, the pattern to be printed is passed through the symmetric sequencing algorithm. The symmetric pixel sequence implementation is different for filled and non-filled cases. A limitation of symmetric printing is that pattern dimensions are constraint to multiples of segments of three drops plus connecting drops. Non-filled issues that have a closed contour, pixel scale design can be critical, as shown in Figure 3-7(b). If the number of segments doesn't match with the multiples of 3n in one direction (x) and 3n+1 in the other direction (y), then the design outcome may lead to irregular shapes like in Figure 3-7(b)(i). Crossed drops are connecting drops with two CDSP relative motion with the previous and next drop. In Figure 3-7(b), a clockwise closed contour pattern printing starts in the down arrowed y-direction with three CDSp relative movements. Then in clockwise x-direction, it travels three CDSP and in up arrowed y-direction four CDSP. The two y-direction travel doesn't face the same number of CDSP movement, and the endpoint doesn't meet with the starting point. Figure 3-7(b)(ii) pattern starting and ending point coincides with the 3n and 3n+1 limitations maintained on y and x-direction.

A limitation of symmetric printing is that pattern dimensions are constrained to multiples of segments of three drops plus connecting drops. Filled forms are composed of multiple neighboring lines in the same or alternate directions. Each line in the same direction is extracted separately and passed through the symmetric sequencing steps. The boundary contour of non-filled

patterns is obtained using the parallel vectorizing sequence shown in Figure 3-5. The whole contour is then processed with the symmetric pixel vectorization routine. A network directed graph is constructed with the generated coordinates as node data, and the distance between two consecutive pixels is the edge attribute for the filled and unfilled patterns. This graph describes the printing scheme with the nodes as the center of each drop with dimensions and drops ordering.

The schema in Figure 3-6(b) splits the symmetric vectorization problem into two phases. The constant DSP pixel map is transformed into DSP and CDSP mapping, as shown in Figure 3-7(c)(ii), Figure 3-7(d)(ii). The pixel space is mapped to micrometer space with the appropriate DSP and CDSP, considering every fourth pixel of the pattern array (Filled pattern: Raster Sequence, Non-filled pattern: Contour Sequence) as a connecting drop. The distances of the connecting drops from the previous and next pixels are changed to CDSP, and the rest are set to DSP. CDSP is smaller than DSP; therefore, there is a shifting in the mapped pixel coordinates shown in Figure 3-7(c)(ii) and Figure 3-7(d)(ii). Figure 3-7(c)(i) is a network diagram for filled shape raster sequencing. This should be taken into account when designing pattern dimensions for symmetric and segmented printing. After mapping, symmetric pixel ordering (1,3,2) is implemented, as shown in Figure 3-7(c)(iii) and Figure 3-7(d)(iii). The procedure retains the first-pixel coordinate at the same location. The previous coordinate of the third pixel receives the second drop. The third drop is printed in between at the previous coordinate of the second drop. The entire pattern coordinates are rearranged with this repeated sequence array internally with a weight tracking column. The column has a repetition of the sequence array [1, 4, 3, 0]. Every multiple of the 4th index values having value 0 are treated as the CDSP, and they are reassigned values starting from the total cumulative sum of that column. The column has value [1, 4, 3, 8]. Each set of pixel coordinates is sorted in descending order according to this weight column, and the new index is now 1,3,2,4. The pixel index having lower weight gets printed first.

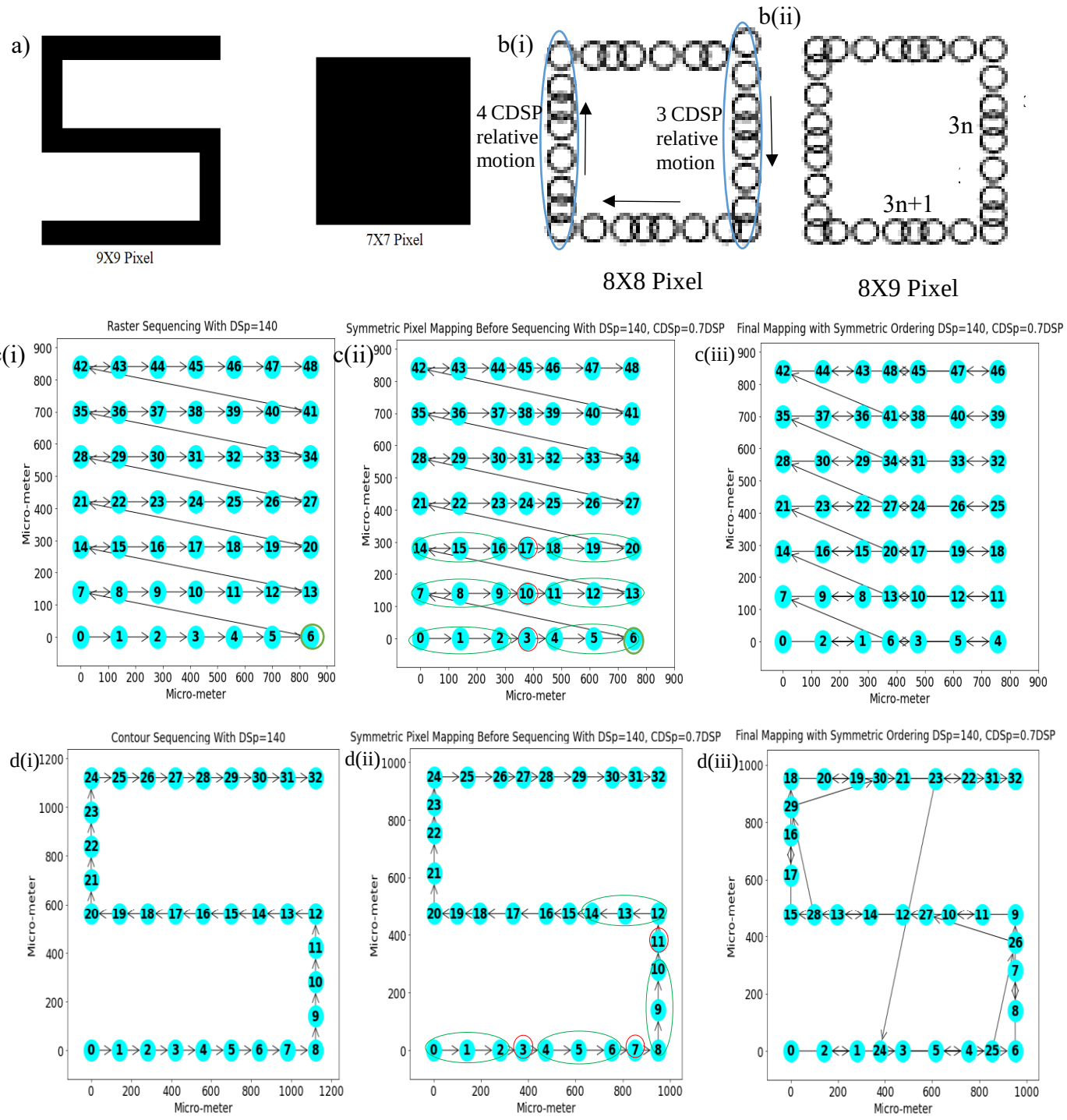


Figure 3-7: Symmetric printing methodology with network graph plot. a) Ground truth image of the filled and non-filled pattern. b) Design constraints for connected contour-based non-filled shapes. c) For filled configuration, pixel mapping with distance transformation and symmetric ordering with DSP=140 µm and CDSP=0.7*DSP. d) For non-filled shapes, distance transformation, final pixel ordering in the micron scale with the same DSP and CDSP.

3.2.2.4. Corner Compensation

For filled shapes, the patterns can bulge in the corner, even though they are patterned through vectorization. A potential solution to this problem is the deletion of some pixels from the corner region before applying raster or vector sequencing. A vision strategy is adopted to determine how many pixels should be deleted and from where. Initially, the original pattern is loaded, pre-processed, and then filtered through a bilateral filter to blur the image, followed by the Harris corner detection procedure. Subsequent thresholding and tuning are applied to the detected corner positions to refine and get the exact coordinates. When the designed pattern dimension becomes small (approximately <15X15 pixel), the number of detected corners might be more than the real number and not in the precise location. For such cases, the corner detection efficiency is further improved with an additional distance measurement step to remove the extra corners within a threshold distance from each other. It gives a reasonably accurate location of the corners. Finally, the detected corner pixels are removed from the design using a shift operation. The optimal shift depends on the input pattern. Vector and raster sequence is subsequently implemented using the algorithms described in section 3.2.2.1 and 3.2.2.2 to generate the relative coordinates from this compensated array for printing.

Figure 3-8: Corner compensation demonstration before applying sequence. Different schemes are shown with varying numbers of drops removed at different positions.

3.2.2.5. Matrix Vectorization

Two different matrix vectorization deposition programs have been developed - one vectorizes patterns block by block, another one is vectorizing each drop of matrix blocks

sequentially. Each matrix block dimension is defined as the number of droplets in the X and Y direction, such as 4x4 or 2x2. Optimum drop spacing and block size have the potential to generate patterns with better resolution. Two different dot sequencing has been adopted for further study, within a single block and between multiple blocks. Promising results have been observed in the case of an entire block printing once at a time, in contrast, to drop level printing between blocks referred to as ‘‘multi-level’’ matrix by Tekin et al.[187]. Figure 3-9 (a)(ii) and Figure 3-9 (b)(ii) show schematic illustrations of the 2x2 and 4x4 matrix vectorization print processes, respectively, at the block level. Considering the first printing starts at the lower-left corner and ends at the upper right, the first 2x2 matrix block, “Block1”, is printed with all four drops. Once “Block-1” is finished, the stage is moved to “Block-2.” The four drops are deposited in raster sequence from low-X, Y coordinates to high X, Y coordinates. The deposition is continued until the last of the 2x2 blocks “Block12”.

In case of filling up the matrix blocks drop by drop, as shown in Figure 3-9 (a)(iii) and (b)(iii), the first drops of the blocks are printed with sequence 1,2,3,.., 12, then the second drops start with the order 13,14,15,16,…,24. This continues until all the drops in Block1 to Block12 goes through sequencing. Block-level printing, as in Figure 3-9(a)(ii) and (b)(ii), exhibits improved patterns in the captured CCD image contrary to printed dots in between blocks as in Figure 3-9(a)(iii) and (b)(iii).

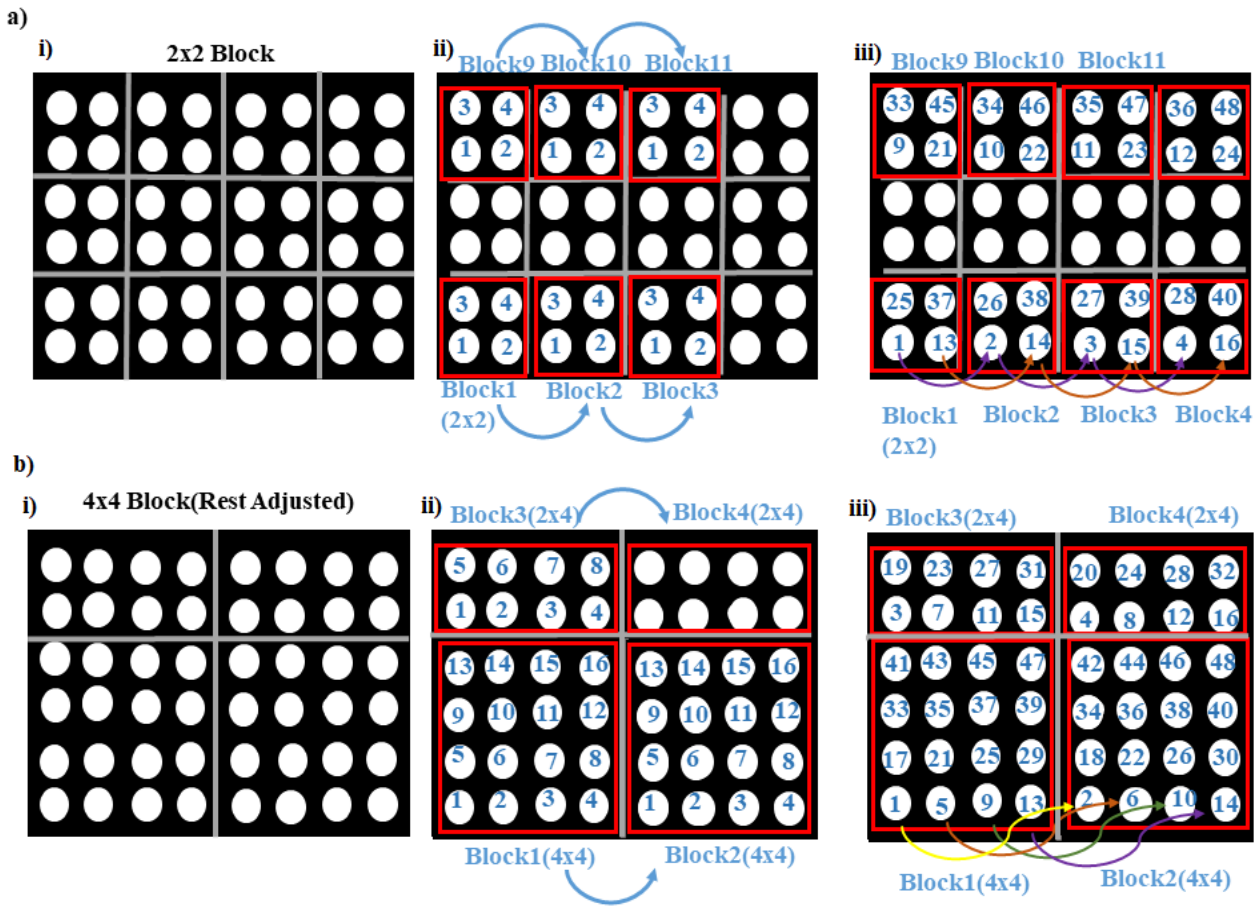


Figure 3-9: Schematic illustration of matrix vectorization block by block and dot by dot. (a) (i) A rectangle pattern of 6 drops (x-direction) x 8 drops (y-direction) is sequenced with twelve 2x2 matrix blocks. (ii) Block by block matrix level drop deposition. The sequence of each drop is denoted with a number written in blue on it. Once all the drops in the same block are deposited, the next block goes through the sequencing until all the blocks are covered. (iii) Schematic illustration of matrix vectorization dot by dot in between blocks. Each drop's order is denoted with a number written in blue, and red lines surround the matrix blocks. Each drop at the same location of each block is printed sequentially before moving to the next locations. (b)(i) A rectangle pattern of 6 drops (x-direction) x 8 drops (y-direction) is sequenced with two 4x4 and two 2x4 matrix blocks. (ii) Matrix level block by block (4x4) drop deposition. (iii) Matrix vectorization dot by dot in between 4x4 blocks.

To extend the promising result, the border pixels of the designed pattern has been extracted, and the contour is fitted on the boundary. The pattern contour border is added to the print file, and

then the rest is sequenced with a matrix-vector before attaching at the end. While printing, the file bounds the pattern edge structure first, and then the inside drops printed as matrix helps improve pattern structure. The distance between the drops inside the matrix and between the matrixes is kept the same and termed as drop spacing (DSP). Contour printing incorporated with block-level matrix printing considerably enhances edge as well as overall pattern homogeneity. The two processes are explained in Figure 3-10 and Figure 3-11.

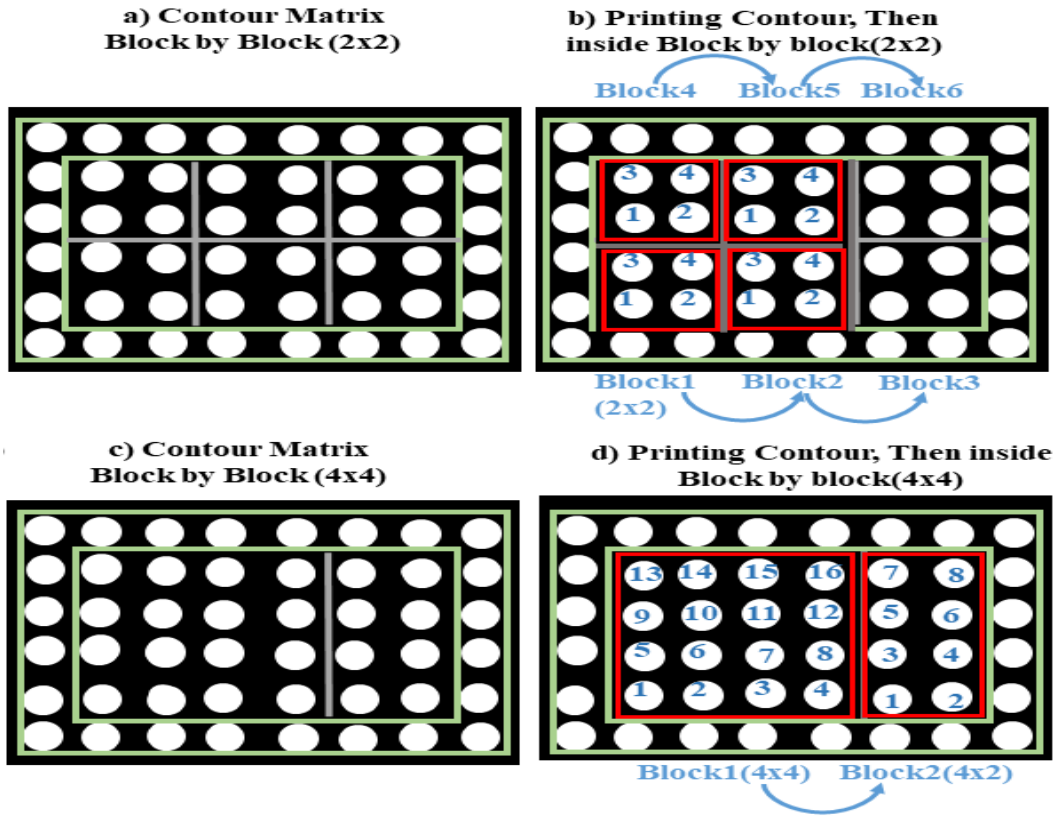


Figure 3-10: Schematic diagram of contour matrix printing (matrix vectorization is blocked by block). The order of each drop is denoted with a number written in blue on it. The contour drops are bounded with green rectangles, while the matrix blocks are shown with red lines. Once all the drops in the same block are deposited, the next block goes through the sequencing until all the blocks are covered. (a)(b) The contour of a rectangle pattern is extracted, and then the inside rectangle of 4 drops (x-direction) x 6 drops (y-direction) is sequenced with six 2x2 matrix blocks. (c)(d) Contour matrix pattern with one 4x4 and one 4x2 matrix block.

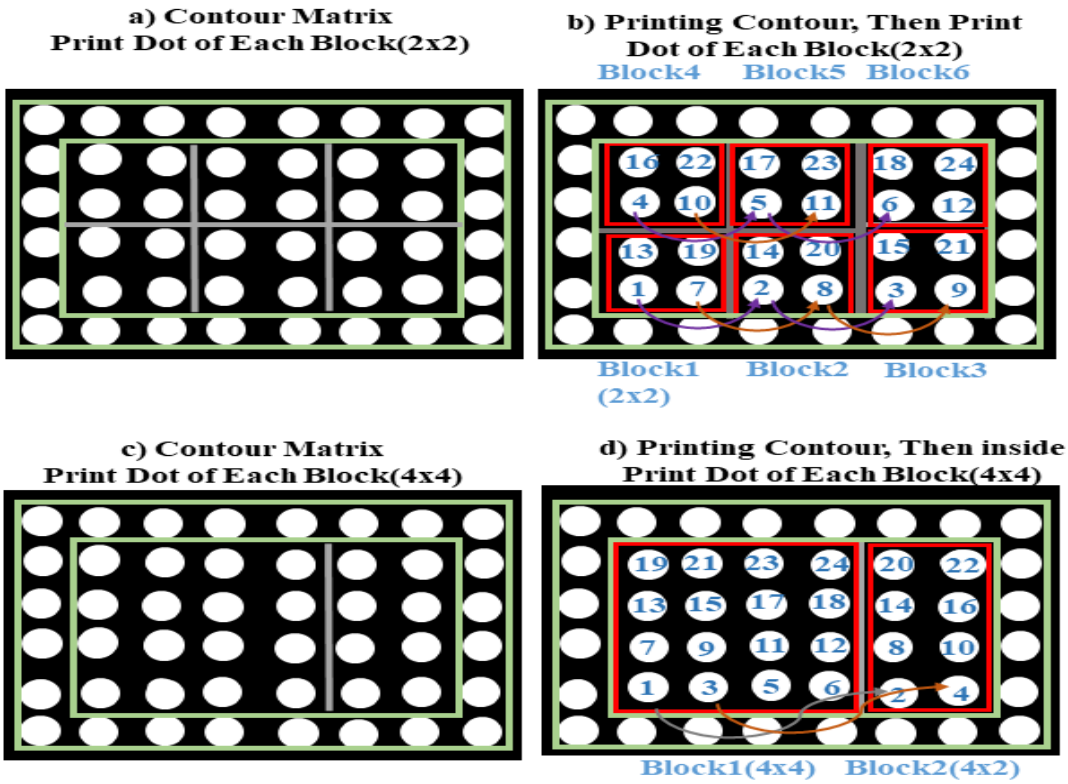


Figure 3-11: Contour Matrix by each block dot sequentially. After the contour has been separated, the rest are sequenced through printing each drop of each matrix block before moving to the next one. (a)(b) The contour of a rectangle pattern is extracted, and then the inside rectangle of 4 drops (x-direction) x 6 drops (y-direction) is sequenced with six 2x2 matrix blocks. (c)(d) Contour matrix pattern with one 4x4 and one 4x2 blocks.

3.2.3. Pattern Segmentation for Improved Quality

Many partitioning algorithms have been developed for geometry partitioning with rectangles [188][189], triangles[190]. These partitioning techniques can easily segment the image into the desired smaller repeatable blocks. This method has already being used by the VLSI layout GDS file for data compression. For printed electronics, this approach is entirely new. As we have experimented with shapes, we have observed promising as well as simplified printing outcomes in the case of rectangular shapes. Rectangular decomposition aids the printing decision process. For large scale printing, this can be a significant advancement. Some ink might contain evaporative materials and dry faster. In that case, printing smaller segments and connect in between immediately is a better option. Again it can restrict materials in segments and resolve unwanted

fluid flow during or after printing. With all these advantages, we resorted to partition pattern before moving forward to vectorization. We attempted two types of portioning: by maximum area rectangle and partition pattern clockwise/ anticlockwise, as shown in Figure 3-12(b),(c).

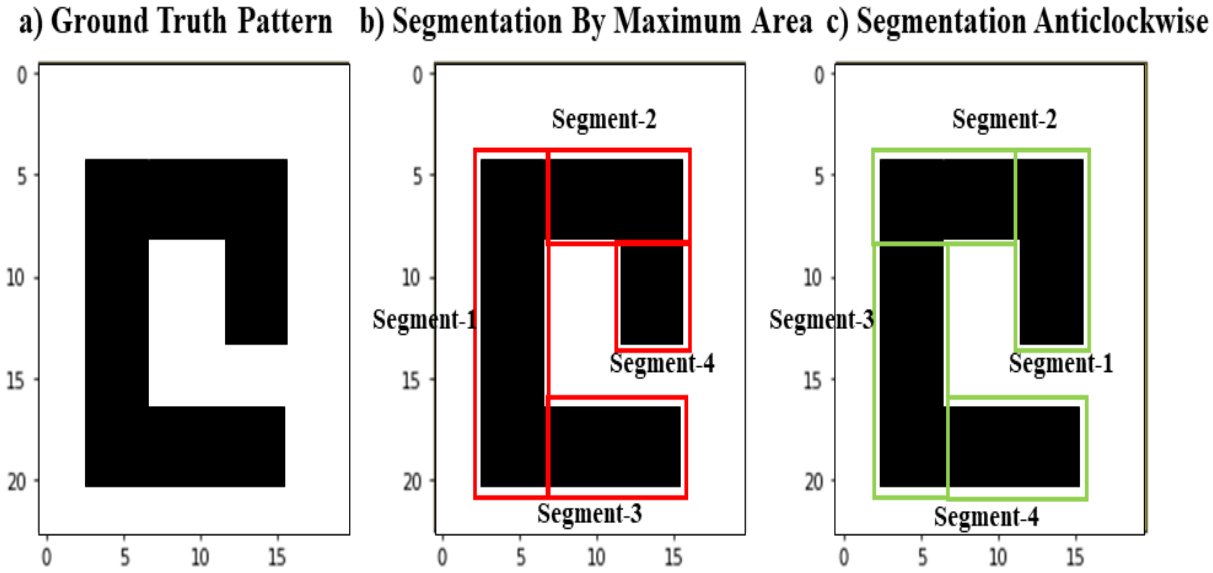


Figure 3-12: Segmenting intricate patterns before applying vectorization. (a) Ground truth design pattern. (b) Segmenting pattern by maximum rectangular area. (b) Segmenting pattern with sorted anticlockwise rectangles.

