Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5539907, 15 pages
https://doi.org/10.1155/2021/5539907

*Research Article*

# Classification of Digital Modulated COVID-19 Images in the Presence of Channel Noise Using 2D Convolutional Neural Networks

**Rahim Khan**,[1] **Qiang Yang**,[1] **Ahsan Bin Tufail**,[1,2] **Alam Noor**,[3] **and Yong-Kui Ma**[1]

[1] *School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China*
[2] *Department of Electrical and Computer Engineering, COMSATS University Islamabad, Sahiwal Campus, Sahiwal, Pakistan*
[3] *CISTER Research Centre, ISEP, Politécnico do Porto, Portugal*

Correspondence should be addressed to Qiang Yang; yq@hit.edu.cn

Received 9 February 2021; Revised 30 May 2021; Accepted 8 June 2021; Published 12 July 2021

Academic Editor: Danfeng Hong

The wireless environment poses a significant challenge to the propagation of signals. Different effects such as multipath scattering, noise, degradation, distortion, attenuation, and fading affect the distribution of signals adversely. Deep learning techniques can be used to differentiate among different modulated signals for reliable detection in a communication system. This study aims at distinguishing COVID-19 disease images that have been modulated by different digital modulation schemes and are then passed through different noise channels and classified using deep learning models. We proposed a comprehensive evaluation of different 2D Convolutional Neural Network (CNN) architectures for the task of multiclass (24-classes) classification of modulated images in the presence of noise and fading. It is used to differentiate between images modulated through Binary Phase Shift Keying, Quadrature Phase Shift Keying, 16- and 64-Quadrature Amplitude Modulation and passed through Additive White Gaussian Noise, Rayleigh, and Rician channels. We obtained mixed results under different settings such as data augmentation, disharmony between batch normalization (BN), and dropout (DO), as well as lack of BN in the network. In this study, we found that the best performing model is a 2D-CNN model using disharmony between BN and DO techniques trained using 10-fold cross-validation (CV) with a small value of DO before softmax and after every convolution and fully connected layer along with BN layers in the presence of data augmentation, while the least performing model is the 2D-CNN model trained using 5-fold CV without augmentation.

## 1. Introduction

An important point to consider in modern wireless communication services is the ability to distinguish between different modulation schemes. Effective communication is essential in bounded network operations. The monitoring of wireless radio communication signals is necessary for the detection, identification, and localization of these signals [1]. The efficient propagation of signals through a wireless channel is of paramount importance to allow the signal energy to be carried optimally. Challenges such as the effects of channel depolarization, intersymbol and co-channel interferences [2], Rician fading [3], and Rayleigh fading channels [2] exist that makes it difficult for the signals to propagate smoothly. The wireless communication system needs to be able to operate in different environments (rural, urban, and suburban), including indoor and outdoor and in all kinds of multipath and time-varying fading channels. Often the transmitter and receiver sources are far away from each other, and a direct line of sight (LOS) path is not possible between them. Hence, the multipath channel is used between the two sources, which is associated with data loss [4].

Modulation schemes are used to provide power and bandwidth-efficient communication satisfying Shannon channel capacity limits and to achieve better efficiency in wireless systems [5]. Single and multicarrier modulation techniques are employed in practice where single or multiple subcarriers carry over information. Modern communication networks are complex and diverse systems, where homogenous and heterogeneous signals coexist as a standard. In such different environments, the detection and recognition of complex signals are necessary to maintain signal fidelity [6].

Deep learning is a computational paradigm that allows models to learn abstractly with applications in images like the classification of objects and regression and model identification for different purposes [7–10], [11, 12]. Zhang et al. and Guo et al. [13, 14] presented a deep learning approach to remove the noise from the images, but it is very necessary to classify the noise-affected patterns to remove those noises. Qiang et al. proposed the Gaussian related spatial-spectral gradient network to remove the mixed noises and the Bayesian posterior deep learning model to remove the non-independent identically distributed noise from the images [15, 16]. There are many studies reported in the literature to study the classification of digital and analog signal schemes deploying deep learning based neural network architectures such as adaptive multistream network incorporating a superposition convolutional unit in each stream [17], adversarial transfer learning architecture [18], polar coordinate approach based network [19], deep neural network consisting of a Convolutional Neural Network (CNN) followed by a long short-term memory network as the classifier which can efficiently explore the temporal and spatial correlations of a signal [20], and exploitation of co-channel signals based on deep learning techniques using a CNN architecture [21]. Different from other works, we deployed novel CNN based architectures on 2D images rather than 1D signals to study multiclass classification problem using COVID-19 lung X-ray samples. Our architectures were designed to study disharmony between Batch Normalization (BN) and dropout (DO) techniques in the presence of data augmentation, to study the impact of different data augmentation techniques such as random rotation, translation, reflection, and shear, without BN and without data augmentation schemes. Higher dimensional signals are known to carry more information and thus can be exploited to achieve better results. To effectively understand the challenges posed by different modulated signals (images modulated by different signals) passed through fading and noisy channel models, we deployed different CNNs to differentiate COVID-19 patients and normal people lungs X-ray images thus solving a multiclass classification problem.

In this paper, our contributions are as follows:

(i) Through development of a dataset with the multiclass (24 classes) modulated images of COVID-19 disease for transmission in Additive White Gaussian Noise(AWGN) and fading channels and classified using deep learning architectures

(ii) We developed a systematic deep learning approach to evaluate the effects of training with a small number of samples for the multiclass classification task

(iii) We estimated our models' competencies on an independent dataset using 5 and 10-fold cross-validation (CV) approaches

(iv) To understand the effect of more data on the classification performance and CNN architectures, we deployed data augmentation methods such as random rotation, translation, reflection, and shear to improve the performance of models

(v) We evaluated the effect of the absence of BN on the classification performance and deployed architectures without it. The architectures are found to have performance bottlenecks such as mean and variance issues in the absence of BN

(vi) Finally, to understand the "variance shift" phenomenon associated with the disharmony (DH) between the BN and DO techniques as mentioned in [22], we deployed a configuration with a small value of DO before softmax and after every convolution and fully connected layer along with BN layers in the presence of data augmentation

The rest of the article is organized as follows. Section 2 presents related work and the mathematical formulation of the modulation schemes, CNNs, AWGN, Rayleigh, and Rician channels. In the methodology Section 3, we present the details of the datasets used in the experiments as well as 2D-CNN architectures for the novel and consistent achievement of the results. Section 4 presents the experimental results of the paper. Section 5 presents the discussion followed by a conclusion in Section 6.

## 2. Related Work

CNNs are believed to learn equivariance, invariance, and equivalence properties [23] effectively. Spatial transformation methods such as per-pixel flow, mean blur, and differentiable bilinear interpolation can also be used to deform the input images benefitting from visual recognition tasks [24].

CNNs are already translation equivariant; that is, small input image translations produce proportionate changes in feature maps, which is not the case for rotations [25]. Aggressive data augmentation helps in improving the performance of translationally variant systems [7]. In response to manually generated perturbations to the input, such as image transformations, a quantitative approach towards analyzing networks measures output changes [26]. However, neither the architectural changes nor the data augmentation may help in achieving the desired invariance [27]. Deformations such as pose, affine transformations such as translation, scaling, rotation, or shear, as well as optical flow, are commonly used for object recognition tasks [28]. Colour information instead of a grayscale image may also improve prediction performance [29]. Visualization of CNN representations is a promising way to explore network representations. It provides a technical foundation for many approaches of CNN representations [30].

