Materials & Design 241 (2024) 112978

Contents lists available at ScienceDirect

Materials & Design

journal homepage: www.elsevier.com/locate/matdes

Enhancing the 3D printing fidelity of vat photopolymerization with machine learning-driven boundary prediction

Yeting Ma [a], Zhennan Tian [a], Bixuan Wang [a], Yongjie Zhao [b], Yi Nie [a,c], Ricky D. Wildman [d], Haonan Li [a,c,*], Yinfeng He [a,c,d,*]

[a] Nottingham Ningbo China Beacons of Excellence Research and Innovation Institute, University of Nottingham Ningbo China, Ningbo, China
[b] Faculty of Mechanical Engineering & Mechanics, Ningbo University, Ningbo 315211, China
[c] Faculty of Science and Engineering, University of Nottingham Ningbo, Ningbo 315100, China
[d] Faculty of Engineering, University of Nottingham, Nottingham NG7 2RD, UK

ARTICLE INFO

Keywords:
Machine learning
CGAN
Vat photopolymerization
Additive Manufacturing

ABSTRACT

Like many pixel-based additive manufacturing (AM) techniques, digital light processing (DLP) based vat photopolymerization faces the challenge that the square pixel based processing strategy can lead to zigzag edges especially when feature sizes come close to single-pixel levels. Introducing greyscale pixels has been a strategy to smoothen such edges, but it is a challenging task to understand which of the many permutations of projected pixels would give the optimal 3D printing performance. To address this challenge, a novel data acquisition strategy based on machine learning (ML) principles is proposed, and a training routine is implemented to reproduce the smallest shape of an intended 3D printed object. Through this approach, a chessboard patterning strategy is developed along with an automated data refining and augmentation workflow, demonstrating its efficiency and effectiveness by reducing the deviation by around 30%.

1. Introduction

Vat photopolymerization technology is a widely used Additive Manufacturing (AM) method, with applications ranging from dental molds [1], 3D printed sports shoes [2] to hearing aids [3]. Users always seek better fidelity. This is, however, very challenging to achieve, not only because the improvement of printing resolution requires upgrading of the projection system, but also because light intended for a specific location often scatters or diffracts into surrounding locations, creating a halo of unwanted polymerization and enlarges features [4]. A commonly used approach is to introduce grayscale pixels to suppress the unwanted halo while maintaining the energy threshold for the printing region [5]. The distribution of such grey pixels can be optimized by simulating the light distribution from each pixel. In addition, a manipulation of the halo distribution can also be used to create partially cured pixels that can help smoothen the transition between pixels, and eliminate stepping or jagged edges, akin to anti-aliasing techniques employed on images [6]. This enhances fidelity and facilitates the production of intricately detailed, structurally robust components with fewer defects, thereby advancing the capabilities of 3D printing and enabling precise applications across various industries. However, a significant challenge is deciding how to distribute grayscale pixels to mitigate steps and achieve smoother edges in the 3D printed structure, which is still an open question. Simulating the light distribution and selectively allocating different greyscale pixels to achieve the target light distribution profile has been confirmed as a promising technique [7]. Such simulation based methods offer high accuracy, but also require significant computations for every layer of projection images and are costly to apply in practice.

Other approaches in photopolymerization based 3D printing include direct or predictive modelling of the cure process to predict the solidification of ink under various UV intensities, often deploying inverse methods to identify the desired light distribution [8]. Such an optimization was used by He et al. [9], deploying an evolutionary algorithm combined with ink jet 3D printing to identify the necessary material distribution to achieve the desired function. These models, however, generally necessitate the measurement of the ink cure degree or mechanical performance. To achieve this, an additional characterization

* Corresponding authors at: Nottingham Ningbo China Beacons of Excellence Research and Innovation Institute, University of Nottingham Ningbo China, Ningbo, China.

E-mail addresses: Haonan.li@nottingham.edu.cn (H. Li), Yinfeng.He@nottingham.edu.cn (Y. He).

https://doi.org/10.1016/j.matdes.2024.112978
Received 21 January 2024; Received in revised form 20 March 2024; Accepted 22 April 2024
Available online 24 April 2024

step is necessary, which can be both time-consuming and intricate. Furthermore, handling the large-scale computations and simulations involved often demands the use of high-performance computers or parallel computing clusters. This requirement significantly increases the complexity of the entire process.

A consideration here is applying machine learning (ML) to help simplify the process of allocating such greyscales to achieve the optimal light distribution. Unlike the simulation method that are often resource-intensive, ML, once trained, offers a data-driven approach to swiftly identify optimal configurations, reducing costs and enhancing printing efficiency. For example, Pattinson et al. [10] introduced an automated dataset refining process to establish a large database for training a model that can better optimize print quality. You et al. [11] and Guan et al. [12] used ML to mitigate scattering effects for significantly improving 3D printing accuracy, enabling substantial enhancements for large-scale features (1 mm*1 mm). Zhao et al. [13] combined ML and an optimization evolutionary algorithm (EA) to obtain objective stress–strain curves fast and efficiently. A recent advance is that of Killgore et al. [14] where they assess the efficacy of a number of neural network based ML models to predict projection patterns, resulting in a substantive increase in the precision of their vat photopolymerization 3D printing. However, their protocol requires a relatively large amount of characterization data together with levels of computing power that may not be available in all labs.

In this paper, a data-driven method is proposed to efficiently achieve greater shape accuracy, i.e., obtain structures that are closer to the desired geometry than is achievable with alternative methods. This is accomplished by introducing a data acquisition and model training strategy that reduces the requirement for data collection whilst improving the printing quality of the manufacturing process. This approach involves two ML steps, the first to acquire and expand the training data, and the second as a model to reveal the correlations between projection and printed pattern, and deploy that as a tool to realize better structures. The approach was tested against three different commercially available vat photopolymerization resins and a variety of challenge structures, showing significant improvements in all cases,

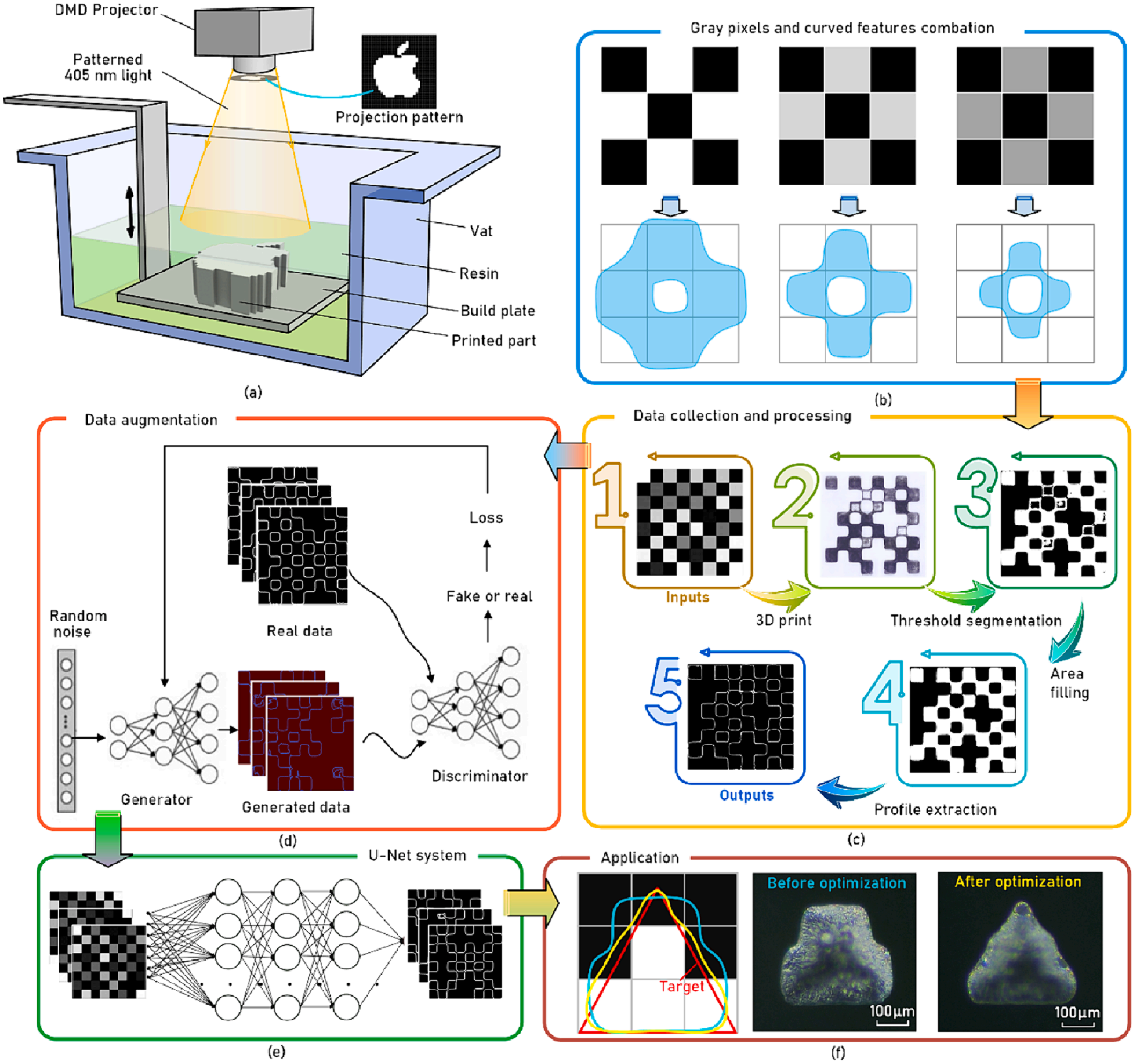


Fig. 1. Conceptual diagram of (a) lab-built vat photopolymerization printing device based on digital micromirror device (DMD) based UV projector. (b) schematic showing the same projection pattern with different combination of greyscale pixels and therefore light intensity can give different solidification profiles; (c) on the basis of the phenomena showing in (b) training data acquisition and processing steps was developed to collect dataset for training a model to between the projection pattern and solidification profile; (d) traditional and CGAN based data augmentation process to extend the number of data points for training; (e) establishes the relationship between the projection pattern and the curing pattern through U-Net system; (f) the final performance assessment of the trained model.

including 3D objects.

