

## Research Article

# Adaptive CNN Ensemble for Complex Multispectral Image Analysis

Syed Muslim Jameel <sup>1</sup>, Manzoor Ahmed Hashmani,<sup>1,2,3,4</sup> Mobashar Rehman,<sup>5</sup> and Arif Budiman<sup>6</sup>

<sup>1</sup>Department of Computer and Information Sciences, Universiti Teknologi PETRONAS (UTP), Perak, Malaysia

<sup>2</sup>Centre for Research in Data Science (CERDAS), UTP, Perak, Malaysia

<sup>3</sup>High Performance Cloud Computing Centre (HPC3), UTP, Perak, Malaysia

<sup>4</sup>Institute of Electrical and Electronics Engineers (IEEE), Piscataway, NJ, USA

<sup>5</sup>Faculty of Information and Communication Technology (FICT), Universiti Tunku Abdul Rahman, Perak, Malaysia

<sup>6</sup>Faculty of Computer Science, University of Indonesia, Depok, West Java, Indonesia

Correspondence should be addressed to Syed Muslim Jameel; muslim\_16000370@utp.edu.my

Received 4 December 2019; Accepted 10 March 2020; Published 15 April 2020

Academic Editor: Danilo Comminiello

Copyright © 2020 Syed Muslim Jameel et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Multispectral image classification has long been the domain of static learning with nonstationary input data assumption. The prevalence of Industrial Revolution 4.0 has led to the emergence to perform real-time analysis (classification) in an online learning scenario. Due to the complexities (spatial, spectral, dynamic data sources, and temporal inconsistencies) in online and time-series multispectral image analysis, there is a high occurrence probability in variations of spectral bands from an input stream, which deteriorates the classification performance (in terms of accuracy) or makes them ineffective. To highlight this critical issue, firstly, this study formulates the problem of new spectral band arrival as virtual concept drift. Secondly, an adaptive convolutional neural network (CNN) ensemble framework is proposed and evaluated for a new spectral band adaptation. The adaptive CNN ensemble framework consists of five (05) modules, including dynamic ensemble classifier (DEC) module. DEC uses the weighted voting ensemble approach using multiple optimized CNN instances. DEC module can increase dynamically after new spectral band arrival. The proposed ensemble approach in the DEC module (individual spectral band handling by the individual classifier of the ensemble) contributes the diversity to the ensemble system in the simple yet effective manner. The results have shown the effectiveness and proven the diversity of the proposed framework to adapt the new spectral band during online image classification. Moreover, the extensive training dataset, proper regularization, optimized hyperparameters (model and training), and more appropriate CNN architecture significantly contributed to retaining the performance accuracy.

## 1. Introduction

In the last decade, the multispectral imagery has manifested as an exigent requirement for analyzing the many critical applications, such as health, military, geology atmosphere, and agriculture [1–3]. Multispectral images divide the electromagnetic spectrum into many spectral bands. Remote sensing using satellites [4], Unmanned Aerial Vehicle (UAV) [5], and Advance Driver Assistance Systems (ADAS) [6] are the fundamental sources to obtain the multispectral imagery

and perform classification and prediction tasks to analyze the behavior of objects or scenes [7]. The distinct types of multispectral images have different processing needs and thus also comes with new challenges to algorithms that analyze the data [7], such as optimization of model computational complexity, accuracy, robustness, and adaptability. A recent survey [8] discusses the potential applications and challenges associated with the nighttime light observations. This study highlights the issues of time-series data analysis for multispectral images. Notably, it

correlates the dynamicity factors of time-series data analysis with various temporal inconsistencies, such as atmospheric conditions, satellite shift, and sensor degradation. Also, this study urges the research community to investigate this area in order to overcome temporal inconsistencies.

In the modern era of digitization, real-time and online analysis of big data is an essential task [9]. Multispectral imagery is one of the dominant and information-intensive types of big data. In a real-time and online learning scenario, the multispectral image classification models accrued a large amount of time-series data from multiple heterogeneous data sources. Data analysis from the multiple data sources is an enabler for a more comprehensive interpretation of objects or scenes. Also, it significantly contributes to the improvement in performance accuracy [10, 11]. However, the current multispectral image classification models are not yet sufficiently robust to various perturbations in the data [12]. For example, there is a high occurrence probability of feature-wise input data changes (virtual drift). Hence, incorporating the adaptability feature is essential to handle the virtual drift.

Moreover, we deduced that the arrival of a new spectral band during online learning is a potential virtual drift issue in multispectral image classification perspective, for example, different combinations of electromagnetic radiation having different wavelengths. These wavelengths form several spectral bands or channels and collectively act as a multispectral image. These spectral bands yield different deep insights, depending on the nature of the problem, such as to get insights from the natural landscape; typically, Red (R), Green (G), and Blue (B) spectral bands are useful, whereas near infrared (NI) and  $R$  spectral bands are essential to detect the healthy vegetation. Also, short infrareds (SIRs) can classify the different soil characteristics [13]. In online learning or multidata sources, scenario change in the existing spectral bands can be observed in the legacy system. It could be due to the change in classification aspect or due to change in sensor technology. The various combinations of different spectral bands can change with a different timestamp, or new spectral band can arrive. The legacy image classification models cannot sustain these changes and become obsolete or less accurate. This issue possesses a massive significance in various future critical applications in the era of the 4<sup>th</sup> Industrial Revolution. Therefore, in this study, we have proposed an adaptive mechanism, which is able to handle such temporal inconsistency during online multispectral classification.

**1.1. Contribution.** The significant contributions of this study involve the following aspects:

- (1) Problem formulation of potential virtual drift (new spectral band arrival issue) in online multispectral image classification scenario
- (2) Proposal of an adaptive CNN ensemble framework for dynamic multispectral image classification with a novel ensemble approach and optimized CNN model

- (3) Validate the adaptive feature and classification performance of adaptive CNN ensemble framework

The rest of the paper is structured as follows.

Section 2 briefly surveys the related work, formulates the new spectral band arrival issue, and raises two (2) potential research questions (RQs) and research objectives (ROs). Section 3 is relevant to Research Objective-1 and proposes an adaptive framework to adapt to the new spectral band. Section 4 is relevant to Research Objective-2 and validation of the proposed framework. Section 4 discusses the experiments, and results are obtained in detail. Section 5 presents the conclusion of this study.

## 2. Related Work and Theoretical Foundation

Based on intensive literature review, we conclude that multispectral data are recognized due to its unique characteristics of spatial, spectral, and temporal features. The data formation is directly linked to (1) characteristics of the spatial resolution (pixel size), (2) characteristics of the spectral resolution (wavelength range), (3) characteristics of the temporal resolution (when and how often images are collected), and (4) input data source (single or multidata sources) [14]. Despite the advantageous characteristics of multispectral data, several challenges are also addressed in the literature which are discussed here after:

- (1) High spatial resolution is an essential element to analyze the ecological process from a different level of abstraction. LiDAR method is an example to extract the structure and physical information of the scene or objects. The high spatial resolution of LiDAR can make the 3D representation of the target but contains a complex structure [15]. LiDAR provides unprecedented performance accuracy if its associated complexities are tackled adequately. Several new approaches have been proposed to handle the associated complexities with LiDAR. A recent study proposes the region-based convolutional neural networks (R-CNNs) for automated detection of multiple classes of archaeological objects in the LiDAR dataset [16]. R-CNN provides a more appropriate strategy for object detection using localized classification. However, LiDAR data formation is an expensive process and needs specialized hardware. Superresolution (SR) [17] method is essential to generate high spatial data without hardware dependency. Initially, SR was proposed to improve the low spatial resolution into high spatial resolution for Landsat remote sensing images. SR approach fuses the multiframe complementary information into a single high spatial resolution. Later, numerous efforts have been made to improvise this method further. For example, a study [18] presents the multiscale residual neural network (MRNN). MRNN adopts the multiscale nature of satellite images to reconstruct high-frequency information accurately for SR.

- (2) The machine learning approaches have utilized to observe the change in environmental and human activities. These approaches compare the historical and present remote sensing data to detect the change in the spatiotemporal pattern [19]. During temporal changes, observing the ground truth values and data preprocessing become a challenging task [19]. This makes the supervised machine learning ineffective. A recent study has proposed an unsupervised approach, called Deep SFA (DSFA) [20]. DSFA is based on slow feature analysis theory and deep network to detect the change for multitemporal remotes sensing images.
- (3) Unlike the color imagery, the multispectral images are captured using reflectance of light in several narrow frequencies (for hyperspectral in hundreds) called spectral bands. The spectral bands give more comprehensive details and measures of the object or scene. Moreover, the individual spectral band has different kinds and levels of details. Machine learning has been an essential tool to analyze this information, such as land-use mapping, land-cover mapping, forest inventory, and urban-area monitoring [21].
- (4) The problem of multidata sources is a significant concern for the remote sensing community. The integration of the data from various heterogeneous data sources provides a powerful approach for generating more detailed characteristics. However, multidata sources also contribute to generating sophisticated attributes in the data. The integration of the different dimensions is highlighted as a critical issue in few earlier studies. Lefsky et al. [22, 23] discuss the possibility of integration of the third dimension in two-dimensional remote sensing data using LiDAR and RADAR. The integration of remotely sensed data into models can provide for model parameter initialization, incorporated as a condition or state variables, or for validation of model outcomes [24]. Hence, the problem of multidata sources has been dramatically escalated in recent years [25].

In summary, the current state of the art for multispectral analysis is online learning analysis. Online learning is most relevant to dynamic data assumptions. In dynamic data assumption, the ever-growing availability of data was captured by heterogeneous sensors and coupled with advance approaches for analysis, such as machine learning. However, there is a high probability of observing changes or inconsistencies in characteristics (spatial, spectral, temporal, or multidata sources) of multispectral/remote sensing data. The traditional machine learning approaches efficiently work on static data assumptions and have limitations in a dynamic environment [26]. The static nature causes performance degradation in terms of performance accuracy or makes them ineffective. Hence, there is a need to develop an adaptive mechanism to work in a dynamic environment. Therefore, this study highlights the spectral band inconsistency (new spectral band arrival issue) and proposes an adaptive framework to adapt the new spectral band during online multispectral image classification.

**2.1. Problem Formulation.** Virtual concept drift applies in online learning, in which the learner/model (machine learning model) trains by the training dataset and then classifies by different input data streams, such as the statistical properties of input data may vary at different time steps (additional features can arrive). For example, the learner  $M$  at time  $t$  ( $M_t$ ) trains with a sample of training data  $D \in Dm$  with  $m$  dimension. Each  $D_t$  contains input features  $\{f_1, f_2, f_3, \dots, f_m\}_t$ . Similarly, in the case of multispectral scenario, the single spectral band can be considered as a single input feature. Hence, the arrival of the new spectral band is a potential virtual concept drift and makes the learner ineffective. Therefore, this study considers a single multispectral channel as a feature of input data (multispectral input image). During the online multispectral classification task, a learner is able to classify the input data. This input data are arriving from multidata sources, such as multiple satellites, and could cause virtual concept drift, as shown in Figure 1(a). Virtual concept drift phenomenon is very critical to handle in the context of the multispectral classification problem. In a multispectral classification problem, the learner is trained to classify on certain spectral bands and cannot recognize the newly arrived spectral bands and drastically degrades its classification performance. As shown in Figures 1(a) and 1(b), the input data are arriving from three different satellites (satellite-1 to satellite-3) with the predefined number of spectral bands, such as Red (R), Green (G), and Blue (B). Later, another data source (new satellite) is introduced in the system with the capability of the new spectral band, such as infrared (IR). However, the learner only possesses the classification ability for the feature from R, G, and B. Learner is not able to correctly classify the new arrival features, such as IR; thus, this will decrease classification performance. Upgradation or addition of new satellite with advanced sensors is only one of the causes of this change. Based on the above discussion, we open some relevant research question (discussed in Table 1) and investigate their research objectives.