A refined invariant representation is a typical image constructed with a cascade of invariants, which retains translation, rotation, skin, and shear information [31]. CNN mainly depends upon satisfying the requirements laid down by the Nyquist sampling theorem. While this will not completely restore rotational equivalence, it shows that the aliasing introduced through the downsampling is significantly reduced [32]. CNNs deal with shift variance far better than scale invariance. At the same time, invariance helps in building robust input transformations through regularization in the network [33, 34]. Furthermore, dataset bias is a major hurdle for the generalization of CNN to the real world and has applications in recognition and detection tasks [35, 36]. Bruna and Mallat presented the invariant scattering technique for the CNNs to reduce the variabilities such as rigid translations, rotations, or scaling as well as nonrigid deformations [37].

Sohn and Lee studied the transformation equivariant architectures trying to infer the best matching filters by transforming them using linear transformation matrices to learn locally invariant features that can be useful in classification tasks [38]. While Ruderman et al. find that the deformation of networks with pooling increases significantly throughout the training process [39]. Bruna et al. stated the relationship between group invariances in CNNs providing an understanding of their classification function performance and explaining why the weight sharing caused by convolutions in the presence of a deformations group is an authentic regularization method [40].

Ngiam et al. show that the sparsity of lifetime is accomplished when the feature is selective and permits examples to be found. High dispersal is achieved for a particular row of features when the distribution has similar statistics for all rows [41]. Cohen and Welling proposed transformation properties of learned visual representations, an invariant CNN group that could be used to develop a scalable representation learning system [42]. Bengio et al. presented the representation learning for complex real-world distributions [43].

When a complex neural network is trained on a small training set, it usually performs poorly on a held-out test set that can be mitigated by a random omission of feature detectors. Overfitting can be reduced by using the DO technique [44]. Methods for applying DO to CNN layers as well as to recurrent neural networks are reported consistently in the literature [45].

The internal covariance shift is a significant problem when training deep networks. BN mitigates this problem by normalizing each training minibatch. Eliminating the internal covariance shift also speeds up the training of deep networks. BN may lead the layer jacobians to have singular values close to identity, which is known to be beneficial for learning. Training without DO but with BN is also a promising approach for achieving higher prediction accuracy but has different train-test calculations [22, 46].

Hong et al. worked on deep learning-based methods such as Graph Convolutional Networks (GCNs) and CNNs which are fused together for hyperspectral image classification tasks [47], for the classification and identification of the materials lying over or beneath the earth's surface by designing a multimodal deep learning framework [48], to address spectral variability [49], for feature extraction of hyperspectral images [50] and semisupervised transfer learning with limited cross-modality data in remote sensing [51].

*2.1. Theoretical Analysis.* This section includes a brief mathematical formulation of the CNNs, modulation schemes including the Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK), 16-Quadrature Amplitude Modulation (16-QAM), and 64-QAM, and AWGN, Rayleigh, and Rician channels.

Modulation is fundamental to all wireless communication systems and the technique of impressing the data to be transmitted on a high-frequency carrier. The objective is to achieve spectral efficiency by squeezing more data into the least amount of spectrum possible. The issues related to interference, hardware, and noise are quickly reduced in the digitally modulated systems as it resists noise and interference and offers bandwidth efficiency, in comparison to the analog modulated systems which need higher bandwidth to transfer symbols. We used BPSK, QPSK, 16-QAM, and 64-QAM digital modulation schemes in this study.

Mathematically, BPSK signal generation can be expressed as

$$Z_m(t) = \sqrt{\frac{2E_b}{T_b}} \cos\left(2\pi f_c t + \pi(1-m)\right), m = 0, 1. \quad (1)$$

Here, $E_b$is energy per bit, $T_b$is bit duration, $\sqrt{2E_b/T_b}$ is the amplitude, $f_c$is carrier signal frequency, and $t$ is the time. The BPSK signal transmits one bit per symbol and is mapped to one of two possible phase states, 0 and $\pi$.

Mathematically, QPSK signal generation can be expressed as

$$Z_m(t) = \sqrt{\frac{2E_s}{T_s}} \cos\left(2\pi f_c t + (2m-1)\frac{\pi}{4}\right), m = 1, 2, 3, 4. \quad (2)$$

Here, $E_s$ is energy per symbol, $T_s$ is symbol duration, $\sqrt{2E_s/T_s}$ is the amplitude,$f_c$is carrier signal frequency, and $t$ is the time. The QPSK signal transmits two bits per symbol and is mapped to one of the four possible phase states, $7\pi/4$, $5\pi/4$, $3\pi/4$, and $\pi/4$ .

The complex envelope of the transmitted waveform with QAM can be written as

$$\tilde{Z}(t) = A\sum_n c(t - nT, \mathbf{X}_n), \quad (3)$$

where $c(t - nT, \mathbf{X}_n) = x_n u_a(t)$, $u_a(t)$ is the amplitude shaping wave, and $x_n = x_{I,n} + jx_{Q,n}$ is the complex-valued data symbol that is transmitted at baud rate $n$.

With noise variance $\sigma^2$ and power constraint $P$, the capacity of the (real) AWGN channel is

$$\mathrm{C}_{\mathrm{AWGN}} = \frac{1}{2}\log\left(1 + \frac{P}{\sigma^2}\right). \quad (4)$$

For the optimal rate adaptation to channel fading with a constant transmitting power, bandwidth B and signal to noise ratio $\gamma$, the Rayleigh channel capacity can be expressed as

$$C_{\text{Rayleigh}} = B\int_0^{\infty} \log_2(1+\gamma)p_{\gamma}(\gamma)d\gamma. \quad (5)$$

The Rician fading channel capacity for an indeterminate number of transmitting/receiving antennas can be written as

$$C_{\text{Rician}} = \int_{\lambda_1,\cdots,\lambda_k} \sum_{i=1}^{k} \log\left(1+\frac{P}{N_T}\lambda_i\right) f_{\lambda_1,\cdots,\lambda_k}(\lambda_1,\cdots,\lambda_k)d\lambda_1 \cdots d\lambda_k. \quad (6)$$

Here, $P$ represents an upper-bound on the total average power, $N_T$ is the number of transmit antennas, and $\lambda$ is any unordered eigenvalue of the noncentral Wishart distributed random matrix.

CNNs are a specialized form of neural networks common with known topology for the processing of data. They use a mathematical operation known as convolution which can be defined as

$$s(t) = \int x(a)w(t-a)da = x(x * w)t. \quad (7)$$

Here, $x$ is the input while $w$ is known as the kernel. The output is widely known in the literature as a feature map. Usually, we use discrete rather than the continuous version of equation (7) defined as

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a). \quad (8)$$

In machine learning applications, the input and the kernel are usually tensors. In the case of two-dimensional input and kernel, the convolution operation can be expressed as

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m,j-n). \quad (9)$$

Usually, the convolution operation is implemented as cross-correlation in the neural network software which is defined as

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m,j+n)K(m,n). \quad (10)$$

## 3. Methodology

*3.1. Dataset and Preprocessing.* We downloaded (https://www.kaggle.com/nabeelsajid917/covid-19-x-ray-10000-images and https://github.com/ieee8023/covid-chestxray-dataset/tree/master/images) a random dataset of images of COVID-19 patients and normal people lungs X-rays from the internet. Sample images are shown in Figure 1. These images are, then, modulated by BPSK, QPSK, 16-QAM, and 64-QAM schemes and passed through AWGN, Rayleigh, and Rician fading channels. We used five and 10-fold CV procedures. The modulated images dataset description for 10-fold and 5-fold CV is given in Tables 1 and 2, respectively. We build datasets to study the 24 class classification problem.

