IMAGE PROCESSING, IMAGE ANALYSIS, AND DATA SCIENCE APPLIED TO PROBLEMS IN PRINTING AND SEMANTIC UNDERSTANDING OF IMAGES CONTAINING FASHION ITEMS

by

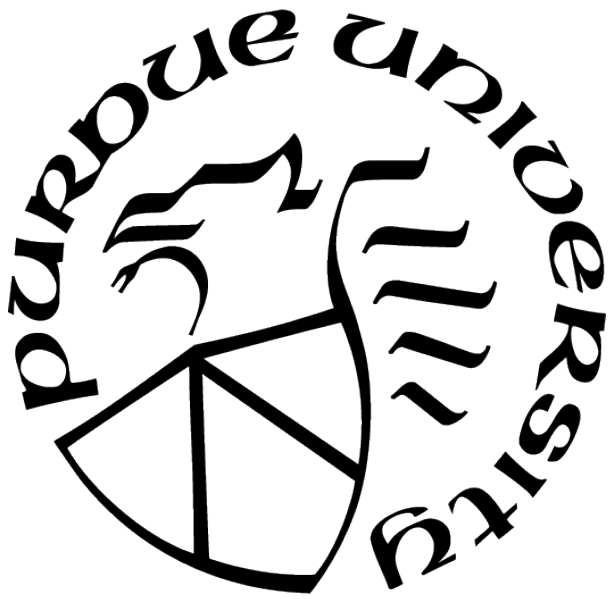
Wan-Eih Huang

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy



School of Electrical and Computer Engineering

West Lafayette, Indiana

December 2020

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Jan P. Allebach, Chair

School of Electrical and Computer Engineering

Dr. George T. Chiu

School of Mechanical Engineering

Dr. Amy Reibman

School of Electrical and Computer Engineering

Dr. Fengqing Maggie Zhu

School of Electrical and Computer Engineering

Approved by:

Dr. Dimitrios Peroulis

To my dearest family

ACKNOWLEDGMENTS

First of all, I would like to thank my major advisor Professor Allebach. His wonderful advice and guidance in those projects I participated helped me a lot in image processing and acoustic signal processing. When I encountered the difficulties in development, he always provided me insightful ideas. In addition, I would like to thank Hewlett-Packard, Boise for providing me opportunities to participate those exciting industry projects. Those were very awesome experiences.

Secondly, I would like to thank all my friends and lab members. They broaden my knowledge and vision. They are always willing to help me not only in research field also in the life. I really appreciate that. I would not have been able to do it without the amazing people I have met along the way.

Lastly, I would like to thank my family for their unconditional love. Even though I cannot spend much time with them since I started my Ph.D. study, they still support every decision I made and encourage me to explore the world.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This thesis aims to address problems in printing and semantic understanding of images.

The first one is developing a halftoning algorithm for multilevel output with unequal resolution printing pixels. We proposed a design method and implemented several versions of halftone screens. They all show good visual results in a real, low-cost electrophotographic printer.

The second problem is related to printing quality and self-diagnosis. Firstly, we incorporated logistic regression for classification of visible and invisible bands defects in the detection pipeline. In addition, we also proposed a new cost-function based algorithm with synthetic missing bands to estimate the repetitive interval of periodic bands for self-diagnosing the failing component. It is much more accurate than the previous method. Second, we addressed this problem with acoustic signals. Due to the scarcity of printer sounds, an acoustic signal augmentation method is needed to help a classifier perform better. The key idea is to mimic the situation that occurs when a component begins to fail.

The third problem deals with recommendation systems. We explored the similarity metrics in the loss function for a neural matrix factorization network.

The last problem is about image understanding of fashion items. We proposed a weakly supervised framework that includes mask-guided teacher network training and attention-based transfer learning to mitigate the domain gap in datasets and acquire a new dataset with rich annotations.

1. INTRODUCTION

Images are an important media to convey information in our daily life. The complex human visual system allows us to perceive them in many applications. Thus, it is valuable to investigate the problems in image applications and propose the solutions to solve them. The work in this dissertation is separated into two parts: printing applications and an online shopping application.

To improve printing quality with the laser, electrophotographic printing technology, we explored the problems in two aspects: a halftoning algorithm and printer self-diagnosis.

In Chapter 2, our halftoning algorithm is introduced. The hybrid screen leverages two halftoning techniques, screening and direct binary search (DBS), to achieve the better quality of the halftoned images, and to enable the algorithm to be integrated into low-cost printers with limited computational resources. This work proposes a complete hybrid screen design method for multilevel output with unequal resolution printing pixels in a laser electrophotographic system. Because of the unstable rendering output of the electrophotographic process, we adopt a clustered-dot screen in our work. We also use the supercell approach to solve the trade-off between screen frequency and the effective number of quantization levels that is inherent to a clustered-dot screen. Moreover, we use subpixel modeling to simulate the unequal resolution printing pixels and multilevel output. This method is well-suited to development of halftoning algorithms for systems with unequal resolution. We also propose several design rules, and evaluate their impact on printing quality.

Not only the halftoning algorithm affects printing quality, but the failing mechanical components can produce unwanted artifacts on printed pages. Therefore, a self-diagnosing printer is needed for fast troubleshooting. We analyzed the data from images and acoustic signals. For the image part, we are particularly interested in band defects. In Chapter 3, we present the work that includes bands detection and repetitive bands analysis. The repetitive interval is a very crucial feature of bands in print quality assessment, because any irregularity on the surface of a rotating component localized in the circumference will cause repetitive defects in the output of the printer [1] [2] [3]. Hence, the repetitive interval can help us diagnose the issues. In previous work, a cost function method provides a robust algorithm

to predict the repetitive interval on less noisy samples. However, if the samples contain more aperiodic bands and noise, the estimation will become a challenge. Moreover, the missing periodic bands will decrease the probability of correct prediction. In this work, we proposes a novel cost-function-based repetitive interval estimation method for periodic bands. By adding synthetic missing bands, we re-evaluate the cost function values to check whether it has a better result. We also show the improvement of accuracy on the print samples with our proposed algorithm.

In addition, acoustic signals can provide information about the root cause of failing components as well. However, it is hard to collect the printer sounds. Especially the abnormal printer sounds. In order to augment acoustic signals to address scarcity of the printing sounds, we will examine the previous work about the detector first in Chapter 4. Then, we will show some analysis result in real printing sounds and synthesized sounds by conventional augmentation methods. Lastly, we explore and discuss some possible augmentation methods for our application.

The second part of this dissertation addresses online shopping applications. Firstly, we talk about experiments for recommender systems in Chapter 5. Matrix factorization is a famous algorithm in the recommender systems field. It is used for discovering underlying user-item relationships. Neural matrix factorization further fuses linear and non-linear interaction between users and items. In this work, we explored how similarity metrics affect the neural matrix factorization network.

Secondly, our work for semantic image understanding of fashion items is presented in Chapter 6. Garment semantic understanding is an important topic in fashion online shopping. It is useful for many applications such as recommender systems and image retrieval. However, it is challenging, because the clothing types would share some similar features. For example, the bottom part of a dress and skirt. Moreover, there are many attributes of clothing items such as sleeve length and neck type. Thus, a dataset with rich annotations is needed. In this work, we proposed a weakly supervised framework that is able to mitigate the domain gap and combine the annotations from two datasets.

In the last chapter we summarize our contributions in this dissertation.

2. MONOCHROME HYBRID, MULTILEVEL, HALFTONE SCREEN WITH UNEQUAL SPATIAL RESOLUTION FOR A LOW-COST ELECTROPHOTOGRAPHIC PRINTER

2.1 Introduction

Monochrome halftoning is a technique for rendering a continuous tone image into an image with binary or a few levels of gray at each pixel.

There are plenty of halftoning algorithms that have been well developed. In general, we can categorize the halftoning algorithms into three basic architectures. First, it is point-to-point processing, such as screening or dithering [4] [5]. Second, neighborhood processing, where the representative method is error diffusion [6]. The last is search-based algorithms, and usually they are iterative. An example is direct binary search (DBS) [7]. Even though the iterative search based algorithms provide the best quality among these three types of architecture, they need massive computations. Thus, it is impossible to generate halftoned images in real time by iterative search based algorithms. In contrast, screening and error diffusion are widely used in practical implementations due to their reduced computational complexity. However, the iterative search based methods are still a crucial tool for designing efficient halftoning algorithms offline. In this paper, we adopt the hybrid screen method [8]. We design the screening algorithm offline by DBS to achieve better quality. Then the halftoned image can be generated in real time by the screening method. As a result, the algorithm is capable to be integrated into a low-cost printer with limited computational resources.

We can classify screens based on the textures they generate. One is the clustered-dot screen, where clustered individual printer addressable dots form a grid-like pattern. It produces gray levels by varying the cluster size. In contrast, a dispersed-dot screen does not form clusters. A number of gray levels are generated by varying the density (frequency) of dots. Clustered dots are more stable than single isolated dots and produce less dot gain as a fraction of area coverage [8]. For our target laser printer based on the electrophotographic process, which has the nature of unstable rendering output, it is more robust to use a clustered-dot screen. Nevertheless, there is a trade-off on the design of a clustered-dot

screen. The frequency of the clustered dots is referred to as the screen frequency. The screen renders the image detail better with a higher screen frequency. On the other hand, a smaller size screen (higher screen frequency) results in fewer gray levels. Moreover, considering the implementation on a digital printer, the screen frequency must be lower than the printer resolution [8] in order to form clusters. To solve the trade-off between screen frequency and effective number of gray levels, a supercell approach has been proposed [9].

For equal resolution, a printer addressable pixel has the same size in both the horizontal and vertical directions. Thus, we can apply the DBS or screening algorithms directly, because the digital image pixels match the printer addressable pixels. However, for unequal resolution, the printer addressable pixel has different dimensions in two directions as illustrated in Fig. 2.1. Therefore, we use the digital image pixel with identical dimension as a subpixel, and then several subpixels grouped with different numbers horizontally and vertically form a simulated unequal resolution printer addressable pixel. In the halftoning algorithm, the subpixels within the same simulated printer pixel have the same binary condition to get close to the real printing output. We will discuss more details in a later section.

2.2 Methodology

2.2.1 Screen Tile Vectors

A periodic screen can be related to two vectors $z = [z_{\mathrm{i}}, z_{\mathrm{j}}]$ and $w = [w_{\mathrm{i}}, w_{\mathrm{j}}]$ defined as screen tile vectors. In order to cover the whole spatial domain with these two vectors, they must be linearly independent. The screen angle is defined as the angle between the tile vector and the j-axis in [8]. We can obtain screen tile vectors from the desired lines per inch (lpi); and the process is described in [8]. Our case is unequal spatial resolution. Assume the height and width of each printing dot are X and Y, respectively. Then we can obtain the screen frequency from the relationship described in Eq. (2.1), where $R_{gcd} = gcd(\frac{1}{X}, \frac{1}{Y})$ and gcd is the greatest common divisor.

$$\text{screen frequency } (\frac{\text{lines}}{\text{inch}})\ \nu = \frac{R_{gcd}}{\|XR_{gcd}|w_{\mathrm{i}}|, YR_{gcd}|w_{\mathrm{j}}|\|} \tag{2.1}$$

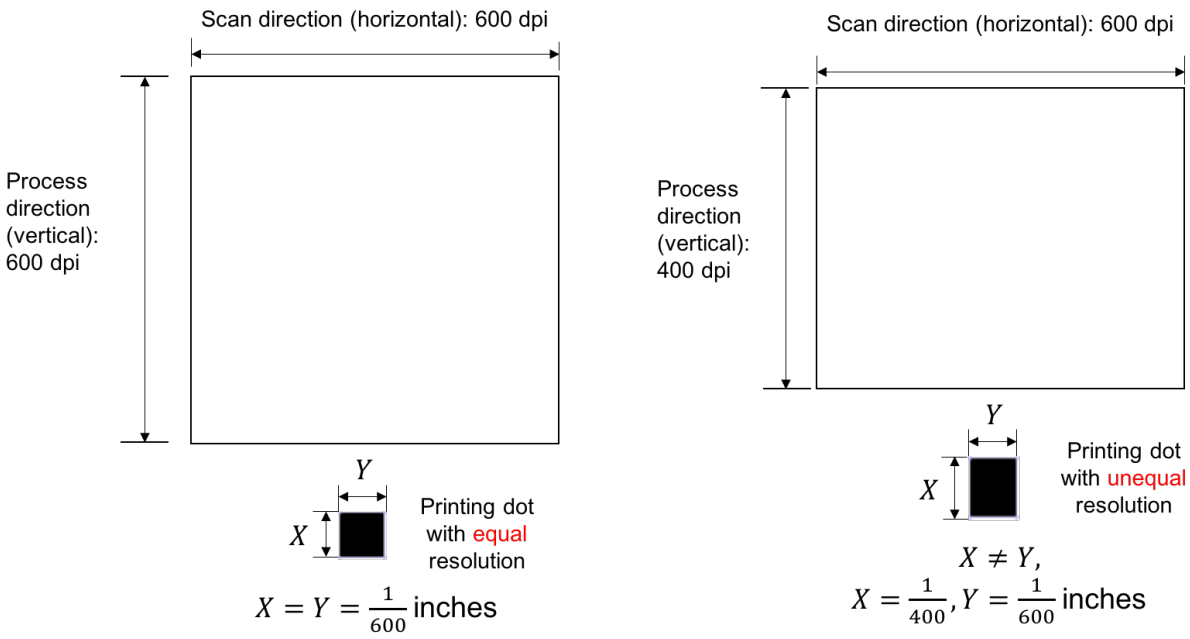


Figure 2.1. Comparison of equal resolution and unequal resolution. For 600×600 dpi, both the height and width of a printing dot is $\frac{1}{600}$; For 400×600 dpi, the height of a printing dot is $\frac{1}{400}$ and the width of a printing dot is $\frac{1}{600}$.

2.2.2 Microcell and Supercell

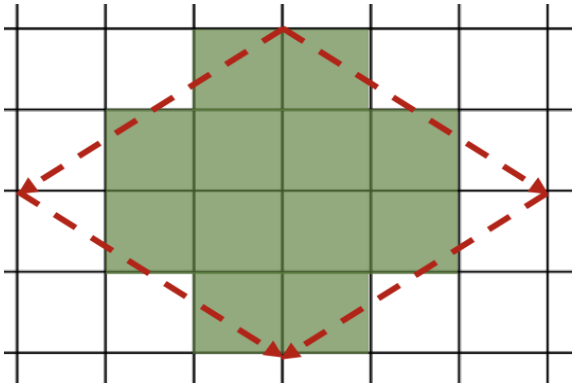
Two screen tile vectors construct a parallelogram called the continuous parameter halftone cell (CPHC), and then we can compute the overlapping area of the CPHC and the 2-D discrete-parameter space to design the microcell [10]. The number of pixels N in a microcell is equal to $|\det [z^T w^T]|$. Fig. 2.2(a) shows the microcell obtained by the tile vectors $z = [2, 3]$ and $w = [2, -3]$; and it contains 12 pixels in a microcell. For the purpose of simplifying the implementation, the smallest rectangle that can be tiled in the horizontal and vertical directions, called the basic screen block (BSB), is a unit used for storing the screen in a 2-D array [8]. Therefore, the height of the BSB is $\frac{N}{gcd(z_j, w_j)}$ and the width of the BSB is $\frac{N}{gcd(z_i, w_i)}$ [11]. An example of a BSB shown in Fig. 2.2(b) which is the corresponding BSB of Fig. 2.2(a).

The supercell obtained by tiling the BSBs in 2-D space is the screen we use for thresholding the continuous tone image. Fig. 2.3(a) is an example of a supercell which has 4 BSBs both in the horizontal and vertical directions. The height and width of a supercell can be arbitrary integers. However, in practical implementation, we need to consider the cost and the quality. If the screen size is bigger, it needs more memory. If the screen size is smaller, the periodic patterns might be more obvious.

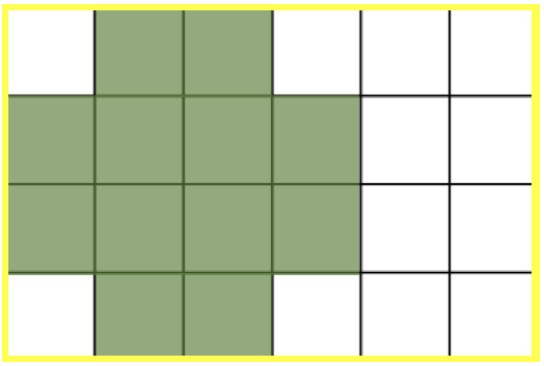
Once the supercell is set, the next step is to design the highlight core and shadow core regions for the stochastic dot texture [5]. That is, the microcell growth sequence of each core might not be the same for each microcell so that it provides spatial freedom in the halftoned patterns. The core shape is determined by different situations. For example, the square core is suitable for a conventional round dot-cluster pattern [5]. In our project, we want to achieve the S shape structure in the midtone, thus we design S shape cores illustrated in Fig. 2.3(b).

2.2.3 Screen Generation

The screen generation process can be divided into 3 phases: highlight, midtone, and shadow. In the beginning, we start with an initial random level for which half of the highlight cores have one dot.



(a) Microcell.



(b) BSB.

Figure 2.2. Microcell and BSB.

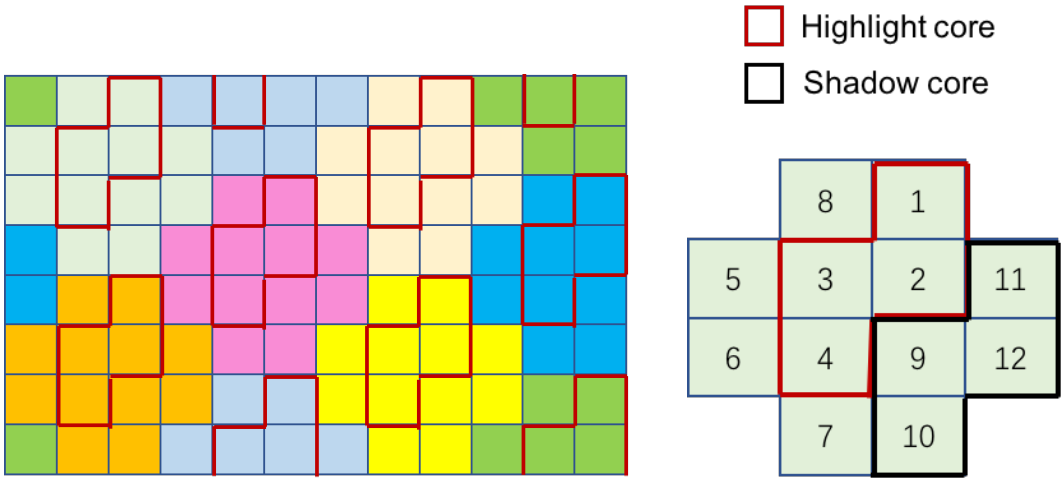


Figure 2.3. (a) 8×12 Supercell. (b) A microcell with S shape cores.

Phase 1: To determine the halftoned pattern for the initial gray level, we use *constrained DBS swap*. The candidate swap pixels can be neighboring pixels within the same highlight core or the pixels in the highlight cores without dots for each trial swap. The trial swap change is allowed only if it minimizes ΔE, the change in visually weighted total squared error, as described later in this chapter. After obtaining the halftoned pattern for the initial level, we remove dots sequentially based on the halftoned pattern generated in the previous step until no dot remains in the halftone pattern. Similarly, we add dots sequentially based on the halftoned pattern generated in the previous step until all highlight cores contain one dot. How to select a dot to remove or add is using *constrained DBS toggle*. The trial toggle (on or off) change is allowed only if it minimizes ΔE. In this stage, the design of the macrocell index array is completed. Then we can apply this macrocell index array to the remaining microcells' levels along with the design constraints within the microcell to generate the halftoned patterns sequentially until all highlight cores are filled. The reason for using the same macrocell index array is to reduce the effort of optimization. It is reasonable that if the macrocell sequence achieved good quality in the lower gray levels (lighter) then the higher gray levels (darker) should be also good with the same macrocell sequence, because the effect of the macrocell sequence is more significant in the lighter region. Figure 2.4 illustrates the design process of Phase 1.

Phase 2: Midtone levels are generated by the same macrocell index array obtained in Phase 1 and the microcell dot growing sequence.

Phase 3: A mirroring method is used in the shadow levels. To put it another way, we invert the dot profile function for each highlight core and offset it for the shadow levels.

2.2.4 Unequal Resolution Printing Model

In order to simulate unequal spatial resolution and multilevel output fairly, we adopt subpixel modeling. A square subpixel in the digital image represents part of an ideal printer addressable dot. Hence, we can repeat this kind of subpixel in both the horizontal and vertical directions to form a simulated unequal resolution printer addressable dot.

Take the specification of our implementation as an example, the printer resolution is 600 dpi in the horizontal direction and 400 dpi in the vertical direction, thus the simulated

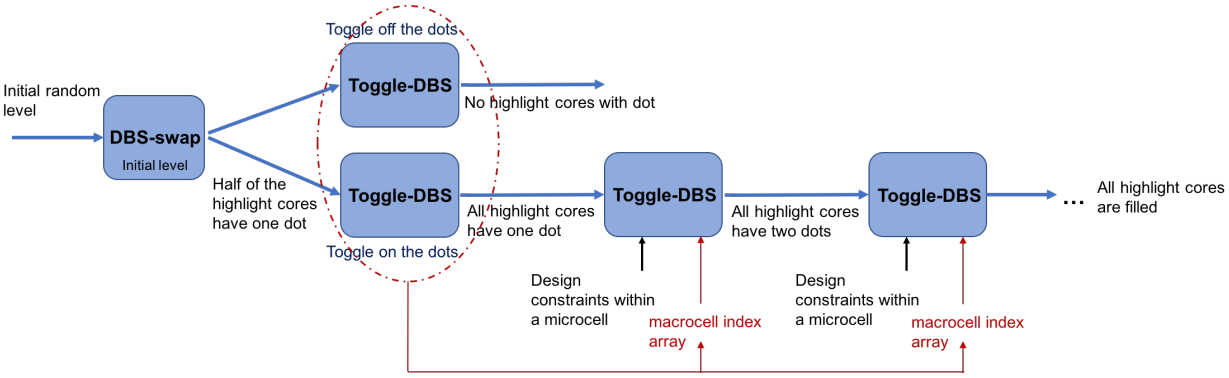


Figure 2.4. Design process of Phase 1.

unequal resolution printing pixel is composed of 3×2 subpixels. Furthermore, considering multilevel output, it is common for laser electrophotographic printers to support multilevel output by PWM (pulse width modulation). In our case, 2 bpp (bits per pixel), a pixel is divided into 3 slivers. Thus, the simulated unequal resolution printing pixel for 2 bpp output is composed of 9×6 subpixels. In addition, each group of 9×2 subpixels in one sliver turns on or off simultaneously. Figure 2.5 shows the subpixels model we use in this project.

2.2.5 DBS for unequal resolution

DBS for multilevel applications [12] and DBS for multiple pixels changes [13] have been proposed. Therefore, we can further extend the concept to the unequal resolution DBS. That is to say, DBS on unequal resolution pixels with $\frac{1}{X} \times \frac{1}{Y}$ dpi can be viewed as DBS on several equal resolution subpixels with $\frac{1}{Z} \times \frac{1}{Z}$ dpi which form a simulated unequal resolution pixel. $\frac{1}{Z}$ is the smallest common multiple of $\frac{1}{X}$ and $\frac{1}{Y}$. On DBS swap or toggle operations, all the subpixels within the same unequal resolution simulated pixel switch to the same binary condition due to the fact that they represent a single unit. Hence, the DBS equations can be rewritten. Assume that there are $m \times n$ subpixels within a simulated pixel. And DBS toggles pixel $\mathbf{m}_0$ or swaps $\mathbf{m}_0$ and $\mathbf{m}_1$. We denote $\mathbf{m}_{0\mathrm{ij}}$ as subpixels of $\mathbf{m}_0$ and $\mathbf{m}_{1kl}$ as subpixels of $\mathbf{m}_1$, where $\mathrm{i} \in \{0, 1, ..., m-1\}$ and $\mathrm{j} \in \{0, 1, ..., n-1\}$. The trial halftone image can be represented by Eq. (2.2). If the operation is to toggle on a dot at $\mathbf{m}_0$, $a_0 = 1$ and $a_1 = 0$. Otherwise, $a_1 = -a_0$ in a swap operation, where it is assumed that there is a dot at $\mathbf{m}_0$ before the swap operation. The change in error due to the trial toggle or swap is given by Eq. (2.3). Equations (2.2) and (2.4) indicate the changes that need to be made should a trial toggle or swap be accepted.

$$g[\mathbf{m}] = g[\mathbf{m}] + a_0 \sum_{\mathrm{i}=0}^{m-1} \sum_{\mathrm{j}=0}^{n-1} \delta[\mathbf{m} - \mathbf{m}_{0\mathrm{ij}}] + a_1 \sum_{k=0}^{m-1} \sum_{l=0}^{n-1} \delta[\mathbf{m} - \mathbf{m}_{1kl}] \quad (2.2)$$

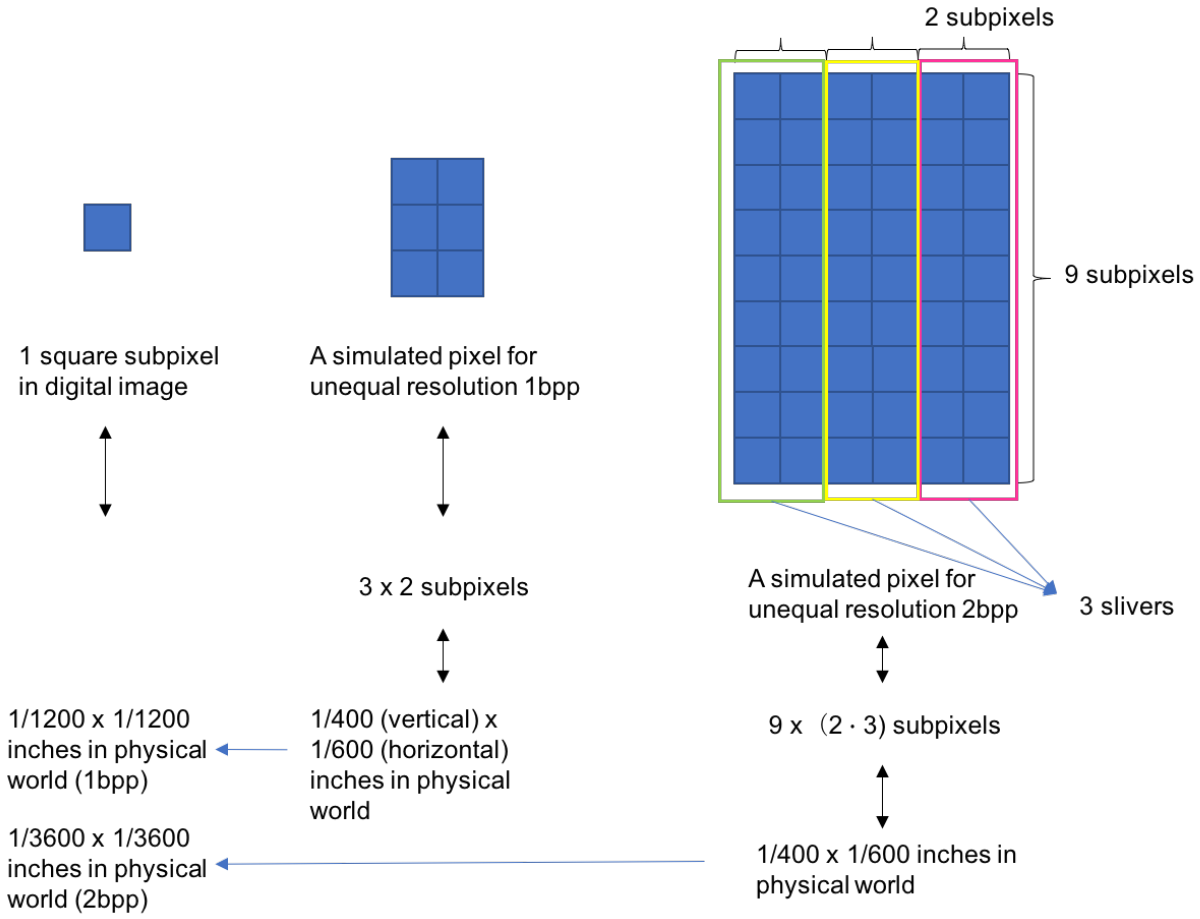


Figure 2.5. Subpixels modeling.

$$\Delta E = 2a_0 \sum_{\mathrm{i}=0}^{m-1} \sum_{\mathrm{j}=0}^{n-1} c_{\tilde{p}\tilde{e}}[\mathbf{m}_{0\mathrm{ij}}] + 2a_1 \sum_{k=0}^{m-1} \sum_{l=0}^{n-1} c_{\tilde{p}\tilde{e}}[\mathbf{m}_{1kl}] + a_0 a_1 \sum_{\mathrm{i}=0}^{m-1} \sum_{\mathrm{j}=0}^{n-1} \sum_{k=0}^{m-1} \sum_{l=0}^{n-1} c_{\tilde{p}\tilde{p}}[\mathbf{m}_{0\mathrm{ij}} - \mathbf{m}_{1kl}] \quad (2.3)$$

$$c'_{\tilde{p}\tilde{e}}[\mathbf{m}] = c_{\tilde{p}\tilde{e}}[\mathbf{m}] + a_0 \sum_{\mathrm{i}=0}^{m-1} \sum_{\mathrm{j}=0}^{n-1} c_{\tilde{p}\tilde{p}}[\mathbf{m} - \mathbf{m}_{0\mathrm{ij}}] + a_1 \sum_{k=0}^{m-1} \sum_{l=0}^{n-1} c_{\tilde{p}\tilde{p}}[\mathbf{m} - \mathbf{m}_{1kl}] \quad (2.4)$$

2.3 Experimental Results

This section shows the experimental results for applying our hybrid screen design with multilevel and unequal spatial resolution output.

