IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 17, 2022, accepted February 7, 2022, date of publication February 10, 2022, date of current version February 24, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3150925

A Study on Halftoning Improvement for Low-Resolution Digital Print Engines With Machine Learning Methods

TAL FRANK[1], OREN HAIK[2], SHANI GAT[2], OREL BAT MOR[2], JAN P. ALLEBACH[3], (Life Fellow, IEEE), AND YITZHAK YITZHAKY[1]

[1]Department of Electro-Optics Engineering, School of Electrical and Computer Engineering, Ben Gurion University of the Negev, Be'er Sheva 84105, Israel
[2]Research and Development Department, HP Inc., Ness Ziona 76101, Israel
[3]School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA

Corresponding author: Yitzhak Yitzhaky (ytshak@bgu.ac.il)

ABSTRACT As today's printing volume worldwide decreases, and most traditional printing engines are expensive non-digital devices (offset), the demand for a low-cost digital replacement is rapidly increasing. A main disadvantage of digital presses is the low-resolution capabilities, introducing a compromise in the print quality (PQ). A key factor of print quality is the halftoning algorithm. A very common halftoning method is amplitude modulation (AM) halftone screening, in which dots are placed on a repetitive lattice, varying in size as a function of the grey level. The main AM screen design PQ challenge for low-resolution devices is the quantization frequencies, a disturbing pattern that usually emerges when a screen is approximated to a rational angle due to low resolution. Fourier-based analysis is a classical rule-based method to filter out screens that suffer from visually disturbing quantization patterns. This work presents a new approach that tackles this challenge by incorporating machine learning with the classic Fourier-based approach. Particularly, we show that a binary decision tree classifier with a Fourier-based feature vector has an accuracy of 95% in identifying quantization-free screens compared to the classic rule-based method, which has an accuracy of 66%. We conclude by demonstrating the use of the screen classifier to design a quantization-free screen set. This is done by first applying the screen classifier to the entire screen pool, that is, the set of all possible screens for a given print engine, followed by a rosette zero-moiré offset-like screen design.

INDEX TERMS Halftoning, moiré, print quality, AM screen, regular screens, irregular screens, quantization, machine learning, classification and regression tree (CART).

I. INTRODUCTION

Halftoning is a technique that simulates a continuous-tone image using dots, varying either in size, shape, or spacing [1], [2]. This is usually done with a pre-computed threshold matrix called a screen [3]. In the screening process, a continuous-tone image is fragmented into identical tiles of pixels, or halftone cells, on which the screen is applied. The result is a binary image where every pixel can be either switched on or off [4]. The common method for designing a screen for an electrophotographic (EP) print engine is amplitude modulation (AM) or periodic, clustered dot [1]. This method produces clusters of pixels (spots) that vary in size on a grid as a function of the gray level [5]. The screen resolution, which is usually measured in lines per inch (LPI), is defined as 1/distance between two adjacent clusters, while the screen angle is the orientation of these clusters with respect to the x-axis. Color printing requires the design of a screen set composed of at least four screens, one for each color plane: cyan, magenta, yellow and black (CMYK) [6]. Superimposing periodic screens may cause low-frequency interference patterns, also known as moiré [7]. Print quality is compromised in the cases where these patterns are evident. Avoiding visual moiré patterns within a screen set requires one to pinpoint screens that do not suffer from these patterns when superimposed. Such screen sets are called moiré free [8], [9], and the most common one is the rosette zero-moiré screen set [8], [9] in which the screens have precisely the same resolution and a fixed thirty-degree difference (CMK) between them. Digital print engine screens are quantized to an

The associate editor coordinating the review of this manuscript and approving it for publication was Rosalia Maglietta.

integer number of pixels and thus can produce only a limited number of unique angles and resolutions (i.e., screens) [1], [10], [11]. The screen pool is the set of different achievable screens per print engine. Having a large screen pool is a prerequisite for designing a high-quality moiré-free screen set, especially a rosette zero-moiré screen set. Further, having a large screen pool is important for another reason: it is very common that a print engine has several screen-sets to enable different screen assignments per job. The choice of screen usually depends on content, application, substrate, or color performance. [4], [8]. The size of the screen pool is proportional to the engine resolution: as the resolution increases, the number of available screens increases as well [11]. Thus, if the resolution is too low, as in the case of digital print engines, it is usually impossible to design a high-quality, moiré-free screen set [11]. One solution to this problem is the method of irregular screens [11]–[13]. Irregular screens form a space-varying cluster grid in which 1/distances and the orientation of two adjacent clusters vary across the grid and are equal on average to the screen resolution and angle, respectively. This is in contrast to regular screens that form an ideal cluster grid in the sense that 1/distances and the orientations of two adjacent clusters are fixed and equal to the resolution and angle, respectively. Irregular screens are usually noisier (grainier) and have inferior PQ in respect to regular screens [11], [13]. The low PQ of irregular screens originates from the fact that these screens have additional frequencies in their spectra. The new frequencies, known as quantization frequencies, cause visible and disturbing patterns on the print [10]. When the patterns are too strong, the screen is considered unacceptable for production. The classical rule-based approaches identify quantization-free screens by performing Fourier analysis of the screen spectrum and avoiding screens that contain low frequencies in the screen design process [12], [13]. These approaches use overly simplistic models to simulate the printing process and the human visual system (HVS). Therefore, they do not sufficiently predict whether a given screen will cause visible disturbing patterns after the printing process. A new and more accurate function is needed that distinguishes between acceptable and unacceptable (with visible disturbing patterns) irregular screens in the screen design process. It is very challenging to build such a function using analytic methods [14]. However, machine learning (ML) algorithms are commonly used to automatically build functions from the available data when analytical description is not available [15]–[17]. ML can be applied to construct both linear and non-linear functions. An example of a linear model is logistic regression, whereas non-linear models comprise support vector machine (SVM), K nearest neighbors (KNN) and classification and regression decision trees (CART) [18]–[20]. Because it is hard to predict which ML model will fit the challenge at hand, it is customary to evaluate different models and select the one with the highest performance on new data (spot checking) [21]. These algorithms are well suited for tabular data, that is, data that are represented in a table by rows and columns where each row is a sample and each column is a feature, while the last column is the target prediction. In this work, we propose a new approach that utilizes machine learning (ML) methods to build a binary screen classifier that takes as an input the screen parameters and classifies whether those parameters produce screens with visually-disturbing quantization patterns. To assess the visibility of the quantization patterns in the print, we have conducted psycho-physical experiments that included printing and evaluating several hundred screens on a typical digital low-resolution print engine. The results of these experiments are used to construct a tabular database for the ML algorithms. The ground truth of the data set is the subjects' evaluations of whether disturbing quantization patterns exist in the print (labeling each as acceptable/unacceptable). For each screen, we define the lowest frequencies in the screen spectrum as the feature vectors. We explore different ML algorithms, hyper parameter tuning and smart feature selection on the experimental data set. Particularly, we show that a binary decision tree with a Fourier-based feature vector has an accuracy of 95% in differentiating between quantization-free screens and those which are not. The classic rule-based method has an accuracy of 66% on the same data set. We further demonstrate using the screen classifier to design a quantization-free screen set for a typical low-resolution digital print engine. This is done by first applying the screen classifier to the entire screen pool, followed by identification of screen sets in which the moiré-free condition is satisfied [7] (out of the quantization-free screen pool). As quantization-free screens are only 5% of the screen pool (hundreds of thousands of screens), designing a quantization-free screen set without the screen classifier is very challenging. Our main contributions are summarized as follows:

- We introduce a novel ML-based screen classifier that predicts whether a given screen will cause visible quantization patterns after the printing process.
- We present a method for training a high performance ML-based classifier with relatively low amount of data by using unique features that are best suited for the screen classification challenge.
- We propose a novel screen design method that enables constructing a quantization-free screen set for a low-resolution digital print engine by incorporating a screen classifier.
- Our method outperforms the baseline rule-based method for predicting the presence of a quantization patterns in the print by a large margin.

The remainder of this paper is organized as follows: Section II reviews the latest work, considering irregular screen noise assessment, rule-based quantization pattern predictors and halftoning using ML methods. Section III reviews the current screen set design approach, starting with a description of the basic halftoning process, the screening methodology with regular and irregular screens, and the rule-based screen set design method. In Section IV, a new method to

design a screen set using an ML-based classifier is introduced. Next, the ML methodology is presented: spectrum-based feature extractor, ML models, evaluation metrics and the data structure are covered. In Section V, both the simulation and experimental results of the method are presented, and a designated screen classifier is constructed (suitable for the HP Indigo print engine). Next, the ML-based screen set design method is applied (via the screen classifier that was found in the experimental part) to construct a high resolution, quantization-free screen set. In Section VI we discuss and analyze the results of the simulation and experimental part together with the proposed screen set design method. Conclusions and future research directions are offered in Section VII.

II. RELATED WORK

A. FOURIER-BASED NOISINESS EVALUATION OF IRREGULAR SCREENS

Several Fourier-based approaches have been suggested to evaluate the noise levels of an irregular screen [22]. It shows that the high noise levels of an irregular screen originate from the presence of low (quantization) frequencies that are absent from the spectrum of regular screens. In [12] it is suggested to quantitatively assess the additional noise of an irregular screen by summing the energy of the quantization frequencies in Fourier space and normalizing the energy in the frequencies associated with it, with a corresponding regular screen. In this method, each irregular screen receives a grade (ratio) that is used to assess the expected noise of the screen in the print. An extension of this work is presented in [13], where the irregular screen noise is evaluated directly from the screen basis lattice vectors, and an HVS model is applied to predict the visibility of the noise in the print. Reference [8] suggested measuring the noise level of an irregular screen by counting the number of different cluster shapes per grey level (a regular screen has only one shape per grey level). The main challenge with these methods is their inability to predict pattern visibility by a typical human observer after the printing process. This leads to an inability to determine which screens will be acceptable for production.

B. RULE-BASED APPROACH TO DIFFERENTIATE BETWEEN ACCEPTABLE AND UNACCEPTABLE IRREGULAR SCREENS

A classic rule-based method to differentiate between acceptable and unacceptable irregular screens was suggested in [10] and [23]. In this approach, the quantization frequencies are calculated directly from the reciprocal screen lattice. For each screen, a vector of the six lowest frequencies that also have the highest amplitudes was calculated. Then, a threshold operation was applied (in the Fourier domain) in which acceptable screens are differentiated from unacceptable ones. Setting the threshold values is one difficulty with this method. It is not clear how to adjust them per print engine and the HVS function. Another difficulty is the over-simplicity of this rule-based model. As both the printing process and the HVS function are highly nonlinear systems [5], [14], this model has a low chance of predicting the quantization pattern's visibility as a function of the screen spectrum.

C. IMPROVING HALFTONING ALGORITHMS WITH MACHINE LEARNING METHODS

In [24], a cycle-GAN (generative adversarial network) is used to generate halftone images and also to reconstruct continuous-tone from halftone. This method does not take into account the visibility of halftone patterns in the print, which is the subject of this study. In [25] and [26], the reconstruction of continuous-tone images from halftone images using ML methods is described. While these algorithms aim to improve the quality of a continuous-tone image, we aim to improve printed halftoned images. Some studies were performed regarding the use of machine learning to assess print quality [27] or to rate the severity of print quality defects [28]–[31]. However, those methods are used for evaluating the artifact severity post printing and cannot be incorporated in the pre-printing screen design process.

