IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 16, 2020, accepted December 14, 2020, date of publication December 24, 2020,
date of current version January 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3047074

Multispectral Image Reconstruction From Color Images Using Enhanced Variational Autoencoder and Generative Adversarial Network

XU LIU[1], ABDELOUAHED GHERBI[1], (Member, IEEE), ZHENZHOU WEI[2], WUBIN LI[3], AND MOHAMED CHERIET[1], (Senior Member, IEEE)

[1]Synchromedia Laboratory, École de Technologie Supérieure (ÉTS), University of Québec, Montreal, QC H3C 1K3, Canada
[2]Department of Electrical and Computer Engineering, McGill University, Montreal, QC H3A 0E9, Canada
[3]Ericsson Research, Montreal, QC H4S 0B6, Canada

Corresponding author: Xu Liu (xu.liu.1@ens.etsmtl.ca)

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), and in part by the Canada Research Chair in Sustainable Smart Eco-Cloud.

ABSTRACT Since multispectral images (MSIs) have much more sufficient spectral information than RGB images (RGBs), reconstructing MS images from RGB images is a severely underconstrained problem. We have to generate colossally different information between the two scopes. Almost all previous approaches are based on static and dependent neural networks, which fail to explain how to supplement the massive lost information. This paper presents a low-cost and high-efficiency approach, "VAE-GAN", based on stochastic neural networks to directly reconstruct high-quality MSIs from RGBs. Our approach combines the advantages of the Generative Adversarial Network (GAN) and the Variational Autoencoder (VAE). The VAE undertakes the generation of the lost variational MS distributions by reparameterizing the latent space vector with sampling from Gaussian distribution. The GAN is responsible for regulating the generator to produce MSI-like images. In this way, our approach can create huge missed information and make the outputs look real, which also solves the previous problem. Moreover, we use several qualitative and quantitative methods to evaluate our approach and obtain excellent results. In particular, with much less training data than the previous approaches, we obtained comparable results on the CAVE dataset and surpassed state-of-the-art results on the ICVL dataset.

INDEX TERMS Generative adversarial network (GAN), variational autoencoder (VAE), VAE-GAN, normal distribution, stochastic neural network, multispectral image, RGB image, image processing, color vision, spectral reconstruction.

I. INTRODUCTION

Lights with different wavelengths have different reflection, refraction, and transmission properties. People use Multi-Spectral Images (MSIs) to record these differences. MSIs consist of several channels or bands, and each band contains the amount of radiation measured in a particular wavelength range [1].

From MSIs' abundant spectral information, many MSIs' applications have been developed, such as land mine detection [2], satellite remote sensing [3], medical imaging [4], weather forecasting [5], and interpretation of ancient documents and artworks [6].

Although MSIs have a wide range of applications, acquiring them is a complicated, costly, and time-consuming process, since MSIs have many bands that have to be taken one by one. Moreover, obtaining each band's data requires a specific wavelength lens to filter out other wavelength lights and to be stored in a dedicated space. Therefore, much time and storage space are consumed in changing the lens and saving each band's image. Moreover, we can synthesize RGBs with high precision from MSIs according to the Color Matching Functions [7]. From MSIs to RGBs, it is a straightforward process.

On the contrary, RGB images only have three bands: red, green, and blue. The three colors are used as the primary colors to constitute other colors (different wavelength light). Most of our daily used devices' cameras can take RGB

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

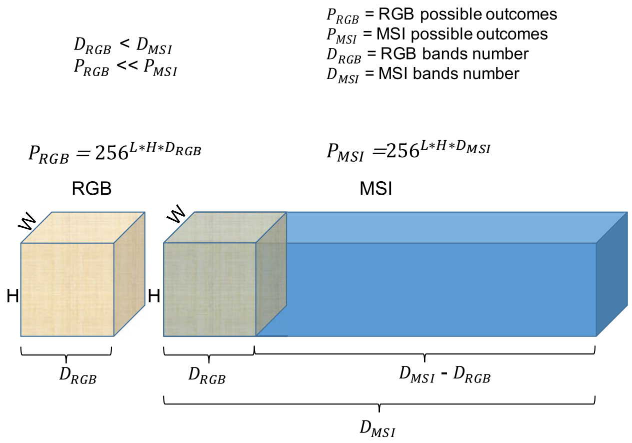
$D_{RGB} < D_{MSI}$
$P_{RGB} << P_{MSI}$
P_{RGB} = RGB possible outcomes
P_{MSI} = MSI possible outcomes
D_{RGB} = RGB bands number
D_{MSI} = MSI bands number
$P_{RGB} = 256^{L*H*D_{RGB}}$
$P_{MSI} = 256^{L*H*D_{MSI}}$
RGB
MSI
W
H
W
H
D_{RGB}
D_{RGB}
D_{MSI} - D_{RGB}
D_{MSI}

FIGURE 1. The schematic diagram of possible spectral space.

images, which are much more convenient and cheaper to obtain. Hence, it is straightforward to consider reconstructing MSIs from RGBs directly. However, the reconstruction process is very complex and challenging.

In order to illustrate this challenge, Figure 1 is the schematic diagram that presents a preliminary idea of the challenge. Supposing we have a 3-bands 512 × 512 RGB image and a 31-bands 512 × 512 MS image, the range of each pixel value is from 0 to 255. According to information theory, the maximum information contained by an RGB image is $-\log 256^{512*512*3}$ (nats), while the maximum information of an MS image is $-\log 256^{512*512*31}$ (nats). There is a vacancy of more than 10 times between the two spaces, which turns the reconstruction of MS images from RGB images into an extremely underconstrained problem. Compared with RGB images, MS images have a much higher spectral resolution, which may cause a difficult problem of one RGB image mapping to many MS images [8].

To tackle this problem, we propose a new approach in this paper, whose fundamental concept is to replace the traditional autoencoder with the VAE when implementing the generator of the GAN and to add an L1 regulator to assist in training the generator. The VAE brings in Gaussian noise by reparameterizing the latent vector, which breaks the direct link from the input to the output. Meanwhile, with sampling from a continuous normal distribution, the generator could create infinite variational output MSI patterns. In this way, one RGB image can generate unexhausted latent vectors, which can create countless MS images. The adversary network helps the generator make real-like MSIs and the L1 regulator collapses multiple possible real-like results into one result. Following the above flow, we can acquire the MS image that we desire.

The rest of the paper is organized as follows: Section II presents some related work. Section III demonstrates the proposed approach. Next, we perform several experiments to evaluate the performance of the proposed method and compare our results with state-of-the-art results in Section IV. In Section V, there is a brief discussion about some limitations of our approach. We summarize our current work and introduce some future work in Section VI. In Appendix VI, we provide all the detailed architectures of the neural network involved.

II. RELATED WORK

MS image reconstruction is not a new field. Early in 2014, Rang *et al.* tried to use synthesized RGBs after white balancing and the radial basis function (RBF) network to reconstruct MSIs. This method behaves well when the reflectance and illumination have a smooth spectrum. However, in the case of a spiky spectrum, the approach yields poor results. Rang *et al.*'s work also involves a limitation because they assumed the use of a uniform illumination to illuminate the scenes [9].

In 2016, Arad *et al.* reconstructed hyperspectral images using a sparse dictionary of hyperspectral signatures and RGB projections. Although their results achieved state-of-the-art, their approach had to make a hyperspectral prior by sampling from each dataset image, restricting the fields of its application. Also, its reconstruction quality relied heavily on the scope and specificity of the hyperspectral prior [10].

These approaches are based on traditional solutions. Such solutions have a common shortfall: they often have too many prerequisites, such as the equipment and the illumination. When the environment or the dataset changes, they need to retune their model's parameters. Deep learning approaches do not have this shortfall. Once the neural network design has been finished, we need to use only the new dataset to train the neural work when the dataset changes. These are data-driven approaches. Many researchers have tried to leverage this new technology to solve the MSI reconstruction problem.

Early in 2017, Zhiwei *et al.* proposed HSCNN based on CNN, which takes the spectrally upsampled image as input and outputs the enhanced hyperspectral images. They claimed their results significantly improved the state-of-the-art. [11].

In 2018, by removing the hand-crafted upsampling in HSCNN, Zhan *et al.* developed HSCNN+, which has two kinds of networks, HSCNN-R and HSCNN-D. HSCNN-R consists of several residual blocks, and HSCNN-D replaces the residual blocks by a dense block with a novel fusion scheme [12]. In the NTIRE 2018 Spectral Reconstruction Challenge, HSCNN-D ranked first, and HSCNN-R ranked second [13].

Meanwhile, in 2018, Berk *et al.* proposed three models based on the Convolutional Neural Network (CNN): a generic model, a conditional model, and a specialized model. The generic model is a direct mapping from RGBs to MSIs. The others two need additional networks to estimate or classify the sensitivities. Moreover, they proved that efficiently estimating the sensitivity function and conditioning the spectral reconstruction model are useful for improving reconstruction accuracy [14]. Although their approach ranked seventh in the NTIRE 2018 Spectral Reconstruction Challenge, they claimed their solution to be the most efficient, with the lowest number of layers and shortest runtime.