In our designs, we apply the compact rule. That is, the newly added sliver should be adjacent to a previous one. In Fig. 2.6, two halftoned patterns corresponding to the same gray level show an example of the impact of the compact rule. The left image was generated without the compact rule. It means that DBS is free to choose any sliver within the highlight cores. On the other hand, the newly added sliver has to be chosen from the slivers adjacent to the existing slivers in right image generated with the compact constraint. From our result, we can see the objectionable diagonal structure in the left image. And the compact rule reduces these artifacts significantly.

We also consider the centroid (symmetric) rule in our design. In other words, the shift of the centroid of the clustered-dots due to the addition of a sliver must be as small as possible. Figure 2.7 shows the impact of the centroid rule for two halftoned patterns with the same gray level. The left halftoned pattern is generated without the centroid rule so there is an objectionable diagonal structure. The right one is generated with the centroid rule and looks more smooth than the left one. Figure 2.8 is the version of Fig. 2.7 after applying a Gaussian filter. These images are closer to the perception through human eyes.

Figures 2.9 and 2.10 show halftoned ramp images thresholding by the screens generated by our approach. The screen Fig. 2.9 used is a square core design; and the screen Fig. 2.10 used is a S shape core design. Both show good quality.

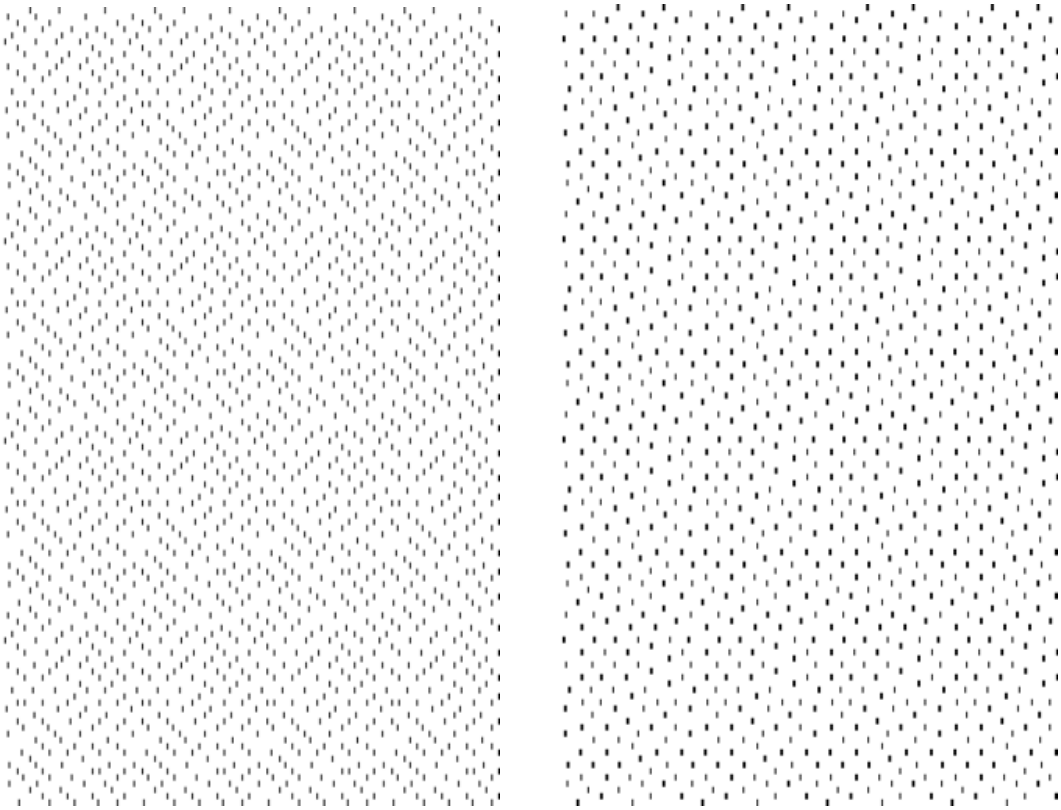


Figure 2.6. Comparison of the halftoned patterns. (a) The halftoned pattern without compact rule. (b) The halftoned pattern with compact rule.

2.4 Conclusion

In this chapter, we developed a novel and complete hybrid screen design method with subpixels modeling. Moreover, we also implemented several version of halftone screens by our approach and they all show good visual results in a real printer. Hence, our method is robust for developing halftoning algorithms for printing systems with unequal spatial resolution.

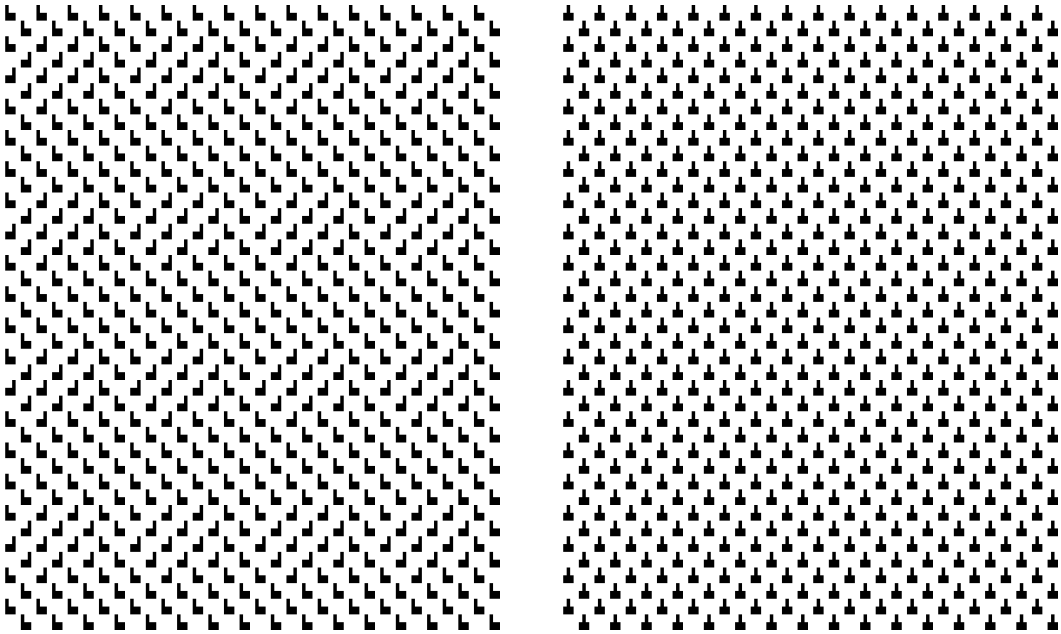


Figure 2.7. Comparison of the halftoned patterns. (a) The halftoned pattern without centroid rule. (b) The halftoned pattern with centroid rule.

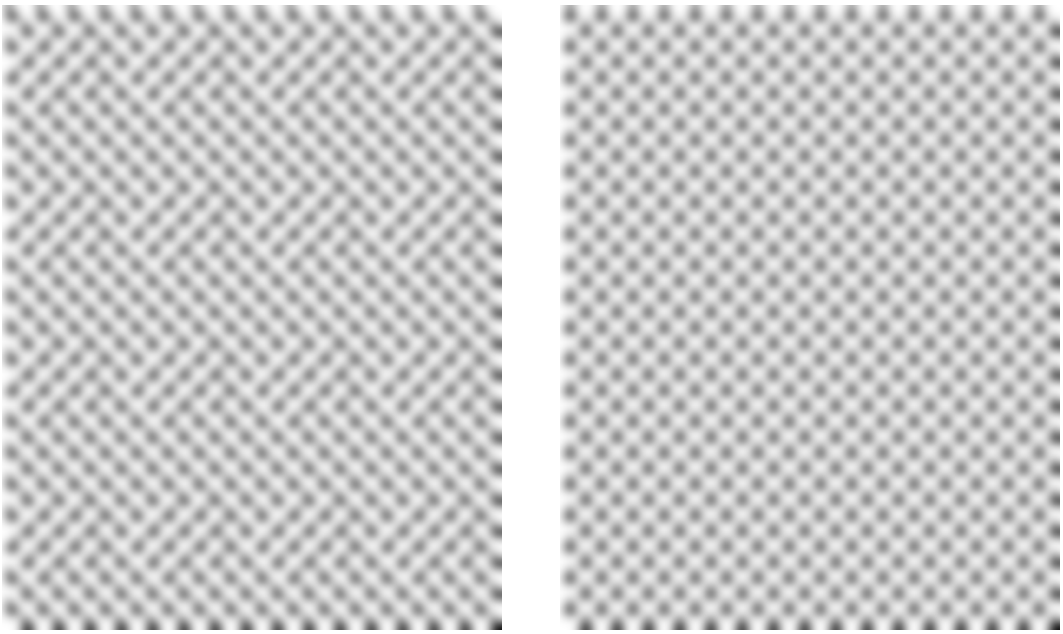


Figure 2.8. Comparison of the Gaussian filtered halftoned patterns shown in Fig. 2.7. The standard deviation of the Gaussian filter was $\sigma = 6$ pixels. (a) The halftoned pattern without centroid rule. (b) The halftoned pattern with centroid rule.

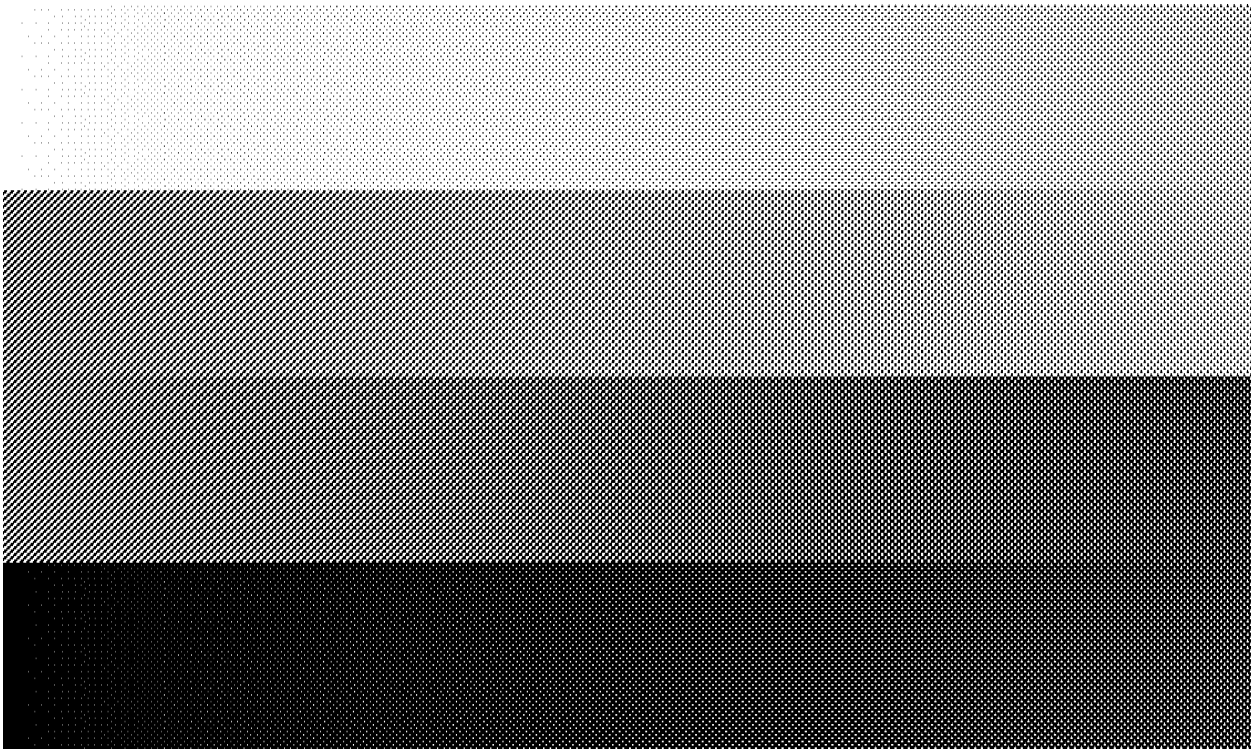


Figure 2.9. 1.88 in $\times$ 1.12 in halftoned image generated by a 20×24 supercell hybrid screen with square core shape, 4-levels, and unequal spatial resolution (600 dpi $\times$ 400 dpi).

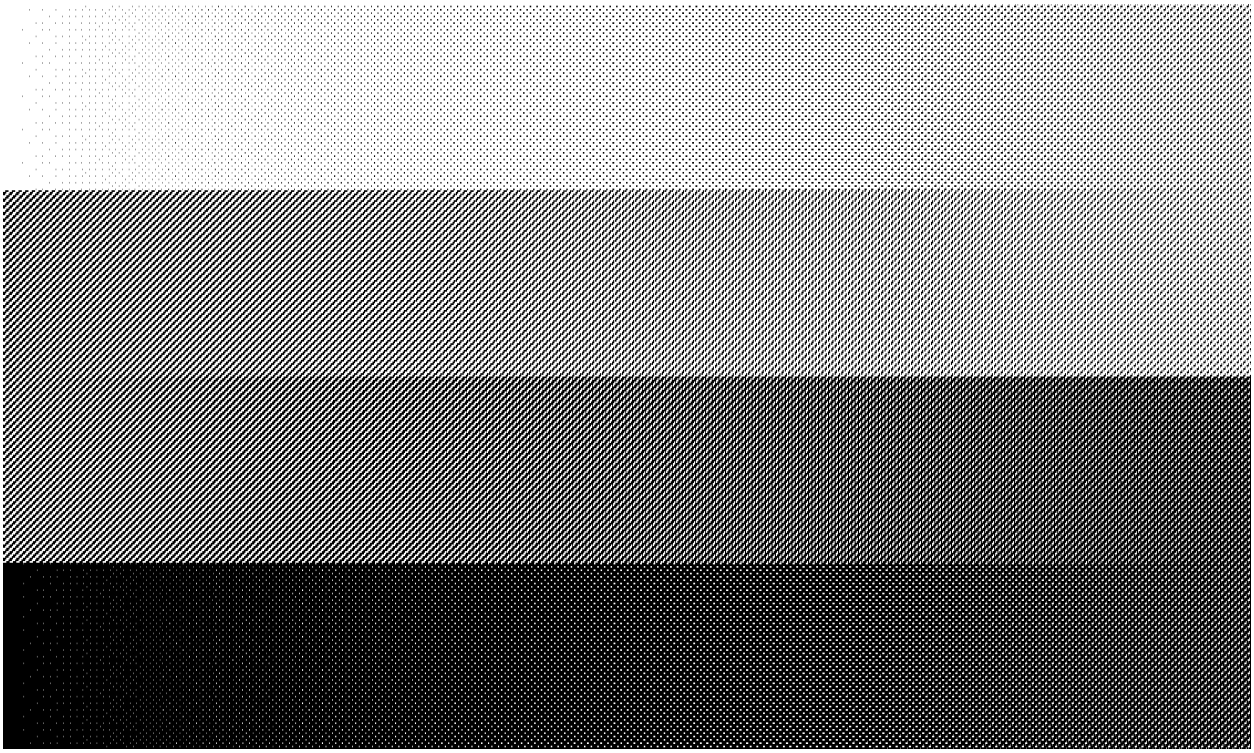


Figure 2.10. 1.88 in $\times$ 1.12 in halftoned image generated by a 20 $\times$ 24 supercell hybrid screen with S core shape, 4-levels, and unequal spatial resolution (600 dpi $\times$ 400 dpi).

3. COST-FUNCTION-BASED REPETITIVE INTERVAL ESTIMATION METHOD WITH SYNTHETIC MISSING BANDS FOR PERIODIC BANDS

3.1 Introduction

Electrophotographic printers have been widely used in the world. There are many print quality (PQ) issues shown in different types of defects, like bands, streaks, and gray spots. Since the electrophotographic process involves multiple delicate components, different appearance of a certain defect might be caused by different components. Hence, the intent of this work is to extract the features of defects so that it could help the diagnosis of the failing components.

One of the most common printing defects in the electrophotographic process is bands, which is the one we want to address in this chapter. It occurs along the scan direction and repeats along the process direction. There are some related works that analyzed the problem of halftone banding [14] [15] [16]. In addition, some works addressed isolated large pitch bands [17] [18] [19]. However, we focus on sharp roller bands in this work. Figure 3.1 shows an example of this kind of bands defect.

Our bands detection is based on the work of Zhang et al. [20]. Some fixed threshold values from observations are used to identify the bands. In this work, we want to explore a new method to identify the bands using a machine learning method, logistic regression, to classify the potential defects.

Moreover, the repetitive interval of periodic bands is a very important feature to diagnose the root cause components. There are a couple of methods to deal with it. One is the histogram method [20], using the histogram of the intervals between neighboring bands. However, if there are some aperiodic bands or noise between two periodic bands, the correct interval will not be chosen. Another method is the cost function method [3], with an exhaustive search for the best solution by evaluating the cost function value. Nevertheless, if the samples are more noisy or corrupted, this method is not able to estimate the repetitive interval correctly. For example, there may be multiple equally spaced bands sequences, and

some of the true periodic bands may be missing. Therefore, we introduce synthetic missing bands in our work to improve the accuracy. We will discuss more details in later sections.

3.2 Methodology

In this work, the pipeline we used can be divided into two parts: bands detection and repetitive bands analysis shown in Fig. 3.2. Our algorithms are specifically tailored to the test page shown in Fig. 3.1

3.2.1 Bands detection

In order to extract the features of bands, we need to identify the bands first. The steps in this part include pre-processing, bands profile extraction, and bands identification. We follow previous work in pre-processing and bands profile extraction. The details are described in [20].

Pre-processing

First, we de-screen the input sample to remove halftone patterns and then mask fiducial dots, the non-printable area, and the bar code which contains confidential information. Because the bands might fade along the scan direction, we partition the sample into three regions, left region, center region, and right region [20] so that we can analyze each region independently.

Bands profile extraction

In the bands profile extraction step, the input is one region of the image from pre-processing. The image is converted from the sRGB color space to the CIE 1931 XYZ color space. Then we use spatial projection of the 2-D image onto the process direction by computing the mean value of each line along the scan direction [20]. Afterwards, we perform color space conversion from CIE XYZ to CIE 1976 L*a*b* on the 1-D projection data. To better distinguish the bands from the background, we subtract the baseline from the 1-D projection data for each channel. The baseline is obtained from the filtered 1-D projection

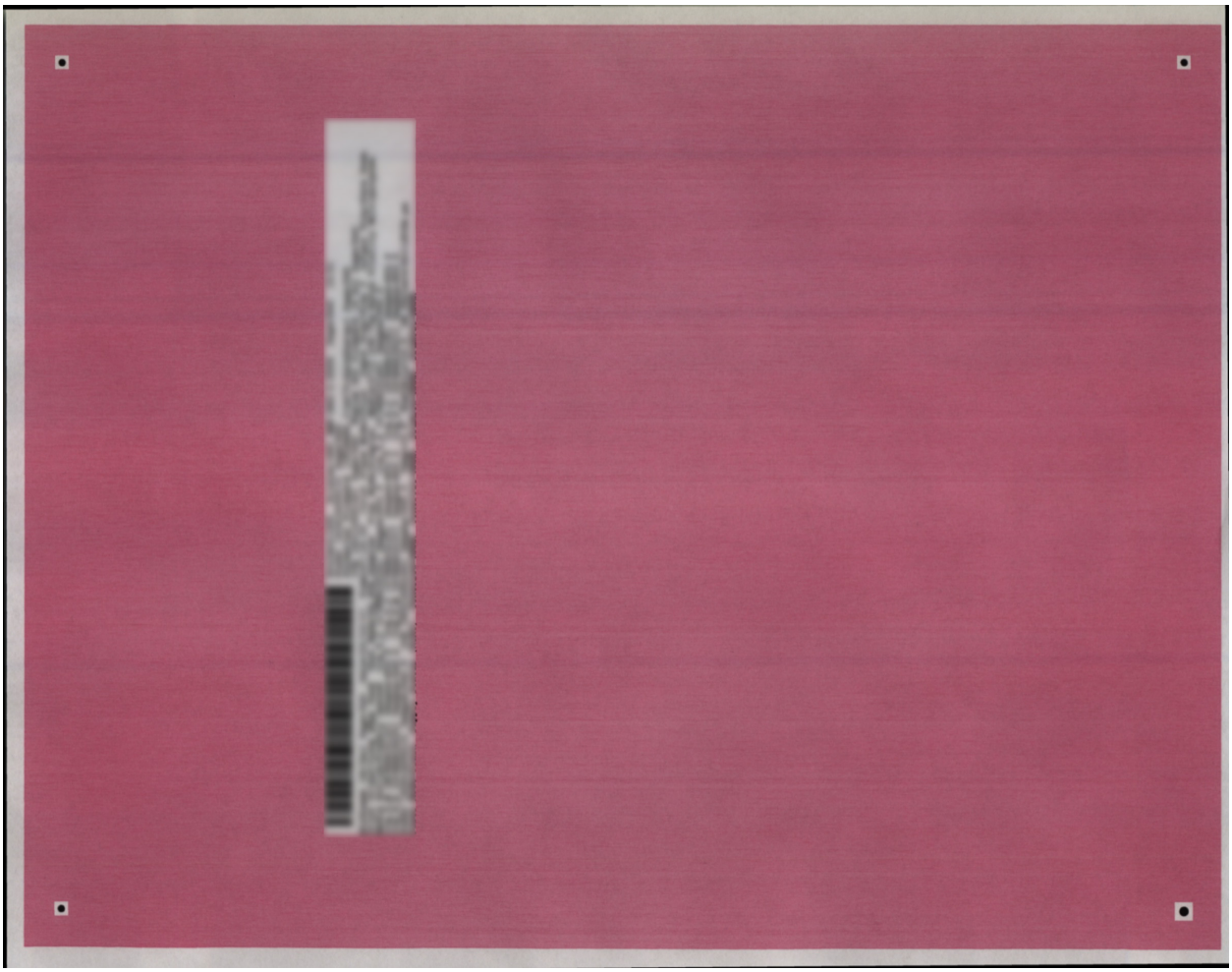


Figure 3.1. Example of bands defect.

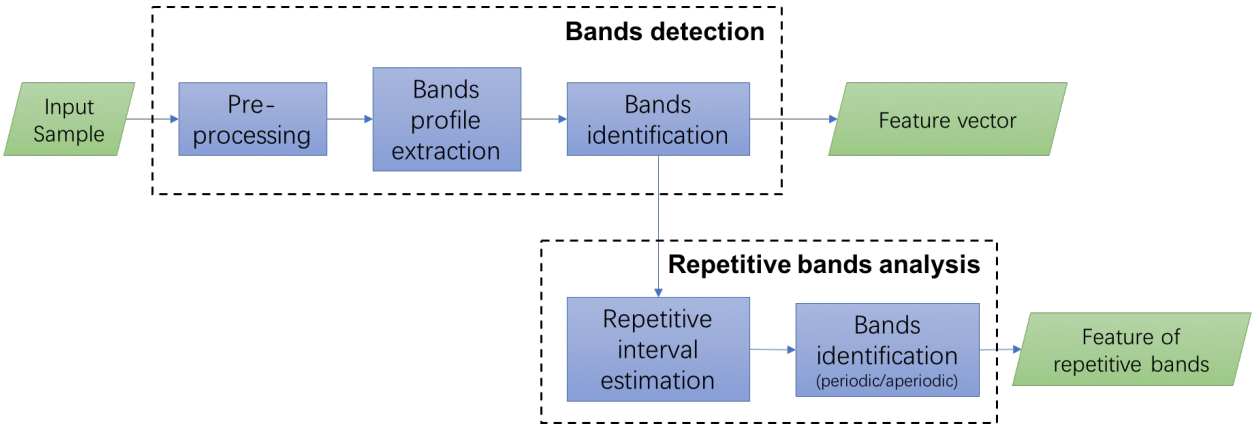


Figure 3.2. Overall pipeline of proposed algorithm.

data. Finally, we can compute ΔE defined in Eq. (3.1) and combine it with the sign of the baseline removed L* channel projection data as our bands profile. The sign value represents whether the ΔE is lighter than the background or darker than the background.

$$\Delta E = \sqrt{(L^*_{proj} - L^*_{base})^2 + (a^*_{proj} - a^*_{base})^2 + (b^*_{proj} - b^*_{base})^2} \tag{3.1}$$

Bands identification

After the 1-D bands profile extraction process, the small fluctuations of ΔE are eliminated by the threshold value. We use the mean value of ΔE plus the standard deviation of ΔE as the threshold value. Next, we locate the edges of peaks. Moreover, if there are positive and negative values of ΔE within a peak, we separate it into a light peak and a dark peak. In other words, each peak is either light or dark. After that, we extract features by computing the center, height, width, area, and sharpness (transition width) for each peak.

However, the human vision system is complicated. Some peaks are invisible among the detected peaks. There are two reasons we want to classify visible and invisible potential defects. First, we obtain better data quality. For example, we lower the false alarm rate. Second, if too many invisible potential defects are included in the repetitive interval estimation in the later process, it would harm the performance and accuracy of the estimation of periodic bands.

In this work, we apply logistic regression to build our classification model. The model is a weighting function to predict the probability that the binary output is visible or invisible. Logistic regression uses a sigmoid function shown in Eq. (3.2) to approximate the hypothesis function. $\mathbf{x}$ is the feature vector and θ is the vector of trainable weighting coefficients. The curve of the sigmoid function is monotonic and it maps all real numbers to the values from 0 to 1. Therefore, it has a good probabilistic interpretation. Assume the estimated probability for label 1 given $\mathbf{x}$ is $h_\theta(\mathbf{x})$, and the estimated probability for label 0 given $\mathbf{x}$ is $1 - h_\theta(\mathbf{x})$. Then, we can try to maximize the likelihood $\mathcal{L}$ in Eq. (3.3), where y_i is the ground truth labels and N is number of samples. To optimize it easily, we can rewrite the maximum likelihood $\mathcal{L}$ by taking logarithm value and changing the sign shown in Eq. (3.4) and Eq. (3.5) [21]. Lastly, we can use gradient descent to train the model.

$$h_\theta(\mathbf{x}) = \frac{1}{1 + \mathrm{e}^{-\theta \mathbf{x}}} \tag{3.2}$$

$$\mathcal{L} =_\theta \prod_{\mathrm{i}=1}^{N} h_\theta(x_\mathrm{i})^{y_\mathrm{i}} (1 - h_\theta(x_\mathrm{i}))^{1-y_\mathrm{i}} \tag{3.3}$$

$$\mathcal{L} =_\theta \log \left[\prod_{\mathrm{i}=1}^{N} h_\theta(x_\mathrm{i})^{y_\mathrm{i}} (1 - h_\theta(x_\mathrm{i}))^{1-y_\mathrm{i}} \right] \tag{3.4}$$

$$=_\theta -\sum_{\mathrm{i}=1}^{N} \{ y_\mathrm{i} \log h_\theta(x_\mathrm{i}) + (1 - y_\mathrm{i}) \log(1 - h_\theta(x_\mathrm{i})) \} \tag{3.5}$$

We choose three features: height (maximum ΔE of a peak), minimum sharpness of two sides, and width for the logistic regression algorithm. Then, we apply this model to our bands identification algorithm. The detailed results will be shown in Section 3.3.