Segmenting patterns with the largest rectangle approximation is a general problem-solving question. As our ground design image consists of ones and zeros only, the binary image matrix has been converted into a histogram consisting of the number of ones divided into bins. The histogram area is solved for finding out the x,y position of the maximum area rectangle. A sample example is shown in Figure 3-13. The brute force method has been used for histograms continuous area optimization. While traversing along the y-axis, it looks like a histogram with a cumulative black pixel count. For finding the maximum area, if the minimum height continues, it needs to be checked. Each time area is calculated and compared with the previous maximum area. The new maximum area is updated until all the pixels are covered. For the segmentation of any designed pattern with the maximum area, the rectangle containing the maximum area is recursively extracted and stored in a queue for further vectorization until all the full image is covered.

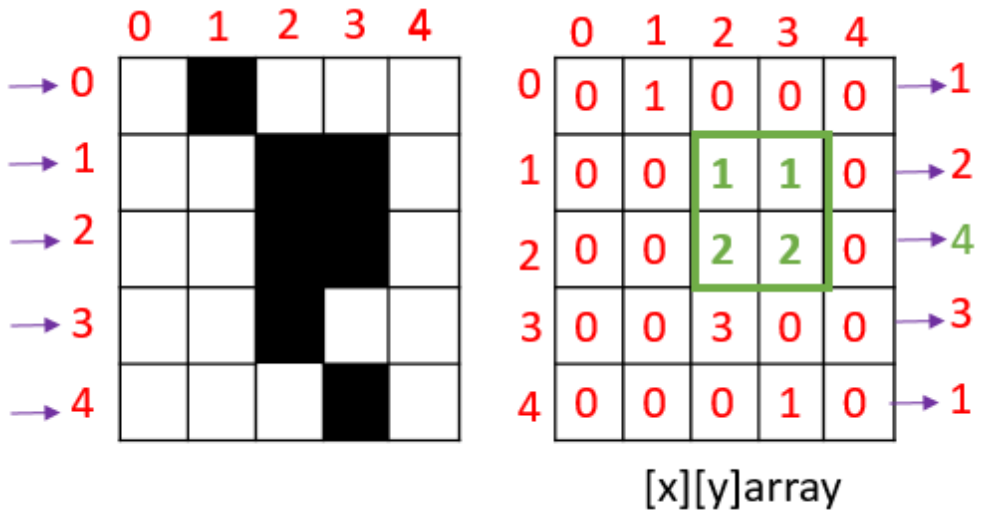


Figure 3-13: Finding the maximum area rectangle in a binary image.

Another idea of pattern segmentation is that it can be segmented clockwise or anticlockwise with rectangles. For the clockwise pattern partitioning, we start traversing from the top rightmost point and traverse downward until we reach any edge; after hitting the edge, we started traveling in the right direction till we have the same image pixel. If the pixel value changes, we start moving upward, with fixed x, increasing y until we reach the edge. From there, it moves again in the right direction until it meets the start point. This process is carried out recursively until all the partitions are collected. The same partitioning process can be carried on from the top left corner in an anticlockwise way

3.3. Experimental Methods

All the models with the generated drop sequences were validated experimentally using a custom-built inkjet printer. Independent x- and y-stages allow any drop sequences to be printed. A piezoelectric inkjet nozzle from Microfab Technologies Inc. (MJ-ATP-01-060, 60 µm diameter) was used. The ink is a commercial silver nanoparticle ink (ANP DGP 40LT-15C) with particle size 35 nm, viscosity 16 cP, and tri(ethylene glycol) monoethyl ether as the major solvent. The vapor pressure of the solvent is low, ensuring stable jetting without ink drying in the nozzle and avoiding the coin stacked morphology, which is undesirable in many microelectronic devices due to the resulting surface roughness. The substrate was glass (Fisherbrand™ Premium Cover Glass). Glass slides were cleaned in an ultrasonic bath for 30 minutes each in isopropanol, deionized water,

and blow-dried in between and afterward. Cameras on the inkjet printer were used for online and offline defect detection and quantification during printing and after drying. After printing, the ink is subsequently dried for 30 minutes on a hotplate at 60°C.). In general, DSP is varied from 95µm to 185µm in the step of 10µm, and test cases are printed for each shape, algorithms. Once the optimum DSP is observed, more test cases are printed to confirm the DSP. The optimum DSP dataset consists of 60 microscope images of printed filled patterns at optimum drop spacing (4 shapes X 3 print test case X 5 algorithms) and 36 nonfilled patterns (4 shapes X 3 print test case X 3 algorithms For other eight drop spacing lying between (90-180), Microscopic images are taken for each of the five sequence models.

3.4. Evaluation Methods

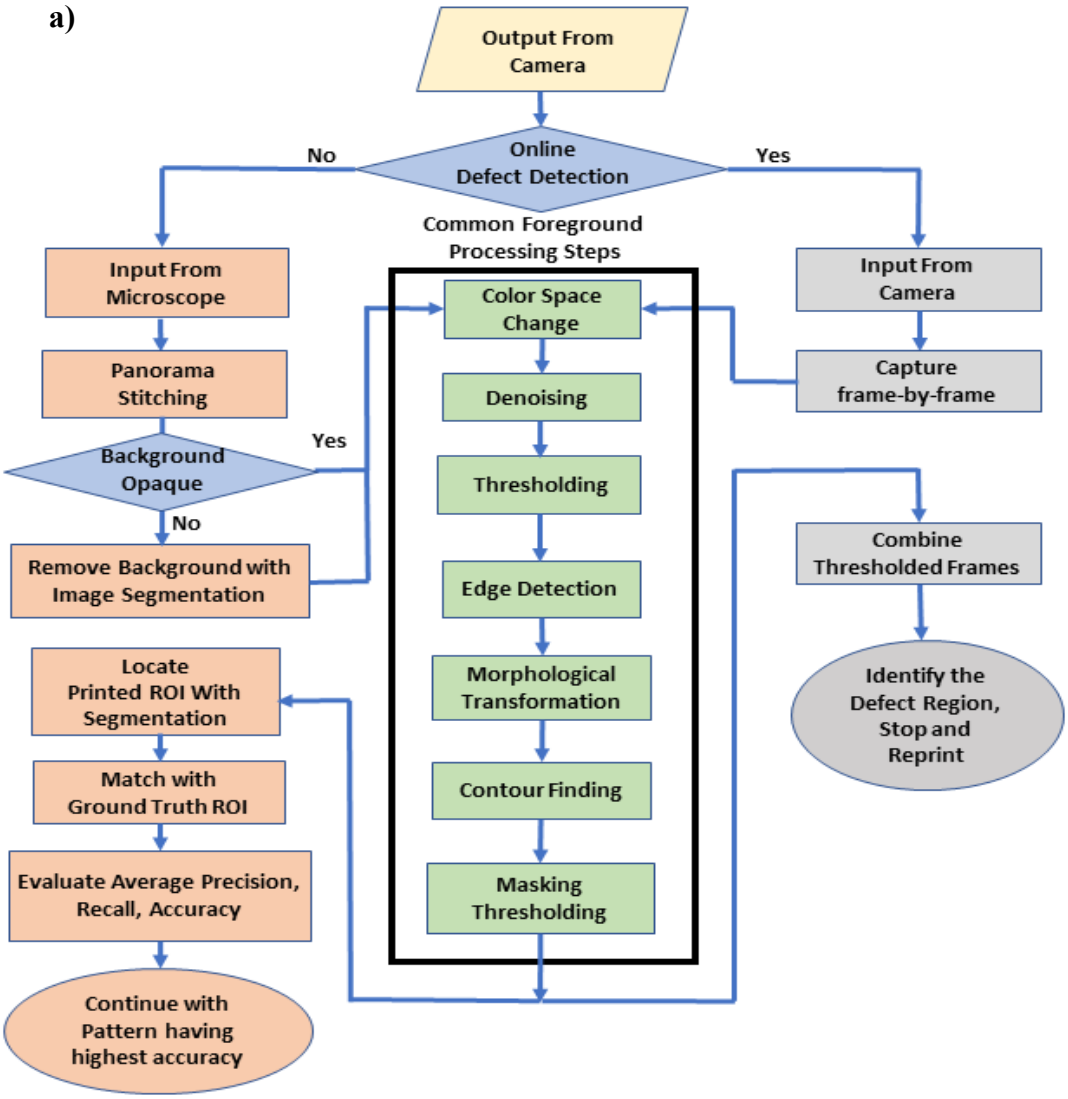
Before analyzing pattern quality, all the captured microscopic still images are passed through background noise elimination and a region of interest (ROI) cropping. Then, precision, recall, and accuracy are calculated and averaged.

3.4.1. Pre-Processing Steps for Printing Evaluation

Figure 3-14 (a) demonstrates the pre-processing steps pursued for defect detection from still images and motion videos. The still pictures are taken by a camera mounted on the printer and passed through a pipeline for stitching if required. Noisy background requires background subtraction and then extracted foreground denoising after RGB to grayscale conversion with the Rudin, Osher, and Fatemi algorithm [84]. There exist a lot of feature-based image segmentation algorithms to subtract the noisy background from the foreground, such as color histogram-based, edges, or boundary-based (based on similarity) [184], region-based (discontinuities) [185]. Printed electronics substrates, materials, and patterns can be very diverse. Prior knowledge helps in extracting the foreground pattern. For instance, the substrate background can be transparent (e.g., glass) or opaque (e.g., coated PET) with the varying color of the foreground ink. Two case examples are demonstrated in Figure 3-14(b) with multiple different image segmentation algorithm results. The transparent substrate creates a noisy background. Region-based clustering (k-means clustering, Gaussian mixture model, graph-cut) in Figure 3-14 (b) (ii-iv) or edge-based foreground processing in Figure 3-14(a) alone is not sufficient to extract the printed pattern. A combined region-based (Graph-cut) and edge-based approach ensure the extraction process works best on a transparent substrate. Graph-cut background elimination performs better than others in

terms of detecting printed objects. With the help of the foreground and background color distribution of the image, two scribbles are set. Then each pixel is assigned a probability of belonging to the foreground or background. A graph is constructed with the foreground scribble as the source and background scribble as the sink vertex. Each image pixel is placed as a node with edges as its distance from the source and the sink. Then graph-partitioning algorithm (fast maxflow min-cut) separates the foreground from the background and returns a logical result, as shown in Figure 3-14(b)(iv). There are a few scattered background areas around the periphery of the extracted foreground, but they are detached after the edge-based foreground processing steps. The final extracted output in Figure 3-14(b)(v) is a binary image, which corresponds well to the original pattern in Figure 3-14(b)(i).

a)



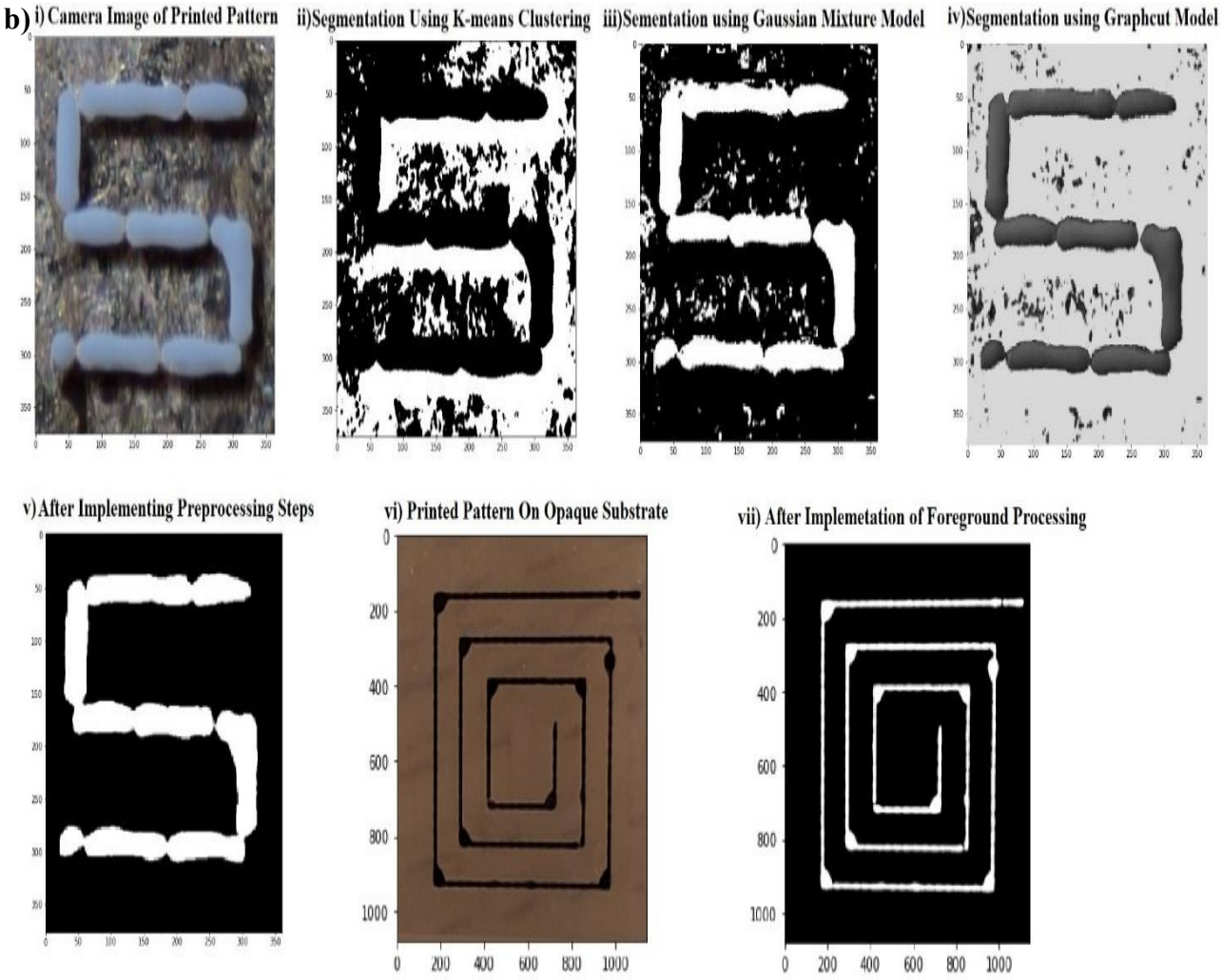


Figure 3-14: a) Processing steps for defect detection in images of the printed patterns. b) Comparison of segmentation approaches for background elimination and foreground processing. i) Printed pattern on a transparent background, ii) Background reduction through k-means clustering-based image segmentation, iii) Background reduction through Gaussian mixture model, iv) Background reduction employing graph-cut model, v) Output with processed foreground on top of graph-cut model separated background, vi) Printed pattern on an opaque substrate, vii) The binary output is obtained only from foreground processing without requiring the background reduction step.

For the opaque background, the background elimination process is not needed, and the foreground pre-processing step alone can yield a well-defined extracted pattern, as shown in Figure 10(b)(vii). The foreground pre-processing routine denoises the image, then thresholds it using

adaptive Gaussian thresholding. Laplacian edge detection is carried out on the thresholded image, and potentially disconnected edges are connected through morphological transformation: closing followed by connected component analysis. Finally, the contour is uncovered from the connected edges, and this array is used as a mask on the original noisy image. The masked image pattern is now separated from the background noise and used for defect quantization.

3.4.2. Defect Detection from Still Pictures

The pre-processed, noise-reduced microscopic image is fed through the final defect quantification pipeline. In our microscope with 4X magnification, the captured image dimension is 1,500x2,000 micrometers. If the pattern dimensions are larger than this, it cannot be captured by a single image. Multiple images are taken by moving the pattern, and they are merged as a pre-processing step. Panorama image stitching is adopted to stitch multiple microscopic images into one so that the test ROI can be easily compared with the ground truth pattern ROI. Figure 3-15(a) shows the microscopic image stitching procedure. Figure 3-15 consists of the left and right sides of the image to be stitched. A functioning stitching method is constructed by using Oriented FAST and Rotated BRIEF (ORB) feature (keypoint) detector. Like other detectors (SIFT or SURF), it efficiently identifies unique features of the left image and matches them across to the right image (shown by green lines in Figure 3-15(a)(i)) with the Brute Force method. Once matched, a 2D projective transformation is carried out to put the two images on the same image in Figure 3-15(a)(ii). The opaque background with the pattern passes through the foreground processing (black box in Figure 3-15(a)), and the processed pattern in Figure 3-15(a)(iv) is generated. Histogram-based segmentation is applied further to crop the patterned ROI from both the ground truth and denoised test image and draw a bounding box around it, as shown in Figure 3-15(b). Cropped ground truth and test ROI are converted to the same scale for defect comparison.

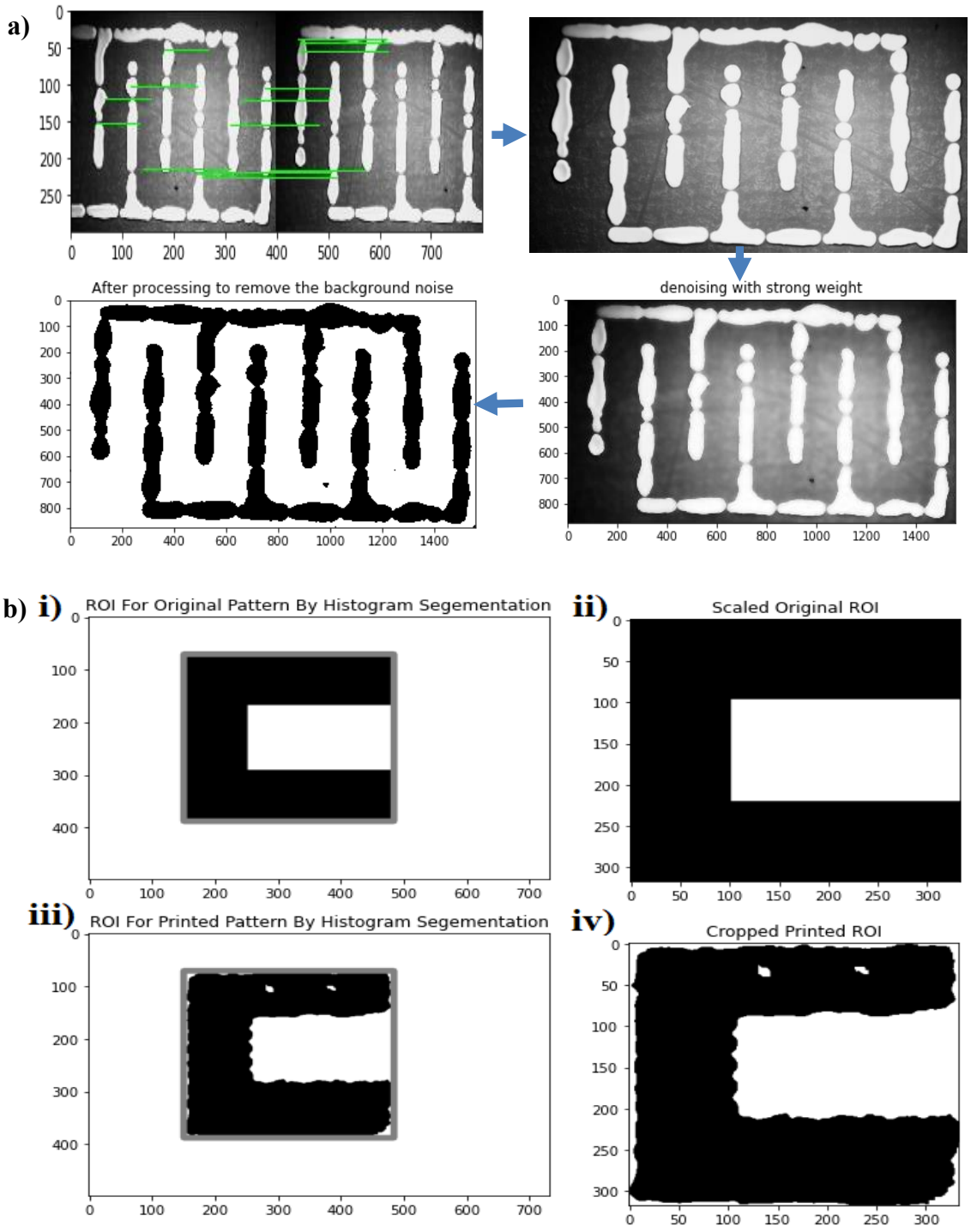


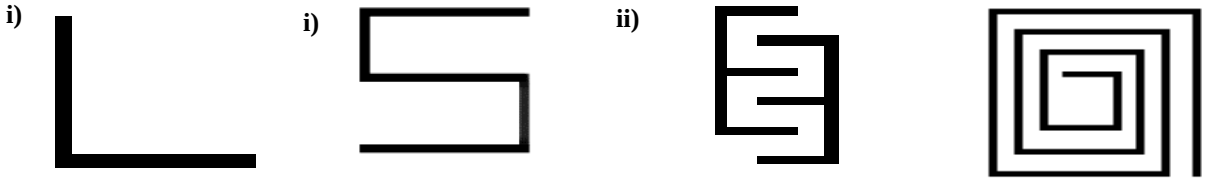
Figure 3-15: a) Stitching microscopic images for defect detection. b) Identifying, scaling, and cropping the ROI from ground truth and printed test pattern.