We performed a series of experiments for the multiclass classification to study the impact of channel fading and noise on the different modulation schemes. In this study, we used BPSK, QPSK, 16-QAM, and 64-QAM representation of the images of COVID-19 patients and normal people lungs X-rays after passing them through AWGN, Rayleigh, and Rician channels as shown in Figure 2. The workflow of the proposed approach is illustrated in Figure 3.

The dataset is a crucial part before training a CNN. The raw datasets have been pre-processed and are then passed through a deep learning algorithm for the multiclass classification task. Normalization ensures a uniform shape of image during image processing to resize and sharpen the image. The preprocessed images are modulated by BPSK, QPSK, 16-QAM, and 64-QAM signals and are then passed through AWGN, Rayleigh, and Rician fading channels to add the effect of the channel. The modulated images are distorted and attenuated by channel effects. The images are then divided into training, validation, and testing sets, and then, training and validation sets are passed through the data augmentation techniques. Data augmentation is a powerful method to prevent overfitting and generates additional training and validation data from the smaller existing datasets. Thereafter, the data augmentation process is used to produce new images for the training of COVID-19 patients and normal people lung X-ray images. Practical data augmentation techniques, including translation, rotation, reflection, shearing, flipping, and so on, are the easiest way of generating new data. We have used random rotation, translation, scaling, reflection, and shear. Finally, augmented data are fed to different 2D-CNN architectures for multiclass (24 classes) classification of modulated images.

*3.1.1. Data Augmentation Techniques.* Big datasets are extremely expensive and are vital to the deep learning model's performance, whereas small datasets overfit during the training process. Pre-trained models are vulnerable to new invisible data and thus may not help in the generalization of the validation set. Data augmentation is used in deep learning models to solve the overfitting problem due to limited data. Data augmentation is a good approach for building better datasets. In general, overfitting does not pose a problem with significant data access. A massive amount of data is required in the training of a deep learning model. It is a difficult task to collect so much amount of data so data augmentation is employed, and the data already present is transformed. It increases the dataset size and adds variability to the dataset. A further enhancement is still required to generalize the efficiency of deep learning models. By using data augmentation, generalization performance can be enhanced. These augmentations usually take the form of geometric or colour augmentations for input images in image processing, which have proven extremely successful in reducing CNN overfitting.

In this work, experiments have been performed to study multiclass (24-classes) classification problems (1) without

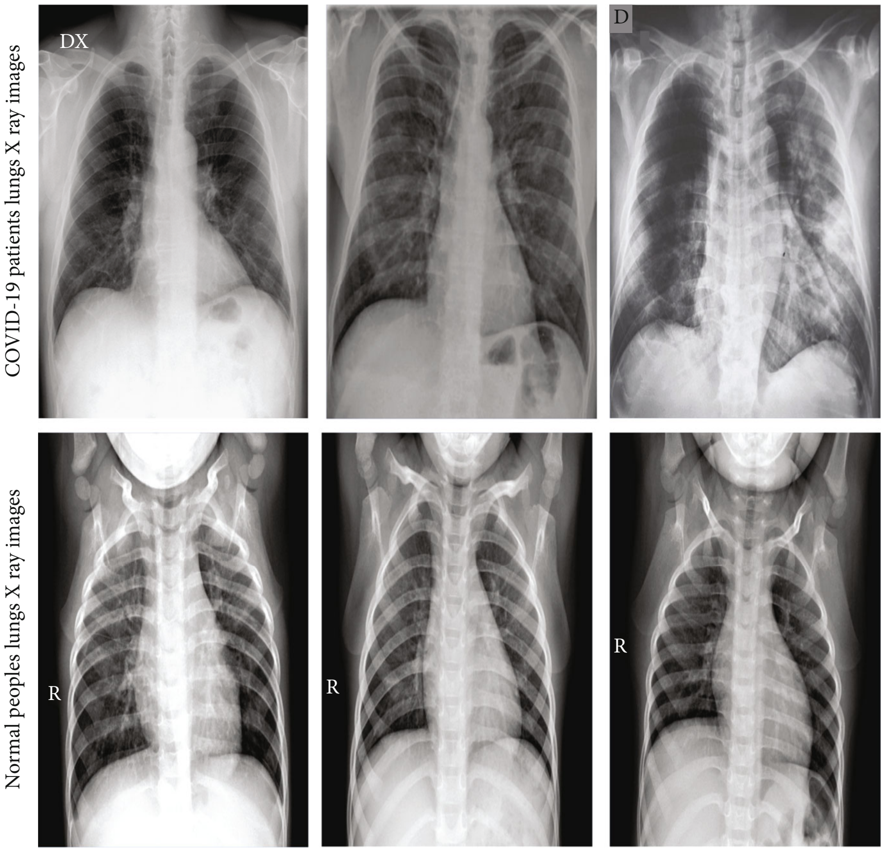

COVID-19 patients lungs X ray images
DX
D
Normal peoples lungs X ray images
R
R
R

Figure 1: Sample of COVID-19 patient's lungs X-ray and normal people lungs X-ray images.

Table 1: COVID-19 and normal lungs X-ray images dataset statistical approach using 10-fold CV.

| Categories | Training set | Validation set | Testing set | Total |
|---|---|---|---|---|
| Number of images | 10368 | 1152 | 480 | 12000 |
| Percentage (%) | 86.4 | 9.6 | 4 | 100 |
| Number of COVID-19 Lungs X-ray images | 5184 | 576 | 240 | 6000 |
| Number of normal Lungs X-ray images | 5184 | 576 | 240 | 6000 |

data augmentation, (2) with data augmentation, (3) absence of BN in the CNN-based classification architectures, and (4) to understand disharmony between BN and DO techniques in which we used a small value of DO before softmax and after every convolution and fully connected layer.

We deployed 5-fold and 10-fold CV [52] approaches to select the optimal set of hyperparameters such as filter size, strides, and DO probabilities. We used grayscale images of size 297 × 167 × 1. The intensity values of images that were inputted into the classifiers were in the range 0-255, and sample modulated images are shown in Figure 2.

### 3.2. Methods

*3.2.1. 2D Convolutional Neural Networks.* The 2D-CNN architectures for this experiment are shown in Figure 4. In the architecture shown in Figure 4, we used zero center normalization to center the data around the origin. Seven convolutional layers have been used to extract features using a filter of size $3 \times 3$ and stride $1 \times 1$ where the number of feature maps varied from 8 to 96. Each convolutional layer has been followed either by a BN layer or not and an Exponential Linear Unit (ELU) nonlinearity activation function with an $\alpha$ value of 1. BN is used for reducing the mean and variance problems and has been employed before ELU nonlinear activation to speed up the training process and to conform to the commonly used practices [52]. Disharmony between DO and BN are contradictory neuronal variances behaviours during the transitioning process of the networks. The deduction of "differential changes" observed in contemporary network bottleneck blocks and finding a sufficient explanation for this confusion between DO and BN has been discussed in the literature [22]. After every nonlinear activation layer, the max pooling layer has been adopted to reduce the number of feature maps. Three dense layers with global averaging with ELU activation function are applied to connect the information extracted by the fully convolutional layers. The dense layer before the softmax classification layer has 24 neurons that are aimed at solving the 24-classes classification problem.

Table 2: COVID-19 and normal lungs X-ray images dataset statistical approach using 5-fold CV.

| Categories | Training set | Validation set | Testing set | Total |
|---|---|---|---|---|
| Number of images | 9216 | 2304 | 480 | 12000 |
| Percentage (%) | 76.48 | 19.2 | 4 | 100 |
| Number of COVID-19 lungs X-ray images | 4608 | 1152 | 240 | 6000 |
| Number of normal lungs X-ray images | 4608 | 1152 | 240 | 6000 |

Three different architectures have been adopted during training. The first one is without BN with employed max pooling after every two convolutional layers, which is given on the top of Figure 4. The second one is to study the data augmentation effect for higher precision during training is shown in the middle of Figure 4. Finally, to study the effect of disharmony (between BN with DO techniques) for best training accuracy is given at the bottom of Figure 4.