III. CURRENT SCREEN SET DESIGN APPROACH

This section describes the current approaches to design screen sets for amplitude modulation halftoning algorithms that are used for digital print engines.

A. HALFTONING

Halftoning is a function $H(I)$ that transforms a multi-bit, multi-tone image $I[m, n]$ into a 1-bit, two-tone image $\tilde{I}[m, n]$. Usually, $I[m, n] \in [0, 255]$ and $\tilde{I}[m, n] \in \{0, 255\}$ [1]:

$$\tilde{I}[m, n] = H(I[m, n]) \tag{1}$$

Halftoning results in a degradation of both tonal and spatial resolution of the image, as can be seen in Fig. 1.

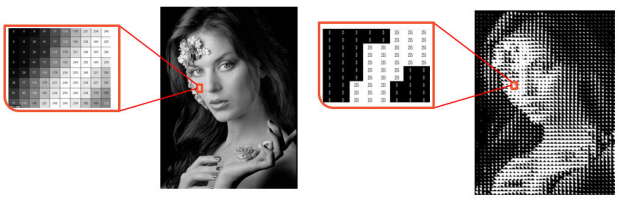


FIGURE 1. Halftoning degrades both the tonal range and the spatial resolution.

Halftoning is a mandatory step in binary systems such as print engines. The halftone image acts as a map of locations that indicate the coordinates on which ink drops or toner clusters will be deposited. A high-enough halftoning resolution will create the illusion of a continuous-tone image, thanks to the low pass filter characteristics of the HVS, as shown in Fig. 2 [32].

B. SCREENING

Screening is one of the fastest, cheapest, and most practical ways to perform halftoning [4]. In this method, a square

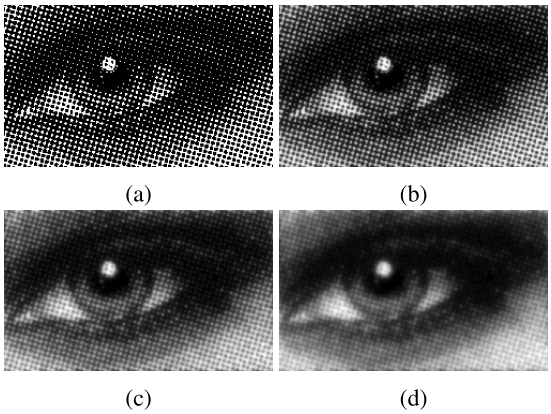


FIGURE 2. A halftone image can be perceived as a continuous-tone image due to the HVS properties. (a) Halftone image, (b) halftone image after applying the Näsänen HVS filter [5] with a viewing distance of 40 cm, (c) 80 cm and (d) 120 cm.

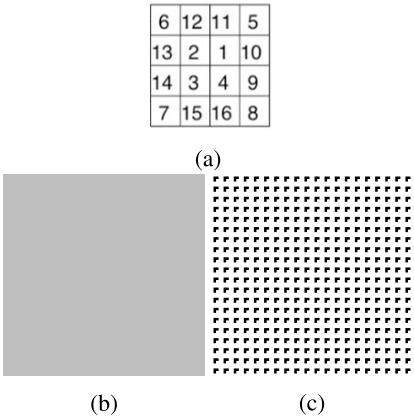


FIGURE 3. The result of the applying a screen (a) on a continuous tone image with constant value (b) with gray level = $\frac{3}{16}$ for all pixels is shown in (c). In this case, all pixels smaller or equal to three in each 4 × 4 non-overlapping region become black.

threshold matrix $S^{k\times k}$ is tiled over a scaled multi-tone image I [m, n], and pixel by pixel thresholding is performed (2).

$$\tilde{I}[m,n] = \begin{cases} 0 & \text{if } I[m,n] \geq S\,[m(mod\ k), n(mod\ k)] \\ 255 & \text{otherwise} \end{cases} \quad (2)$$

The area of the screen matrix k^2 is equal to the maximal number of tones that the screen emulates. Every threshold value $s_{m,n}$ in the screen matrix $S[m, n]$ satisfies $s_{m,n} \in [1, k^2]$. The multi-tone image $I[m, n]$ is scaled such that each pixel will satisfy $i_{m,n} \in [1, k^2]$ pre-screening [3]. The elements of $S[m, n]$ indicate the order in which pixels change tone as a function of the input tone. An example of a screen threshold matrix is shown in Fig. 3a. In this example, $k = 4$; thus, the number of emulated tones of this screen is 17, counting both full white (no thresholds exceeded) and full black (all thresholds exceeded). Fig. 3c shows the result of applying the screen on a continuous-tone image as a function of the input tone (gray level= $\frac{3}{16}$). (The continuous-tone image I [m, n] is scaled such that $i_{m,n} \in [1, 16]$.)

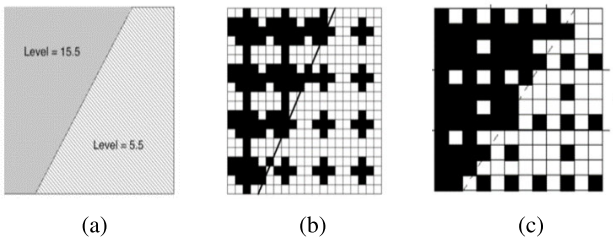
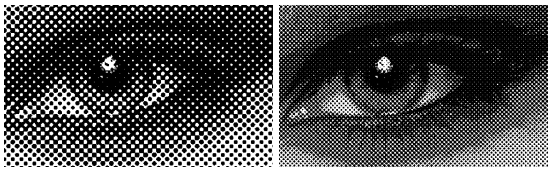


FIGURE 4. Dispersed dot screen vs. clustered dot screen. (a) continuous tone image, (b) image after clustered dot screening, (c) image after dispersed dot screening. Note that (b) has higher spatial resolution (smaller pixels) than (c). Although dispersed dot screening renders edges more accurately than the clustered dot screening, it is not applicable for electrophotographic print engines due to the high usage of isolated dots, which are not stable.

The screen is the basic repetitive halftoning unit. In the case of an AM screen, it is composed of halftone cells [1]. It is customary to use a cost function to determine the $S[m, n]$ values within the halftone cell [1]. Dispersed dot cost functions tend to distribute pixels away from each other with some typical wavelength, λ_{gl}, to create a visually pleasing distribution [2], [33]. Clustered dot screen cost functions distribute dots around the halftone cell center [8]. Dispersed dot screens render large portions of the tonal range with isolated areas of colorant pixels. These screens are applicable in print engines that can produce stable and small isolated ink drops, for example, ink jet-based engines. In the case of electrophotographic engines, the common practice is to use clustered dot halftoning [34]. Fig. 4 shows the result of applying a dispersed dot screen vs. a clustered dot screen on a continuous-tone image. As dispersed dot screens are composed mainly of isolated dots (and not clusters), these screens render edges much more accurately than clustered dot screens. However, dispersed dot screens are not applicable for electrophotographic engines because the electrophotographic marking process is not as stable as the offset or ink jet marking process. What this means is that there is more variation from dot-cluster to dot-cluster, there is more noise due to scattered colorant, and small dot-clusters may not consistently develop. The overall effect is to create the appearance of graininess in the printed image, which customers find objectionable [8]. The ability of a clustered screen to render edges relates to the number of clusters per unit area—the screen resolution [1] (usually measured in LPI).

1) REGULAR SCREEN

In a regular screen [1], the halftone cell size s^2 is proportional to the screen tonal resolution T and inversely proportional to the screen's spatial resolution (known as the screen frequency, or LPI) f_s factored by R^P, the print engine resolution (3) [8]. We define HVS^{min} as the highest frequency noticeable by a typical human observer. A halftoned image is perceived as a continues tone when the screen resolution satisfies $f_s > HVS^{min}$ ($HVS^{min} = 80\ lpi$ for a viewing distance of 20 cm [5], [8]). As f_s increases, more details from the pre-halftoned image are preserved (Fig. 5). However, as T decreases, the



(a) (b)

FIGURE 5. As f_s increases, more details from the pre-halftoned image are preserved. (a) image after AM halftoning with screen of $f_s = 70$ LPI (b) image after AM halftoning with screen of $f_s = 140$ LPI. (The reader is advised to zoom in until the display artifacts that appear in (b) disappear.)

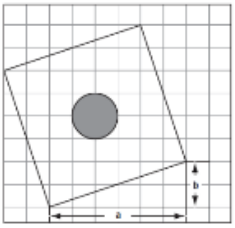


FIGURE 6. The angle of the screen s^{ang} is defined by the right triangle that the halftone cell forms with respect to the horizontal axis.

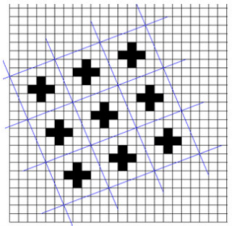


FIGURE 7. The halftone cell legs (a, b) define a rotated square lattice on the pixel grid R^P.

tonal range of the halftoned image deteriorates. The f_s-T trade-off is usually addressed by grouping multiple halftone cell dimensions into a super-cell [1], [8].

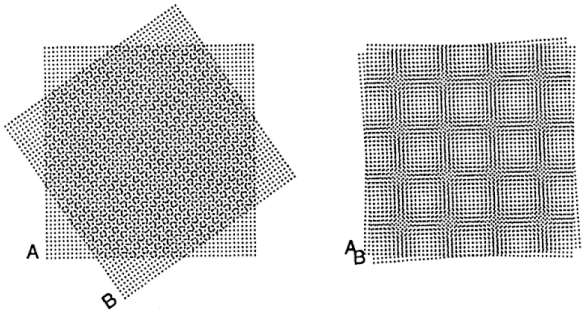
$$f_s = \frac{R^P}{s};\ T = s^2 \tag{3}$$

The angle of the screen s^{ang} is defined by the right triangle that the halftone cell forms (Fig. 6) on the pixel grid R^P. Equation (4) shows the relationships between the halftone cell s, the screen angle s^{ang} and the right triangle legs: (a, b) [1].

$$s^{ang} = \tan^{-1}\left(\frac{b}{a}\right);\quad s = \sqrt{a^2 + b^2} \tag{4}$$

The halftone cell legs define a square lattice on the pixel grid R^P (Fig. 7). We define $\{\vec{v}_i\}_{i=1}^2$ as the basis vectors that span the halftone square lattice. The Fourier spectrum of the screen is also a 2D lattice, the reciprocal lattice, and the frequencies of the screen are spanned by the reciprocal lattice basis vectors $\{f_i\}_{i=1}^2$:

$$\begin{aligned} &\vec{v_1} = (a, b);\quad \vec{v_2} = (-b, a) \\ &\vec{f_1} = \frac{R^P}{a^2 + b^2}(-b, a);\quad \vec{f_2} = \frac{R^P}{a^2 + b^2}(a, b) \end{aligned} \tag{5}$$



(a) (b)

FIGURE 8. When the screen lattices are placed on top of each other, the screens frequencies interfere, and moiré patterns start to emerge. Here, we show two identical screens superimposed with two different relative angles (a) and (b), which significantly influence the appearance of the moiré. In the case that the moiré pattern frequency is low enough (b), that is, $|f^{moire}| < HVS^{min}$, the pattern might be visible, and the print unacceptable.