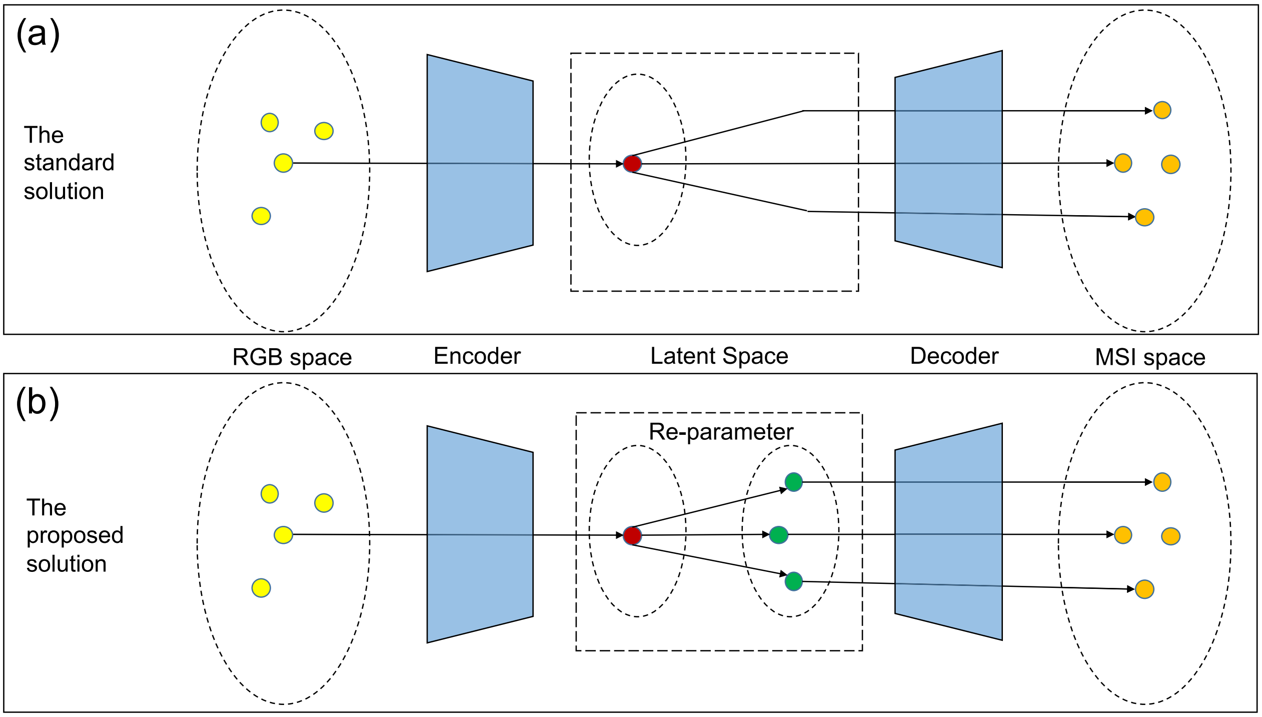

FIGURE 2. The comparison of the standard solution and the proposed solution.

Meanwhile, in 2018, Xiaolin *et al.* suggested utilizing K-means classification to separate an RGB image into different classes according to their spectrums and then applying backpropagation neural networks (BPNNs) to reconstruct the corresponding hyperspectral image. Their approach had to establish a mapping between the RGB and the MSI for each class as a foundation [15].

Furthermore, at the 2019 CVPR workshop, Kin *et al.* demonstrated a way of directly reconstructing MSIs from RGBs by using conditional GAN. Since their method is purely data-driven, it could easily lead to hallucinatory results [16].

Meanwhile, some contributed solutions appeared at the NTIRE 2020 Challenge on Spectral Reconstruction from an RGB Image [17]. For example, Jiaojiao *et al.* proposed a novel adaptive weighted attention network for spectral reconstruction. They stacked multiple dual residual attention blocks to build the backbone of the approach. Their entries obtained first rank on the ''Clean'' track and third place on the ''Real World'' track [18].

Although the above mentioned papers have achieved some progress in reconstructing MSIs from RGBs, almost all the methods except Kin *et al.*'s are based on static and dependent neural networks. Since there are no random elements in their neural networks, their models cannot generate new information or previously unseen distributions. Thus, they failed to answer the crucial question: how to supplement the lost information between RGBs and MSIs with their approaches. However, in the following sections, we will answer it with our work.

III. THE PROPOSED METHOD

A. THE PROBLEM ANALYSIS AND THE PROPOSED SOLUTION

From the previous introduction, we know that the tremendous challenge of reconstructing MSIs from RGBs is how to use a small spectral space to represent an ample spectral space. It is a severely underconstrained problem that will result in the metamerism phenomenon [19]. The main idea of metamerism is that different multispectral distributions map to the same RGB distribution. Thus, the metamerism prevents the synthesis of the correct MSIs from the RGBs, since different MSI labels with the same RGB input may cause the gradient to descend in different directions and lead to problematic convergence. Figure 2 (a) is the standard solution based on Auto Encoder, one RGB pixel maps to one latent vector; and the latent vector maps to many MS pixels, which make the model training hard to converge. It is also very similar to a multi-directional tug of war; as different teams of people exert force in different directions, the center of the rope swings in different directions.

Since the problem has been described clearly, the solution is also apparent: try to create more variational inputs to map to multiple outputs to reduce the input entanglement effect. There are two places to add variations. One is on the input side, and the other is in the latent space. However, if we add the variations on the input side, they are easily ignored. The explanation is that the structure of the Auto Encoder is good at denoising, since it compresses the input information, including the variations, and the minor variations are easily thrown out in the compression process. Several authors also have

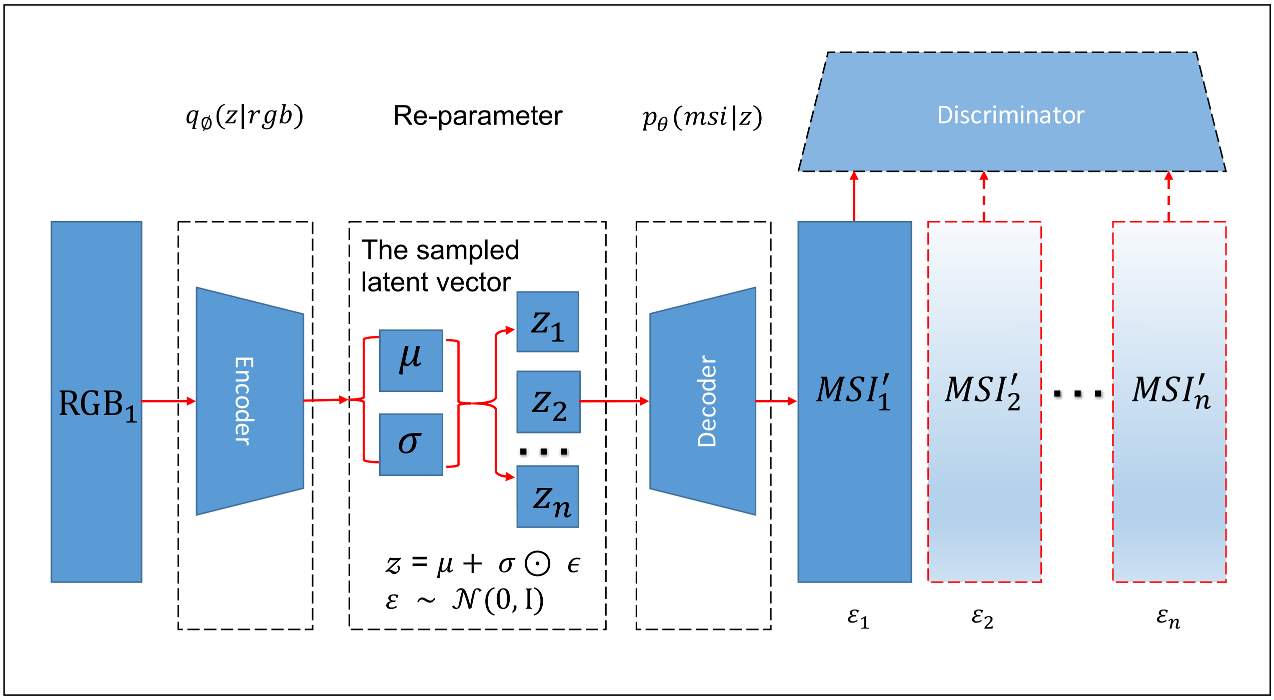

FIGURE 3. **The detailed implementation of the proposed method.**

reported this phenomenon [20], [21]. Meanwhile, we have verified this side effect in our experiments. However, if we insert the noises (variations) in the latent space, they escape the lossy compression process and are enlarged directly by the decoder. To this end, we strive to add variations to the latent vector.

Figure 2 (b) depicts our proposed solution. We use the Variational Auto Encoder instead of the typical Auto Encoder. In this way, one RGB pixel still maps to one latent vector in the first phase. However, this latent vector will be re-parametered into several different latent vectors by later random sampling from the normal distribution. The number of re-parametered latent vectors can be unlimited, greater than the limited number of possible MSI pixels. Therefore, each input with one random latent vector can map to at least one output. The re-parameterization step turns the input space from limited into unlimited. Furthermore, with the re-parameterization step, the proposed approach is equivalent to building a bridge between the two deterministic networks, encoder and decoder, and making the gradients backpropagate from output to the input. In this way, the previous underconstrained problem can be solved.