### 3. Adaptive Convolutional Neural Network (CNN) Ensemble Framework

This section elaborates the proposed adaptive CNN ensemble framework. The intuition behind this framework is to adapt the new spectral band during an online multispectral image classification scenario. In the proposed framework, the diversity of the ensemble mechanism helps to handle the possible arrival of the new spectral band. More precisely, this framework proposes a novel ensemble approach which contributes the diversity to the ensemble system in a simple yet effective manner. This proposed framework has five (5) modules which are as follows:

- (1) Dynamic ensemble classifier (DEC) module
- (2) Performance feedback module
- (3) Training sample repository module
- (4) New instance initiate and train module
- (5) Add instance initiate module

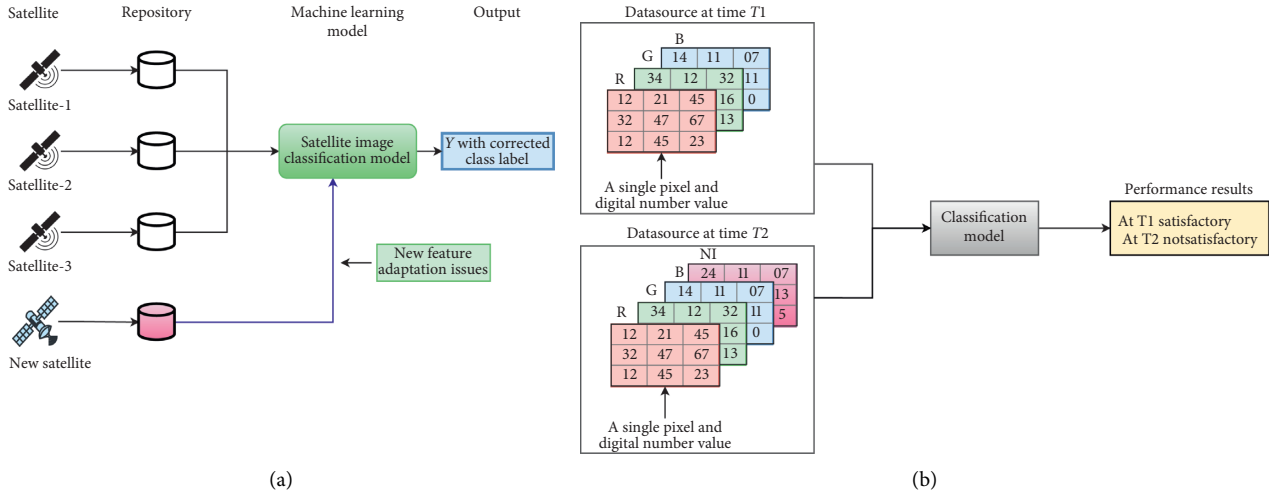


FIGURE 1: Virtual concept drift illustration scenario: (a) introducing advanced remote sensing satellite to the existing system; (b) new spectral band arrival (near-infra (NI)) scenario during online multispectral image classification.

TABLE 1: The potential research questions and research objectives of this study.

Research questions	Research objectives	Section
RQ1: how can a multispectral model adapt the new features (spectral bands) during the online classification scenario?	RO1: to propose an adaptive multispectral image classification framework to detect and adapt the newly arrived features (spectral bands) during the online classification scenario	Section 3
RQ2: how can an online multispectral model retain its performance (in terms of accuracy) after noticing a new feature (spectral band)?	RO2: to validate effectiveness, diversity, and classification performance of the proposed framework after noticing a new feature (spectral band)	Section 4

**3.1. DEC Module.** DEC module contains several instances of optimized CNN model. The instances in the DEC module are subject to increase after detecting the new spectral band. Each instance is responsible for classifying the individual channel (spectral band) separately. The classification outputs from each channel is passed to the performance feedback (PF) module. PF module utilizes this information to detect the new spectral band arrival. Also, the individual output from each classifier is aggregated using the voting ensemble approach to generate the single classification output for each multispectral image, using equation (1). We compute weighted majority vote by associating a weight  $w_j$  with classifier  $C_j$ , where  $X_A$  is the characteristic function, such that  $C_j(X) = i \in A$ , and  $A$  is the set of unique class label:

$$\hat{y} = \arg \max \sum_{j=1}^m w_j x_A(C_j(x) = i). \quad (1)$$

In the DEC module, we have presented a novel ensemble approach (single instance for the single band) for the ensemble. In this approach, the individual optimized CNN model handles the individual spectral band. The simplicity of this ensemble approach makes it more convenient to adapt new spectral band in an online scenario. DEC module updates dynamically after observing the new spectral band. Furthermore, the section below discusses the architecture of the optimized CNN model in detail.

**3.1.1. The Architecture of Optimized CNN Model.** The optimized architecture of CNN was found after several model hyperparameter tuning, such as image size, the number of filters, filter size, number of layers, activation function, types, and configuration of layers. Besides, we also performed the training for hyperparameter tuning, for example, change in EPOCH, batch size, and learning rate. Finally, we found twenty (20) layered CNN architecture most appropriate, as shown in Figure 2. The optimized CNN model possesses twenty (20) layers, fourteen (14) layers for the feature extraction, and six (6) layers for classification. The optimized CNN model uses six (6) different types of layers with tuned parameters, such as (1) batch-normalization layer, (2) convolutional layer, (3) Gaussian noise layer, (4) pooling layer, (5) dropout layer, and (6) dense layer. The optimized CNN model takes  $50 \times 50 \times 1$  size of image input and performs multiclassification tasks (classifies the ten (10) classes, 0–9 handwritten digit). The architecture is carefully designed and fine-tuned during many experiments. In the feature extraction process, the optimized CNN model extracts the relevant feature from the input image channel and sends for classification purpose.

(1) *Batch-Normalization (BN) Layer.* BN layer helps to reduce internal covariate shift in optimized CNN model using equation (2). The covariant shift leads to the possible usage of higher learning rates [27]. Therefore, it was essential to use the BN layer in our CNN model. We kept  $50 \times 50$  size of the

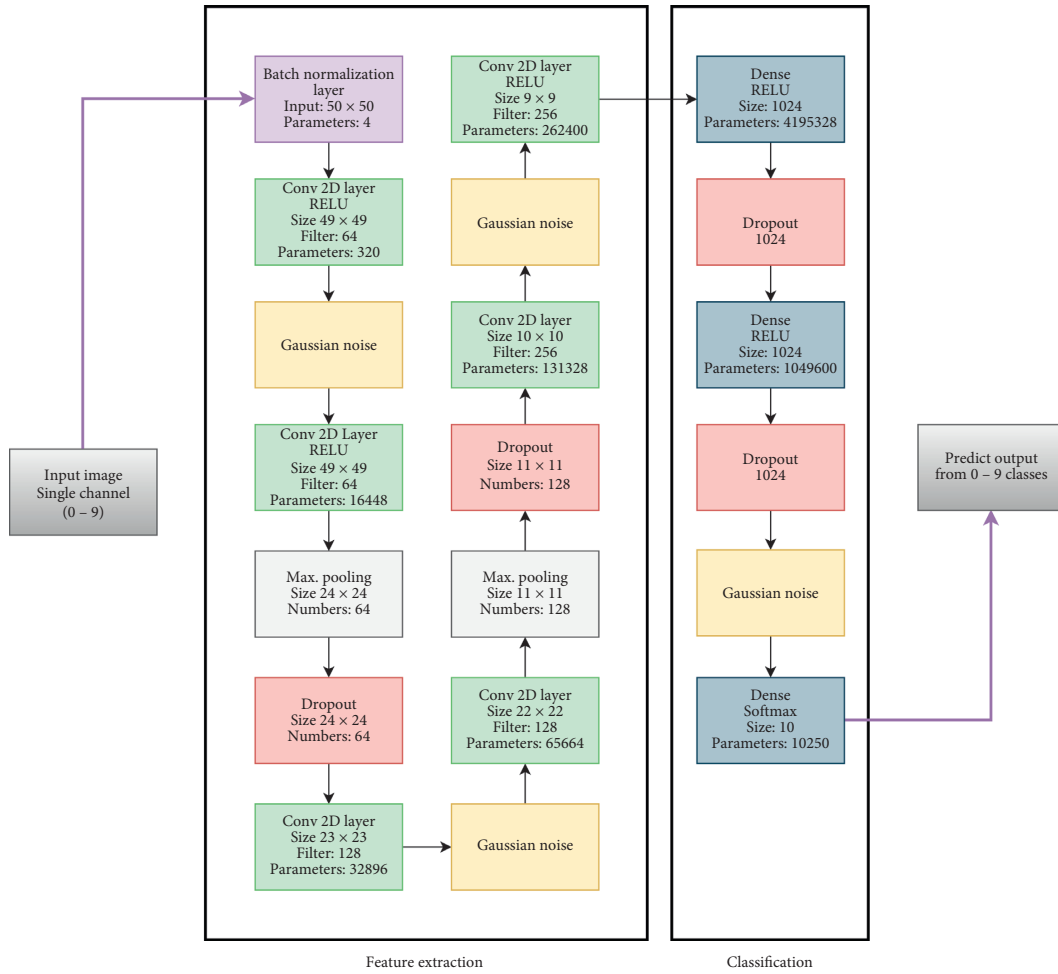


FIGURE 2: Optimized (20 layered) CNN architecture used as an instance of the ensemble in DEC module.

input (the same size as of input image channel) and four (4) parameters including gamma ( $\gamma$ ) weights, beta ( $\beta$ ) weights, moving mean ( $\mu$ ), and moving variance ( $\sigma$ ). Gamma ( $\gamma$ ) and beta ( $\beta$ ) weights only learn during training (along with the original parameters of the network), whereas moving mean ( $\mu$ ) and moving variance ( $\sigma$ ) are not learnable:

$$y = BN\gamma, \beta(x_i). \quad (2)$$

(2) *Convolution Layer.* Our optimized CNN model uses six (06) different convolutional layers to extract the low-level and high-level input image features. The filters are convolved onto the image sample and added biases. We used the  $2 \times 2$  filter size to obtain more depth image features. In literature, it is recommended to keep the number of filters in the range of powers of two (2). Therefore, we kept it as 64, 128, and 256 and got a satisfactory performance, as shown in Table 2. The input size of subsequent convolutional layers reduces due to pooling and dropout layers. The output of each convolution operation produces a 2D activation map. 2D activation map possesses linear values. Subsequent feature map values are calculated, using equation (3), where the input image is denoted by  $f$ , and our filter is denoted by  $h$ . The indexes of rows and columns of

the result matrix are marked with  $m$  and  $n$ , respectively. To introduce nonlinearity in each convolutional layer, we applied the RELU activation function [28] using equation (4). For each convolutional layer, the trainable parameters are calculated using equation (5), where  $F$  is the number of filters (kernels),  $F_{pre}$  is the number of the previous filter,  $S_f$  is the kernel size, and  $b$  is the bias. In the case of first convolutional layer,  $F_{pre} = 1$ :

$$\text{Conv} = (f * h)_{[m,n]} = \sum_j \sum_k h_{[m,n]} f_{[m-j, n-k]}, \quad (3)$$

$$\text{Relu}(\text{Conv})_z = \max(0, z), \quad (4)$$

$$P = F \times (F_{pre} \times S_f) + b. \quad (5)$$

(3) *Gaussian Noise (GN) Layer.* Many studies [29, 30] have noted that adding small amounts of input noise (jitter) to the training data often aids generalization and fault tolerance. In our optimized CNN model, in total, four (4) GN layers were used (3 in feature extraction and 1 in classification part) to reduce the overfitting issue and ensure the regularization effect. We found that our model showed best generalization

TABLE 2: The architecture of each convolutional layers in optimized CNN model.

Layers	Input size	Filter size	Number of filters	Trained parameters	Padding	Stride	Activation function
Conv2D 1	49 × 49	2 × 2	64	320	Zero	1	RELU
Conv2D 2	48 × 48	2 × 2	64	16,448	Zero	1	RELU
Conv2D 3	23 × 23	2 × 2	128	32,896	Zero	1	RELU
Conv2D 4	22 × 22	2 × 2	128	65,664	Zero	1	RELU
Conv2D 5	10 × 10	2 × 2	256	131,328	Zero	1	RELU
Conv2D 6	9 × 9	2 × 2	256	262,400	Zero	1	RELU

performance when the noise value is 0.01 using the following equation:

$$P_G(\text{noise}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-((\text{noise}-\mu)^2/2\sigma^2)}. \quad (6)$$

(4) *Pooling Layer*. In pooling layer, the max pooling approach is used with the patch size = 2 × 2. Here, we downsampled feature maps by summarizing the presence of a feature (by a factor of 2) in patches of the feature map, using equation (7). In our optimized CNN model, the pooling layer operates upon  $x$  channel with the matrix  $H \times W$ . The pooling operator maps a subregion into a single number for maximum pooling, as recommended in the literature [31]:

$$\max: y_{i^{l+1}, j^{l+1}, d} = \max x_{i^{l+1}}^l \times H + i, j^{l+1} \times W + j, d, \quad (7)$$

$$0 \leq i < H, 0 \leq j < W.$$

(5) *Dropout Layer (DL)*. To avoid overfitting and code-dependencies among the neurons, we have utilized the DL. We used four (4) DL, two (2) in feature extraction, and two (2) in classification part. DL randomly deactivates specific units (neurons) in a layer with a certain probability  $p$  from a Bernoulli distribution. Also, dropout momentarily switches off some neurons in a layer so that they do not contribute any information or learn any information during those updates, and the onus falls on other active neurons to learn harder and reduce the error [32]. We keep the dropout rate  $p = 0.5$  for achieving maximum regularization in the following equation:

$$r_j^l \sim \text{Bernoulli}(p). \quad (8)$$

(6) *Dense Layer*. Primarily, the feature matrix is flattened (feature vector) and fed into dense layer. Every neuron from the feature vector connected with each neuron of the dense layer and followed by a nonlinear activation function. Equations (9)–(11) represent the mathematical relationship between the feature map and the dense layer, where  $F_{ci}$  is a fully connected layer,  $F_{cm}$  is fully connected layer as the middle layer, and  $F_{cfi}$  is fully connected layer as the final layer. We used RELU and SoftMax (last layer) nonlinear activation functions. Besides, we used three fully connected layers in our proposed architecture and used the dropout approach to eliminate the drawback of complex computations due to more parameters [33]:

$$F_{ci} = F_{vec} = \sum_{j=1}^n x_j, \quad \text{where } i \text{ is } R \text{ and } F_{vec} \ni F_c, \quad (9)$$

$$F_{cm} = \text{Relu}(F_c)_z = \max(0, z), \quad (10)$$

$$F_{cfi} = \text{SoftMax}(F_{ci}) = \frac{e^{F_{ci}}}{\sum_j e^{F_{ci}}}. \quad (11)$$

3.2. *Performance Feedback Module*. The performance feedback module is responsible for detecting the possible arrival of a new spectral band. A significant decrease in the error distance implies a change in the input data stream and suggests that the learning model is no longer appropriate to handle it. It is essential to mention a threshold level (Th) of classification accuracy to detect the new spectral band. The wrongly classified input samples will be considered as new spectral details (image) if  $\text{Th} < 50\%$ . Each misclassified sample (with classification accuracy  $< 50\%$ ) will be transferred to the training sample repository module.