2) SCREEN SET DESIGN

Color printing requires the design of a screen *set*, where each screen is assigned to one channel (colorant): $\{s_i\}_{i=1}^{\#inks}$ (usually CMYK) [6]. When the screen lattices are placed on top of each other, the screens' frequencies interfere, and patterns start to emerge (moiré) [7], [9]. If a moiré pattern frequency is low enough, that is, $|f^{moire}| < HVS^{min}$, the pattern will be visible, and the print's quality might be unacceptable [8]. (Fig. 8)

The moiré pattern frequency f^{moire} is a linear combination of the screens basis vectors (6), where $\vec{f}_j^{\,i}$ is the j^{th} basis vector of color plane i, α_j^i are integer coefficients, and the sum $\sum \alpha_j^i$ denotes the moiré order [8].

$$\vec{f}^{moire} = \sum_{i=1}^{N}\sum_{j=1}^{2}\alpha_j^i \vec{f}_j^{\,i} \tag{6}$$

Finding a moiré-free screen set requires that the following condition (7) be satisfied [8].

$$|\vec{f}^{moire}| > HVS^{min} \text{ or } |\vec{f}^{moire}| = 0 \tag{7}$$

The most common moiré-free screen set for high resolution analog print engines (offset) is the rosette zero-moiré screen set [1], [4], [8]. This set is considered to be the best all-around compromise for most images and applications as it forms the least objectionable moiré—the rosette patterns [4]. The screens in this set have equal resolution and a 30-degree angle difference (usually 15°/45°/75° for the CMK planes and 0° for the yellow plane) [4]. The zero-moiré condition $|f^{moire}| = 0$ dictates that both $\{|f_i|\}$ and $\{s_i^{ang}\}$ should be carefully pin-pointed to hold the condition. Even a small deviation in the screen angle and resolution will cause severe moiré (the HVS function is most sensitive to low-frequency patterns [5]) [7]. A typical offset screen set has $\{|f_i|\}_{i=1}^4 = 175$ LPI and $\{s_i^{ang}\}_{i=1}^4 = \{0, 15, 45, 75\}$ [4]. Digital print engine screens are quantized to an integer number of pixels and,

TABLE 1. The screen pool of regular screens for $R^P = 812.8$ DPI. The rosette zero-moiré screen, commonly used with offset presses, is not feasible.

Screen#	$a_{[pixel]}$	$b_{[pixel]}$	$f_{s_{[lpi]}}$	$R^{angle}_{[deg]}$	$R^{P}_{[dpi]}$
1	5	2	150	21	812.8
2	5	1	159	11	812.8
3	4	3	162	36	812.8
4	4	2	181	26	812.8
5	3	3	191.57	45	812.8
6	1	4	197.13	75	812.8
7	3	2	225.43	33	812.8

therefore, can produce only a few unique angles and resolutions (i.e., regular screens) [1], [10], [11]. For this reason, designing a rosette zero-moiré screen set with regular screens is not feasible. Table 1 shows the screen pool (available regular screens) with $150 < f_s < 225$ for a print engine with $R^p = 812.8$ DPI (Indigo print engine [35]). It is evident that the pool of regular screens is very limited, and the rosette zero-moiré screen set is not feasible. Generally, any moiré-free screen set is not feasible within this screen pool [7].

3) INCORPORATING IRREGULAR SCREENS IN THE SCREEN SET DESIGN

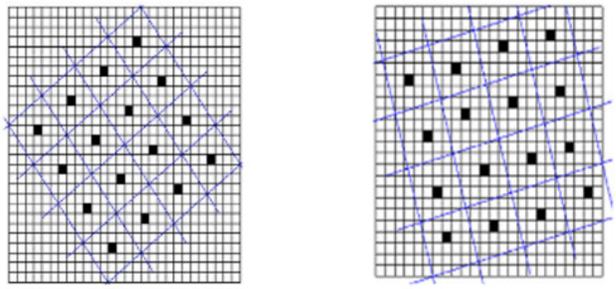
It is customary to use irregular screens [11] to cope with the lack of available screens in low-resolution print engines. In these screens, the halftone cell legs $\{(a^i, b^i)\}_{i=1}^{clusters}$ and the halftone cell centers $\{(x^i_{center}, y^i_{center})\}_{i=1}^{clusters}$ do not fall necessarily on the R^P grid (Fig. 9).

The use of irregular screens dramatically increases the number of available screens per print engine, thus enabling one to design a screen set that satisfies (7) (moiré-free condition) [8]. The screen pool size $|P|$ for $f_s \in [150, 225]$ and $R^p = 812.8$ DPI is given in Table 2 for both regular and irregular screens. The screen pool is constructed by varying the halftone screen basis vector (a, b), such that $f_s \in [150, 225]$, $(a, b) \in \mathbb{Q}$ and the halftone cells can be combined to a zero angle dither array super cell (tile) [8].

In the case of an irregular screen, the lattice basis vectors $\{\vec{v}_i^{\,ireg}\}_{i=1}^2$ can be non-integer, rational numbers $(a, b) \in \mathbb{Q}$ [8]. As pixels must have integer numbers, the halftone cell center positions are rounded to the closest pixels when spanning a halftone grid:

$$\{(x^i_{center}, y^i_{center}\}_{i=1}^{clusters} = span\{round(n_1 \cdot \vec{v}_1^{\,ireg} \pm n_2 \cdot \vec{v}_2^{\,ireg})\}_{i=1}^{clusters} \quad (8)$$

The rounding causes the irregular screen distance between cells to be non-uniform (Fig. 9b); hence, irregular screens are usually much noisier than regular screens [22]. The vast majority (approximately 95% for a typical low-resolution print engine [10]) suffer from disturbing patterns (quantization patterns) that are visible on the print and which customers find to be objectionable (unacceptable screens) [8]. The irregular screen pool for a typical low-resolution print engine is of the order of hundreds of thousands; thus, it is impractical to identify unacceptable screens by printing and visually inspecting all the screens in the pool. Thus unacceptable screens need to be identified in the screen design process pre-printing. This is usually done by a spectrum analysis of the irregular screen. Fig. 10 shows the screen spectrum of an irregular screen vs. a regular screen (equal resolution and angle). The irregular screen spectrum contains additional frequencies that are absent from the regular screen spectrum. These additional frequencies cause quantization patterns to appear in the print. The common practice to incorporate irregular screens in the screen design is to calculate these patterns directly from the reciprocal screen lattice (Fourier representation of the screen lattice vectors (11)) followed by a threshold, rule-based operation that determines if the screen is in the acceptable or unacceptable domain [8], [11]–[13], [10].



(a) (b)

FIGURE 9. The halftone cell centers (x_0, y_0) fall on R^P in the case of a regular screen (a) and do not fall on the pixel grid in the case of an irregular screen (b). As pixels must have integer numbers, the halftone cell center is rounded to the closest pixel.

TABLE 2. Comparison of the size of the screen pool for regular and irregular screens with f_s in different ranges and high and low R^P values.

$f_{s_{[lpi]}}$	$R^{P}_{[dpi]}$	Regular	Irregular
$80 - 150$	812.8	25	11706
	3251.2	155	46050
$150 - 225$	812.8	7	20237
	3251.2	73	79682
$225 - 300$	812.8	4	28168
	3251.2	40	110907

4) SPECTRUM ANALYSIS OF IRREGULAR SCREENS

The spectrum of a regular screen is composed of frequencies $\vec{f}_{n_1,n_2}$, which are a linear combination of the screen reciprocal lattice basis vectors $\{\vec{f}_i\}_{i=1}^2$ and all their harmonics $\{n_i \cdot \vec{f}_i\}$ (9) [7].

$$\vec{f}_{n_1,n_2} = n_1 \cdot \vec{f}_1 + n_2 \cdot \vec{f}_2,\ n_1, n_2 \in \mathbb{Z} \quad (9)$$

In the case of a regular screen, each frequency in the spectrum satisfies $|\vec{f}_{n_1,n_2}| > f_s$, thus setting $f_s > HVS^{min}$ ensures that the screen will not suffer from disturbing visible low frequencies. (See (3) and (5) for the relation between f_s and $\vec{f}_1$ and $\vec{f}_2$.) This is not the case for irregular screens.

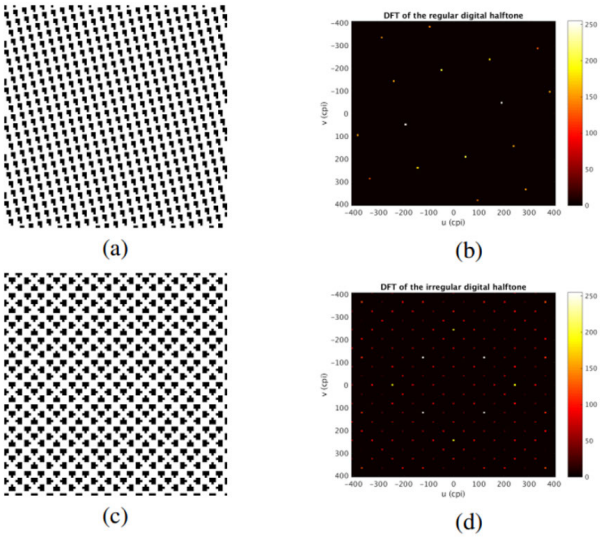


FIGURE 10. Comparison of spectra. Regular screen (a) and its corresponding spectrum (b) and irregular screen (c) and its corresponding spectrum (d). The irregular screen spectrum contains additional frequencies that are absent from the regular screen spectrum and are the cause for visible patterns on the print.

In the case of irregular screens, the rounding effect (8) generates non-uniformities in the pixel locations, and additional frequencies, that is, quantization frequencies, start to appear. Fig. 11 shows an example of the quantization effect for the irregular screen in Table 3. After the rounding, as seen in Fig. 11b, a new low frequency $\vec{q}_i$ appears. $\vec{q}_i$ was not present before the rounding, as observed in Fig. 11a.