3.5. Results and Discussion

3.5.1. Non-filled Printed Pattern Analysis for Different Printing Sequence Generation Algorithms

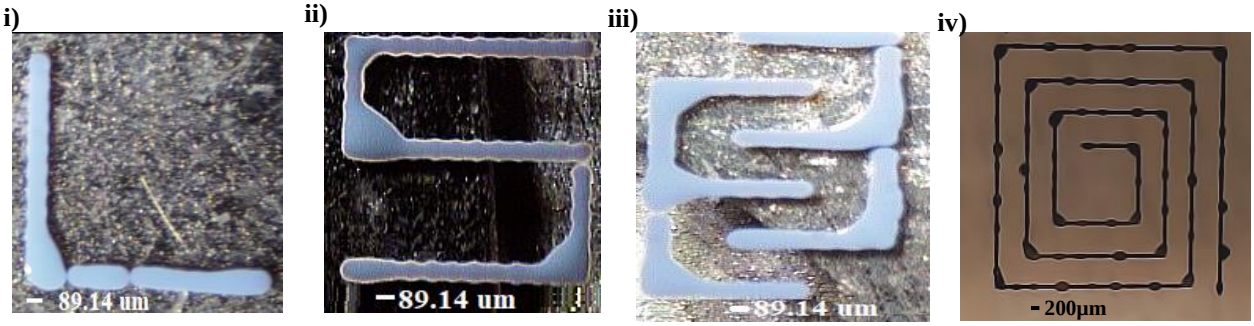
Representative experimental printing results for non-filled patterns are shown in Figure 3-16. Four small-scale patterns (L, S, Interdigitated, H) and one large-scale rectangular spiral pattern were investigated to study the impact of vectorizations on pattern quality. In all cases, DSP was optimized, and the best results are shown here. For all of the raster printing outcomes in Figure 3-16(b), some bulging is observed at the beginning of line segments. The large-scale pattern in Figure 3-16(b)(iv) suffers from bulging at regular intervals. Despite the fact that this repetitive bulging is not observed for small-scale printing in Figure 3-16(b)(i-iii), patterns exhibit abrupt disconnects at the corners and intersections. This arises from instant position changes in the positive y-direction resulting in a pressure imbalance between the already deposited large fluid bead and the impinging drop with a small radius at the next y-location. Consequently, these deviations from the intended pattern make raster printing a non-ideal choice for shapes comprising single lines with frequent direction changes such as corners, T-junctions, or crossing lines. Conversely, for vector printing, as shown in Figure 3-16(c) improved patterning results can be observed. Small-scale patterns exhibit perfect edges and corners with a noteworthy decrease in breakage at the junction. There is a systematic fluid flow with this pattern generation method as the edge drops are printed in a sorted clockwise fashion. Each time new drops impact a previously printed track, a stable transition is created, although some bulging can still be noticed at intervals for the large-scale spiral shape. The non-uniformity of large-scale patterns can be further improved by symmetric sequencing. Figure 3-16(d)(iv) shows that the large-scale line pattern exhibits almost no bulging along lines with limited bulging at the corners. For the small-scale design in Figure 3-16(c)(i-iii), the improvements are also distinctly visible together with uniform connectivity. Symmetric printing necessarily comprises reduced drop spacing due to the connecting drop spacing of 0.75 times of the drop spacing in contrast to raster and linearly vectorized DSP in Figure 3-16(b-c).

a) Non-Filled Shapes



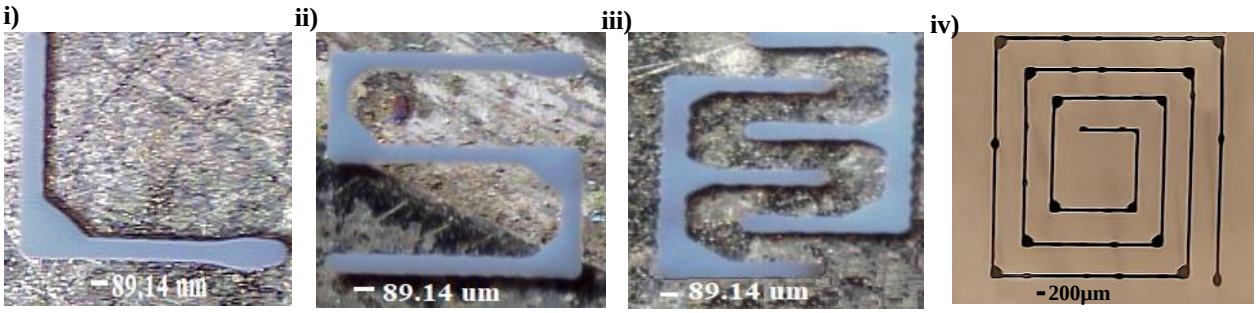
11x13 Pixel | 8x8 Pixel | 13x6 Pixel | 84x84 Pixel

b) Raster Printing Result



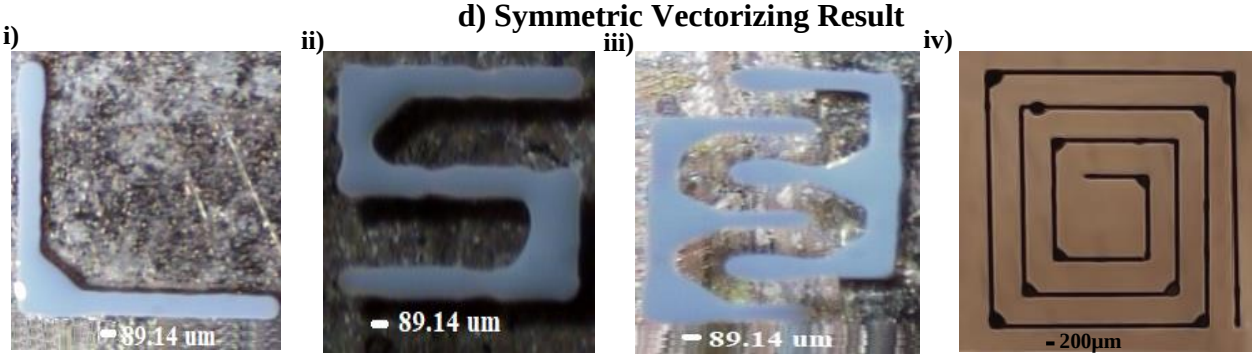
DSP=130µm | DSP=130 µm | DSP=130 µm | DSP=130 µm

c) Contour Vectorizing Result



DSP=125 µm | DSP=125 µm | DSP=125 µm | DSP=125 µm

d) Symmetric Vectorizing Result



DSP=120, CDSP=0.75 µm | DSP=120, CDSP=0.75 µm | DSP=120 µm, CDSP=0.75 µm | DSP=120 µm, CDSP=0.75 µm

Figure 3-16: Optical micrographs of experimental printing results for four different sequencing techniques with the optimum DSP and CDSP for non-filled cases a) Ground truth patterns, b)

Raster printing results, c) Linear vectorizing results, d) Symmetric vectorizing results for four different types of non-filled line shapes.

It is clear from Figure 3-16 that small-scale patterns with multiple intersections can be improved in terms of bulging with vectorized printing. Significant improvement in bulging has been observed for single-line patterns when symmetric sequencing is adopted. Large-scale line patterns (rectangular spiral) in Figure 3-16 (iv) with raster and linear vectorized printing show a regular bulging pattern while symmetric printing shows the best outcome with the least bulging except at the corners.

3.5.2. Filled Printed Pattern Analysis for Generation Algorithms

The first corner compensation algorithm performance was explored for filled structures, as shown in Figure 3-17. C shape has in total of eight corners; two of them are such that they enclose the print fluid by more than 90º. These types are referred to as inner corners, and the other ones which surround material less than or equal to 90º are termed as outer corners. G shape possesses ten corners, three inner and seven outer corners. In the case of L shape, it contains six corners, including one inner corner. The rectangle is composed of four corners, all outer. To test the effect of inner and outer corners separately, C, G, L shapes were printed with 1-pixel inner corner compensation only, and the rectangle was printed with 1-pixel outer corner compensation. It is clear from Figure 3-17 that neither inner nor outer compensation gives a good pattern for small-scale patterns. If inner and outer compensation are applied together, it will aggravate the pattern even more. At a glance, for any filled shapes, inner corner compensation is way better than outer corner compensation. Outer compensation not only changes the pattern shape but also opens up the edge boundaries. Lower DSP (110µm) is somewhat better than higher DSP (115µm) in the case of inner corner compensation. As in the case of C (one out of two), G (one out of three) printed with 110 µm DSP some corners are nicely compensated, while the others are not. Even though, for the same patterns printed with 115 µm DSP, none of the printed corners are correctly compensated, after applying for inner compensation.

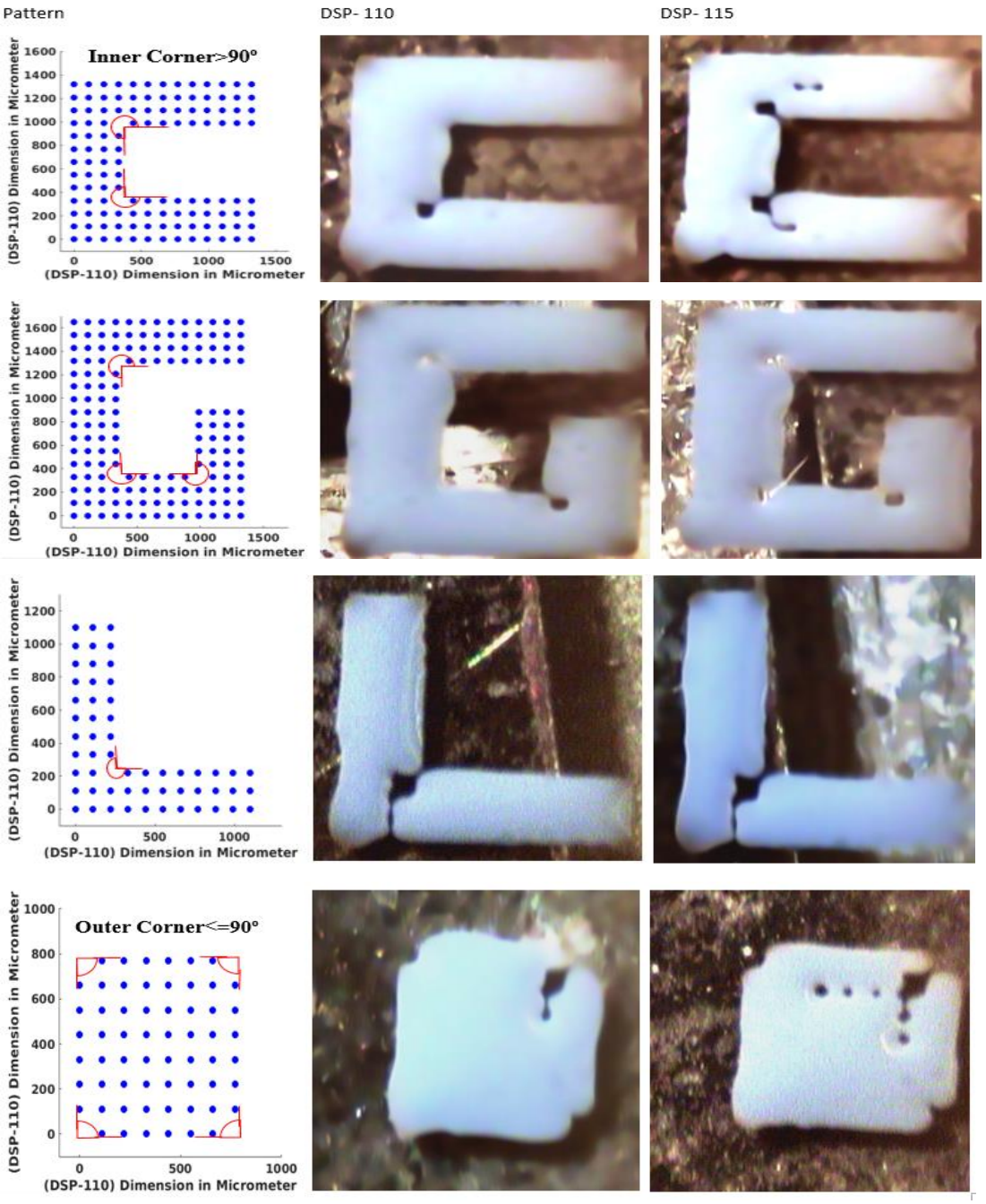
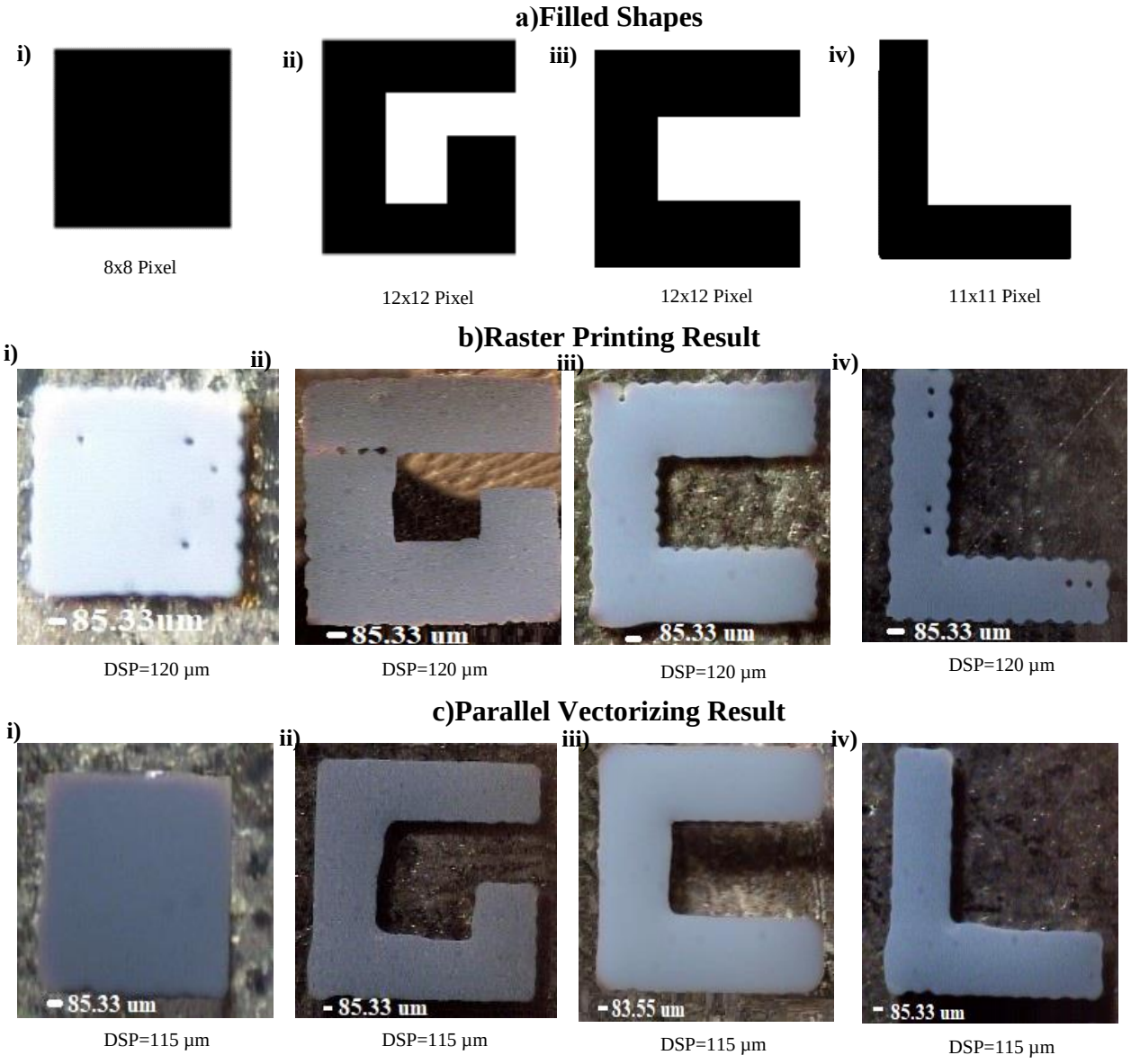


Figure 3-17: Corner compensation for different patterns. Inner compensation has been applied to pattern C, G, L, and outer compensation is applied to the rectangle.

Similarly, pattern generation algorithm performance was explored for filled structures, as shown in Figure 3-18. Four types of filled shapes (L, C, G, and Rectangle) were printed, and the results with optimized drop spacing are discussed here. Ground truth patterns are shown in Figure 3-18(a). When filled patterns are raster printed, they exhibit a number of abrupt holes, and also, the edges

are not uniform, as shown in Figure 3-18(b). These problems with raster printing can be overcome with contour-based vector sequencing, as shown in Figure 3-18(b)(c). Both radial and parallel filling exhibit similar results. Both methods enhance the quality of filled shapes compared to raster printing. The biggest advantage of contour printing is even and uniform edges. The initial printing of the boundary confines the rest of the drops subsequently printed to fill the pattern. These mechanics control the fluid pressure within the borderline region and considerably reduce bulging and abrupt holes. Images of patterns printed with symmetric ordering are shown in Figure 3-18(e). Symmetric printing improves the corners, although small inconsistencies around the edge profile and a few holes can be observed.



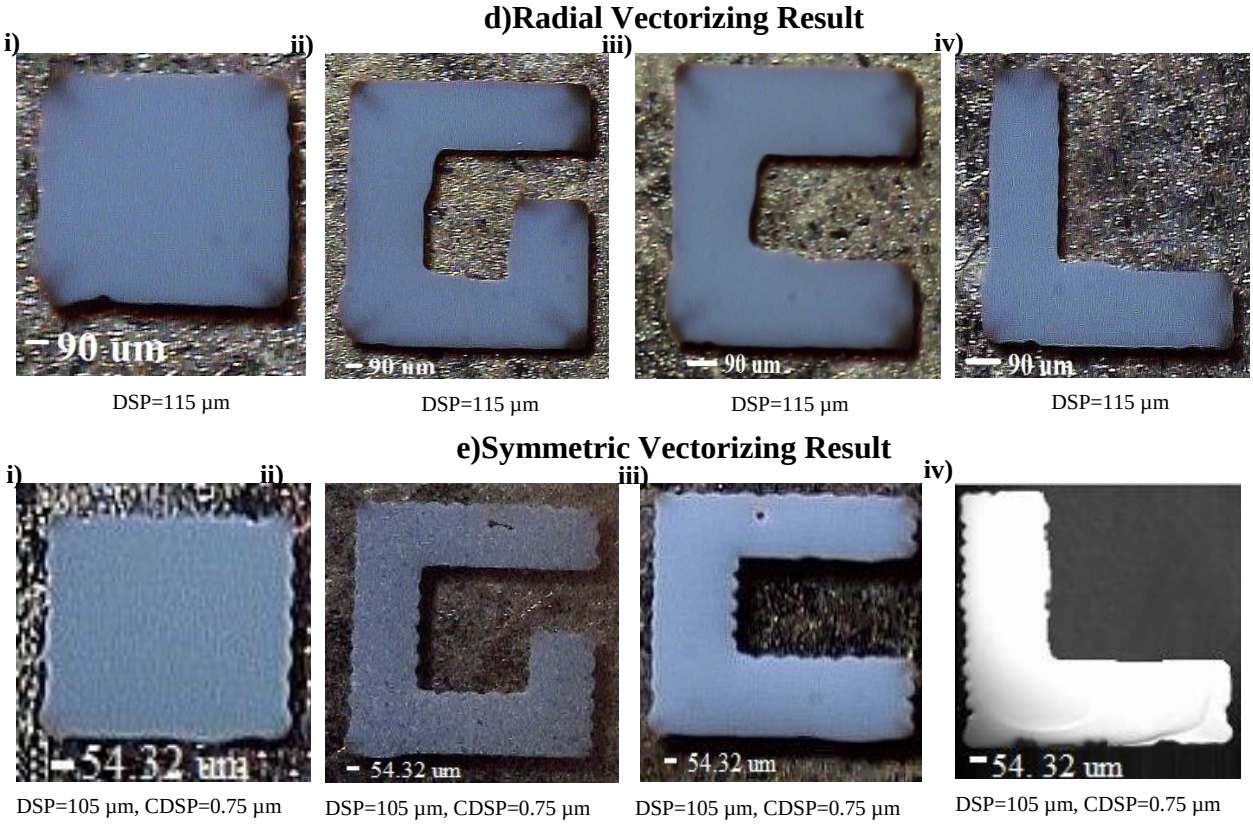


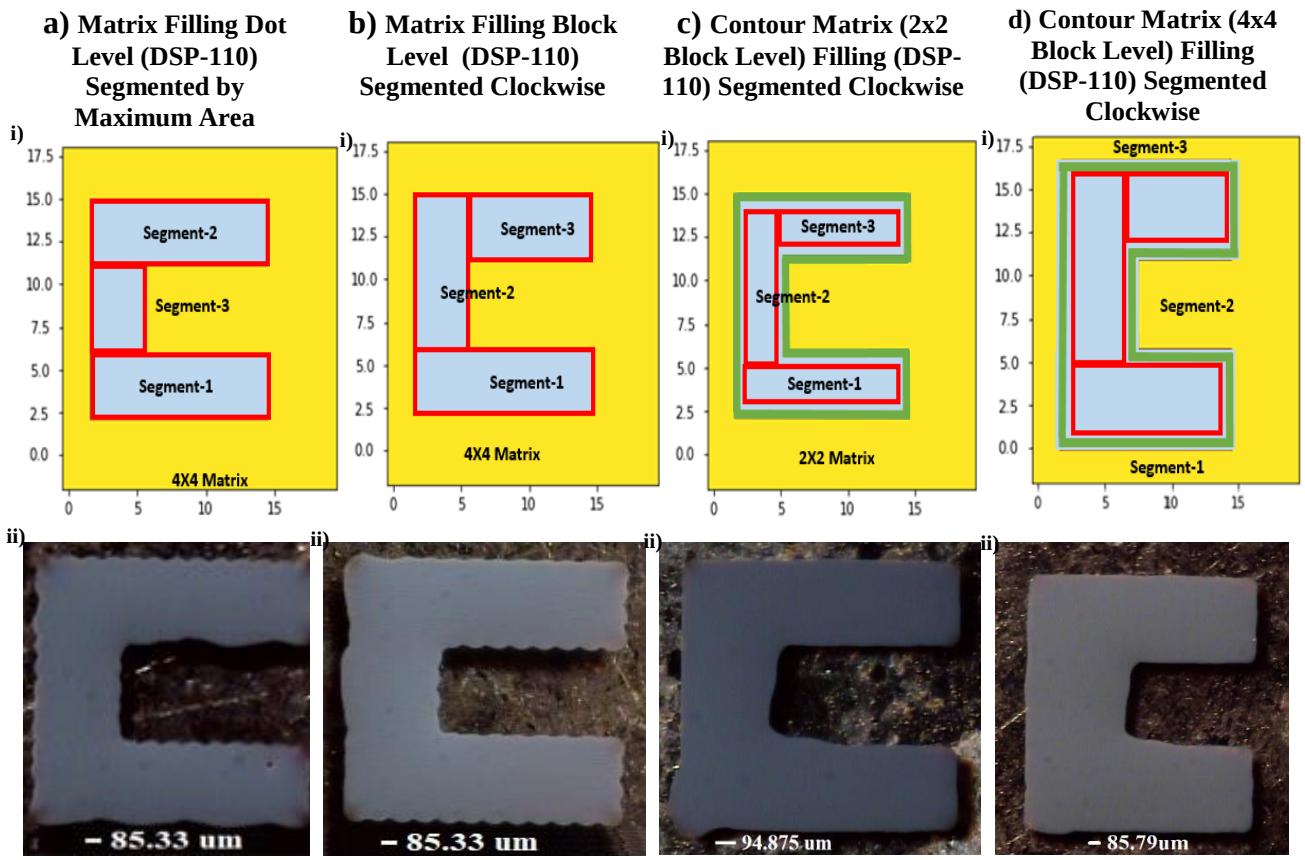
Figure 3-18: Optical micrographs of experimental printing results of six different sequencing techniques with optimized DSP and CDSP for filled cases a) Ground truth patterns, b) Raster printing results, c) Parallel contour vectorizing results, d) Radial contour vectorizing results, e) Symmetric vectorizing results.

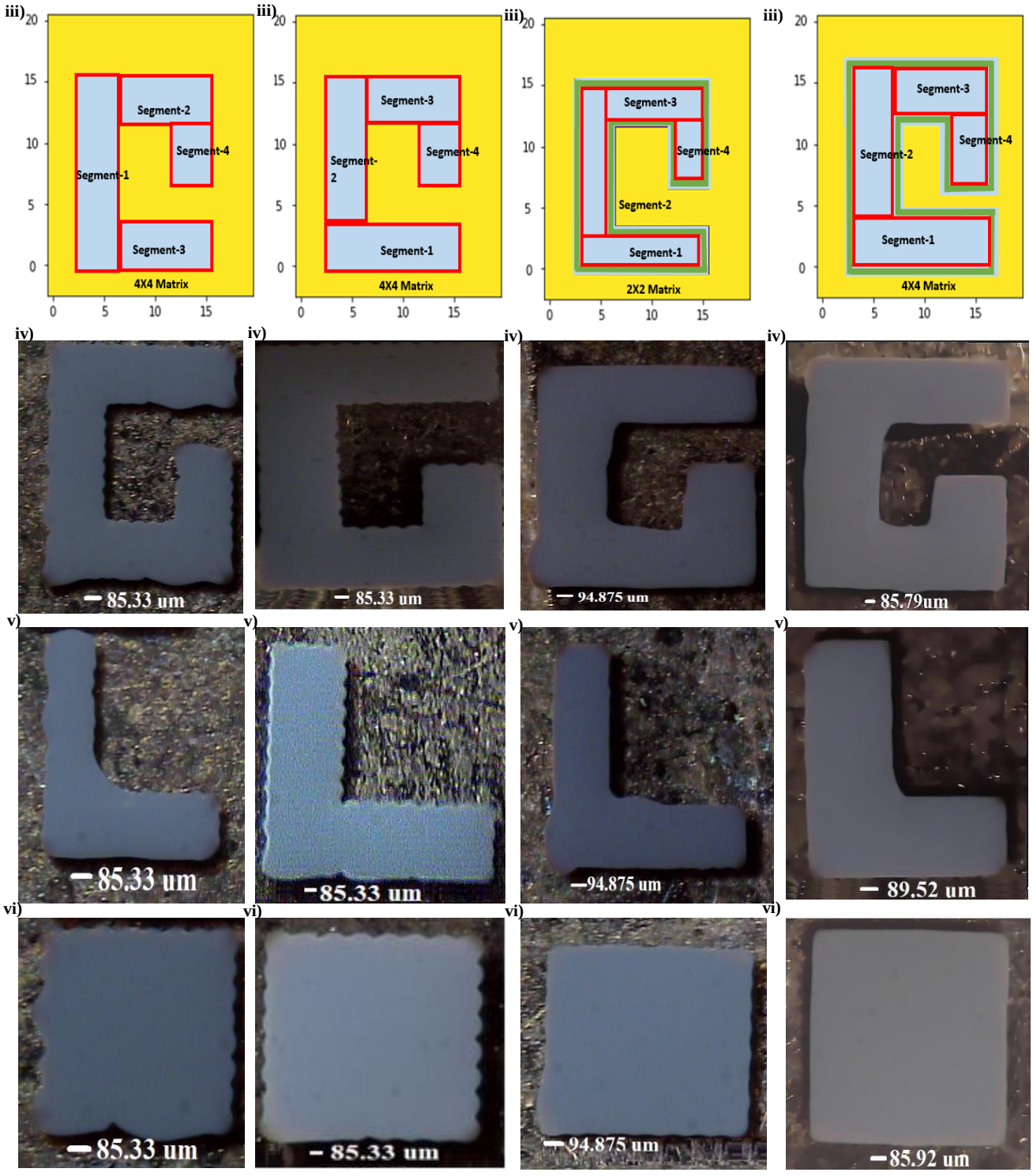
Depending on the desired printed features, contour-based vectorizing with parallel or radial filling enhances the edge profile and reduces pattern variance. However, these patterns suffer from reduced junction sharpness. For devices that can compromise in terms of edge smoothness but require strict corner localization, symmetric printing is a better choice.