B. THE DETAILED IMPLEMENTATION OF THE PROPOSED METHOD

1) THE MODEL's ARCHITECTURE

According to the previous analysis, we designed a special GAN, as shown in Figure 3. The most significant difference from the typical GAN is that we used the VAE to substitute the AE. The AE has a fixed latent space, which means that the model's parameters are fixed when the training finishes. One input can lead to only one specific output. Because of the fixed neural network, when the metamerism phenomenon occurs, different labels with the same input cause the gradient to descend in different ways and make the training process challenging to converge. It is difficult to learn the right mapping between the input and the output.

Figure 3 shows that the proposed approach consists of three main parts: encoder, re-parameter, and decoder. The encoder is a neural network that compresses an input sample RGB_1 into a hidden representation μ-mean vector and a σ-standard deviation vector. And ϕ stands for the weights and biases. We denote the encoder by $q_\phi(z|rgb)$. And then, we re-parameterize the latent space vector z with Equation (1). In Equation (1), ϵ is sampled from the normal distribution $\mathcal{N}(0, I)$ and supplies the variations to generate diverse latent space vectors "$z_1, z_2, \ldots, z_n$". The decoder is another neural network that generates possible MSI distributions "$MSI'_1, MSI'_2, \ldots, MSI'_n$" with varied $\epsilon_n s$ "$\epsilon_1, \epsilon_2, \ldots, \epsilon_n$", and θ denotes the weights and biases. We denote the decoder with $p_\theta(msi|z)$.

$$\begin{aligned} z &= \mu + \sigma \odot \epsilon \\ \epsilon &\sim \mathcal{N}(0, I) \end{aligned} \tag{1}$$

From Equation (1), it can be seen that all the variations of latent space vector z are from sampling the normal distribution. The normal distribution can make the VAE latent space have a continuous variation compared with the AE fixed latent space. An input produces different latent vectors, and the decoder creates corresponding variational outputs with these variations. In this way, each input is tagged with a random Gaussian noise number label and re-parameterized into a unique latent vector; and one latent vector generates one particular output. Besides, the Gaussian distribution space is infinite, which guarantees that one latent vector can map to at least one output. The latent space will finally be disentangled.

2) THE MODEL's TRAINING

As noted above, our approach has three main parts: encoder, re-parameter, and decoder, which constitute the generator. To train the generator, we use the adversary network and Kullback–Leibler divergence (KL divergence) together.

Specifically, in order to get the generator, we designed 3 losses to train the model: Loss of VAE-$\mathcal{L}_{VAE}$, Loss of discriminator-$\mathcal{L}_D$, and Loss of GAN-$\mathcal{L}_{GAN}$.

$$\begin{aligned}\mathcal{L}_{VAE} &= -\mathbb{E}_{z\sim q_\phi(z|rgb)}[\log p_\theta(msi|z)] \\ &\quad + \beta\mathbb{KL}(q_\phi(z|rgb)||p_\theta(z|msi)) \qquad (2)\end{aligned}$$

$$\mathbb{E}_{z\sim q_\phi(z|rgb)}[\log p_\theta(msi|z)] = \left\|MSI - MSI'\right\|_1 \qquad (3)$$

Equation (2) is the definition of $\mathcal{L}_{VAE}$, which contains a negative log-likelihood reconstruction loss, a KL divergence regularizer and a hyperparameter β. Equation (3) is the implementation of the reconstruction loss, and here we use the Mean Absolute Error (MAE) to calculate the error between the generated MSI and the original MSI. If the decoder fails to re-build the MSIs well, this loss becomes large. In this way, this loss helps the decoder learn to reconstruct the MSIs.

Meanwhile, the KL divergence measures how much information is lost when using $q_\phi(z|rgb)$ to represent $p_\theta(z|msi)$. It also shows how close $q_\phi(z|rgb)$ is to $p_\theta(z|msi)$. In the VAE, $p_\theta(z|msi)$ respects the standard normal distribution $N(0, I)$. The encoder will receive a penalty in the loss if the output representations $p_\theta(z|msi)$ are different from those from the standard normal distribution. Meanwhile, this regularizer tries to keep the representations $p_\theta(z|msi)$ of each datapoint sufficiently different. Without the regularizer, the encoder could learn to cheat and give each input RGB pixel the same representation in a different Euclidean space region [22].

Furthermore, β is the regularization coefficient that tunes the available ratio of negative log-likelihood reconstruction loss and KL divergence. Higher β means the KL divergence will take more role, which will bring in more variations but more noise [23].

$$\mathcal{L}_{GAN} = \mathbb{E}_{rgb\sim p_{data}(rgb)}[\log(1 - D(G(rgb)))] \qquad (4)$$

In addition to Equation (2), we use an adversarial network to train the generator. Equation (4) defines the loss of generator. It works like other GANs by trying to cheat the discriminator and letting the discriminator think the generated MS images are real.

$$\begin{aligned}\mathcal{L}_D = \mathbb{E}_{msi\sim p_{data}(msi)}&[\log D(msi)] \\ &+ \mathbb{E}_{rgb\sim p_{data}(rgb)}[\log(1 - D(G(rgb)))] \qquad (5)\end{aligned}$$

Equation (5) represents the loss of the discriminator, which consists of two log-likelihood parts. The former tries to use the real MSI to train the model, and the latter allows the model learn fake examples from the results of the generator.

TABLE 1. The hardware list. [25], [26].

Device	Type	Number
CPU	Intel(R) Xeon(R) CPU E5-1620 v4 @ 3.50GHz	1
Memory	Samsung DDR4 8g	4
Solid State Disk Drive	INTEL SSDSC2BB48 480g	1
Mechanical Hard Disk Drive	WDC WD40EFRX-68N 4TB	1
GPU	NVIDIA TITAN Xp	1

TABLE 2. The software list.

Software	Version
Ubuntu	16.04.6 LTS (Xenial Xerus)
Python	3.6.7
Tensorflow-GPU	2.1.0
CUDA	10.2.89

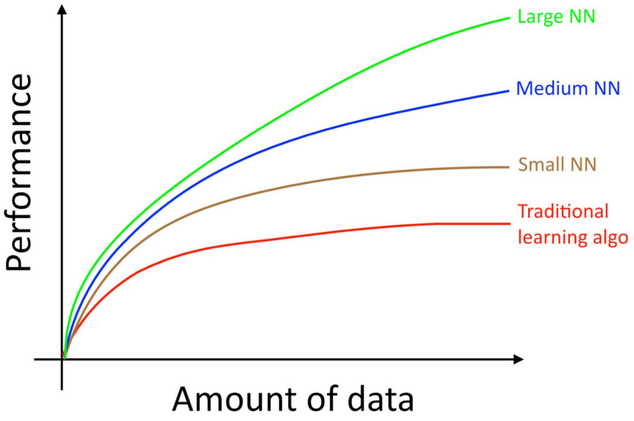

FIGURE 4. The relation between model performance and training data. [27].

Combining all the above losses, we obtain the total loss $\mathcal{L}_{Total}$ (6). There are 2 hyper-parameters, β, γ, which have different impacts on the training process. β tunes the ratio of KL divergence, γ adjusts the GAN's functional percentage. If we want the KL divergence to have more effects, we need to increase the value of β; and if we want to use the GAN more to train the generator, we need to tune up the γ. GAN and KL divergence have different advantages and disadvantages for training the model. We will give a detailed discussion and analysis in the Section V.

$$\mathcal{L}_{Total} = \left\|MSI - MSI'\right\|_1 + \beta\mathbb{KL} + \gamma\mathcal{L}_{GAN} \qquad (6)$$

IV. EXPERIMENTS

To thoroughly evaluate our approach, we selected two classical datasets: CAVE [24] and ICVL [10]. The detailed experiment description and results are in the following sections.

A. EXPERIMENT ENVIRONMENT

We list the hardware devices used in Table 1 and the software involved in Table 2.

In this project, our graphic card is NVIDIA TITAN Xp. Its architecture is Pascal and has 3840 parallel computing cores. Each core has a 1582 MHz frequency. Its memory is

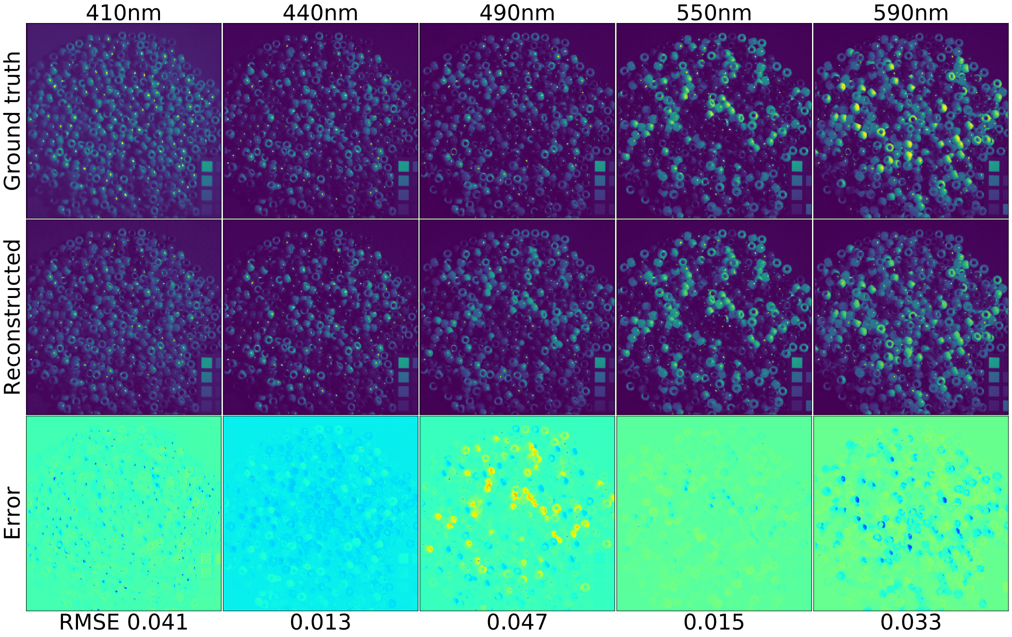

FIGURE 5. The reconstruction of five selected spectral bands using an RGB image. (The input RGB is the top left image of Figure 6.)