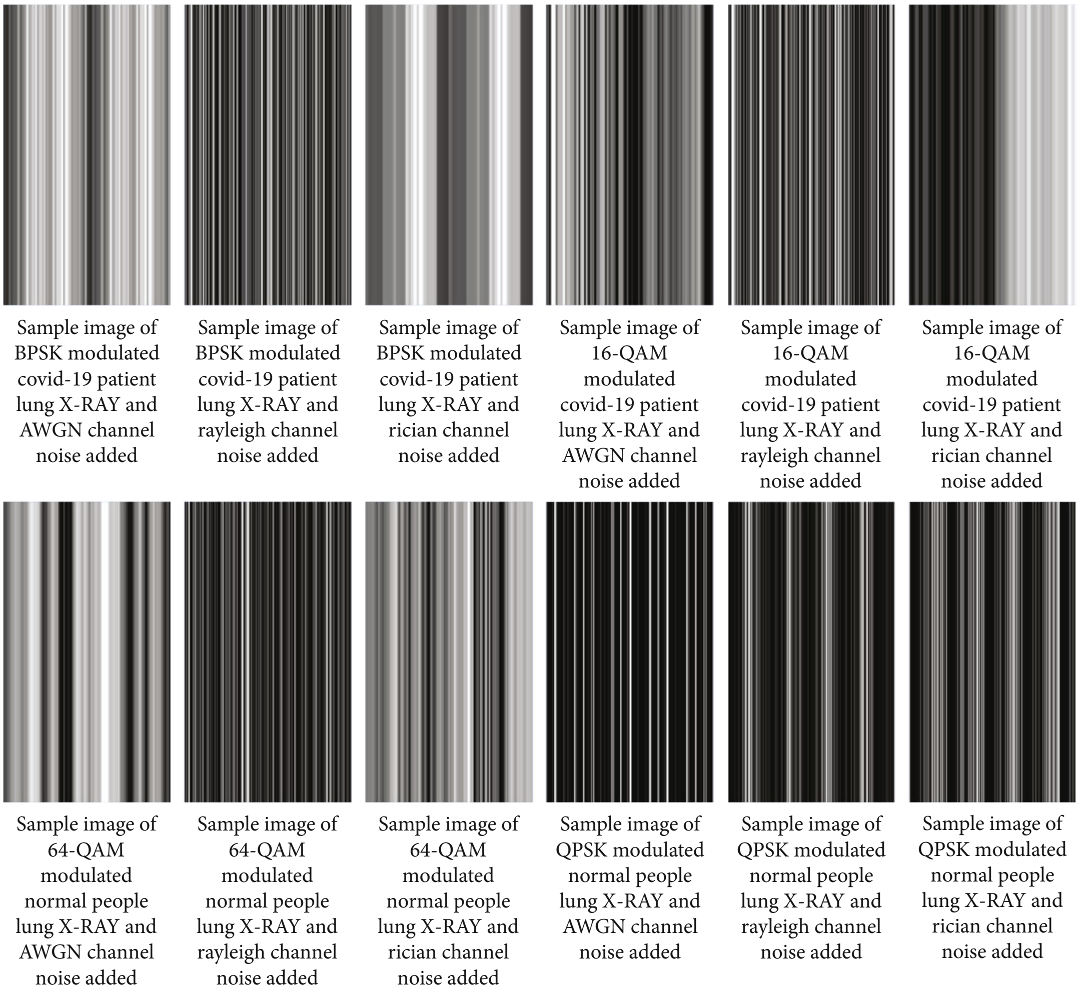
The first architecture without BN has an input of size 297 × 167 × 1 and employed a zero-center normalization procedure. After this input layer, there is a block employing a 2D convolutional layer with filter size 3 × 3 and stride 1 followed by an ELU activation layer with an alpha value of 1 followed by a 2D convolutional layer with filter size 3 × 3 and stride 1 followed by an ELU activation layer with an alpha value of 1 followed by a 2D max pooling layer with filter size 2 × 2 and stride 2. This block is repeated 3 times such that the number of feature maps in the convolutional layers are 8, 16, 32, 48, 64, and 80, respectively. After that there is a convolutional layer with filter size 3 × 3, a number of feature maps equal 96, with stride size 1 followed by an ELU activation layer with an alpha value of 1 followed by a 2D max pooling layer with filter size 2 × 2 and stride 2 followed by a DO layer with ratio 40% followed by 3 dense layers with the number of neurons equal to 100, 50, and 24, respectively, followed by a global average pooling layer with an ELU activation function, followed by a softmax probability layer and a classification layer.

The second architecture with and without data augmentation has an input of size 297 × 167 × 1 and employed zero-center normalization procedure. After this input layer, there is a block employing a 2D convolutional layer with filter size 3 × 3 and stride 1 followed by a BN layer followed by an ELU activation layer with an alpha value of 1 followed by a 2D max pooling layer with filter size 2 × 2 and stride 2. This block is repeated 7 times such that the number of feature maps in the convolutional layers are 8, 16, 32, 48, 64, 80, and 96, respectively. Finally, there is a DO layer with a probability of 50%, followed by three dense layers with 100, 50, and 24 neurons each, followed by a global average pooling layer with an ELU activation function, followed by a softmax layer and a classification layer.

The third architecture that is designed to study the disharmony between BN and DO techniques has an input layer with size 297 × 167 × 1 and employed a zero-center normalization procedure. After this layer, there is a block employing a 2D convolutional layer with filter size 3 × 3 and stride 1 followed by a BN layer followed by an ELU activation layer with an alpha size of 1 followed by a 2D max pooling layer with filter size 2 × 2 and stride 2 followed by a DO layer with a ratio of 10%. This block is repeated 7 times such that the number of feature maps in the convolutional layers are 8, 16, 32, 48, 64, 80, and 96, respectively. After that, there are 3 dense or fully connected layers with a number of neurons equal to 100, 50, and 24, respectively, followed by a global average pooling layer with an ELU activation function, followed by a softmax probability layer and a classification layer. After every dense layer, there is a DO layer with a ratio of 10%.

*3.2.2. Effect of Batch Normalization.* Note that we used a simplified architecture in comparison to the previous architectures to speed up the training process as the removal of BN layers slows down the network training by a significant margin. This architecture took the most amount of time to run. The architecture without BN can slow the training cycle as shown at top of Figure 4, and also, the model consistency is disrupted by means and variance issues. It reduces the sum by the covariance of the hidden unit values. BN allows each network layer, separate from other layers, to learn by itself. It adds a little noise to activations in the hidden layers. It is important to use less DO if BN is used, because a lot of data is lost with a higher DO ratio. Nonetheless, even BN is not the last hope, it is better to use it with DO. BN norms the output of previous activation layers by subtracting the batch mean and dividing by the batch standard deviation, thus enhancing neural network stability. But the weights in the next layer are no longer suitable after changing activation outputs with other arbitrarily initialized parameters. Adam optimizer reverses this normalization because it is a way to reduce the loss function. BN adds two trainable parameters to each layer, then multiplying the standard output by the parameter “standard deviation” (gamma) and adding the parameter “means” (beta). BN allows Adam to denormalize by adjusting these beta and gamma weights for each activation instead of sacrificing all the network weights. But still, the CNNs are unstable due to the inconsistency of variance shifts.