3.5.3. Effect of Segmentation and Matrix Vectorization in Pattern Improvement

The fabrication method described in section 3.3 was followed to print the patterns generated with matrix vectorization and rectangular partitioning. Printing was carried out using several values for drop spacing, and print images are displayed only for optimum drop spacing. Well-defined and smooth films are obtained with contour printing first and then filling the inside drops with 4x4 blocks. At optimum drop spacing of 110µm, the dot level printing Figure 3-19(a)

in doesn't result in a better result; instead, block-level printing outperforms (b) dot-level printing for all the filled shapes, even without considering the segmentation effects (visible for rectangles in (a)(vi) and (b)(vi)). One promising result is that 2x2 block printing doesn't perform better than 4x4 blocks with all four shapes. Bulging starts at the intersection of two segments. Alternatively, a large scale monopole antenna was investigated in (vii) to study the impact of vectorizations and partitioning more evidently on pattern quality. All of the dot level printing outcomes in (a) are observed, significant bulging at edges where the first few drop in each matrix block lands. Besides, significant holes are observed when the large scale pattern is segmented by maximum area recursively. The fluid flow is not homogeneous in this case of segmentation. Most of the time, there is no connection between segmented rectangles. For block-level printing, the large scale shape is less prone to holes and edge deformation. At the same time, the small patterns are better in terms of edge deviations. Yet, bulging is observed at the beginning of each segment. Printing the contour first and then filling up inside the block-level matrix enhances pattern quality furthermore.




iii)
Segment-1
Segment-2
Segment-3
Segment-4
4X4 Matrix
iii)
Segment-1
Segment-2
Segment-3
Segment-4
4X4 Matrix
iii)
Segment-1
Segment-2
Segment-3
Segment-4
2X2 Matrix
iii)
Segment-1
Segment-2
Segment-3
Segment-4
4X4 Matrix
iv)
85.33 um
iv)
85.33 um
iv)
94.875 um
iv)
85.79um
v)
85.33 um
v)
85.33 um
v)
94.875 um
v)
89.52 um
vi)
85.33 um
vi)
85.33 um
vi)
94.875 um
vi)
85.92 um

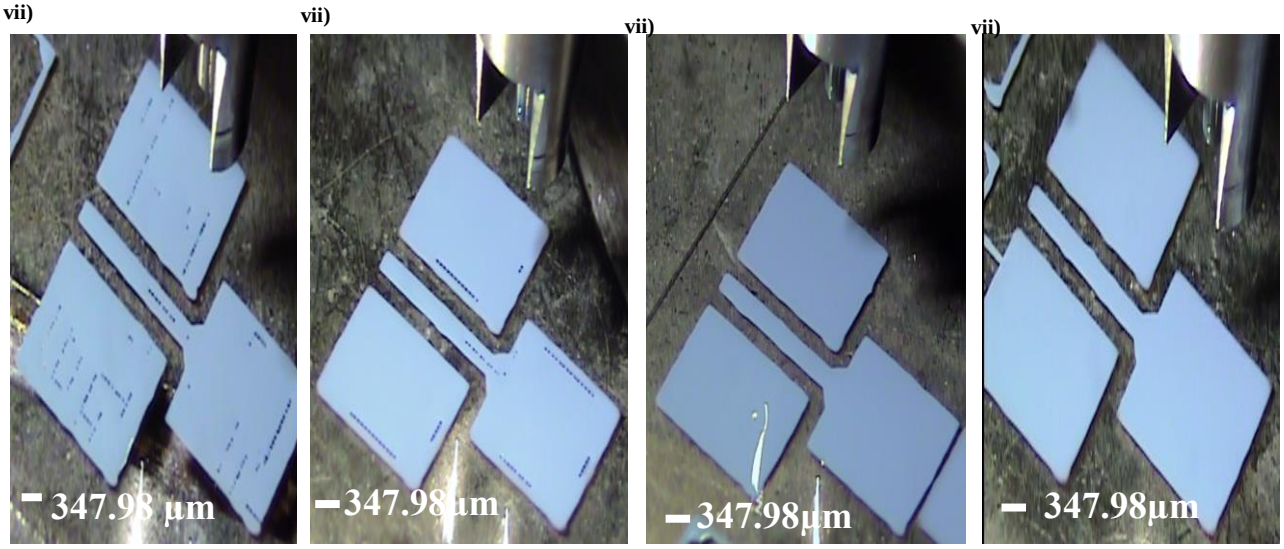


Figure 3-19: Optical micrographs of experimental printing results for matrix vectorization combined with segmentation techniques with the optimum DSP for four filled cases. Ground truth pattern segmentations are displayed by (i),(iii). (a) Matrix printing results at dot level with rectangles partitioned by maximum area, (b) Matrix printing results at block level with rectangles partitioned clockwise, (c) Contour Matrix printing results at block-level (2x2) with rectangles partitioned clockwise, (d) Contour Matrix printing results at block-level (4x4) with rectangles partitioned clockwise.

Matrix is a promising way of patterning by inkjet printing. Provided that the contour is printed first, and the inside is filled with a small block (2x2) matrix vectorization, the results are similar to radial or parallel printing. As the block size increases, the contour matrix patterning result improves. For devices that can compromise in terms of edge smoothness but require strict corner localization, symmetric, or block matrix printing is a better choice. All the printing schema results in Figure 3-21. We can see that the printed C pattern is improving while contour-based vectorization is adopted. With a block size of 4x4 and clockwise segmentation, the printing output is improving. In a word, as the pattern scale size increases, segmentation combined with contour-based block vectorization yields better printing.

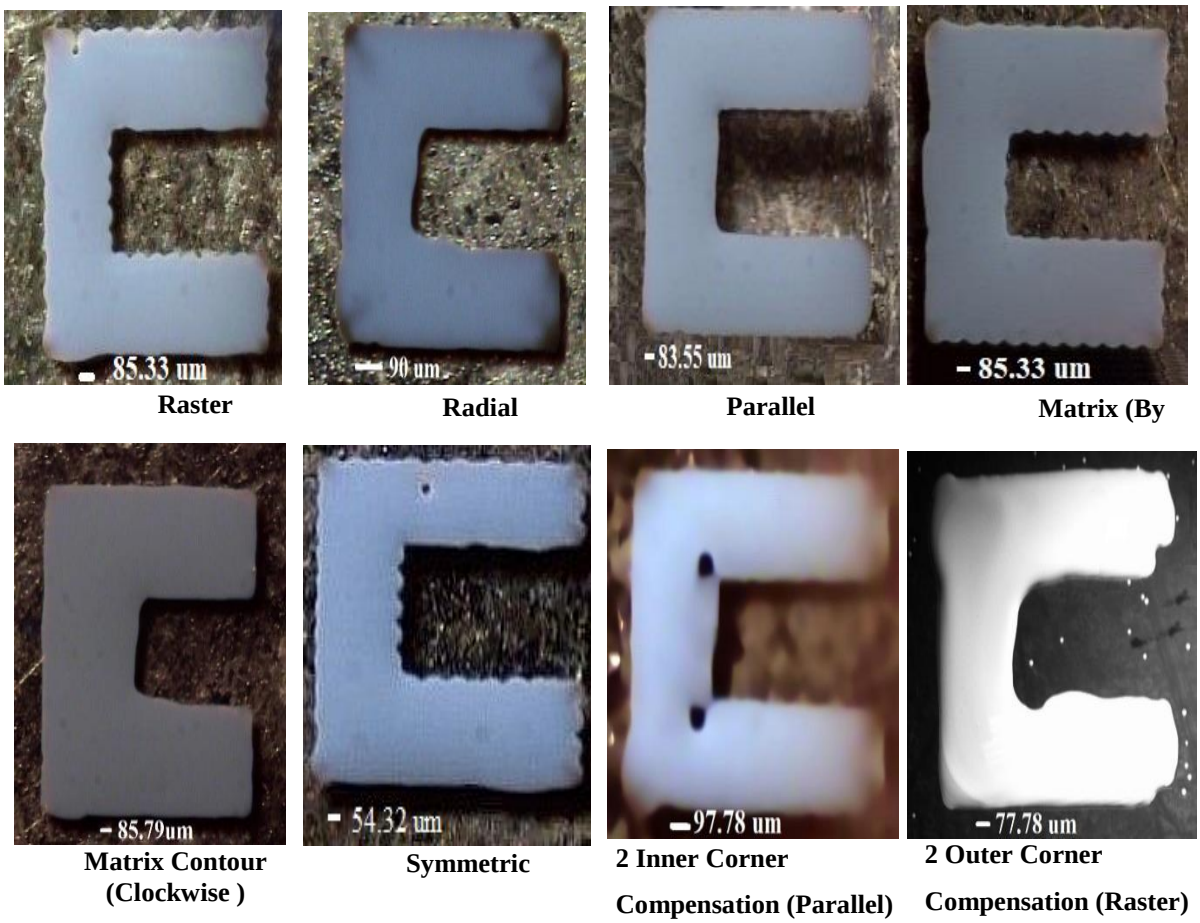


Figure 3-20: Images of patterns printed through five different drop sequencing algorithms. Pictures have been taken right after printing without the application of thermal annealing. Pictures have been processed through vision-based non-contact pattern evaluation before sintering at 180°C.

3.5.4. Numerical Evaluation of Printed Patterns

Patterns generated with the four models were evaluated through computer vision-based defect detection, superior to traditional qualitative human inspection, lacking standardization and efficiency. It consists of two parts: the offline image segmentation module and the online motion tracking module, responsible for different quality assessment types. The online module performs early quality monitoring along with capturing data for later offline processing. Offline testing performs the final defect quantification. Table 3-1 lists a comparison of the pattern generation models applied to the performance evaluation test.

Table 3-1: Comparison of different pattern generation methods.

Pattern Type	Shape Type	Precision	Recall	Accuracy (%)	Standard Deviation of Accuracy
Raster	Filled (DSP-120 µm)	0.935	0.773	88.70%	3.49%
	Non-Filled (DSP-130 µm)	0.891	0.837	84.034%	1.77%
Contour Vector (Radial, Parallel Average)	**Filled** (DSP-115 µm)	0.903	0.999	**93.420%**	**2.91%**
	Small Scale	0.931	0.871	**88.548%**	**1.66%**
	Large Scale → Average	0.919	0.825	85.552%	0.029%
	Non-Filled → (DSp-125µm)	0.887	0.848	87.05%	0.846%
Symmetric Vector	Filled (DSP-105 µm, CDSP-0.75 DSP)	0.889	0.976	89.14%	3.35%
	Small Scale →	0.837	0.839	83.36%	1.165%
	Large Scale → Average	0.965	0.897	**94.18%**	**0.011%**
	Non-Filled → (DSP-120 µm, CDSP-0.75 DSP)	0.901	0.868	88.77%	0.588%
Corner Compensation (1 Pixel)	Filled (DSP-120 µm)	0.88	0.596	79.84%	2.07%
	Non-Filled (DSP-130 µm)	0.42	0.351	51.57%	1.75%
Matrix (4x4 Block) (Clockwise Segmented)	Filled (DSP-110 µm)	0.895	0.939	92.18%	2.87%
Contour Matrix (4x4 Block) (Clockwise Segmented)	Filled (DSP-110 µm)	0.916	0.983	**94.89%**	**2.14%**

Radial and parallel filling results were averaged and the row labeled as contour vectorizing. Each sequencing model's performance is evaluated based on the average precision, recall, and accuracy calculation (equations (39),(40),(41)) of two binary image sets (ground truth, test) based on the number of true positive (TP), false positive (FP), true negative (TN) and false-negative (FN) pixels. Positive refers to white pixels (background), negative refers to dark pixels (pattern) in the binarized images.

$$\text{Precision}=\frac{TP}{TP+FP} \tag{39}$$

$$\text{Recall}=\frac{TP}{TP+FN} \tag{40}$$

$$\text{Accuracy}=\frac{TP+TN}{TP+TN+FP+FN} \tag{41}$$

When the ground truth non-print white (1) pixels are not white (0) in the corresponding test image, it implies the printed patterns have bulged over the intended pattern boundaries, and false negative is high, recall is low. Precision becomes low when false positive is high due to some ground truth black pixels (0) transformed to white pixels (1) in the test image exhibiting shrinking, i.e., hole defects. Filled shapes have lower recall (i.e., exhibit bulging defects) compared to non-filled shapes prone to lower Precision (i.e., exhibit shrinking or disconnect defects). Accuracy increases with high true positive and true negative values, which means the number of black pixels in the printed image is close to the number of black pixels in the ground truth image. The white pixel number similarly matches. At the same time FP, FN is low, signifying reduced bulging, holes, and shrinking, generating the optimum printed pattern. These calculations of print quality metrics are carried out with the DSP that gives the best result for a particular method and pattern type. Figure 3-21 gives an overview of the key performance measures of the four vision models. In the case of filled shapes, contour vectorization has the highest accuracy. While contour is filled first, then the inside drops are filled with 4x4 matrix, optimum DSP 110 µm clockwise, block by block, the printed results show further improvements over shapes. For non-filled shapes, the symmetric sequencing model outperforms other methods showing better average accuracy, although having some holes and shrinking defects. These quantitative findings agree well with the qualitative findings described above.

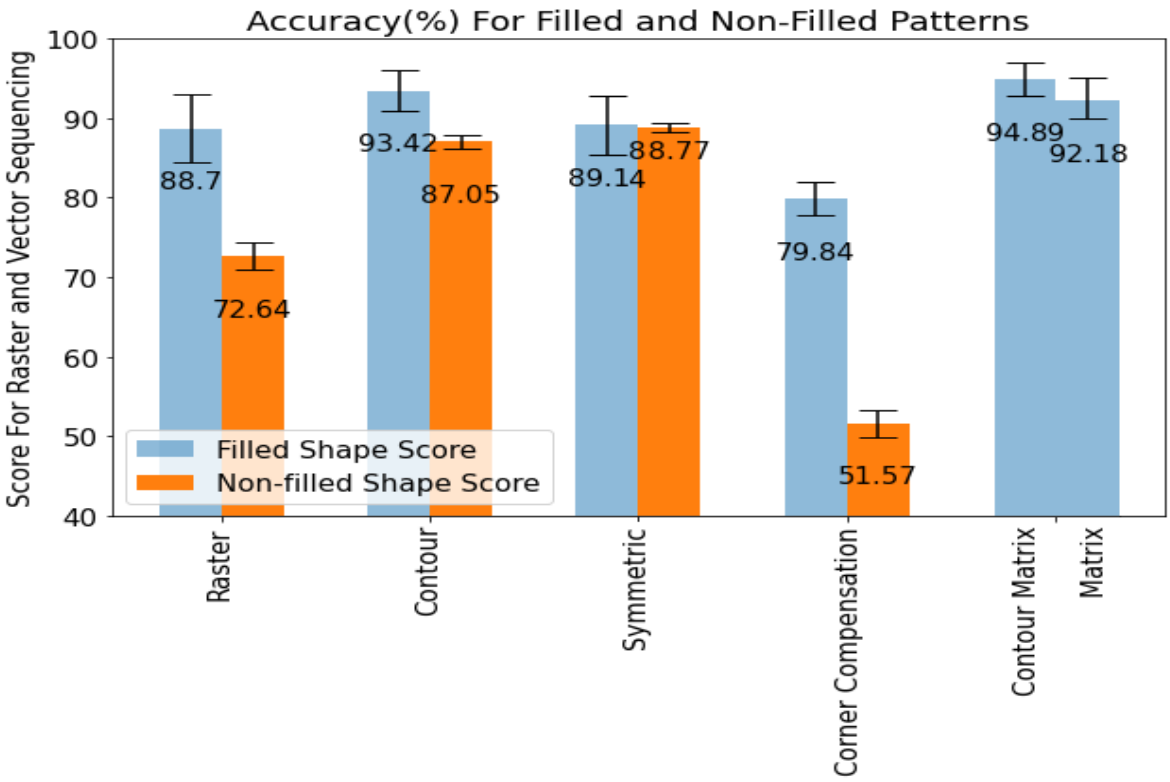


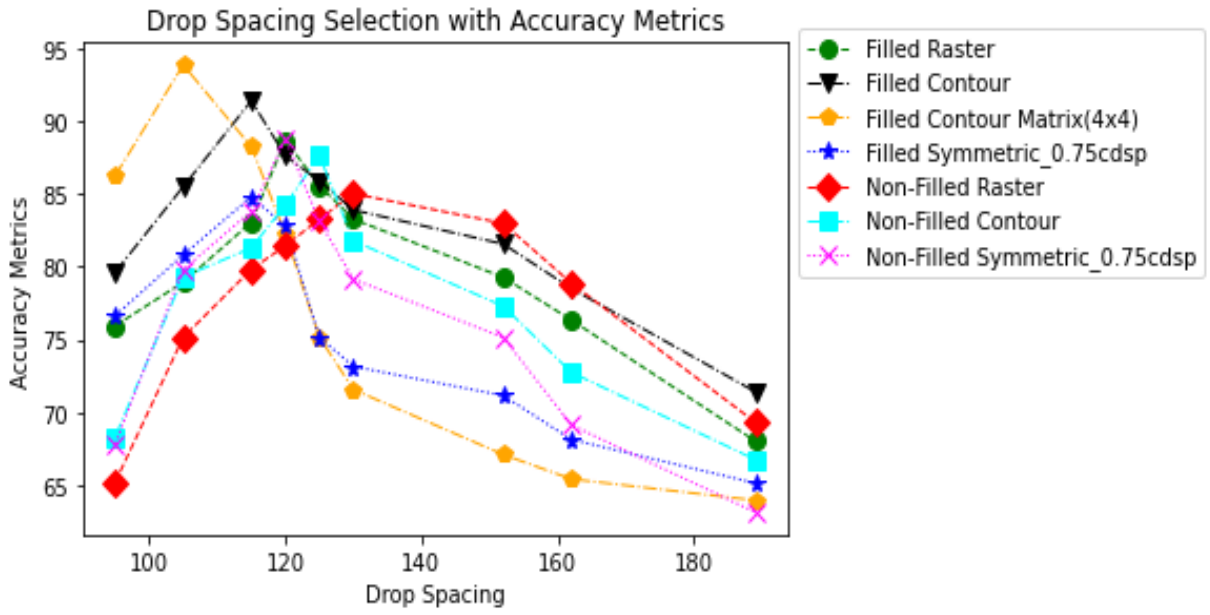
Figure 3-21: Accuracy of different pattern generation methods from still pictures.

Figure 3-21 attributes very similar accuracy to non-filled patterns printed with contour and symmetric schema. This difference has been statistically tested with a t-test at 95% confidence interval. We obtained T-score=2.89, p-value=.0077 < α (0.05). As it indicates that the difference between contour and the symmetric non-filled group is 2.89 times the difference within the group, so we reject the null hypothesis. And say that symmetric and contour sequencing results in different results for nonfilled cases. Similarly, for the matrix and contour-matrix filled cases, we conclude that the print results are different with a T-score 3.028, p-value=0.0049< α (0.05), and rejected null hypothesis. Feedback System

The defect detection system described in the previous section is not only to study the merit of different printing sequence generation methods but also to provide useful feedback on the pattern design. Generating optimum printed patterns also depends on the DSP and for symmetric printing CDSP, in addition to the drop sequencing method. This parameter needs to be re-optimized whenever a new ink-substrate combination is used or when printing onto inhomogeneous substrates with already existing printed layers. Incorrect DSP selection can lead to bulging, shrinking,

scalloping, and beads [155]. Using these evaluation metrics as feedback can efficiently identify the optimum DSP and CDSP for a specific substrate and ink. This selection has always been difficult, especially for complex and diverse shapes and large-scale patterns. Previously, most studies have identified the best DSP and CDSP only for single lines or simple designs. Figure 3-22(a) shows some of our DSP optimization results. Filled and non-filled shapes are averaged over three different patterns each (filled: C, Rectangle, L; non-filled: Interdigitated, L, S) for different values of DSP. The optimum DSP giving the highest accuracy is different for filled and non-filled shapes and the different sequencing methods. For filled shapes, raster, contour, and symmetric printing should use a drop spacing of 120 µm, 115 µm, and 105 µm, respectively, for this ink-substrate system. For non-filled shapes, raster, contour, and symmetric printing should use a drop spacing of 120 µm, 115 µm, and 105 µm. Non-filled shapes require larger DSP. Even with the optimum DSP and CDSP values and the optimum sequencing method, some of the printed features can still have small abrupt holes. Large-scale pattern printing is more prone to holes. The accuracy-based feedback system can be used for backfilling abrupt holes at their centroid location. After finding the centroid of the void contour, the coordinates are merged, and the next time the printer prints the complete pattern, it will compensate for the hole by adding an additional drop at that location. Figure 3-22(b) is an example of center coordinate detection for abrupt micron-scale holes.

a)



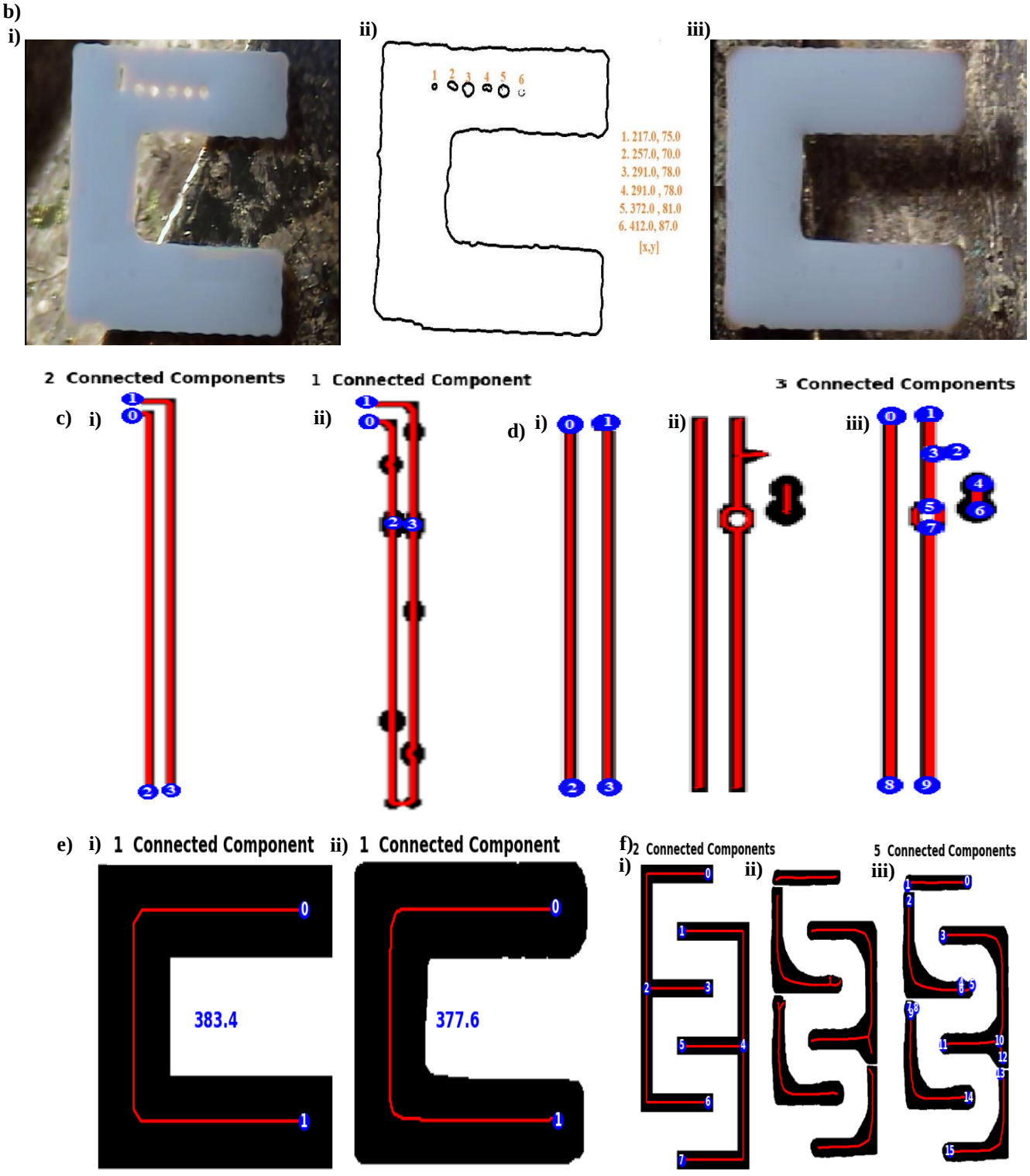


Figure 3-22: Pattern evaluation and error pruning using a graphical approach. a) Optimum drop spacing selection using accuracy as the evaluation metric. b) Holes coordinate findings and reprinting with appended hole coordinates. c) Short circuit path determination for two lines in close

proximity. (i) Ground truth pattern. (ii) Printed lines are short-circuited between nodes 2 and 3 due to significant bulging that changes the skeleton and nodes. d) Void space and additional printed track tracing. (i) Ground truth design pattern, (ii) Skeleton determined in the printed lines, (iii) Graph nodes are created based on the skeleton connectivity. e) (i) Ground truth pattern skeleton, (ii), and network graph for a filled shape. f) (i) Ground truth interdigitated pattern skeleton components, (ii) Printed pattern skeleton components, (iii) Routing through pattern framework, and open circuit path tracing in interdigitated structure.