3.2.2 Repetitive bands analysis

The features are extracted after the bands identification process. Hence, we can use the information of each band to predict which band belongs to a set of periodic bands. In our method, the maximum value of ΔE, light/dark, and center position of each band are used in our repetitive interval estimation algorithm.

Repetitive interval estimation

There are two phases in the proposed estimation algorithm: an initial guess and adding synthetic missing bands. We apply the cost function method [3] to make an initial guess; and then we improve the accuracy by adding synthetic missing bands.

Phase 1: Initial guess. Since the repetitive bands are generated from one single component [3], they should look similar. Thus, we compare the strength of light bands and the strength of dark bands. Here, the strength is the maximum value of ΔE. The larger one determines whether our candidate set of repetitive bands will be dark or light. All following processes operate on the center positions of this candidate set of repetitive bands.

First, we want to present the cost function briefly, and summarize the algorithm steps. In order to introduce the cost function method clearly, we start with the notation. N is the total number of bands in the candidate set; and the input data is the positions of candidate bands denoted by $\vec{b} = [b_1, b_2, \ldots, b_N]$. Assume there are p periodic bands and define the membership vector $\vec{m}$ to indicate which band belongs to the set of repetitive bands. That is,

$$\vec{m} = [m_1, m_2, \ldots, m_N], \quad m_{\mathrm{i}} = \begin{cases} 1, & \textit{periodic} \\ 0, & \textit{aperiodic} \end{cases} \qquad \mathrm{i} = 1, 2, \ldots, N.$$

There are two variables in the cost function. One is o which is the position of the first periodic band. The other is the repetitive interval Δb. Therefore, the predicted positions of the periodic bands can be expressed as

$$b_k = o + (k-1)\Delta b, \quad k = 1, \ldots, p.$$

The cost function is defined as the mean square error between the predicted positions of the periodic bands and the true data. It can be represented as shown in Eq. (3.6).

$$\phi = \frac{1}{p} \sum_{\mathrm{i}=1}^{N} m_{\mathrm{i}} \left(o + \Delta b (\sum_{\mathrm{j}=1}^{\mathrm{i}} m_{\mathrm{j}} - 1) - b_{\mathrm{i}} \right)^2 \tag{3.6}$$

To find the optimal solution of o and Δb for a given membership vector, we take the first derivatives of the cost function with respect to two variables, o and Δb, respectively. Then, we apply the first order necessary condition (FONC). The closed-form optimal solution for a fixed membership vector $\vec{m}$ can be obtained by solving the linear equations. The solutions are shown in Eq. (3.7) and Eq. (3.8).

$$\hat{o}^{(\vec{m})} = \frac{2(2p-1)}{p(p+1)} \sum_{\mathrm{i}=1}^{N} m_{\mathrm{i}} b_{\mathrm{i}} - \frac{6}{p(p+1)} \sum_{\mathrm{i}=1}^{N} (\sum_{\mathrm{j}=1}^{\mathrm{i}} m_{\mathrm{j}} - 1) m_{\mathrm{i}} b_{\mathrm{i}} \tag{3.7}$$

$$\hat{\Delta b}^{(\vec{m})} = \frac{12}{p(p+1)(p-1)} \sum_{\mathrm{i}=1}^{N} m_{\mathrm{i}} (\sum_{\mathrm{j}=1}^{\mathrm{i}} m_{\mathrm{j}} - 1) b_{\mathrm{i}} - \frac{6}{p(p+1)} \sum_{\mathrm{i}=1}^{N} m_{\mathrm{i}} b_{\mathrm{i}} \tag{3.8}$$

However, p and $\vec{m}$ are unknown. In order to find the best solution, this algorithm uses an exhaustive search with these two variables. The flow is described in Fig. 3.3. The input is the true data $\vec{b}$ and the total number of candidate bands N. The initial value for parameter p is 3. For a given p, we have $\binom{N}{p}$ possible combinations of the membership vector. $M_{possible}$ denotes the set of possible membership vectors. For each membership vector in $M_{possible}$, we compute the optimal position of the first periodic band and the optimal repetitive interval by Eq. (3.7) and Eq. (3.8). Then, the cost function value can be obtained from these two values for a given membership vector. The relationship is described in Eq. (3.9).

$$\phi^{(\vec{m})} = \frac{1}{p}\sum_{\mathrm{i}=1}^{N} m_{\mathrm{i}} \left(\hat{o}^{(\vec{m})} + \hat{\Delta b}^{(\vec{m})} (\sum_{\mathrm{j}=1}^{\mathrm{i}} m_{\mathrm{j}} - 1) - b_{\mathrm{i}} \right)^2 \tag{3.9}$$

After obtaining the cost function values for all $\binom{N}{p}$ possible membership vectors in $M_{possible}$, the optimal result for the given p is the minimum cost function value ϕ_p. So we save its corresponding estimated repetitive interval $\hat{\Delta b}_p$ and membership vector $\vec{m}_p$. We repeat the above process until we finish the computations for $p = 3, 4, \ldots, N$. In the last step, we compute the fitting error defined in Eq. (3.10), which is the cost function value normalized to its repetitive interval. At the end of this phase, we have the fitting error ϵ_p and its corresponding repetitive interval and membership vector, for $p = 3, 4, \ldots, N$. How do we choose the best solution from this set of possible solutions? There are two aspects that we need to consider. The first is the fitting error. However, we cannot just choose the minimum one. Because when p is small, it is easy to find equally spaced bands. Thus, the fitting error is small. In previous work, the criterion was to choose the maximum p from those with fitting error less than 5% [3].

$$\epsilon_p = \frac{1}{\hat{\Delta b}_p}\sqrt{\phi_p} \tag{3.10}$$

Since our samples are more noisy, there is a larger probability to find multiple equally spaced band sequences with different repetitive intervals. In addition, missing periodic bands might affect the estimation because the criterion tends to choose the solution with maximum p. In our previously described method, we only determined a single repetitive interval. That is, we selected from the candidate sets of periodic bands with fitting error less than 5% that

set with the maximum number p of periodic bands. To solve this problem, we introduce synthetic missing bands and re-evaluate the cost function value to find the best solution. To begin with, we choose candidate sets of periodic bands with fitting error less than 5% as our candidates for Phase 2.

Phase 2: Adding synthetic missing bands. For each candidate set, we check whether there is an existing band at twice the interval away from either end of the set of periodic bands. If no such band exists, we check the next candidate set. On the contrary, if such a band at twice the interval does exist on either end of the candidate set of bands, we add that band and a new synthetic band that is equally spaced between the band at the end of the set and the new band to update this candidate set of periodic bands. Note that it is possible that additional bands are located at the repetitive interval, beyond the new band that is added to the candidate set. In this case, these bands will also be added to the candidate set. In fact, we may end up merging two candidate sets of periodic bands. After that, we have a new set of fitting errors corresponding to our updated sets of periodic bands. If the smallest fitting error is smaller than our best-so-far fitting error, then we update the best-so-far solution. We repeat this process until all candidate sets have been checked. Finally, we apply the same criterion as in the previous work [3] to select the best one. That is, we choose the set with maximum p from those sets for which the fitting error is less than 5%. The overall repetitive interval estimation algorithm is described in Fig. 3.4.

In our previous work, we also found missing bands. However, we only used one candidate set. We chose the one best result from the first cost function estimation and then continue adding bands one interval away until the set of periodic bands expands to fill the whole page. The purpose for our previous method is to find as many periodic bands as possible. However, here our target is to improve the accuracy. The key idea of our proposed method is searching more possibilities and using less synthetic data. We use more candidates from Phase 1 initial guess and only add one or two synthetic bands in Phase 2 for each candidate set.

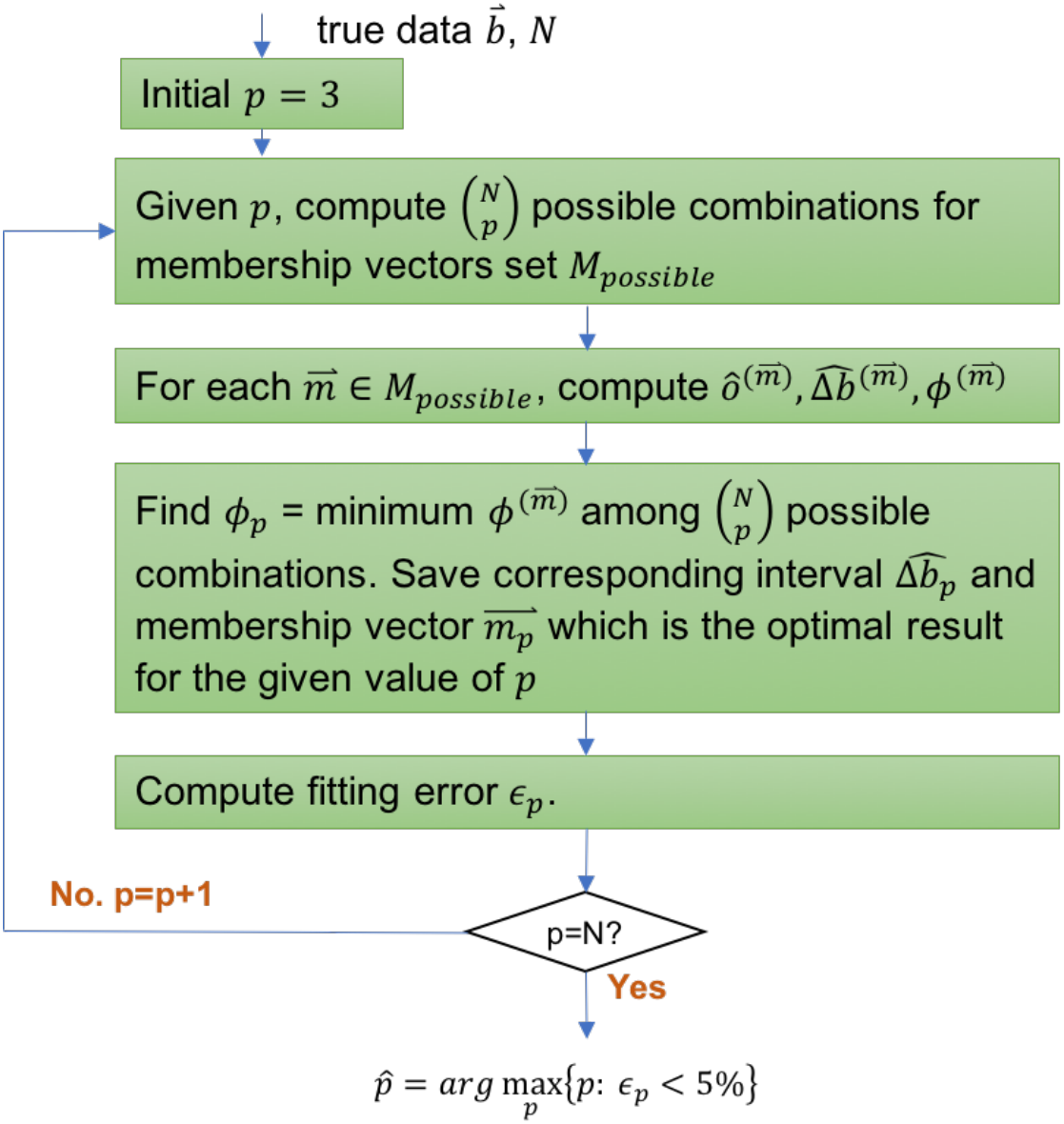


Figure 3.3. Cost function estimation algorithm.

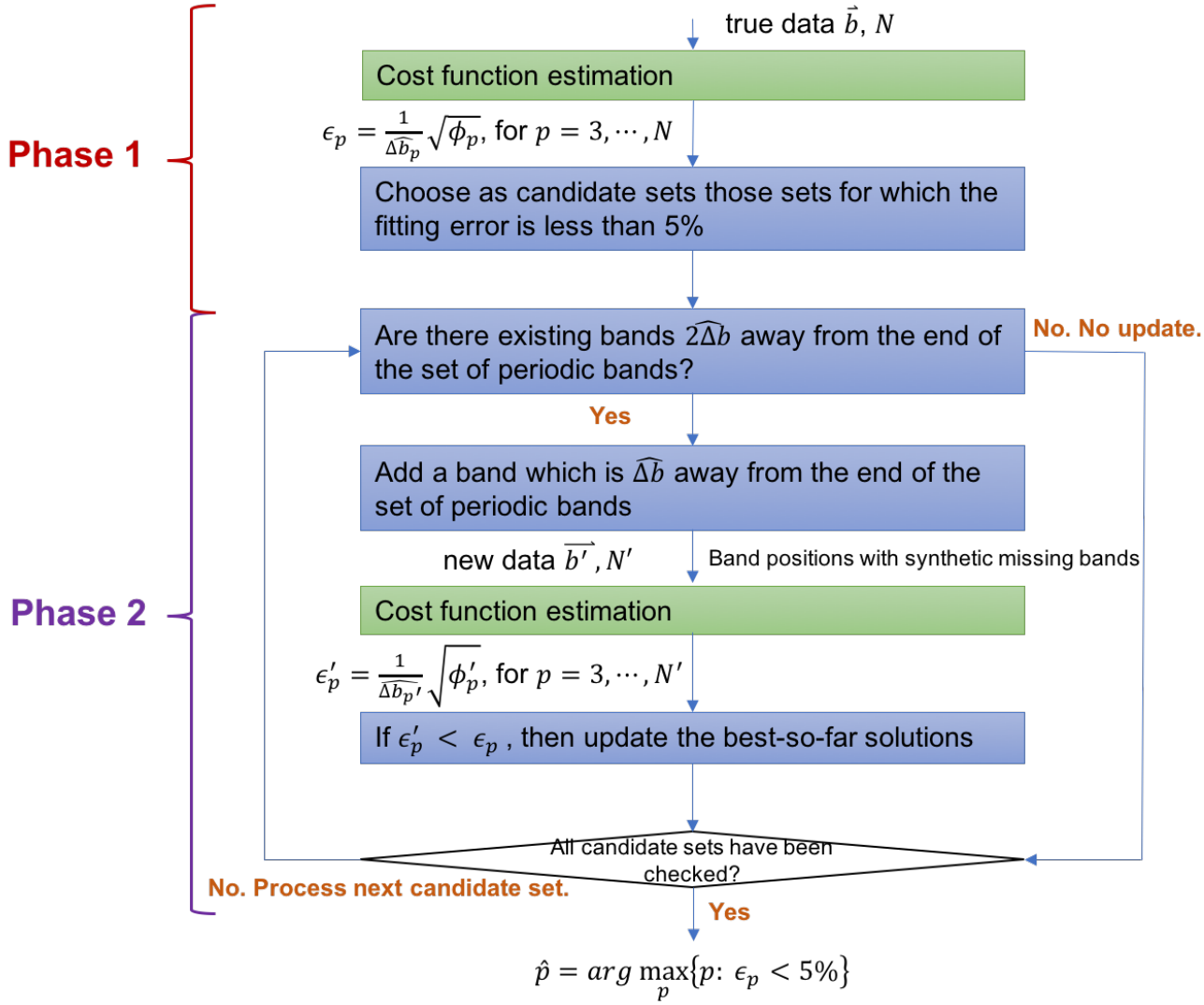


Figure 3.4. Proposed repetitive interval estimation algorithm.

Bands identification for periodic and aperiodic bands

After the estimated repetitive interval is determined, the corresponding membership vector is used to identify the periodic and aperiodic bands. Thus, we can collect the statistic values, like maximum, and average strength of periodic bands for our feature vector.

3.3 Experimental Results

This section shows the experimental results of logistic regression and applying our algorithm on the test pages.

To classify whether the potential defects are visible or invisible, we use maximum ΔE, minimum sharpness of two sides, and width as our selected features in the logistic regression algorithm. We have a total of 18 sample pages and 1693 labeled bands on those sample pages. We apply K-fold cross validation; K is 9 in this work. First, we partition the data into 9 groups randomly; 8 groups are used for training and 1 group is used for testing. We repeat for all 9 groups. The result is shown in Table 3.1. Here, we provide several measures of performance in addition to accuracy. TP and FP are true positive and false positive, respectively. Similarly, TN and FN are true negative and false negative, respectively. Precision is defined as $\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$. Specificity is defined as $\frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}$. And sensitivity is given by $\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$. Then, we reshuffle the data 100 times. From Fig. 3.5, we can see the accuracy is very stable. The average accuracy of K-fold cross validation for visible-invisible band defects classification is 93.4%.

In this project, the test pages are softcopies of constant tone printed from a color laser electrophotographic printer and scanned at 600 dpi. We are only interested in the smooth area since the bands defect in smooth areas is more obvious for our perception [22].

Figure 3.6 is an example of the detection result. The yellow lines separate the three regions. The blue lines are the projection data in signed ΔE. The black lines are the threshold values we use to find the peaks. The red bars are periodic bands; and the green bars are aperiodic bands.

Figure 3.6(a) and Figure 3.6(b) are the same test page, but estimated by different methods. Usually, the bands defect is most obvious in the center part. Thus, we check the center

part only. In Figure 3.6(a), the estimated repetitive interval is 12.2 mm by the cost function method without adding synthetic missing bands. However, the estimated repetitive interval is 33.62 mm by our proposed algorithm, which is the cost function method with added synthetic missing bands. The blue bar shown in Figure 3.6(b) is the position where we add the synthetic missing band. The ground truth is 34 mm. Similarly, Fig. 3.7 is another example for the two estimation methods applied on the same test page. The cost function method predicted the repetitive interval to be 42.64 mm on this sample page. But the repetitive interval estimated by our proposed method is 33.66 mm. The ground truth is the same: 34 mm. Therefore, our method can estimate the repetitive interval better than the previous method on these test pages.

The total number of test pages with obvious periodic bands we have in this project is 15. We apply our proposed method to these test pages and the results are shown in Table 3.2.

3.4 Conclusion

In this chapter, we built a classification model by logistic regression to determine whether the potential defects are visible or invisible. The average result achieves 93.4% accuracy. In addition, we proposed a new cost-function-based repetitive interval estimation method. We re-evaluate the cost function values by combining true data with synthetic missing bands to improve the accuracy on noisy and corrupted test sample pages.

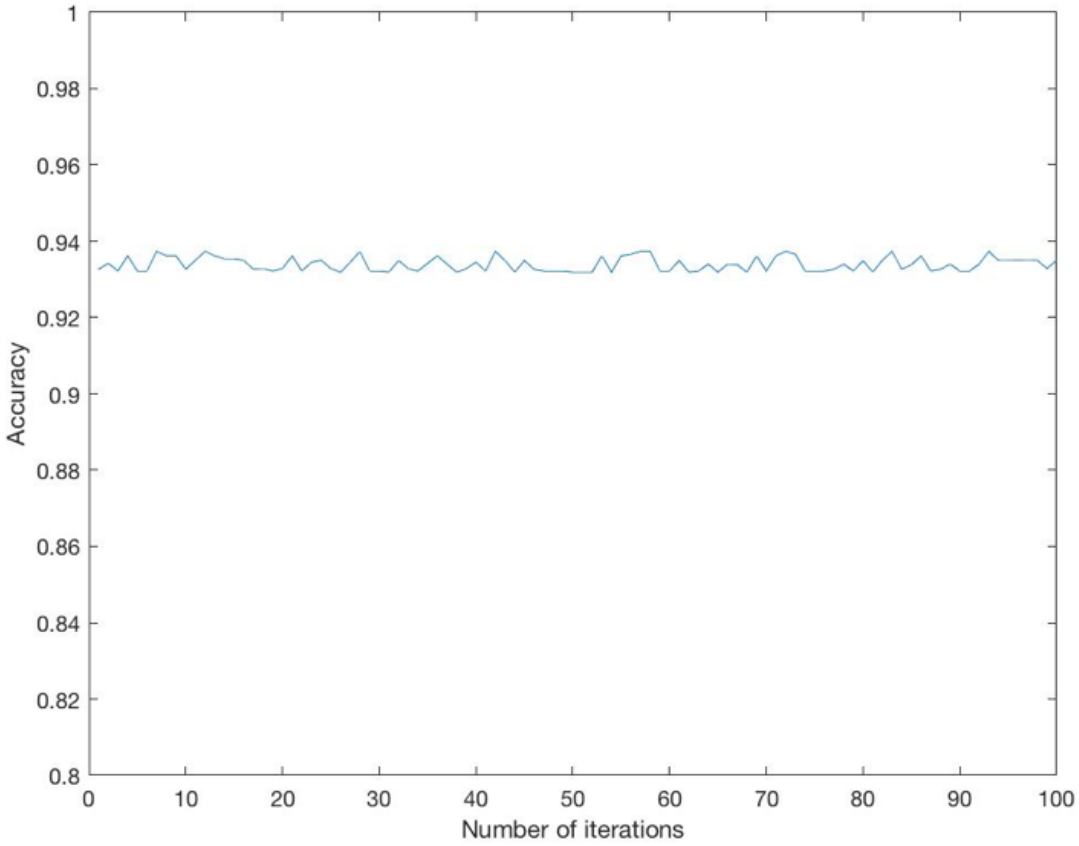


Figure 3.5. 9-fold cross validation with reshuffling the data 100 times. Each iteration is the average of 9 tests for one partitioning of the data.

Table 3.1. Logistic regression with K-fold cross validation result for classification of visible and invisible potential defects.

Fold index	FN	FP	TN	TP	Total	Accuracy	Precision	Specificity	Sensitivity
1	4	7	101	85	197	0.9442	0.9239	0.9352	0.9551
2	10	2	73	78	163	0.9264	0.9750	0.9733	0.8864
3	5	7	88	69	169	0.9290	0.9079	0.9263	0.9324
4	7	6	113	77	203	0.9360	0.9277	0.9496	0.9167
5	11	0	107	60	178	0.9382	1	1	0.8451
6	1	5	115	85	206	0.9709	0.9444	0.9583	0.9884
7	6	9	114	88	217	0.9309	0.9072	0.9268	0.9362
8	10	1	98	70	179	0.9385	0.9859	0.9899	0.8750
9	9	7	110	55	181	0.9116	0.8871	0.9402	0.8594
					Average	0.9362	0.9399	0.9555	0.9105

Table 3.2. Comparison of repetitive interval estimation result by histogram method, cost function method, and cost function method with adding synthetic missing bands. The ground truth interval for these samples is 34 mm.

Sample ID	Histogram	Cost function	Cost function with adding synthetic missing bands
1	19.81	33.67	33.67
2	32.43	33.91	33.91
3	11.47	33.88	33.88
4	NaN	32.99	32.99
5	28.87	33.76	33.76
6	28.66	31.91	31.91
7	13.38	33.68	33.68
8	30.95	33.97	33.64
9	25.23	33.66	33.68
10	NaN	12.2	33.62
11	33.82	33.83	33.83
12	26.67	33.58	33.58
13	32.72	42.64	33.66
14	9.19	33.97	33.8
15	14.1	33.71	33.71

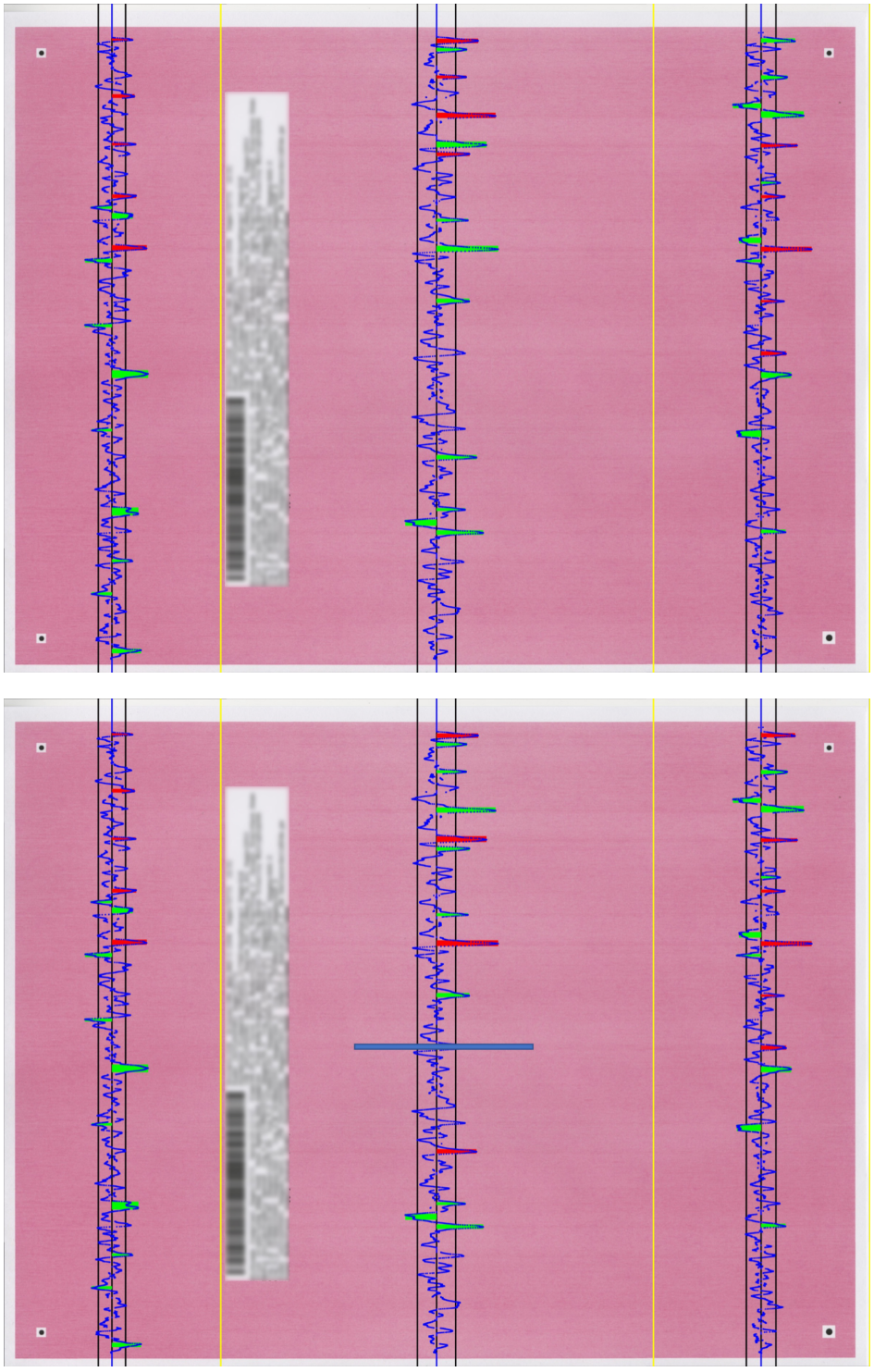


Figure 3.6. Comparison of estimated repetitive interval on the same test page. The red bands have been identified as periodic bands. The green bands are aperiodic bands. The ground truth repetitive interval is 34 mm. (a) Estimated repetitive interval is 12.2 mm by cost function method. (b) Estimated repetitive interval is 33.62 mm by cost function method with adding synthetic missing bands. The blue bar is the synthetic missing band.