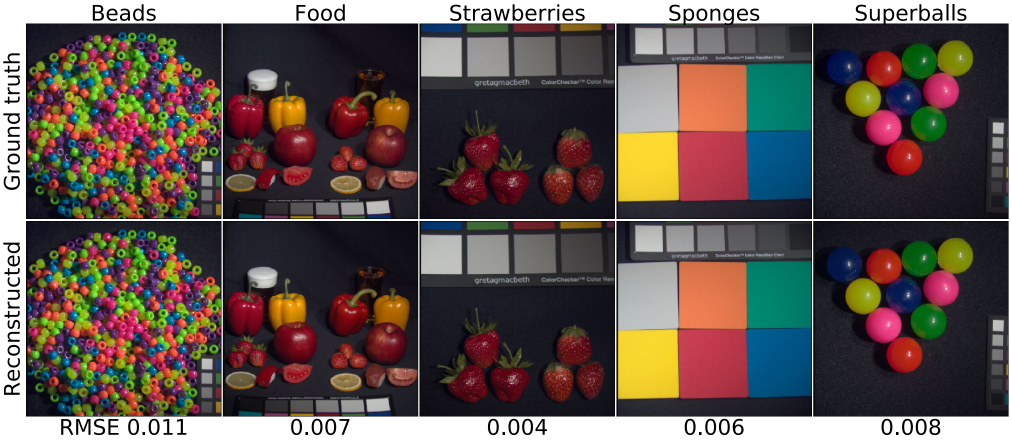

FIGURE 6. The reconstruction of five selected RGB images using MS images.

made of 12 GB GDDR5X, and its speed achieves 11.4 Gbps. The bandwidth attains a high point of 547.7 GB/s. According to our previous research results, the GPU's training speed is about 10 times faster than the CPU's [25], [26].

B. THE CAVE DATASET

1) THE DATASET INTRODUCTION AND THE HYPERPARAMETER SETTING

Columbia University Computer Vision Laboratory made the CAVE dataset. The CAVE dataset has 32 indoor scenes, including 5 categories: stuff, skin and hair, paints, food and drinks, real and fake. Moreover, each image consists of a 31-band MS image and a 3-band RGB image. The 31 bands cover visible light from 400nm to 700nm at 10nm steps. The spatial resolution of each band is 512 × 512 pixels [24].

First, we convert the CAVE dataset from 32 (scenes) × 34 (bands) png images into a 32 (samples) × 512 (height) × 512 (width) × 34 (band-columns) numpy array, where the 1-31 columns map to the 400nm-700nm bands. Meanwhile, the 32-34 columns match the R, G, B bands individually.

Then, we split the dataset into two equal-size groups according to the image's index in the dataset. Images with odd indexes, 1, 3, 5, . . . , 31, are put in one group; images with even indexes, 2, 4, 6, . . . , 32, are put in another group. We rotate these two groups to be the training dataset and the evaluating dataset to thoroughly verify our approach. Generally speaking, more training data often lead to a higher performance model. Andrew Ng has given a detailed explanation

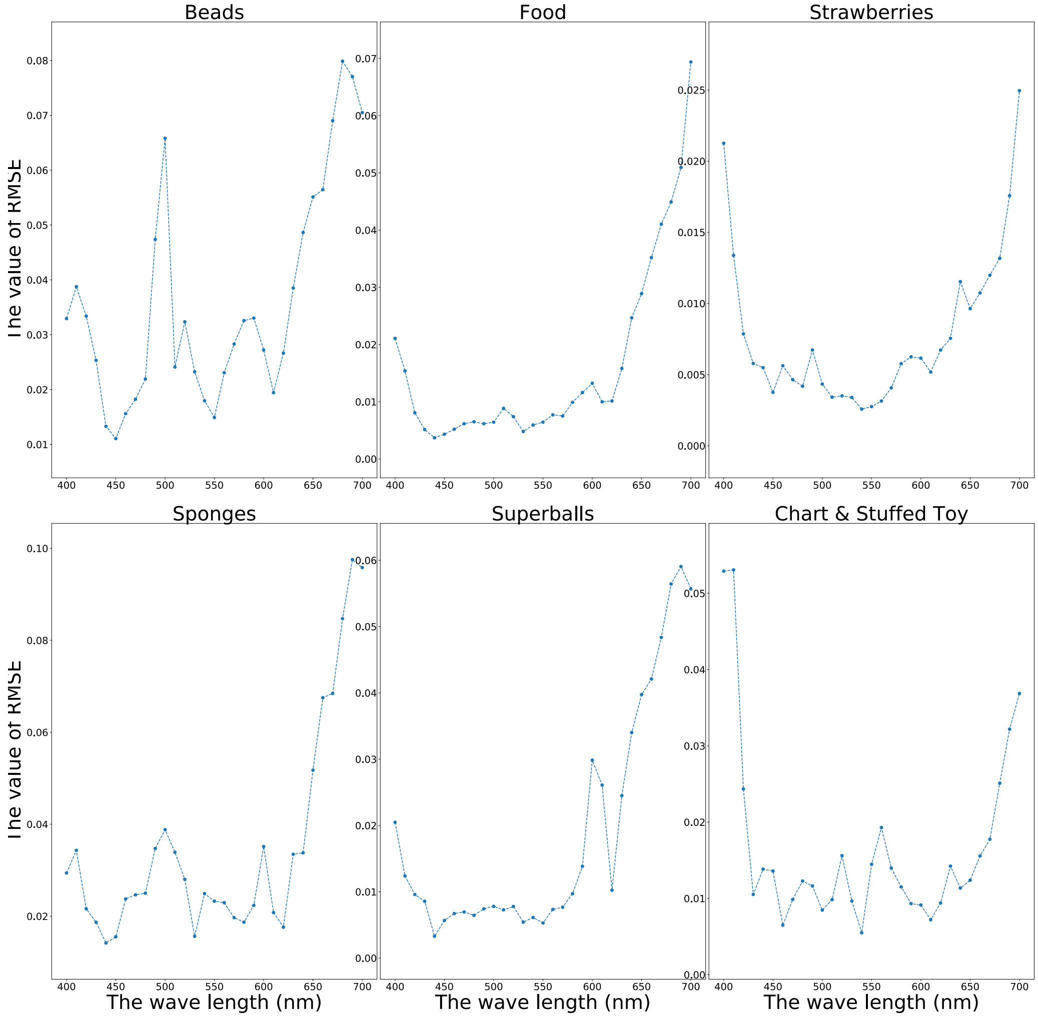

FIGURE 7. The RMSE curves of MSI reconstruction.

with Figure 4 in his book "Machine learning yearning" [27]. So, the most frequent ratio of training and evaluation is 80%:20% or 70%:30%. However, the more training data required, the fewer application domains the approach has. We choose a more challenging dataset split ratio, 50%:50%. Only a few works, like Kin *et al.*'s [16], have chosen this kind of split ratio to the best of our knowledge. Moreover, the approach of Kin *et al.* also contains a GAN loss like ours. Therefore, we use their work as the primary baseline. We also compare our results with Arad *et al.*'s and Berk *et al.*'s, since Arad *et al.* claimed their results are state-of-the-art [10] and Berk *et al.* claimed theirs is the first successful estimation of the spectral data from a single RGB image captured in unconstrained settings [14].

To thoroughly evaluate our method and compare it with the previous baseline, we also performed bi-directional translations between RGBs and MSIs and conducted qualitative and quantitative evaluations, respectively.

Furthermore, we normalized the data from the range [0.0-1.0] to [-1.0-1.0] before training and recovered the image data to [0.0-1.0] after training. After many trials, we found that the best setting to train the generator: $\beta = 0$, $\gamma = 10$, the training epoch is 300, and the optimizer is Adam with the learning rate $1e^{-4}$.

2) THE QUALITATIVE EVALUATION

In this subsection, the qualitative results of the two phases: RGB to MSI and MSI to RGB, are shown. Since we chose Kin *et al.*'s work [16] as the baseline, all the selected images and training configurations are the same as theirs.