Another method for defect detection in complex printed micron-scale patterns is skeleton-based graph inspection similar to optical inspection methods for PCBs [91][92][93]. Figure 3-16(c)-(e) shows defects patterns such as short and open circuits, bulging, and holes. Such defects can lead to catastrophic loss of functionality in electronics systems. To analyze all connected components in the printed structure, Zhang's skeletonization [181] is adopted. The ROI is reduced to a 1-pixel wide representation by recursive identification and removal of border pixels as long as the structure's connectivity is conserved. Network graph-based inspection is performed on the skeletons with central nodes and edges with weights as the distance between connecting nodes of the printed patterns. The shortest path distance within the network is utilized for short circuit tracing. A list of connected and disconnected nodes and their distance in μm is used to detect and visualize fault patterns. Comparing ground truth and printed pattern skeletons become much easier through fault tracing from one node to another. Figure 3-22(c)(i) shows a path in the printed pattern image between nodes 0 and 1 through nodes 2 and 3. The path distance is 417.56 μm, while the ground truth pattern is designed as two distinct lines without any connection between 0 and 1. The printed pattern in Figure 3-22(c)(ii) contains a single component signifying a short circuit fault. The pattern in Figure 3-22(d)(ii) exhibits a three-component skeleton with one separate track due to unwanted material deposition. When the printed pattern graph coincides with the ground truth with little deviation as in Figure 3-22(e)(i) and (ii), it can be considered defect-free. This filled C-shape structure corresponds to Figure 3-18(a)(iii) printed with DSP 115 μm and contour algorithm. Ground truth and the printed pattern have the same number of connected components. The distance between the farthest nodes is nearly identical, with a 1.51% deviation. An interdigitated raster printing result with 130 μm DSP (cf. Figure 3-16(a)(iii)) is tested for fault identification in Figure 3-22(f). It demonstrates the presence of an open circuit fault in the constructed graph. The ground

truth pattern in Figure 3-22(f)(i) has two different connected skeletons with nodes 0,2,3,6 and 1,5,4,7. The printed design skeleton in Figure 3-22(f)(ii) is segregated into five component skeletons clearly exhibiting open circuits. These results agree with the qualitative and quantitative analysis of pattern quality for different sequencing methods. Additionally, with the skeleton method, it is straightforward to find the shorted fault path in the pattern and optimize the design pattern. This graph approach is particularly helpful in finding skeleton anomalies of the designed pattern due to fluid flow and the exact location of the short- and open-circuit fault from a specific node.

3.6. Conclusion

A novel computer vision-based scheme is developed and evaluated to optimize the control of inkjet printing at the micrometer scale. A pattern image's pixels are taken as the input and converted to a drop sequence by means of feature mapping and shape detection algorithms. Finally, printed results are analyzed using automated defect detection. Four ways to generate the sequence of drops are compared. Contour based vectorization (parallel and radial filling) gives the best results for filled shapes, and symmetric vectorization gives the best results for non-filled shapes. Defect quantification and identification are vital for quality monitoring and final output judgment. The print output is evaluated concerning the input image, and they are compared to detect defects. The developed computer vision-based electronics printing improves print quality for complex shapes in an automated fashion, which is impossible or requires extensive manual intervention with traditional methods. In the future, this algorithm can be extended to combine different sequence generation methods for different parts of a layer that have different requirements.

4. Chapter 4: Optimized Pattern Comparison with Contact Based Pattern Evaluation

4.1. Introduction

In Chapter 3, we have analyzed several vectorization techniques to improve pattern printing. Next, we moved forward to carrying some experiments on how these patterning techniques change the surface morphology and the electrical property of the print. The results are presented in the following sections. Pattern structure of different shapes and scale are printed using solvent-based silver Nano-particle ink using a customized MicroFab inkjet printer with a nozzle radius of 30µm. Then they have been thermally annealed and then processed for characterization. This annealing process of any ink material to its solid form is termed as sintering. Sintering helps to achieve desirable electronic properties. Metallic ink responds well to sintering. During the process, heat transfers to the ink particles. Although as metallic inks, mostly gold (Au) [191], silver (Ag) [192], copper (Cu) [193] are used mainly for their high electrical conductivity, particle-free metal precursor inks are also quite common [194]. Our designed system perfectly aligns with nanoparticle inks; it can also be utilized for other inks such as metal-organic decomposition (MOD) and metallic salt-based inks. We have used thermal sintering via a hotplate baking for the glass substrate. However, cheap plastic substrates such as PET, PI can't sustain hot plate annealing temperature more than 150°C. In that case, alternative non-contact sintering adaptation is suggested that will selectively heat the printed materials without heating the substrate [191][194]. Before printing, stable jetting has been ensured using the ML-driven feature settings. And the patterns are obtained with optimized DSP for each of the sequencing schemas, as discussed in Chapter 4. Overall, an optimized homogeneous and reproducible pattern structure with the fewest defects has been used as a sample of each of the printing methods during characterization. The results obtained represent an essential step towards the application of automated industrial inkjet printing.

4.2. Methodology

4.2.1. Pattern fabrication

Patterns are fabricated using a commercial silver nanoparticle ink (ANP DGP 40LT-15C). The diameter of the nanoparticle is 35 nm, and the primary solvent is triethylene glycol monoethyl

ether (TGME). The same ink was used throughout chapter 3 and 4. We used fisher 2''x 2'' cut glass mostly and, in some cases, PET as a substrate. Before printing, the substrates were ultrasonicated for 5 minutes in DI water, then 5 minutes in IPA, and the last 5 minutes in DI water. In between each ultrasonication, the substrate is dried with compressed air. We have demonstrated patterning with a metal nanoparticle-based ink. The vectorized pattern optimization of nanoparticle-based conductive ink has not been studied before extensively but can yield significant insights for optimizing the patterning conditions. For example, during printing, the pattern defect quantity monitoring through precision, recall, and accuracy can help evaluate the device layer before going to the next manufacturing level. As the pattern generation algorithm changes, so do the pattern quality. The pattern assessment results of chapter 3 are corroborated with sheet resistance and surface topography profile in this chapter. The pattern quality metrics and electrical properties are correlated for different pattern generation schema. Figure 4-1 shows a summary of the fabrication process.

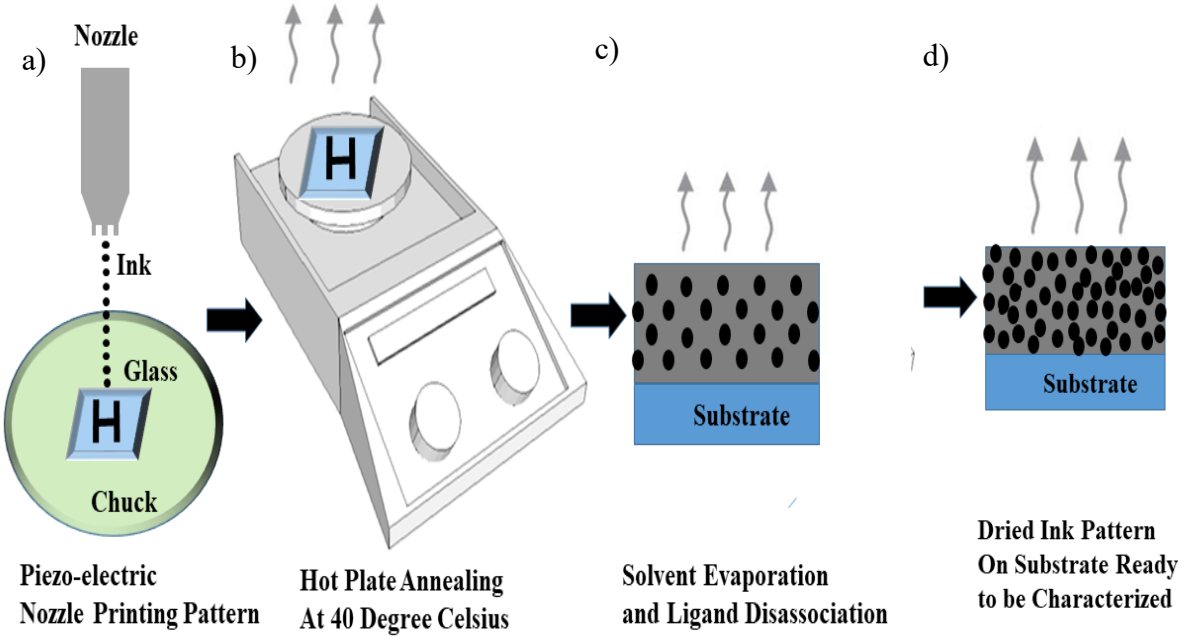


Figure 4-1: Sample pattern fabrication steps. (a) Patterns are fabricated on cleaned glass with a specific pattern generation process. (b) Samples are subjected to the first step, drying at 40°C temperatures. (c) Organic ligands in the nanoparticle ink start dissociating, and the solvent starts lessening. (d) Samples are subjected to the second step annealing at 180°C temperatures. The nanoparticles agglomerate, the region grows, leading to complete sintering.

For electrical and surface property estimation, the prepared patterns are subjected to a 40°C drying step for 20 minutes on a hotplate. Then they are annealed on a hotplate at a constant temperature of 180°C. The first drying step removes the excess solvent from the patterns. The high-temperature sintering is carried out for 30 minutes. This two-step annealing prevents rapid drying and generates better surface topology.

4.2.2. Sheet Resistance Characterization of Printed Pattern

Sheet resistance effectively relates thin layer resistivity to the layer thickness. This characterization is the primary physical parameter for quality assurance of any conductor, semiconductor or insulator printed films, photovoltaics, OLED, sensors, packaging, semiconductor, and many more industries [195][196][197]. In the following section, sheet resistance characterization has been conducted to infer the quality of printed patterns regarding resistance. With the printed pattern sintered at the same temperature of 180°C having similar resistivity, the sheet resistance R_s changes mainly with the pattern thickness given by equation (43).

$$\text{Volume resistivity, } \rho = \frac{ResistanceXArea}{Length} = \frac{ResistanceXWidthXThickness}{Length} = \frac{RxA}{L} \quad (42)$$

$$\text{Sheet Resistance, } R_s = \frac{\rho}{t} = \frac{ResistanceXWidth}{Length} \quad (43)$$

Here, R_s and t are the sheet resistance and printed pattern thickness.

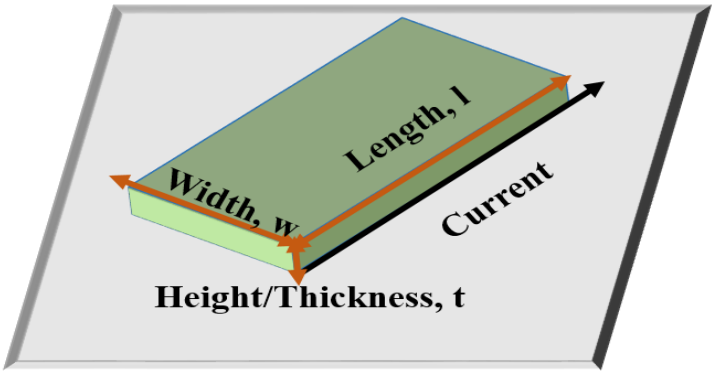


Figure 4-2: Pattern thickness, width, length definition for sheet resistance estimation.

There are many contact-based sheet resistance characterization techniques such as Four-Point-Probe, Van der Pauw. Again in the collinear test, four equally spaced, co-linear probes are sited into the pattern; the current is driven between the outer two probes while voltage is measured between the inner two probes. Four-Point-Probe can be also be placed in a non-collinear way. Yet,

4-pt measurements are limited to un-patterned films and not applicable to intricate patterns and shapes. For small, complex patterns, current pathways can be changed by the pattern geometry. Considering our varied shape study, Van der Paw's method [197][196] has been implemented for estimating filled pattern sheet resistance. Van der Paw measurement has been carried with the setup in Figure 4-3(a-d). We have used the 4200A-SCS Parameter Analyzer to carry out the Van der Paw measurements, as shown in Figure 4-3 Van der Paw (a)(c). The four probes eliminate measurement errors due to the probe resistance, the spreading resistance under each probe, and the contact resistance between each metal probe and the beneath printed material. Throughout the experiments, the current is forced between points 1 and 2, and the voltage difference has been measured between 3 and 4 for a 4-point probe.

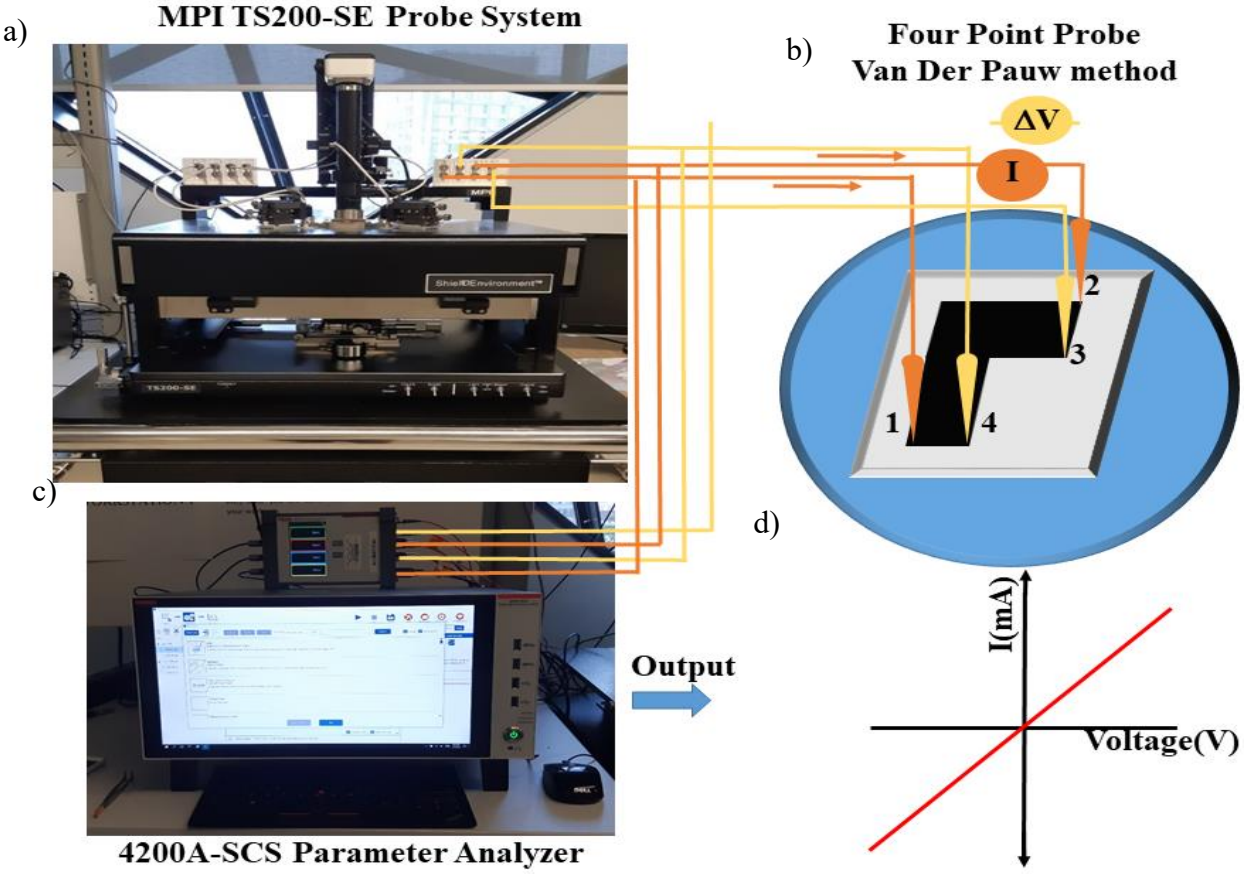


Figure 4-3: Pattern evaluation through resistance measurement using contact-based four-point probing.

Using this method, the sheet resistance is derived from the measured voltage differences and current values around the periphery of the sample with the configurations shown in Figure 4-3 (b). Once all the voltage differences are measured, two values of sheet resistance, R_{s1} and R_{s2} are calculated using the equations (44)-(46):

$$R_{s1}=\frac{\pi}{\ln 2} * f_1 * \frac{(V_{43}-V_{34}+V_{14}-V_{41})}{4I} = \frac{\rho_1}{t} \quad (44)$$

$$R_{s2}=\frac{\pi}{\ln 2} * f_2 * \frac{(V_{21}-V_{12}+V_{32}-V_{23})}{4I} = \frac{\rho_2}{t} \quad (45)$$

$$R_s=\frac{R_{s2}+R_{s1}}{2} \quad (46)$$

The parameter analyzer measures the voltage and the current through the sample. The f_1, f_2 geometrical factors changes with the pattern shape symmetry. These values are derived from the following f-Q plot. The Q values are related to the ratio of two subsequent edge resistance. For perfect symmetric printed patterns such as a rectangle $f_1=f_2=1$. As a reference, the same ink's sheet resistance value is cross-checked against one previous work of the same group [192]. All the non-filled line patterns studied in Chapter 3, three small-scale patterns (L, S, Interdigitated), and four types of filled shapes (L, C, G, and Rectangle) were investigated to study the impact of vectorizations on pattern sheet resistance. Values are calculated only with the patterns printed with optimized DSP. For each filled, non-filled shape, four samples with optimized DSP have been tested, and the results are illustrated in the result section.

In the next stage, pattern surface roughness is measured using a stylus profilometer. We have utilized Alpha-Step D-600 Stylus profiler systems to inspect and analyze the printed pattern surfaces. There are varieties of electronics applications that are highly sensitive to surface topology variation. Surface profiling is one of the best available methods to determine the printed layer step height, roughness, and waviness. Once the thickness profile, Z, is obtained in x or y direction over the shape's length, the mean thickness is calculated. Then mean thickness deviation, Ra, is calculated using the equation (47). The standard deviation of the thickness is estimated with the root means square of thickness deviation using the equation

(48).

$$R_a=\frac{1}{L}\int_0^L |Z(x)-Mean|\,dx \quad (47)$$

$$\text{Standard Deviation} = \sqrt{\frac{1}{L}\int_0^L |Z(x)-Mean|^2\,dx} \quad (48)$$

4.3. Results and Discussions

4.3.1. Surface Topology of the Patterns

The Alpha-Step D-600 profilometer can detect height deviation even down to 10 angstroms to as large as 1.2 mm. The profiler was tuned to use a load equivalent to 0.03 mg to carry out the experiments with its inbuilt optical deflection height measurement mechanism and magneto static force control system. Sample thickness measurement is carried out along the Z-axis with 400 data points per profile. All the printed patterns evaluated in chapter 3 have been profiled to cross-check the validity of the results. A sample pattern surface characterization schema is shown in Figure 4-4. The stylus travel distance is plotted in the x-axis with the millimeter dimension, and the estimated thickness is plotted in the y-axis with micrometer units.

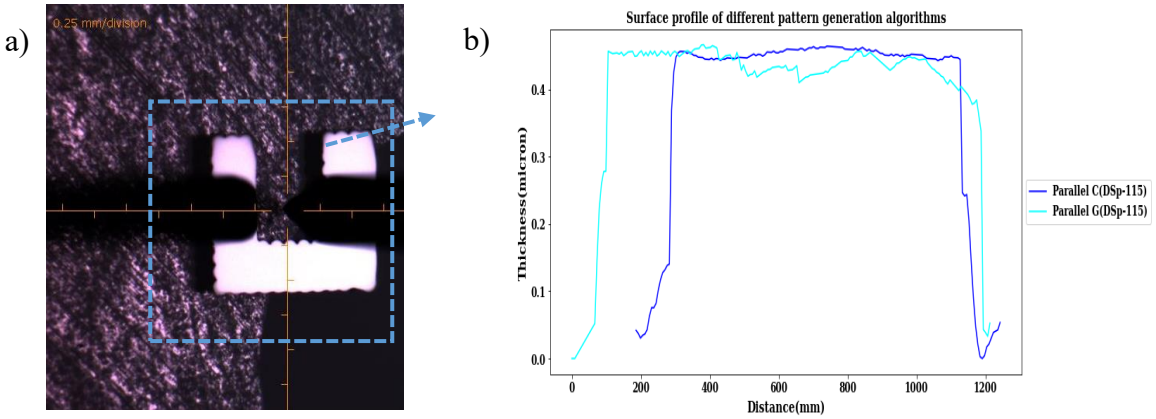


Figure 4-4: A representative thickness measurement example of the sample. a) The black needles are the stylus head and its shadow touching the print pattern surface. The scales are showing the surface dimension in millimeters. b) Thickness profile of C and G shapes printed with a parallel printing method with an optimum DSP of 115µm.

The characterization of the pattern surfaces was conducted along different cross-sections (x-directions) over the print surface area with 1 nm z-resolution. In total, 120 different samples were analyzed. The representative experimental results are summarized in Table 4-1. 36 different samples constituting three small-scale non-filled patterns (L, S, Interdigitated) printed with three pattern generation schema at optimized DSP were studied for their thickness profile. Similarly, approximately 80 thickness profiling was conducted for the four types of filled shapes (L, C, G, Rectangle) printed with five vectorization schema with optimized drop spacing.

Table 4-1: Surface profiling results for several shapes and pattern generation schema.