2. Methodology

The approach of this paper aims to create a ML model that is able to understand the correlation between different combination of greyscale pixels and their cured pattern profile (Fig. 1a and 1b). Such a model can then be used to optimize the printed pattern profile by judicious spatial distribution of greyscale levels of pixels. A set of chessboard projection patterns was chosen as such an arrangement can maximize the breadth of adjoining pixel greyscale level combinations. The printed samples were then washed and their appearance captured using an optical microscope and processed in a semi-automated high-throughput manner for data refinement (Fig. 1c). They were then paired with the projection data to build a database for feeding and training the model. To expand this database to a more useful size, a trained conditional Generative Adversarial Network (CGAN) [15–17] (Fig. 1d) was also introduced to generate artificial data and therefore expand the database without additional experiments. The accumulation of data generation strategies led to 1200 pairs of chessboard data. The final U-net based neural network was trained with mixed experimental and artificial data, allowing us to uncover the projections needed to realize the desired features in manufactured objects (Fig. 1e)[18–21].

This approach was then tested by challenging the model to replicate intricate structures and devices, including filters and microfluidic channels, and determining the level of improvement that can be achieved via a machine learning method. This showed that the approach could improve the shape accuracy by 200 % for both 2D patterns (Fig. 1f) and 3D devices.

2.1. 3D printer development and sample post-processing

A lab-built vat photopolymerization printer (as shown in Fig. 2a) was configured with a UV projector (GVINDA, Production name PDC05-70, printing resolution 66–75 μm, Optical distortion 0.10 %) at 405 nm wavelength and 0.65 in. 6500 DMD chip. The resolution of the chip was 1920 pixels by 1080 pixels. The build stage was immersed in the resin tank and after each layer was 3D printed, the build stage moved downward to allow a new layer of liquid resin to recoat the top surface. The UV pattern was projected onto the release film to form a solidified pattern on the build platform. (see Fig. 2b). During the 3D printing process, the UV projector was operated at a constant 25.6 mW/cm^2 UV intensity with a 3 s exposure time.

Three commercially available resins were used. The model was initially trained using transparent UV resin (Resin 'A', Anycubic Classic), followed by training with two other UV resins (Resins 'B', Creality UV sensitive resin White and 'C', Anycubic ABS-like Resin + Grey). These resins differ in material composition, color, and translucency, and were selected to demonstrate the model's versatility in adapting to resin variations.

2.2. Projection strategy

The projected patterns were generated using a chessboard strategy, where one set of diagonals was fixed to black pixels (0) whilst the other set was composed of pixels where the greyscale values were randomly distributed from 0 to 255. The chessboard pattern enables maximization of the grayscale value distribution, with its interlocking light and dark areas effectively simulating and balancing the distribution of light across the printing surface. Moreover, the simple and regular structure of the chessboard pattern serves as an excellent approximation for the illumination and curing processes of complex geometrical shapes. By analyzing the curing effects of these basic patterns through ML, the insights gained can be extended to more intricate patterns and structures, thereby optimizing the entire printing process. Each 2D 'chessboard pixel' was defined to be composed of a 4x4 set of 'projector pixels', and then a projection was generated to form an 8x8 chessboard. This strategy was chosen to ease fabrication and characterization but can be scaled trivially to 2x2 projector pixels or further as desired. Subsequently in this paper, the term 'pixel' will be used to refer to a 'chessboard pixel'. Twenty chessboard patterns were produced in a single batch, and in total 300 were produced.

2.3. Data acquisition

The 3D printed samples were washed using an ultrasonic cleaning machine and then dried naturally. The samples were imaged with an optical microscope (KEYENCE, VHX-1000), followed by threshold processing with a threshold value of 127 to generate binary images (see the Supplementary Information). Registration of the image was ensured via fixing of the top-left corner and bottom-right corner pixels to value 255 in all projections (see Fig. 3a).

The binary image and the projection pattern were then aligned by shifting and rotating each image in discrete steps and selecting the

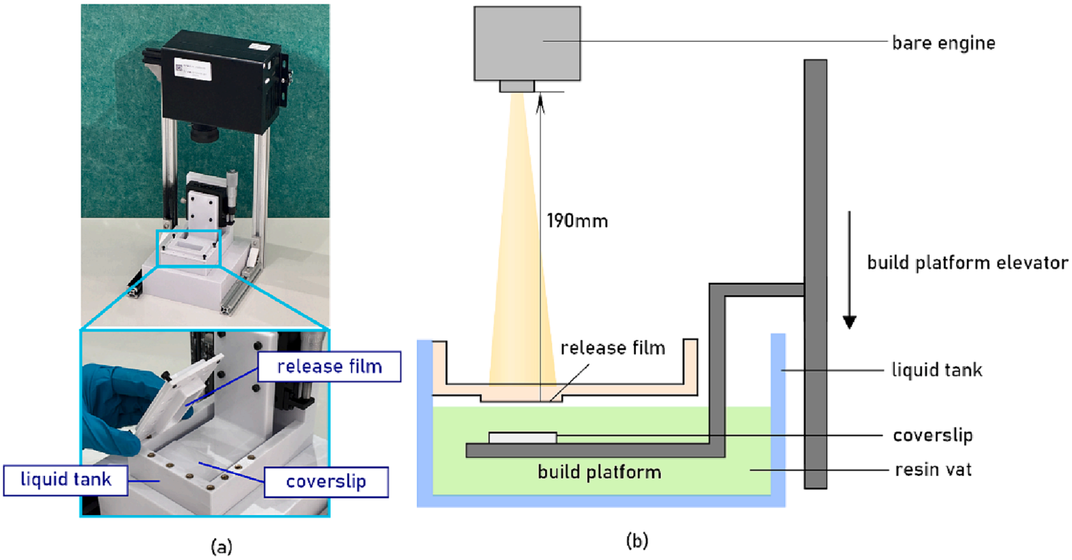


Fig. 2. Schematic diagram of the lab-built vat photopolymerization printer: (a) and the appearance of the whole system and the printing section (b) schematic of the printer's setup.

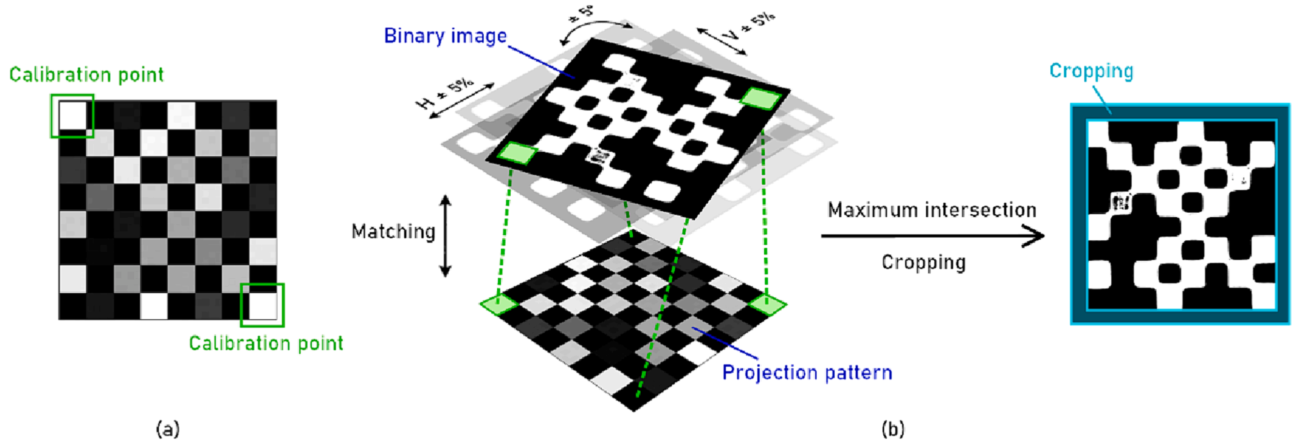


Fig. 3. Calibration of binary image to form datasets with the projection pattern: (a) chess board design with calibration points; (b) automated fine-tune the binary image to match the projection pattern.

combination that gave rise to the strongest overlap (see Fig. 3b and Supplementary Section 2).

2.4. Data refining

Due to the noisiness of the data emerging from the manufacturing / data collection workflow, the next step was data refinement and curation (Supplementary Figure S2). The detailed strategy can be found in Supplementary Section 3. In summary, this process is able to identify the error data caused by reflection of light during optical microscopy process without falsely removing any real data (Supplementary Fig. S2b-B).