*3.2.3. Effect of Disharmony between Batch Normalization and Dropout Techniques.* Overfitting and long training time are two significant issues in multi-layered neural network training, especially in deep learning. Two well-known approaches to addressing these issues are DO and BN. DO works because the mechanism produces many implicit sets of weight sharing. The concept is that one randomly removes neurons for each training set. Indeed, one has a first neural net subset that runs inferences and updates its weights. To accomplish the classification, you have more neural networks, which work as ensembles. BN works by normalizing the inputs dynamically per minibatch. A study shows that the effect is much quicker to learn without a loss of generalization when removing DO while using BN. One of the benefits of DO is that it can reduce mutual information quadratically, and the correlation between any neuron pair about the DO layer



Sample image of BPSK modulated covid-19 patient lung X-RAY and AWGN channel noise added

Sample image of BPSK modulated covid-19 patient lung X-RAY and rayleigh channel noise added

Sample image of BPSK modulated covid-19 patient lung X-RAY and rician channel noise added

Sample image of 16-QAM modulated covid-19 patient lung X-RAY and AWGN channel noise added

Sample image of 16-QAM modulated covid-19 patient lung X-RAY and rayleigh channel noise added

Sample image of 16-QAM modulated covid-19 patient lung X-RAY and rician channel noise added

Sample image of 64-QAM modulated normal people lung X-RAY and AWGN channel noise added

Sample image of 64-QAM modulated normal people lung X-RAY and rayleigh channel noise added

Sample image of 64-QAM modulated normal people lung X-RAY and rician channel noise added

Sample image of QPSK modulated normal people lung X-RAY and AWGN channel noise added

Sample image of QPSK modulated normal people lung X-RAY and rayleigh channel noise added

Sample image of QPSK modulated normal people lung X-RAY and rician channel noise added

Figure 2: Random sample modulated images of COVID-19 patient's lungs X-ray and normal people lungs X-ray used in the experiments.

parameter can be reduced linearly. Although these two methods share universal principles of design, multiple research findings have shown that they have distinct strengths to improve deep learning. Many tools simplify both approaches as a simple call function, enabling flexible stacking to build deep learning architectures. Although its usage use directives are available, there are unfortunately no defined guidelines or detailed studies on network configuration, data input, accuracy, and learning efficiency to investigate them. It is unclear when users should consider using both DO and BN, and how they can be combined (or used as an alternative) to obtain optimized deep learning performance. In CNNs, BN and DO should be used with precaution and experimentation.

The phenomenon of variance shift causes the disharmony between BN and DO techniques. DO's behaviour is different between the training and the testing stages, which shift the input statistics that are learned in BN. DO will change the variance of a specific neural unit as we switch the network position from train to test. BN will, therefore, preserve the statistical variation accumulated throughout the test phase during the learning procedure. The inconsistency of such variance ("variance shift") results in numerically unstable behaviour in the inference that ultimately leads to more incorrect predictions when DO is applied before the BN.

BN technique is a way to achieve deterministic information flow where each neuron participates in a process to achieve zero mean and unit variance. Let values of variable $x$ over a minibatch be represented with $m$ instances $(B = \{x^{(1)\cdots(m)}\})$. Mathematically, we can express the normalize part as

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}, \sigma^2 = \frac{1}{m}\sum_{i=1}^{m}\left(x^{(i)} - \mu\right)^2, x\wedge^{(i)} = \frac{x^{(i)} - \mu}{\sqrt{\sigma^2 + \epsilon}}, \quad (11)$$

where $\mu$ and $\sigma^2$ appears in the backpropagation. Normalization of activations based on the minibatch enables efficient training but is neither required nor desired during inference. As a result, BN accumulates moving averages of neural means and variances throughout learning to track a model accuracy as it trains which can be expressed as

$$E^{\text{moving}}(x) \longleftarrow E_B(\mu), \text{Var}^{\text{Moving}}(x) \longleftarrow \acute{E}_B\left(\sigma^2\right). \quad (12)$$

Here, $E_B(\mu)$ represents the expectation based on multiple training minibatches, and $\acute{E}_B(\sigma^2)$ signifies the expectation

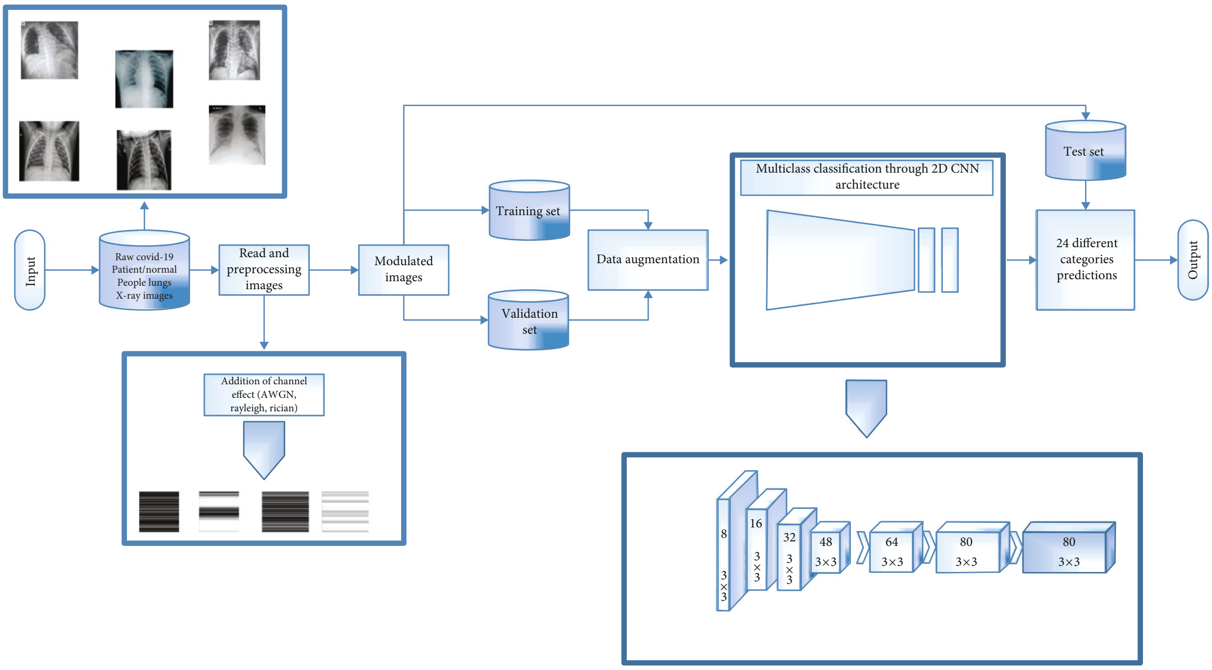

Input
Raw covid-19 Patient/normal People lungs X-ray images
Read and preprocessing images
Modulated images
Training set
Validation set
Data augmentation
Test set
Multiclass classification through 2D CNN architecture
24 different categories predictions
Output
Addition of channel effect (AWGN, rayleigh, rician)
8
3×3
16
3×3
32
3×3
48
3×3
64
3×3
80
3×3
80
3×3

Figure 3: Workflow of the proposed approach for multiclass (24-classes) modulated images classification.

based on the unbiased variance estimate over multiple training minibatches. They are all obtained by moving averages implementations and are fixed during inference for linear transform which can be expressed mathematically as

$$\hat{x} = \frac{x - E^{\text{moving}}(x)}{\sqrt{\text{Var}^{\text{Moving}}(x) + \epsilon}}. \tag{13}$$

We will now present theoretical analysis for the case shown in Figure 5, where there is a single convolutional layer sandwiched between a BN and a DO.