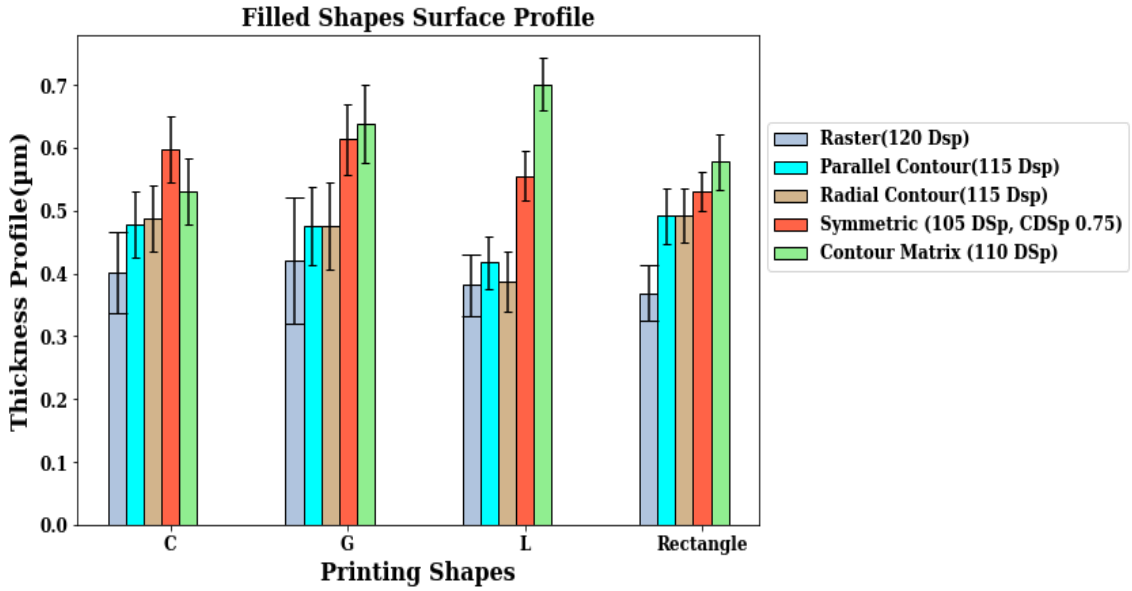
Type	Algorithm	Shape	Mean Thickness, t(µm)	Mean Deviation (Ra) (µm)	Mean Thickness Over Different Patterns (µm)	Mean (Ra/t) (%)	Standard Deviation of Ra (%)
Filled	Raster (120-DSP)	C	0.401	0.064	0.393	16.47%	2.552%
		G	0.460	0.101			
		L	0.361	0.050			
		Rectangle	0.33	0.044			
	Parallel (115-DSP)	C	0.456	0.052	0.426	10.76%	**0.910%**
		G	0.464	0.062			
		L	0.395	0.042			
		Rectangle	0.387	0.044			
	Radial (115-DSP)	C	0.487	0.053	0.476	11.53%	**1.144%**
		G	0.536	0.069			
		L	0.387	0.047			
		Rectangle	0.492	0.043			
	4x4 Matrix Contour (110-DSP)	C	0.598	0.053	0.574	**7.87%**	**1.136%**
		G	0.613	0.056			
		L	0.555	0.039			
		Rectangle	0.53	0.032			
	Symmetric (105-DSP)	C	0.637	0.069	0.651	11.27%	1.225%
		G	0.701	0.091			
		L	0.577	0.062			
		Rectangle	0.689	0.072			
Non-Filled	Raster (130-DSP)	L	0.135	0.008	0.174	7.09%	0.83%
		S	0.148	0.007			
		Interdigitated	0.240	0.022			
	Contour (125-DSP)	L	0.231	0.009	0.249	**5.01**%	**0.503%**
		S	0.263	0.013			
		Interdigitated	0.315	0.019			
	Symmetric (115-DSP)	L	0.325	0.017	0.362	6.11%	0.88%
		S	0.350	0.018			
		Interdigitated	0.410	0.0328			

Based on our measured thickness, the surface roughness parameter Ra has been calculated and reported in Table 4-2. As we can see, there is a mean thickness variation among the profiles printed with different algorithms. For filled shapes, the thickness increases from raster to symmetric. It is because optimum DSP decreases, and as a result, the mean thickness increases overall. For the non-filled cases also, a similar pattern is observed from raster to symmetric. Interestingly, the height deviation percentage result is similar to what we have found for non-

contact precision-recall-based pattern evaluation. Mean Ra divided by an average thickness of four filled shapes for each of the five pattern generation methods reveals that raster printed pattern is undoubtedly experiencing the most variation. Contour 4x4 matrix printed block by block has the least variation, and it expresses the pattern homogeneity over the print dimension. Next, radial and parallel are exhibiting the same trend of roughness variation. Same as filled shapes, raster printed non-filled patterns show a little more deviation than the contour and the symmetric ones. Overall the filled shapes contain more roughness variation than the non-filled cases, which is expected due to their increasing dimensions. Also, the thickness standard deviations are decreasing with contour printing for any type of pattern. The table results are visualized further using Figure 4-5(a)-(b) and Figure 4-6.

The bar charts in Figure 4-5 (a) give an overview of crucial thickness measures of filled patterns printed with five vision-based pattern generation models. Overall, any shapes generated with contour-matrix show the lowest mean thickness deviation (Ra). Better surface homogeneity arises when the outside contour is printed, and the inside blocks are filled with a 4x4 block of matrix. Radial and parallel filling also display very low thickness abnormality compared to raster. So, the conclusion derived from the vision-based evaluation agrees well with the thickness profile. Again, as the area and complexity increases, the deviation increases from rectangle to L, C, G.

a)



b)

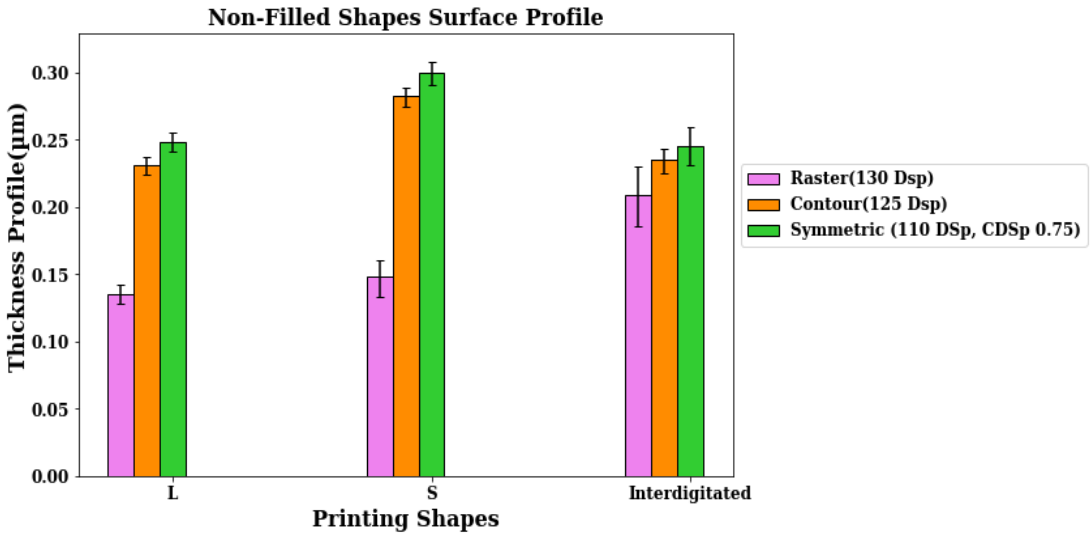


Figure 4-5: Surface mean thickness of experimental printing results of five different sequencing techniques with optimized DSP and CDSP. The mean thickness deviation, Ra, is shown by the black error bar. (a) Four filled patterns are printed with raster, parallel contour, radial contour, symmetric, block contour matrix vectorizing. (b) Three non filled cases printed with raster, contour, symmetric vectorizing.

Among the three non-filled test cases, interdigitated thickness varies the most for all three types of patterning. Raster printed L, S, Interdigitated samples show the lowest thickness with the highest DSP among all methods. Also, they show inferior surface quality in comparison to the other two. For the interdigitated structure, all of the patterns show the most variation. As the complexity increases with more junctions and edges, the thickness is also varied in those regions. For L shape, variation is almost the same for all the three algorithms. Raster, as usual, shows the lowest thickness with the highest drop spacing of all. The greater the drop spacing, the more the ink material spreads in on the substrate; this lowers the Z dimension's material thickness. Contour and symmetric both show the best uniformity, with the lowest deviations. However, for the interdigitated structure, the thickness profile shows a lot of variation. Due to the higher number of junctions, three drop segments, and connecting drops, these thickness results agree well with the results described in Chapter 3.

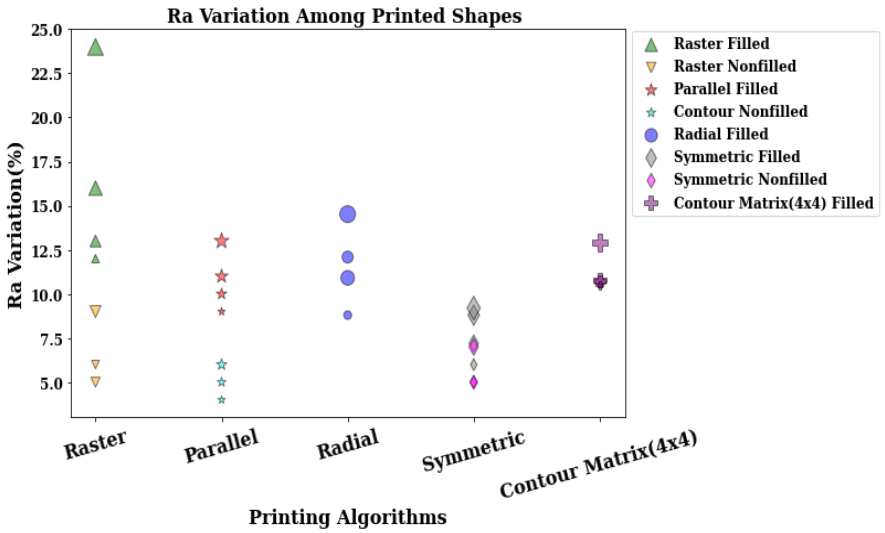


Figure 4-6: Ra variation percentage among pattern types, shapes, and printing methods.

Figure 4-6 summarizes the whole finding in a single scatter plot. The increasing size of the filled case markers denotes Rectangle, L, C to G. Non-filled type markers also exhibit the same rise trend L, S, Interdigitated. At a glance, non-filled thickness varies the least among different patterns. Non-filled Ra variation is less than filled cases. It is attributable to higher optimal print drop spacing accompanied by lower print area. It is noticeable that the variation increases almost for every printing schema with larger and more complex patterns. For filled cases, G structure, and non-filled cases, an interdigitated structure is the most susceptible to variation. Raster characteristically contains the highest deviation from the mean calculated thickness.

4.3.2. Pattern Sheet Resistance Measurement

The van der Pauw method is used to measure sheet resistance using Keithley 4200A-SCS Parameter Analyzer. The sheet resistance effectively infers to the characteristics of the printed pattern. Then sheet resistance can be converted to resistivity with the measured film thickness, as mentioned in the previous section. Figure 4-7 shows a sample sheet resistance result from our experimental setup. We ensured that the samples are homogeneous without isolated holes; contacts are placed close to the sample's edges. Samples are mostly symmetrical, and the contacts are numbered from 1 to 4 in counter-clockwise order. The current I12 is into contact one and taken out of contact 2. The voltage V34 is measured between contacts 3 and 4.

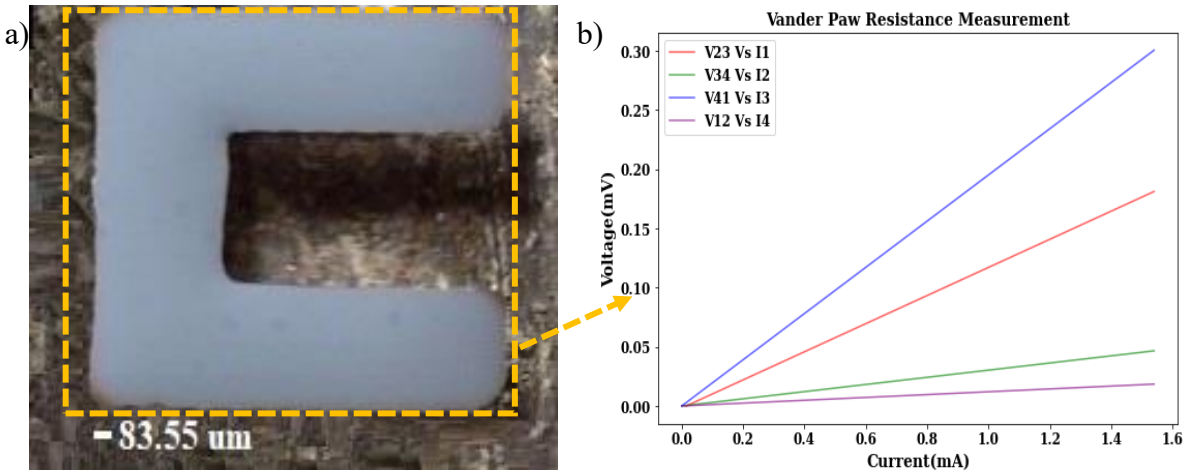


Figure 4-7: Vander-Paw sample sheet resistance schema. (a) Profile of the C pattern printed through parallel contour printing using DSP-115 µm and silver ink. (b) The voltage and current are measured through the four probes.

It is evident in Figure 4-7(b) that the voltage differences are not the same along between every two points of the C shape. According to the Van der Pauw setup rule, the geometric factor needs to be updated to ensure correct sheet resistance. For the non-filled cases, the width is so small that we have to take a 4-pt sheet measurement for the two points, so they don't have any geometric correction factor. For C shape, factors of 0.98 and 0.624 were used. In the case of G, 0.857, and 0.42 were used. For L shape, the values of 0.458 and 0.501, and for rectangle 1 and 0.9878 were used. The results of all the patterns are displayed in Table 4-2. It is visible from Table 4-2 that there is little dispersion in the measured sheet resistance over the shapes. Overall, the mean sheet resistance over the patterns for filled and non-filled shapes is almost the same irrespective of pattern sequencing schema. The mean sheet resistance is between 0.2-0.3 Ω/sq for any printed patterns and sintered at 180ºC. The variation in the sheet resistance on glass is mostly dependent on pattern type and structure. As we have used different optimal drop spacing for different printing methods, a mentionable change is thickness variation. The influence of the sheet thickness is not negligible, and it will change the pattern sheet resistance.

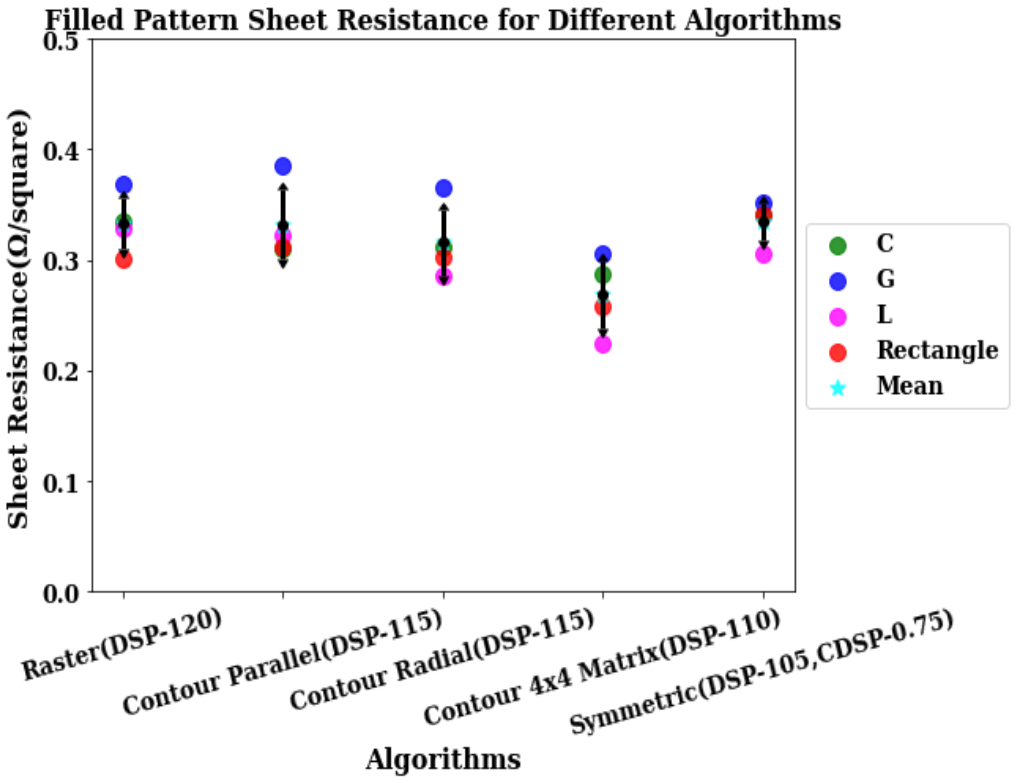
Table 4-2: Sheet resistance measurement results for printed patterns.

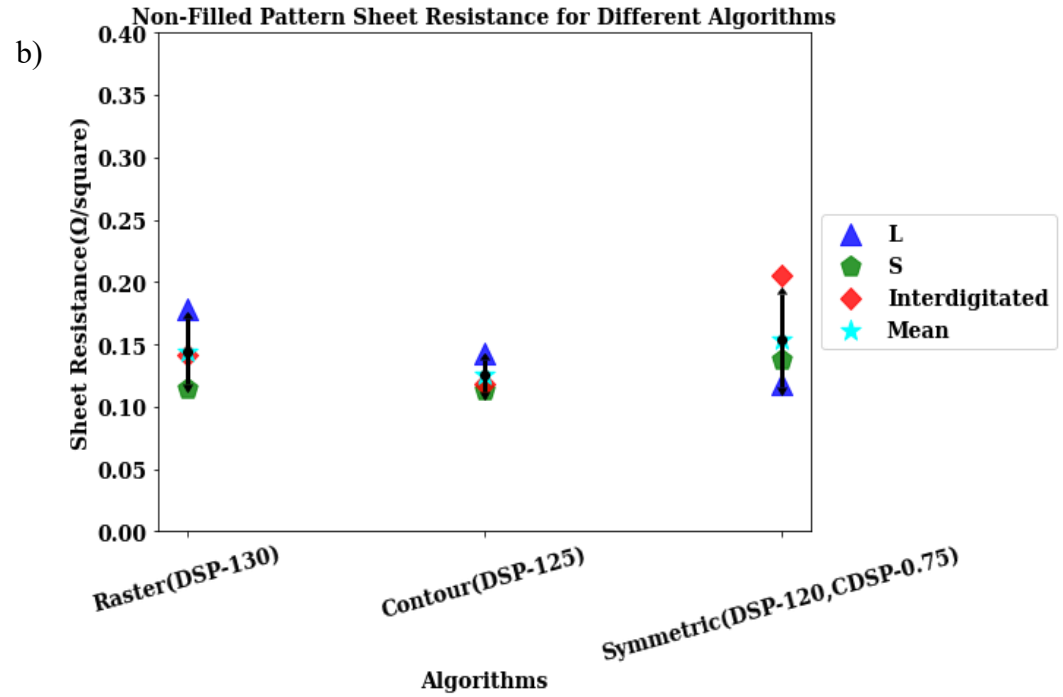
Type	Algorithm	Shape	W/L	R_s (Ω/sq)	Mean R_s Over Patterns (Ω/sq)	Standard Deviation Over Patterns (%)
Filled	Raster (120-DSP)	C	0.123	0.335	0.269	3.56%
		G	0.100	0.358		
		L	0.164	0.359		
		Rectangle	0.500	0.332		
	Parallel (115-DSP)	C	0.145	0.310	0.332	3.61%
		G	0.112	0.386		
		L	0.172	0.322		
		Rectangle	0.451	0.312		
	Radial (115-DSP)	C	0.337	0.311	0.316	3.45%
		G	0.265	0.365		
		L	0.221	0.285		
		Rectangle	0.452	0.303		
	4x4 Matrix Contour (110-DSP)	C	0.298	0.288	0.388	2.46%
		G	0.212	0.306		
		L	0.281	0.225		
		Rectangle	0.470	0.258		
	Symmetric (105-DSP)	C	0.160	0.340	0.335	1.99%
		G	0.123	0.352		
		L	0.230	0.306		
		Rectangle	0.435	0.340		
Non-Filled	Raster (130-DSP)	L	0.097	0.178	0.145	3.19%
		S	0.055	0.114		
		Interdigitated	0.059	0.141		
	Contour (125-DSP)	L	0.103	0.143	0.126	1.61%
		S	0.049	0.113		
		Interdigitated	0.080	0.119		
	Symmetric (115-DSP)	L	0.085	0.118	0.1543	4.59%
		S	0.062	0.138		
		Interdigitated	0.073	0.205		

The results from Table 4-2 can be analyzed further, as shown in Figure 4-8(a)-(c). Figure 4-8(a) states that the sheet-resistance is the same for all the print shapes irrespective of the printing profile. However, the standard deviation of sheet resistance measurement varies. For the G shape, variation is the most. It might come from a lack of pattern symmetry. Again, C, L deviation is also noticeable. The overall mean sheet resistance between the patterning methods are pretty similar. Interdigitated pattern measured resistance has the largest difference from the mean value, as shown

in Figure 4-8(b). It comes from lots of edges and corners of the pattern. As the shape gets complex, the standard deviation increases as well. Figure 4-8(c) shows that filled and non-filled patterns exhibit the same mean sheet resistance. The non-filled pattern measurement standard deviation is lower than the filled case as expected. The overall standard deviation might arise from non-symmetric approximation and process parameter variation, such as annealing temperature variation. Substrate property is the same throughout the experiments, so the variation cannot be attributed to substrate roughness or porosity, although there could be experimental variations in cleaning.

a)





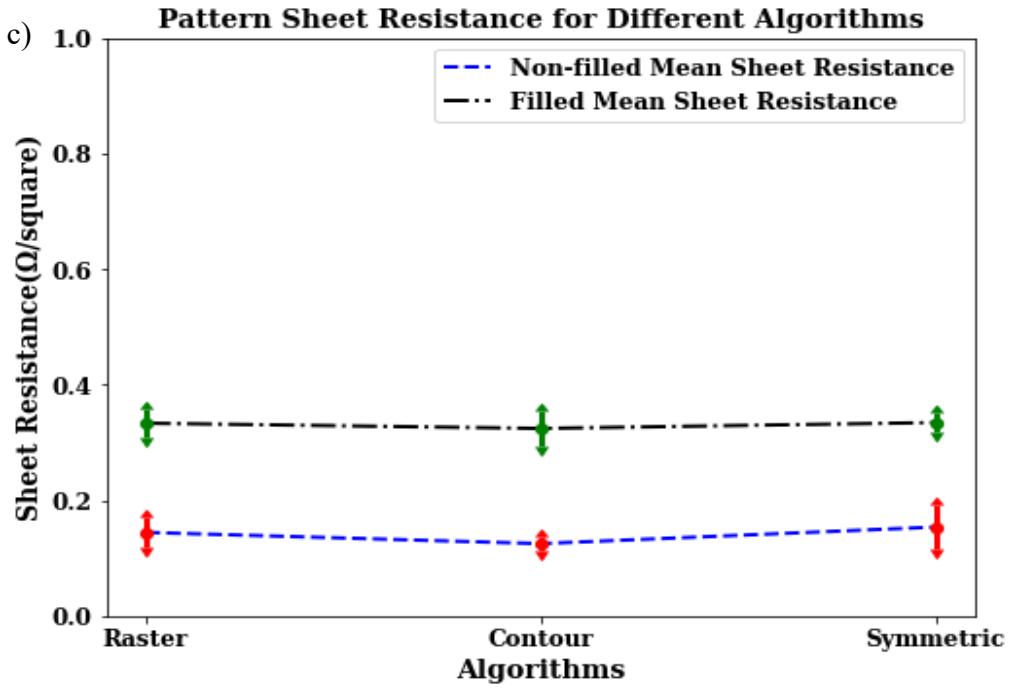


Figure 4-8: (a) Sheet resistance of filled patterns for different algorithms. As an optimal pattern, DSP decreases from raster to symmetric, standard deviation rises over complex shapes. (b) The sheet resistance of non-filled patterns for different algorithms. Though optimal pattern DSP decreases from raster to symmetric, the sheet resistance is similar for the patterns. (c) The sheet resistance of both filled and non-filled patterns for different algorithms. Optimal pattern DSP

decreases from raster to symmetric; sheet resistance is almost the same for both cases. The filled case is susceptible to a higher standard deviation than the non-filled.

Finally, an independent two-sample t-test has been carried on to conclude whether the sheet resistance varies among the filled and non-filled groups printed with raster, contour, symmetric vectorization techniques. With 95% confidence interval, the t-score for raster printing is 0.0894 with p-value 0.93> α=0.05. This means the null hypothesis can't be rejected, and we can say there is no difference between sheet resistances of filled and non-filled groups while printed with raster sequencing. Similar results are observed for contour and symmetric printing as well. Filled parallel, radial results are averaged to compare against the non-filled contour printing. With the same confidence interval p-value of 0.81, 0.91 are obtained for contour and symmetric algorithms. Again, a t-test has been carried on for filled groups printed with different algorithms and non-filled cases. For all the cases, the p-value is found greater than α. We can conclude that the sheet resistance is the same irrespective of sequencing techniques and pattern types.