In addition, directly using the image to train the model would cost a

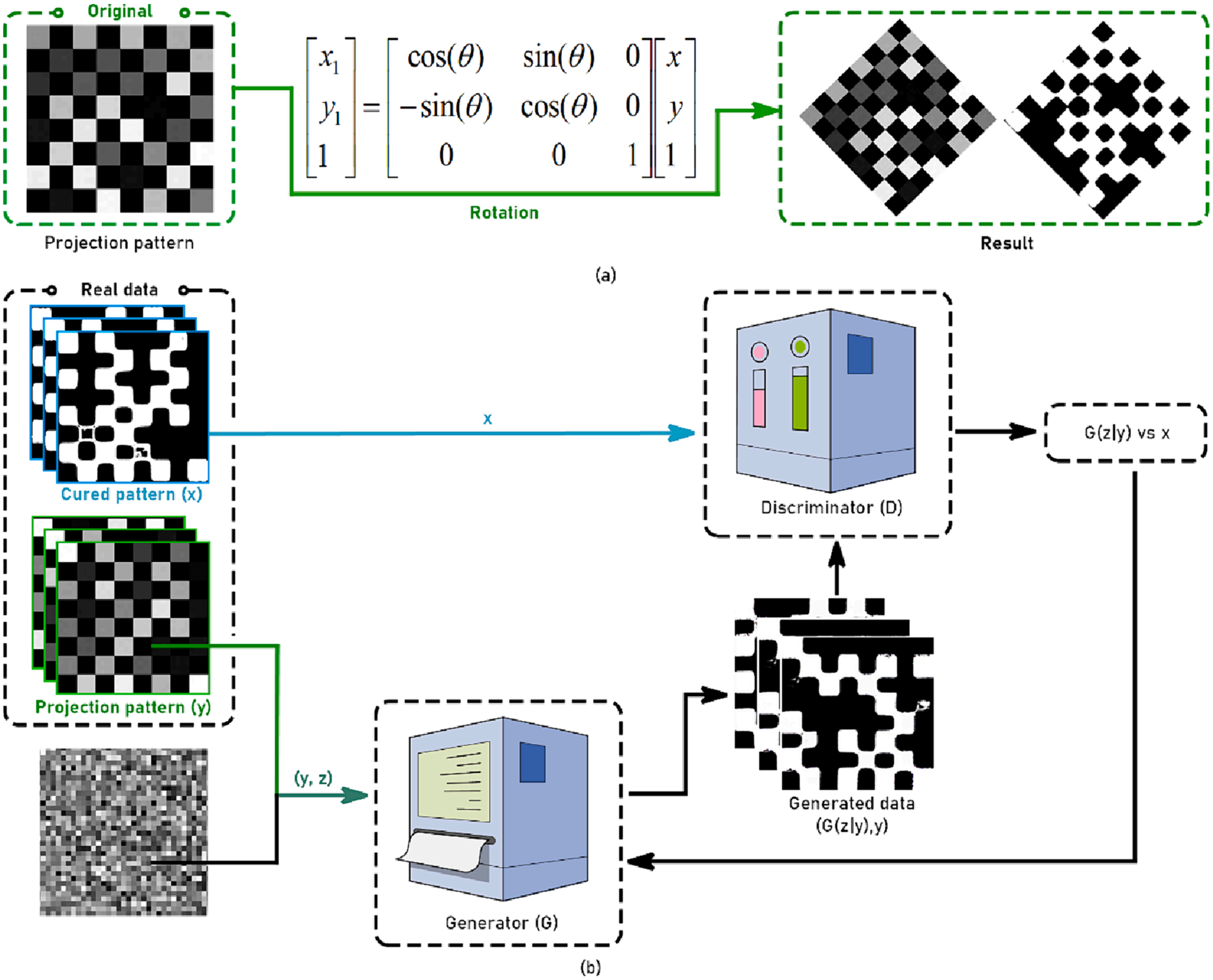

Original
Projection pattern
$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
Rotation
Result
(a)
Real data
Cured pattern (x)
Projection pattern (y)
x
(y, z)
Discriminator (D)
G(z|y) vs x
Generated data
(G(z|y),y)
Generator (G)
(b)

Fig. 4. Schematic of the data argumentation strategies used in this study: (a) Traditional data augmented methods to expand data through geometric transformation and (b) Conditional Generative Adversarial Network (CGAN) was trained and used to generates data that can further expand data pattern diversity.

considerable amount of time. To increase the efficiency, the image profiles can be extracted according to the gradient value and direction: where the edge of the shape appears, its gradient is always perpendicular to the edge. The approach accurately converted a figure pattern into profiles (see Supplementary Fig. S2f), which considerably reduces the computational burden during the training process [22] and as a result, the training speed was increased by 5 times so that the processing time of each pair of data is reduced from 890 ms to 178 ms (Supplementary Section 4 Table S2).

2.5. Data augmentation

During the ML process, the size of the training dataset significantly influences the performance of the final model. However, in practice, collecting a large training dataset by physically 3D printing thousands of samples is time-consuming and can substantially increase both time and labor costs. According to the experimental process of data collection, it was calculated that it takes approximately 65 min for one batch (20 samples), including 2 min for operating the projection device and placing the coverslip, 3 min for projection, cleaning, and air drying, and 60 min for capturing microscope images. Therefore, data augmentation [23] was employed to expand the dataset using both traditional data augmentation and CGAN.

2.5.1. Traditional data augmentation

Using traditional data augmentation techniques, rotation was initially applied to the 300 sets of experimental data, including projection images and its corresponding microscope images after threshold segmentation (see Fig. 4a). These methods can effectively increase the diversity of training data and improve the generalization ability of the model. However, traditional data augmentation is constrained by the data distribution, and the generated samples cannot cover the entire data space. Therefore, when the traditional data augmentation reaches a certain amount of data, it not only has no further benefits for model training, but also increases the model's training time[23,24]. As a consequence, traditional data augmentation methods were used to increase the training dataset from 300 to 600 sets of data.

2.5.2. Conditional Generative Adversarial networks (CGAN)

In order to enhance the pattern diversity and increase the data available, a Pix2Pix model, which is a type of CGAN, was integrated into the approach to augment the training data. A trained Pix2Pix CGAN aims to generate corresponding output images from a conditioned input image [25]. It normally contains a U-net based architecture and discriminator. The training process include feeding the system with paired input data and output data to help the model establishing correlations between them and therefore when it receives an input data, it is able to predict and draw an output data based on the how it was trained. Although widely used in AI drawing, this could also be used in our data augmentation as our data pairs are paired conditioned patterns. This can provide an effective solution to the problem of limited data and greatly improve the accuracy. Such a protocol has been used before by Lakmal [26] and Liu [27] for data augmentation in which they confirmed that Pix2Pix can perform precise image-to-image mapping, such as converting sketches to detailed images or transforming daytime photos into nighttime scenes, thereby enhancing the diversity of the dataset. It has also demonstrated the capability to generate realistic synthetic images, effectively augmenting the amount of data available for model training.

The CGAN [28] is capable of generating data based on a generator. In our case, the generator accepts a randomly generated pattern alongside a grayscale chessboard pattern as inputs to produce a profile of the cured pattern. It is the role of the discriminator to determine whether the data is generated or real (see Fig. 4b).

The loss function used in the training process of CGAN is as follows:

$$\min_{G}\max_{D} V'(D,G) = \mathrm{E}_{x\ \mathrm{P_{data(x)}}}[\log \mathrm{D}(x|y)] + \mathrm{E}_{z\ \mathrm{P_{z(z)}}}[\log(1-\mathrm{D}(x,\mathrm{G}(y|z)))] \quad (1)$$

Here, y represents the given condition, x is the real data corresponding to condition y, z is the random noise input to the generator, G(y|z) is the data generated by the generator based on condition y and noise z, and D (x|y) is the probability estimate by the discriminator that the given data x under condition y is real.

In order to generate the 3D printed profile that is similar to the real situation, we utilized the Stochastic Gradient Descent [29] method to optimize the generating parameters (detailed below):

$$\theta_{G}{}' \leftarrow \theta_{G} + \alpha\frac{1}{m}\sum_{i=1}^{m}\frac{\partial \log(1-\mathrm{D}(\mathrm{G}(z_i|y_i)))}{\partial\theta_{G}} \quad (2)$$

where θ_G is the adjustable parameter in the generated model which controls the behavior of the generator and determines how the generator generates samples from random noise. (By adjusting the value of θ_G, we can change characteristics such as the quality, diversity, and authenticity of the samples generated by the generator). α is the learning rate, m is the number of samples, D is the discriminator and z_i and y_i represent separately a generator input and the corresponding condition label.

By implementing the CGAN strategy, the 600 datasets obtained from traditional data augmentation were further expanded to a total of 1200 datasets. It is worth noting that due to limitations in computing power, it is not practical to train a dataset larger than 1200 in our case, but in principle the dataset can be enlarged with more computational power.

2.6. Model training with u-net model

The U-Net architecture was originally customized for biomedical image segmentation, distinguished by its ability to capture global information and preserve detailed features within images. This structure enhances the understanding of input images and enables accurate pixel-level predictions. Therefore, it is highly suitable for image translation tasks requiring high precision, facilitating the precise establishment of relationships between grayscale projection images and binary images, corresponding to the profile printed by different grayscale pixels, to achieve accurate pixel-level predictions.

In the U-NET implementation, a hyperbolic tangent activation function [30] was used, which is subsequently mapped to 0 to 255 to achieve the final grayscale image. As per the training, a cross entropy loss function [31] (Supplementary Equation S5) was used to compare the difference between the predicted and the real data and similarly a stochastic gradient descent [32] was employed to update model parameters (see Supplementary Section 6).

Fig. 5 illustrates the model training process. The training dataset was partitioned into two groups: the training group (contained 80 % of the dataset: 960 pairs of data) and the validation group (contained the remaining 20 % dataset: 240 pairs of data). The training group was employed to train the ML model, thus establishing the correlation between the projected pattern grayscale image and the 3D printed profile data. The validation group is tasked with testing the model's performance on unseen data to adjust its training parameters. The detailed steps are given in the supplementary information.