Figure 5 shows the result of the RGB to the MSI. We chose image "beads" and selected 5 spectral bands to reconstruct the MSI from the RGB image. To make the paper's layout look tidy, we put the ground truth input RGB image in Figure 6 top left. Moreover, we selected the same five bands as Kin *et al.* did to compare the results equally. The five bands are between 400nm and 700nm with approximately equal intervals. Since it is hard to identify nearby wavelength lights' subtle differences with the naked eyes, the five bands are

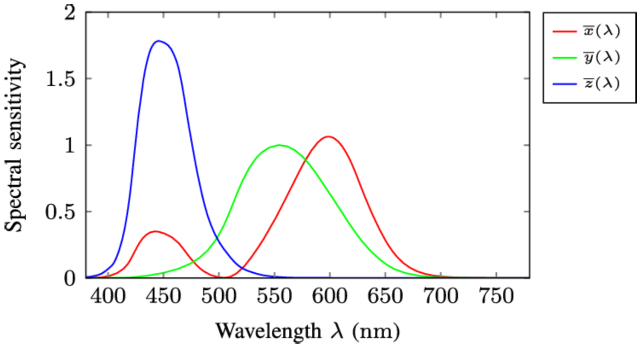

FIGURE 8. CIE 1931 color matching functions [28].

enough to demonstrate the qualitative evaluation of spectral reconstruction. Furthermore, it is a prevalent way to select several bands to show qualitative results [10], [14]. Otherwise, in the quantitative evaluation section, we demonstrate the full RMSE varieties against the whole spectrum (400nm-700nm) reconstruction results in Figure 7.

Besides, we got Error maps using the prediction image minus the ground truth image and then pseudocolored the error images with the "jet" colormap. Red, green, and blue indicate negative, zero, and positive errors, respectively. After comparing our results with Kin *et al.*'s work, we found that our model behaves better than theirs, especially in the 410nm, 550nm, and 590nm bands.

Moreover, we can quickly reconstruct the RGB images from MS images with a similar neural network when we reverse the input and output of the VAE generator and make some small changes, such as resetting the input and output size. Figure 6 is the result of reconstructing the RGB from the MSI. We find that the reconstructed RGB images and ground truth images are too close to judge with the naked eyes. Furthermore, the reconstruction behaves better than Kin *et al.*'s work [16].

3) THE QUANTITATIVE MEASUREMENT

To compare our results fairly with the related work, we chose three classical quantitative metrics: Root Mean Square Error (RMSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM), to evaluate our model's performance.

As stated above, we practiced with 16 MS and RGB images with the odd number position to train the model and another 16 scenes with the even number position to test the model, and vice versa. We performed the previous process 3 times and then calculated the average value. Furthermore, the reconstruction experiment is bi-directional, which means that we measure the results of converting both the RGB to the MSI and the MSI to the RGB. In order to highlight the comparison easily, we include our results with Berk *et al.*'s [14], Kin *et al.*'s [16], and Arad *et al.*'s [10] in the Table 3 and Table 4, respectively.

TABLE 3. The average RMSE, PSNR and SSIM of reconstructing MSI from RGB.

Metrics	**Berk**	**Kin**	**Arad**	**Ours**
Approach	CNNs	cGANs	Sparse coding	**VAE-GAN**
Ratio of training and testing	N/A	50%:50%	N/A	**50%:50%**
RMSE∼(0-255)	N/A	8.0622	5.4	**5.741**
RMSE∼(0-1)	0.038	N/A	N/A	**0.023**
PSNR	28.78	N/A	N/A	**34.00**
SSIM	0.94	N/A	N/A	**0.98**

TABLE 4. The average RMSE, PSNR and SSIM of reconstructing RGB from MSI.

Metrics	**Berk**	**Kin**	**Ours**
Approach	CNNs	cGANs	**VAE-GAN**
Ratio of training and testing	N/A	50%:50%	**50%:50%**
RMSE∼(0-255)	2.55	5.649	**1.943**
RMSE∼(0-1)	0.038	N/A	**0.0076**
PSNR	28.78	N/A	**42.96**
SSIM	0.94	N/A	**0.99**

From Table 3, it can be seen that Arad *et al.* have the lowest RMSE. However, our results are very close to theirs and rank second. For this result, we provide the following explanation.

First, the ratio of splitting the CAVE dataset for training and evaluation is not explicit in Arad *et al.*'s paper. As noted above, in most cases, more training data usually yield a higher performance model.

Second, before reconstruction, their method had to make a hyperspectral prior by sampling from each image. The sampling ratio of the CAVE dataset is 3.8% of each image. Their approach needs to gather information from the whole dataset, including the training dataset and the test dataset, which will help their model improve its performance. However, the requirements for more information will limit the application scope of their approach. On the contrary, the ratios of Kin *et al.* and our approaches are 50% for training and 50% for testing. The training dataset and the testing dataset are entirely isolated, which means that the model knows nothing about the test dataset when doing the testing. So, our approaches' application fields will be more prevalent. Given the above analysis, it is not fair to compare our and Kin *et al.*'s results with Arad *et al.*'s results. By comparing our results with Kin *et al.*'s, it can be seen that the RMSE of reconstructing the MSI from the RGB has been reduced by 29%, and the RMSE of reconstructing the RGB from the MSI has been reduced by 66%.

Besides, we selected the same six images as Kin *et al.* did in their paper [16]: "Beads," "Food," "Strawberries," "Sponges," "Superballs," and "chart & Stuffed Toy" to calculate their RMSEs of MSI 31 bands reconstruction and draw the curves of RMSEs against the wavelengths in the Figure 7.

Figure 7 reveals that the model behaves with different prediction abilities when facing different scenes. Moreover,

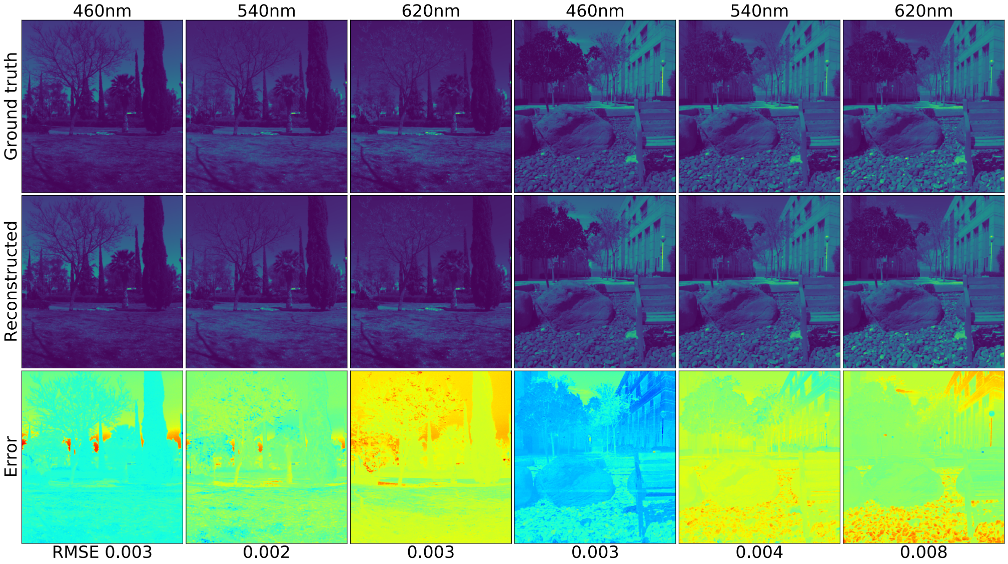

FIGURE 9. The reconstructions of three selected spectral bands images of two scenes. (The two input RGBs are the top left two images of Figure 10.)

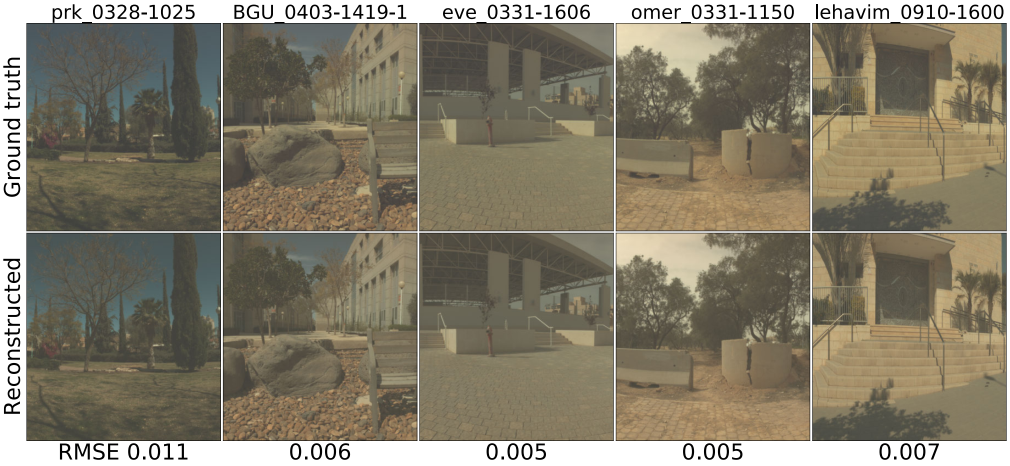

FIGURE 10. The reconstruction of five selected RGB images using MS images.

the model performs better when reconstructing a band image whose wavelengths are in the middle range. Our explanations are as follows. From Figure 8 CIE 1931 standard observer color matching functions [28], we notice that most of R, G, and B components are distributed principally from 450nm to 650nm wavelengths and a few ingredients are allocated to near the two ends of the spectrum. A color pixel constituted of R, G, and B bands holds little information about the two ends of the spectrum, which may cause the low reconstructing ability at the two ends of the spectral zone.