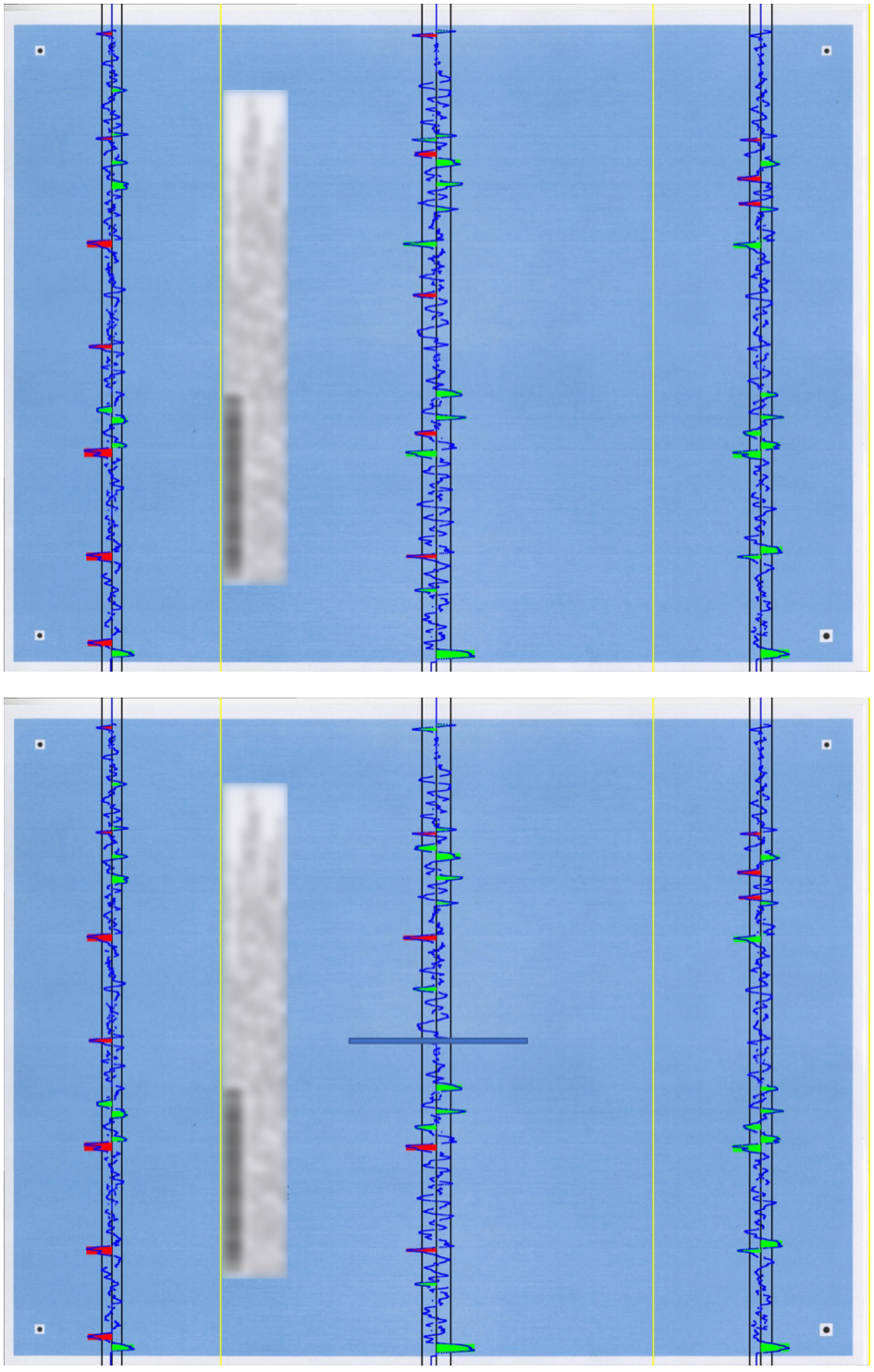


Figure 3.7. Comparison of estimated repetitive interval on the same test page. The red bands have been identified as periodic bands. The green bands are aperiodic bands. The ground truth repetitive interval is 34 mm. (a) Estimated repetitive interval is 42.64 mm by cost function method. (b) Estimated repetitive interval is 33.66 mm by cost function method with adding synthetic missing bands. The blue bar is the synthetic missing band.

4. ACOUSTIC SIGNAL AUGMENTATION

4.1 Acoustic signal analysis

4.1.1 Previous work

Yutong Xue and Xihui Wang proposed an acoustic signal detector for HP printers to characterize the printer sounds [23]. The detector can be divided into two parts as shown in Fig. 4.1. One is strong tone detection and the other is modulation detection.

In strong tone detection, firstly, the power spectral density (PSD) is estimated by Welch's method. That is, the time-varying signal is segmented into equal-length overlapping time frames and then the PSD of each frame is computed. The final estimation is the average value over all time frames. Once the estimated PSD is obtained, the detector computes a dynamic threshold by a moving average filter. Last, we compare the PSD with the dynamic threshold. The frequencies where the peaks are located are called strong tone frequencies.

In modulation analysis, the detector analyzes the information within a frequency range around each strong tone frequency. To achieve this, a Butterworth bandpass filter is applied to the raw signal. After filtering the signal, the detector uses the Hilbert transform to generate the imaginary part of the complex analytic signal. Thus, the instantaneous amplitude can be extracted by taking the absolute value of the analytic signal. Then, the PSD of the instantaneous amplitude is computed by the fast Fourier transform (FFT). And the modulation depth, which is defined in this work, as the PSD of the instantaneous amplitude divided by the estimated PSD of the strong tone frequency can be evaluated. The frequencies where the peaks of the modulation depth are located are called modulation frequencies.

4.1.2 Conventional augmentation methods

There are plenty of acoustic data augmentation methods adopted in various works and applications. They can be categorized roughly into six types.

1. Time perturbation: change the speed of sound while keeping the pitch and spectral envelope unchanged [24]. Some works argue that simply resampling the signal with shifts in the frequency domain can achieve better performance [25] [26].

2. Frequency perturbation: vocal tract length perturbation (VTLP) is widely used in speech

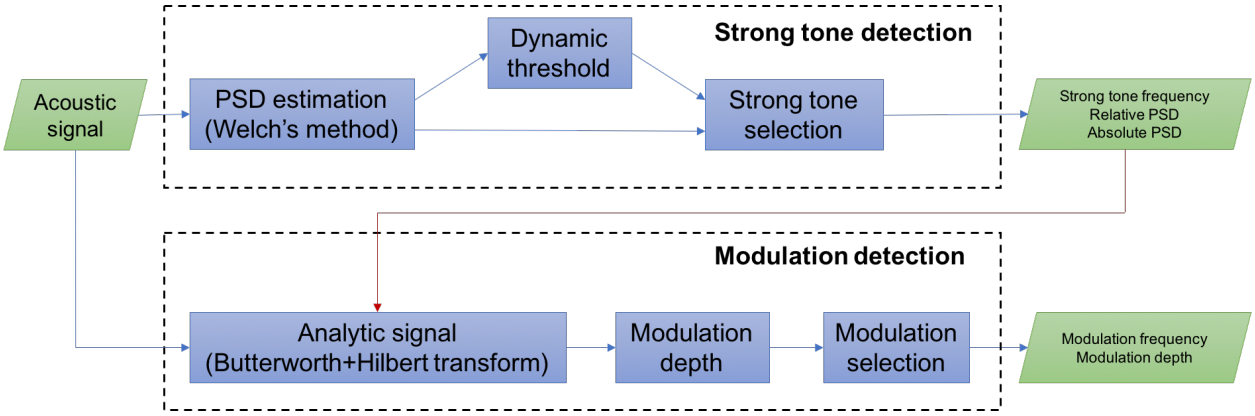


Figure 4.1. Detector.

recognition. The frequency is warped by a random warp factor for each utterance [27]. In environmental sounds classification [24], the pitch is changed while keeping the duration unchanged.

3. Amplitude scaling and gain: apply different linear functions to different input amplitude levels to boost or suppress the magnitude of loudness [24].

4. Adding background noise from external data: mix the sample with another background sound to generate new samples [24].

5. Mixing within class: the assumption is that if the two different sounds from the same class are mixed, the result still belongs to the same class [28] [29].

6. Cropping: there are many variations for cropping techniques: [30] randomly delay the samples; [31] split each sample into overlapping fixed length segments; [32] zero pad two sides of the sound and randomly crop it.

In the initial analysis, we conduct an experiment with a real printer sound. The detector result of the original sound is shown in Fig. 4.2. The bottom figure is the power spectral density and the black crosses are detected peaks. The corresponding frequencies on the x-axis are strong tone frequencies. The top figure shows modulation frequency versus strong tone frequency. For example, one of the strong tone frequencies is 3628 Hz, which has modulation frequency 9.6 Hz.

In order to know how conventional augmentation methods affect the characteristics of the printer sound, we picked time stretching and pitch shifting in this experiment. In time stretching, we apply the short time Fourier transform (STFT) first to separate temporal information from spectral information. Then, we use a phase vocoder to maintain the pitch [33]. More specifically, the phase vocoder constructs new time steps based on the speed ratio and interpolates the magnitudes. For example, if we speed up the original signal, we will get a smaller total number of time frames. At the same time, we track the phase difference of the two original time frames to keep the same rate of angular rotation. Lastly, the synthesized acoustic signal is obtained by the inverse short time Fourier transform (ISTFT) of the modified result of the STFT. Pitch shifting is a combination of time stretching and resampling [34]. First, we use time stretching to obtain a speed changed but pitch unchanged version of the original signal. Then, we resample the intermediate result to recover the

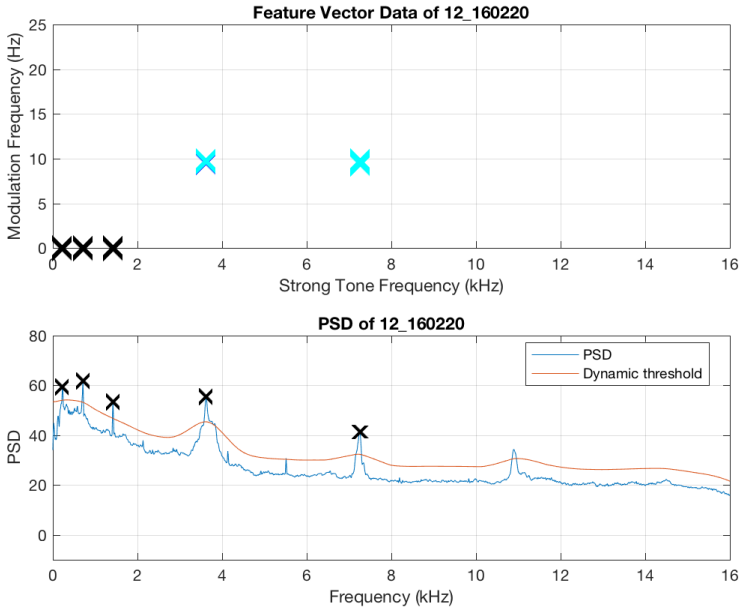


Figure 4.2. Detector result of original sound.

original speed. Because the second step is just simple resampling, it causes the pitch shifting. The two methods are described in Fig. 4.3.

Next, we applied time stretching on the real printer sound. We used speed up and slow down 20% as the examples. The results are shown in Fig. 4.4. The strong tone frequencies are unchanged on both versions. On the other hand, the modulation frequencies changed along with the speed. For the sample of speed up 20%, the modulation frequency increased from 9.6 Hz to 11.5 Hz. Similarly, the modulation frequency decreased from 9.6 Hz to 7.6 Hz for the slow down version. The other trial is pitch shifting shown in Fig. 4.5. We raise and lower 4 semi-tones which are equivalent to an increase of 25.99% and a decrease of 20.63% in Hz. The modulation frequencies are unchanged on both versions. However, the strong tone frequencies changed along with the pitch. The strong tone frequency 3628 Hz in the original sound shifted to 4565 Hz in the higher pitch version and to 2875 Hz in the lower pitch version.

Therefore, we conclude that the strong tone frequency is related to the spectral envelope; and the modulation frequency is related to the speed.

4.1.3 Instantaneous amplitude and instantaneous frequency

To further understand the printer sounds, the analysis of instantaneous amplitude and instantaneous frequency is applied. Firstly, we leverage the second part of the detector. The narrow band signal is extracted by a Butterworth filter for each strong tone frequency. Then, we use the Hilbert transform to generate the complex analytic signal. Thus, we can obtain the instantaneous amplitude by simply taking the absolute value of the analytic signal. The instantaneous amplitude is the envelope of the narrow band signal. Lastly, we compute the unwrapped phase of the analytic signal, and use a 121-point differentiator to evaluate the instantaneous frequency. The process is shown in Fig. 4.6.

Here, we use the real printer sound whose file name is "blank1.wav" as our example. Figures 4.7 and 4.8 show instantaneous amplitude and instantaneous frequency, respectively. The detector lists the five strong tone frequencies with largest amplitude. Therefore, Fig. 4.7 shows the instantaneous amplitude computed by the narrow band filters around the strong tone frequencies 11895, 13148, 14414, 1992, and 352 Hz. Figure 4.8 shows the corresponding

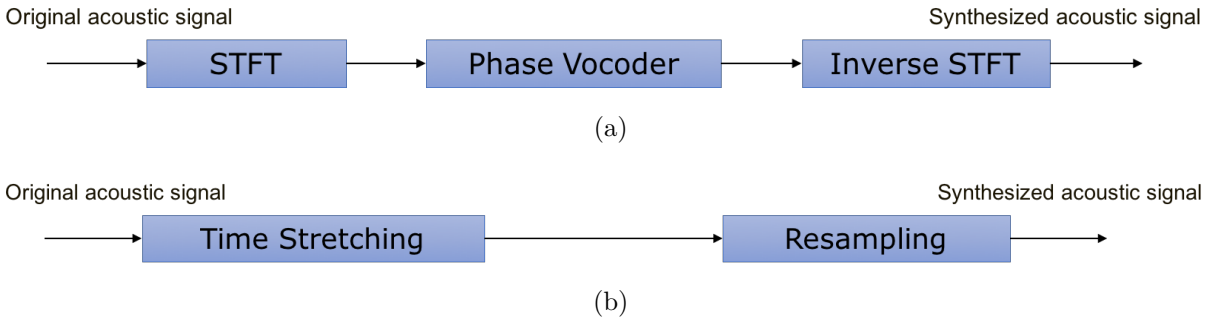
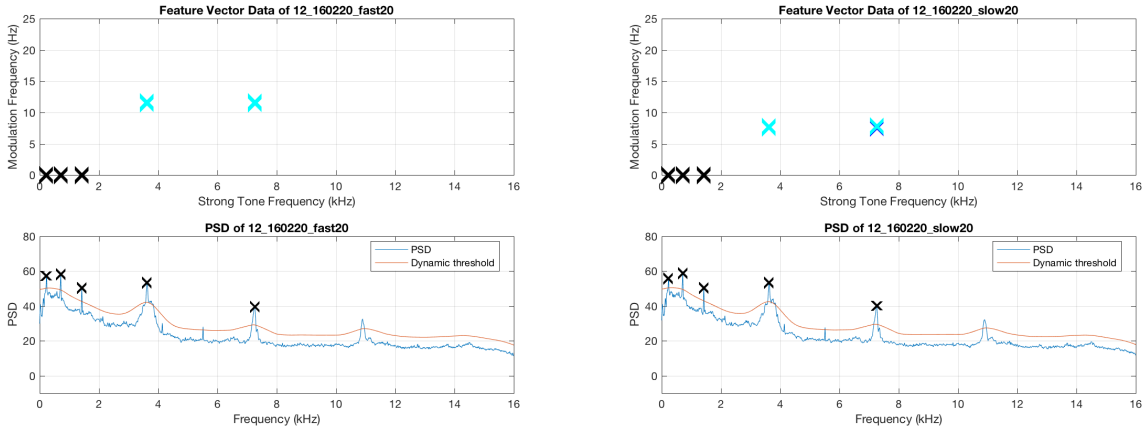
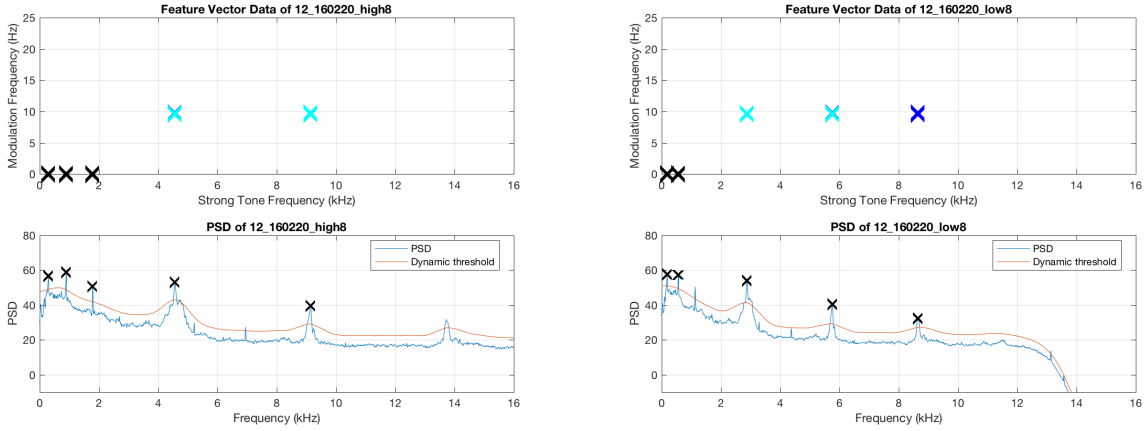


Figure 4.3. (a) Time stretching. (b) Pitch shifting.



(a) (b)

Figure 4.4. (a) Speed up 20%. (b) Slow down 20%.



(a) (b)

Figure 4.5. (a) Higher pitch +25.99%. (b) Lower pitch -20.63%.

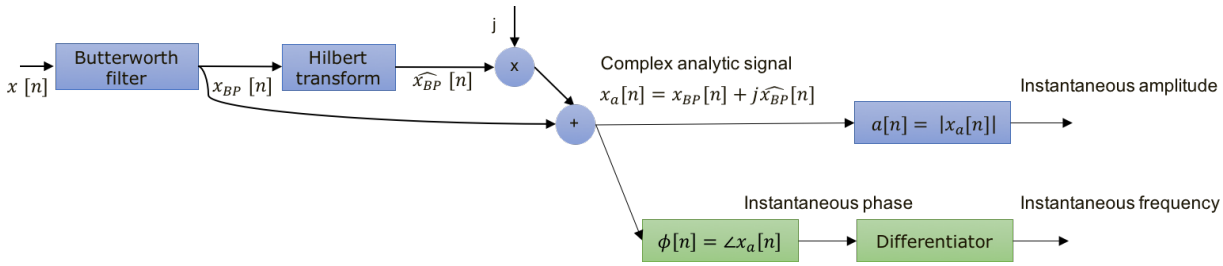
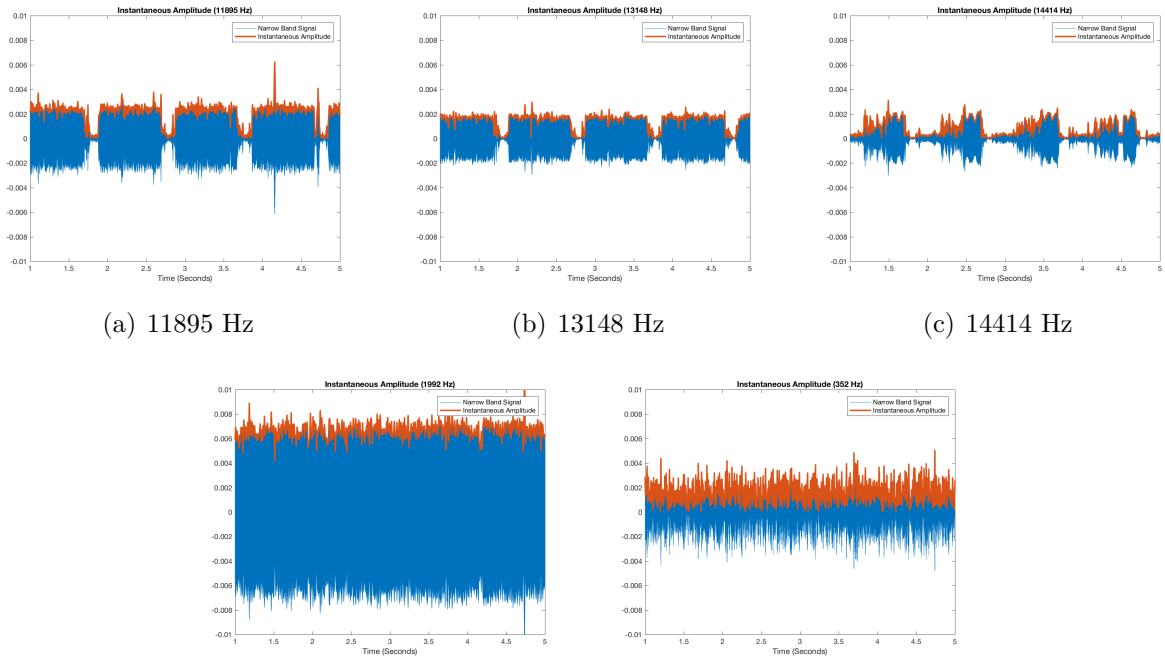


Figure 4.6. Instantaneous amplitude and instantaneous frequency analysis.

instantaneous frequency signals. We observe some interesting results in this analysis. The three narrow bands (a) - (c) show periodicity. Even though (d) and (e) look like noise, they still have some very different characteristics. The narrow band of (d) has very small variation in instantaneous frequency. However, the narrow band of (e) shows very large variation in instantaneous frequency.

4.1.4 Histogram of dataset

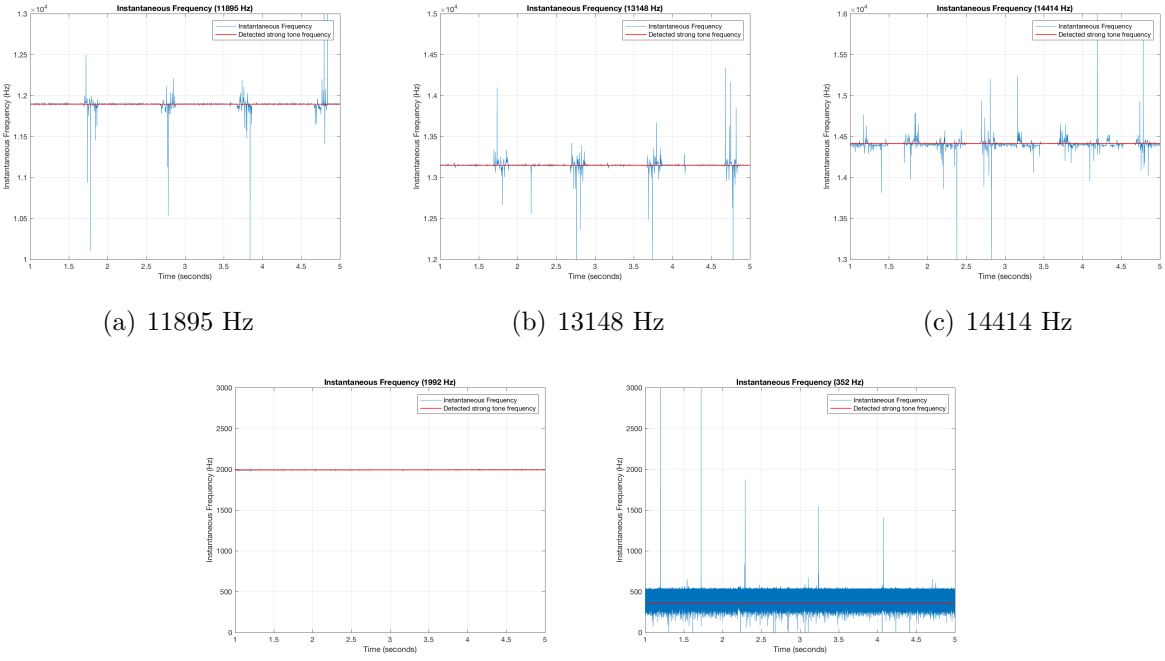
Lastly, we would like to examine the histogram of the whole dataset to better understand the variations between the samples. We have a total of 416 sound files. Each is around 10 seconds duration. Figure 4.9 shows the histogram of strong tone frequency. Around 2000 Hz and 14500 Hz, there is a spread in the distributions in the histogram. We interpret this that there are larger variations sample by sample for strong tone frequencies around 2000 Hz and 14500 Hz. Moreover, the four dominant strong tone frequencies among the samples are 10641 Hz, 11895 Hz, 12012 Hz, and 13148 Hz. On the other hand, there is not much variation in the modulation frequency. The histogram of the modulation frequencies is shown in Fig. 4.10. The detector usually detects 1 Hz and 2 Hz for all samples.



(a) 11895 Hz (b) 13148 Hz (c) 14414 Hz

(d) 1992 Hz (e) 352 Hz

Figure 4.7. Instantaneous amplitude. The blue part is the narrow band signal and the orange envelope is the instantaneous amplitude.



(a) 11895 Hz (b) 13148 Hz (c) 14414 Hz

(d) 1992 Hz (e) 352 Hz

Figure 4.8. Instantaneous frequency. The blue part is the instantaneous frequency and the red part is the detected strong tone frequency.

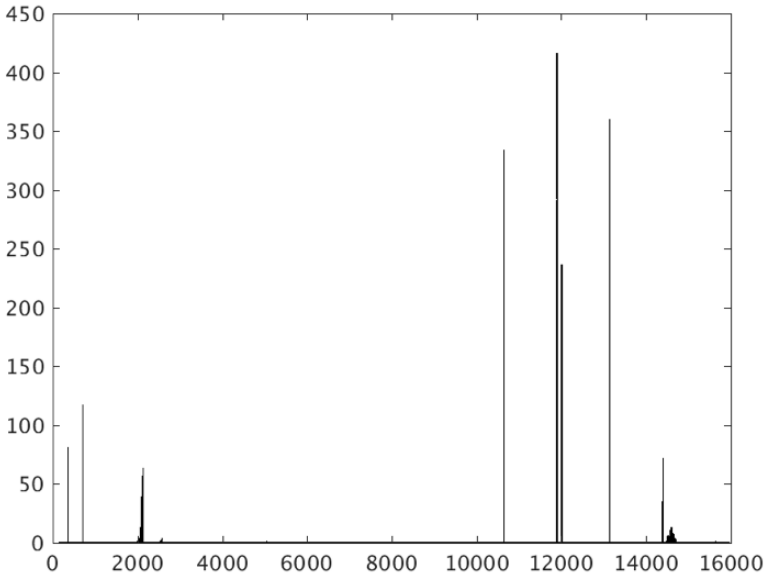


Figure 4.9. Histogram of strong tone frequencies.

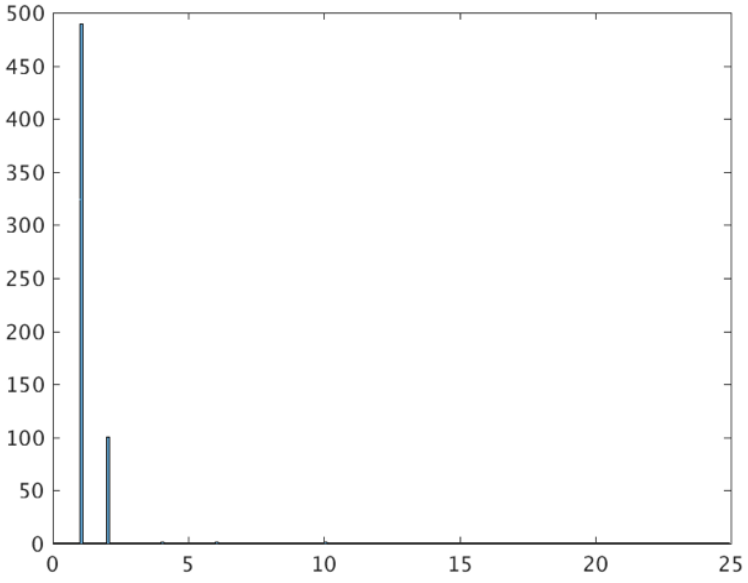


Figure 4.10. Histogram of modulation frequencies.

4.2 Acoustic signal augmentation

4.2.1 Sinusoidal model with amplitude modulation

The Fourier series models a complicated signal as a linear combination of simple sine and cosine waves. Hence, we use the sinusoidal model as our first step, since the goal is to generate an additional strong tone frequency with corresponding modulation properties. We also need to incorporate amplitude modulation in the model. The carrier signal $c(t)$ is defined in Eq. (4.1) which is a sine wave with strong tone frequency f_{ST} and amplitude A. Then, the modulating signal $m(t)$ is another sine wave with modulation frequency f_m and amplitude M. We can rewrite M as the product of A and m, where m is the amplitude ratio of the modulating signal to the carrier signal. That is, $m = \frac{M}{A}$. The modulating signal is defined in Eq. (4.2). Therefore, the modulated signal $y(t)$ is given by Eq. (4.3), based on amplitude modulation [35].

$$c(t) = A\sin(2\pi f_{ST}t) \tag{4.1}$$

$$m(t) = M\sin(2\pi f_m t) = Am\sin(2\pi f_m t) \tag{4.2}$$

$$y(t) = \left[1 + \frac{m(t)}{A}\right] c(t) = [1 + m\sin(2\pi f_m t)]c(t) \tag{4.3}$$

Because amplitude modulation is linear, we can extend Eq. (4.3) to Eq. (4.4). The summation of the multiple modulating signals achieves multiple modulation frequencies. We use this model as our defect generator and add the artificial defect to the original acoustic signal to generate the synthesized acoustic signal. The process is described in Fig. 4.11.

$$y(t) = \left[1 + \sum_{\mathrm{i}=1}^{N} m_{\mathrm{i}}\sin(2\pi f_{m_{\mathrm{i}}}t)\right] c(t) \tag{4.4}$$

For example, if we want an additional strong tone frequency 5000 Hz with modulation frequencies 1 Hz and 2 Hz, we can set $f_{ST} = 5000$, $f_{m_1} = 1$, and $f_{m_2} = 2$. In addition, the amplitude of carrier A and modulation indices m_{i} can be tuned to obtain the magnitude of

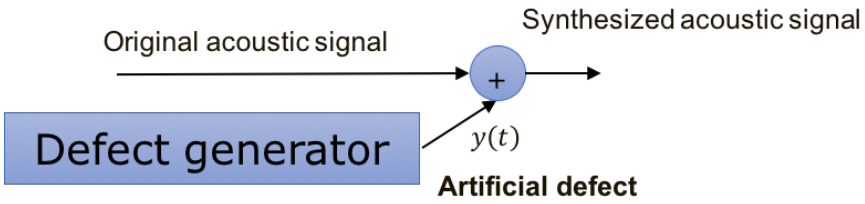


Figure 4.11. Synthesis of acoustic signal corresponding to a defect by the sinusoidal model with amplitude modulation.