Subsequently, the 3D printed profiles from the validation group were input into the trained model to generate the corresponding projection pattern. By comparing the generated projection pattern with the original projected pattern, the deviation could be measured and the model refined. To evaluate the discrepancy between the computed projected pattern and the original projected pattern, the Mean Squared Error (MSE) mode was utilized. It is worth noting that MSE is a method used to observe the training process and does not replace the loss function. The MSE is given by

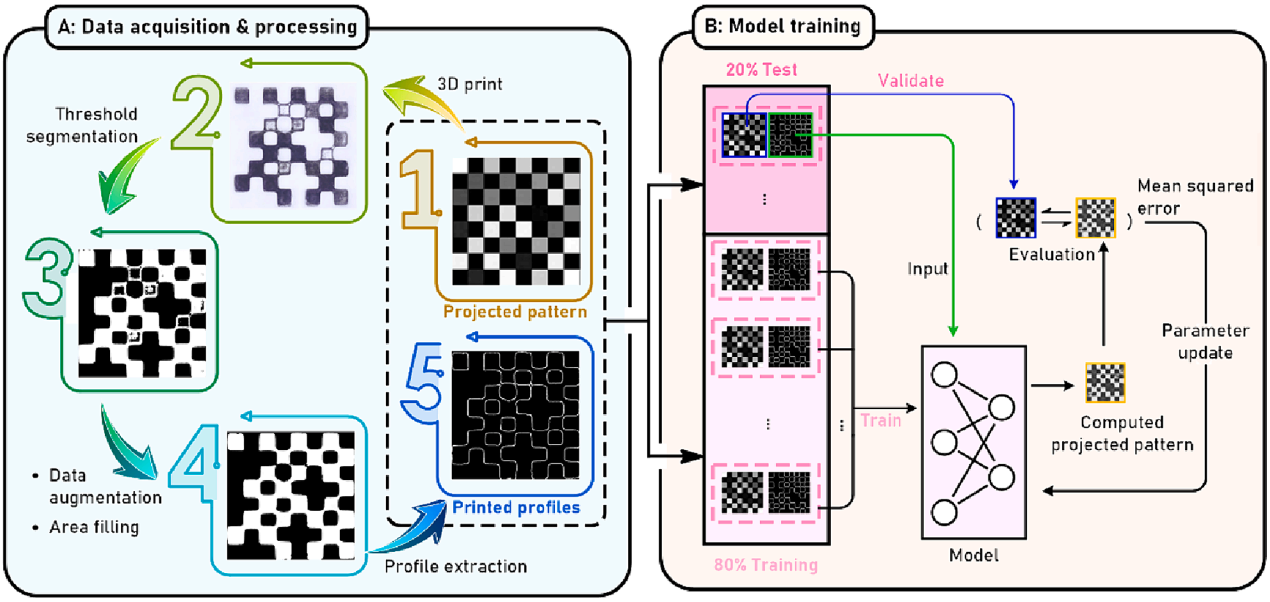


Fig. 5. Model training process: (A) acquisition and processing of chessboard data to obtain final training data; (B) datasets partitioning, model training and validation.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(\mathrm{Y}_i - \widehat{\mathrm{Y}}_i\right)^2 \tag{3}$$

where n is the number of samples, Y_i is the actual data, $\widehat{\mathrm{Y}}_i$ is the predicted data from the trained model.

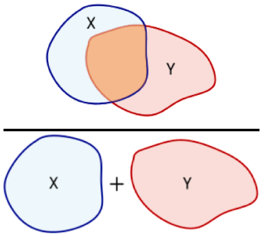
2.7. Metrics for quantitatively evaluation

In order to evaluate the performance of the model and quantifying its accuracy, two coefficients were introduced to quantitatively assess shape accuracy of the 3D printed structure. The computation of these two coefficients is carried out directly by an existing module in an open-source Python library. However, it is important to note that prior to invoking the module for computation, the printed output needs to undergo data processing for automatic calibration (Section 2.4).

Dice coefficient

Dice coefficient is commonly used in image processing to assess the amount of overlap and therefore similarity between two pictures. Therefore, this coefficient was introduced to quantify the performance of ML process.

$$\mathrm{Dice}(X, Y) = \frac{2\times|X\cap Y|}{|X|+|Y|} = 2\times$$

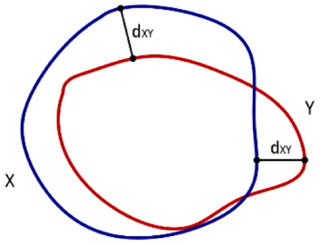


(4)

Where X and Y represent two sets, |X| denotes the cardinality of set X, |Y| denotes the cardinality of set Y, and $|X\cap Y|$ denotes the cardinality of the intersection of X and Y. The range of the Dice coefficient is between 0 and 1, the coefficient approaching 1 means the two images have high similarity.

Hausdorff distance

In addition, another metric, the Hausdorff distance, was incorporated. The Hausdorff distance places greater emphasis on accurate matching and is widely employed in the detection of positional errors and other feature extraction methods.



$$\mathrm{Hausdorff}\,(X, Y) = \max\left[\sup_{x\in X}\inf_{y\in Y} d(x, y), \sup_{y\in Y}\inf_{x\in X} d(x, y)\right] \tag{5}$$

In this context, $d(x, y)$ represents the distance between x and y, while 'sup' and 'inf' denote the upper and lower limits, respectively.

While both the Dice coefficient and Hausdorff distance are essential for evaluating image shape accuracy, they address distinct aspects of segmentation accuracy. The Dice coefficient emphasizes overall similarity between target and actual outcomes, serving as a reliable indicator of general accuracy. On the other hand, the Hausdorff distance highlights the maximum boundary difference, offering crucial insights into key point precision. By combining these metrics, a holistic evaluation is achieved, ensuring results that are not only accurate overall but also precisely delineated. For more details, please refer to our supplementary information.

3. Result and discussion

3.1. The improvement of the model versus the training data

Fig. 6 illustrates the evolving 3D printed sample quality as the number in the training dataset increased. It's evident that when 100 datasets were fed into the model, the snowflake printed from the generated data still contained a significant number of errors. Some detailed structures, such as branches were merged together. However, as the number of training data increases, the quality of the printed snowflake started to improve, exhibiting enhanced details and accuracy. The branches of the snowflake became clearer, and the overall pattern appeared more symmetrical than before. To describe this evolution more accurately, the evolution of snowflake fidelity with the number of

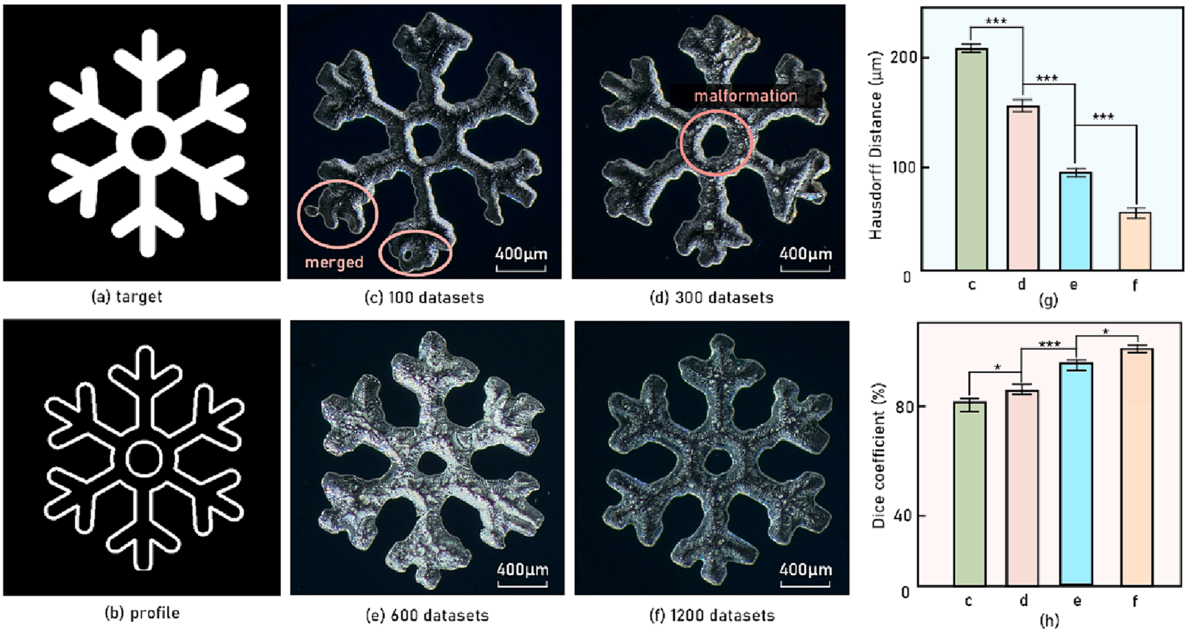


Fig. 6. The effects of data augmentation methods: (a) desired target and (b) the profiles of target; (c) snowflakes 3D printed by a model trained using 100 datasets and (d), (e), (f) snowflakes 3D printed by a model trained on 300, 600, 1200 sets of training data, respectively; (g) and (h) quantify the evolution of snowflake fidelity with the number of datasets by Hausdorff distance and Dice coefficient. Significance was assessed using a one-way Anova with post hoc Tukey's test (* denotes $p < 0.05$ between the two, ** denotes $p < 0.01$, and *** denotes $p < 0.001$).

datasets was quantified using the Dice coefficient [33] and Hausdorff distance [34] (see Section 2.7). Hausdorff distance significantly decreases with the increase of model training data, while Dice coefficient gradually increases (see Fig. 6 g-h), indicating that the snowflake pattern 3D printed is getting closer to the target. Significance testing also confirms that this improvement is statistically significant.

In addition, with the help of MSE, it is possible to monitor the training progress and it was found that a lower MSE value was achieved as the number of epoch increase. This indicated a more accurate prediction of the projected pattern when a target structure is input, which means a greater proficiency of the model at capturing the correlation between the projected pattern and the actual 3D printing profile (Supplementary Figure S6).