4.4. Conclusion

We observed that for filled patterns, matrix vectorization combined with clockwise segmentation generates the most promising results. Radial and parallel printing is the next adoptable pattern generation schema. Patterns printed with this method exhibits almost zero bulging and better corners. The same patterns have been thermally annealed, and sheet resistance, surface topology have been characterized. Meaningful surface roughness variation has been observed among different pattern generation schemas. Contour-based filling overall gives a smoother edge profile and less abrupt thickness variation. For the non-filled cases, a similar surface topology trend has been witnessed. Throughout the work, patterns are sintered at 180°C as the used silver ink has been characterized to conduct at this temperature with a sheet resistance value of 0.2-0.4 Ω/sq. Whatever the pattern sequencing plan, the sheet resistance maintained a constant value throughout all shapes and types. To conclude, this is an excellent observation that patterning techniques do not change the ink material's inherent conductive quality.

5. Chapter 5: Conclusion and Future Works

In this chapter, a summary of all the conclusions from different chapters is provided. Then, future work and an outlook are suggested based on the works in this thesis.

5.1. Discussions and Conclusions

The first chapter of this thesis gives an overview of the theory and algorithms used in the subsequent chapters.

In the second chapter of this thesis, machine learning is explored as an efficient method of electronics print parameter optimization and dynamic jetting window forecasting. Prediction of drop behavior before the experimental printing can reduce printing work. Different ensemble approaches are presented as a promising method of foretelling drop speed, radius, and category. Contribution of this prediction module along with the limitations are discussed as follows-

- In contrast to the recent work [134] that has used a similar predictive modeling technique to forecast drop speed, the volume of a single material (using three features- voltage, pulse duration, rise time), we have made our model more robust over twelve materials and eleven features in total. Previous works have reported drop formulation simulation results in terms of the single printer and either in terms of a unipolar or bipolar waveform. Our data expand over multiple printers, as well as unipolar and bipolar signal type. We have seen overall voltage, pulse duration, nozzle size, frequency exerts more influence on the velocity and radius output of a material.
- Weighted averaging of gradient boosting, random forest models generate velocity prediction results with an RMSE of 0.3163m/s, and in the case of radius prediction, 2.91μm. The neural network model categorizes drop with an overall accuracy of 91.84%.
- Our contribution in this context is that we demonstrated a more comprehensive machine learning approach that considers various ink and printer parameters to predict jetting categories. Not only that we have developed a more precise, cost, and time-saving jetting window in terms of predicted velocity, as we have shown, the traditional jetting window boundaries might not fit all experimental data.
- One limitation of this drop behavior prediction is that jetting accuracy is 91.84%, which can be improved further with increased data. Jetting classification data comes from our lab data only, as studied literature didn't report any class category data. We have seen the majority of the

misclassification comes from the single drop class. If the data set is augmented, the jetting prediction output will be more interpretable in terms of feature to output relationships. Another constraint is that a measurement error associated with the target drop radius data. Because the data has been collected in image format using a camera (EZ-grabber version 3, 720×480 pixels resolution), one pixel is equivalent to less than one μm, so the diameter measurement error is ± 5 % [141].

The third chapter discusses different drop sequence generation methods for optimizing pattern printing. The pros and cons of five major algorithms- raster, contour, symmetric, matrix, contour-matrix- have been analyzed in terms of the generated pattern's accuracy evaluation. For filled shapes, contour-based algorithms- matrix inside, radial, parallel brings promising improvement with the highest evaluation accuracy of 94.07%. Prominent contributions and significant barriers to the vision-based printing module are-

- Significant works have been conducted on pattern non-idealities, however, most of the experiments are conducted on simple 2D designs [75][79]-[84] such as lines or squares with manual optimized drop sequence design and not implemented for automated industrial manufacturing. Promising results have been published by our group for sequential design and pattern splitting [78]. We have contributed with a new image feature based pattern generation and non-contact precision, recall, accuracy based pattern evaluation method.
- Contour-based vectorising brings in pattern homogeneity with an enhanced edge profile. In the case of a small scale, contour-based radial, parallel, matrix patterns lose a little junction sharpness where the notch dimension is low. However, with increased block size and pattern dimension, quality is improved with contour-matrix vectorization. For devices that can compromise in terms of edge smoothness but require strict corner localization, symmetric or matrix printing is a better choice. In a word, as the pattern scale size increases, segmentation combined with contour-based block vectorization yields better printing.
- Non-filled patterns are very compatible with symmetric printing while printed on a large scale and patterns with fewer intersections. Smaller-scale designs with a higher number of corners are more adapted to the contour method.
- A combination of multiple vectorizing methods develops filled pattern homogeneity. Graph-based and skeletonization approach are shown to distinguish successfully between different kind of pattern defects.

- The resolution of the nozzle plays a significant impact on drop sequencing. While planning any inkjet pattern, the designer should balance minimum design width, spacing, notch with nozzle diameter, optimum drop spacing for the ink material, and the substrate [198]. We have demonstrated the optimized patterns in 60µm resolution (nozzle size) and optimized drop spacing between (105 µm -130 µm). We have shown 1 DSP as minimum design width (Non-filled Line Shape), 4 DSP for the minimum notch (G Shape), 3DSP for minimum spacing (Monopole Antenna) for silver ink and glass substrate. If the designed dimensions are lower and the nozzle diameter is higher, the printing method might fail irrespective of the best-suited sequencing method. The design has to be carried on carefully considering optimum drop-spacing and nozzle resolution.
- Vision-based pattern sequence generation and evaluation solely rely on the designed image as well as the captured printed pattern image quality. The camera's adjustment (brightness, contrast, sharpness) regarding the printing material and substrate background is crucial during printed image capture. Lower pixel resolution will degrade the performance of the image processing and segmenting algorithms. On the whole generation and evaluation, performance will lessen.

Finally, the fourth chapter introduces sheet resistance characterization, and surface profilometry results are presented as a supporting evaluation of the vision-based pattern generation schema. Overall, patterning techniques don’t change the sheet resistance. It only changes the surface topology. A homogeneous pattern gives less surface height deviation.

- Before sheet resistance and surface profiling, the pattern needs to be sintered at the same temperature so that they are comparable. As we are sintering at a high temperature, two-step sintering should be carried on to avoid the coffee-ring effect coming that can happen due to fast solvent evaporation. This effect can lead to a significant deviation in surface topology along different directions.
- While using the probe station, probing of the four needles should be done carefully so that ink material doesn’t come out with the needle head. Besides, the probe’s contact needs to be cross-checked from the observation of the I-V curve so that there is no open circuit. For filled patterns Vander Paw method, needles should be placed on the edge. The material loss may vary sheet resistance, surface roughness measurement.

In a nutshell, our understanding of the drop generation in terms of machine and material parameters leads to a more generic and diverse choice of materials and machine property. Pattern sequencing, segmentation and evaluation realization open up the opportunity to automatically optimize complex and layered printing.

5.2. Future Work

A couple of interesting future research works can be conducted based on this thesis's findings. First of all, we have implemented and investigated ensemble learning for drop modulation forecasting. The collected data can be further augmented by inserting additional materials. Not only that, we have mainly worked with a nozzle size of 60 µm. Data with some more nozzle sizes and other electrical wave shapes such as sinusoidal, triangular would make the prediction more generic and take the prediction output to higher accuracy. The output from this ML data model can be merged with the vision-based pattern evaluation. With optimal drop spacing, contact angle data for some more ink material and substrate combinations for different types of pattern optimization techniques can be converted to a comprehensive vision and learning-based model that would predict optimal drop spacing and preferred choice of printing method for any given ground truth design.

Adversarial networks could be built for 2D inkjet printing with data collected on several substrates such as PET, PEN, PI with several inks such as copper, gold as the contact angle varies so that DSP will vary. With the input features, contact angle, DSP, and different printing algorithms, the precision, recall, accuracy output can be predicted using generative adversarial networks which would have some original dataset and also generate synthetic data for newer materials, so that we can know beforehand which range of DSP, contact angle, generation algorithm, and pattern shape will give the best accuracy[199].

6. References

[1] Horn, T.J. and Harrysson, O.L., 2012. Overview of current additive manufacturing technologies and selected applications. Science progress, 95(3), pp.255-282.

[2] Stoyanov, S. and Bailey, C., 2017, May. Machine learning for additive manufacturing of electronics. In 2017 40th international spring seminar on electronics technology (ISSE) (pp. 1-6). IEEE.

[3] ASTM F2792 Standard, "Standard Terminology for Additive Manufacturing Technologies", available at: http://www.astm.org

[4] Josten, M., 2017. BREAK IT DOWN: WHAT IS BINDER JETTING?. [Online] Available at: https://news.pminnovationblog.com/blog/what-is-binder-jetting

[5] Peels, J., 2017. Comparison of Metal 3D Printing — Part Two: Directed Energy Deposition. [Online] Available at: https://3dprint.com/182367/directed-energy-deposition/

[6] Bourell, D. et al., 2017. Materials for additive manufacturing. CIRP Annals - Manufacturing Technology, Volume 66, pp. 659-681.

[7] Ligon, S. C. et al., 2017. Polymers for 3D Printing and Customized Additive Manufacturing. Chemical Reviews, 117(15), pp. 10212-10290.

[8] Varotsis, A. B., 2018. Introduction to Material Jetting 3D Printing. [Online] Available at: https://www.3dhubs.com/knowledge-base/introduction-material-jetting-3d-printing.

[9] Michalik, J., Joyce, J., Barney, R. & McCune, G., 2015. 3D opportunity for product design. [Online] Available at: https://www2.deloitte.com/insights/us/en/focus/3d-opportunity/3d-printing-product-design-and-development.html.

[10] Sun, S., M.Brandt & M.Easton, 2017. Powder bed fusion processes: An overview. Laser Additive Manufacturing, pp. 55-77.

[11] Silbernagel, C., 2018. Additive Manufacturing 101-6: What is sheet lamination?. [Online] Available at: http://canadamakes.ca/what-is-sheet-lamination/.

[12] Cullen, A.T. and Price, A.D., 2019. Fabrication of 3D conjugated polymer structures via vat polymerization additive manufacturing. Smart Materials and Structures, 28(10), p.104007.

[13] McKerricher, G., Vaseem, M. and Shamim, A., 2017. Fully inkjet-printed microwave passive electronics. Microsystems & nanoengineering, 3(1), pp.1-7.

[14] Correia, V., Mitra, K.Y., Castro, H., Rocha, J.G., Sowade, E., Baumann, R.R. and Lanceros-Mendez, S., 2018. Design and fabrication of multilayer inkjet-printed passive components for printed electronics circuit development. Journal of Manufacturing Processes, 31, pp.364-371.

[15] Yasin, T. and Baktur, R., 2010, July. Inkjet printed patch antennas on transparent substrates. In 2010 IEEE Antennas and Propagation Society International Symposium (pp. 1-4). IEEE.

[16] Subramanian, V., Fréchet, J.M., Chang, P.C., Huang, D.C., Lee, J.B., Molesa, S.E., Murphy, A.R., Redinger, D.R. and Volkman, S.K., 2005. Progress toward development of all-printed RFID tags: materials, processes, and devices. Proceedings of the IEEE, 93(7), pp.1330-1338.

[17] Rida, A., Yang, L., Vyas, R. and Tentzeris, M.M., 2009. Conductive inkjet-printed antennas on flexible low-cost paper-based substrates for RFID and WSN applications. IEEE Antennas and Propagation Magazine, 51(3), pp.13-23.

[18] Koo, H.S., Chen, M., Pan, P.C., Chou, L.T., Wu, F.M., Chang, S.J. and Kawai, T., 2006. Fabrication and chromatic characteristics of the greenish LCD colour-filter layer with nano-particle ink using inkjet printing technique. Displays, 27(3), pp.124-129.

[19] Chang, S.C., Liu, J., Bharathan, J., Yang, Y., Onohara, J. and Kido, J., 1999. Multicolor organic light-emitting diodes processed by hybrid inkjet printing. Advanced Materials, 11(9), pp.734-737.

[20] Chung, S., Cho, K. and Lee, T., 2019. Recent progress in inkjet-printed thin-film transistors. Advanced science, 6(6), p.1801445.

[21] Tseng, H.Y. and Subramanian, V., 2011. All inkjet-printed, fully self-aligned transistors for low-cost circuit applications. Organic Electronics, 12(2), pp.249-256.

[22] P. F. Moonen, I. Yakimets, and J. Huskens, “Fabrication of transistors on flexible substrates: From mass-printing to high-resolution alternative lithography strategies,” Adv. Mater., vol. 24, no. 41, pp. 5526–5541, Nov. 2012.

[23] Hodgson, A., 2007. The role of paper in the future of printed electronics. CoPADD 2007, p.5.

[24] MacDonald, W.A., Looney, M.K., MacKerron, D., Eveson, R., Adam, R., Hashimoto, K. and Rakos, K., 2007. Latest advances in substrates for flexible electronics. Journal of the Society for Information Display, 15(12), pp.1075-1083.

[25] He, B., Yang, S., Qin, Z., Wen, B., and Zhang, C., 2017, “The Roles of Wettability and Surface Tension in Droplet Formation During Inkjet Printing,” Sci. Rep., 7(1), p. 11841.

[26] Meyer, S., Pham, D.V., Merkulov, S., Weber, D., Merkulov, A., Benson, N. and Schmechel, R., 2017. Soluble metal oxo alkoxide inks with advanced rheological properties for inkjet-printed thin-film transistors. ACS applied materials & interfaces, 9(3), pp.2625-2633.

[27] Nallan, H.C., Sadie, J.A., Kitsomboonloha, R., Volkman, S.K. and Subramanian, V., 2014. Systematic design of jettable nanoparticle-based inkjet inks: Rheology, acoustics, and jettability. Langmuir, 30(44), pp.13470-13477.

[28] Kim, E. and Baek, J., 2012. Numerical study on the effects of non-dimensional parameters on drop-on-demand droplet formation dynamics and printability range in the up-scaled model. Physics of Fluids, 24(8), p.082103.

[29] Barton, K., Mishra, S., Shorter, K.A., Alleyne, A., Ferreira, P. and Rogers, J., 2010. A desktop electrohydrodynamic jet printing system. Mechatronics, 20(5), pp.611-616.

[30] Daly, R., Harrington, T.S., Martin, G.D. and Hutchings, I.M., 2015. Inkjet printing for pharmaceutics–a review of research and manufacturing. International journal of pharmaceutics, 494(2), pp.554-567.

[31] Le, H.P., 1998. Progress and trends in inkjet printing technology. Journal of Imaging Science and Technology, 42(1), pp.49-62.

[32] Martin, G.D., Hoath, S.D. and Hutchings, I.M., 2008, March. Inkjet printing-the physics of manipulating liquid jets and drops. In J. Phys.: Conf. Ser (Vol. 105, No. 1, p. 012001).

[33] Strutt, J.W. and Rayleigh, L., 1878. On the instability of jets. Proc. London Math. Soc, 10(4).

[34] "Rotech Machines", Rotechmachines.com, 2020. [Online]. Available: https://www.rotechmachines.com/news/cij-vs-tij/2018/08/20/cij-vs-tij. [Accessed: 31- Aug- 2020]

[35] Hutchings, I.M. and Martin, G.D. eds., 2012. Inkjet technology for digital fabrication. John Wiley & Sons.

[36] Wijshoff, H., 2006, January. Manipulating drop formation in piezo acoustic inkjet. In NIP & Digital Fabrication Conference (Vol. 2006, No. 1, pp. 79-82). Society for Imaging Science and Technology.

[37] Bogy, D.B. and Talke, F.E., 1984. Experimental and theoretical study of wave propagation phenomena in drop-on-demand ink jet devices. IBM Journal of research and development, 28(3), pp.314-321.

[38] Hoath, S. D., Hsiao, W.-K., Jung, S., Martin, G. D., Hutchings, I. M., Morrison,N. F., and Harlen, O. G., 2013, “Drop Speeds From Drop-on-Demand Ink-Jet Print Heads,” J. Imaging Sci. Technol., 57(1), pp. 1–11.

[39] Nallan, H.C., Sadie, J.A., Kitsomboonloha, R., Volkman, S.K. and Subramanian, V., 2014. Systematic design of jettable nanoparticle-based inkjet inks: Rheology, acoustics, and jettability. Langmuir, 30(44), pp.13470-13477.

[40] Liu, Y.-F., Tsai, M.-H., Pai, Y.-F., and Hwang, W.-S., 2013, “Control of Droplet Formation by Operating Waveform for Inks With Various Viscosities in Piezoelectric Inkjet Printing,” Appl. Phys. A, 111(2), pp. 509–516.

[41] Liu, Y., and Derby, B., 2019, “Experimental Study of the Parameters for Stable Drop-on-Demand Ink-jet Performance,” Phys. Fluids, 31(3), p. 032004.

[42] Lai, J.-M., Huang, C.-Y., Chen, C.-H., Linliu, K., and Lin, J.-D., 2010, “Influence of Liquid Hydrophobicity and Nozzle Passage Curvature on Microfluidic Dynamics in a Drop Ejection Process,” J. Micromech. Microeng., 20(1), p. 015033.

[43] Reis, N., Ainsley, C., and Derby, B., 2005, “Ink-Jet Delivery of Particle Suspensions by Piezoelectric Droplet Ejectors,” J. Appl. Phys., 97(9), p. 094903.

[44] Yang, Q., Li, H., Li, M., Li, Y., Chen, S., Bao, B., and Song, Y., 2017, “Rayleigh Instability-Assisted Satellite Droplets Elimination in Inkjet Printing,” ACS Appl. Mater. Interfaces, 9(47), pp. 41521–41528.

[45] Wu, H.-C., Shan, T.-R., Hwang, W.-S., and Lin, H.-J., 2004, “Study of Micro-Droplet Behavior for a Piezoelectric Inkjet Printing Device Using a Single Pulse Voltage Pattern,” Mater. Trans., 45(5), pp. 1794–1801.

[46] Wu, D. and Xu, C., 2018. Predictive modeling of droplet formation processes in inkjet-based bioprinting. Journal of Manufacturing Science and Engineering, 140(10), p.101007.

[47] Alpaydin, E., 2020. Introduction to machine learning. MIT press.

[48] Murdoch, W.J., Singh, C., Kumbier, K., Abbasi-Asl, R. and Yu, B., 2019. Interpretable machine learning: definitions, methods, and applications. arXiv preprint arXiv:1901.04592.

[49] A. Dror, "Additive Manufacturing Trends to Know in the Electronics Industry in 2020", Nano-di.com, 2020. [Online]. Available: https://www.nano-di.com/blog/2020-additive-manufacturing-trends-to-know-in-the-electronics-industry-in-2020. [Accessed: 24- Aug- 2020]

[50] Brownlee, J., 2018. A gentle introduction to k-fold cross-validation. Machine Learning Mastery, 2019.

[51] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn : Machine learning in Python. the Journal of machine Learning research, 12, pp.2825-2830.

[52] Provost, F. and Kohavi, R., 1998. Glossary of terms. Journal of Machine Learning, 30(2-3), pp.271-274.

[53] Chai, T. and Draxler, R.R., 2014. Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. Geoscientific model development, 7(3), pp.1247-1250.

[54] M. Rogati, "The AI Hierarchy of Needs | Hacker Noon", Hackernoon.com, 2020. [Online]. Available: https://hackernoon.com/the-ai-hierarchy-of-needs-18f111fcc007. [Accessed: 24- Aug- 2020]

[55] Dey, A., 2016. Machine learning algorithms: a review. International Journal of Computer Science and Information Technologies, 7(3), pp.1174-1179.

[56] Kotsiantis, S.B., Zaharakis, I.D. and Pintelas, P.E., 2006. Machine learning: a review of classification and combining techniques. Artificial Intelligence Review, 26(3), pp.159-190.

[57] Heinl, M., Walde, J., Tappeiner, G. and Tappeiner, U., 2009. Classifiers vs. input variables—The drivers in image classification for land cover mapping. International Journal of Applied Earth Observation and Geoinformation, 11(6), pp.423-430.

[58] J. Brownlee, "A Tour of Machine Learning Algorithms", Machine Learning Mastery, 2020. [Online]. Available: https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/. [Accessed: 22- Sep- 2020]

[59] Hastie, T., Tibshirani, R. and Tibshirani, R.J., 2017. Extended comparisons of best subset selection, forward stepwise selection, and the lasso. arXiv preprint arXiv:1707.08692.

[60] Breiman, L., 2001. Random forests. Machine learning, 45(1), pp.5-32.

[61] Gupta, B., Awasthi, S., Gupta, R., Ram, L., Kumar, P., Prasad, B.R. and Agarwal, S., 2018. Taxi travel time prediction using ensemble-based random forest and gradient boosting model. In Advances in Big Data and Cloud Computing (pp. 63-78). Springer, Singapore.

[62] Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. Annals of statistics, pp.1189-1232.

[63] Kadiyala, A. and Kumar, A., 2018. Applications of python to evaluate the performance of decision tree-based boosting algorithms. Environmental Progress & Sustainable Energy, 37(2), pp.618-623.

[64] Chen, T., He, T., Benesty, M., Khotilovich, V. and Tang, Y., 2015. Xgboost: extreme gradient boosting. R package version 0.4-2, pp.1-4.

[65] Ogunleye, A.A. and Qing-Guo, W., 2019. XGBoost model for chronic kidney disease diagnosis. IEEE/ACM transactions on computational biology and bioinformatics.

[66] McKerricher, G., Vaseem, M. and Shamim, A., 2017. Fully inkjet-printed microwave passive electronics. Microsystems & nanoengineering, 3(1), pp.1-7.

[67] Correia, V., Mitra, K.Y., Castro, H., Rocha, J.G., Sowade, E., Baumann, R.R. and Lanceros-Mendez, S., 2018. Design and fabrication of multilayer inkjet-printed passive components for printed electronics circuit development. Journal of Manufacturing Processes, 31, pp.364-371.

[68] Yasin, T. and Baktur, R., 2010, July. Inkjet printed patch antennas on transparent substrates. In 2010 IEEE Antennas and Propagation Society International Symposium (pp. 1-4). IEEE.

[69] Subramanian, V., Fréchet, J.M., Chang, P.C., Huang, D.C., Lee, J.B., Molesa, S.E., Murphy, A.R., Redinger, D.R. and Volkman, S.K., 2005. Progress toward development of all-printed RFID tags: materials, processes, and devices. Proceedings of the IEEE, 93(7), pp.1330-1338.

[70] Rida, A., Yang, L., Vyas, R. and Tentzeris, M.M., 2009. Conductive inkjet-printed antennas on flexible low-cost paper-based substrates for RFID and WSN applications. IEEE Antennas and Propagation Magazine, 51(3), pp.13-23.

[71] Koo, H.S., Chen, M., Pan, P.C., Chou, L.T., Wu, F.M., Chang, S.J. and Kawai, T., 2006. Fabrication and chromatic characteristics of the greenish LCD colour-filter layer with nano-particle ink using inkjet printing technique. Displays, 27(3), pp.124-129.

[72] Chang, S.C., Liu, J., Bharathan, J., Yang, Y., Onohara, J. and Kido, J., 1999. Multicolor organic light-emitting diodes processed by hybrid inkjet printing. Advanced Materials, 11(9), pp.734-737.

[73] Chung, S., Cho, K. and Lee, T., 2019. Recent progress in inkjet-printed thin-film transistors. Advanced science, 6(6), p.1801445.

[74] P. F. Moonen, I. Yakimets, and J. Huskens, “Fabrication of transistors on flexible substrates: From mass-printing to high-resolution alternative lithography strategies,” Adv. Mater., vol. 24, no. 41, pp. 5526–5541, Nov. 2012.

[75] Soltman, D., and Subramanian, V., 2008, “Inkjet-Printed Line Morphologies and Temperature Control of the Coffee Ring Effect,” Langmuir, 24(5), pp. 2224–2231.

[76] Stucchi, M., Bamal, M. and Maex, K., 2007. Impact of line-edge roughness on resistance and capacitance of scaled interconnects. Microelectronic Engineering, 84(11), pp.2733-2737.

[77] Sowade, E., Polomoshnov, M. and Baumann, R.R., 2016. The design challenge in printing devices and circuits: Influence of the orientation of print patterns in inkjet-printed electronics. Organic Electronics, 37, pp.428-438.