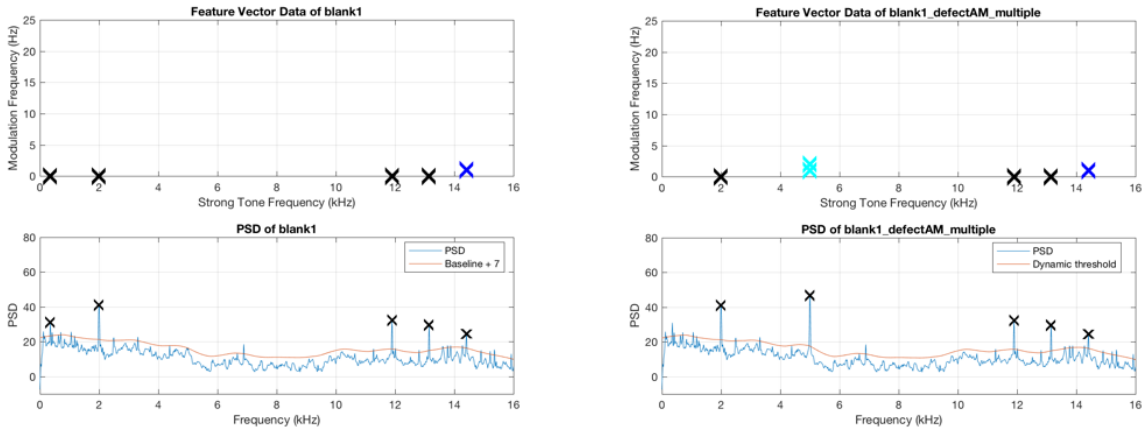
the strong tone and magnitude of modulation that we want. In order to have a magnitude that is large enough for the detector to detect, we use $A = 0.01$, $m_1 = 0.6$, and $m_2 = 0.6$ in this example. The result is shown in Fig. 4.12. In the synthetic sound, the detector detects an additional strong tone frequency around 5000 Hz; and this strong tone frequency has modulation frequencies 1 Hz and 2 Hz.

In our other trial, we approximate one of the real strong tone components. We choose strong tone component 11895 Hz detected from "blank1.wav". The parameters are $A = 0.01$, $f_{ST} = 5000$, $f_{m_1} = 1$, $f_{m_2} = 2$, $f_{m_3} = 3$, $m_1 = 0.32$, $m_2 = 0.25$, and $m_3 = 0.16$. The result is shown in Fig. 4.13. Even though we can get similar a result from the detector, the narrow band signals have different shapes.

4.2.2 Mixing with external dataset

It is very hard to collect abnormal sounds for anomaly detection in machine sounds. There are two reasons. First, it is infeasible to damage the target machine deliberately to emulate the failing conditions naturally. Second, the actual anomalous sounds occur rarely and may vary a lot. Therefore, Koizumi et al. [36] proposed a method to synthesize abnormal sounds. They observed that abnormal machine sounds can categorized into two types: collision sounds and sustained sounds. They used the external dataset DCASE2016 [37] to generate synthetic abnormal samples. However, we should rethink the problem in our case. For example, the sounds of pages being turned is used as abnormal sustained sounds for their case. On the contrary, this should be normal in the recording environment of a functioning printer. We can generate noisy samples by mixing pages turning sounds. And the noisy samples should be included in dataset to make the classifier more robust.

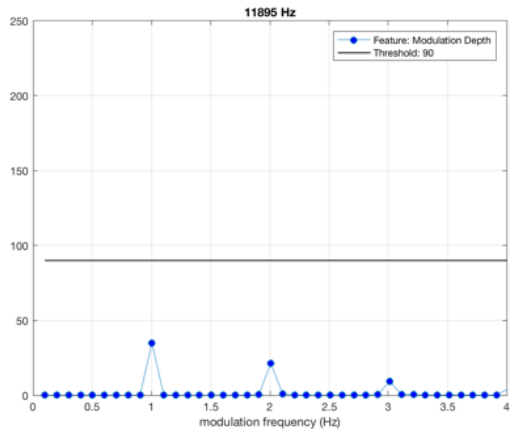
Because the external dataset and our dataset are obtained in different recording conditions, we need to adjust the energy to make it reasonable. Given a desired SNR_{db} as defined in Eq. (4.5). Then the noise is the energy of the external sound scaled by the energy adjustment ratio r which are defined in Eq.(4.6) and Eq. (4.7). Here E_{signal} is the median of the squared amplitude of the printer sound and $E_{external}$ is the median of the squared amplitude of the external sound [36].



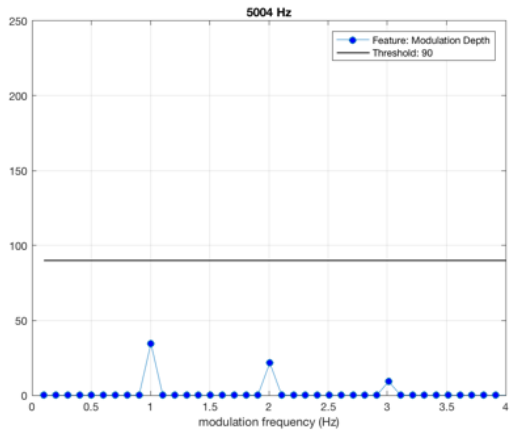
(a) Original sound

(b) Synthetic sound

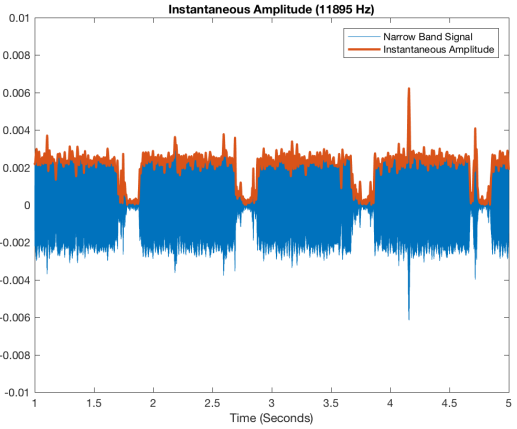
Figure 4.12. Example of sinusoidal model with amplitude modulation.



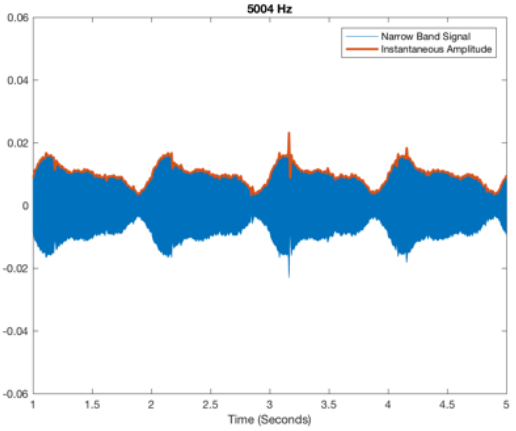
(a) Original sound



(b) Synthetic sound



(c) Instantaneous amplitude of original sound



(d) Instantaneous amplitude of synthetic sound

Figure 4.13. Example of sinusoidal model with amplitude modulation used to synthesize a strong tone with three modulation frequencies.

$$SNR_{db} = 10 \cdot \log\left(\frac{E_{sigmal}}{E_{noise}}\right) \tag{4.5}$$

$$E_{signal}10^{-\frac{SNR_{db}}{10}} = E_{noise} = E_{external} \cdot r \tag{4.6}$$

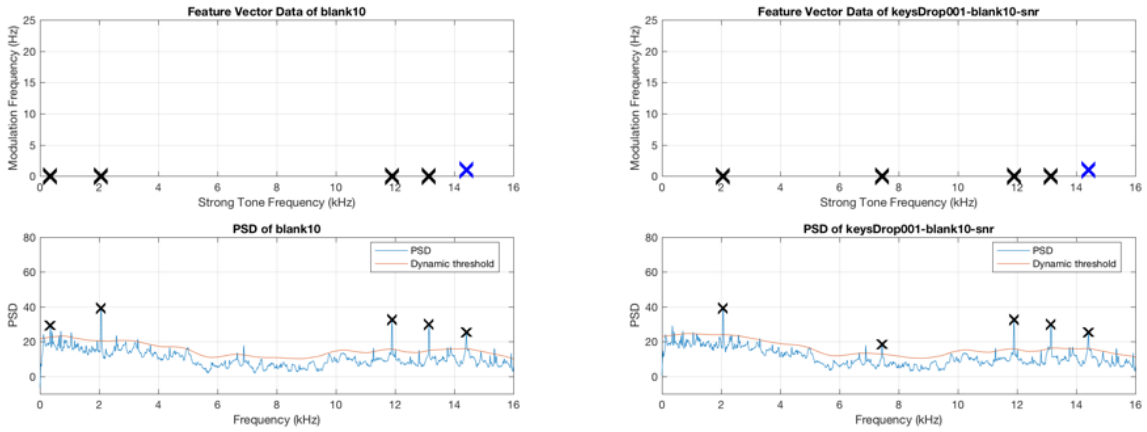
$$r = \frac{E_{signal}10^{-\frac{SNR_{db}}{10}}}{E_{external}} \tag{4.7}$$

We show two examples for a collision sound and a sustained sound, respectively. We use the keys dropping sound as a collision sound mixing with the printer sound. The result is shown in Fig. 4.14. When the energy of the external sound is large enough, the detector detects a new strong tone frequency (around 7500 Hz). In the other example, we use the pages turning sound as a sustained sound mixing with the printer sound. It is more difficult for the detector to detect the weakest strong tone frequency when the energy of the external sound is large enough. This is because the collision sound has concentrated energy in a narrow band, whereas the energy of the sustained sound is spread over a wide low spectrum.

4.2.3 Synthetic abnormal data

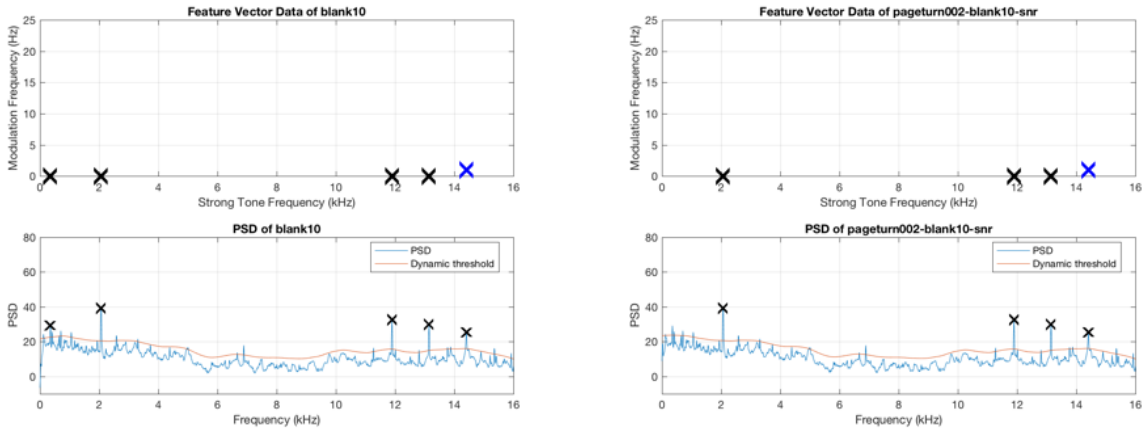
From the analysis of instantaneous amplitude and instantaneous frequency in Section 4.1.3, the different narrow-band signals exhibit different properties, such as the shape of the envelope, periodicity, and the variations in instantaneous frequency. Therefore, our key idea is that if the narrow-band signals can be related to some components, the abnormal printer sounds can be synthesized by changing a single narrow-band signal in frequency, time, and amplitude aspects.

First of all, we construct the short time Fourier transform (STFT) to obtain frequency-time information. Then, we choose one of the strong tone frequencies from the detector's result as our source frequency. To extract the source component from the spectrogram, a rectangular window centered at the corresponding source frequency bin with window length 7 is applied to the frequency domain for each time frame.



(a) Original sound (b) Synthetic with collision sound

Figure 4.14. Mixing printer sound with keys dropping sound.



(a) Original sound (b) Synthetic with sustained sound

Figure 4.15. Mixing printer sound with pages turning sound.

In frequency aspect, we shift the extracted source component to the destination frequency by replacing the frequency components centered at the corresponding destination frequency bin with the same window size in the spectrogram. Moreover, we eliminate the original strong tone frequency by replacing the frequency component at the source frequency bin with the average frequency component of the two side bins.

In the time aspect, we change the speed of the single narrow-band signal by a phase vocoder. Then, we replace the source component by the new time stretched version of it.

In the amplitude aspect, we increase the loudness of the single strong tone frequency component by just simply scaling the magnitude of the extracted source component.

Lastly, we reconstruct the signal using the inverse short time Fourier transform (ISTFT). Then, we have synthetic abnormal sounds in frequency, speed, and loudness. The process is shown in Fig. 4.16.

Figure 4.17 shows an example of the synthetic abnormal sound of frequency shifting. Our source component is the narrow band signal around strong tone frequency 14414 Hz and we shifted it to 5000 Hz. The detector result shows that we have new strong tone frequency at 4992 Hz; and the original strong tone frequency 14414 Hz disappeared. In addition, we examined the instantaneous amplitude of the narrow band signal around the new strong tone frequency 4992 Hz. It captured the characteristics of its source component well.

The speed perturbation examples are shown in Figs. 4.18 and 4.19. The original sound has around 10 red segments in the spectrogram. However, the speeded up version has around 12 segments in the spectrogram, and the slowed down version has around 8 segments in the spectrogram. The effect is also reflected in the detector's results. The source component strong tone frequency 14414 Hz has modulation frequency 1 Hz. In the speeded up synthetic sound, its corresponding modulation frequency changed to 1.2 Hz. And for the slowed down version, the source component has modulation frequency 0.8 Hz.

The last example is scaling the magnitude of a single strong tone frequency shown in Fig. 4.20. The source component is the same. The detector's result for the synthetic sound shows a much larger peak than the other strong tone frequencies.

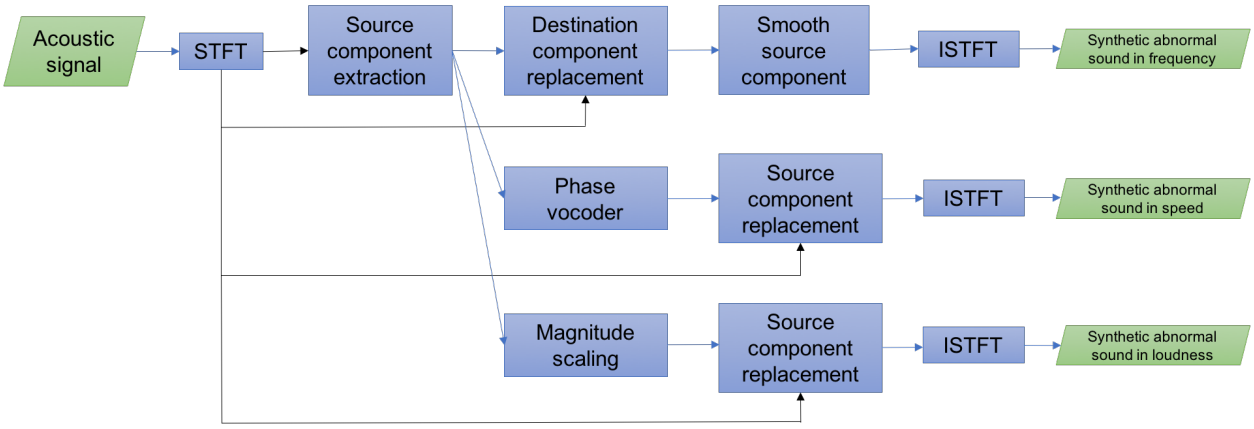
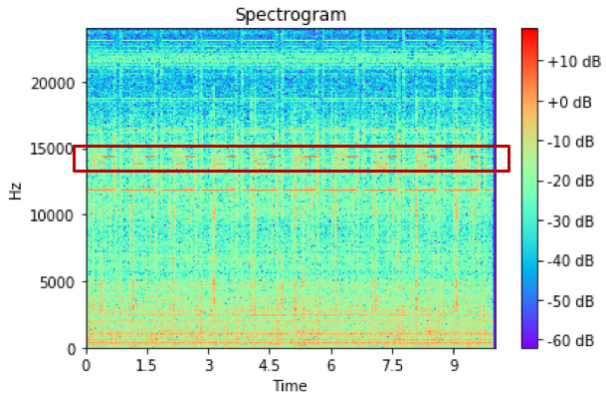
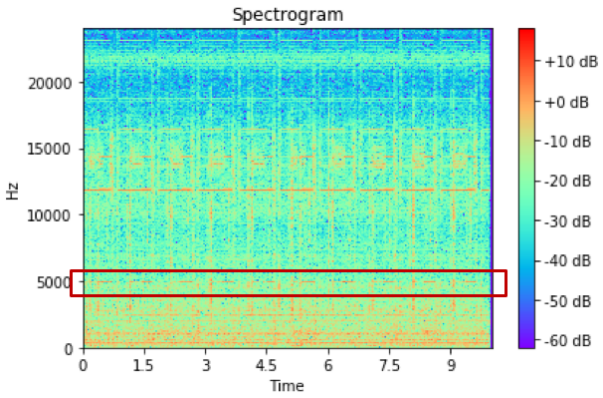


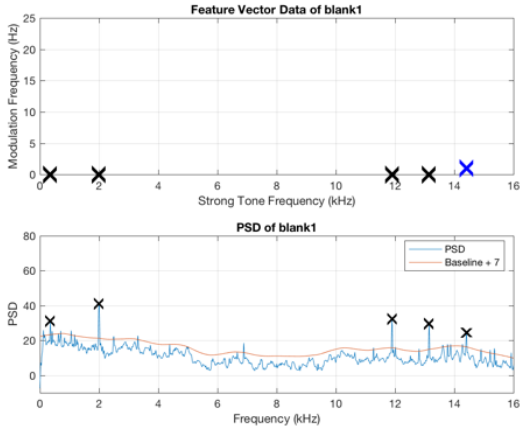
Figure 4.16. Proposed synthetic abnormal sounds.



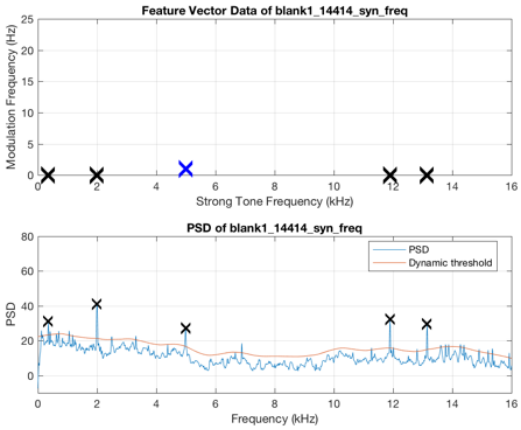
(a) Original spectrogram



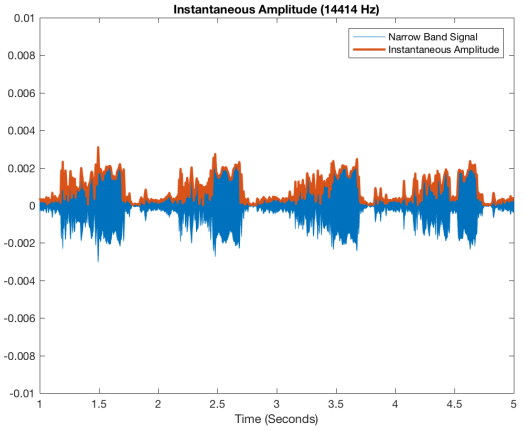
(b) Synthetic spectrogram



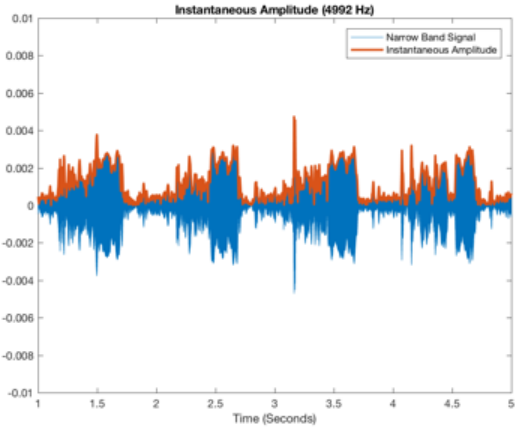
(c) Original detector result



(d) Synthetic detector result

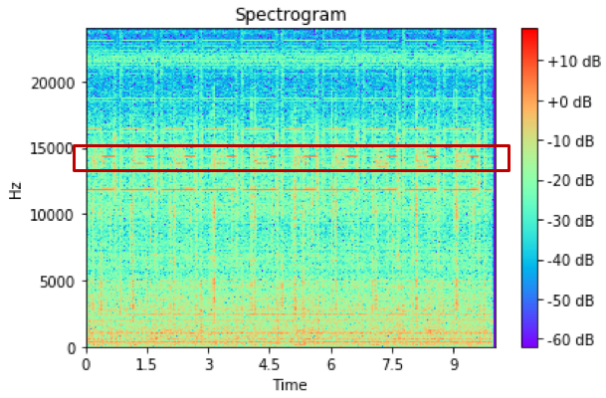


(e) Original instantaneous amplitude

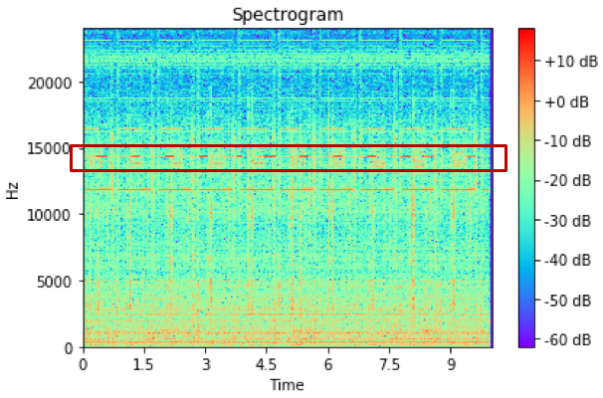


(f) Synthetic instantaneous amplitude

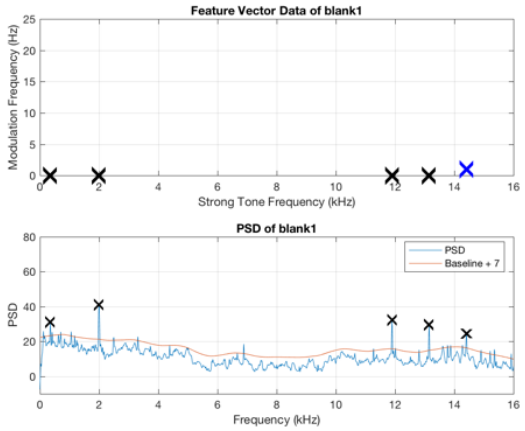
Figure 4.17. Example of frequency shifting of a single strong tone component.



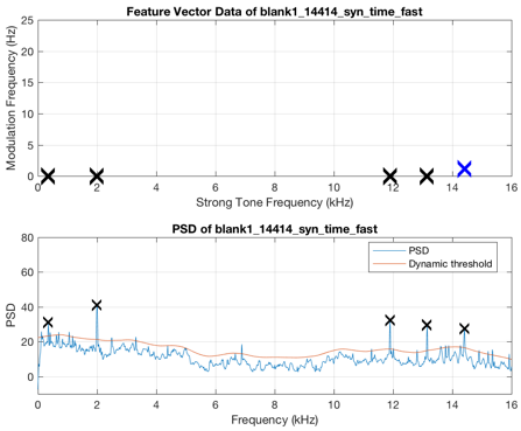
(a) Original spectrogram



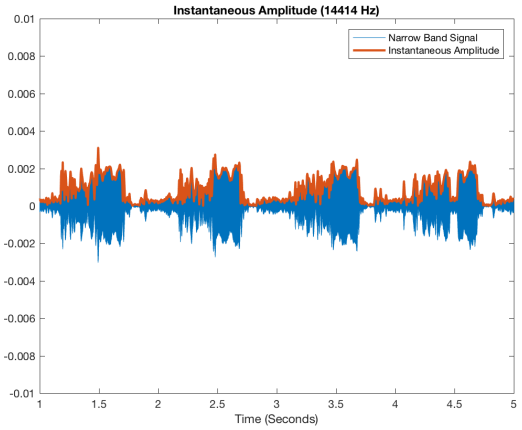
(b) Synthetic spectrogram



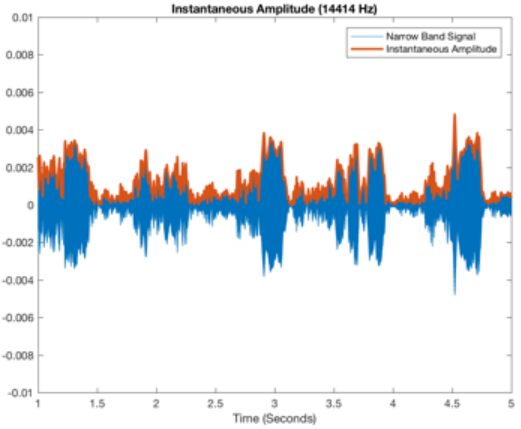
(c) Original detector result



(d) Synthetic detector result

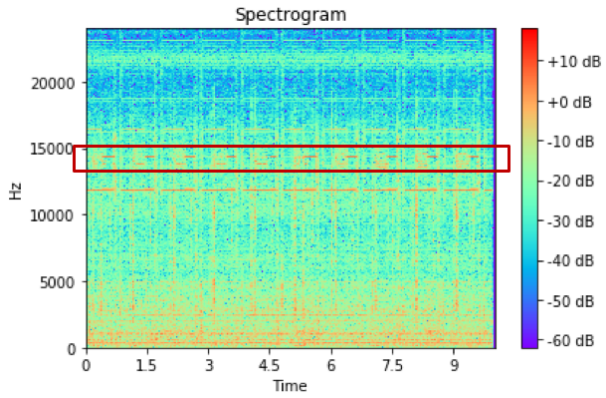


(e) Original instantaneous amplitude

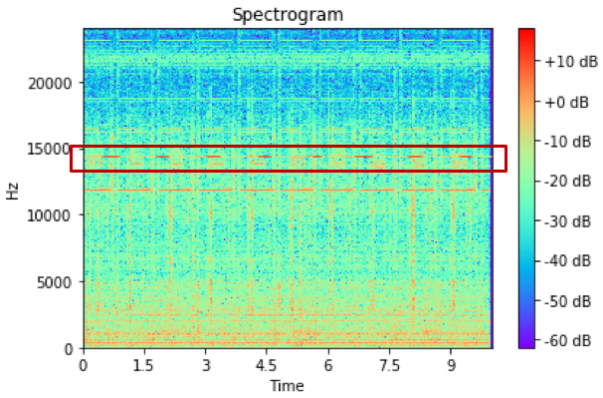


(f) Synthetic instantaneous amplitude

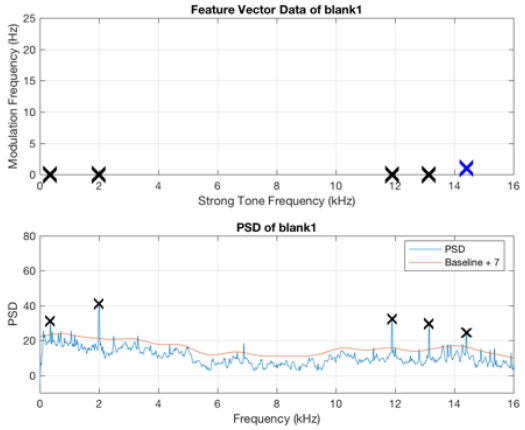
Figure 4.18. Example of speeding up a single strong tone component.