The spectrum of an irregular screen is composed of both regular $\vec{f}_{n_1,n_2}$ (9) and irregular $\vec{q}_{n_1,n_2}$ frequencies. The latter are generated due to the residual error of the rounding operation (8), which varies within the screen lattice (Fig. 9). The error function is the residual of the spatial halftone basis vectors $\vec{v}_1^{\,ireg}, \vec{v}_2^{\,ireg}$:

$$\begin{aligned} e(n_1,n_1) &= (n_1\vec{v}_1^{\,ireg} + n_2\vec{v}_2^{\,ireg}) mod 1 \quad n_1,n_2 \in \mathbb{Z} \\ x mod\ 1 &= x - round(x) \end{aligned} \tag{10}$$

$\vec{v}_1^{\,ireg}$ and $\vec{v}_2^{\,ireg}$ can be expressed as irreducible fractions via $p_1, q_1, p_2, p_2 \in \mathbb{N}$:

$$\vec{v}_1^{\,ireg} = \left(\frac{p_1}{q_1}, \frac{p_2}{q_2}\right); \quad \vec{v}_2^{\,ireg} = \left(-\frac{p_2}{q_2}, \frac{p_1}{q_1}\right) \tag{11}$$

Thus, the error function takes the form:

$$e(n_1,n_2) = \left(\frac{n_1p_1q_2 - n_2p_2q_1}{q_1q_2}, \frac{n_1p_2q_1 - n_2p_1q_2}{q_1q_2}\right) mod 1 \tag{12}$$

The error function is a discrete 2D saw-tooth function with fundamental period $T = lcm(q_1,q_2) \cdot \frac{1}{f_s}$ (lcm - least common multiple). The frequencies of the error function, that is, the quantization frequencies $\vec{q}_{n_1,n_2}$, and their corresponding amplitudes $|A|_{n_1,n_2}$, are given by the Fourier series expansion

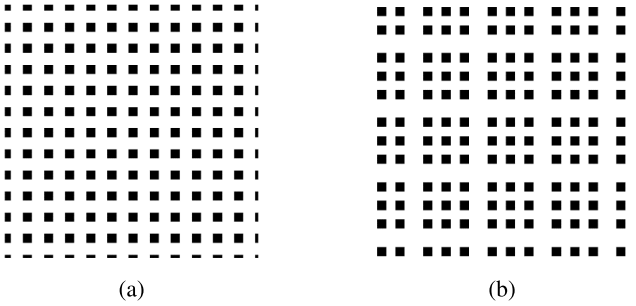


FIGURE 11. An example of the effect of the rounding operation (8) for an irregular screen, introducing quantization frequencies to the screen (a) an irregular screen before the rounding operation (b) an irregular screen after the rounding operation. The latter suffers from a low-frequency quantization pattern. The parameters for this screen are shown in Table 3.

TABLE 3. An irregular screen for $R^P = 812.8$ DPI.

$\vec{v}_1^{ireg}$	$\vec{v}_2^{ireg}$	$\vec{f}_1^{ireg}$	$\vec{f}_2^{ireg}$	$f_{s_{[lpi]}}$	$R^{angle}_{[deg]}$	$R^{P}_{[dpi]}$
$(\frac{7}{3},0)$	$(0,\frac{7}{3})$	$(\frac{3}{7},0)$	$(0,\frac{3}{7})$	348.34	90°	812.8

of a saw-tooth function with period T (13):

$$\begin{aligned} \vec{q}_{n_1,n_2} &= \frac{n_1}{T}\cdot\vec{f}_1 + \frac{n_2}{T}\cdot\vec{f}_2,\ n_1,n_2 \in \mathbb{Z} \\ A_{n_1,n_2} &\propto \frac{1}{n_1+n_2} \end{aligned} \tag{13}$$

Fig. 12 presents the error function in the $\vec{f}_1$ direction for the irregular screen that is given in Table 3. In this case, $T = 7$ and $\vec{q}_{n_1,n_2} = \left(\frac{n_1}{7}, \frac{n_2}{7}\right)$; $A_{n_1,n_2} \propto \frac{1}{n_1+n_2}$

5) THE RULE-BASED APPROACH

Current screen set design is carried out using the rule-based approach. This approach predicts whether a screen will suffer from visible quantization patterns when printed. This approach has two steps: first, the lowest and strongest quantization frequencies are calculated via (13). Then each frequency $\vec{q}^{\,i}_{n_1,n_2}$ is compared with a dedicated harmonic dependant threshold th'_{n_1,n_2}. The condition for screen acceptability is given by:

$$\forall i : \vec{q}^{\,i}_{n_1,n_2} > th'^{i}_{n_1,n_2} \tag{14}$$

The main idea behind this approach is the fact that the HVS behaves as a low pass filter; thus, low quantization frequencies are more visible to the human observer—and, therefore, riskier [5], [32]. As the visibility is also a function of the amplitude, which is inversely proportional to the order of the harmonic (13), the thresholds $th'^{i}_{n_1,n_2}$ are more permissive as (n_1,n_2) increases.

It is not clear how to set the threshold values. As the thresholds are a function of the printing process and the HVS response to the print, the common practice is to set them with an initial guess (based on a simplistic HVS function) and fine-tune them using empirical measures (printing and evaluating the patterns' severity on the print) [1], [10]. For the Indigo print engine [35] ($R^p = 812.8$ DPI), the threshold values that were found to give the best result are given

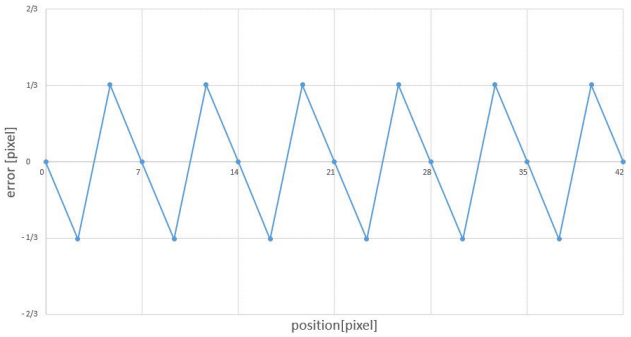


FIGURE 12. Error function of the irregular screen from Table 3 in the $\vec{f}_1$ direction.

in Table 4 [10], [23]. Defining th_i as the period of lowest frequency, that is, the largest period, allowed per harmonic: $th_i = \frac{1}{th'_i}$. The thresholds th'_i are presented in units of the length of the halftone cell *hc*.

IV. PROPOSED METHOD

A. SCREEN SET DESIGN WITH SCREEN CLASSIFIER

We propose a novel method for incorporating irregular screens in the screen set design. The main idea behind our approach is to divide the screen pool P into two non-overlapping classes: P^{acc}, the set of all screens that are acceptable for production, and P^{unacc}, the set of all screens that are unacceptable for production:

$$P = P^{acc} \cup P^{unacc} \tag{15}$$

Thus, the task of differentiating irregular screens for a certain print engine boils down to a two-class classification problem. Therefore, our goal is to find a classifier C that takes as input a screen that is defined by two numbers (f_s, s^{ang}), and classifies it as either acceptable or unacceptable for a given print engine:

$$C(f_s, s^{ang}) \in P^{acc} \; or \; C(f_s, s^{ang}) \notin P^{acc} \tag{16}$$

The main steps of the method are listed below:

1) Construct the screen pool P for a given print engine.
2) Build a screen classifier $C(f_s, s^{ang})$ using ML methodology and psycho-physical experiments.
3) Apply the screen classifier to the screen pool.
4) Define a target screen set resolution range (3).
5) Find a set of three points in the screen pool that satisfy:
   - *Quantization-free:* $\forall i \; C(f_s^i, s_i^{ang}) \in P^{acc}$.
   - *moiré-free:* $\{(f_s^i, s_i^{ang})\}_{i=1}^3$ satisfy (7).
6) Assign the set $\{(f_s^i, s_i^{ang})\}_{i=1}^3$ to the colorants $\{C, M, K\}$.
7) Assign the colorant Y as follows: $f_s^y = f_s^k \cdot 1.10 \; s_{ang}^y = \frac{1}{2}\left(s_{ang}^k - s_{ang}^c\right)$.

This method is designed to yield a quantization-free, rosette-like zero-moiré $\{C, M, K, Y\}$ screen set within the target resolution range for a dedicated print engine

TABLE 4. Threshold values found empirically for the Indigo print engine [35] for the rule-based approach in units of halftone cell *hc*.

$th'_1[hc]$	$th'_2[hc]$	$th'_3[hc]$	$th'_4[hc]$	$th'_5[hc]$	$th'_6[hc]$
2.5	3.5	3.5	8	8	8

(as patterns in yellow ink have low visibility, we followed a rule-based approach to design the screen for yellow ink [4]). Fig. 13 shows the result of Step 1 for a typical low resolution $R^p = 812.8$ DPI print engine (the pool is constructed by varying the halftone screen basis vector parameters (a, b), such that $f_s \in [0, 250]$, $(a, b) \in \mathbb{Q}$ and the halftone cells can be combined to a zero angle dither array super cell (tile) [8]). Each point in the 2D plot corresponds to one screen defined by its LPI and angle (f_s, s^{ang}). Blue points correspond to regular screens (which are acceptable by definition), and red points correspond to irregular screens on which the classifier will be imposed.

The success of this method depends on the ability of the screen classifier to distinguish between acceptable and unacceptable irregular screens. In what follows, we will show how to use ML methodology and psycho-physical experiments to build the screen classifier. Fig. 14 shows a schematic flowchart of the proposed method.

B. FOURIER-BASED FEATURE EXTRACTOR

To extract the screen features for the ML classifiers, we calculate a 6-dimensional feature vector $\vec{Q}_s$ (the quantization vector) for each screen. This is done by using (13) in the following way: the 1^{st} feature is the lowest quantization frequency that satisfies $\sum_j |n_j| = 1$ (1^{st} harmonic); The 2^{nd} feature is the lowest quantization frequency that satisfies $\sum_j |n_j| = 2$ (2^{nd} harmonic); we continue to the 6^{th} harmonic (17).

$$\begin{aligned}
Q_s^1 &= argmin_{n_1,n_2}\{|\vec{q}_{n_1,n_2}|\}^{|n_1|+|n_2|=1} \\
Q_s^2 &= argmin_{n_1,n_2}\{|\vec{q}_{n_1,n_2}|\}^{|n_1|+|n_2|=2} \\
Q_s^3 &= argmin_{n_1,n_2}\{|\vec{q}_{n_1,n_2}|\}^{|n_1|+|n_2|=3} \\
&\cdots \\
Q_s^6 &= argmin_{n_1,n_2}\{|\vec{q}_{n_1,n_2}|\}^{|n_1|+|n_2|=6} \\
\vec{Q}_s &= \left(Q_s^1, Q_s^2, Q_s^3, Q_s^4, Q_s^5, Q_s^6\right)
\end{aligned} \tag{17}$$

The quantization vector stores the 6 lowest and strongest quantization frequencies in the screen spectrum; thus, the 6-dimensional space it spans is the most suited for the screen classification. The structure of the quantization vector implies the feature importance: Q_s^i importance decreases as i decreases. This is because the amplitude of the quantization pattern is inversely proportional to the order of the harmonic (13). As the HVS behaves as a low pass filter, we take only the lowest frequency in each harmony [5], [32]. The quantization vector is also used in the rule-based screen set design approach, although it is defined a bit differently from this derivation [10].

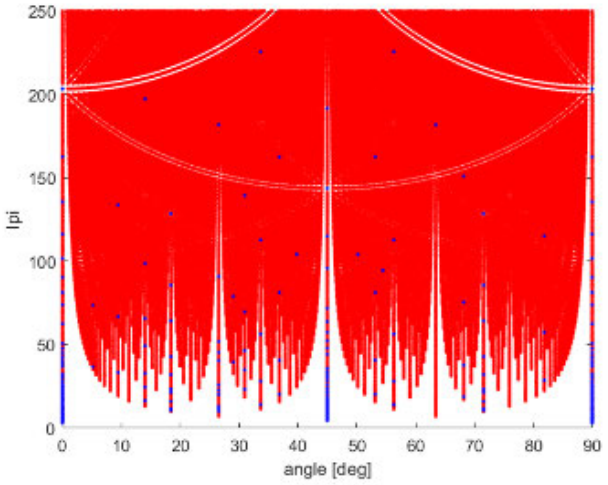


FIGURE 13. The screen pool of the HP Indigo print engine. Constructing this pool is the first step of the proposed method. Each red or blue point (f_s, s^{ang}) in this graph represents an optional screen for this print engine. Blue dots correspond to regular screens, and red dots correspond to irregular screens. In what follows, a screen classifier will be developed and applied to each point-screen in the pool.