Here, $X$ is obtained by $\sum_{i=1}^{d} w_i a_i (1/p) x_i$ during training, where $w$ denotes the corresponding weights for $x$ taking into consideration the fact that DO has been applied. To ease the analysis, we assume that weights of $w$ remain constant so that the gradients approach to zero. We can expand $\text{Var}^{\text{Train}}(X)$ as follows:

$$\begin{aligned}\text{Var}^{\text{Train}}(X) &= \text{Cov}\left(\sum_{i=1}^{d} w_i a_i \frac{1}{p} x_i, \sum_{i=1}^{d} w_i a_i \frac{1}{p} x_i\right) \\ &= \left(\frac{1}{p}\left(c^2 + v\right) - c^2\right)\left(\sum_{i=1}^{d} w_i^2 + \rho^{ax} \sum_{i=1}^{d} \sum_{j\neq i}^{d} w_i w_j\right),\end{aligned} \tag{14}$$

where $\rho^{ax} = \text{Cov}(a_i x_i, a_j x_j)/\sqrt{\text{Var}(a_i x_i)}\sqrt{\text{Var}(a_j x_j)} \in [-1, 1]$.

Similarly, $\text{Var}^{\text{Test}}(X)$ can be written as follows:

$$\text{Var}^{\text{Test}}(X) = Cov\left(\sum_{i=1}^{d} w_i x_i, \sum_{i=1}^{d} w_i x_i\right) = v\left(\sum_{i=1}^{d} w_i^2 + \rho^{x} \sum_{i=1}^{d} \sum_{j\neq i}^{d} w_i w_j\right), \tag{15}$$

where $\rho^{x} = \text{Cov}(x_i, x_j)/\sqrt{\text{Var}(x_i)}\sqrt{\text{Var}(x_j)} \in [-1, 1]$. Finally, variance shift can be expressed as

$$\Delta(p, d) = \frac{\text{Var}^{\text{Test}}(X)}{\text{Var}^{\text{Train}}(X)} \tag{16}$$

Ideally, we would like $\Delta(p, d) \longrightarrow 1$ which can be achieved by eliminating DO or by growing the width of the channel.

## 4. Experimental Results

As given in Tables 1 and 2, we deployed 5-fold and 10-fold CV approaches to study the Automatic Modulation Classification (AMC) problem. We performed experiments to study disharmony between BN and DO techniques in the presence of data augmentation methods, without BN, without data augmentation, and with different data augmentation schemes. We performed a total of 66 experiments. 60 experiments were done as part of 5- and 10-fold CV approaches to select the optimum set of hyperparameters, while 6 experiments were done on the testing dataset. Note that we did not perform any experiments on the testing dataset while tuning hyperparameters done on the experiments without BN. We augment only the training dataset and validation

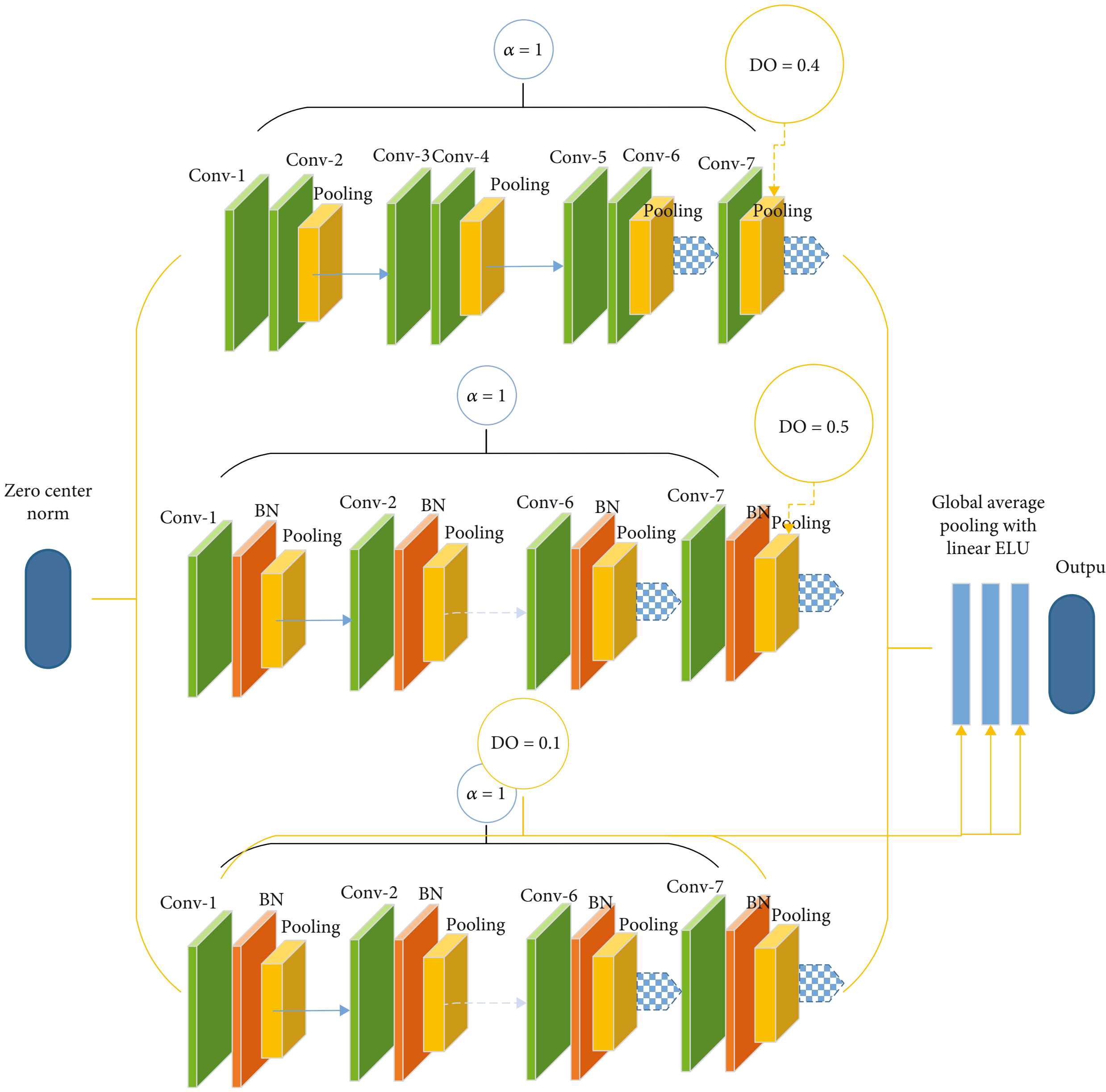


Figure 4: Three different architectures used for analyzing the effect of data augmentation, BN and disharmony between BN and DO techniques.

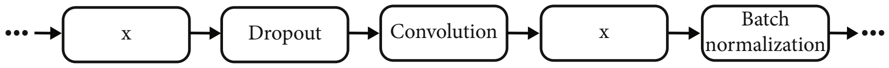


Figure 5: A representation of disharmony between BN and DO techniques.

datasets in the experiments while the testing dataset is never augmented. The experiments were done in order to show the effectiveness of the proposed methods. We note that deploying 10-fold CV leads to a model that has better performance as compared to its 5-fold counterpart. We performed a sufficient number of experiments to ensure that our results are representative of real-time scenarios at par with other studies reported in the literature.