3.2. Performance test with simple patterns

In order to evaluate the performance of the model trained through the approach, the model was challenged with two pixel-scale designs: a circle and a triangle (see Fig. 7). It can be seen from Fig. 7A1 and A3, due to limitations in 3D printing resolution, a circle with a diameter of 3 pixels will be sliced into a simple cross, thus failing to achieve the intended design. However, after optimization by the trained model, such small curvatures could be achieved with the correct combination of grayscale pixels.

In Fig. 7A3 and B3, the outer profile of the designs as well as the actual 3D printed sample profile were extracted from the projection patterns before and after optimization. It can be seen that the maximum deviation distance before optimization (H1) is considerably larger than that with the optimized pattern (H2).

Using the two coefficients allowed for quantitative comparison of shape accuracy improvements. The Dice coefficient and Hausdorff distance quantitatively assess the enhancement from two different perspectives. Following model optimization, both simple pattern's Dice coefficients increased by around 10 % (as seen in Fig. 7C1), indicating a similarly overall improvement. However, in terms of the maximum boundary difference measured by the Hausdorff distance, the improvement in the triangular design was not as good as the circular structure, the accuracy of the design improved by 80 % for the triangle and around 300 % for the circle design (as seen in Fig. 7C2). It was noticed that the key deviation came from the conner of the triangle structure. Such shapes with sharp corners are more sensitive to small deviations and this can be better captured by Hausdorff distance. The significant reduction of both parameters suggests that the 3D printed shape now possesses much higher fidelity than before. Remarkably, this improvement was achieved without any hardware upgrades.

3.3. Performance assessment with complex patterns

To further assess the effectiveness of the optimization method, the model was challenged with complex, small patterns (see Fig. 8).

From Fig. 8a, it can be observed that the antennae of the butterfly (location i) more closely matched the designed pattern when using the model. While both patterns were unable to reproduce the hollow patterns in the butterfly wings fully, the optimized version still exhibits improved printing quality and captures more hollow structures than the non-optimized version (locations ii and iii).

Similar enhancements are evident in the bug and watermelon patterns, demonstrating the higher fidelity achievable with the trained model. In the bug pattern, the bug's leg, measuring only around 20 µm, showed considerable improvement, in regard to the curvature and void. For the bug 3D printed with the non-optimized pattern, the bug's legs tended to merge together, with significantly poorer detailing (locations iv and v).

Similar improvements are also observable in the watermelon pattern, which contains both large (location vii) and small curvatures (location vi). The sample 3D printed by the non-optimized patterns tends to reproduce these curvatures as either right angles (location vii) or zig-zag paths (location vi). The trained model improves this, eliminating such step-like appearances and therefore helped to better reproduce the curvature in the original design.

As previously, the Dice coefficient was utilized to quantify the improvement in the samples 3D printed with and without optimization. As can be seen from Fig. 8b, the shape accuracy of the pattern improved by approximately 10–20 % on average. The Hausdorff distance also

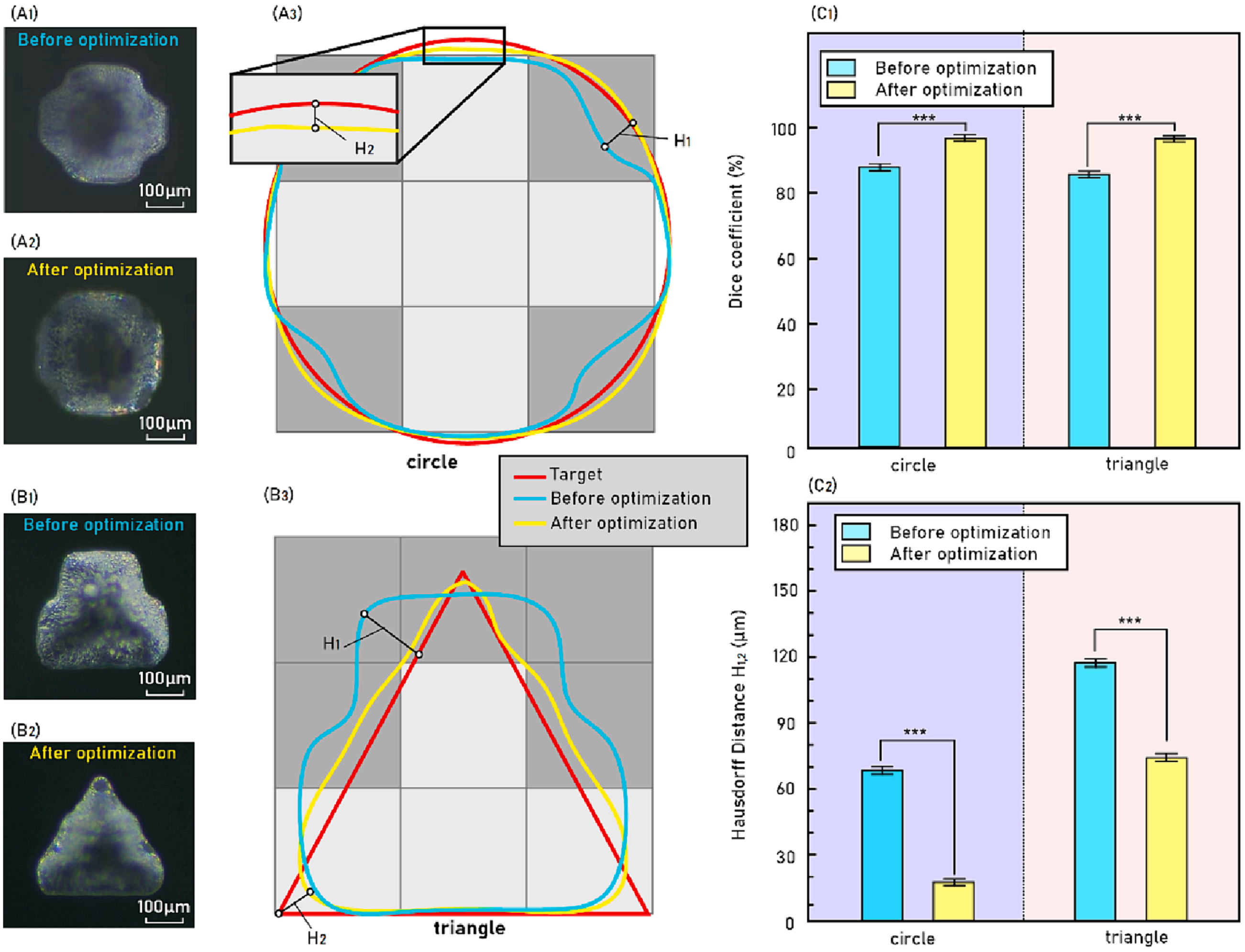


Fig. 7. Two pixel-scale designs 3D printed to verify the model effect: (A1) a 3D printed circle with a diameter of 3 pixels before optimization and (A2) a 3D printed circle with a diameter of 3 pixels after optimization by the proposed method; (A3) the overlap of the 3D printed circle and the target circle; (B1) a 3D printed triangle with sides 3 pixels long before optimization and (B2) a 3D printed triangle with sides 3 pixels long after optimization; (B3) the overlap of the 3D printed triangle and the target triangle; (C1) Dice coefficient and (C2) Hausdorff distance. Significance was assessed using a one-way Anova with post hoc Tukey's test (* denotes $p < 0.05$ between the two, ** denotes $p < 0.01$, and *** denotes $p < 0.001$).

showed similar trends as given in Supplementary Figure S7. In terms of the Dice coefficient performance, both simple and complex patterns showed improvements of around 10–20 % after optimization. However, a comparison based on the Hausdorff distance revealed that the improvement in the simple patterns was 4–15 times greater than that in the complex patterns. This disparity may be attributed to the diverse features present in complex patterns, such as very sharp corners, for which our current training data may not be sufficient for optimal optimization. Future research efforts could focus on augmenting the training data with detailed structures to further enhance optimization accuracy.

Experimental results with Hausdorff distance and Dice coefficient highlight the efficacy of grayscale processing in smoothing edges and closely approximating the target image, underlining the importance of precise pixel-level control to prevent edge inaccuracies [6]. The study demonstrates the ML model's effectiveness, though it reveals that while both metrics are crucial for assessing image fidelity, they yield different degrees of improvement due to their focus on varying aspects of segmentation accuracy (see Section 2.7).

3.4. Validating the model performance with multi-layer structures

To demonstrate the model can help improve the print quanlity in vat polymerization process, two potential applications were introduced: 3D printed filter and microfluidic channels as demonstration.

3.4.1. Filter

The first application of this involves fabricating filters with smaller, more accurate holes. Using vat photopolymerization to fabricate filter or similar mesh structures is challenging. Although fabricating holes with diameters spanning dozens of pixels isn't problematic, when this diameter is reduced to a few pixels, traditional methods fail to reproduce the correct structures, often leading to completely obstructed voids.

Here, two different types of holes were induced: circular and triangular with 3D printing of multi-layer samples (1–3 layers) to produce a 2 mm*2mm*0.15 mm filter. As shown in Fig. 9, when the size of the circular hole was around 5.2 pixels, the sliced pattern began to display a diamond pattern instead of a circular one. This situation worsened as the hole diameter decreased further, resulting in a 2.6 pixel hole sliced to only one pixel (Fig. 9). This led to the designed hole being fully blocked. However, with the trained model, a new set of projection patterns containing grayscale combinations at the border of the designed hole was generated to better reproduce the intended shape. It's evident that the accuracy of the final hole is improved, and the system is capable of achieving smaller through holes that were previously unattainable.