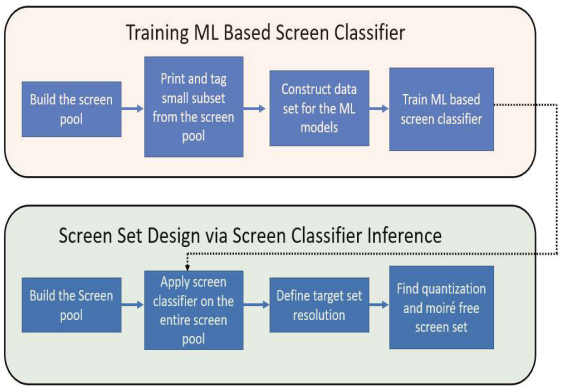


FIGURE 14. A schematic flowchart of the proposed method. The method can be divided into two main parts: training the screen classifier to a small, selected subset of the pool, followed by screen set design via inference of the screen classifier to the entire pool. The typical number of screens for the training is on the order of several hundred while the entire screen pool is on the order of hundreds of thousands.

C. APPLIED MACHINE LEARNING MODELS

It is hard to know in advance which ML model will yield the best screen classifier. Therefore, we tested and compared several ML models [21]. As both the print engine and the HVS are non-linear functions [14], most of the models we explored were non-linear as well. The models that we explored were:

Logistic regression (LG): is a linear model that predicts the probability of a discrete outcome given the input variables. For the screen classifier case, the variables are the screen parameters (usually the quantization vector), and the outcome is the probability that the screen will suffer from visible quantization patterns (unacceptable).

K nearest neighbors (KNN): is a non-linear machine learning model in which data samples are classified according to their distances from other data points. The usage of a KNN classifier makes sense as acceptable screens should be closer to each other in the quantization feature space. The number of nearest neighbors, K, that participated in the voting was set to 5, the default value in the KNN implementation [36].

Support vector machine (SVM): is an ML model in which data can be transformed to a higher-dimensional space, such that linear separation can be applied by a hyper-plane [16]. Utilizing SVM requires one to determine the best data transformation and to fine-tune the hyper-plane parameters per the classification challenge [18]. We have used an exponential kernel with different preprocessing methods to improve the SVM classifier performance.

Classification and regression decision trees (CART): are greedy non-linear ML models in which the target cost function, the Gini purity index, is minimized in the learning process. These models were preferred for several reasons: First, the algorithm finds and sorts the features according to importance during the learning process via the purity metric. Second, CART models are highly non-linear models, thus perform well even with challenging classification problems (given enough data). Third, if the tree is shallow enough, the CART model acts as a white box model; that is, the prediction mechanism can be interpreted, and insights can be gained. We have used simple visualization of the trees to interpret the classifying mechanism of the ML model. One main disadvantage of CART models is their tendency to overfit, especially in cases where the models are too complicated. In our case, the trees were relatively shallow, so we did not have to limit the tree depth to regulate the tree. [37]. We preferred to use this relatively simple model over more sophisticated ones (e.g., XGBoost, random forest and deep learning) because it requires less data, is explainable, and (as we will show) yields good results.

D. EVALUATION METRICS

We divided the data into 80% training and 20% testing sets. As usual, the training set was used to explore different ML models and to fine-tune hyper parameters while the testing set was saved for the final evaluation of the selected and fine-tuned ML model. We performed 10-fold cross validation [37], in which the training set was divided (again) into training and validation sets 10 times, and the ML models were compared between the different folds [37]. Accuracy, which is the number of correct predictions out of the total number of predictions, was used to compare the performance between ML classifiers. Models that showed low variation (low variance) between the different folds and high accuracy were preferable. In our case, false positives (FPs) are screens that were tagged unacceptable by subjective evaluation and acceptable by the classifier, false negatives (FNs) are screens that were tagged acceptable by subjective evaluation and unacceptable by the classifier, true negatives (TNs) are screens that were tagged unacceptable by subjective evaluation and unacceptable by the classifier, and

true positives (TPs) are screens that were tagged acceptable by subjective evaluation and acceptable by the classifier. A confusion matrix was built for the testing set that provides detailed information on the ML classifier performance. The elements of the matrix are FP, FN, TN and TP, where high values in the diagonal elements of the confusion matrix indicate a high model performance [15].

E. DATA STRUCTURE

Obtaining large amounts of real data is challenging because generating the ground truth of the screen class required access to a print engine, installing dedicated software patches, printing several target jobs per screen and tagging each screen by a human observer. We estimate approximately 0.5 day per 10 screens. Thus, our first step was to explore the ML models via simulation. In the simulation, the screen class (acceptable/unacceptable) was tagged via the legacy rule-based model (Section III-B5). In the second step, dedicated experiments and physical evaluations were done to tag the screen class.

1) SIMULATION DATA STRUCTURE

We did the following to generate the data set for the simulation: the screen geometry parameters a and b (screen basis vectors elements as shown in (5)) were varied between [0, 20] with steps of 0.03 to create a screen pool of approximately 300,000 screens in the ranges of $f_s \in [0, 300]$ *lpi* and $s^{ang} \in [0°, 90°]$. Then, a smaller pool of 11,500 screens was constructed with uniform distribution within $f_s \in [0, 300]$ *lpi* and $s^{ang} \in [0°, 90°]$. Next, for each screen, the 6-dimensional quantization vector (6 lowest and strongest quantization frequencies) was calculated via (17). To determine the screen class, we followed the legacy rule-based approach [8], [10], [12]: if the six lowest quantization frequencies of a given screen were higher than Indigo's empirical thresholds (Table 4), the screen was tagged as class 1 (acceptable); otherwise, it was tagged as class 0 (unacceptable). Two types of vector features were used to construct the data set. The first feature vector is the 6-dimensional quantization vector that was calculated (per screen) as described above. The second feature vector is a 4-dimensional vector that simply contained the screen basis vector $\vec{v}_1, \vec{v}_2$ elements (the vectors were calculated via (5)). Table 5 shows the first four data samples of the data set with the 6-dimensional quantization vector as the feature vector. Each row corresponds to one screen: $[Q_s^1, Q_s^2, ..Q_s^6]$ are the features, and the class is the screen tag generated according to the methods described above.

2) EXPERIMENT DATA STRUCTURE

To create data set for the ML training, we conducted psychophysical experiments in which 443 designated screen threshold matrices (2) were installed on the digital HP Indigo 10000 print engine [35], printed and evaluated. It was not feasible to use more samples due to limited printing resources. For that purpose, we created a pool of screens geometries in a way similar to that presented in the previous section. The screen geometry parameters a and b (screen basis vectors elements as shown in (5)) varied between [0, 20] with steps of 0.03 to create a screen pool of approximately 300,000 screens in the ranges of $f_s \in [0, 300]$ *lpi* and $s^{ang} \in [0°, 90°]$. Then, a smaller pool of 443 screens was constructed out of the 300,000 pool (this is the maximum number of screens we were able to print) with uniform distribution within $f_s \in [0, 300]$ *lpi* and $s^{ang} \in [0°, 90°]$. The screens' distributions are shown in Figs. 15a-15b. The class of a screen (ground truth) was found as follows: For each screen in the pool $\{(f_s^i, s_i^{ang})\}_{i=1}^{443}$, the screen geometry parameters were used as an input to the Indigo threshold screen generator that yielded a threshold matrix according the input parameters [4], [8], [35]. Each threshold screen matrix was installed on the target HP Indigo press [35]. Once the threshold matrix screen was installed on the Indigo print engine, a dedicated target job was printed once using that threshold screen matrix. The job was composed of one page that contained 30 $5.5 \times 5.5\ cm^2$ patches with values of K (monochrome black ink) increasing from 0% to 95% coverage. Three human subjects (HP Indigo engineers) visually inspected and evaluated each printed job. A screen was tagged as class 1 (acceptable) if all the subjects agreed that *none* of the 30 patches contain visual quantization patterns. Otherwise, the screen was tagged as class 0 (unacceptable). The results of the screen ground truth tagging are presented in Fig. 15c. Of the 443 screens evaluated, 100 were judged to be unacceptable, and 343 were judged to be acceptable. The f_s and s^{ang} distributions are uniform within the target ranges by design (Figs. 15a,15b). The features are standardized to have zero mean and standard deviation of one. The tagged data distribution, that is, the ground truth, reveals that our data are imbalanced (Fig. 15c). As stated earlier, the number of screens classified by the subjects as belonging to class 1, that is, acceptable and free of visible quantization frequencies, is 343, while only 100 samples belong to class 0. To balance the data set, we randomly duplicated samples from the two classes such that: $N^{class=0} = N^{class=1} = 350$. After the duplication, $N^{real} = 700$. In addition to standardization of the features, we have used two more preprocessing methods: scaling, where the data are scaled to the range [0–1]; and normalization, where the data are transformed such that each sample will have a unit norm length.

TABLE 5. Four samples from the screen data set used to train the ML models. Each row corresponds to one screen. $[Q_s^1, Q_s^2, ..Q_s^6]$ is the feature vector of each screen (quantization vector, in halftone-cell units). "Class" is the screen class. For the 300,000 screen data set, tagging was done by simulation using the rule-based model. For the 433-screen data set, tagging is based on subjective evaluation.

N	Q_s^1	Q_s^2	Q_s^3	Q_s^4	Q_s^5	Q_s^6	Class
1	1.76	2.19	3.1	1.9	3.5	3.7	1
2	2.2	3.42	1.8	2	5.7	2.4	0
3	1.78	2	2.8	1.7	4	4	0
4	1.87	2.82	3.9	4.9	2.8	2.2	1

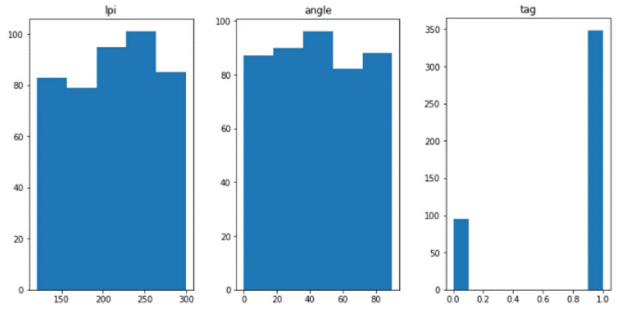


FIGURE 15. Screen data distribution for the 443 screens that were evaluated by the human subjects. (a) Screen resolution f_S [lpi], (b) Screen angle s^{ang} [deg], (c) Tagged data – ground truth (0 = unacceptable, 1 = acceptable).