The computational work was carried out with MATLAB 2019b (9.7) on the Window Server with only a single NVIDIA GPU Titan RTX. Datasets were divided into training, validation, and testing structures, and we used different 2D-CNN architecture parameters. As given in Tables 1 and 2, the distribution of data is 86.4% for training, 9.6% for validation, and 4% for testing for 10-fold CV, while the distribution of data is 76.8% for training, 19.2% for validation, and 4% for testing for the 5-fold CV. The techniques of with/without data augmentation, absence of BN, and disharmony between BN and DO are adopted during training. While CV methods for 5-fold and 10-fold are used to get high accuracy in the modulated data collection of limited samples. Each model has a different performance and the 2D-CNN model using disharmony between BN and DO techniques has the highest efficiency shown in Table 3. The training and validation accuracy of different models are shown in Figures 6 and 7, respectively.

We used data augmentation to increase the diversity of data for training the models. The parameters used for data augmentation are random rotation with a range of -5 to +5, horizontal and vertical translations from -3 to +3 pixels, random reflections in the top-bottom and left-right directions with a probability of 50%, as well as horizontal and vertical shear in the range -5 to +5. We used a minibatch size of 10 and a piecewise learning rate schedule in which we multiply the last learning rate with 0.1 after every 5 epochs. We trained the model for 200 epochs and shuffled the training dataset after every epoch, and the initial learning rate of 0.01 was chosen to train the model with the Adam optimization algorithm [53].

Confusion matrices evaluation results for the architectures presented earlier in the methodology section are shown in Table 4. The confusion matrices are obtained by testing the best model in all of the considered scenarios on the testing set. The confusion matrix evaluation results for the CNN architectures trained without BN using both 5- and 10-fold CV approaches are not presented as they are severely overfitted to just one class. The error rate for the test split is given in equation (17).

$$\text{Error Rate} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}, \tag{17}$$

where false positive (FP) is incorrect positive prediction, false negative (FN) is incorrect negative prediction true positive (TP) is the correct positive prediction, and true negative (TN) is correct negative prediction.

Sensitivity is also known as recall (REC) or true positive rate (TPR) rate and is given in equation (18).

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{18}$$

where true positive (TP) is the correct positive prediction and FN is the false negative.

Specificity, which is the ratio of TN values to TN+FP is given in equation (19).

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}, \tag{19}$$

where FP is false positive and true negative (TN) is correct negative prediction.

Precision, ratio of TP, and TP+FP are shown in equation (20).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{20}$$

Cohen's kappa is a calculation of how well the test is conducted in comparison to how well it should have been done at all. In other words, a model would have high kappa scores if the model has a significant difference between its accuracy and error rate.

$$\text{Cohen}'\text{s kappa} = \frac{\text{P}_0 - \text{P}_\text{e}}{1 - \text{P}_\text{e}}, \tag{21}$$

where $P_0$ is the overall model accuracy and $P_e$ is the measure of the agreement between the predictions of the model and the values of the actual class.

F-Score is typically useful than accuracy, mainly when the class distribution is inconsistent. Accuracy works better if the FN and the FP of the model have the same costs. If the costs of FP and FN are significantly different, both precision and recall need to be considered. The formula of F-Score is given in equation (22).

$$\text{F} - \text{SCORE} = \frac{2 * (\text{Sensitivity} * \text{Precision})}{\text{Sensitivity} + \text{Precision}}. \tag{22}$$

Furthermore, we considered the following seven metrics for the evaluation and comparison of the performance of architectures for all classes of signal classification: Relative Classifier Information (RCI), Strength of Agreement Matthews' benchmark (SOA-Matthews) for all categories, Matthews' correlation coefficient (MCC), class-wise Index of Balanced Accuracy (IBA), class-wise Geometric Mean (GM), class-wise Confusion Entropy (CEN), and F2 Score.

GM is defined as the square root of the product of true positive rate (TPR) and true negative rate (TNR). It focuses only on the recall of each class. Algorithms that completely misidentify one class will receive a GM assessment value of zero. Alone, it may not be a sufficient metric for model assessment.

RCI is an information-theoretic approach designed expressly to summarize how distinctly classes have been separated. This measure has a range between zero and one where large values indicate better classification. The performance of different classifiers on the same domain can be measured by comparing RCI values. RCI is essential for ranking uniformity of predictions while ignoring if the classes have been predicted correctly or not. A hazardous quality of RCI is that both perfect misclassification and perfect classification return the same value.

MCC is a correlation coefficient between the observed and predicted binary classifications; it returns a value between -1 and +1. A factor of +1 represents a perfect prediction, 0 represents no better than a random forecast, and -1 indicates total disagreement between prediction and observation. Though MCC has been established as one of the best binary classification task measures, its performance in

Table 3: The training, testing, and validation accuracy using different 2D-CNN models.

| Models | 5-FCV | 10-FCV | Aug | BN | Disharmony | Training accuracy (%) | Testing accuracy (%) | Validation accuracy (%) |
|---|---|---|---|---|---|---|---|---|
| 2D-CNN | ✓ | | | ✓ | | 80.91 | 79.54 | 77.03 |
| 2D-CNN | | ✓ | | ✓ | | 84.22 | 81.12 | 82.17 |
| 2D-CNN | ✓ | | ✓ | ✓ | | 88.89 | 85.83 | 85.26 |
| 2D-CNN | | ✓ | ✓ | ✓ | | 90.05 | 88.41 | 88.60 |
| 2D-CNN | ✓ | | ✓ | ✓ | ✓ | 91.0 | 89.03 | 89.78 |
| 2D-CNN | | ✓ | ✓ | ✓ | ✓ | 96.59 | 92.31 | 93.13 |

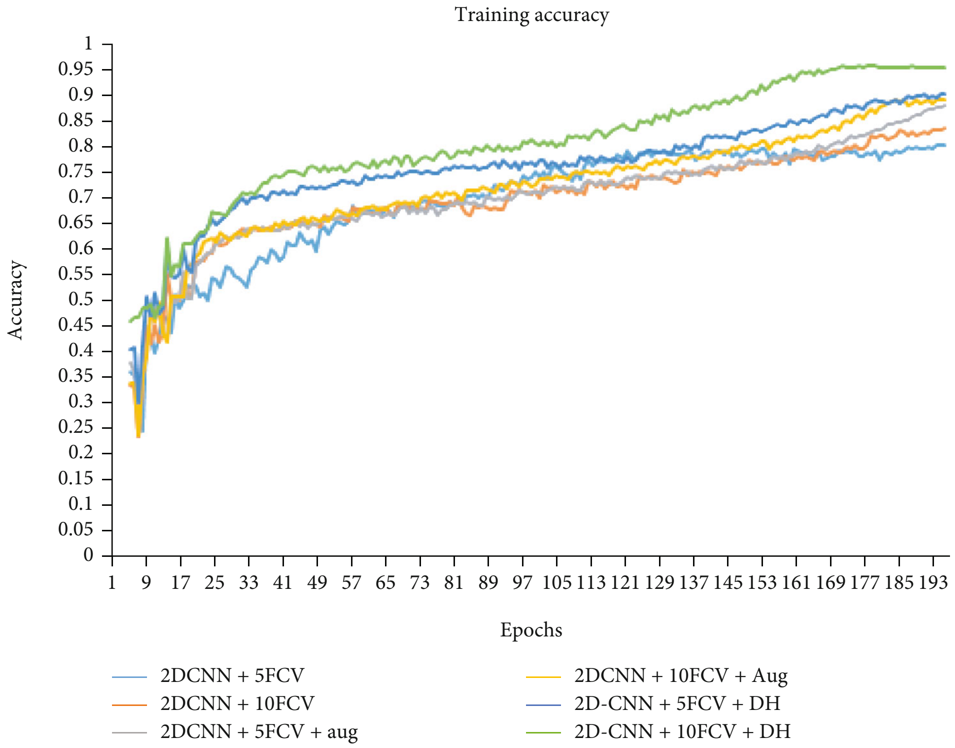


Figure 6: Comparison of the training accuracy of 6 different parameters models.

multiclass settings is far less studied. One of the strengths of MCC is that it can correctly identify random assignments of data with more consistency than the other measures. Hence, it can be best reserved for understanding if a classifier is randomly assigning class labels.