In addition to single-layer patterns, the impact of multi-layers was also examined (see Fig. 9). The holes were assessed, and the partially

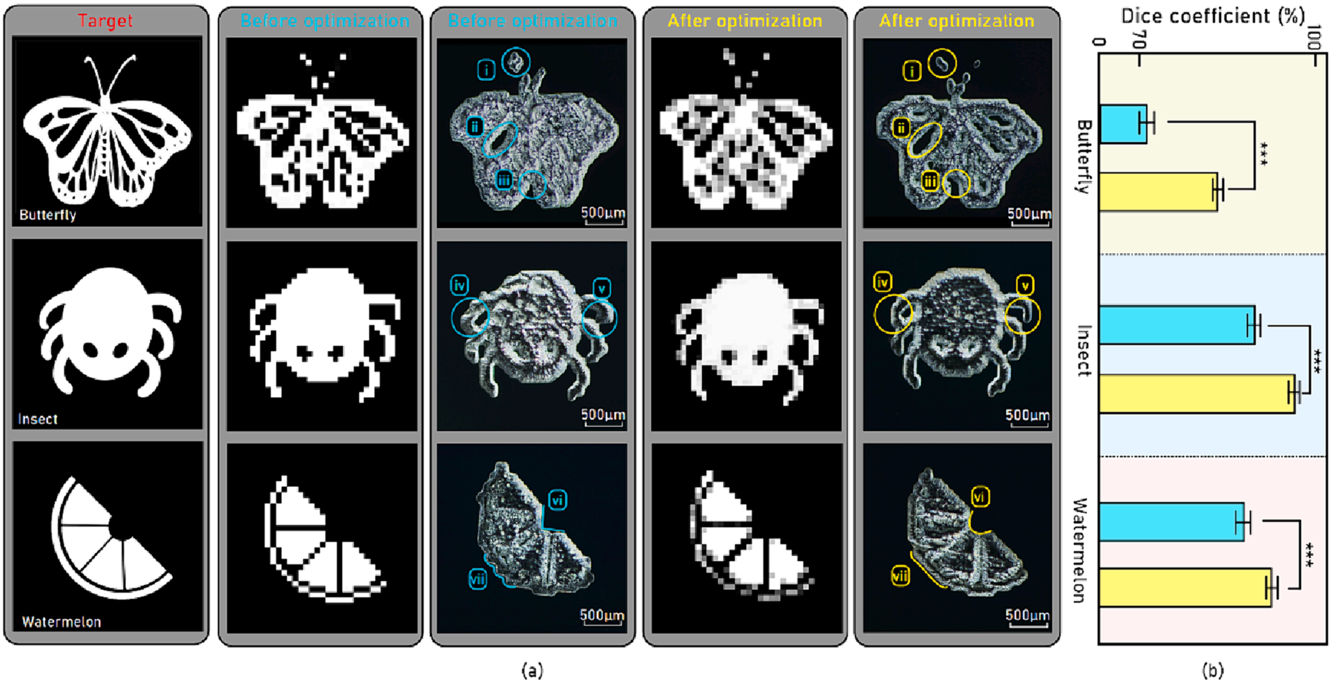


Fig. 8. Complex and small patterns printed to verify the model performance: (a) target pattern, projected image and 3D printed structure before and after optimization; (b) Dice coefficient of the printed patterns. Significance was assessed using a one-way Anova with post hoc Tukey's test (* denotes $p < 0.05$ between the two, ** denotes $p < 0.01$, and *** denotes $p < 0.001$).

blocked holes are labelled by the symbol ⊙ and fully blocked ones by ●. It can be seen that, with the assistance of the model, the blockage of holes was significantly reduced for both single-layer and multi-layer structures, suggesting a good efficiency of the model. A similar effect was also observed in triangular holes. With the proposed model, smaller through holes with better fidelity were achieved compared to the non-optimized patterns. For the 'large' holes (3–5 pixels), their shape accuracy also improved. However, it is worth noting that even though our model is able to increase the success rate of printing small holes, smaller holes (2–3 pixels) tended to block as the number of layers increased. This was owing to our model being trained on 2D data that doesn't incorporate interaction between layers. In actual printing, the projected light can penetrate through layers and cause curing in previously generated layers.

3.4.2. Microfluidics

Microfluidic chips [35] present another application for AM that requires high fidelity of 3D printing for achieving small channels. In DLP printing, small channels (2–4 pixels) could be blocked by the halo of surrounding white pixels for structuring the side wall of the channels. Given this, the model was challenged with a designed mini microfluidic device that included a reservoir and a set of branched channels with stepped channel dimensions, ranging from 7 pixels down to just 2 pixels. The 3D printed chip (as shown in Fig. 10f-g) has a size of 9 mm*9 mm*0.25 mm with channels embedded. All the channels contain three layers, each layer being 50 μm in height, resulting in a total channel height of 0.15 mm.

When 3D printing with non-optimized patterns, blockages started to occur when the channel diameter reached the 5–6 pixel range. As demonstrated in Fig. 10a-e, upon flowing a fluorescent dye through the microfluidic device, it could only penetrate through channels with a minimum of 5 or 6 pixels in diameter, with channels smaller than this being partially blocked.

Conversely, when the printing pattern was optimized with the trained model, the generated grayscale pattern was able to reproduce through channels with better accuracy. Besides, the optimized channel width is also closer to the designed width. As shown in Fig. 10*h*, the unoptimized channel width has a larger deviation from the design and this deviation will increase the width decreases. In the meanwhile, the channels printed from the optimized pattern are more consistent and similar to the designed width. This is likely due to the introduction of greyscale pixels on the edge of the channel helped suppressing the halo of previously used white pixels and therefore suppressed the excessive curing that was causing channel blockage.

The enhancement of printing precision is important for 3D printed microfluidic device as it allows fabricating more compacted device with better control of their flow. However, achieving this is not that straightforward as it include either purchasing a more expensive projector or careful tuning the distribution of grey pixels [36] which could be either expensive or time consuming. In contrast, the utilization of ML methods to predict grayscale projection images enables the printing of details around 100 μm while accelerating the entire process.

3.5. Compatibility of the model with different inks

To ascertain the compatibility of the machine learning model with various ink formulations, the 3D printing process was extended to two additional inks: Resin 'B' (White, Brittle), Resin 'C' (Grey, ABS-Like) (Supplementary Fig. S8b). Different inks have different reactivities, necessitating adjustments to the intensity of the grayscale pixel in the optimized pattern. We selected the 3D printing parameter used for training as the standard and designed a crosshair map to determine the required adjustment level for alternative inks. The crosshair map contains an array of 256 cross-hairs with grayscale values ranging from 0 to 255 (Supplementary Fig. S8a). For the control ink (used for model training), the last successfully 3D printed crosshair was at an intensity of 96 (Supplementary Fig. S8c left). Implementing the same 3D printing parameters for the two additional ink candidates revealed the disappearance of the cross-hair at intensities of 82 and 69 for Resin 'B' and Resin 'C', respectively (Supplementary Fig. S8c mid and right). Based on these results, the overall intensity of the optimized grayscale pattern was modified for each resin. As depicted in Fig. 11b-d, after adjusting the

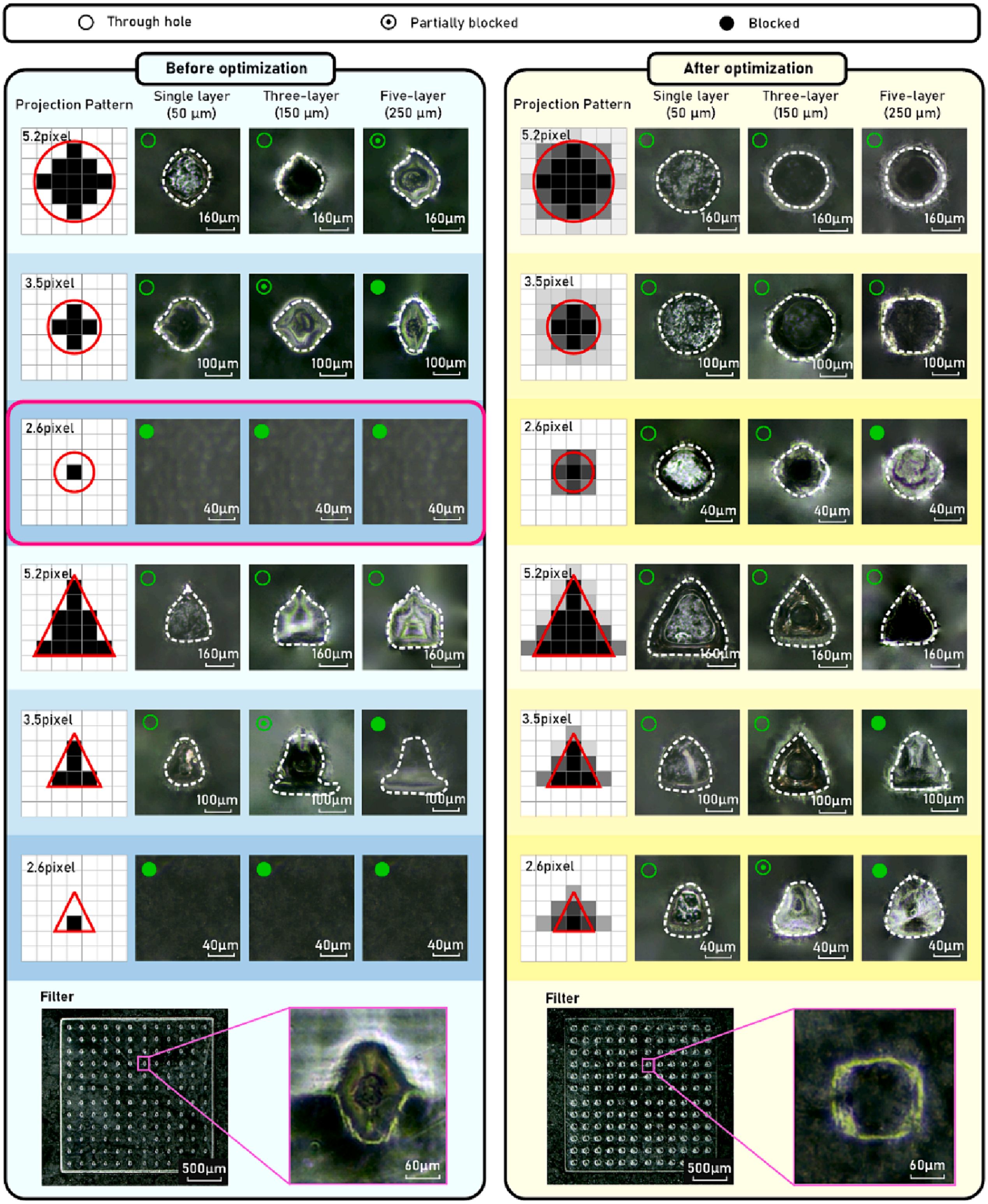


Fig. 9. Analysis of circle (left) and triangle (right) meshes of different sizes on the printed filter like structure. In each row, the projection pattern and its corresponding 3D printed structure was shown with a different number of layers, the height of each layer is 50 µm. The bottom image shows the filter structure 3D printed with non-optimized and optimized patterns.