V. RESULTS

A. SIMULATION RESULTS

We randomly chose 200 samples from the simulated data set (Section IV-E), $100 \in class = 1$ and $100 \in class = 0$, scaled the features to be within the range $\in [0, 1]$ and fed them to the following ML classification models: support vector machine (SVM), logistic regression (LR), linear discriminant (LDA), K-nearest neighbors (KNN) and classification and regression trees (CART). Each model was applied with the default set of hyperparameters [36], and with the feature vectors pre-processed using standardization. Table 6 shows the classification accuracy performances and the variance between the different folds after applying 10-fold cross validation. This result implies that the binary decision tree (CART) is the best model for our challenge. In addition, a binary decision tree has another important advantage: in case the tree is small enough, its visualization might provide important insights [37]. A visualization of the tree T^{200}_{quant} is given in Fig. 16. Each leaf node contains the following information: threshold value of the feature Q_i, the Gini index—the cost function grade [37], the total number of samples that reached the node, the number of samples that reached the node that belong to class 0/class 1: the $value = [N^{\text{class }0}, N^{\text{class }1}]$, node classification: pass/fail stands for class 0/class 1 (the final classification is set, by the leaf at the very bottom node). The features are ordered in decreasing order of the Gini grade; that is, important features appear at the top of the tree while less important features appear at the bottom. In our case, the features have a physical interpretation: The i^{th} feature $\vec{Q}^i_s$ corresponds to the i^{th} harmonic (Section IV-B). Thus the importance of the features decreases as i increases. Indeed, the tree T^{200}_{quant} was able to capture part of this behavior.

The thresholds the tree found for the features presented in each node correspond to the threshold empirical values th'_i that were shown in Section III-B5. The ability of T^{200}_{quant} to restore the thresholds is given in Table 7. It is clear that the model was able to restore the first threshold value th'_1 (the most important feature) and found relatively close values for th'_2, th'_4. The thresholds th'_3, th'_5 were quite different from the simulation thresholds, and the tree did not use the 6^{th} feature. Thus, the accuracy on classification tree T^{200}_{quant} is good, at 94%, and it can restore the threshold vector to a certain extent. Could we do better? To answer this question, we re-trained the CART with 500 samples and created a new tree: T^{500}_{quant}. The accuracy obtained with T^{500}_{quant} is 99%. The tree visualization is given in Fig. 17. The ability of T^{500}_{quant} to restore the thresholds is given in Table 8. Not only is the T^{500}_{quant} accuracy higher, but it also captures the feature importance

TABLE 6. Results of applying different ML models as screen classifiers with the simulated data set of 100 samples from each class (total of 200 samples). The data was preprocessed using standardization (the features were standardize to have zero mean and standard deviation of one).

ML model	mean accuracy	variance
LR	0.66	0.05
SVM	0.90	0.02
LDA	0.51	0.03
KNN	0.91	0.01
CART	0.94	0.015

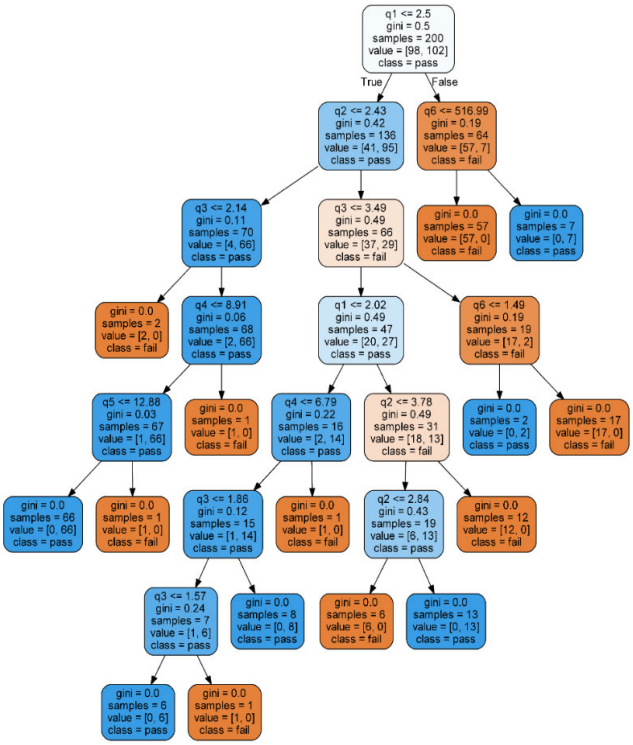


FIGURE 16. Visualization of the T^{200}_{quant} tree.

TABLE 7. Empirical thresholds that were used by the rule-based Fourier model for the simulation vs. T^{200}_{quant} class 1 (blue nodes) threshold values (q_i) taken from the left-most side of the tree in Fig 16. The level of the node (depth) corresponds to the importance of the threshold in the classification. The threshold values are in halftone-cell units.

	th'_1	th'_2	th'_3	th'_4	th'_5	th'_6
Simulation thresholds	2.5	3.5	3.5	8	8	8
T^{200}_{quant} Restored	2.5	2.4	2.14	8.9	12.8	-

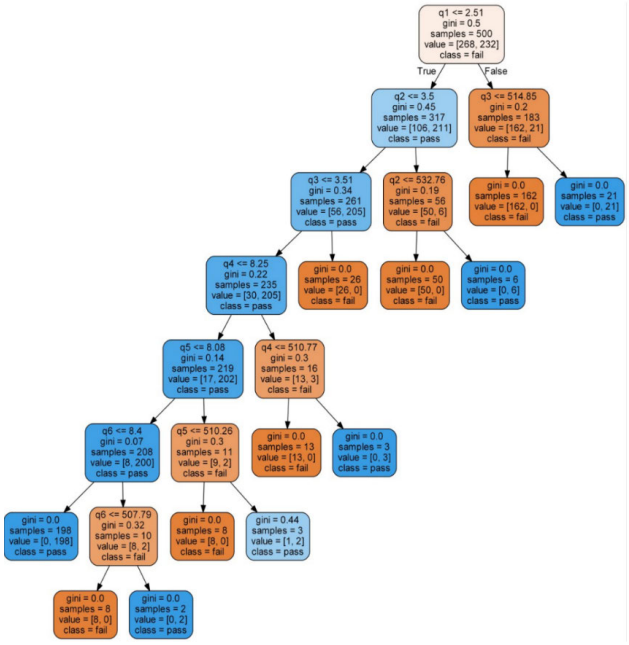


FIGURE 17. Visualization of the T^{500}_{quant} tree.

TABLE 8. T^{500}_{quant} restored thresholds vs. the empirical threshold that were used by the rule-based Fourier model for the simulation. The threshold values are in halftone-cell units.

	th'_1	th'_2	th'_3	th'_4	th'_5	th'_6
Simulation thresholds	2.5	3.5	3.5	8	8	8
T^{500}_{quant} Restored	2.51	3.5	3.51	8.25	8.09	8.4

and restores the thresholds quite accurately. Although both T^{200}_{quant} and T^{500}_{quant} act as good classifiers, T^{500}_{quant} is preferable, as it acts as a white box model and fits perfectly to our physical model. What would happen if we used the screen lattice vectors instead of the quantization vector as the feature vector? In this case, the model should learn the features from the data; hence the number of samples in the data should be larger. To understand how much larger, we did the following: a new data set was created with four features: $v^x_1, v^y_1, v^x_2, v^y_2$, where $\vec{v}^x_1, \vec{v}^x_2$ are the screen lattice basis vectors (5) (in fact there are only two independent features because the vectors are orthogonal and have the same length). The screen class was simulated as before (Section IV-E1). Table 9 shows the first four data samples from this new data set.

Next, we fed the CART with 500 samples from this new data set ($250 \in class = 1$ and $250 \in class = 0$) and developed a new tree $T^{500}_{lattice}$. This tree obtained an accuracy of 88%, which is lower than T^{200}_{quant} or T^{500}_{quant}. To determine the number of samples needed to reach the accuracies achieved with T^{200}_{quant} and T^{500}_{quant}, we increased the number of samples N and re-trained the CART. The results are presented in Fig. 18.

About 5000 samples are needed to obtain 95% with $T^{N}_{lattice}$, which is not feasible in our case. In addition, the trees $T^{N}_{lattice}$ are much more complicated than T^{200}_{quant} and T^{500}_{quant}, and do not provide insights about the model behaviour. A comparison

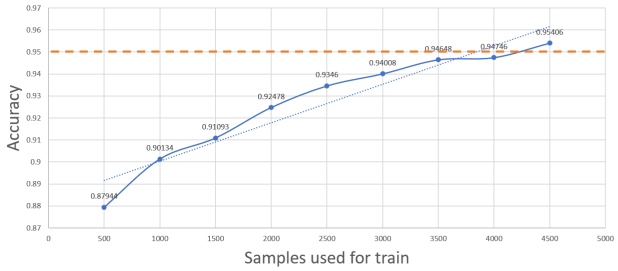


FIGURE 18. Accuracy of the CART model as a function of N using the screen basis vectors as the feature vector applied to the simulation data set. Achieving accuracy of 95% using this feature vector requires more than 5000 samples.

TABLE 9. First four samples of the data set with only the screen lattice basis vectors as features. Each row corresponds to one screen. The feature vectors are the screen lattice basis vectors, $\vec{v}_1, \vec{v}_2$ of each screen in units of pixels *pix*, and the screen class, which is simulated by tagging each screen as one of the two classes via the rule-based model.

N	v^x_1	v^y_1	v^x_2	v^y_2	Class
1	4.5	4.4	-4.4	4.5	1
2	2.4	4.1	-4.1	2.4	1
3	6.3	5.5	-5.5	6.3	0
4	2.2	2.6	-2.6	2.2	1

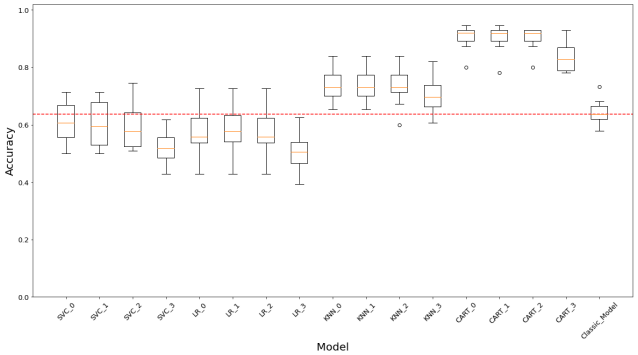
TABLE 10. Accuracy comparison between the two different models that were obtained by training with the quantization vector and the lattice basis vectors, respectively, as a function of the number of samples N. Using the quantization frequencies as feature vector dramatically reduced the required number of samples needed to reach an accuracy of 95%.

	$N = 200$	$N = 500$	$N = 5000$
T^{N}_{quant} accuracy	94%	99%	99.7%
$T^{N}_{lattice}$ accuracy	82%	88%	95%

between the models as a function of feature vectors and N is presented in Table 10. The comparison validates that using the quantization frequencies as the feature vector dramatically reduced the required number of samples needed to reach an accuracy of 95%. We conclude that the CART model with the quantization vector as the feature vector requires several hundred samples to reach high performance as a screen classifier on the simulated data. Next, we will use the ML methodologies to build a screen classifier using real data, that is, data that were tagged via visual inspection.