C. THE ICVL DATASET

1) THE DATASET INTRODUCTION AND THE HYPERPARAMETERS SETTING

The Ben-Gurion University interdisciplinary Computational Vision Lab made the ICVL dataset. The latest version contains 201 images (Most scenes are outdoor.) taken by a Specim PS Kappa DX4 hyperspectral camera. Moreover, each image has 1392 × 1300 spatial resolution over 519 spectral bands ($400-1,000nm$ at roughly $1.25nm$ increments) [10].

In the experiment, we used the reduced ICVL dataset supplied by Arad *et al.* They reduced the 519 bands to 31 bands of roughly 10 nm in 400–700 nm for two reasons: reducing computational cost and facilitating comparison to previous benchmarks that employ this kind of representation [10]. Moreover, we calculated the RMSE between the RGB image generated by 519 bands and the RGB image generated by 31 bands. The RMSE is about 0.00014, which means the two formats have few differences.

To evaluate our approach thoroughly and non-overlappingly, we first randomized the order of 201 images and then divided them into 6 groups. Each of the first 5 groups, 0, 1, 2, 3, 4, has

32 images. However, the sixth group, group 5, has 41 images. We rotated one of the first five groups, $0 - 4$, as the training dataset and the remain five groups as the testing dataset. For example, if we take group 1 as the training dataset, groups 0, 2, 3, 4, and 5 are the testing dataset, which means that the training and testing dataset splitting ratio reaches $32 : 169$, 32 images for training and 169 images for testing.

We split each training dataset of 32 images into two equal subgroups according to the images' index in the dataset. Images with odd indexes, $1, 3, 5, \ldots, 31$, are put in one subgroup, whereas images with even indexes, $2, 4, 6, \ldots, 32$, are put in another subgroup. We rotate these two subgroups to be the training dataset and the validating dataset.

Furthermore, we reduce the 32 training images' spatial resolution from 1392 × 1300 to 512 × 512 by sampling randomly one of the four parts (top left, top right, bottom left, bottom right) of the original image. The final training dataset consists of 32 images with a spatial resolution of 512 × 512. Meanwhile, the testing dataset includes 169 images with a spatial resolution of 1392 × 1300. To the best of our knowledge, we are the first to use so few images to train the ICVL model.

To thoroughly evaluate our method, we also performed bi-directional translations between RGBs and MSIs and conducted quantitative and qualitative evaluations. Besides, we selected three related excellent works, Zhiwei *et al.* [11], Arad *et al.* [10], and Berk *et al.* [14], for comparison. Among them, Zhiwei *et al.*'s HSCNN claimed their results were state of the art.

Furthermore, we normalized the data from the range [0.0-1.0] to [−1.0-1.0] before training and recovered the image data to [0.0-1.0] after training. After many trials, we found the best setting to train the generator: $\beta = 0, \gamma = 10$, the training epoch is 300, and the optimizer is Adam with the learning rate $1e^{-4}$.

2) THE QUALITATIVE EVALUATION

In this subsubsection, we continue to demonstrate the qualitative results of the two reconstruction phases: RGB to MSI and MSI to RGB. We emphasized again that the training data is completely isolated from the testing data.

Since Arad *et al.* created the ICVL dataset and claimed their results are state-of-the-art, we chose their results as the baseline. Figure 9 is made like Arad *et al.*'s Figure 4. We selected the same two scenes and the same three bands (460nm, 540nm, and 620nm) to demonstrate our model's reconstruction performance. To make the paper's layout look tidy, we put the two ground truth input images in the top left first image "prk_0328-1025" and top left second image "BGU_0403-1419-1" of Figure 10.

Moreover, we got Error maps using the prediction image minus the ground truth image and then pseudocolored the error images with the "jet" colormap. Red, green, and blue indicate negative, zero, and positive errors, respectively.

When comparing Figure 9 and Figure 10 with Figure 5 and Figure 6, a phenomenon can be easily observed; the same approach works better on the ICVL dataset than on the CAVE dataset. Our explanation is that most of the CAVE dataset images contain large dark background areas, which provide little information for training the model.

3) THE QUANTITATIVE MEASUREMENT

To compare our work fairly with the previous work, we chose four classical quantitative metrics: Root Mean Square Error (RMSE), Normal or Relative Root Mean Square Error (nRMSE or rRMSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM), to evaluate our model's performance.

Moreover, we prepare two experiments and several comparisons with the state-of-the-art to demonstrate our approach's superiority.

In the first experiment, we compared our approach, VAE-GAN, with the two state-of-the-art studies: Arad *et al.*'s "sparse coding" and Zhiwei *et al.*'s HSCNN. For equal comparison, we did some processes on the ICVL dataset.

According to the report from Arad *et al.*, the whole experiment was conducted on the "complete set" of 102 images. After double-checking, we found that image "lst_0408-0924" does not exist in their supplied dataset [29]. So, we deleted image "lst_0408-0924" from the complete set. The number of the complete set images thus became 101. Moreover, 59 domain-specific images belong to the 101 images. There are 5 domains, or subsets: Park, Indoor, Urban, Rural, and Plant-life. Arad *et al.* selected one image from a set for testing and the rest of the set images to train the dictionary. Furthermore, they repeated the previous step until every image in the set had been chosen for testing. In the last step, they calculated the average relative RMSE (rRMSE). In this way, Arad *et al.* achieved better results in the domain-specific subsets than with the complete set [10].

Zhiwei *et al.*'s HSCNN performed in a more generalizable way. They used a total of 200 images, including the complete set to train a CNN model with 141 images, excluding the domain-specific subsets. They tested the model on the 59 domain-specific images. Then, they trained another model with 159 images, excluding the non-domain-specific images in the complete set, and tested the obtained model on these 41 images. In this way, the images for training and testing were rigorously separated, and HSCNN eliminated the domain-specific restriction imposed in sparse coding [11].

Our dataset splitting is similar to Zhiwe *et al.*'s but has more challenges. We used the up to date dataset, which contains 201 images. Moreover, we used only the 100 images, excluding the 101 complete set's images to train the model. We then tested the model on the 101 complete set images, including the 59 domain-specific images. In this way, we isolated the training and testing images rigorously, eliminated the domain-specific restriction, and reduced the number of training images.

Table 5 compares the results of the above three approaches. We find that our approach, VAE-GAN, surpasses the two other approaches in the complete set and most of the subsets

TABLE 5. The comparison of RGB to hyperspectral conversion.

	Arad et al. (Sparse coding) Train:Test N/A	Zhiwe et al. (HSCNN) Train:Test 141:59	Ours (VAE-GAN) Train:Test 100:101
Data Set	**rRMSE**	**rRMSE**	**rRMSE**
Complete set	0.0756	0.0388	**0.0348**
Park subset	0.0589	0.0371	**0.0334**
Indoor subset	0.0507	0.0638	**0.0463**
Urban subset	0.0617	0.0388	**0.0322**
Rural subset	0.0354	**0.0331**	0.0381
Plant-life subset	0.0469	0.0445	**0.0426**
Subset average	0.05072	0.04346	**0.03852**

TABLE 6. The bi-directional conversion results of VAE-GAN on ICVL dataset.

VAE-GAN Train:Test ——32:169					
Metrics	**PSNR**	**SSIM**	**RMSE**	**rRMSE**	**RMSE_INT**
RGB to MSI	43.388	0.999	0.008	0.037	1.921
MSI to RGB	46.809	1.000	0.005	0.013	1.349

TABLE 7. The CAVE-RMSE changes against the β variations.

β	PSNR	SSIM	RMSE (0-1)	RMSE (0-255)
0	34.037	0.975	0.022	5.696
0.01	34.033	0.973	0.022	5.635
0.1	33.783	0.961	0.023	5.777
1	30.585	0.825	0.031	8.003
10	20.764	0.375	0.097	24.802
100	16.314	0.526	0.169	43.032
$\alpha = 100, \gamma = 1000, \delta = 10$, train_epoch=300				

TABLE 8. The ICVL-RMSE changes against the β variations.

β	PSNR	SSIM	RMSE (0-1)	RMSE (0-255)
0	44.432	0.998	0.007	1.794
0.01	36.727	0.884	0.015	3.821
0.1	24.098	0.446	0.070	17.954
1	24.047	0.826	0.072	18.260
10	24.274	0.784	0.075	19.239
100	24.274	0.808	0.075	19.116
$\alpha = 100, \gamma = 1000, \delta = 10$, train_epoch=300				

TABLE 9. The generator of RGB_to_MSI.

	Layer	Input	Output	Activation
Encoder	Dense	RGB(3)	512	Leaky_Relu
	Dense	512	512	Leaky_Relu
	Dense	512	100+100	N/A
Z	Latent Re-parameter	100+100	100	N/A
Decoder	Dense	512	512	Leaky_Relu
	Dense	512	512	Leaky_Relu
	Dense	512	MSI(31)	Tanh

in the table. Moreover, this record was created with the lowest ratio of training and testing.

In the second experiment, we handled an extreme challenge to thoroughly explore our approach's full capacity. We did not use the above tactic of splitting the dataset. Rather, we used the splitting tactic introduced in subsubsection IV-C1.