intensity of the optimized grayscale patterns, the trained model demonstrated a similar optimization level with different ink formulations. Calculating the Dice coefficient of the pattern 3D printed with different resins (Fig. 11e) revealed that by adjusting the pattern's intensity based on the crosshair map, improvements to similar levels was achieved for each resin. This suggests that this model, following a calibration process, is applicable to different ink formulations without the necessity to retrain the entire model.

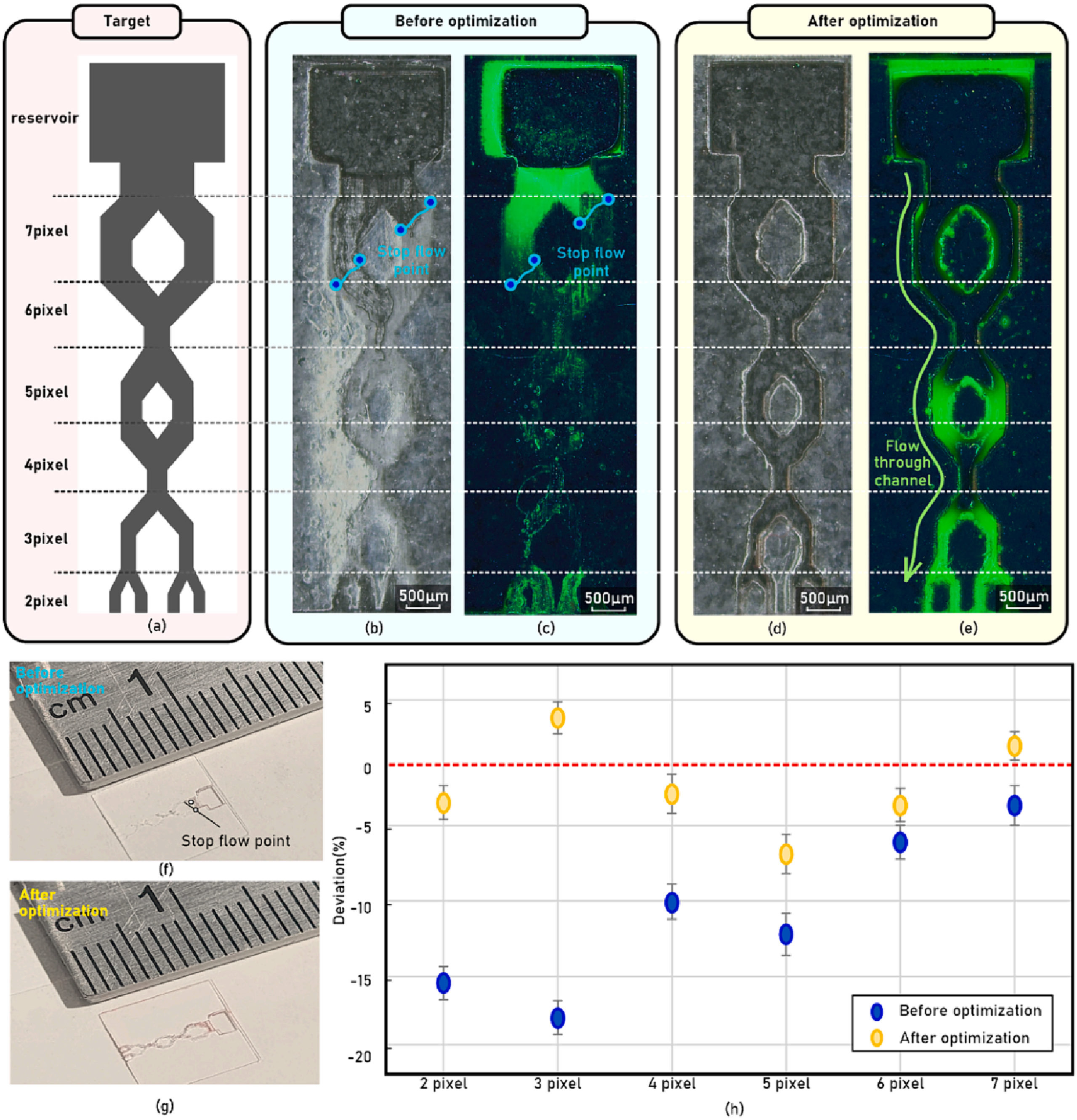


Fig. 10. Printing assessment through fabricating microfluidic channels: (a) the design of channel pattern; (b) 3D printed channel before model optimization and (c) fluorescent ink injected showed blockage of channel in the printed structure; (d) 3D printed channel after model optimization and (e) fluorescent ink was injected showing the channels were succeed; (f), (g) Appearance of the device before and after optimization; (h) the deviation of the 3D printed channel width from the theoretical width.

3.6. Discussion

This work introduces an innovative method for data collection and processing to train an ML model based on U-Net, thereby enhancing the precision of vat photopolymerization in 3D printing. By utilizing a regularized chessboard pattern and an automated calibration strategy, we have significantly reduced the model training time by a factor of five. The effectiveness of this method is demonstrated through the reduction of 3D printing feature sizes and the increased throughput in printing complex structures, such as filters and microfluidic channels, without the need for hardware modifications. The model's adaptability across different materials is validated through simple calibration.

Comparing with simulation-based strategies, we see that the ML approach is able to achieve similar levels of fidelity [6]. Clearly there are advantages and disadvantages of each approach. A simulation can capture the essential physics of a process, and often be easily modified to accommodate changes, but need to be operated as an inverse method to identify optimal conditions. ML methods, however, front load this analysis and can swiftly analyze large datasets, identify patterns to optimize DLP printing, accelerate iteration speed, and potentially reduce the number of required experimental trials. Although physical simulation methods offer more precise optimization results by simulating material behavior and light-material interactions, they necessitate detailed analysis of the simulated printing process, which can be