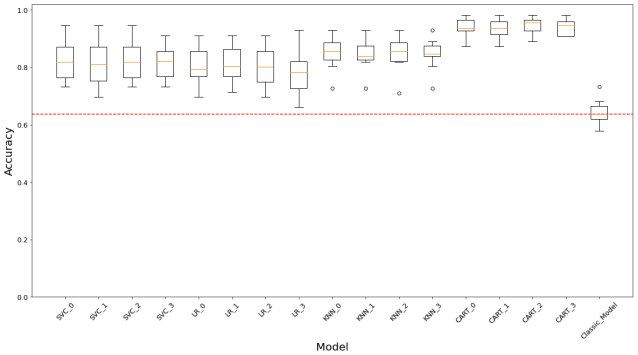
B. EXPERIMENTAL RESULTS

Next, we evaluated the performance of the ML models on the experimental data set (based on subjective evaluation, Section IV-E2). The following models were checked: logistic regression (LR), support vector machine (SVM), K-nearest neighbors (KNN) and classification and regression tree (CART). Each model was trained with the default set of hyperparameters [36] four times: three times, each with a different preprocessing operation and one time without preprocessing. We divided our data into 80% training and 20% testing and used 10-fold cross validation on the training set



Model	mean accuracy	Variance
classic model	0.64	0.04
SVC_2	0.6	0.05
LR_1	0.54	0.05
KNN_0	0.75	0.06
$CART\_0$	0.82	0.06

FIGURE 19. Accuracy comparison of ML classification models for predicting acceptable and unacceptable irregular screens applied to the experimental data set. The feature vector that was used is the screen basis vectors. *CART*_0 gave the best results with an accuracy of 82%.



Model	Mean accuracy	Variance
classic model	0.64	0.04
SVC_3	0.79	0.04
LR_1	0.78	0.05
KNN_1	0.82	0.03
$CART\_3$	0.95	0.02

FIGURE 20. Summary of the ML model results with the quantization vector as the feature vector applied to the experimental data set. All the models have better performance than the rule-based Fourier model. *CART*_3 achieved the best results with an accuracy of 95%.

to evaluate the models. Later, the best model was applied to the testing set to validate generalization to new data. The accuracy of each model with the vector lattice-based features when evaluated on the validation set with 10-fold cross-validation is presented in Fig. 19. For each model and preprocessing method, a box plot summarises the model performance [21], followed by a table showing the results with the best preprocessing method for each ML method. The line in each box represents the median of the accuracy. The lower and upper edges represent the 25^{th} and 75^{th} percentiles, respectively. The whiskers show extreme values for accuracy. The following conventions for the preprocessing methods are used: 0- no preprocessing, 1- re-scaling, 2- standardization and 3- normalization. The dashed red line in Fig. 19 indicates the accuracy of the rule-based model on this data set. The dots indicate outliers.

Next, we retrain all the models with the quantization vector as the feature vector. Fig. 20 shows a comparison of the different ML models' accuracy with the four different preprocessing methods, based on 10-fold cross-validation applied to the validation data set. The red line indicates the performance of the classic rule-based model on this data set.

Using the quantization vector as a feature vector improved the performance of all the ML models. With this feature vector, all the models outperformed the rule-based model. The best results of each model are summarized in the table at the bottom of Fig. 20. These results agree with the simulation results (Section V-A): using the quantization vector as a feature vector with a binary decision tree (*CART*_3) and preprocessing by normalization is the preferred configuration with an accuracy of 95% and variance of 2%.

Next, we applied the trained CART classifier T^{443}_{quant}, with preprocessing by normalization on the testing set, which was not used until this part. The confusion matrix is presented in

TABLE 11. The confusion matrix of the T^{443}_{quant} model (*CART*_3) on the testing set. *gt* denotes the ground truth class, and *p* is the class prediction by the model. The accuracy for the T^{443}_{quant} model on the testing set was 94%.

	$p=0$	$p=1$
$gt=0$	67	4
$gt=1$	5	64

Table 11. The accuracy for the T^{443}_{quant} model on the testing set was 94%, whereas the accuracy of the classic model on the testing set was 64%. As the CART performance of the test set is very similar to the cross-validation results of the training set, we conclude that the model generalizes well to new data. The maximal depth of the tree was chosen to be 25 after a grid search [21] on the training set.

The Tree T^{443}_{quant} is more complicated than the trees found in Section V-A (Fig. 21). Pruning the tree for a simpler model [21] deteriorates the model performance. Thus, we conclude that this is the simplest, highest performance tree for this data set. The fact that the T^{443}_{quant} is more complicated than the trees found with the simulation-based data set (Section V-A) can be explained as follows: First, our rule-based model is too simple and does not capture the complexity of the problem. Second, we were not able to produce a sufficiently large number of samples. T^{443}_{quant} can still be interpreted as being relatively small. The tree nodes represent physical quantities: frequency thresholds and their corresponding harmonics.

1) DESIGNING A SCREEN SET WITH THE SCREEN CLASSIFIER

Next, we followed the screen design steps from Section IV-A.

1) We constructed the screen pool by varying the halftone screen basis vector (a, b), such that $f_s \in [0, 250]$, $(a, b) \in \mathbb{Q}$. The result of this step is shown in Fig. 13.

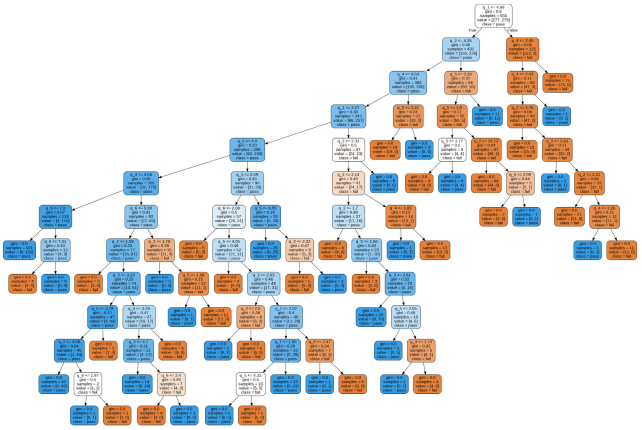


FIGURE 21. The tree T^{443}_{quant}: This tree is more complicated than the trees obtained with the simulated data but still small enough to be interpreted.

TABLE 12. Result of applying the classification function *CART_3* with tree $T_{quant}{}^{433}$ to the set of irregular screens. The fraction of acceptable irregular screens is only 7%.

	Acceptable irregular	Unacceptable irregular	Total irregular
screens	7063	100,651	107,727

2) The tree T^{443}_{quant} that was found in the previous part is used as the screen classifier $C(f_s, s^{ang})$.
3) We applied $C(f_s, s^{ang})$ on the screen pool. The result is presented in Fig. 22: blue points correspond to regular screens, that is, $C(f_s, s^{ang})) \in P^{acc}$ by definition. Red points correspond to unacceptable irregular screens, that is, $C(f_s, s^{ang}) \notin P^{acc}$. Green points corresponds to acceptable irregular screens, that is, $C(f_s, s^{ang}) \in P^{acc}$. Table 12 shows the result quantitatively of the classification function applied to the set of irregular screens: there are only 7% acceptable irregular screens out of 107,727 possible screens.
4) We refined the screen resolution range; we used $f_s \in [225, 255]$ *lpi*.
5) A dedicated software was used to swipe the screen pool within the f_s range to find screen triplets $\{(f_s^i, s_i^{ang})\}_{i=1}^3$ that satisfy both the quantization-free condition: $\forall i \;\; C(f_s^i, s_i^{ang}) \in P^{acc}$ and the moiré-free condition 7).
6) {C, M, K} colorants were assigned to the screen set found in the previous part. The screen set is presented in Table 13 (the Y colorant was found as described in Step 7 in Section IV-A).

This screen set was tested on an HP Indigo print engine [35] and proved to be quantization-free and moiré-free with high print quality features. An attempt to design such a screen set with the legacy rule-based approach resulted in visible quantization frequencies (Fig. figure23) and thus, inferior print quality. (We chose a challenging f_s^i range. As f_s^i increases, the quantization patterns become much stronger [12], [13].)

VI. DISCUSSION

The simulation results showed that ML models can be used as screen classifiers with high accuracy for the simulated data.

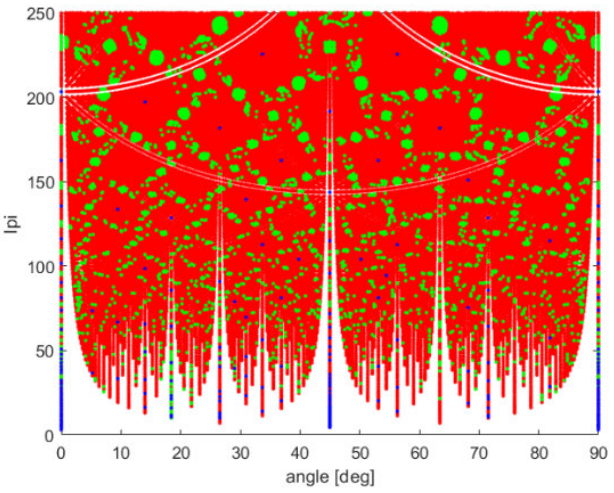


FIGURE 22. Result of applying the classification function T^{443}_{quant} with the quantization vector as features for $R^P = 812.8$ DPI. Blue points correspond to regular screens, red point correspond unacceptable irregular screens, green points correspond to acceptable irregular screens.

TABLE 13. High resolution $f_s = 230$ LPI rosette zero-moiré screen set for a low-resolution print engine $R^P = 812.8$ DPI that was designed using the proposed method. The screen set is moiré- and quantization-free (only green points in Fig. 22 for each screen in the set).

Ink	$f_s[lpi]$	$R^{ang}[deg]$
C	230.3	15
M	230.3	75
K	230.3	45
Y	253.5	30

The CART model gave the best results in the spot checking. The CART model showed much better performance when given smart features (the quantization vector), but even simpler features (screen basis vectors) gave reasonable results. The performance of the CART model as a function of the number of samples was also explored. We showed that with smart features, 500 samples were required to achieve an accuracy of 95%, whereas achieving similar results using simpler features required several thousand samples. The CART model was able to restore the threshold values that were used to tag the data by the simulation and was also able to learn the features' importance (low and strong frequencies are more important). *The experimental results* also showed that ML models can be used as screen classifiers with high accuracy for the data set that was evaluated by the human subjects. Even with simple features, both CART and KNN classifiers outperformed the classic rule-based method (with Indigo empirical thresholds), while SVC and LG were inferior. Using smart features boosted the performance of all the ML models above that of the classic model. The CART (the best model with both features) had an accuracy of 95% in screen classification, whereas the classic rule-based approach achieved 66% on the same data set. We validated the classifier performance on a testing set, which yielded similar results.

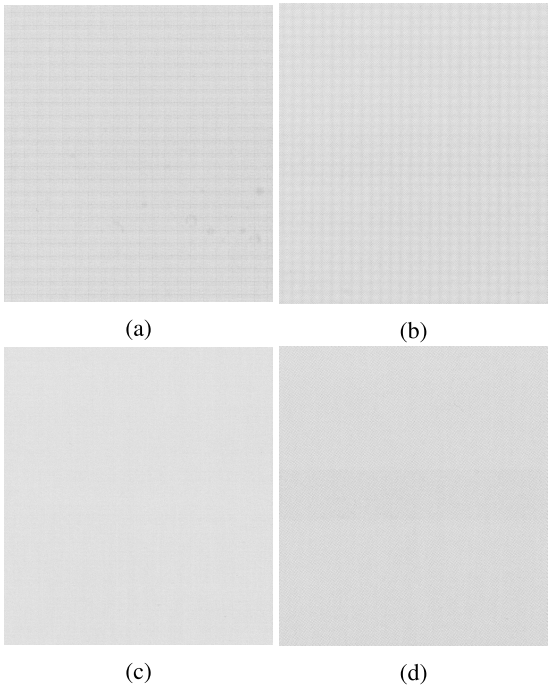


FIGURE 23. A comparison between the proposed method and the current legacy rule-based method applied to HP Indigo print engines. Each image shows a scanned patch of a halftone screen printed with the target printer [35], with a constant gray level of 25%. Attempting to design a screen set with the current approach resulted in visible quantization frequencies: (a), (b) screens that were found to be acceptable by the legacy rule-based method and unacceptable by the proposed method. (c), (d) screens that were found to be acceptable by the proposed method and unacceptable by the legacy rule-based method.