3.3. *Training Sample Repository Module*. Training sample repository module contains the wrongly classified sample images (provided by performance feedback module). Sample images are considered as the potential newly arrived spectral band (because our model did not recognize these samples). Once the number of misclassified images reached fifty (50), they will be clustered using K-mean clustering approach by keeping the value of  $K = 1$ . As per literature, fifty (50) images are enough for training a module. More specifically, K-mean clustering approach and cosine distance measure are used to determine the new class. The centroid of the obtained cluster is compared with the individual sample image using the cosine distance. The intuition behind measuring the cosine distance of cluster centroid and input sample is to determine more relevant samples. Cosine similarity formula can be derived from equation (12), where  $\vec{a}$  and  $\vec{b}$  are two vectors containing the information of cluster centroids and input sample image. Later, all similar image samples are categorized into a hypothetical class, such as  $X$ . The hypothetical class  $X$  contains the samples of new spectral band images. Finally, the new instance (optimized CNN model) of ensemble initiates and trains using the class  $X$  in the new instance training module:

$$\text{Similarity}(A, B) = \cos \theta = \frac{A \cdot B}{|A| \cdot |B|},$$

$$\vec{a} \cdot \vec{b} = |\vec{a}| \cdot |\vec{b}| \cos \theta, \quad (12)$$

$$\cos \theta = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|}.$$

**3.4. New Instance Online Training Module.** New instance online training module suggests the idea of online training during classification. This module initiates a new instance (optimized CNN model) and trains using the obtained class  $X$  (from the training sample repository module). After completing the training process, this module forwards the new instance towards the add instance module. However, manual intervention is mandatory to determine the actual class label for hypothetical class  $X$ . This can be replaced manually by the original value such as  $X = \text{infraray}$ .

**3.5. Add Instance Module.** The add instance module adds the new instance in our existing ensemble DEC module. The dynamicity in DEC module allows adding a new instance.

**3.6. How It Works?** The core part of the adaptive CNN ensemble framework is the DEC module. DEC module consists of multiple optimized CNN instances, such as  $I(i_1, i_2, \dots, i_n)$ . However, every single instance is responsible for handling the individual spectral band of the multispectral image, such as  $B_n = (B_1, B_2, \dots, B_n)$ . For each time period  $t$ , the individual CNN instance classifies the individual spectral band, via Algorithm 1. For each classification, the individual CNN instance forwards their classification accuracy to the performance feedback module to be evaluated as a potential new spectral band. The performance feedback module follows specific criteria to initiate the new spectral band detection process, such as if the classification accuracy is less than the threshold value ( $\text{Th} < 50\%$ ). Also, the DEC module aggregates the individual classification performance using the weighted majority ensemble approach, via Algorithm 2. Later, performance feedback module forwards the misclassified sample images to the training sample repository module. Training sample repository module segregates the misclassified sample image into a single cluster using the K-mean clustering ( $K = 1$ ). Also, this module measures the cosine distance [34] of each misclassified samples with the centroid of the cluster to identify further relevant sample images. The available sample images in the cluster are given a hypothetical class name, such as  $X$ . Class  $X$  is a potential new spectral band class. New spectral band class  $X$  (along with its sample images) is forwarded to the new instance online training module. This module initiates and trains the new instance of optimized CNN model and forwards to add instance module. Add instance module adds newly trained model to the DEC module. Figure 3 presents the graphical

representation, and Algorithm 3 defines the detail steps of the proposed adaptive CNN ensemble framework.

## 4. Experimental Results

To verify the effectiveness and performance of the proposed framework, three (3) subsections are presented here: (1) Section 4.1 details data preparation and transformation of multispectral dataset. (2) Section 4.2 introduces the experimental criteria, experimental setup, and performance measures, and (3) Section 4.3 displays the results of the experiment and discusses the analysis and deduction.

**4.1. Data Preparation and Transformation.** Most of the multispectral datasets are domain-specific and are only valid for particular applications. However, it is desirable to validate the proposed model using the generic dataset. For example, the MNIST dataset [36] is widely used as a benchmark for different image analysis tasks. The MNIST dataset is composed of greyscale handwritten images, but it is not appropriate for multispectral image analysis. The idea of a generic dataset for multispectral images was firstly proposed by Xiaozhou Wang at Kaggle, named as Multispectral MNIST dataset (the dataset is available at <https://drive.google.com/drive/folders/1HwAcRdtDSba68u-lMDDqzwBYLjfhIoWj?usp=sharing>). Multispectral MNIST dataset contains seven hundred thirteen (713) multispectral images of handwritten numbers between 0 and 9, from six (06) different peoples, using two (02) different pens. The ten (10) grayscale images, with  $350 \times 350$  pixel dimension, represent ten (10) channels for the multispectral image (each greyscale image value 0–255 represents a channel). During initial experiments, we found multispectral MNIST dataset failed to classify the image samples correctly and showed not more than 20% of training accuracy and 16% validation accuracy, as shown in Figure 4. We hypothesized class imbalance and a smaller number of data samples (713 data samples) as a reason for this performance degradation. Therefore, to overcome this issue, we have prepared an extended version of multispectral MNIST dataset called extended multispectral MNIST dataset (EMMD). To improve the performance of this dataset, firstly, we have removed the data samples which were causing the class imbalance problem. Later, we have increased the number of samples up to sixty thousand (60,000) samples using the image augmentation techniques, such as image flipping, random cropping, random scaling, center zooming, increasing/decreasing brightness, sharpness, and introducing slight noises into the image pixel values. To further enhance its performance, we normalized data from 0–255 to 0–1, while preserving the aspect ratio and achieved breakthrough towards the improvement in performance accuracy which is more than 97%, as shown in Figure 5. During model training and testing, we have reduced the image size (feed input) up to  $50 \times 50$  pixels to lower the computational complexity, as shown in Table 4. The obtained results proved suitability of EMMD dataset for the multispectral image classification. However, the low classification performance for the

**Input:** input sample  $S_o$  with spectral bands  $B: (b_1, b_2, \dots, b_n)$ ,  $p = 0.5$ , filter size  $F$  (64, 128, 256)

//For data preprocessing

- (1) Apply batch-normalization process, via equation (2)

//For feature extraction task

- (2) Perform convolution operation with  $F=64$  and nonlinearity, via equations (3) and (4)
- (3) Add Gaussian noise, via equation (6)
- (4) Apply max-pooling, via equation (7)
- (5) Activate and deactivate neurons using dropout with  $p$ , via equation (8)
- (6) Perform convolution operation and nonlinearity, with  $F=64$ , via equations (3) and (4)
- (7) Perform convolution operation and nonlinearity,  $F=128$ , via equations (3) and (4)
- (8) Add Gaussian noise, via equation (6)
- (9) Perform convolution operation and nonlinearity,  $F=128$ , via equations (3) and (4)
- (10) Apply max-pooling, via equation (7)
- (11) Activate and deactivate neurons using dropout with  $p$  via equation (8)
- (12) Perform convolution operation and nonlinearity,  $F=256$ , via equations (3) and (4)
- (13) Add Gaussian noise, via equation (6)
- (14) Perform convolution operation and nonlinearity,  $F=256$ , via equations (3) and (4)

//For classification tasks

- (15) Flatten feature vector and connect with dense layer, via equations (9) and (10)
- (16) Activate and deactivate neurons using dropout with  $p$ , via equation (8)
- (17) Apply dense layer, via equations (9) and (10)
- (18) Activate and deactivate neurons using dropout with  $p$ , via equation (8)
- (19) Add Gaussian noise, via equation (6)
- (20) Calculate the probability score, equations (9) and (11)

**Output: classify (B)**

ALGORITHM 1: Single instance classification algorithm.

**Input:** classify ( $B_n$ )

- (1) Determine the accuracies from single instances,  $I: (i_1, i_2, \dots, i_n)$ , via Algorithm 1
- (2) Apply weighted majority criteria to determine classification accuracy, via equation (1)
- (3) Show the sample output

**Output: ensemble\_Vote (Classify ( $B_n$ )),  $B_n = (B_1, B_2, \dots, B_n)$**

ALGORITHM 2: Dynamic ensemble classifier.

channel-0 and channel-1 is due to noisy data, which was inherited from the multispectral MNIST dataset during the image augmentation process. Moreover, we also purposely added some noisy data to make the EMD dataset more challenging and to avoid the low variance and high biased (underfitting) issue in our trained model. Figure 6 shows the aggregation view of all ten channels (0–9) for a few noisy image samples. Figure 7 shows the aggregation view of all ten channels (0–9) for a few real image samples.

The EMD dataset is also a step towards forming a generic testbed for all types of multispectral image analysis, which will also be helpful for future research in the multispectral domain. The further details of EMD dataset is presented in Supplementary Material (Supplementary Material.pdf) (Available here).

**4.2. Experimental Criteria and Performance Measures.** Extended multispectral MNIST dataset (EMMD) contains ten (10) individual channels (channel-0 to channel-9). These

individual channels act as individual input pipeline and demonstrate the new spectral band adaptation problem in a better way. To validate adaptability of classification performance of the proposed framework, three (3) experimental queries have been formulated (as shown in Table 5) followed by an inference algorithm. In response to Query 1, Case 1 validates the performance of the proposed framework in stable condition. In response to Query 2, Case 2 and Case 3 validate the performance of the model after observing a new spectral band in the system. Finally, in response to the Query 3, Case 4 validates the ensemble performances in all the cases. Moreover, six (06) different performance measures are used, which are commonly adopted as a performance indicators [37]. The details of the inference algorithm and performance measures are discussed in the Supplementary Material (Available here).

**4.2.1. Platform and Libraries.** The experiments are carried out on the Google Cloud Platform (GCP). In the GCP server (us-west1-b region), we installed the Compute Engine



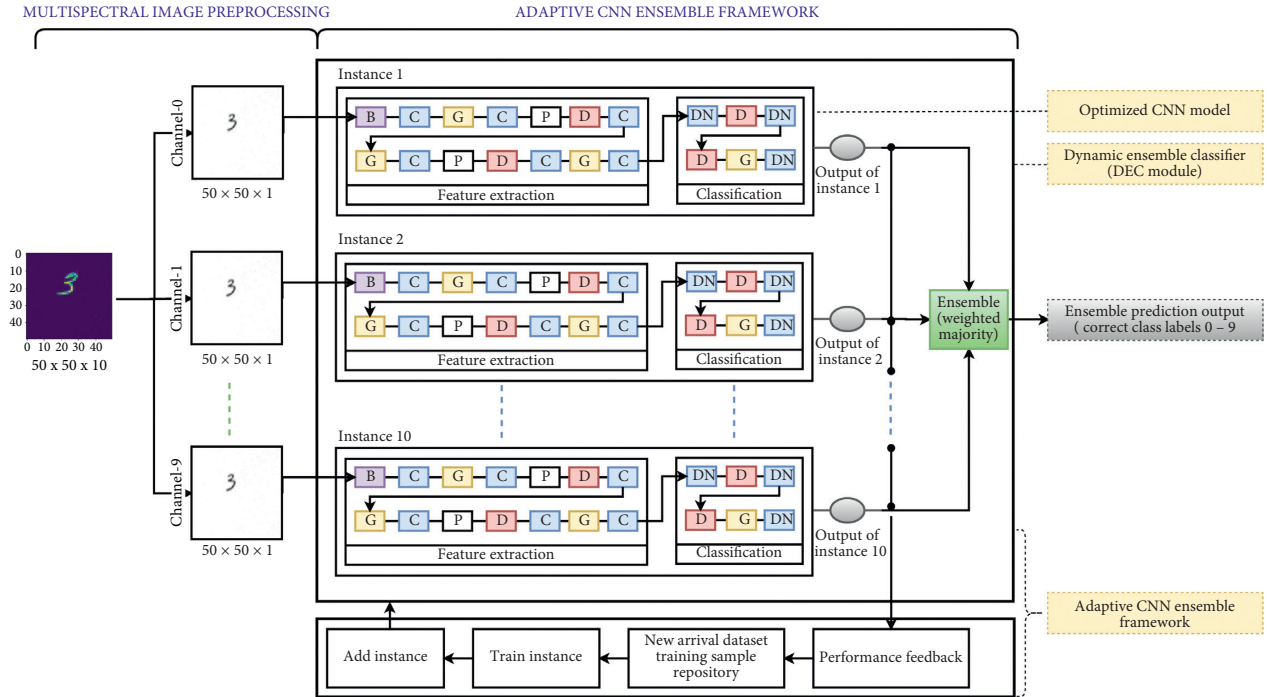


FIGURE 3: Proposed adaptive convolutional neural network (CNN) ensemble framework to adapt new spectral band during online image classification.