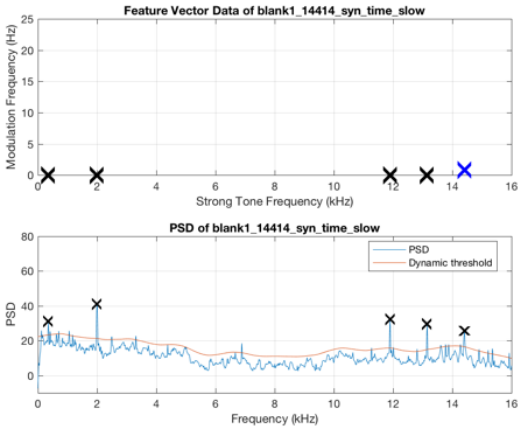
(a) Original spectrogram



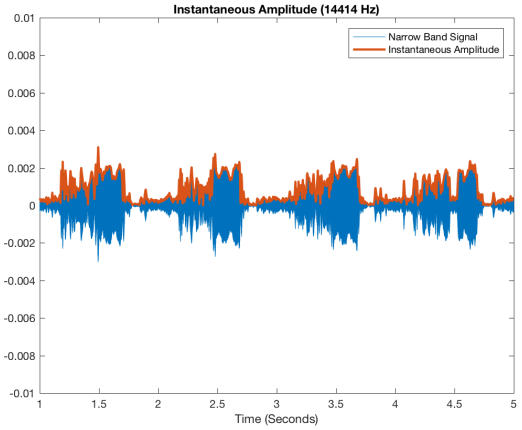
(b) Synthetic spectrogram



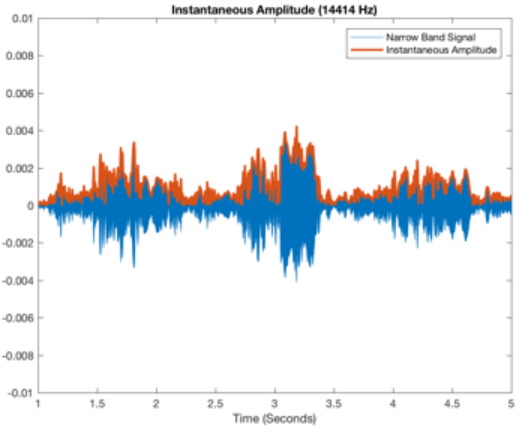
(c) Original detector result



(d) Synthetic detector result

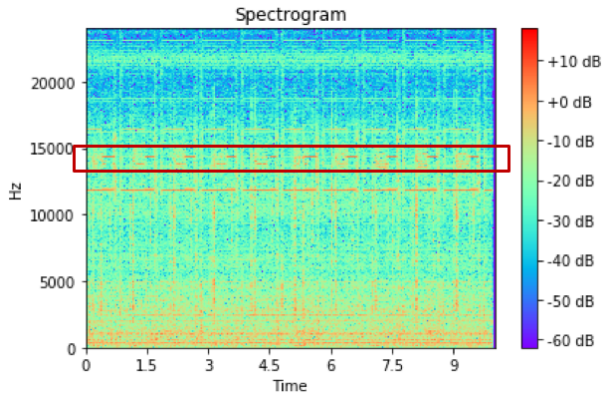


(e) Original instantaneous amplitude

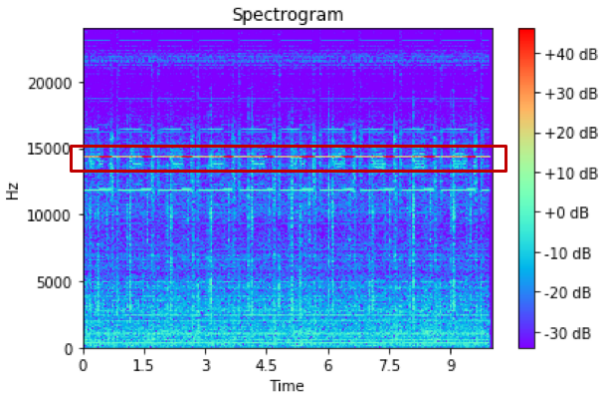


(f) Synthetic instantaneous amplitude

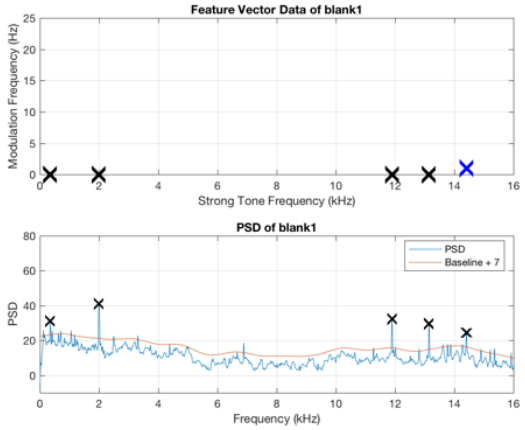
Figure 4.19. Example of slowing down a single strong tone component.



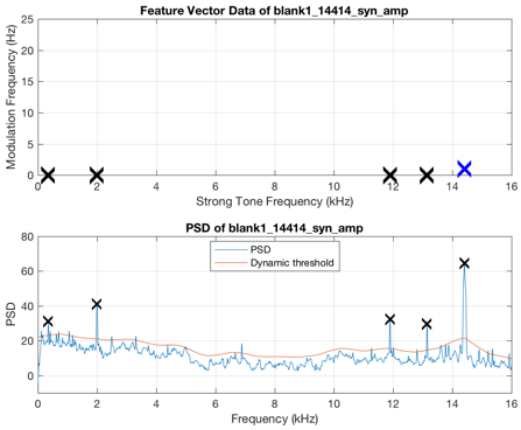
(a) Original spectrogram



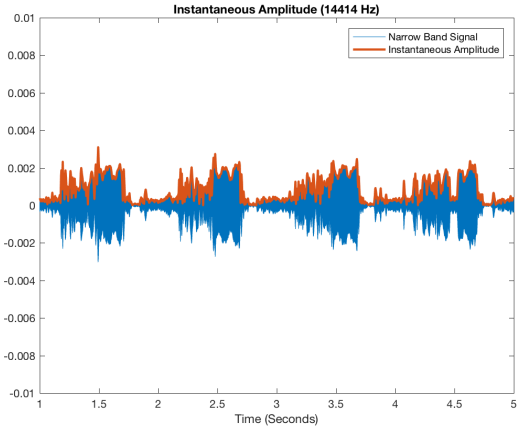
(b) Synthetic spectrogram



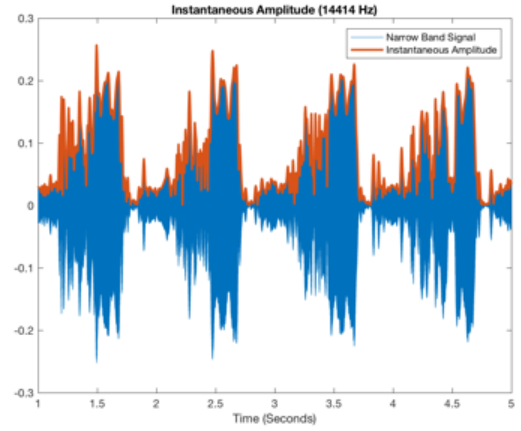
(c) Original detector result



(d) Synthetic detector result



(e) Original instantaneous amplitude



(f) Synthetic instantaneous amplitude

Figure 4.20. Example of scaling the amplitude of a single strong tone component.

4.3 Classification framework

In order to see how different anomaly models affect the classification results, we construct the framework shown in Fig. 6.8. We used two anomaly models. The first one is the mixing external dataset introduced in Section 4.2.2. We set the SNR to be 10 dB and randomly choose 8 samples from the real normal printer sounds mixed with 50 samples from the external dataset (DCASE2016). The 50 samples include 5 classes (door knocking, keys put on a table, keystrokes on a keyboard, pages being turned, drawers being opened) and 10 samples for each class. Therefore, we generate 400 synthetic abnormal sounds. The other anomaly model is our proposed synthetic method in Section 4.2.3. We randomly choose 80 samples from the real normal printer sounds. For each narrow-band signal related to a strong tone frequency, we synthesize one abnormal version. For frequency shifting, we randomly select the destination frequency in the range of $[6000, 9000]$ Hz, because no strong tone frequencies were detected in this range from our normal dataset. In speed aspect, we randomly choose speed up or slow down, but the variation is fixed to 20%. Lastly, in the amplitude aspect, we scale the magnitude of the DFT results by $10\times$.

During the preprocessing, we clip the sound from the second to the seventh second. Thus, the length of each sample is 5 seconds.

Because our goal in the current stage is to construct a baseline classification system, we adopt the method in [38], which is the baseline for environmental sound classification with the public dataset ESC. Firstly, we apply the short time Fourier transform to the signal and compute the power. Then, we apply a Mel filter bank [39] to each time frame. Lastly, we take the logarithm of the Mel power spectrogram and use the discrete cosine transform to extract the Mel frequency cepstral coefficients (MFCCs) [40] for each time frame. The features are the average and standard deviation of each MFCC over the time frames.

For classification, we use the supervised learning support vector machine (SVM) and the semi-supervised learning one-class support vector machine (OCSVM) [41]. SVM is a popular and robust learning method in supervised learning. SVM can use nonlinear function to map the data points to a higher dimensional space, and tries to find a hyperplane with maximal margin to separate the classes linearly. However, for applications to anomaly detection such

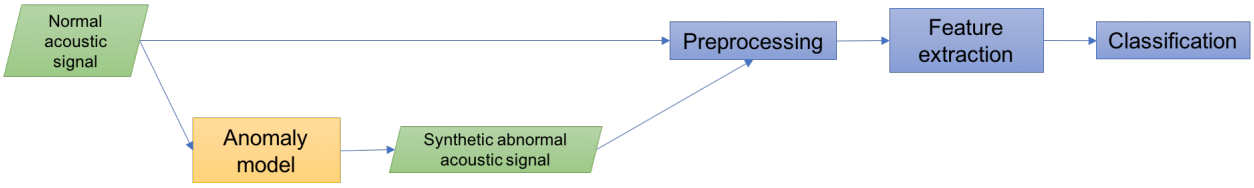


Figure 4.21. Framework for generating and processing synthetic abnormal acoustic signals..

as fraud detection and intrusion detection, the abnormal data is rare and has high variability. Therefore, we also consider this possibility in our task. In OCSVM, the algorithm tries to find a hyperplane with maximal margin between the data points and the origin. We will use the real normal printer sounds to train the model, and test the model with the normal and synthetic abnormal data.

Figure 4.22 shows the results of two anomaly models and two classifiers. Intuitively, we expect the overall performance of supervised SVM to be better than semi-supervised OCSVM on both datasets, because supervised learning has more information to learn the boundaries of the data. What we find is that supervised SVM performs better when the mixing external dataset is used as the abnormal dataset. However, semi-supervised OCSVM performs better on our second proposed synthetic abnormal dataset.

Therefore, we compute the standard deviation of each feature within the same dataset to determine the variations among the samples. From Fig. 4.23, we see our second proposed method has higher variations. In learning theory, if the training samples and the test samples are not from the same distribution, the model cannot capture the data representation well. Hence, that is why SVM performs better on the smaller variations dataset. The other analysis method that we consider is principal component analysis (PCA). We apply PCA to the features and choose two principal components for visualization as shown in Fig. 4.24. The blue dots are the normal data. The green dots are synthetic abnormal data generated by the mixing external dataset. The orange dots are synthetic abnormal data generated by our second proposed method. We can see that the green dot are very similar to the blue dots. That is, the abnormal data synthesized by the mixing external dataset is very similar to real normal sounds in the feature space. However, semi-supervised learning OCSVM is trying to find a behavior model of the normal data. It is going to distinguish the new incoming data that exhibits a very different behavior. Thus, that is why OCSVM performs better on our second proposed synthetic abnormal dataset.

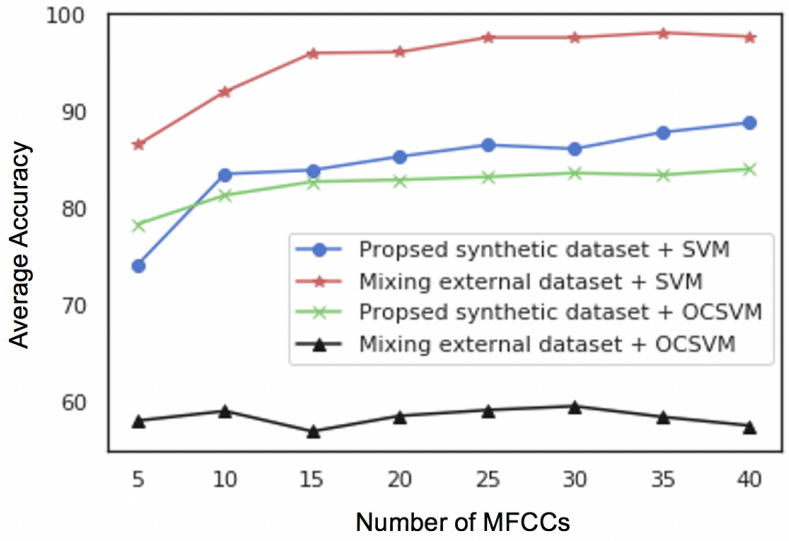


Figure 4.22. Classification results.

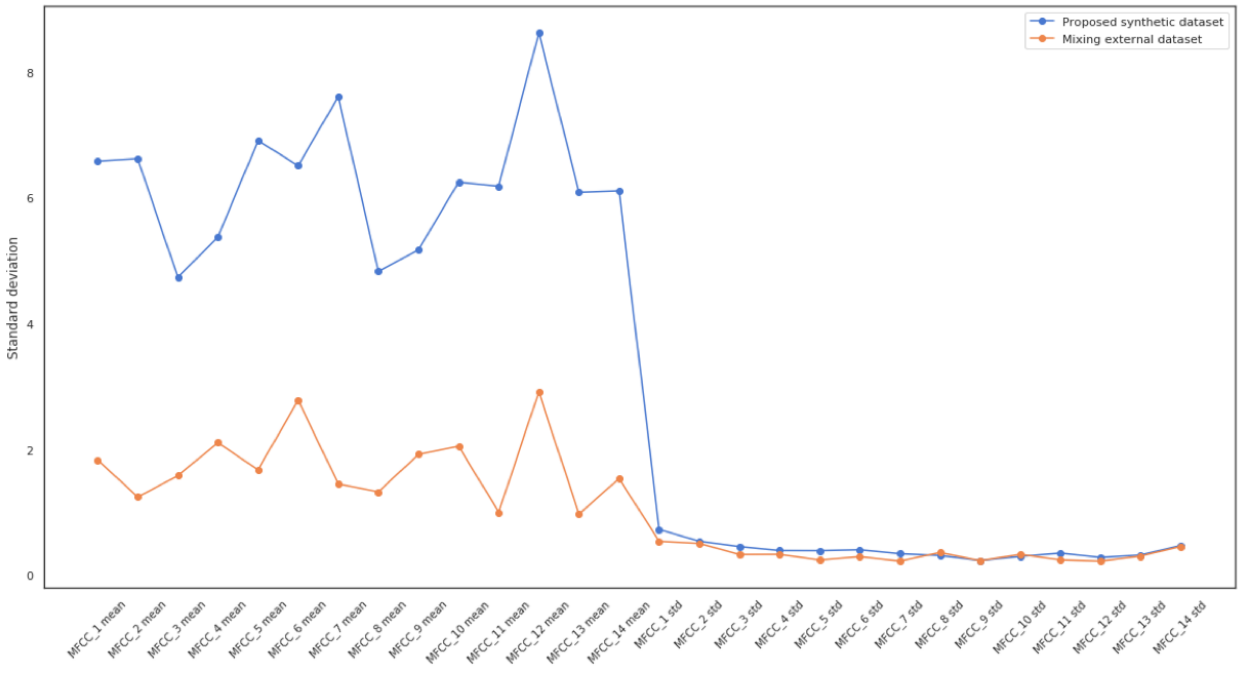


Figure 4.23. Standard deviation of each feature within the dataset.

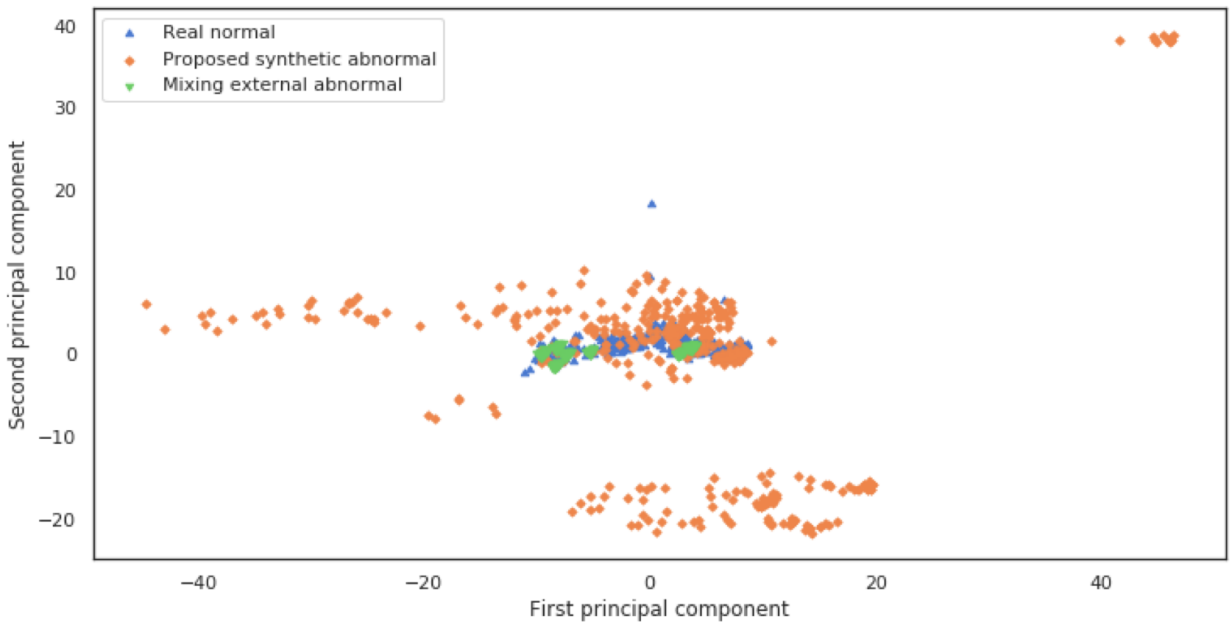


Figure 4.24. Principal component analysis of the features.

5. INCORPORATING A SIMILARITY METRIC IN A NEURAL MATRIX FACTORIZATION NETWORK

5.1 Introduction

There are many research works to deal with recommender systems such as [42] [43] [44]. However, they leverage additional information either from user side or item side. In some cases, we only have interaction, such as buying or watching, history in the dataset. Matrix factorization has proven its effectiveness in the Netflix Prize competition [45] which only uses watching history in development. It maps users and items to a joint latent factor space and models the user-item interactions as inner products in this space. A neural network extension of matrix factorization is proposed in [46], which utilizes a multi-layer perceptron to model more complex non-linear interactions between users and items. Moreover, the overall architecture is a fusion of linear and non-linear kernels shown in Fig. 5.1. The linear kernel is a general representation of conventional matrix factorization to model linear interactions. The non-linear kernel is implemented by fully-connected layers with non-linear activation functions to model complex non-linear interactions.

Here, we care more about whether the user is interested in the specific item than the accuracy of the rating scores. In addition, most of the data we can have is implicit real world data. Hence, the task can be formulated as a binary classification problem. The model predicts whether the user will be interested in the item. Cross-entropy is used as the loss function, as shown in Eq. (5.1). Here, m is the total number of samples, $\hat{y}_{ui}$ is the prediction, and y_{ui} is the ground truth.

$$L = -\frac{1}{m}\sum y_{ui}\log\hat{y}_{ui} + (1 - y_{ui})\log(1 - \hat{y}_{ui}) \tag{5.1}$$

5.2 Methodology

The motivation for incorporating a similarity metric in the optimization process is to make the model more controllable. In addition, this is the original idea behind matrix factorization. If two entities are more relevant, their features should be closer to each other.

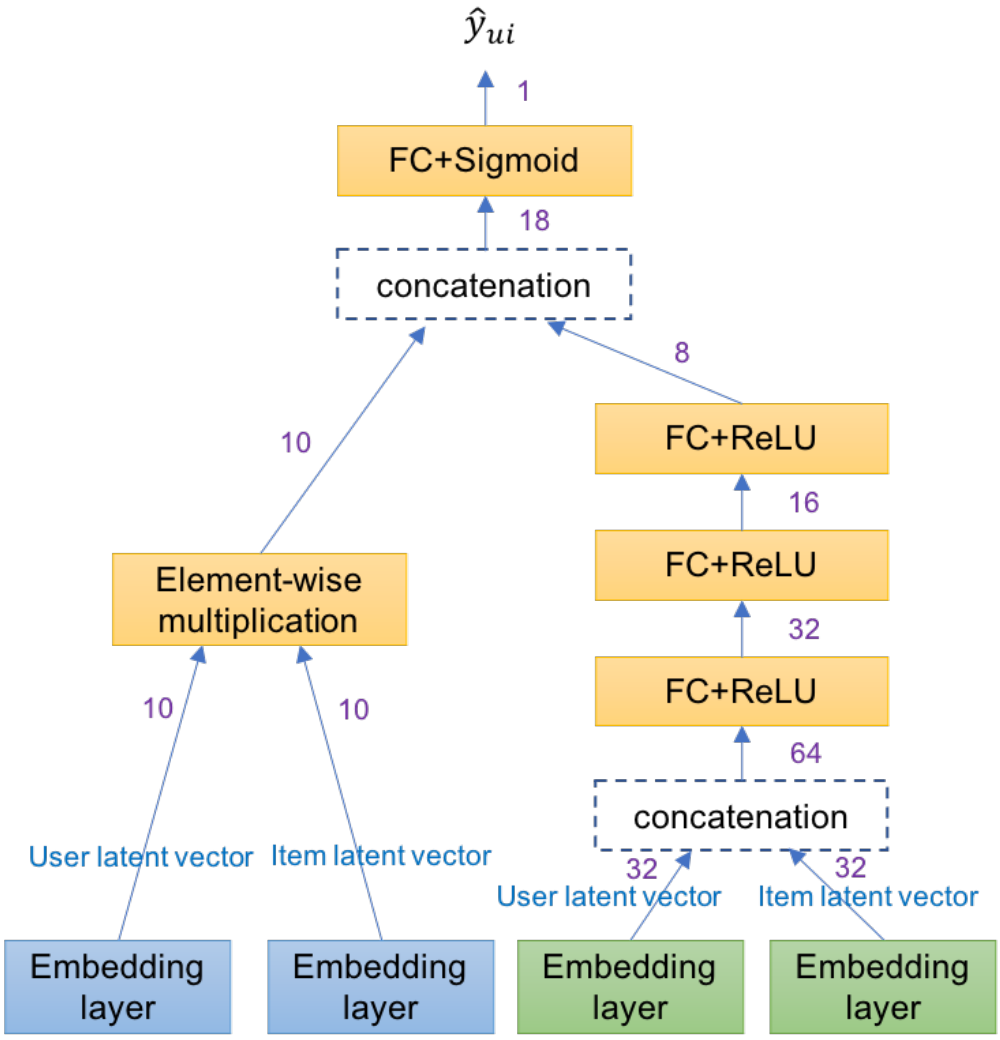


Figure 5.1. Neural matrix factorization.

We explore three ways including latent vectors of matrix factorization, latent vectors of multi-layer perceptron, and latent vectors of both. Cosine similarity is used to measure the similarity of two latent vectors. It is defined in Eq. (5.2). Here, x_u is the latent vector of user u and x_{i} is the latent vector of item i. Because the numerical value of cosine similarity is between -1 to 1, we transform the target to fit this range, as shown in Eq. (5.3). This is the ground truth of the similarity.

$$\mathrm{Sim}(u, \mathrm{i}) = \frac{x_u \cdot x_{\mathrm{i}}}{\|x_u\| \|x_{\mathrm{i}}\|} \tag{5.2}$$

$$\mathrm{Sim}_{ref} = y_{u\mathrm{i}} * 2 - 1 \tag{5.3}$$

5.2.1 Similarity metric with mean squared error

The loss functions can be extended as shown in Eqs. (5.4), (5.5), and (5.6) for latent vectors of matrix factorization, latent vectors of multi-layer perceptron, and latent vectors of both, respectively. $\mathrm{Sim}_{mf}(u, \mathrm{i})$ is a function of the latent vectors of matrix factorization. Similarly, $\mathrm{Sim}_{mlp}(u, \mathrm{i})$ is a function of the latent vectors of multi-layer perceptron.

$$\begin{aligned} L = &- \frac{1}{m} \sum y_{u\mathrm{i}} \log \hat{y}_{u\mathrm{i}} + (1 - y_{u\mathrm{i}}) \log(1 - \hat{y}_{u\mathrm{i}}) \\ &+ \frac{1}{m} \sum (\mathrm{Sim}_{ref} - \mathrm{Sim}_{mf}(u, \mathrm{i}))^2 \end{aligned} \tag{5.4}$$

$$\begin{aligned} L = &- \frac{1}{m} \sum y_{u\mathrm{i}} \log \hat{y}_{u\mathrm{i}} + (1 - y_{u\mathrm{i}}) \log(1 - \hat{y}_{u\mathrm{i}}) \\ &+ \frac{1}{m} \sum (\mathrm{Sim}_{ref} - \mathrm{Sim}_{mlp}(u, \mathrm{i}))^2 \end{aligned} \tag{5.5}$$

$$\begin{aligned} L = &- \frac{1}{m} \sum y_{u\mathrm{i}} \log \hat{y}_{u\mathrm{i}} + (1 - y_{u\mathrm{i}}) \log(1 - \hat{y}_{u\mathrm{i}}) \\ &+ \frac{1}{m} \sum (\mathrm{Sim}_{ref} - \mathrm{Sim}_{mf}(u, \mathrm{i}))^2 \\ &+ \frac{1}{m} \sum (\mathrm{Sim}_{ref} - \mathrm{Sim}_{mlp}(u, \mathrm{i}))^2 \end{aligned} \tag{5.6}$$

5.2.2 Similarity metric with mean absolute error

Furthermore, we explore the loss function incorporating similarity metric with mean absolute error. The loss functions are defined in Eqs. (5.7), (5.8), and (5.9) for latent vectors of matrix factorization, latent vectors of multi-layer perceptron, and latent vectors of both, respectively.

$$\begin{aligned} L = &- \frac{1}{m} \sum y_{ui} \log \hat{y}_{ui} + (1 - y_{ui}) \log(1 - \hat{y}_{ui}) \\ &+ \frac{1}{m} \sum |\text{Sim}_{ref} - \text{Sim}_{mf}(u, \text{i})| \end{aligned} \tag{5.7}$$

$$\begin{aligned} L = &- \frac{1}{m} \sum y_{ui} \log \hat{y}_{ui} + (1 - y_{ui}) \log(1 - \hat{y}_{ui}) \\ &+ \frac{1}{m} \sum |\text{Sim}_{ref} - \text{Sim}_{mlp}(u, \text{i})| \end{aligned} \tag{5.8}$$

$$\begin{aligned} L = &- \frac{1}{m} \sum y_{ui} \log \hat{y}_{ui} + (1 - y_{ui}) \log(1 - \hat{y}_{ui}) \\ &+ \frac{1}{m} \sum |\text{Sim}_{ref} - \text{Sim}_{mf}(u, \text{i})| \\ &+ \frac{1}{m} \sum |\text{Sim}_{ref} - \text{Sim}_{mlp}(u, \text{i})| \end{aligned} \tag{5.9}$$

5.3 Experiments

The architecture is shown in Fig. 5.1. It has a 3 layer multi-layer perceptron for the non-linear kernel. The numbers shown in purple are the dimensions of each layer. Hence, the dimension of the latent vectors of matrix factorization is 10 and the dimension of latent vectors of the multi-layer perceptron is 32.

In these experiments, we use the MovieLens 1M [47] dataset which has 6,040 users, 3,706 items, and 1,000,209 ratings. Moreover, it has at least 20 ratings for each user. Because we are only interested in whether the user showed some interest in the item, we treat all the ratings as positive samples and all unobserved interactions as negative samples. In order to predict the 'future', we draw out the latest interaction of each user as the test set and use the remaining positive samples for training. We follow the training and evaluation methods in [46]. For the training, we randomly sample four negative instances for each positive instance. During the evaluation, it is time-consuming to rank all items for every user. So, we randomly

sample 99 negative instances for each positive instance. The detailed numbers are shown in Table 5.1 and Table 5.2.