[78] R. Abunahla, M. S. Rahman, P. Naderi and G. Grau, “Inkjet Printing Without Bulges: A Segmented and Symmetric Drop Placement Method,” Journal of Micro- and Nano-Manufacturing, 2020. In press.

[79] Duineveld, P. C., 2003, “The Stability of Ink-Jet Printed Lines of Liquid with Zero Receding Contact Angle on a Homogeneous Substrate,” Journal of Fluid Mechanics, 477.

[80] Stringer, J., and Derby, B., 2009, “Limits to Feature Size and Resolution in Ink Jet Printing,” Journal of the European Ceramic Society, 29(5), pp. 913–918.

[81] Stringer, J., and Derby, B., 2010, “Formation and Stability of Lines Produced by Inkjet Printing,” Langmuir, 26(12), pp. 10365–10372.

[82] Diaz, E., Ramon, E., and Carrabina, J., 2013, “Inkjet Patterning of Multiline Intersections for Wirings in Printed Electronics,” Langmuir, 29(40), pp. 12608–12614.

[83] Oh, Y., Kim, J., Yoon, Y. J., Kim, H., Yoon, H. G., Lee, S.-N., and Kim, J., 2011, “Inkjet Printing of Al2O3 Dots, Lines, and Films: From Uniform Dots to Uniform Films,” Current Applied Physics, 11(3), pp. S359–S363.

[84] Hsiao, W.-K., Martin, G. D., and Hutchings, I. M., 2014, “Printing Stable Liquid Tracks on a Surface with Finite Receding Contact Angle,” Langmuir, 30(41), pp. 12447–12455.

[85] Kang, H., Soltman, D., and Subramanian, V., 2010, “Hydrostatic Optimization of Inkjet-Printed Films,” Langmuir, 26(13), pp. 11568–11573.

[86] Soltman, D., Smith, B., Morris, S. J. S., and Subramanian, V., 2013, “Inkjet Printing of Precisely Defined Features Using Contact-Angle Hysteresis,” Journal of Colloid and Interface Science, 400, pp. 135–139.

[87] Du, Z., Xing, R., Cao, X., Yu, X., and Han, Y., 2017, “Symmetric and Uniform Coalescence of Ink-Jetting Printed Polyfluorene Ink Drops by Controlling the Droplet Spacing Distance and Ink Surface Tension/Viscosity Ratio,” Polymer, 115, pp. 45–51.

[88] Soltman, D., Smith, B., Kang, H., Morris, S. J. S., and Subramanian, V., 2010, “Methodology for Inkjet Printing of Partially Wetting Films,” Langmuir, 26(19), pp. 15686–15693.

[89] Chen, C.-T., and Hung, T.-Y., 2016, “Morphology and Deposit of Picoliter Droplet Tracks Generated by Inkjet Printing,” Journal of Micromechanics and Microengineering, 26(11), p. 115005.

[90] Vila, F., Pallares, J., Ramon, E., and Teres, L., 2016, “A Systematic Study of Pattern Compensation Methods for All-Inkjet Printing Processes,” IEEE Transactions on Components, Packaging and Manufacturing Technology, 6(4), pp. 630–636.

[91] Ito, M. and Nikaido, Y., 1991, September. Recognition of pattern defects of printed circuit board using topological information. In [1991 Proceedings] Eleventh IEEE/CHMT International Electronics Manufacturing Technology Symposium (pp. 202-206). IEEE.

[92] Rau, H. and Wu, C.H., 2005. Automatic optical inspection for detecting defects on printed circuit board inner layers. The International Journal of Advanced Manufacturing Technology, 25(9-10), pp.940-946.

[93] Ito, M., Nikaido, Y. and Hoshino, M., 1992, September. Divide and Merge Image Processing For Pattern Defect Analysis of Printed Circuit Boards. In Thirteenth IEEE/CHMT International Electronics Manufacturing Technology Symposium (pp. 178-182). IEEE.

[94] A. Koschan and M. Abidi. “Detection and Classification of Edges in Color Images”. IEEE Signal Processing, 22(1):64–73, January 2005.

[95] Savant, Shubhashree. "A review on edge detection techniques for image segmentation." International Journal of Computer Science and Information Technologies 5, no. 4 (2014): 5898-5900.

[96] Gong, X.Y., Su, H., Xu, D., Zhang, Z.T., Shen, F. and Yang, H.B., 2018. An overview of contour detection approaches. International Journal of Automation and Computing, 15(6), pp.656-672.

[97] Eriksson, A.P., Barr, O. and Astrom, K., 2006. Image segmentation using minimal graph cuts.

[98] Uzkent, B., Hoffman, M.J., Cherry, E. and Cahill, N., 2014, November. 3-d mri cardiac segmentation using graph cuts. In 2014 IEEE Western New York Image and Signal Processing Workshop (WNYISPW) (pp. 47-51). IEEE.

[99] Zhao, Z.-Q., Zheng, P., Xu, S.-t. & Wu, X. Object detection with deep learning: a review. IEEE Transactions on Neural Networks and Learning Systems 30, 3212–3232 (2019).

[100] Ren, S., He, K., Girshick, R. & Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. Advances in Neural Information Pro- cessing Systems, 91–99 (Neural Information Processing Systems Foundation, 2015).

[101] Girshick, R. Fast R-CNN. Proceedings of the IEEE International Conference on Computer Vision, 1440–1448 (IEEE, 2015).

[102] Girshick, R., Donahue, J., Darrell, T. & Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 580–587 (IEEE, 2014).

[103] Liu, W. et al. SSD: Single shot multibox detector. European Conference on Com- puter Vision, 21–37 (Springer, 2016).

[104] Garcia-Garcia, A., Orts-Escolano, S., Oprea, S. O., Villena-Martinez, V. & Garcia-Rodriguez, J. A review on deep learning techniques applied to semantic seg- mentation. Preprint at https://arxiv.org/abs/1704.06857 (2017).

[105] Badrinarayanan, V., Kendall, A. & Cipolla, R. SegNet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39, 2481–2495 (2017).

[106] Ronneberger, O., Fischer, P. & Brox, T. U-Net: convolutional networks for bio- medical image segmentation. International Conference on Medical Image Com- puting and Computer-assisted Intervention, 234–241 (Springer, 2015).

[107] Long, J., Shelhamer, E. & Darrell, T. Fully convolutional networks for semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3431–3440 (IEEE, 2015).

[108] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. & Yuille, A. L. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. 40, 834–848 (2017).

[109] He, K., Gkioxari, G., Dollár, P. & Girshick, R. Mask R-CNN. Proceedings of the IEEE International Conference on Computer Vision, 2961–2969 (IEEE, 2017).

[110] Goodfellow, I. et al. Generative adversarial nets. Advances in Neural Information Processing Systems, 2672–2680 (Neural Information Processing Systems Founda- tion, 2014).

[111] Goodfellow, I., Bengio, Y. & Courville, A. Deep Learning. (MIT Press, 2016).

[112] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. Nature 521, 436–444 (2015).

[113] Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 1097–1105 (Neural Information Processing Systems Foundation, 2012).

[114] Litjens, G. et al. A survey on deep learning in medical image analysis. Med. Image Anal. 42, 60–88 (2017).

[115] Falk, T. et al. U-Net: deep learning for cell counting, detection, and morphometry. Nat. Methods 16, 67–70 (2019).

[116] Moen, E. et al. Deep learning for cellular image analysis. Nat. Methods https://doi. org/10.1038/s41592-019-0403-1 (2019).

[117] Karpathy, A. Software 2.0. https://medium.com/@karpathy/software-2-0- a64152b37c35 (2017).

[118] Novoselov, K. S., Mishchenko, A., Carvalho, A. & Castro Neto, A. H. 2D materials and van der Waals heterostructures. Science 353, aac9439 (2016).

[119] Novoselov, K. S. et al. Two-dimensional atomic crystals. Proc. Natl Acad. Sci. USA102, 10451–10453 (2005).

[120] Novoselov, K. S. et al. Electric field effect in atomically thin carbon films. Science 306, 666–669 (2004).

[121] Tseng, H.Y. and Subramanian, V., 2011. All inkjet-printed, fully self-aligned transistors for low-cost circuit applications. Organic Electronics, 12(2), pp.249-256.

[122] Subramanian, V., Fréchet, J.M., Chang, P.C., Huang, D.C., Lee, J.B., Molesa, S.E., Murphy, A.R., Redinger, D.R. and Volkman, S.K., 2005. Progress toward development of all-printed RFID tags: materials, processes, and devices. Proceedings of the IEEE, 93(7), pp.1330-1338.

[123] McKerricher, G., Perez, J.G. and Shamim, A., 2015. Fully inkjet printed RF inductors and capacitors using polymer dielectric and silver conductive ink with through vias. IEEE Transactions on Electron Devices, 62(3), pp.1002-1009.

[124] Hoath, S. D., Hsiao, W.-K., Jung, S., Martin, G. D., Hutchings, I. M., Morrison,N. F., and Harlen, O. G., 2013, "Drop Speeds From Drop-on-Demand Ink-Jet Print Heads," J. Imaging Sci. Technol., 57(1), pp. 1–11.

[125] Nallan, H. C., Sadie, J. A., Kitsomboonloha, R., Volkman, S. K., and Subramanian,V., 2014, "Systematic Design of Jettable Nanoparticle-Based Inkjet Inks: Rheology, Acoustics, and Jettability," Langmuir, 30(44), pp. 13470–13477.

[126] Liu, Y.-F., Tsai, M.-H., Pai, Y.-F., and Hwang, W.-S., 2013, "Control of Droplet Formation by Operating Waveform for Inks With Various Viscosities in Piezoelectric Inkjet Printing," Appl. Phys. A, 111(2), pp. 509–516.

[127] Liu, Y., and Derby, B., 2019, "Experimental Study of the Parameters for Stable Drop-on-Demand Ink-jet Performance," Phys. Fluids, 31(3), p. 032004.

[128] Lai, J.-M., Huang, C.-Y., Chen, C.-H., Linliu, K., and Lin, J.-D., 2010, "Influence of Liquid Hydrophobicity and Nozzle Passage Curvature on Microfluidic Dynamics in a Drop Ejection Process," J. Micromech. Microeng., 20(1), p. 015033.

[129] Reis, N., Ainsley, C., and Derby, B., 2005, "Ink-Jet Delivery of Particle Suspensions by Piezoelectric Droplet Ejectors," J. Appl. Phys., 97(9), p. 094903.

[130] Yang, Q., Li, H., Li, M., Li, Y., Chen, S., Bao, B., and Song, Y., 2017, "Rayleigh Instability-Assisted Satellite Droplets Elimination in Inkjet Printing," ACS Appl. Mater. Interfaces, 9(47), pp. 41521–41528.

[131] He, B., Yang, S., Qin, Z., Wen, B., and Zhang, C., 2017, "The Roles of Wettability and Surface Tension in Droplet Formation During Inkjet Printing," Sci. Rep., 7(1), p. 11841.

[132] Kim, E. and Baek, J., 2012. Numerical study on the effects of non-dimensional parameters on drop-on-demand droplet formation dynamics and printability range in the up-scaled model. Physics of Fluids, 24(8), p.082103.

[133] Nallan, H.C., Sadie, J.A., Kitsomboonloha, R., Volkman, S.K. and Subramanian, V., 2014. Systematic design of jettable nanoparticle-based inkjet inks: Rheology, acoustics, and jettability. Langmuir, 30(44), pp.13470-13477.

[134] Wu, D. and Xu, C., 2018. Predictive modeling of droplet formation processes in inkjet-based bioprinting. Journal of Manufacturing Science and Engineering, 140(10), p.101007.

[135] F.Brishty,"Machine-Learning-Based-Data-Driven-Approach-for-Optimized-Inkjet-Printed-Electronics-Codes", github.com, 2020. [Online]. Available: https://github.com/fahmida185/Machine-Learning-Based-Data-Driven-Approach-for-Optimized-Inkjet-Printed- Electronics/tree/master/Final%20Codes. [Accessed: 27- Aug- 2020]

[136] Microfab technote, Fluid Properties Effects on Ink-Jet Device Performance, 99-02, 1999.

[137] Jpmtr.org,2020.[Online].Available:http://jpmtr.org/iarigai_conference/3A1_Scattareggia_Fujifilm%20Dimatix-%20Scattareggia-%20Iarigai%202015.pdf. [Accessed: 27- Aug- 2020]

[138] Microfab technote, Drive waveform effects on inkjet device performance, 99-03, 1999.

[139] Microfab technote, Orifice Diameter Effects, 99-04, 1999.

[140] "WebPlotDigitizer - Extract data from plots, images, and maps", Automeris.io, 2020. [Online]. Available: https://automeris.io/WebPlotDigitizer. [Accessed: 27- Aug- 2020]

[141] Equipment – Ink-jet – Imaging-based measurement of droplet volume, Printed electronics – Part 302-2.

[142] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn : Machine learning in Python. the Journal of machine Learning research, 12, pp.2825-2830.

[143] Breiman, L., 2001. Random forests. Machine learning, 45(1), pp.5-32.

[144] Hastie, T., Tibshirani, R. and Tibshirani, R.J., 2017. Extended comparisons of best subset selection, forward stepwise selection, and the lasso. arXiv preprint arXiv:1707.08692.

[145] Aslam, J.A., Popa, R.A. and Rivest, R.L., 2008, July. On Auditing Elections When Precincts Have Different Sizes. In EVT.

[146] Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. Annals of statistics, pp.1189-1232.

[147] "1.1. Linear Models — Scikit-learn 0.23.2 documentation", Scikit-learn .org, 2020. [Online]. Available: https://Scikit-learn .org/stable/modules/linear_model.html. [Accessed: 27- Aug- 2020]

[148] Duineveld, P.C., De Kok, M.M., Buechel, M., Sempel, A., Mutsaers, K.A., Van de Weijer, P., Camps, I.G., Van de Biggelaar, T., Rubingh, J.E.J. and Haskal, E.I., 2002,

February. Inkjet printing of polymer light-emitting devices. In Organic Light-Emitting Materials and Devices V (Vol. 4464, pp. 59-67). International Society for Optics and Photonics.

[149] Brownlee, J., 2018. A gentle introduction to k-fold cross-validation. Machine Learning Mastery, 2019.

[150] Sill, J., Takács, G., Mackey, L. and Lin, D., 2009. Feature-weighted linear stacking. arXiv preprint arXiv:0911.0460.

[151] Hwang, J.Y., Kim, M.K., Lee, S.H., Kang, K., Kang, H. and Cho, Y.J., 2007, October. A study on driving waveform of a piezoelectric inkjet print head. In Optomechatronic Micro/Nano Devices and Components III (Vol. 6717, p. 67170T). International Society for Optics and Photonics.

[152] Tsai, M.H., Hwang, W.S., Chou, H.H. and Hsieh, P.H., 2008. Effects of pulse voltage on inkjet printing of a silver nanopowder suspension. Nanotechnology, 19(33), p.335304.

[153] Molnar, C., 2019. Interpretable machine learning. Lulu. com.

[154] P. F. Moonen, I. Yakimets, and J. Huskens, "Fabrication of transistors on flexible substrates: From mass-printing to high-resolution alternative lithography strategies," Adv. Mater., vol. 24, no. 41, pp. 5526–5541, Nov. 2012.

[155] Soltman, D., and Subramanian, V., 2008, "Inkjet-Printed Line Morphologies and Temperature Control of the Coffee Ring Effect," Langmuir, 24(5), pp. 2224–2231.

[156] Stucchi, M., Bamal, M. and Maex, K., 2007. Impact of line-edge roughness on resistance and capacitance of scaled interconnects. Microelectronic Engineering, 84(11), pp.2733-2737.

[157] Sowade, E., Polomoshnov, M. and Baumann, R.R., 2016. The design challenge in printing devices and circuits: Influence of the orientation of print patterns in inkjet-printed electronics. Organic Electronics, 37, pp.428-438.

[158] Papari, G. and Petkov, N., 2011. Edge and line oriented contour detection: State of the art. Image and Vision Computing, 29(2-3), pp.79-103.

[159] Iverson, L.A. and Zucker, S.W., 1995. Logical/linear operators for image curves. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(10), pp.982-996.

[160] R. Abunahla, M. S. Rahman, P. Naderi and G. Grau, "Inkjet Printing Without Bulges: A Segmented and Symmetric Drop Placement Method," Journal of Micro- and Nano-Manufacturing, 2020. In press.

[161] Duineveld, P. C., 2003, “The Stability of Ink-Jet Printed Lines of Liquid with Zero Receding Contact Angle on a Homogeneous Substrate,” Journal of Fluid Mechanics, 477.

[162] Stringer, J., and Derby, B., 2009, “Limits to Feature Size and Resolution in Ink Jet Printing,” Journal of the European Ceramic Society, 29(5), pp. 913–918.

[163] Stringer, J., and Derby, B., 2010, “Formation and Stability of Lines Produced by Inkjet Printing,” Langmuir, 26(12), pp. 10365–10372.

[164] Diaz, E., Ramon, E., and Carrabina, J., 2013, “Inkjet Patterning of Multiline Intersections for Wirings in Printed Electronics,” Langmuir, 29(40), pp. 12608–12614.

[165] Oh, Y., Kim, J., Yoon, Y. J., Kim, H., Yoon, H. G., Lee, S.-N., and Kim, J., 2011, “Inkjet Printing of Al2O3 Dots, Lines, and Films: From Uniform Dots to Uniform Films,” Current Applied Physics, 11(3), pp. S359–S363.

[166] Hsiao, W.-K., Martin, G. D., and Hutchings, I. M., 2014, “Printing Stable Liquid Tracks on a Surface with Finite Receding Contact Angle,” Langmuir, 30(41), pp. 12447–12455.

[167] Kang, H., Soltman, D., and Subramanian, V., 2010, “Hydrostatic Optimization of Inkjet-Printed Films,” Langmuir, 26(13), pp. 11568–11573.

[168] Soltman, D., Smith, B., Morris, S. J. S., and Subramanian, V., 2013, “Inkjet Printing of Precisely Defined Features Using Contact-Angle Hysteresis,” Journal of Colloid and Interface Science, 400, pp. 135–139.

[169] Du, Z., Xing, R., Cao, X., Yu, X., and Han, Y., 2017, “Symmetric and Uniform Coalescence of Ink-Jetting Printed Polyfluorene Ink Drops by Controlling the Droplet Spacing Distance and Ink Surface Tension/Viscosity Ratio,” Polymer, 115, pp. 45–51.

[170] Soltman, D., Smith, B., Kang, H., Morris, S. J. S., and Subramanian, V., 2010, “Methodology for Inkjet Printing of Partially Wetting Films,” Langmuir, 26(19), pp. 15686–15693.

[171] Chen, C.-T., and Hung, T.-Y., 2016, “Morphology and Deposit of Picoliter Droplet Tracks Generated by Inkjet Printing,” Journal of Micromechanics and Microengineering, 26(11), p. 115005.

[172] Vila, F., Pallares, J., Ramon, E., and Teres, L., 2016, “A Systematic Study of Pattern Compensation Methods for All-Inkjet Printing Processes,” IEEE Transactions on Components, Packaging and Manufacturing Technology, 6(4), pp. 630–636.

[173] Ito, M. and Nikaido, Y., 1991, September. Recognition of pattern defects of printed circuit board using topological information. In [1991 Proceedings] Eleventh IEEE/CHMT International Electronics Manufacturing Technology Symposium (pp. 202-206). IEEE.

[174] Rau, H. and Wu, C.H., 2005. Automatic optical inspection for detecting defects on printed circuit board inner layers. The International Journal of Advanced Manufacturing Technology, 25(9-10), pp.940-946.

[175] Ito, M., Nikaido, Y. and Hoshino, M., 1992, September. Divide and Merge Image Processing For Pattern Defect Analysis of Printed Circuit Boards. In Thirteenth IEEE/CHMT International Electronics Manufacturing Technology Symposium (pp. 178-182). IEEE.

[176] Soltman, D.B., 2011. Understanding inkjet printed pattern generation (Doctoral dissertation, UC Berkeley).

[177] Papari, G. and Petkov, N., 2011. Edge and line oriented contour detection: State of the art. Image and Vision Computing, 29(2-3), pp.79-103.

[178] Iverson, L.A. and Zucker, S.W., 1995. Logical/linear operators for image curves. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(10), pp.982-996.

[179] Jaglan, P., Dass, R. and Duhan, M., 2019. A comparative analysis of various image segmentation techniques. In Proceedings of 2nd International Conference on Communication, Computing and Networking (pp. 359-374). Springer, Singapore.

[180] Zaitoun, N.M. and Aqel, M.J., 2015. Survey on image segmentation techniques. Procedia Computer Science, 65, pp.797-806.

[181] Zhang, T.Y. and Suen, C.Y., 1984. A fast parallel algorithm for thinning digital patterns. Communications of the ACM, 27(3), pp.236-239.

[182] https://github.com/fahmida185/Computer-Vision/tree/master/Project%20Vision

[183] Pau LF. Computer vision for electronics manufacturing. Springer Science & Business Media; 2012 Dec 6.

[184] Jaglan, P., Dass, R. and Duhan, M., 2019. A comparative analysis of various image segmentation techniques. In Proceedings of 2nd International Conference on Communication, Computing and Networking (pp. 359-374). Springer, Singapore.

[185] Zaitoun, N.M. and Aqel, M.J., 2015. Survey on image segmentation techniques. Procedia Computer Science, 65, pp.797-806.

[186] "jetlab II-Precision", Microfab.com, 2020. [Online]. Available: http://www.microfab.com/complete-systems/jetlab-ii-tabletop. [Accessed: 28- Sep- 2020]

[187] Tekin, E., de Gans, B.J. and Schubert, U.S., 2004. Ink-jet printing of polymers–from single dots to thin film libraries. Journal of Materials Chemistry, 14(17), pp.2627-2632.

[188] Wu, S.Y. and Sahni, S., 1994. Fast algorithms to partition simple rectilinear polygons. VLSI Design, 1(3), pp.193-215.

[189] Keil, J.M. and Sack, J.R., 1985. Minimum decompositions of polygonal objects. In Machine Intelligence and Pattern Recognition (Vol. 2, pp. 197-216). North-Holland.

[190] Neggazi, B., Haddad, M. and Kheddouci, H., 2012, October. Self-stabilizing algorithm for maximal graph partitioning into triangles. In Symposium on Self-Stabilizing Systems (pp. 31-42). Springer, Berlin, Heidelberg.

[191] D. V. Leff, P. C. Ohara, J. R. Heath, and W. M. Gelbart, "Thermodynamic control of gold nanocrystal size: Experiment and theory," J. Phys. Chem., vol. 99, no. 18, pp. 7036–7041, 1995.

[192] M. S. Rahman, M. Rahman, S. Pisana, and G. Grau, "Effect of sintering conditions on the thermal properties of printable metal nanoparticle ink studied by thermoreflectance," 2019, p. 28.

[193] S. Magdassi, M. Grouchko, and A. Kamyshny, "Copper Nanoparticles for Printed Electronics: Routes Towards Achieving Oxidation Stability," Materials (Basel)., vol. 3, no. 9, pp. 4626–4638, Sep. 2010.

[194] S. Wünscher, R. Abbel, J. Perelaer, and U. S. Schubert, "Progress of alternative sintering approaches of inkjet-printed metal inks and their application for manufacturing of flexible electronic devices," Journal of Materials Chemistry C, vol. 2, no. 48. Royal Society of Chemistry, pp. 10232–10261, 28-Dec-2014.

[195] Zikulnig, J., Roshanghias, A., Rauter, L. and Hirschl, C., 2020. Evaluation of the Sheet Resistance of Inkjet-Printed Ag-Layers on Flexible, Uncoated Paper Substrates Using Van der Paw's Method. Sensors, 20(8), p.2398.

[196] "Sheet Resistance Equations and Theory | Complete Guide", Ossila, 2020. [Online]. Available: https://www.ossila.com/pages/sheet-resistance-theory. [Accessed: 02- Oct- 2020]

[197] "Van der Pauw method", En.wikipedia.org, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Van_der_Pauw_method. [Accessed: 02- Oct- 2020]

[198] Ramon i Garcia, E., 2015. Inkjet printed microelectronic devices and circuits. Universitat Autònoma de Barcelona,.

[199] Li, L., 2020. Improving Precision of Material Extrusion 3D Printing by in-situ Monitoring and Predicting 3D Geometric Deviation Using Conditional Adversarial Networks (Doctoral dissertation, University of California, Los Angeles).