**Input:** multiple datasources DS:  $(ds_1, ds_2, \dots, ds_n)$ , input sample  $S_o$  with spectral bands  $B: (b_1, b_2, \dots, b_n)$  at time  $t$ .  $I: (i_1, i_2, \dots, i_n)$  are instances of DEC at time  $t$ , such that  $I$  is trained using respective  $B$ . However,  $B+1: (b_1, b_2, \dots, b_{n+1})$  are the spectral bands at time  $t+1$ .  $M_c (m_{c1}, m_{c2}, \dots, m_{cn})$  are misclassified samples (% accuracy of  $M_c < Th$ ) in training repository.

**Initialization:**  $Th = 50$  (threshold value for performance)

- (1) Counter: 1
- (2) **While** data source  $>$  null and  $B = I$   
     //Valid input data source
- (3) Classify  $(B)$ , via Algorithm 1
- (4) Forward accuracy to the performance feedback module
- (5) Determine the ensemble accuracies using voting, via Algorithm 2  
     //Spectral band stream are classifying with their respective instances in the DEC module
- (6) if % accuracy for  $B \geq Th$       //if sample does not misclassify
- (7)     Repeat steps 3, 4, and 5
- (8) if % accuracy for  $B \leq Th$       //if sample misclassify
- (9)     Save the  $B$       //Save misclassified sample in training repository  
     //as potential new spectral band
- (10) Counter++
- (11) Repeat steps 3, 4, and 5
- (12) if counter = 50      //number of misclassified instances reached to 50
- (13) Cluster  $M_c$  using  $K$ -mean where  $K = 1$ , [35].  
     //Cluster all the misclassified data samples using the  $K$ -means approach with  
     // $K = 1$ ,  $K = 1$  the case is assigned to the class of its nearest neighbor
- (14) Determine optimized centroid, [35]//to optimize the similar sample instances
- (15) Compare cosine distance cluster sample with cluster centroid, via equation (12)  
     //to segregate most relevant samples in cluster
- (16) Assign the all nearest samples a hypothetical class  $X = B_{n+1}$   
     //A new class with additional spectral band information
- (17) Create new instance classifier  $i_{n+1}$   
     //New Single Instance, 20 layered architecture
- (18) Train new instance classifier  $I = i_{n+1}$  with hypothetical class  $X = B_{n+1}$ , via Table 3  
     //Online training with selected hyperparameters as depicted in Table 3

ALGORITHM 3: Continued.

- ```

(19)   Add new instance classifier to DEC,  $I = I + 1$ 
        //Add the new instance in DEC
(20)   Update  $B = B + 1$  //Update the list of spectral band
(21)   Repeat step 3
(22)   End while

```

Output: DEC module with  $(i_{n+1})$  instances, such that  $B + 1$  must equal to  $I + 1$ :  $(i_1, i_2, \dots, i_{n+1})$ , such that  $i_{n+1}$  is trained using  $b_{n+1}$  and performs classification using  $\sum(i_1, i_2, \dots, i_{n+1})$ .

ALGORITHM 3: Adaptive CNN ensemble framework.

TABLE 3: Training hyperparameters (tuning values and optimized values).

| Training hyperparameters             | Tuning values        | Optimized values |
|--------------------------------------|----------------------|------------------|
| Mini-batch size                      | 16, 32, 64, 128, 256 | 120              |
| Learning rate                        | 0.1, 0.01, 0.001     | 0.001            |
| L1 regularization (Lambda parameter) | 0.001, 0.0003        | 0.0003           |
| Number of EPOCH                      | 10–100               | 100              |
| Optimization function                | Adam                 | Adam             |
| Cross-entropy                        | One-hot encoded      | One-hot encoded  |

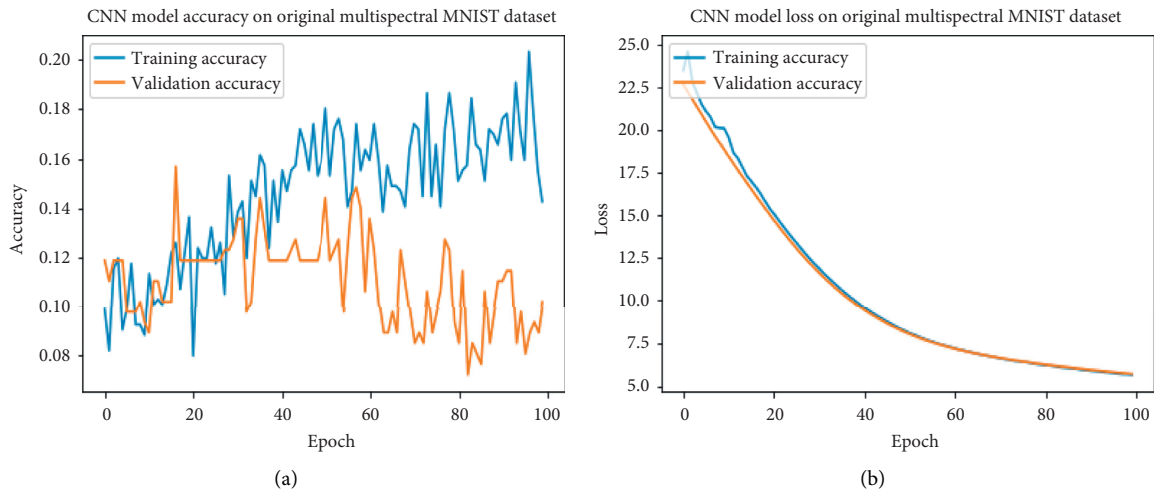


FIGURE 4: Performance of CNN classifier on multispectral MNIST dataset: (a) training and testing accuracy on multispectral MNIST dataset (713 data samples); (b) training and testing loss on multispectral MNIST dataset (713 data samples).

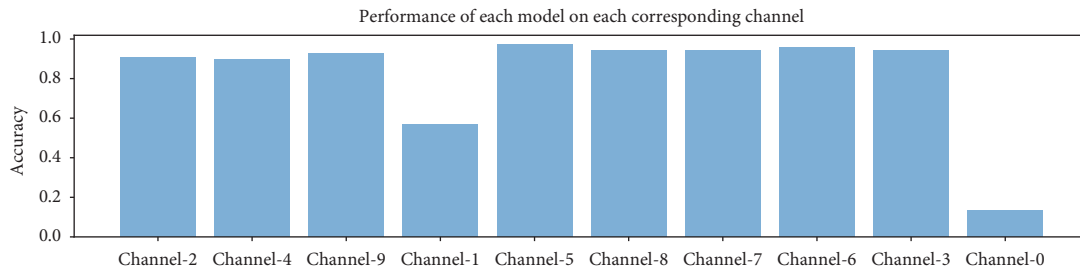


FIGURE 5: The classification accuracies of the CNN model on each corresponding channels on EMD dataset.

Virtual Machine with additional machine learning and deep learning libraries. To speed up the complex computing jobs, we used 16 vCPUs, 104 GB RAM with single NVIDIA GPU Tesla K80. The experiments are implemented using the Python 3 programming language and libraries below:

- (1) Scikit learn library to perform basic machine learning tasks
- (2) OpenCV to perform image processing tasks
- (3) NumPy and Pandas for performing data processing tasks

TABLE 4: Data architecture of multispectral MNIST and extended multispectral MNIST datasets.

| Dataset                      | Image size                 | Feed input               | Output   | No. of samples | Max. testing accuracy (%) |
|------------------------------|----------------------------|--------------------------|----------|----------------|---------------------------|
| Multispectral MNIST          | $350 \times 350 \times 10$ | $50 \times 50 \times 10$ | 10 (0-9) | 713            | 20                        |
| Extended multispectral MNIST | $350 \times 350 \times 10$ | $50 \times 50 \times 10$ | 10 (0-9) | 60,000         | 97                        |

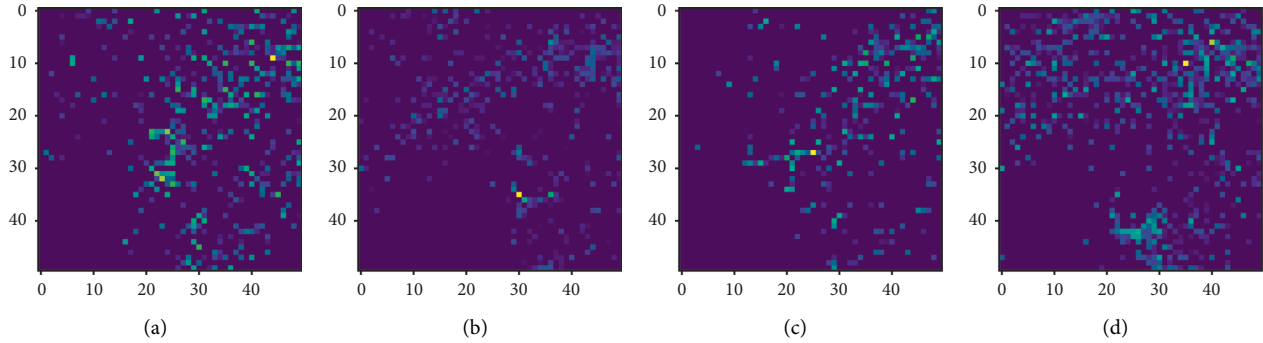
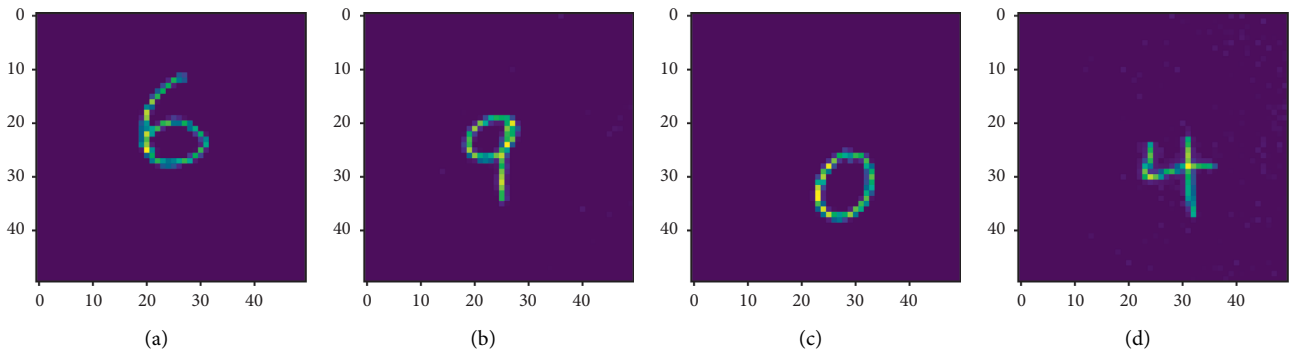
FIGURE 6: The aggregation view of few noisy images for all ten channels (0-9) in the EMDM dataset with image size  $50 \times 50$ : (a) image for digit two (2); (b) image for digit eight (8); (c) image for digit four (4); (d) image for digit zero (0).FIGURE 7: The aggregation view of few images for all ten channels (0-9) in EMDM dataset with image size  $50 \times 50$ : (a) image for digit six (6); (b) image for digit nine (9); (c) image for digit zero (0); (d) image for digit four (4).