Hit rate and NDCG (Normalized Discounted Cumulative Gain) are used to evaluate the performance of recommender systems in this work. If the true interest item shows up in Top-N recommendation, we called 'hit'. On the other hand, NDCG is used to evaluate whether the true interest item has the higher rank. If the system ranks the item higher, it has higher score. The DCG (Discounted Cumulative Gain) is defined by Eq. (5.10). And IDCG (Idealized Discounted Cumulative Gain) is computed by the same formula, but it assumes that the interest items are ordered by decreasing relevance. So, we can have $\mathrm{NDCG} = \frac{\mathrm{DCG}}{\mathrm{IDCG}}$.

$$\mathrm{DCG} = r_1 + \sum_{\mathrm{i}=2}^{N} \frac{r_{\mathrm{i}}}{\log_2(\mathrm{i})} \tag{5.10}$$

The result is shown in Table 5.3. Incorporating similarity metric on latent vectors of multi-layer perceptron with mean squared error has the best result.

Table 5.1. Training set.

Training set	
number of positive	994,169
number of negative	3,976,676
total	4,970,845

Table 5.2. Test set.

Test set	
number of positive	6,040
number of negative	597,960
total	604,000

Table 5.3. Comparison result of different loss functions.

	Hit Rate@10	NDCG@10
Baseline	0.660	0.384
MF (MSE)	0.655	0.381
MLP (MSE)	0.674	0.395
MF+MLP (MSE)	0.670	0.396
MF (MAE)	0.653	0.374
MLP (MAE)	0.661	0.384
MF+MLP (MAE)	0.641	0.371

6. SEMANTIC UNDERSTANDING OF IMAGES CONTAINING FASHION ITEMS

6.1 Introduction

Garment semantic understanding as shown in Fig. 6.1 is a crucial topic in fashion product shopping on e-commerce websites. It is essential on a wide variety of applications, such as recommender systems, image retrieval, and image caption. Semantic understanding provides the information to cluster the products with similar features, then we can utilize it to recommend the new products to the customers. Another example is that the customers can provide the image of fashion product from their fashion icons, then the semantic information help us to do visual search and retrieve the products that the customers are looking for. Furthermore, we can apply the semantic understanding to image tagging which can further extend to product title generation or auto-correction.

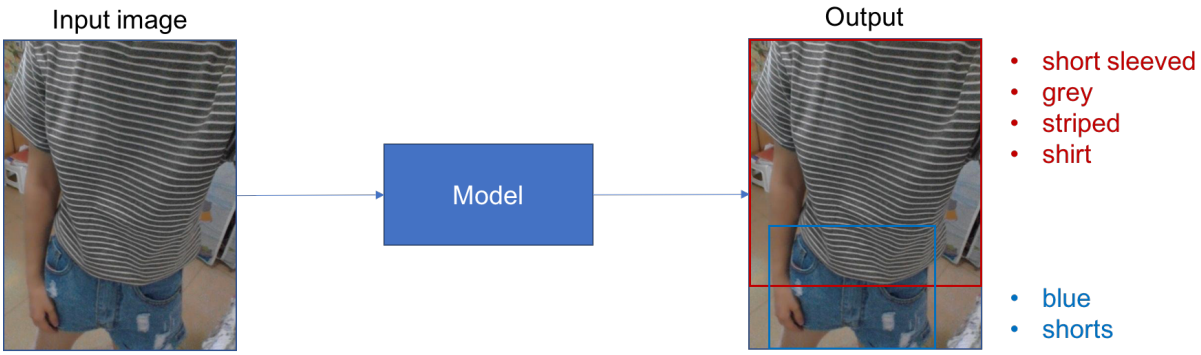


Figure 6.1. An example of fashion items semantic understanding. The model detects the clothes and outputs the corresponding attributes.

Supervised deep learning has proven its effectiveness on a wide range of fields such as computer vision and natural language processing. However, this data-driven method relies on a large amount of labeled data. In practical, it is hard to find a dataset with rich annotations include many attributes of clothing items. For example, sleeve length, color, pattern, and neck type. One possible way is to combine the datasets which are collected from different domains and with different annotations. Therefore, here comes the challenges. First is the domain gap between the datasets. Even these deep neural networks have high learning

capacity, they are still suffering from poor transferability. [48] shows an experiment. They trained a convolutional neural network on SVHN (The Street View House Numbers) dataset which achieves 98% accuracy. However, the model performed poorly on MNIST handwritten digits dataset. The accuracy is only 67.1%, despite in general MNIST dataset is easier task. This demonstrates the challenge of domain gap. Figure 6.2 shows the image examples of the two datasets.

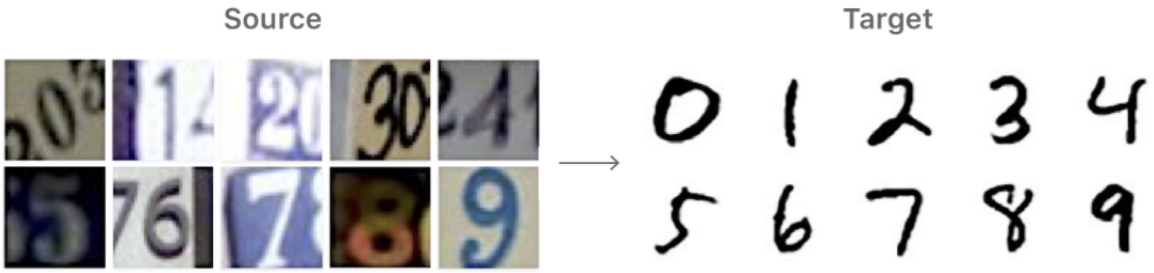


Figure 6.2. An illustration of domain gap: the source images are from SVHN and the target images are from MNIST. [49]

In this work we utilized two datasets: DeepFashion2 [50] and Kaggle fashion product images [51]. Let us take a glance of these two datasets which are shown in Fig. 6.3 and we will give more details in next section. The images in DeepFashion2 are scene photos which are taken by professional photography or selfie. They can be taken from many different angles with cluttered background. The annotations include bounding boxes and segmentation masks for detection task. Moreover, DeepFashion2 also provide garment attributes such as sleeve length. On the other hand, Kaggle fashion product images are catalog photos with simple background and taken from frontal view. It provides pattern attribute annotations. Hence, in our work, the problem we want to deal with is how do we bridge the datasets from different domain and different annotations.

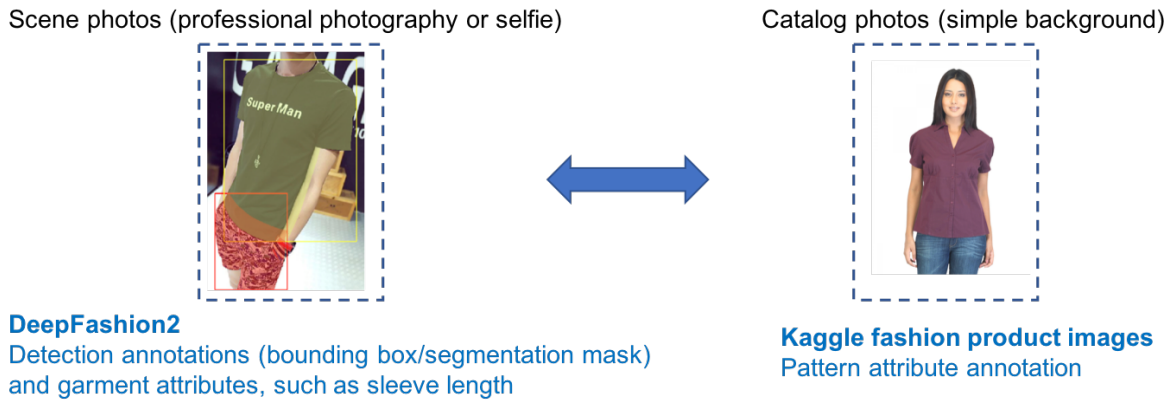


Figure 6.3. Two datasets from different domain and different annotations. (DeepFashion2 and Kaggle fashion product images)

6.2 Preliminary

6.2.1 Datasets

DeepFashion2 is a versatile fashion dataset released in 2019 for detection, segmentation, clothing image retrieval, etc. It provides 13 fine-grained categories for clothing items as shown in Fig. 6.4 and their corresponding bounding boxes and segmentation masks. Figure 6.5 is an example of bounding box annotations. One image can contain multiple objects, that is, multiple clothing items. The images in DeepFashion2 show different variations such as scale, occlusion level, and view points. On the other hand, Kaggle fashion product images are catalog images with white or clean background. The images not only include garment but also shoes, watches, belts, etc. In this work we only focus on garment. The dataset originally provides annotations comprise category, gender, season, and product description. However, we are interested in other garment attributes. We further mined the data from the provided meta-data and created a dataset for pattern annotations. There are 4 patterns: Printed, Solid, Striped, and Checked. The sample images are shown in Fig. 6.6.

6.2.2 Weak supervision

State-of-the-art models in a form of conventional supervision rely on massive sets of hand-labeled training data which are expensive and time-consuming. It is not practical in real-

Figure 6.4. DeepFashion2: fine-grained categories. (Two categories with red text have fewer samples)

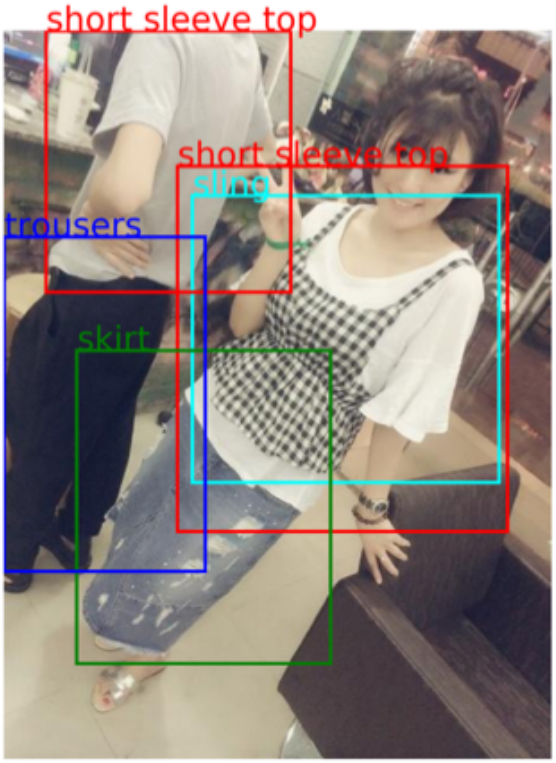


Figure 6.5. DeepFashion2: bounding box example.

Figure 6.6. Kaggle fashion product images: pattern attribute.

world applications. Recently, weak supervision aroused the practitioners' interests. Usually it starts from a problem: how to label the unlabeled dataset? Weak supervision is weaker forms of supervision leveraging higher-level or noisier input. The labels/annotations can be imprecise, inexact, or inaccurate. [52] proposed Snorkel, a system to support programming training data. The pipeline asks users to define labeling functions which could come from external domain knowledge, rule, or classifiers. Obviously, these labels can be noisy and conflict. Then Snorkel uses a generative model to learn the weighting of each labeling function and output a probabilistic training labels. This implies the model gives more weights to the labels generated by more accurate labeling functions. The final discriminative model is trained on this set of probabilistic training labels. Because the ground truth is generated by the program, this target model is trained in a form of weak supervision. Figure 6.7 illustrates the pipeline.

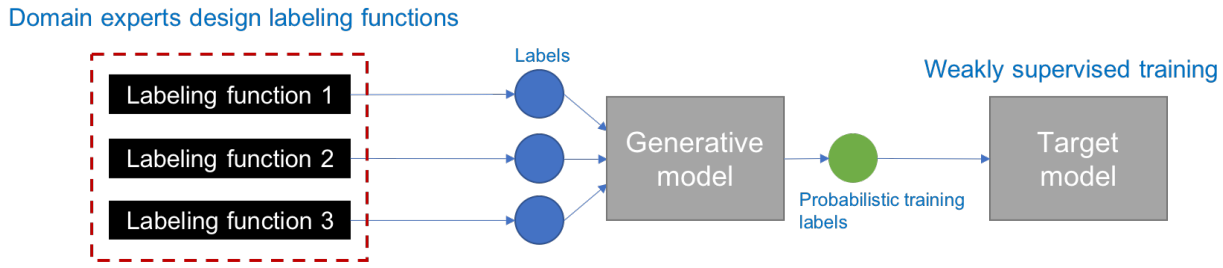


Figure 6.7. Weak supervision example: Snorkel's pipeline.

6.3 Overview of proposed weakly supervised framework

To solve domain gap and fuse annotations, we proposed a weakly supervised framework as illustrated in Fig. 6.8. The key idea is that we want to create a model by one dataset and use it to label the other dataset. Then we can create a new dataset with rich annotations. The framework consists of three steps. Firstly, we use attention-based transfer learning to train a pattern prediction model for clothing items. Then, we use this model to label DeepFashion2 dataset for pattern attribute. Now we have a new dataset with bounding boxes, fine-grained categories, and patterns annotations. Lastly, we can use this new dataset to train our target model which is able to perform clothing item semantic understanding.

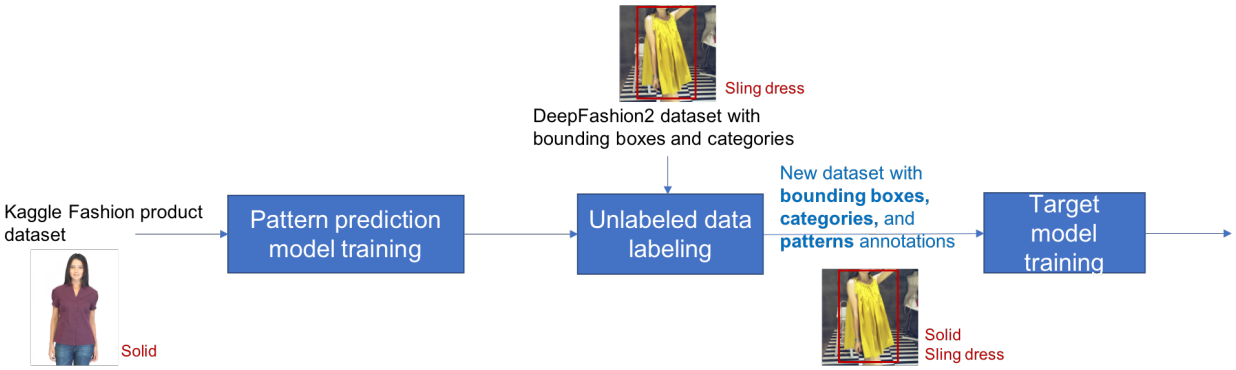


Figure 6.8. Proposed weakly supervised framework.

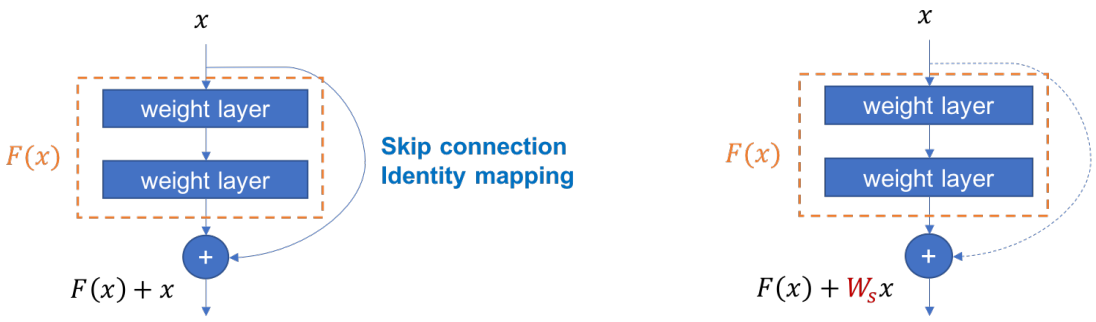
6.4 Baseline and Related work

6.4.1 Introduction of backbone network: ResNet

Residual neural network (ResNet) [53] is a breakthrough of very deep neural networks. Before residual neural network, the classic neural network simply cascade convolutional layers. When the number of layers exceeds to a certain number, the performance becomes worse. The author argues that the deeper network should not perform worse than shallow network. The residual block uses skip connection from input to the output of convolutional layers as shown in Fig. 6.9. It gives an opportunity to the neural network to learn identity mapping. That is, it should be equivalent to shallow network. If the dimension of input is the same as the dimension of output of convolutional layers, residual block just simply add them together as the final output. In contrast, the convolutional layers sometimes shrink spatial size to extend receptive field and increase the channel size for more features of each spatial position. This causes the input and output of convolutional layers have different dimensions. Then, we need to apply linear projection to fit the dimension between them. In practice, the linear projection is implemented by 1×1 convolution.

6.4.2 Baseline

Snorkel [52] uses user-designed labeling functions and gathers ensemble result to create a new dataset. Our baseline is using a convolutional neural network to label the dataset. The first step of the process is training a CNN model (ResNet18) on Kaggle fahsion product



(a) Input and output have the same dimension.

(b) Input and output have different dimension.

Figure 6.9. Residual block.

images for pattern prediction task. Then, we use this model to make predictions on DeepFashion2 dataset. These predicted labels provide new annotations to DeepFashion2 dataset. The architecture of ResNet18 is shown in Fig. 6.10. We apply the same backbone, ResNet18, to all the approaches in this work to compare them fairly.

6.4.3 Learn to pay attention

Attention plays an important role in human visual perception. Recently in many research works, attention concept also shows significant influence for natural language understanding in artificial intelligence such as transformer architecture [54]. A related work about attention we use to compare with our method is Learn to Pay Attention [55]. This paper is inspired by the success of attention mechanism in natural language processing (NLP). In NLP applications there is a query feature vector, and computing the correlation between the query feature vector and input context provides the information that which part in the context the model should pay more attention to. Learn to Pay Attention uses global features which are from last layer to serve as query feature vector in attention mechanism. And the context is intermediate representation. The idea of this paper is to compute attention by 'compatibility' between intermediate representations and global feature vector.

The compatibility score is given by Eq. (6.1). g is a global feature vector which is a summarized result from last layer. For backbone ResNet18, the dimension of last layer is $512 \times 7 \times 7$, if the dimension representation is (channel size, height, width). Then, this

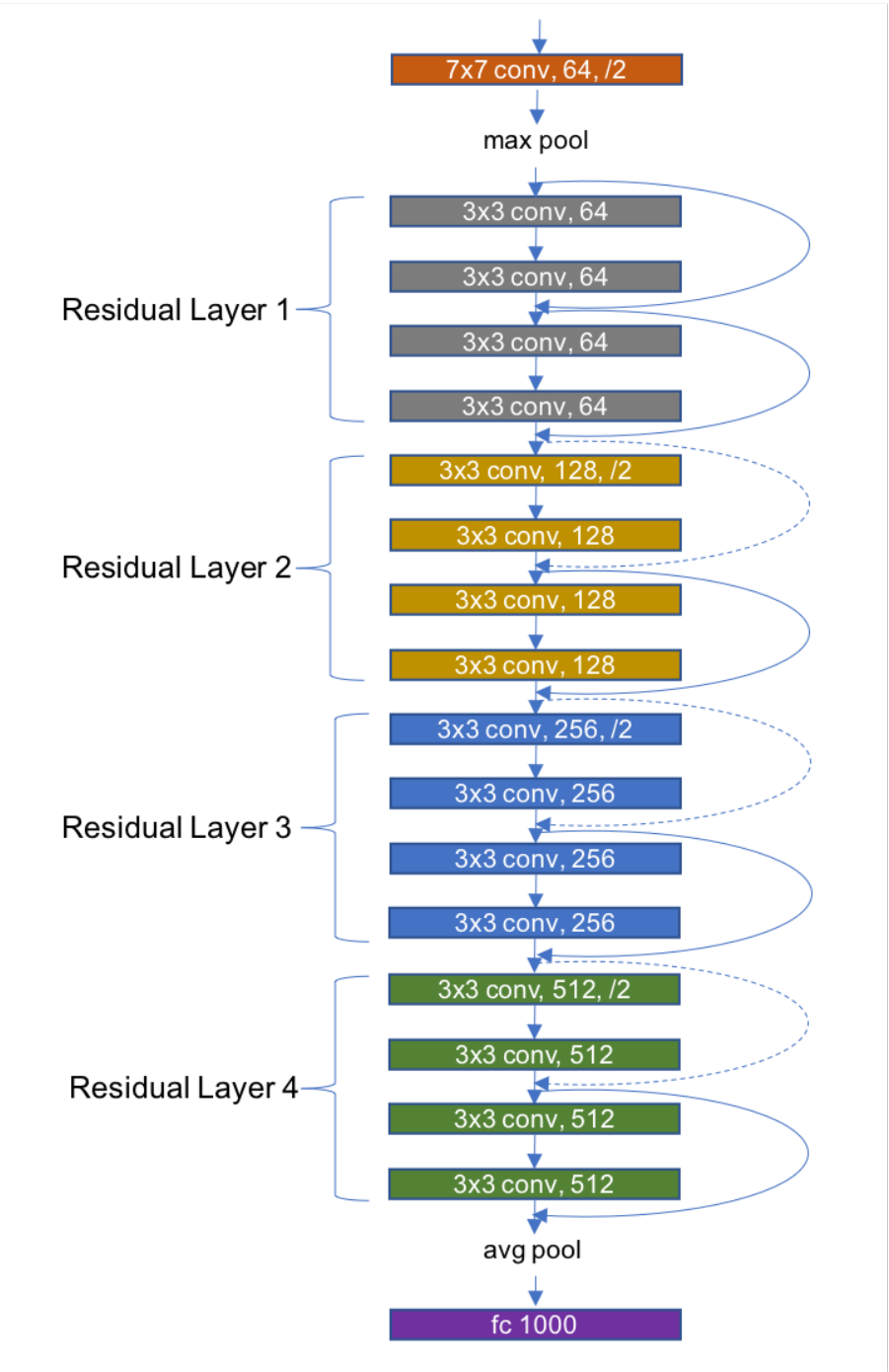


Figure 6.10. Architecture of ResNet18. The solid lines represent skip connections when the input and output of the convolutional layers have the same dimension size. The dash lines represent skip connections when the input and output of the convolutional layers have different dimension size and a linear projection is involved.

method applies a convolutional layer and a max pooling to compress this representation to a 512-dimension feature vector. Here, l_{i} is a local feature vector where i indicates the spatial position. And u is a learnable vector. The compatibility score is the inner product of the learnable vector and element-wise sum of the local feature vector and the global feature vector. Softmax function is applied to the compatibility score in order to let all attention values sum to 1. Attention value is give by Eq. (6.2). Then, the final feature vector is a weighted average of local features as shown in Eq. (6.3). If we also consider the earlier layer to make final prediction, the channel size (number of features of each spatial position) usually is less than the channel size of last layer. This method applies 1×1 convolutional layer to reduce the dimension of the global feature vector and concatenate the results from layers to a fully connected layer. The final prediction is from a weighted average of local features by attention values. The original average pooling and fully connected layer in backbone are eliminated. Figure 6.11 illustrates our implementation of Learn to Pay Attention.

$$c_{\mathrm{i}} = \langle u, l_{\mathrm{i}} + g \rangle \tag{6.1}$$

$$a_{\mathrm{i}} = \mathrm{softmax}(c_{\mathrm{i}}) \tag{6.2}$$

$$g_a = \sum_{\mathrm{i}=1}^{n} a_{\mathrm{i}} l_{\mathrm{i}} \tag{6.3}$$

6.5 Methodology

6.5.1 Attention-based transfer learning

Teacher-student learning is one of the transfer learning approaches. They can be applied for domain adaptation [56]. For example, the face recognition from studio photography and surveillance camera. The images are from different domains but with the same annotations. In addition, transfer learning is able to improve the performance of simple network by transferring domain knowledge from complex teacher network. For example, [57] proposed a method of transferring attention by properly defining attention from intermediate layers of

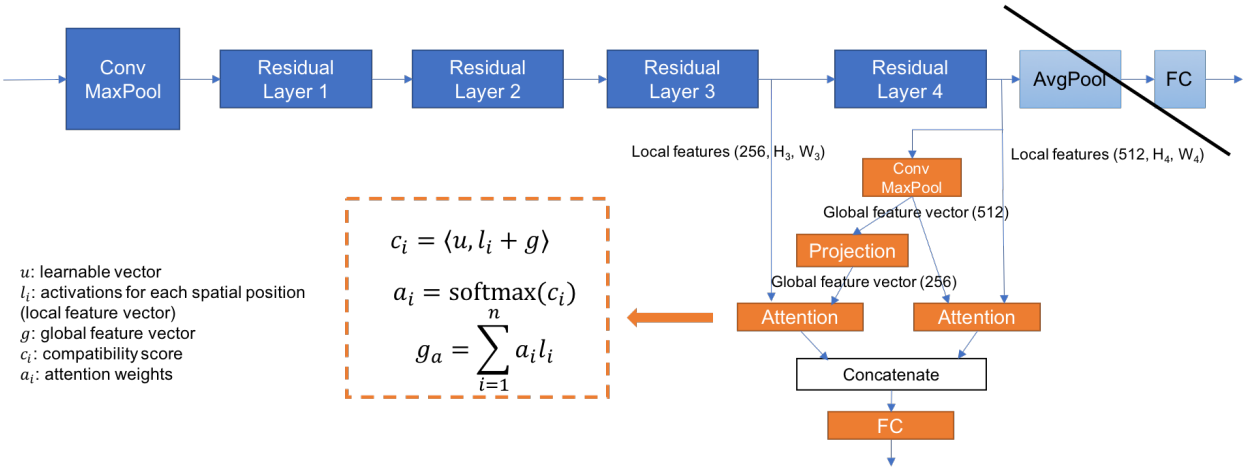


Figure 6.11. Implementation of Learn to Pay Attention.

convolutional neural network. And force a student network to mimic the attention maps of a powerful teacher network in order to improve the performance of a student network. The advantage is to use a simpler network through the knowledge transferring from a more complex network to achieve better performance.

Attention maps knowledge transferring inspires us to leverage it to mitigate the domain gap between two datasets with different annotations. In common notion, DeepFashion2 is a harder dataset. The images have cluttered background. Moreover, some of them are taken from weird angles. The deformation could make the same clothing category look very different.

There are two stages in our attention-base transfer learning. The first stage is the teacher network training. We train a teacher network to perform fine-grained categories classification on DeepFashion2 dataset. In general, a deep convolutional neural network consists of several convolutional layers for feature extraction. In addition, there is an average pooling and fully connected layer that summarizes the features and makes final prediction as illustrated in Fig. 6.12.

In the second stage, we assume the teacher network can provide the guidance to that part in spatial position to which we should pay more attention. Therefore, we freeze the teacher network and train student network on Kaggle fashion product images for patterns classifica-

tion task. We do not care the output of teacher network. Instead, we use the feature map of intermediate layers as shown in Fig. 6.13. There are two learning goals for student network training. One is the patterns classification task, the other is we want the attention map generated by student network is close to the attention map generated by teacher network. If the dimension representation is (channels, height, width), the dimensions of feature map in the first convolutional layer, residual layer 2, and residual layer 4 for ResNet18 are (64, 56, 56), (128, 28, 28), and (512, 7, 7) respectively. In order to fit attention maps between teacher network and student network, the vectorized attention map is computed by Eq. (6.4). j is the layer index, S represents student network, and T is teacher network. F is attention map function which sums the activation values cross the channels. That is, $a_{x,y} = \sum_{c=1}^{c=C} a_{c,x,y}$, $\mathcal{R}^{C\times H\times W} \rightarrow \mathcal{R}^{H\times W}$. vec() means converting 2D matrix to 1D vector.

$$\begin{aligned} Q_S^{\mathrm{j}} &= \mathrm{vec}(F(A_S^{\mathrm{j}})) \\ Q_T^{\mathrm{j}} &= \mathrm{vec}(F(A_T^{\mathrm{j}})) \end{aligned} \tag{6.4}$$

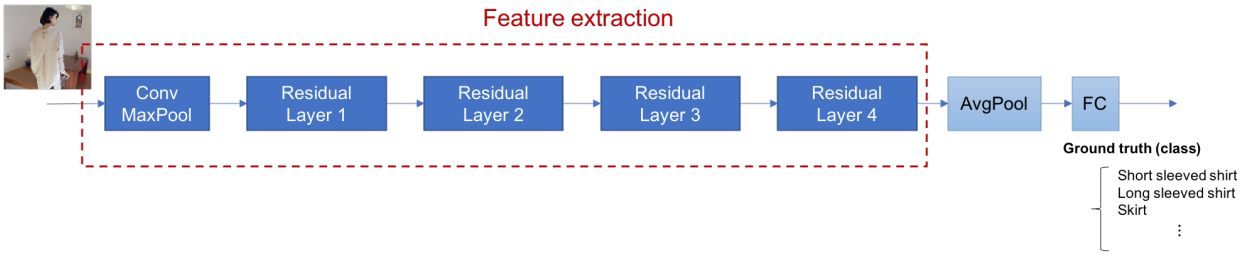


Figure 6.12. Attention-based transfer learning - stage 1: Teacher network training on DeepFashion2 for fine-grained categories classification.