As the CART tree was relativity shallow, it was interpretable, and we extracted the frequencies thresholds the tree used to perform the screen classification. *The screen design method with classifier inference* showed that only **7%** out of approximately 100,000 screens are quantization-free. A software program was used to find a quantization-free screen set (out of the 7% quantization-free screens that were found) that also satisfies the moiré-free conditions. Although we chose a challenging screen set resolution target, the proposed method yielded a quantization-free screen set. Using the rule-based approach to design a similar screen set resulted in a screen set that suffered from visible quantization patterns.

VII. CONCLUSION

In this study, we have shown that machine learning methods can be used to construct a screen classifier that predicts (with high accuracy) whether a screen set will suffer from visible quantization patterns when printed on a specific print engine. The method to combine the classifier in the screen design process was introduced, and a quantization-free screen set was designed. Although the classifier we found fits the HP Indigo print engine, the approach can be leveraged to any digital print engine. In this case, a new classifier will have to be constructed, and the experimental part will need to be revisited using the target print engine. One disadvantage of this method is that it requires tedious data collection, which in some cases can be challenging. This method can be further extended to include chrome, that is, to take into account the effect color has on pattern visibility (this study included only black ink). Another possible extension is to build a screen *set* classifier to predict whether a screen set will produce visible moiré patterns *per print engine*. Similar to this study, it will require an experimental part with the designated print engine. The quantization-free screen set we found in this study is now under qualification, and we plan to release it in one of the future HP Indigo products.

REFERENCES

[1] D. L. Lau and G. R. Arce, *Modern Digital Halftoning*, vol. 1. Boca Raton, FL, USA: CRC Press, 2018.
[2] P. Li and J. P. Allebach, "Tone-dependent error diffusion," *IEEE Trans. Image Process.*, vol. 13, no. 2, pp. 201–215, Feb. 2004.
[3] T. M. Holladay, "Electronic halftone screening," U.S. Patent 4 185 304, Jan. 22, 1980.
[4] P. Fink, *Postscript Screening: Adobe Accurate Screens*. Upper Saddle River, NJ, USA: Prentice-Hall, 1992.
[5] F. A. Baqai and J. P. Allebach, "Computer-aided design of clustered-dot color screens based on a human visual system model," *Proc. IEEE*, vol. 90, no. 1, pp. 104–122, Jan. 2002.
[6] J. A. C. Yule, *Principles of Color Reproduction, Applied to Photomechanical Reproduction, Color Photography, and the Ink, Paper, and Other Related Industries*. New York, NY, USA: Wiley, 1967.
[7] I. Amidror and R. D. Hersch, "Analysis of the superposition of periodic layers and their moiré effects through the algebraic structure of their Fourier spectrum," *J. Math. Imag. Vis.*, vol. 8, no. 2, pp. 99–130, 1998.
[8] T. Frank, O. Haik, A. Jumabayeva, J. P. Allebach, and Y. Yitzhaky, "New design for compact color screen sets for high-end digital color press," *IEEE Trans. Image Process.*, vol. 29, pp. 3023–3038, 2020.
[9] Y.-Y. Chen, T. Kashti, M. Fischer, D. Shaked, R. Ulichney, and J. P. Allebach, "The lattice-based screen set: A square N-color all-orders Moiré-free screen set," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1873–1886, Apr. 2016.
[10] N. Mosenson, "Quantization frequencies in AM screens," *Proc. SPIE*, vol. 2009, pp. 371–372, Jan. 2009.
[11] C. Tang, A. Veis, R. Ulichney, and J. Allebach, "Irregular clustered-dot periodic halftone screen design," *Proc. SPIE*, vol. 9015, Feb. 2014, Art. no. 90150S.
[12] A. Jumabayeva, Y.-T. Chen, T. Frank, R. Ulichney, and J. Allebach, "Design of irregular screen sets that generate maximally smooth halftone patterns," *Proc. SPIE*, vol. 9395, Jan. 2015, Art. no. 93950K.
[13] A. Jumabayeva, T. Frank, Y. Ben-Shoshan, R. Ulichney, and J. Allebach, "Single separation analysis for clustered-dot halftones," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 4383–4387.
[14] P. Goyal, M. Gupta, C. Staelin, M. Fischer, O. Shacham, T. Kashti, and J. Allebach, "Electro-photographic model based stochastic clustered-dot halftoning with direct binary search," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 1721–1724.
[15] R. Tibshirani, T. Hastie, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer Series in Statistics), vol. 1, 2nd ed. Cham, Switzerland: Springer, 2016.
[16] P. Harrington, *Machine Learning in Action*. New York, NY, USA: Simon and Schuster, 2012.
[17] G. L. Serrano, *Grokking Machine Learning*, vol. 1. Shelter Island, NY, USA: Manning Publications, 2021.
[18] P. Harrington, *Machine Learning in Action*, vol. 1, 1st ed., P. Harrington, Ed. Shelter Island, NY, USA: Manning Publications, 2012.
[19] C. M. Bishop, *Pattern Recognition and Machine Learning*, vol. 1. New York, NY, USA: Springer, 2011.
[20] F. Chollet, *Deep Learning With Python, Manning*. Shelter Island, NY, USA, Nov. 2017.
[21] J. Brownlee, "Machine learning mastery with Python," *Mach. Learn. Mastery Pty Ltd*, vol. 527, pp. 100–120, Apr. 2016.
[22] Y.-T. Chen, T. Kashti, T. Frank, R. Ulichney, and J. P. Allebach, "Fourier-based analysis of regular and irregular periodic clustered-dot halftone textures," in *Proc. Int. Congr. Imag. Sci. (ICIS)*, 2014.

[23] N. Mosenson, "Screening method," International Patent WO2 001 058 140 A1, Feb. 6, 2000.
[24] J.-M. Guo and S. Sankarasrinivasan, "H-GAN: Deep learning model for halftoning and its reconstruction," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2020, pp. 1–2.
[25] C.-H. Son, "Inverse halftoning through structure-aware deep convolutional neural networks," *Signal Process.*, vol. 73, Aug. 2017, vol. 107591.
[26] Y. Yi, R. Li, C. Yu, and Y. Yuan, "Quality evaluation metric for greyscale error diffusion halftone images based on texture and visual characteristics," *Imag. Sci. J.*, vol. 65, no. 6, pp. 315–326, Aug. 2017.
[27] O. Haik, O. Perry, E. Chen, and P. Klammer, "A novel inspection system for variable data printing using deep learning," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 3541–3550.
[28] M. Q. Nguyen and J. P. Allebach, "Feature ranking and selection used in a machine learning framework for predicting uniformity of printed pages," *Electron. Imag.*, vol. 2017, no. 12, pp. 166–173, Jan. 2017.
[29] W. Wang, G. Overall, T. Riggs, R. Silveston-Keith, J. Whitney, G. Chiu, and J. P. Allebach, "Figure of merit for macrouniformity based on image quality ruler evaluation and machine learning framework," *Proc. SPIE*, vol. 8653, Feb. 2013, Art. no. 86530P.
[30] Y. Yang, U. Sarkar, I. Borrell, and J. P. Allebach, "Ink quality ruler experiments and print uniformity predictor," *Image Quality and System Performance XVII. International Society for Optics and Photonics*, 2020.
[31] Z. Hu, L. Hu, P. Bauer, T. Harris, and J. Allebach, "Relation between image quality and scan resolution: Part I," *Electron. Imag.*, vol. 2020, no. 9, pp. 1–322, 2020.
[32] J. Mannos and D. J. Sakrison, "The effects of a visual fidelity criterion of the encoding of images," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 4, pp. 525–536, Jul. 1974.
[33] R. A. Ulichney, "Void-and-cluster method for dither array generation," *Proc. SPIE*, vol. 1913, pp. 332–343, Sep. 1993.
[34] W.-E. Huang, T. Liu, K. Bengtson, and J. Allebach, "Monochrome hybrid, multilevel, halftone screen with unequal spatial resolution for a low-cost electrophotographic printer," *Electron. Imag.*, vol. 2018, no. 16, pp. 1–428, 2018.
[35] *HP Indigo 10000 Digital Press Specification*. Accessed: Sep. 30, 2021. [Online]. Available: http://h20195.www2.hp.com/V2/GetPDF.aspx/4AA3-6493ENW.pdf
[36] *Scikit-Learn*. Accessed: Sep. 30, 2021. [Online]. Available: https://scikit-learn.org/stable/
[37] J. Brownlee, *XGBoost With Python: Gradient Boosted Trees With XGBoost and Scikit-Learn* (Machine Learning Mastery). ebook, 2016.

TAL FRANK received the M.Sc. degree in physics from the Weizmann Institute, Israel. He is currently pursuing the Ph.D. degree with the Ben Gurion University of the Negev. For the last ten years, he has been working at HP Indigo, where he has developed halftoning algorithms, experimental physics for Indigo presses, and an inspection system for Indigo presses.

OREN HAIK received the Ph.D. degree in computer vision from the Ben Gurion University of the Negev, Israel. For the last 11.5 years, he has been working at HP Indigo, where he has developed a real-time inspection systems, print quality measures, time-series prediction metrics, and halftoning algorithms. He is currently developing various machine and deep learning-based solutions for HP Indigo presses.

SHANI GAT received the M.Sc. degree in electrical engineering from Tel Aviv University, Israel. For the last three and a half years, she has been working at HP Indigo and has developed halftoning algorithms, print quality measures, and a real-time inspection system for Indigo presses.

OREL BAT MOR received the B.Sc. degree in physics from Bar-Ilan University, Israel. For the last three and a half years, he has been working at HP Indigo and has developed halftoning algorithms, PQ digital enhancements filters, and experimental physics for fine tuning Indigo presses.

JAN P. ALLEBACH (Life Fellow, IEEE) is currently a Hewlett-Packard Distinguished Professor of electrical and computer engineering at Purdue University. He is a fellow of IS&T and SPIE. He was elected to a membership in the National Academy of Engineering and a fellowship in the National Academy of Inventors. He was named SPIE/IS&T Electronic Imaging Scientist of the Year. He has received the IS&T Honorary Membership, the IEEE Daniel E. Noble Award, the OSA/IS&T Edwin Land Medal, and the IS&T Johann Gutenberg Prize.

YITZHAK YITZHAKY received the Ph.D. degree in electrical and computer engineering from the Ben Gurion University of the Negev, Israel, in 2000. He was a Postdoctoral Research Fellow at the Harvard Medical School, Schepens Eye Research Institute, Boston, MA, USA. He is currently an Associate Professor at the Electro-Optics Engineering Department, Ben-Gurion University of the Negev. His research interests include imaging, image enhancement, and computer vision.