TABLE 5: Experimental queries and test case scenarios.

| Experimental queries                                                                             | Test case scenarios                                                                                                                                                                                                                                                                            |
|--------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Q1: how does proposed framework perform in a stable state?                                       | Case 1: three (3) spectral bands tested (channel-1, channel-2 and channel-3) and no new spectral band arrives                                                                                                                                                                                  |
| Q2: what is the effectiveness of the proposed framework to adapt to the new spectral band?       | Case 2: six (6) spectral bands tested (channel-1 to channel-6). Three (03) new spectral bands arrive (channel-4, channel-5, and channel-6)<br>Case 3: all spectral bands tested (channel-0 to channel-9). Four (04) new spectral bands arrive (channel-0, channel-7, channel-8, and channel-9) |
| Q3: does the ensemble performance of proposed framework is increased when having more instances? | Case 4: to perform a comparative analysis on the ensemble performance from case 1, case 2, and case 3                                                                                                                                                                                          |

(4) Matplotlib for result visualization

(5) TensorFlow (1.13) and Keras (as backend) for deep learning tasks

**4.2.2. Training and Testing Dataset Preparation.** To simulate the online scenario (where the multiple channels will arrive at different timestamps), we have separated dataset into its ten (10) channels (0-9). Furthermore, it is essential to report final accuracy on unseen data (to which trained model never

exposed). Therefore, we have divided the dataset (each channel) into training and testing with the ratio 3:1 and applied the cross-validation. However, for testing the model, two hundred thirty-six (236) image samples are randomly generated from the testing dataset.

**4.2.3. Hyperparameter Optimization.** To select hyperparameters for training a model, we used a manual search strategy [38]. Using this strategy, we acquired the optimized

training hyperparameters after various tuning iterations, as shown in Table 3. Here, we also get benefits from the best practices by the research community, for example, selection of optimization function (Adam) and selection of cross-entropy (one-hot encoded).

**4.2.4. Performance Measures.** The classification accuracy is the suitable metric to evaluate the learner performance in the presence of virtual concept drift [39]. Moreover, five (5) different performance measures are used, which are commonly adopted as a classification performance indicator by the research community [37]. These include testing accuracy, testing loss, f1-score, confusion matrix, and classification report.

**4.3. Experimental Results and Discussion.** The details of the experimental results are discussed below.

**4.3.1. Case 1 Results (No Arrival of New Spectral Band).** In Case 1, the acquired testing accuracies and losses are desirable for Model\_2 and Model\_3 (90.67% and 94.4%, respectively). However, for Model\_1, the testing accuracy is 56.70%, which is not satisfactory. Interestingly, the loss of Model\_1 is lesser than Model\_3, which justify the performance of the Model\_1, as shown in Table 6. The argument behind the less accuracy of Model\_1 is noisy data of channel-1 at EMD (refer to Figure 1, in the Supplementary Material (Available here)); the visualization shows noisy (less) data at channel-1). Furthermore, the testing accuracies, losses, and f1-scores are depicted from Figures 8–10.

Figure 11 shows the classification report and confusion matrix of all evaluated models in this case. The detail (precision, recall, and f1-score for each class output) performance of Model\_1 can be observed by classification report. The Model\_1 performed well for predicting the class 0 and did not perform satisfactorily for class output 5 (0.3 precision). The Model\_1 for the class 0 and class 5 did not find optimal relevant instances in a dataset (less recall) and support represents the samples of the true response that lie in that class, as shown in Figure 11(a). However, through confusion matrix of Model\_1, it can be noticed that, among the 236 testing samples, the correct number of predictions are 134, and the false prediction of the Model\_1 is 102, as shown in Figure 11(b).

Similarly, for the Model\_2 and Model\_3, the correct predictions are 215, and 223, respectively, as shown in Figures 11(d) and 11(e). However, Model\_2 predicted 100% correct values for the 0 and 1 classes (1.0 precision) and is able to recall all the required samples for class 6 and class 9, as shown in Figure 11(c).

**4.3.2. Case 2 Results (Arrival of Three (3) New Spectral Bands).** In Case 2, initially, the model is in the state of Case 1. Later, the model has been provided with three (3) more channels, which are channel-4, channel-5, and channel-6. After the arrival of three (03) additional channels, the three newly optimized CNN models initiated and were trained

using the respective channels. Later, the trained models are added in the previous ensemble (DEC module). Each new instance is performed for prediction (the training of these instances was performed offline) for their respective channels. Table 7 shows the comparisons of the performance accuracies and loss for all the participant models. From Table 7, it can be observed that all new instances (Model\_4, Model\_5, and Model\_6) performed well. However, the loss of Model\_5 is exceptionally better (only 0.20). Figures 12–14 visualize the comparison of testing accuracies, testing loss, and f1-score among all participant models.

The detail performance of these models can be observed from the classification report and confusion matrix, as shown in Figure 15. In the classification report, the maximum precision and recall (1.0) can be seen in the Model\_5. In this model, the class six (06) output shows 100% precisions (class outputs 0, 1, 4, 5, 8, and 9) and class five (05) output presents 100% recall (class outputs 0, 1, 3, 6, and 8). However, the remaining classes' precision and recalls' are also significant in the Model\_5, as shown in Figure 15(c). Moreover, the confusion matrix of Model\_5 shows that, among the 236 prediction samples, Model\_5 only made four (04) false predictions (232 predictions are correct), as shown in Figure 15(d). The performance of Model\_4 and Model\_6 is also satisfactory with correct predictions 212 and 226, respectively.

**4.3.3. Case 3 Results (Arrival of Four (4) New Spectral Bands).** Case 3 is in continuation of the Case 2. In case three (03), we have supplied further four (04) new spectral bands (channel-0, channel-7, channel-8, and channel-9) to our model and verified the diversity of our proposed framework. Through the experimental results as shown in Table 8, it can be analyzed that our proposed framework handles the issue of new spectral band adaptation with efficiency, and performance of every model is remarkable. The performance of Model\_0 is not satisfactory, and Model\_0 did not even make accuracy more than 13%. The bad performance by Model\_0 is a similar case of as Model\_1, where the performance issue is due to noisy (less) data which is presented in this channel (refer to Figure 1, in the Supplementary Material (Available here)); the visualization shows no data at channel-0). However, the performance of other new added instances (Model\_7, Model\_8, and Model\_9) is above 92.5%, and loss is much less. Figures 16–18 represent the comparison of testing accuracies, testing losses, and f1-scores among all participants models.

Figure 19 presents the classification report and confusion matrix of newly arrived spectral bands (Model\_0, Model\_7, Model\_8, and Model\_9). It represents the performance details of each newly added model in the DEC module. As discussed earlier, due to noise, the Model\_0 instance performance is far more below than satisfactory. Therefore, the precision, recall, and f1-scores are also not satisfactory for Model\_1, and in most of the classes, the precision and recall are 0%, as shown in Figure 19(a). Similarly, Model\_0 only correctly classified 32 samples out of 236 test samples, as shown in Figure 19(b). On the contrary, the performance of Model\_7, Model\_8, and Model\_9 is exceptionally better as

TABLE 6: Participated models, testing accuracy, and testing loss.

| Models  | Testing accuracy (%) | Testing loss |
|---------|----------------------|--------------|
| Model_1 | 56.70                | 1.94         |
| Model_2 | 90.67                | 1.65         |
| Model_3 | 94.4                 | 2.44         |

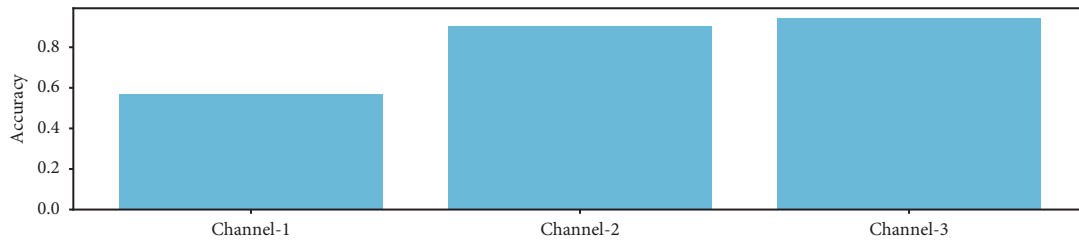


FIGURE 8: Accuracy of Model\_1 to Model\_3 for corresponding channels.

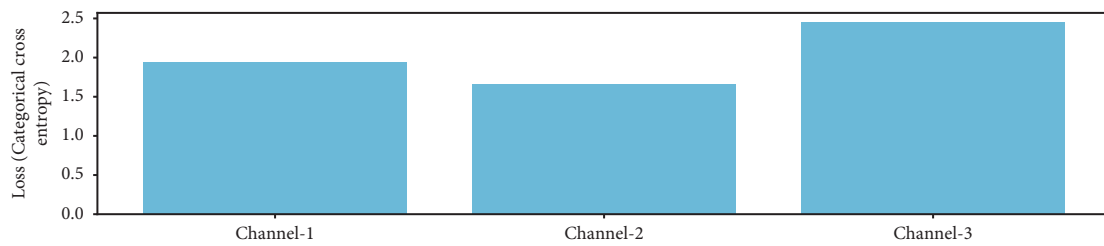


FIGURE 9: Categorical cross-entropy loss of Model\_1 to Model\_3 for corresponding channels.

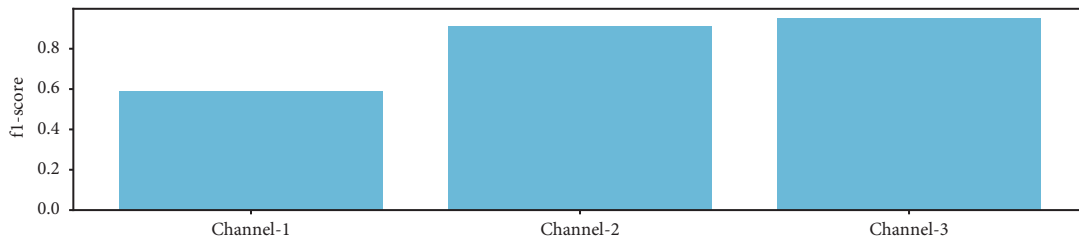


FIGURE 10: F1-score of Model\_1 to Model\_3 for corresponding channels.

reported in classification report with only thirteen (13) false prediction for Model\_7 and Model\_8 and eighteen (18) false prediction for Model\_9, as shown in Figures 19(d), 19(f), and 19(h), respectively. Moreover, for Model\_7, Model\_8, and Model\_9, most of the precisions, recalls, and f1-scores of the classes are more than 90%.

#### 4.3.4. Case 4 Results (Ensemble Performances of All Cases).

The objective of experiment Case 4 is to study the ensemble performance in the new spectral band adaptation scenarios. It is interesting to study the impact of the ensemble over classification accuracies of all three cases (Case 1, Case 2, and Case 3). For example, it is observed that the ensemble performance is increased with a higher number of instances in the ensemble. Moreover, even after the lousy performance of a few incompetent instances in the ensemble, the encouraging results were found. The bad performances of the instances (Model\_0

and Model\_1) are saturated due to the excellent performance of other instances (Model\_2 to Model\_9). The classification accuracy of ensemble Case 3 was recorded at 96.61%, which is maximum in all other cases, as shown in Table 9.

Figure 20 represents the classification report and confusion matrix of all ensemble cases. The individual class wise performance can be observed using the classification reports, from Figures 20(a), 20(c), and 20(e). In ensemble Case 3, six (06) out of ten (10) classes reported 100% precision, recall, and f1-scores (for class outputs 0, 1, 2, 4, 8, and 9) with worst precision reported 88% for class output 3. Similarly, to ensemble Case 3, the worst recall and f1-scores are 82% and 88% for class 5. However, unlike the classification report of ensemble Case 1, the classification report of ensemble Case 2 is near to the ensemble Case 3 performance. Also, the incorrect predictions for ensemble Case 1, ensemble Case 2, and ensemble Case 3 are seventeen (17), nine (09), and eight