The loss function to optimize the student network in stage 2 are given by Eq. (6.5) and Eq. (6.6). Eq. (6.5) is the cross entropy for classification loss. Here, M is number of classes, $y^{(\mathrm{i})}$ is the ground truth, and $o^{(\mathrm{i})}$ is the output activation value. The second term in Eq. (6.6) is the attention loss which is used to minimize the distance between the teacher attention map and the student attention map.

$$L_{CE} = -\sum_{\mathrm{i}=1}^{M} y^{(\mathrm{i})} \log(\mathrm{softmax}(o^{(\mathrm{i})})) \tag{6.5}$$

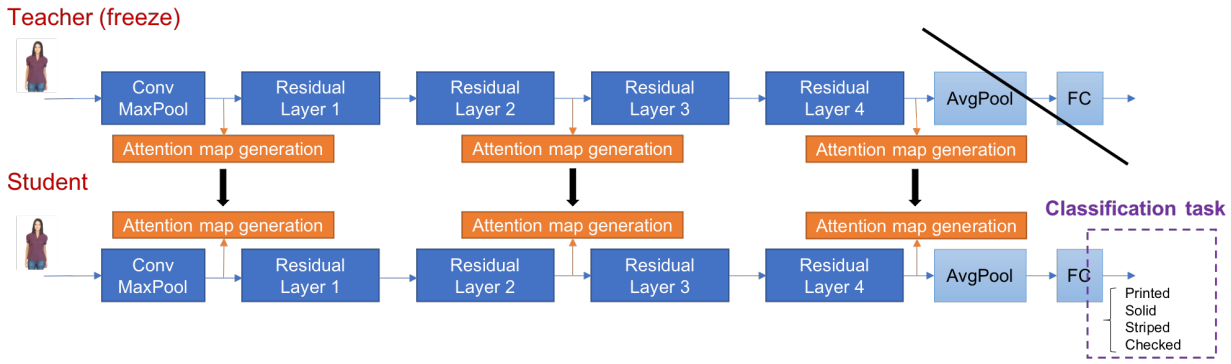


Figure 6.13. Attention-based transfer learning - stage 2: Student network training by transfer learning on Kaggle fashion product images for patterns classification.

$$L = L_{CE} + \frac{\beta}{2} \sum_{j \in I} \left\| \frac{Q_S^j}{\|Q_S^j\|_2} - \frac{Q_T^j}{\|Q_T^j\|_2} \right\| \tag{6.6}$$

6.5.2 Mask-guided teacher network training

In order to improve the performance of teacher network, we further leverage the annotations of segmentation masks provided by DeepFashion2. We proposed mask-guided teacher network training as shown in Fig. 6.14. We add additional block to convert feature map to attention mask and use the segmentation mask as ground truth. Thus, the loss function will have an additional term which is responsible to fit this attention map.

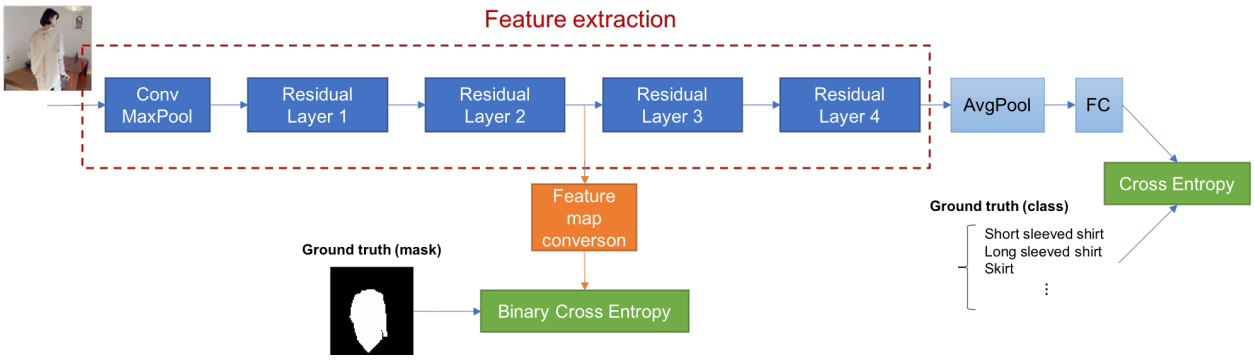


Figure 6.14. Attention-based transfer learning - stage 1: Mask-guided teacher network training on DeepFashion2 for fine-grained categories classification.

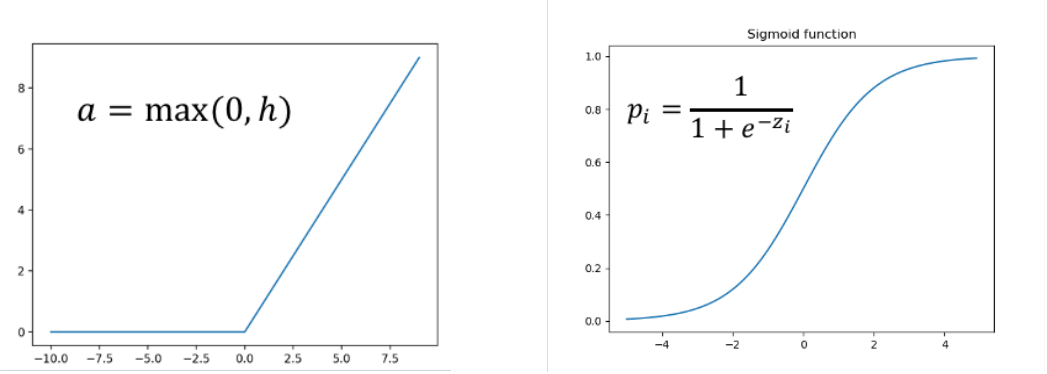
Feature maps are from intermediate layers of neural network, and they are three dimensional tensors which can be interpreted as several features (channels) for each spatial position. The ground truth segmentation mask is either 0 or 1 to represent background or object. Hence, we need to convert feature maps to fit the ground truth. To be specific, the conversion result is the probability of the object for each spatial position. Firstly, we want to compress features to a single value for each spatial position. We apply max function to the activation values cross the channels as Eq. (6.7), where $a_{c,x,y}$ is the activation value of feature map. The idea is to suppress the activation values which belong to background. Second, we apply batch normalization which is given by Eq. (6.8). Because the non-linear activation function of convolutional layer for ResNet18 is ReLU, the output values span the entire range of the positive numbers as shown in Fig. 6.15(a). We want to re-center the values to zero for next step. Then, we apply Sigmoid function which is given by Eq. (6.9) and the output value of each spatial position is interpreted as the probability of object. The additional loss for mask-guided teacher network training is simply binary cross entropy as shown in Eq. (6.10). The total loss is cross entropy for fine-grained categories classification task plus the mask binray cross entropy. The process is illustrated in Fig. 6.16.

$$a_{x,y} = \max_{1 \le c \le C} a_{c,x,y} \tag{6.7}$$

$$z = \frac{a - E[a]}{\sqrt{Var[a] + \epsilon}} \cdot \gamma + \beta \tag{6.8}$$

$$p_{\mathrm{i}} = \frac{1}{1 + \mathrm{e}^{-z_{\mathrm{i}}}} \tag{6.9}$$

$$L_{mask} = \frac{1}{N} \sum_{\mathrm{i}=1}^{N} y_{\mathrm{i}} \cdot \log(p_{\mathrm{i}}) + (1 - y_{\mathrm{i}}) \cdot \log(1 - p_{\mathrm{i}}) \tag{6.10}$$



(a) ReLU. (b) Sigmoid.

Figure 6.15. Activation functions.

6.6 Experiments and results

Our work is implemented by Python and PyTorch. The experiments are conducted on the Purdue Gilbreth cluster that consists of Dell compute nodes with Intel Xeon processors and Nvidia Tesla GPUs (P100 or V100).

6.6.1 Notation for semantic understanding

In our experiments, we have 13 categories include short sleeved shirt, long sleeved shirt, vest, sling, short sleeved outwear, long sleeved outwear, shorts, trousers, skirt, short sleeved dress, long sleeved dress, vest dress, and sling dress provided by DeepFashion2. And 4 patterns: printed, solid, striped, and checked. Our target model perform the classification task for 52 (13×4=52) classes. For example, solid skirt and printed short sleeved shirt.

6.6.2 Pattern prediction model training and evaluation

The first part of our proposed framework is to train a pattern prediction model by attention-based transfer learning in order to label unlabeled dataset. In our experiment, the unlabeled dataset is DeepFashion2, because originally it does not provide pattern annotations. Hence, we hand-labeled the ground truth for subset of DeepFashion2 to see how well our work performs. We trained this network on Kaggle fashion product images. The training set includes 10225 images. Each image with a label of one of four patterns: printed, solid,

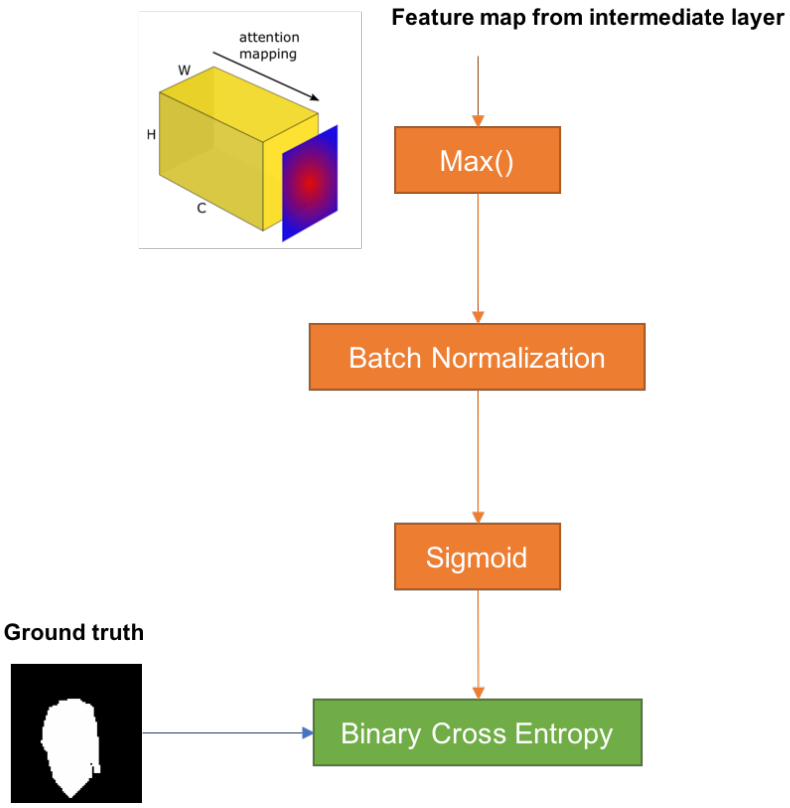


Figure 6.16. Feature map conversion block.

striped, and checked. Then, we evaluated the model on two test datasets: Kaggle fashion product images (2555 images) and DeepFashion2 (3300 images). Because DeepFashion2 has bounding box annotations. We also experimented another setup; using this annotation to crop the original image as pre-processing. The comparison result is shown in Table 6.1. The solo attention mechanism: Learn to Pay Attention and data augmentation are able to improve the classification accuracy of the Kaggle dataset which is the dataset the model is trained on. However, the good performance can not be transferred to another dataset well. They are not able to improve the accuracy on unseen dataset DeepFashion2. On the other hand, our proposed attention-based transfer learning improves both datasets. And incorporating mask-guided teacher network training has the best performance among all the approaches. It is also noteworthy that our proposed method mitigates the performance gap between original images and cropped images. The difference of accuracy between original

images and cropped images is reduced from 5.75% to 2.21%. It means the model has higher ability to focus on our interested objects.

6.6.3 Semantic understanding model training and evaluation

In the second part of our proposed framework, we use trained pattern prediction model to label DeepFashion2 training set includes 16500 images. By this way, we obtain a new dataset with rich annotations contain bounding boxes, segmentation masks, fine-grained categories, and patterns. We simply extend MaskRCNN [58] to 52 classes (13 fine-grained categories $\times$ 4 patterns) as our target model: semantic understanding model for clothing items. The evaluation test set is 3300 images from DeepFashion2 validation subset. The new added pattern annotations are obtained by hand labeled. We generate new training datasets by pattern prediction models trained by different approaches mentioned in last section. And use the same neural network (MaskRCNN) to train on those different training datasets. This experiment is to evaluate which generated dataset can provide better performance for our target model training.

We follow the standard detection performance metrics in COCO dataset [59]. In detection or segmentation, a prediction is considered to be True Positive if Intersection-over-Union (IoU) is larger than threshold. IoU is computed as the area of intersection of the model's output with the ground truth, divided by the area of their union as shown in Fig. 6.17. Average precision (AP) is the average precision values over different classes and different thresholds. The precision is defined in Eq. (6.11).

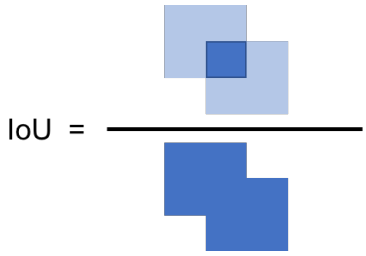


Figure 6.17. Intersection-over-Union.

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6.11}$$

In Table 6.2 and Table 6.3 we compare all the methods with three AP values using different IoU thresholds. $\text{AP}^{@0.50:0.95}$ is also called mean AP which is computed by averaging 10 precision values over 52 classes from threshold 0.5 to threshold 0.95 with step size 0.05. $\text{AP}^{@0.5}$ and $\text{AP}^{@0.75}$ are the average precision over 52 classes at IoU threshold 0.5 and 0.75 respectively. Detection result has the same phenomenon with pattern prediction evaluation. The solo attention mechanism: Learn to Pay Attention and data augmentation perform worse than baseline. On the other hand, our proposed attention transfer learning has the best performance among the methods. It implies that a new training dataset generated by better pattern prediction model achieves better performance, because the newly generated training dataset has better data quality.

Table 6.1. Comparison result of pattern prediction on two datasets.
[1] [55] Learn to pay attention ($H_4 \times W_4$, $H_3 \times W_3$). Dimensions of layer 4 and layer 3.
[2] Proposed method.

	Kaggle	DeepFashion2	DeepFashion2 (crop)
Baseline	86.26%	66.55%	72.30%
Attention[1] (28×28, 56×56)	89.00%	64.97%	71.15%
Attention[1] (7×7, 14×14)	88.29%	65.91%	68.27%
Data augmentation	89.27%	66.76%	67.27%
Attention transfer learning[2]	90.99%	70.45%	73.58%
Attention transfer learning[2] (Mask-guided)	90.45%	72.94%	75.15%

Table 6.2. Comparison result of clothing items semantic understanding: bounding box average precision.

	$AP^{@0.50:0.95}$	$AP^{@0.50}$	$AP^{@0.75}$
Baseline	32.7	39.8	38.0
Baseline (crop)	36.9	45.2	43.1
Attention (28×28, 56×56)	31.6	38.7	36.8
Attention (7×7, 14×14)	29.4	35.9	34.3
Data augmentation	26.0	31.9	30.7
Attention transfer learning	40.4	48.3	46.9
Attention transfer learning (Mask-guided)	42.1	51.2	48.8

Table 6.3. Comparison result of clothing items semantic understanding: segmentation mask average precision.

	$AP^{@0.50:0.95}$	$AP^{@0.50}$	$AP^{@0.75}$
Baseline	33.2	39.7	38.3
Baseline (crop)	37.7	45.1	43.0
Attention (28×28, 56×56)	32.4	38.6	37.3
Attention (7×7, 14×14)	30.0	35.9	34.3
Data augmentation	26.4	31.9	30.8
Attention transfer learning	40.6	48.2	46.1
Attention transfer learning (Mask-guided)	42.6	51.1	48.9

7. CONCLUSIONS

In this dissertation, we investigated problems in printing quality and online-shopping. To improve printing quality we proposed solutions from two aspects: halftoning algorithm and printer self-diagnosis. For online-shopping, we first explored different loss functions in recommender systems. Then, we deal with the problem of semantic image understanding of fashion items.

In Chapter 2, we developed a novel hybrid halftoning screen design method with subpixels modeling for unequal resolution. The design was implemented into a real printer in the market. This low cost laser printer produces high quality documentation at efficient speeds from the Printerland review. In Chapter 3, we developed a new cost-function-based repetitive bands interval estimation algorithm. Adding synthetic missing bands in exhaustive search achieves better performance than the previous method. Accurate estimation helps to diagnose root cause of failing mechanical component in the printer. In Chapter 4, we investigated the printer self-diagnosis problem with acoustic data. We proposed three flexible methods to synthesize abnormal data. Then, we can increase the size of dataset to help the classifier be more robust.

In Chapter 5, we explored how the similarity metric in the loss function affects the performance of a neural matrix factorization network. In Chapter 6, we proposed a weakly supervised framework to create a new dataset with rich annotations for our semantic image understanding model training. The new dataset is generated by proposed mask-guided teacher network training in attention-based transfer learning, which is effective in knowledge transfer between two datasets from different domain and with different annotations.

To summarize, the major contributions of this thesis are listed as follows:

1. Halftoning algorithm development with unequal resolution
    - Develop S shape cores. It has better visual results than the traditional square cores on our product
    - Improve the design process which is applying the same macrocell index array to each stage. It shows the better result as well.

- Develop design rule to require compact and symmetric dot clusters.
- Design was implemented in a low cost monochrome laser printer.

2. Periodic banding analysis

   - Build a classification model by logistic regression to determine whether the potential defects are visible or invisible. The average result achieves 93.4% accuracy.
   - Proposed a new cost-function-based repetitive interval estimation method to improve the accuracy on noisy and corrupted test sample pages.

3. Acoustic signal augmentation

   - Proposed a flexible anomaly model to synthesize abnormal data.
   - Built a framework to analyze the relationship between anomaly models and classifiers.

4. Recommender systems

   - Explored the similarity metrics in the loss function for a neural matrix factorization network.

5. Semantic image understanding of fashion items

   - Proposed a weakly supervised framework to mitigate the domain gap in datasets and acquired a new dataset with rich annotations
   - Proposed a mask-guided teacher network training in attention-based transfer learning that improved the performance of semantic image understanding for fashion items

REFERENCES

[1] W. Jang and J. P. Allebach, "Simulation of print quality defects," *Journal of Imaging Science and Technology*, vol. 49, no. 1, pp. 1–18, 2005.

[2] R. Kumontoy, K. Low, M. Ortiz, C. Kim, P. Choe, S. Leman, K. Oldenburger, M. Lehto, X. Lehto, H. Santos-Villalobos, H. Park, and J. Allebach, "Web-based diagnosis tool for customers to self-solve print quality issues," *Journal of Imaging Science and Technology*, vol. 54, no. 4, pp. 40 503–1, 2010.

[3] J. Zhang and J. P. Allebach, "Estimation of repetitive interval of periodic bands in laser electrophotographic printer output," in *Image Quality and System Performance XII, SPIE*, vol. 9396, 2015.

[4] Q. Lin and J. P. Allebach, "Color FM screen design using DBS algorithm," *Proc. SPIE, Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts III*, vol. 3300, pp. 353–361, 1998. DOI: 10.1117/12.298298.

[5] G. Lin and J. P. Allebach, "Generating stochastic dispersed and periodic clustered textures using a composite hybrid screen," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3746–3758, Dec. 2006, ISSN: 1057-7149. DOI: 10.1109/TIP.2006.881968.

[6] P. Li and J. P. Allebach, "Tone-dependent error diffusion," *IEEE Transactions on Image Processing*, vol. 13, no. 2, pp. 201–215, Feb. 2004, ISSN: 1057-7149. DOI: 10.1109/TIP.2003.819232.

[7] T. N. Pappas, J. P. Allebach, and D. L. Neuhoff, "Model-based digital halftoning," *IEEE Signal Processing Magazine*, vol. 20, no. 4, pp. 14–27, Jul. 2003, ISSN: 1053-5888. DOI: 10.1109/MSP.2003.1215228.

[8] C. Lee and J. P. Allebach, "The hybrid screen improving the breed," *IEEE Transactions on Image Processing*, vol. 19, no. 2, pp. 435–450, Feb. 2010, ISSN: 1057-7149. DOI: 10.1109/TIP.2009.2032941.

[9] G. Sharma, *Digital Color Imaging Handbook*. Boca Raton, FL, USA: CRC Press, Inc., 2002, ISBN: 084930900X.

[10] F. A. Baqai and J. P. Allebach, "Computer-aided design of clustered-dot color screens based on a human visual system model," *Proceedings of the IEEE*, vol. 90, no. 1, pp. 104–122, Jan. 2002, ISSN: 0018-9219. DOI: 10.1109/5.982409.

[11] T. M. Holladay, "An optimum algorithm for halftone generation for displays and hard copies," *Proc. Society Information Display*, vol. 21, pp. 185–192, 1980.

[12] G.-Y. Lin and J. P. Allebach, "Multilevel screen design using direct binary search," *Journal of the Optical Society of America A*, vol. 19, no. 10, pp. 1969–1982, Oct. 2002.

[13] D. Kacker, T. Camis, and J. P. Allebach, "Electrophotographic process embedded in direct binary search," *IEEE Transactions on Image Processing*, vol. 11, no. 3, pp. 243–257, Mar. 2002, ISSN: 1057-7149. DOI: 10.1109/83.988958.

[14] S. Hu, H. Nachlieli, D. Shaked, S. Shiffman, and J. P. Allebach, "Color-dependent banding characterization and simulation on natural document images," in *Color Imaging XVII: Displaying, Processing, Hardcopy, and Applications*, International Society for Optics and Photonics, vol. 8292, 2012, 82920W.

[15] X. Jing, H. Nachlieli, D. Shaked, S. Shiffman, and J. P. Allebach, "Masking mediated print defect visibility predictor," in *Image Quality and System Performance IX*, International Society for Optics and Photonics, vol. 8293, 2012, 82930R.

[16] A. H. Eid, M. N. Ahmed, B. E. Cooper, and E. E. Rippetoe, "Characterization of electrophotographic print artifacts: Banding, jitter, and ghosting," *IEEE Transactions on Image Processing*, vol. 20, no. 5, pp. 1313–1326, 2011.

[17] R. Rasmussen, E. N. Dalal, and K. Hoffman, "Measurement of macro-uniformity: Streaks, bands, mottle and chromatic variations," in *PICS*, 2001, pp. 90–95.

[18] D. R. Rasmussen, K. D. Donohue, Y. S. Ng, W. C. Kress, F. Gaykema, and S. Zoltner, "Iso 19751 macro-uniformity," in *Image Quality and System Performance III*, International Society for Optics and Photonics, vol. 6059, 2006, 60590K.

[19] D. R. Rasmussen, "Tent-pole spatial defect pooling for prediction of subjective quality assessment of streaks and bands in color printing," *Journal of Electronic Imaging*, vol. 19, no. 1, p. 011 017, 2010.

[20] J. Zhang, S. Astling, R. Jessome, E. Maggard, T. Nelson, M. Shaw, and J. P. Allebach, "Assessment of presence of isolated periodic and aperiodic bands in laser electrophotographic printer output," in *Image Quality and System Performance X, SPIE*, vol. 8653, 2013.

[21] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning From Data*. New York, USA: AMLBook, 2012.

[22] J. Zhang, H. Nachlieli, D. Shaked, S. Shiffman, and J. P. Allebach, "Psychophysical evaluation of banding visibility in the presence of print content," in *Image Quality and System Performance IX, SPIE*, vol. 8293, 2012.

[23] X. Wang, "Harmonic scrubber for detected modulation frequencies," M.S. thesis, Purdue University, West Lafayette, Indiana, 2019.

[24] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[25] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[26] J. Billa, "Improving LSTM-CTC based ASR performance in domains with limited training data," *arXiv preprint arXiv:1707.00722*, 2017.

[27] N. Jaitly and G. E. Hinton, "Vocal tract length perturbation (VTLP) improves speech recognition," in *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*, vol. 117, 2013.

[28] N. Takahashi, M. Gygli, B. Pfister, and L. Van Gool, "Deep convolutional neural networks and data augmentation for acoustic event detection," *arXiv preprint arXiv:1604.07160*, 2016.

[29] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 6440–6444.

[30] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, IEEE, 2015, pp. 1–6.

[31] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *Advances in Neural Information Processing Systems*, 2016, pp. 892–900.

[32] Y. Tokozume, Y. Ushiku, and T. Harada, "Learning from between-class examples for deep sound recognition," *arXiv preprint arXiv:1711.10282*, 2017.

[33] M. Dolson, "The phase vocoder: A tutorial," *Computer Music Journal*, vol. 10, no. 4, pp. 14–27, 1986.

[34] J. Laroche and M. Dolson, "New phase-vocoder techniques for pitch-shifting, harmonizing and other exotic effects," in *Proceedings of the 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 1999, pp. 91–94.

[35] U. Madhow, *Introduction to Communication Systems*. Cambridge University Press, 2014.

[36] Y. Koizumi, S. Saito, H. Uematsu, Y. Kawachi, and N. Harada, "Unsupervised detection of anomalous sound based on deep learning and the Neyman–Pearson lemma," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 1, pp. 212–224, 2019.

[37] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *24th European Signal Processing Conference (EUSIPCO)*, 2016, pp. 1128–1132.

[38] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *Proceedings of the 23rd ACM International Conference on Multimedia*, 2015, pp. 1015–1018.

[39] M. Sahidullah and G. Saha, "Design, analysis and experimental evaluation of block based transformation in mfcc computation for speaker recognition," *Speech Communication*, vol. 54, no. 4, pp. 543–565, 2012.

[40] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[41] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[42] S. Rendle, "Factorization machines," in *2010 IEEE International Conference on Data Mining*, IEEE, 2010, pp. 995–1000.

[43] Y. Juan, Y. Zhuang, W.-S. Chin, and C.-J. Lin, "Field-aware factorization machines for ctr prediction," in *Proceedings of the 10th ACM Conference on Recommender Systems*, ACM, 2016, pp. 43–50.

[44] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, "Xdeepfm: Combining explicit and implicit feature interactions for recommender systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ACM, 2018, pp. 1754–1763.

[45] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, no. 8, pp. 30–37, 2009.

[46] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*, International World Wide Web Conferences Steering Committee, 2017, pp. 173–182.

[47] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems (TIIS)*, vol. 5, no. 4, pp. 1–19, 2015.

[48] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht, "Sliced Wasserstein discrepancy for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 285–10 295.

[49] *Bridging the Domain Gap for Neural Models*, 2019 June. [Online]. Available: https://machinelearning.apple.com/research/bridging-the-domain-gap-for-neural-models.

[50] Y. Ge, R. Zhang, X. Wang, X. Tang, and P. Luo, "Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5337–5345.

[51] 2018. [Online]. Available: https://www.kaggle.com/paramaggarwal/fashion-product-images-dataset.

[52] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré, "Snorkel: Rapid training data creation with weak supervision," in *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, NIH Public Access, vol. 11, 2017, p. 269.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[54] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017.

[55] S. Jetley, N. Lord, N. Lee, and P. Torr, "Learn to pay attention," International Conference on Learning Representations, 2018.

[56] V. Manohar, P. Ghahremani, D. Povey, and S. Khudanpur, "A teacher-student learning approach for unsupervised domain adaptation of sequence-trained asr models," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2018, pp. 250–257.

[57] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," International Conference on Learning Representations, 2017.

[58] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969.

[59] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *European Conference on Computer Vision*, Springer, 2014, pp. 740–755.

VITA

Wan-Eih Huang received her BS in Engineering and System Science and MS in Electrical Engineering from National Tsing Hua University, Taiwan, in 2008 and 2010 respectively. She is currently pursuing a Ph.D. degree, and working on image processing, image analysis, data science, and machine learning in the School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, USA. Her research interest includes image processing, computer vision, and deep learning.