TABLE 10. The discriminator of RGB_to_MSI.

	Layer	Input	Output	Activation
Discriminator	Dense	MSI(31)	64	Leaky_Relu
	Dense	64	64	Leaky_Relu
	Dense	64	1	Sigmoid

TABLE 11. The generator of MSI_to_RGB.

	Layer	Input	Output	Activation
Encoder	Dense	MSI(31)	512	Leaky_Relu
	Dense	512	512	Leaky_Relu
	Dense	512	100+100	N/A
Z	Latent Re-parameter	100+100	100	N/A
Decoder	Dense	512	512	Leaky_Relu
	Dense	512	512	Leaky_Relu
	Dense	512	RGB(3)	Tanh

TABLE 12. The discriminator of MSI_to_RGB.

	Layer	Input	Output	Activation
Discriminator	Dense	RGB(3)	64	Leaky_Relu
	Dense	64	64	Leaky_Relu
	Dense	64	1	Sigmoid

We randomized the order of 201 images and then divided them into 6 groups. Each of the first 5 groups, 0, 1, 2, 3, 4, has 32 images. The sixth group, group 5, has 41 images. We rotated one of the first five groups, $0 - 4$, as the training dataset and the remain five groups as the testing dataset. For example, if we take group 1 as the training dataset, groups 0, 2, 3, 4, and 5 are the testing dataset, which means the training and testing dataset splitting ratio reaches $32 : 169$, 32 images for training and 169 images for testing.

Moreover, we split each training dataset 32 images into two equal subgroups according to the image's index in the dataset. Images with odd indexes, $1, 3, 5, \ldots, 31$, were put into one subgroup, and images with even indexes, $2, 4, 6, \ldots, 32$, were put in another subgroup. We rotated these two subgroups to be the training dataset and the validating dataset.

Furthermore, we reduced the 32 training images' spatial resolution from 1392×1300 to 512×512 by sampling randomly one of the four parts (top left, top right, bottom left, bottom right) of the original image. The final training dataset consists of 32 images with a spatial resolution of 512×512. And the testing dataset includes 169 images with a spatial resolution of 1392×1300. To the best of our knowledge, we are the first to use so few images to train the ICVL model.

With the above training and testing tactic, we got the experimental results of RGB to MSI conversion and MSI to RGB conversion, respectively; these results are listed in Table 6. This is the first novel experiment with this kind of dataset splitting to achieve the best results to our knowledge. In comparison with previous CAVE results, Table 3 and Table 4, it can be quickly seen that the ICVL results are much better than the CAVE results, which is consistent with the previous ICVL Qualitative results' analysis.

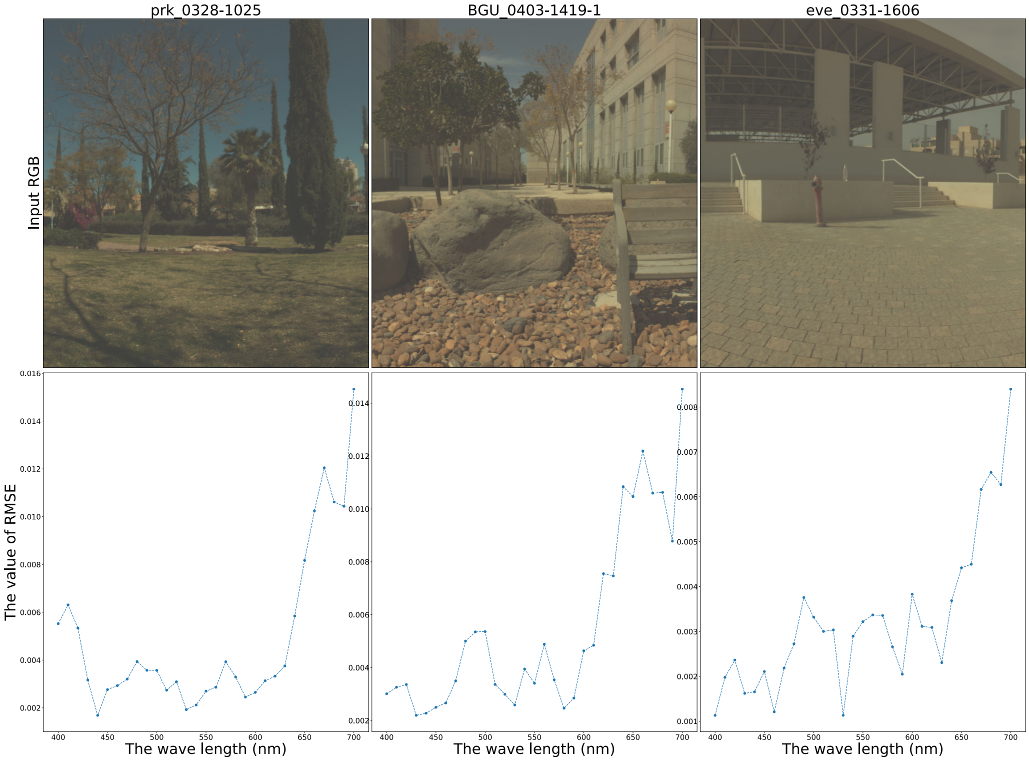

FIGURE 11. The RMSE curves of MSI reconstruction.

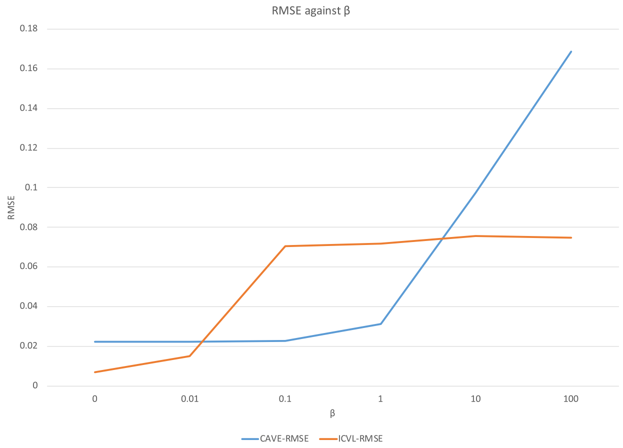

FIGURE 12. The RMSE against the β.

Besides, we selected three images: “prk_0328-1025”, “BGU_0403-1419-1”, and “eve_0331-1606” to calculate their MSI reconstruction RMSE of 31 bands individually and drew their RMSE curves in the Figure 11. “prk_0328-1025” and “BGU_0403-1419-1” appeared in Arad *et al.*’s paper [10], and Zhiwei *et al.* adopted “eve_0331-1606” as an example [11].

It is worth noting in considering Figure 11, that the model behaves worse in the long-wavelength range. Our explanation of that phenomenon is that because the three images were taken outdoors in daylight, the light with long wavelengths was rapidly immersed in the background noise. With a regular camera, it is hard to capture the subtleties of long-wavelength light. Thus, because the RGB image contains few features of long-wavelength light, which is the chief reason for the poor MSI reconstruction performance in the long-wavelength spectral zone.

V. LIMITATIONS

VAE-GAN has many hyperparameters. Therefore, there are numerous means of optimizations.

For example, we used two losses, KL-divergence and GAN, to train the model; however, we did not know which is better. β is the key hyper-parameter, which can tune the functional percentage of the two losses. We performed two interesting experiments to investigate how the model’s reconstruction ability alters with β’s change on the CAVE and ICVL datasets, respectively.

Table 7 is the experiment data for the RMSE change against the β variations on the CAVE dataset, and Table 8 is the data for the RMSE change against the β variations on the ICVL dataset. Furthermore, for convenient visual and comparison purposes, we included the above two tables’ data in Figure 12.

From them, we can find a common rule: The model’s prediction performance will decrease as the value of β increases, which indicates the GAN loss has a better efficiency on training the generator to synthesize real-like MSIs or RGBs.

For the above phenomenon, we explain that KL-divergence tunes the encoder by forcing the latent vector of z generated to be close to the normal distribution. The learning gradients only pass through the encoder. Only the encoder has to be updated. However, GAN adjusts the whole generator network, including the encoder and decoder, by making the outputs more like the real ones. The gradients go through the whole generator. The decoder and encoder have to be updated together. So, the GAN has higher efficiency to train the generator.

However, the above explanation is only our hypothesis; it lacks a strong theory and experimental supports. We need to undertake further research to address this problem soon.

VI. CONCLUSION AND FUTURE WORK

In this paper, we first introduced the challenge of MSIs' reconstruction from RGBs and the related works' common shortages. Because their approaches are based on static and dependent neural networks, they cannot generate new variations to supplement the lost information between the MSIs and RGBs. Next, we analyzed the bottle neck of the problem and elaborated on our proposed approach, which leverages re-parameterizing latent vectors and GAN tricks to create new variational MSI-like images. In this way, one latent vector can evolve into many latent vectors respecting the normal distribution. One input RGB image with random latent space vectors can be created out of the lost possible multiple outputs. We then used the GAN and L1 regulator to make the possible multiple outputs convergence into one real-like MS image output. Thus, we were able successfully to solve the metamerism problem and transform the one-to-many problem into a one-to-one problem by bringing in random latent vectors. We used qualitative and quantitative methods to evaluate our approach to the CAVE and ICVL datasets. With much less training data than the previous approaches, we got comparable results on the CAVE dataset and surpassed the state-of-the-art results on the ICVL dataset.