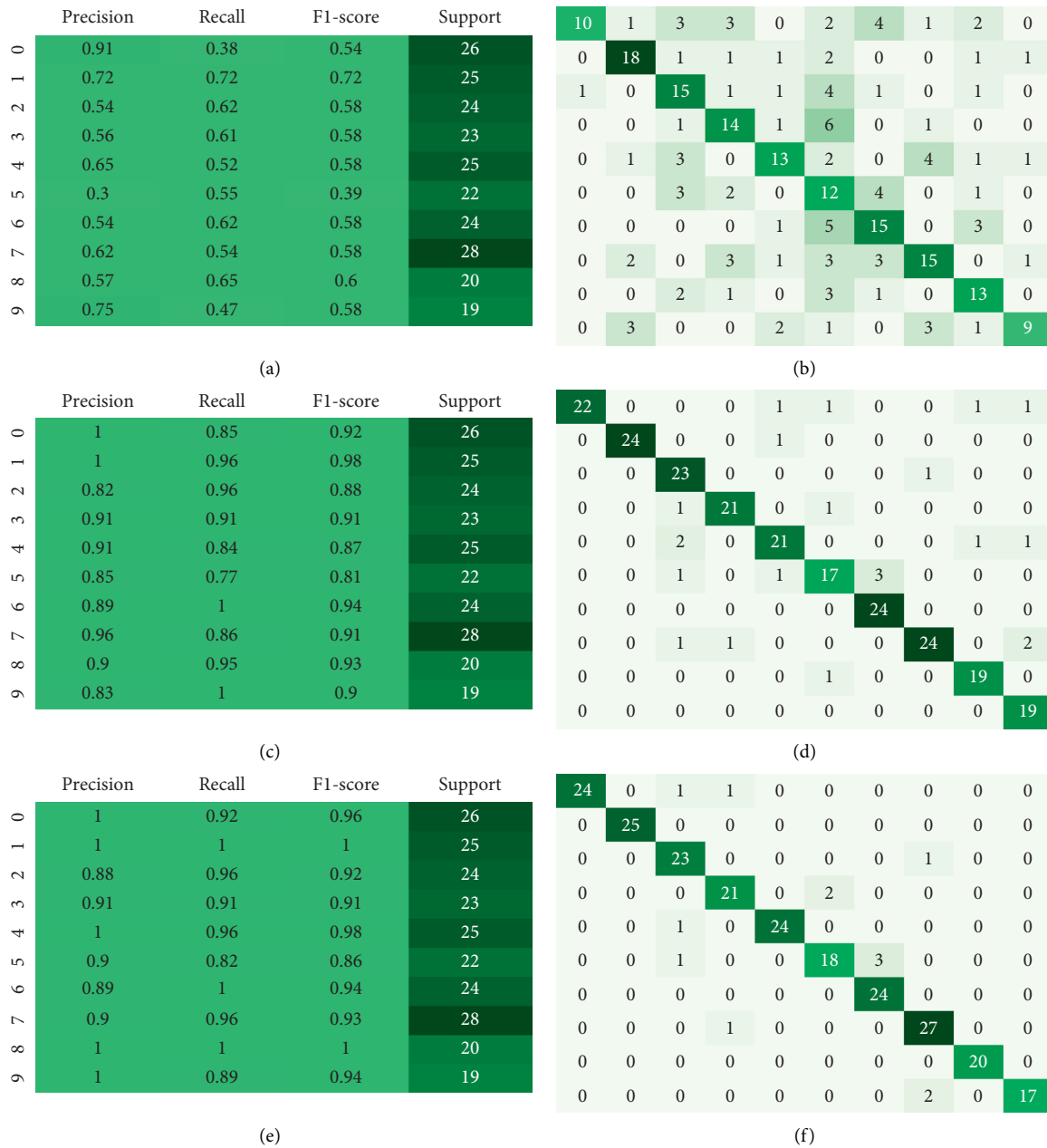


FIGURE 11: The classification report and confusion matrix: (a) classification report of Model\_1; (b) confusion matrix of Model\_1; (c) classification report of Model\_2; (d) confusion matrix of Model\_2; (e) classification report of Model\_3; (f) confusion matrix of Model\_3.

TABLE 7: Participated models, testing accuracy, and testing loss.

| Models  | Testing accuracy (%) | Testing loss |
|---------|----------------------|--------------|
| Model_1 | 56                   | 1.94         |
| Model_2 | 90.60                | 1.65         |
| Model_3 | 94.5                 | 2.44         |
| Model_4 | 89.83                | 2.13         |
| Model_5 | 97                   | 0.26         |
| Model_6 | 95                   | 2.0          |

(08), respectively, which is the evidence of remarkable classification performance, as shown in Figures 20(b), 20(d), and 20(f). To sum, it is worth underlining that the

combination of the diversity technique (with more ensemble instances) is beneficial for performance improvement in adaptive classification models.

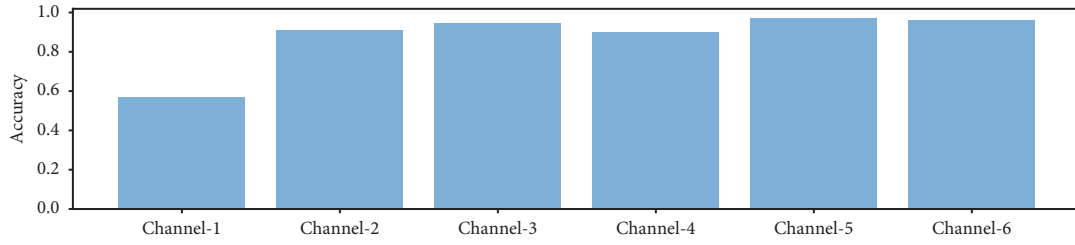


FIGURE 12: Accuracy of Model\_1 to Model\_6 for the corresponding channel.

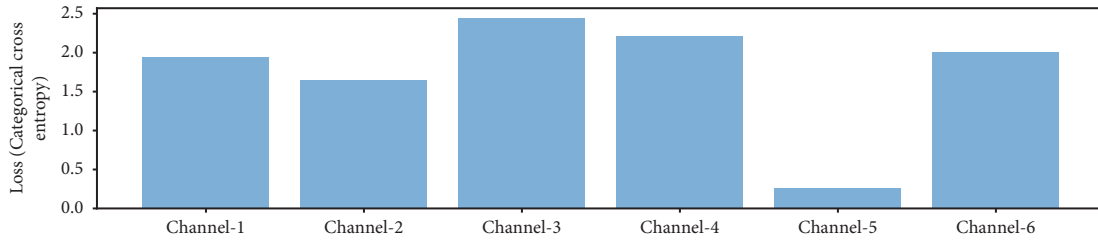


FIGURE 13: Categorical cross-entropy loss of Model\_1 to Model\_6 for corresponding channels.

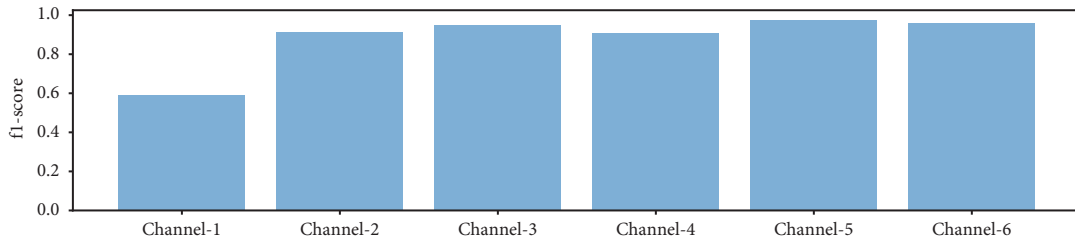


FIGURE 14: F1-score of Model\_1 to Model\_6 for corresponding channels.

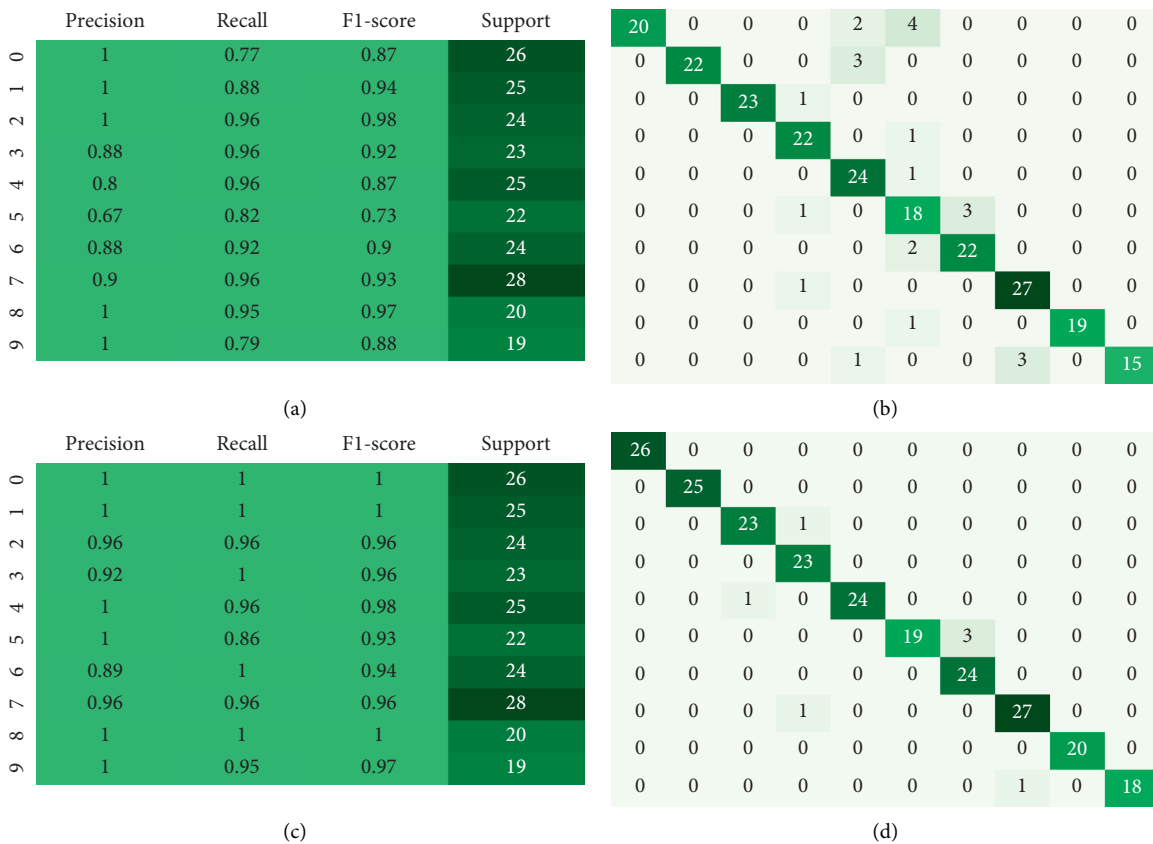


FIGURE 15: Continued.

|   | Precision | Recall | F1-score | Support |
|---|-----------|--------|----------|---------|
| 0 | 1         | 1      | 1        | 26      |
| 1 | 1         | 1      | 1        | 25      |
| 2 | 0.96      | 1      | 0.98     | 24      |
| 3 | 0.91      | 0.91   | 0.91     | 23      |
| 4 | 0.96      | 1      | 0.98     | 25      |
| 5 | 0.9       | 0.82   | 0.86     | 22      |
| 6 | 0.86      | 1      | 0.92     | 24      |
| 7 | 1         | 0.89   | 0.94     | 28      |
| 8 | 1         | 0.95   | 0.97     | 20      |
| 9 | 1         | 1      | 1        | 19      |

(e)

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 26 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 0  | 25 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 0  | 0  | 24 | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 0  | 0  | 0  | 21 | 0  | 2  | 0  | 0  | 0  | 0  |
| 0  | 0  | 0  | 0  | 25 | 0  | 0  | 0  | 0  | 0  |
| 0  | 0  | 1  | 0  | 0  | 18 | 3  | 0  | 0  | 0  |
| 0  | 0  | 0  | 0  | 0  | 0  | 24 | 0  | 0  | 0  |
| 0  | 0  | 0  | 2  | 1  | 0  | 0  | 25 | 0  | 0  |
| 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 19 | 0  |
| 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 19 |

(f)

FIGURE 15: The classification report and confusion matrix: (a) classification report of Model\_4; (b) confusion matrix of Model\_4; (c) classification report of Model\_5; (d) confusion matrix of Model\_5; (e) classification report of Model\_6; (f) confusion matrix of Model\_6.

TABLE 8: Participated models, testing accuracy, and testing loss.

| Models  | Testing accuracy (%) | Testing loss |
|---------|----------------------|--------------|
| Model_0 | 13.5                 | 2.36         |
| Model_1 | 56.7                 | 1.94         |
| Model_2 | 90.60                | 1.65         |
| Model_3 | 94.5                 | 2.44         |
| Model_4 | 89.83                | 2.21         |
| Model_5 | 97                   | 0.26         |
| Model_6 | 95.70                | 2.0          |
| Model_7 | 94.50                | 1.56         |
| Model_8 | 94.49                | 0.81         |
| Model_9 | 92.79                | 0.94         |

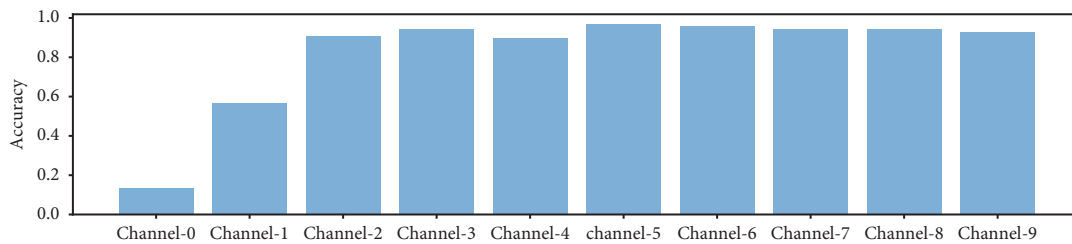


FIGURE 16: Accuracy of Model\_0 to Model\_9 for corresponding channels.