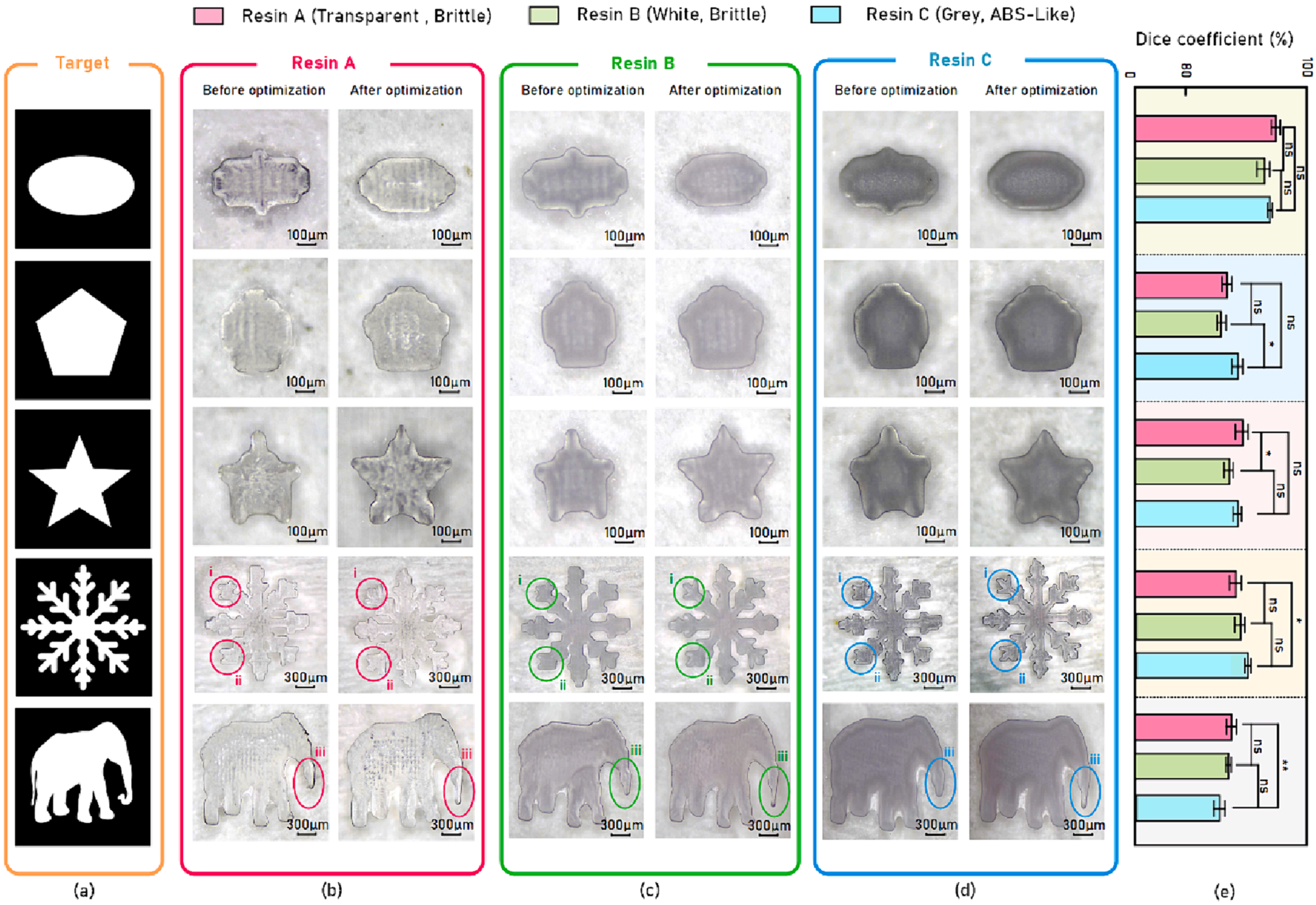


Fig. 11. The performance of the machine learning model on different ink formulations: (a) Target pattern; b-d) the comparison of 3D printed structure with different inks, (b) Resin 'A' (Transparent, rigid), (c) Resin 'B' (White, rigid), (d) Resin 'C' (Grey, ABS-Like); (e) the calculated Dice coefficient of the optimized and intensity adjusted patterns 3D printed with different ink formulations and significance test. Significance was assessed using a one-way Anova with post hoc Tukey's test (* denotes $p < 0.05$ between the two, ** denotes $p < 0.01$, and *** denotes $p < 0.001$).

extremely time-consuming [6]. Moreover, ML models can adapt to different materials and environmental conditions through calibration, providing versatility in optimizing printing precision for various scenarios. In contrast, physical simulations require complex computational resources and extensive computational simulations, limiting their real-time applicability in optimizing DLP printing processes.

However, this strategy still faces limitations that as the training data are limited in 2D, it does show some limitations when dealing with very small or sharp structures or when the print contains multiple layers. We speculate the expanding the dataset to include a more diverse set of materials (e.g., hydrogels) and shapes (e.g., training on non-chessboard pattern) could improve our approach still further.

4. Conclusion

In conclusion, our approach of improving the design of projection patterns, enhances the accuracy and efficiency of vat photopolymerization in 3D printing. Owning to the use of ML, once this model is trained, it could be easily shared and used by many users without the need of significant computing resources. Although, further refining of this process can help this strategy achieving better performance, it still paves the way for advanced possibilities in AM, including broader material adaptability and optimization of automated production processes, thereby enabling the production of higher-performance parts for more complex applications.

CRediT authorship contribution statement

Yeting Ma: Data curation, Investigation, Methodology, Software, Validation, Writing – original draft. **Zhennan Tian:** Software, Validation, Methodology. **Bixuan Wang:** Visualization. **Yongjie Zhao:** Supervision, Writing – review & editing. **Yi Nie:** Supervision. **Ricky D. Wildman:** Supervision, Writing – review & editing. **Haonan Li:** Supervision, Writing – review & editing, Funding acquisition. **Yinfeng He:** Supervision, Writing – review & editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The code and relevant files related to this study are openly available in digitalcommonsdata at 10.17632/6rkxx9j2wr.1.

Acknowledgments

This work was supported by the Engineering and Physical Sciences Research Council Award "Dialling up Performance for on Demand Manufacturing" [grant number EP/W017032/1]; the Nottingham Ningbo China Beacons of Excellence Research and Innovation Institute [grant number I01211200002, I01220800006]; the National Natural Science Foundation of China [grant number 51975302, 52175417]; the K. C. Wong Magna Fund in Ningbo University.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.matdes.2024.112978.

References

[1] A.B. Xepapadeas, et al., Technical note on introducing a digital workflow for newborns with craniofacial anomalies based on intraoral scans-part I: 3D printed and milled palatal stimulation plate for trisomy 21, BMC Oral Health 20 (1) (2020) 1–8.

[2] T. Gong, L. Kang, Application analysis of 3D printing technology in design field: taking shoe design as an example, Sci. Program. 2021 (2021) 1–8.

[3] M. Vivero-Lopez, et al., Anti-biofilm multi drug-loaded 3D printed hearing aids, Mater. Sci. Eng. C 119 (2021) 111606.

[4] Q. Zhang, et al., Design for the reduction of volume shrinkage-induced distortion in digital light processing 3D printing, Extreme Mech. Lett. 48 (2021) 101403.

[5] X. Kuang, et al., Grayscale digital light processing 3D printing for highly functionally graded materials, Sci. Adv. 5 (5) (2019) p. eaav5790.

[6] S.M. Montgomery, et al., Pixel-Level Grayscale Manipulation to Improve Accuracy in Digital Light Processing 3D Printing, Adv. Funct. Mater. (2023) 2213252.

[7] J.E. Hergert, et al., Grayscale Digital Light Processing and Post-Treatment for the Fabrication of 3D-Printed Polymer Blends, Adv. Eng. Mater. 24 (8) (2022) 2101543.

[8] P. Zhao, et al., Modelling the influence of UV curing strategies for optimisation of inkjet based 3D printing, Mater. Des. 208 (2021) 109889.

[9] Y. He, et al., Exploiting generative design for 3D printing of bacterial biofilm resistant composite devices, Adv. Sci. 8 (15) (2021) 2100249.

[10] D.A. Brion, S.W. Pattinson, Generalisable 3D printing error detection and correction via multi-head neural networks, Nat. Commun. 13 (1) (2022) 4654.

[11] S. You, et al., Mitigating scattering effects in light-based three-dimensional printing using machine learning, J. Manuf. Sci. Eng. 142 (8) (2020) 081002.

[12] J. Guan, et al., Compensating the cell-induced light scattering effect in light-based bioprinting using deep learning, Biofabrication 14 (1) (2021) 015011.

[13] B. Zhao, et al., Design of grayscale digital light processing 3D printing block by machine learning and evolutionary algorithm, Compos. Commun. 36 (2022) 101395.

[14] J.P. Killgore, et al., A data-driven approach to complex voxel predictions in grayscale digital light processing additive manufacturing using U-nets and generative adversarial networks, Small (2023) 2301987.

[15] I. Goodfellow, et al., Generative adversarial networks, Commun. ACM 63 (11) (2020) 139–144.

[16] A. Creswell, et al., Generative adversarial networks: An overview, IEEE Signal Process Mag. 35 (1) (2018) 53–65.

[17] J. Gui, et al., A review on generative adversarial networks: Algorithms, theory, and applications, IEEE Trans. Knowl. Data Eng. 35 (4) (2021) 3313–3332.

[18] X. Zhao, et al., Use of unmanned aerial vehicle imagery and deep learning unet to extract rice lodging, Sensors 19 (18) (2019) 3859.

[19] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation. Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18, Springer, 2015.

[20] Y. Weng, et al., Nas-unet: Neural architecture search for medical image segmentation, IEEE Access 7 (2019) 44247–44257.

[21] H. Huang, et al., Unet 3+: A Full-Scale Connected Unet for Medical Image Segmentation. ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP), IEEE, 2020.

[22] T. Zhang, S. Wei, S. Ji, E2ec: An end-to-end contour-based method for high-quality high-speed instance segmentation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.

[23] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, Journal of Big Data 6 (1) (2019) 1–48.

[24] Perez, L. and J. Wang, The effectiveness of data augmentation in image classification using deep learning. arXiv preprint arXiv:1712.04621, 2017.

[25] Lee, H., et al., Pix2Pix-based data augmentation method for building an image dataset of black ice. Available at SSRN 4044502, 2022.

[26] H. Lakmal, M. Dissanayake, Illuminating the Roads: Night-to-Day Image Translation for Improved Visibility at Night. International Conference on Asia Pacific Advanced Network, Springer, 2023.

[27] B. Liu, et al., Sketch-to-art: Synthesizing stylized art images from sketches. Proceedings of the Asian Conference on Computer Vision, 2020.

[28] Mirza, M. and S. Osindero, Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784, 2014.

[29] L. Bottou, Stochastic gradient descent tricks, in: Neural Networks: Tricks of the Trade, Second Edition, Springer, 2012, pp. 421–436.

[30] T. Liu, T. Qiu, S. Luan, Hyperbolic-tangent-function-based cyclic correlation: Definition and theory, Signal Process. 164 (2019) 206–216.

[31] P.-T. De Boer, et al., A tutorial on the cross-entropy method, Ann. Oper. Res. 134 (2005) 19–67.

[32] Bharadhwaj, H., K. Xie, and F. Shkurti. Model-predictive control via cross-entropy and gradient-based optimization. in Learning for Dynamics and Control. 2020. PMLR.

[33] Rame, A. and M. Cord, Dice: Diversity in deep ensembles via conditional redundancy adversarial estimation. arXiv preprint arXiv:2101.05544, 2021.

[34] D.P. Huttenlocher, G.A. Klanderman, W.J. Rucklidge, Comparing images using the Hausdorff distance, IEEE Trans. Pattern Anal. Mach. Intell. 15 (9) (1993) 850–863.

[35] P. Pattanayak, et al., Microfluidic chips: recent advances, critical strategies in design, applications and future perspectives, Microfluid. Nanofluid. 25 (2021) 1–28.

[36] P.J. van der Linden, A.M. Popov, D. Pontoni, Accurate and rapid 3D printing of microfluidic devices using wavelength selection on a DLP printer, Lab Chip 20 (22) (2020) 4128–4140.