In the future, we plan to conduct more research on optimizing the hyperparameter settings. We will expand our approach to more datasets to verify its versatility.

APPENDIX
THE DETAILED NEURAL NETWORK ARCHITECTURE

In this section, we list all the detailed structures of the neural networks dealt with in Section III in Tables 9, 10, 11, and 12 respectively. The training batch size is 10240 and the testing batch size is 262144 (512 × 512). The training epoch is 300.

REFERENCES

[1] K. Ose, T. Corpetti, and L. Demagistri, "2—Multispectral satellite image processing," in *Optical Remote Sensing of Land Surface*, N. Baghdadi and M. Zribi, Eds. Amsterdam, The Netherlands: Elsevier, 2016, pp. 57–124. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9781785481024500028

[2] I. Makki, R. Younes, C. Francis, T. Bianchi, and M. Zucchetti, "A survey of landmine detection using hyperspectral imaging," *ISPRS J. Photogramm. Remote Sens.*, vol. 124, pp. 40–53, Feb. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0924271616306451

[3] C. M. Gevaert, J. Suomalainen, J. Tang, and L. Kooistra, "Generation of Spectral–Temporal response surfaces by combining multispectral satellite and hyperspectral UAV imagery for precision agriculture applications," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 3140–3146, Jun. 2015.

[4] S. Andersson-Engels, J. Johansson, and S. Svanberg, "Medical diagnostic system based on simultaneous multispectral fluorescence imaging," *Appl. Opt.*, vol. 33, no. 34, pp. 8022–8029, Dec. 1994. [Online]. Available: http://ao.osa.org/abstract.cfm?URI=ao-33-34-8022

[5] G. P. Ellrod, "Advances in the detection and analysis of fog at night using goes multispectral infrared imagery," *Weather Forecasting*, vol. 10, no. 3, pp. 606–619, 1995, doi: 10.1175/1520-0434(1995)010<0606:AITDAA>2.0.CO;2.

[6] S. Baronti, A. Casini, F. Lotti, and S. Porcinai, "Multispectral imaging system for the mapping of pigments in works of art by use of principal-component analysis," *Appl. Opt.*, vol. 37, no. 8, pp. 1299–1309, 1998. [Online]. Available: http://ao.osa.org/abstract.cfm?URI=ao-37-8-1299

[7] M. Magnusson, J. Sigurdsson, S. Armannsson, M. Ulfarsson, H. Deborah, and J. Sveinsson, "Creating RGB images from hyperspectral images using a color matching function," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2020, pp. 1–4.

[8] C. Abraham. (2020). *A Beginner's Guide to (CIE) Colorimetry*. [Online]. Available: https://medium.com/hipster-color-science/a-beginners-guide-to-colorimet%ry-401f1830b65a

[9] R. M. Nguyen, D. K. Prasad, and M. S. Brown, "Training-based spectral reconstruction from a single RGB image," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 186–201.

[10] B. Arad and O. Ben-Shahar, "Sparse recovery of hyperspectral signal from natural RGB images," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 19–34.

[11] Z. Xiong, Z. Shi, H. Li, L. Wang, D. Liu, and F. Wu, "HSCNN: CNN-based hyperspectral image recovery from spectrally undersampled projections," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 518–525.

[12] Z. Shi, C. Chen, Z. Xiong, D. Liu, and F. Wu, "HSCNN+: Advanced CNN-based hyperspectral recovery from RGB images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 939–947.

[13] J. Wu, J. Aeschbacher, and R. Timofte, "In defense of shallow learned spectral reconstruction from RGB images," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 1042–1051.

[14] B. Kaya, Y. B. Can, and R. Timofte, "Towards spectral estimation from a single RGB image in the wild," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Seoul, South Korea, 2019, pp. 3546–3555, doi: 10.1109/ICCVW.2019.00439.

[15] X. Han, J. Yu, J.-H. Xue, and W. Sun, "Spectral super-resolution for RGB images using class-based BP neural networks," in *Proc. Digit. Image Comput., Techn. Appl. (DICTA)*, Dec. 2018, pp. 1–7.

[16] K. G. Lore, K. K. Reddy, M. Giering, and E. A. Bernal, "Generative adversarial networks for spectral super-resolution and bidirectional RGB-To-Multispectral mapping," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 926–933.

[17] B. Arad, R. Timofte, O. Ben-Shahar, Y.-T. Lin, and G. D. Finlayson, "Ntire 2020 challenge on spectral reconstruction from an RGB image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, Jun. 2020, pp. 446–447.

[18] J. Li, C. Wu, R. Song, Y. Li, and F. Liu, "Adaptiveweighted attention network with camera spectral sensitivity prior for spectral reconstruction from RGB images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 462–463.

[19] S. E. Palmer, *Vision Science: Photons to Phenomenology*. Cambridge, MA, USA: MIT Press, 1999.

[20] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, May 2016, pp. 1125–1134.

[21] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," Cornell Univ., Ithaca, NY, USA, Tech. Rep., 2015, doi: 1511.05440.

[22] J. Altosaar. (2020). *Tutorial—What is a Variational Autoencoder*. [Online]. Available: https://jaan.io/what-is-variational-autoencoder-vae-tutorial/

[23] I. Higgins, L. Matthey, X. Glorot, A. Pal, B. Uria, C. Blundell, S. Mohamed, and A. Lerchner, "Early visual concept learning with unsupervised deep learning," Cornell Univ., Ithaca, NY, USA, Tech. Rep., 2016, doi: 1606.05579.

[24] F. Yasuma, T. Mitsunaga, D. Iso, and S. Nayar, "Generalized assorted pixel camera: Post-capture control of resolution, dynamic range and spectrum," Columbia Univ., New York, NY, USA, Tech. Rep. CUCS-061-08, Nov. 2008.

[25] X. Liu, H. A. Ounifi, A. Gherbi, Y. Lemieux, and W. Li, "A hybrid gpu-fpga-based computing platform for machine learning," *Procedia Comput. Sci.*, vol. 141, pp. 104–111, Dec. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050918318052

[26] X. Liu, H.-A. Ounifi, A. Gherbi, W. Li, and M. Cheriet, "A hybrid GPU-FPGA based design methodology for enhancing machine learning applications performance," *J. Ambient Intell. Hum. Comput.*, vol. 11, no. 6, pp. 2309–2323, Jun. 2019, doi: 10.1007/s12652-019-01357-4.

[27] A. Ng. (2017). *Machine Learning Yearning*. [Online]. Available: http://www.mlyearning.org/(96)

[28] M. Amara, F. Mandorlo, R. Couderc, F. Gerenton, and M. Lemiti, "Temperature and color management of silicon solar cells for building integrated photovoltaic," *EPJ Photovolt.*, vol. 9, p. 1, Dec. 2018.

[29] ICVL. (2020). *Icvl Dataset*. [Online]. Available: http://icvl.cs.bgu.ac.il/hyperspectral/

XU LIU received the B.S. degree in physics from Jilin University, Jilin, China, in 2005, and the master's degree in electronics and communication engineering from Peking University, Beijing, China, in 2011. He is currently pursuing the Ph.D. degree in software engineering with the École de Technologie Supérieure (ÉTS), Montreal, QC, Canada.

From 2017 to 2018, he was a Research Intern with Ericsson Research, Montreal. Since 2019, he has been a Research Assistant with the Synchromedia Laboratory. His research interests include machine learning, image processing, and high-performance computing.

ABDELOUAHED GHERBI (Member, IEEE) received the Ph.D. degree in computer engineering from Concordia University, Canada. He is currently an Associate Professor with the Software and IT Engineering Department, École de Technologie Supérieur (ETS), Montreal, QC, Canada. He was a Visiting Researcher with the Defense Research and Development Canada (DRDC), Valcartier, QC, Canada. His research interests include mainly model-driven software engineering, modeling and analysis techniques for real-time and critical software systems, software performance, high availability, and security.

ZHENZHOU WEI is currently pursuing the bachelor of software engineering (B.S.E.) degree with McGill University, Canada. Her research interests include image processing, cloud computing, computer vision, machine learning, and natural language processing.

WUBIN LI received the Ph.D. degree from Umeå University, Sweden. He is currently working as a Researcher of cloud technologies with Ericsson Research, Montreal, QC, Canada. His research interests include storage systems, distributed systems, cloud computing, high availability, and applied mathematics. His research interests also include enhancing edge/cloud systems with analytics and machine intelligence.

MOHAMED CHERIET (Senior Member, IEEE) received the B.Sc. degree in CE from Bab Ezzouar University, Algiers, and the DEA and Ph.D. degrees from the University of Paris 6, Paris 6, France.

As a Scientist and an Educator, he has taken an active role in publishing technical articles and authoring books. He has published 70 international journal articles and 135 international conference papers, and has delivered 17 invited talks. In addition, he has authored and published six books on pattern recognition, document image analysis and understanding, and computer vision. Among them, the book entitled *Character Recognition Systems: A Guide for Students and Practitioners*, a Textbook for Students and Practitioners, is highly acclaimed. He is also recognized for his activities in technical journal editorial writing, organizing, and taking part in many conferences. He has contributed to the training of 65 high qualified personnel. He has also served as the Chair of the IEEE's Montreal CIS Chapter.