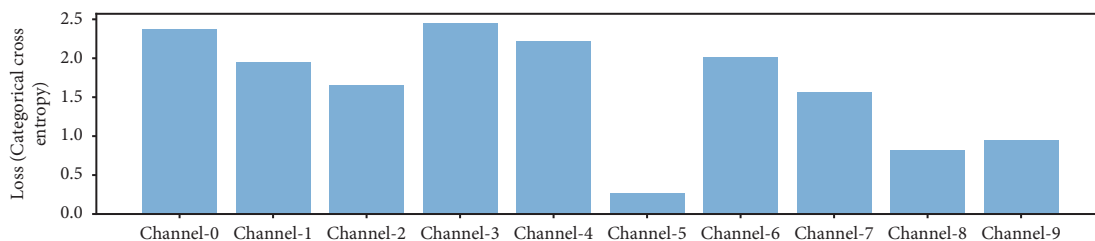


FIGURE 17: Categorical cross-entropy loss of Model\_0 to Model\_9 for corresponding channels.

4.3.5. *Analysis and Deduction on Experimental Queries.* In response to Query 1, we can safely conclude that the performance of the ensemble is satisfactory in stable condition. However, Model\_1 classification is less due to the noisy data in channel\_1, which does not portray the model

deficiency. The accuracy of the Model\_1 can improve if it is trained on clean data.

In response to Query 2, we can safely conclude that the diversity feature of the proposed framework is effective enough to maintain the classification performance of the



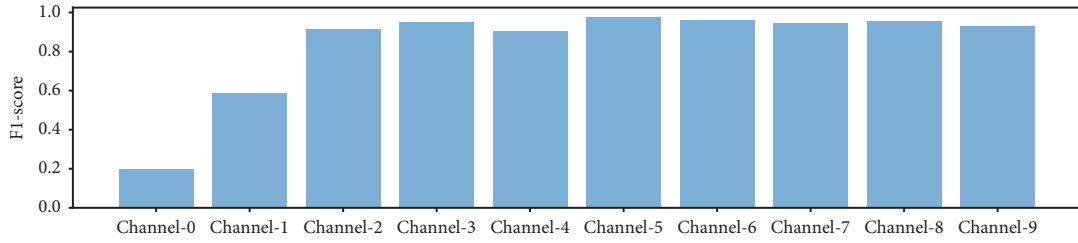


FIGURE 18: F1-score of Model\_0 to Model\_9 for corresponding channels.

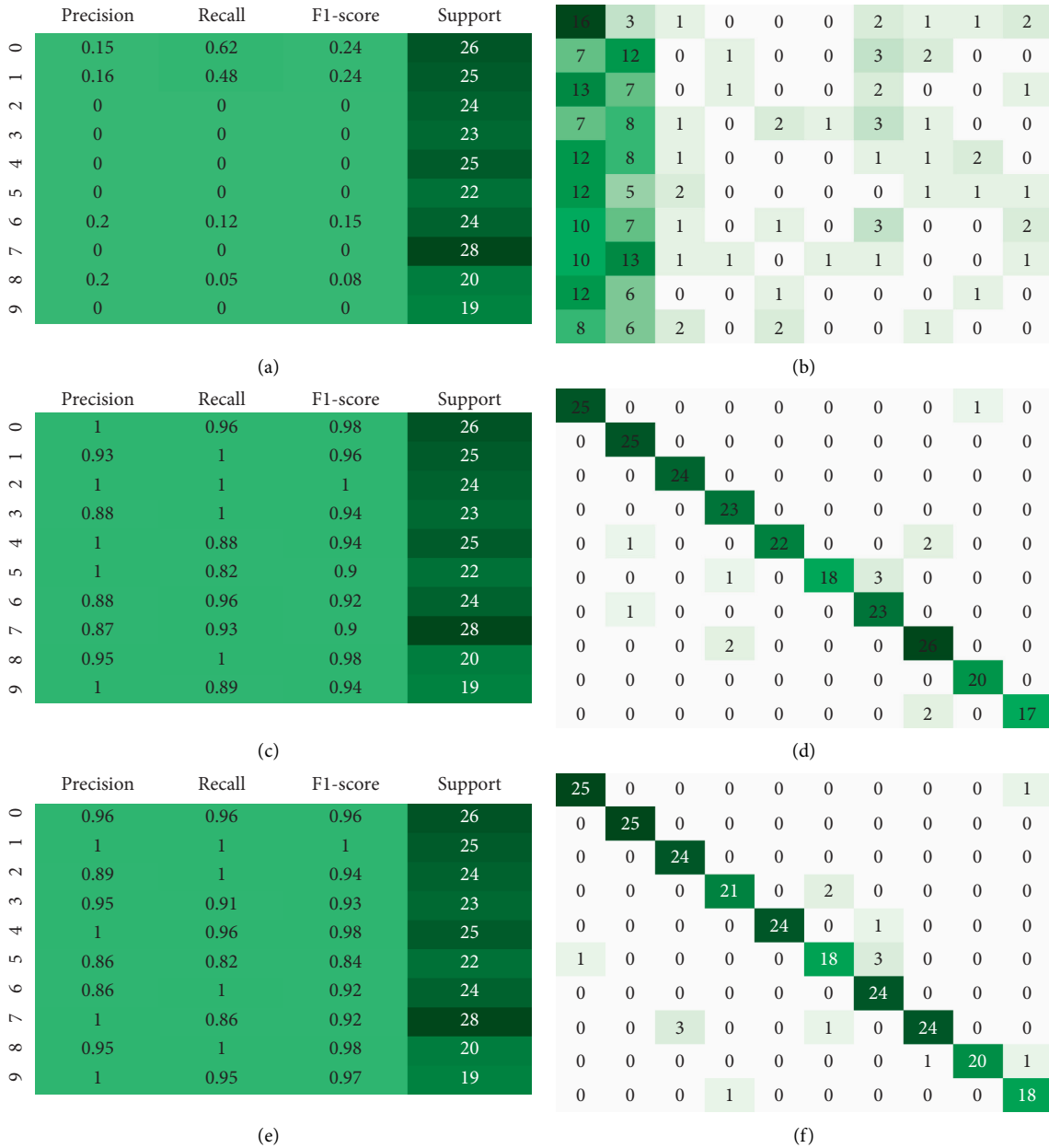


FIGURE 19: Continued.

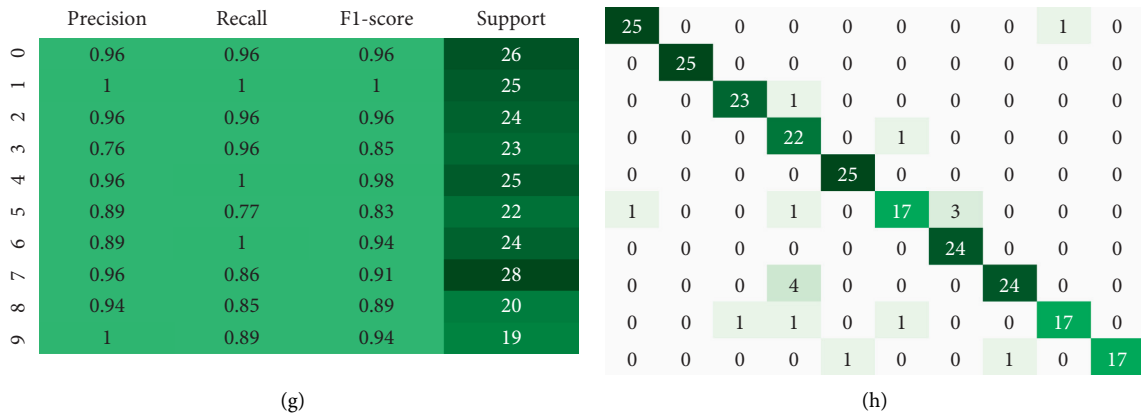


FIGURE 19: The classification report and confusion matrix: (a) classification report of Model\_0; (b) confusion matrix of Model\_0; (c) classification report of Model\_7; (d) confusion matrix of Model\_7; (e) classification report of Model\_8; (f) confusion matrix of Model\_8; (g) classification report of Model\_9; (h) confusion matrix of Model\_9.

TABLE 9: Ensemble testing accuracies of all three (03) experimental cases.

| Testing samples | Ensemble case 1 | Ensemble case 2 | Ensemble case 3 |
|-----------------|-----------------|-----------------|-----------------|
| 236             | 92.79%          | 96.18%          | 96.61%          |

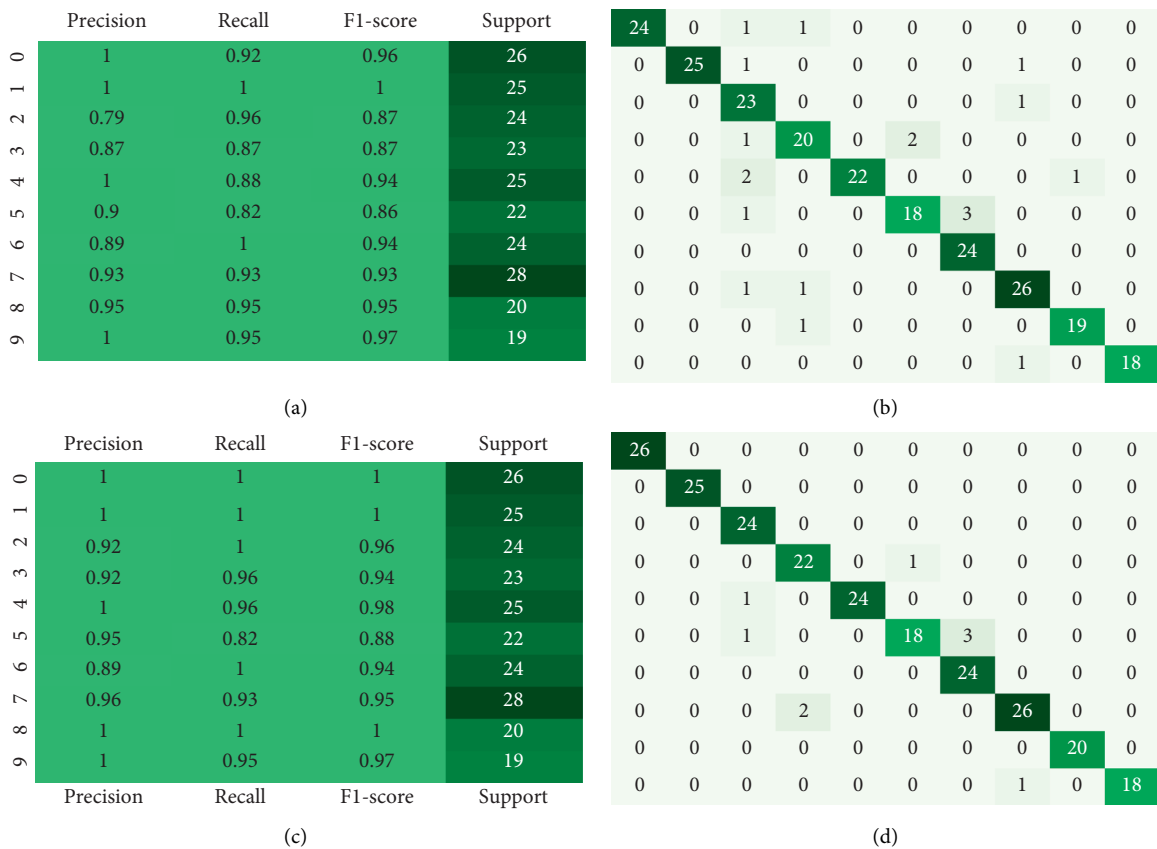


FIGURE 20: Continued.

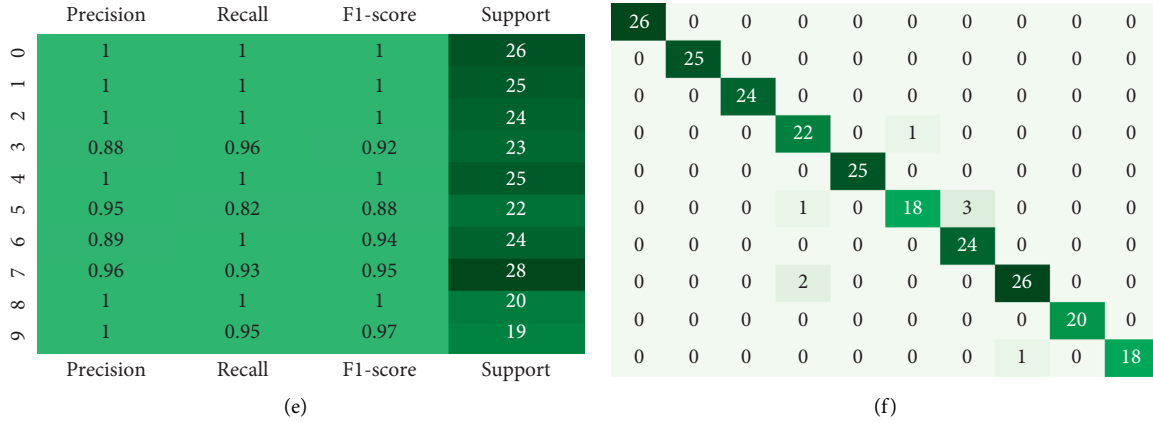


FIGURE 20: The classification report and confusion matrix: (a) classification report of ensemble case 1; (b) confusion matrix of ensemble case 1; (c) classification report of ensemble case 2; (d) confusion matrix of ensemble case 2; (e) classification report of ensemble case 3; (f) confusion matrix of ensemble case 3.

model during the new spectral band adaptation. Interestingly, the simplicity of the proposed ensemble approaches more conveniently tuned model according to the new input structure. The proposed framework classifies the newly arrived input data efficiently, which is desirable.

In response to Query 3, we can safely conclude that the classification performance (in terms of accuracy) is better with more ensemble instances. Therefore, our proposed framework will be more effective when dealing with a higher number of spectral bands, such as hyperspectral imageries. Moreover, the bad performances of few instances can be saturated due to the excellent performance of majority instances in the ensemble.

### 5. Conclusion and Future Work

Multispectral image classification needs to have dynamic data assumptions instead of static data assumptions. This paper formulates the dynamic data assumption scenario, such as the new spectral band arrival issue. Also, it proposes an adaptive CNN ensemble framework as an attempt to handle new spectral band arrival issue. Proposed framework can retain classification accuracy while adapting new spectral band. A novel ensemble approach is proposed to ensure the ensemble diversity and adaptability. In this ensemble approach, the single optimized CNN model handles the single spectral band. This novel approach contributes towards the diversity of the ensemble system in a simple yet effective manner.

Adaptive CNN ensemble framework comprises of five modules. The intuition behind these modules is to explicitly detect and handle new spectral band issue during online classification. Dynamic ensemble classifier (DEC) module is based on multiple optimized CNN models. However, the optimized CNN model is found after several experiments by performing model and training hyperparameter tuning. The novel ensemble approach works under the DEC module. DEC module uses weighted voting ensemble approach to aggregate the classification performances from all individual models/instances. The models/instances in DEC module are subject to increase after adapting the new spectral band.

Performance feedback module detects the arrival of a new spectral band through the misclassified images. Also, we have proposed the idea of online class formation and labelling in the training repository module, which uses the K-mean clustering and cosine distance metrics. The intuition behind the online training module is to perform online training during classification. However, in the initial experiments, we did not find the promising achievements to train the model online, and we performed offline training using the full batch of the respective channels. We consider online training as the future concern of this research.

Our proposed framework found encouraging classification results. The proposed framework not only retained but improved the classification accuracy after adapting new spectral band. Also, the ensemble performances are found more desirable with a large number of ensemble instances. Therefore, the ensemble approach will outperform when dealing with hyperspectral image classification. Interestingly, the large training dataset, proper regularization, optimized model, and training hyperparameters and more appropriate convolutional neural network architecture significantly contributed towards retaining the classification performance after observing and adapting the new spectral band. It is worth underlining that, in the present investigation, the ensemble size, that is, the maximum number of ensemble members, was fixed up to ten (10). Also, the deletion of the existing spectral band was not in the scope. Hence, it is interesting, for future work, to perform a strategy for more the ten (10) ensemble size (for online hyperspectral image classification) and add mechanism in the adaptive CNN ensemble framework for deleting useless spectral bands.

### Data Availability

The prepared and transformed dataset (EMMD) for this experiment can be found at <https://drive.google.com/drive/folders/18Yp-QFJj4rYFQQMfg-sKcMbzfCWbutd>. The multispectral MNIST dataset is available at <https://drive.google.com/drive/folders/1HwAcRdtDSba68u-lMDDqzWB YLjfHlOwJ?usp=sharing>.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Acknowledgments

The authors acknowledge Xiaozhou Wang (Chief Data Scientist and Cofounder at Quartic.ai, Canada) for their contribution in original multispectral MNIST dataset preparation and Douglas Duhaime (Digital Humanities Software Developer at Yale University, USA) for providing their guidance in Training Sample Repository Module. This research study is conducted in Universiti Teknologi PETRONAS (UTP), Malaysia, as a part of the research project “Correlation between Concept Drift Parameters and Performance of Deep Learning Models: Towards Fully Adaptive Deep Learning Models” under Fundamental Research Grant Scheme (FRGS) Ministry of Higher Education (MOHE), Malaysia.

## Supplementary Materials

The trained models are available at . The inference model coding can be found at “Inference-Model-Coding.pdf,” and the training pipeline (with training accuracies) is available at “Training-Pipeline.pdf.” The details of inference algorithm, performance measures, and EMD dataset are available at “Supplementary-Material.pdf.” (*Supplementary Materials*)

## References

- [1] D. Roblyer, R. Richards-Kortum, K. Sokolov et al., “Multi-spectral optical imaging device for in vivo detection of oral neoplasia,” *Journal of Biomedical Optics*, vol. 13, no. 2, Article ID 024019, 2008.
- [2] D. Starodubov, K. McCormick, M. Dellosa, and L. Volfson, “Ship-relative instant multispectral positioning system,” in *Proceedings of the Radar Sensor Technology XXII*, vol. 10633, International Society for Optics and Photonics, Orlando, FL, USA, May 2018.
- [3] M. Gilcher, T. Ruf, C. Emmerling, and T. Udelhoven, “Remote sensing based binary classification of maize. Dealing with residual autocorrelation in sparse sample situations,” *Remote Sensing*, vol. 11, no. 18, p. 2172, 2019.
- [4] C. Toth and G. Józków, “Remote sensing platforms and sensors: a survey,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 115, pp. 22–36, 2016.
- [5] G. Balamurugan, J. Valarmathi, and V. P. S. Naidu, “Survey on UAV navigation in GPS denied environments,” in *Proceedings of the 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, IEEE, Paralakhemundi, India, October 2016.
- [6] K. Małeckı, N. Adam, and P. Forczmański, “Multispectral data acquisition in the assessment of driver’s fatigue,” in *Proceedings of the International Conference on Transport Systems Telematics*, Springer, Katowice-Ustroń, Poland, April 2017.
- [7] J. E. Ball, D. T. Anderson, and C. S. C. , “Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community,” *Journal of Applied Remote Sensing*, vol. 11, no. 4, Article ID 042609, 2017.
- [8] M. Zhao, Y. Zhou, X. Li et al., “Applications of satellite remote sensing of nighttime light observations: advances, challenges, and perspectives,” *Remote Sensing*, vol. 11, no. 17, p. 1971, 2019.
- [9] M. Obitko and V. Jirkovský, “Big data semantics in industry 4.0,” in *Proceedings of the International Conference on Industrial Applications of Holonic and Multi-Agent Systems*, Springer, Valencia, Spain, September 2015.
- [10] S. Sukhanov, D. Budylyskii, I. Tankoyeu, R. Heremans, and C. Debes, “Fusion of LiDAR hyperspectral and RGB data for urban land use and land cover classification,” in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, IEEE, Valencia, Spain*, pp. 3864–3867, July 2018.
- [11] A. L. Brun, A. S. Britto, L. S. Oliveira, F. Enembreck, and R. Sabourin, “A framework for dynamic classifier selection oriented by the classification problem difficulty,” *Pattern Recognition*, vol. 76, pp. 175–190, 2018.
- [12] M. Li, S. Huang, and A. Pizurica, “Robust dynamic classifier selection for remote sensing image classification,” in *Proceedings of the 2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP)*, IEEE, Wuxi, China, July 2019.
- [13] Z. Xie, Y. Chen, D. Lu, G. Li, and E. Chen, “Classification of land cover, forest, and tree species classes with ZiYuan-3 multispectral and stereo data,” *Remote Sensing*, vol. 11, no. 2, p. 164, 2019.
- [14] W. Turner, S. Spector, N. Gardiner, M. Fladeland, E. Sterling, and M. Steininger, “Remote sensing for biodiversity science and conservation,” *Trends in Ecology & Evolution*, vol. 18, no. 6, pp. 306–314, 2003.
- [15] M. A. Wulder, R. J. Hall, N. C. Coops, and S. E. Franklin, “High spatial resolution remotely sensed data for ecosystem characterization,” *BioScience*, vol. 54, no. 6, pp. 511–521, 2004.
- [16] W. B. Verschoof-van der Vaart and K. Lambers, “Learning to look at LiDAR: the use of R-CNN in the automated detection of archaeological objects in LiDAR data from The Netherlands,” *Journal of Computer Applications*, vol. 2, no. 1, p. 10, 2019.
- [17] R. Y. Tsai and T. S. Huang, “Multiframe image restoration and registration,” *Advances in Computer Vision and Image Processing*, vol. 1, pp. 317–339, 1984.
- [18] T. Lu, J. Wang, Y. Zhang, Z. Wang, and J. Jiang, “Satellite image super-resolution via multi-scale residual deep neural network,” *Remote Sensing*, vol. 11, no. 13, p. 1588, 2019.
- [19] G. Jianya, “A review of multi-temporal remote sensing data change detection algorithms,” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 37, no. B7, pp. 757–762, 2008.
- [20] B. Du, “Unsupervised deep Slow feature analysis for change detection in multi-temporal remote sensing images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 12, pp. 9976–9992, 2019.
- [21] X. Cao, Z. Xu, and D. Meng, “Spectral-spatial hyperspectral image classification via robust low-rank feature extraction and markov random field,” *Remote Sensing*, vol. 11, no. 13, p. 1565, 2019.
- [22] M. A. Lefsky and W. B. Cohen, “Selection of remotely sensed data,” in *Remote Sensing of Forest Environments: Concepts and Case Studies*, M. Wulder and S. Franklin, Eds., pp. 13–46, Kluwer Academic, Boston, MA, USA, 2003.
- [23] R. N. Treuhaft, B. E. Law, and G. P. Asner, “Forest attributes from radar interferometric structure and its fusion with optical remote sensing,” *BioScience*, vol. 54, no. 6, pp. 561–571, 2004.
- [24] N. C. Coops and J. D. White, “Modeling forest productivity using data acquired through remote sensing,” in *Remote Sensing of Forest Environments: Concepts and Case Studies*,

- M. Wulder and S. Franklin, Eds., pp. 411–431, Kluwer Academic, Boston, MA, USA, 2003.
- [25] P. Ghamisi, R. Gloaguen, P. M. Atkinson et al., “Multisource and multitemporal data fusion in remote sensing: a comprehensive review of the state of the art,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 1, pp. 6–39, 2019.
- [26] N. Sisodiya, N. Dube, and P. Thakkar, “Next-generation artificial intelligence techniques for satellite data processing,” *Artificial Intelligence Techniques for Satellite Image Analysis*, pp. 235–254, Springer, Cham, Switzerland, 2020.
- [27] S. Ioffe and C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift,” 2015, <https://arxiv.org/abs/1502.03167>.
- [28] A. F. Agarap, “Deep learning using rectified linear units (relu),” 2018, <https://arxiv.org/abs/1803.08375>.
- [29] G. An, “The effects of adding noise during backpropagation training on a generalization performance,” *Neural Computation*, vol. 8, no. 3, pp. 643–674, 1996.
- [30] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning: Adaptive Computation and Machine Learning*, MIT Press, Cambridge, MA, USA, 2016.
- [31] J. Wu, *Introduction to Convolutional Neural Networks*, vol. 5, National Key Lab for Novel Software Technology. Nanjing University, Nanjing, China, 2017.
- [32] N. Srivastava, “Dropout: a simple way to prevent neural networks from overfitting,” *The Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.
- [33] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *Proceedings of the 2017 International Conference on Engineering and Technology (ICET)*, IEEE, Antalya, Turkey, August 2017.
- [34] J. R. Smith, *Integrated spatial and feature image system: retrieval, analysis and compression*, Ph.D. thesis, Columbia University, New York, NY, USA, 1997.
- [35] V. K. Dehariya, S. K. Shrivastava, and R. C. Jain, “Clustering of image data set using  $K$ -means and fuzzy  $K$ -means algorithms,” in *Proceedings of the 2010 International Conference on Computational Intelligence and Communication Networks*, pp. 386–391, Bhopal, India, 2010.
- [36] Y. LeCun, C. Cortes, and C. J. S. Burges, “The MNIST database of handwritten images,” *This Dataset Was Introduced in LeCun*, vol. 10, p. 34, 1998, <http://yann.lecun.com/exdb/mnist>.
- [37] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.
- [38] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.
- [39] J. Gama, R. Sebastião, and P. P. Rodrigues, “On evaluating stream learning algorithms,” *Machine Learning*, vol. 90, no. 3, pp. 317–346, 2013.