SOA-Matthews is a way of calculating the Pearson product-moment correlation coefficient. It has the same interpretation as MCC.

CEN is an information-theoretic approach that discriminates among confusion matrices. It may fail to recognize randomness in a classifier's information and may be used in cases where discrimination between the confusion matrix is essential. Small values of CEN represent less information loss and better classification performance.

IBA is a method that combines an unbiased index of its overall accuracy and a measure of how dominant the class with the highest individual accuracy rate is. Like GM, it focuses only on the recall of each class. A significant shortcoming of IBA is that it will neglect how well the classifier is performing the predictions. A loss in the measure's discriminatory power is a direct result of this oversight and manifests itself when trying to compare two models with similar per class performance. IBA may tend to uplift models that predict well across all categories while ignoring those that cannot.

The CNN models trained without BN severely overfitted to just one class, and hence, their results are omitted.

The RCI metric for the 2D-CNN model trained using 5-fold CV and 10-fold CV without data augmentation is 82.3% and 75.5%, while the 2D-CNN model trained using 5-fold CV and 10-fold CV with data augmentation is 85.4% and 76.6%, respectively. 2D-CNN using disharmony between BN and DO techniques trained using 5-fold CV with data augmentation with a small value of DO before softmax and after every convolution and fully connected layer along with BN is 86.2% and trained using 10-fold CV with the same parameters are 90.2% which is given in Table 5.

Based on the RCI metric alone, the best performing models from the highest to lowest order are the 2D-CNN

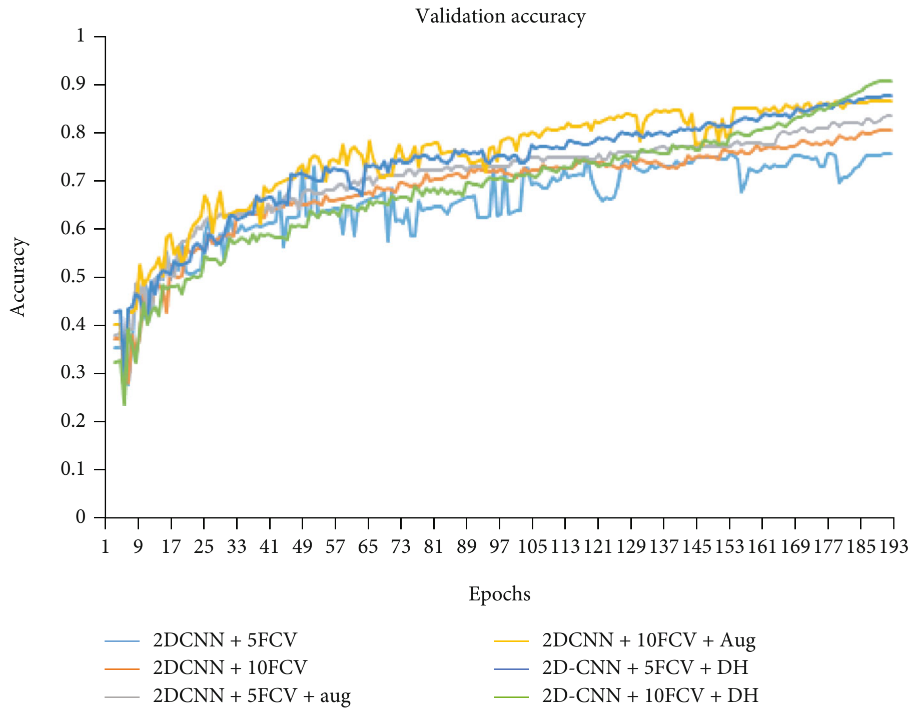


Figure 7: Comparison of the validation accuracy of 6 different parameters models.

Table 4: The error rate, specificity, sensitivity and precision, Cohen's kappa, F0.5 Score, and F1 Score using different 2D-CNN models.

| Models | Error rate (%) | Specificity (%) | Sensitivity (%) | Precision (%) | Cohen's kappa (%) | F0.5 Score (%) | F1 Score (%) |
|---|---|---|---|---|---|---|---|
| 2D-CNN+5FCV | 40.918 | 99.11 | 79.53 | 80.43 | 78.651 | 80.05 | 79.69 |
| 2D-CNN+10FCV | 28.333 | 99.38 | 85.84 | 86.50 | 85.217 | 86.21 | 85.93 |
| 2D-CNN+5FCV+Aug | 23.184 | 99.49 | 88.49 | 89.12 | 87.902 | 88. 86 | 88. 60 |
| 2D-CNN+ 10FCV+Aug | 8.260 | 99.17 | 81.14 | 82.13 | 80.30 | 81.66 | 81.23 |
| 2D-CNN+5FCV+DH | 2.282 | 99.51 | 88.86 | 89.53 | 88.358 | 89. 25 | 88.97 |
| 2D-CNN+10FCV+DH | 1.539 | 99.66 | 92.32 | 93.02 | 91.973 | 92.03 | 92.44 |

Table 5: The Relative Classifier Information (RCI), Strength of Agreement Matthews' benchmark (SOA-Matthews), Matthews correlation coefficient (MCC), Index of Balanced Accuracy (IBA), Geometric Mean (GM), Confusion Entropy (CEN), and F2 Score using different 2D-CNN models.

| Models | RCI (%) | SOA-Matthews (%) | MCC (%) | IBA (Avg) (%) | GM (%) | CEN (%) | F2 Score (%) |
|---|---|---|---|---|---|---|---|
| 2D-CNN+5FCV | 82.3 | Strong | 85.23 | 74.05 | 92.31 | 13.68 | 79.54 |
| 2D-CNN+10FCV | 75.2 | Strong | 78.67 | 63.6 | 92.44 | 19.44 | 85.82 |
| 2D-CNN+5FCV+Aug | 85.4 | Strong | 87.92 | 78.55 | 93.80 | 11.24 | 88.48 |
| 2D-CNN+10FCV+Aug | 76.6 | Strong | 80.33 | 66.48 | 89.62 | 18.25 | 81.09 |
| 2D-CNN+5FCV+DH | 86.2 | Strong | 88.37 | 79.15 | 94.01 | 10.66 | 88.85 |
| 2D-CNN+10FCV+DH | 90.2 | Very strong | 91.99 | 85.35 | 95.90 | 7.4 | 92.27 |

model trained with disharmony between BN and DO techniques with 10-fold CV and data augmentation, 2D-CNN model trained with disharmony between BN and DO techniques with 5-fold CV with data augmentation, 2D-CNN using 5-fold CV with data augmentation, 2D-CNN using 5-fold CV without data augmentation, 2D-CNN using 10-fold CV with data augmentation, and 2D-CNN using 10-fold CV without data augmentation.

Based on combined metrics such as SOA-Matthews, MCC, IBA, GM, CEN, F2 Score, F1 Score, F0.5 Score, Error Rate, Specificity, Sensitivity, Precision, and Cohen's kappa for multiclass modulated signal classification, the best performing model is the 2D-CNN model trained with disharmony between BN and DO techniques, while the least performing model is the 2D-CNN trained using 5-fold CV without data augmentation.

## 5. Discussion

From the results, it is pertinent that the comparison based on a single performance metric is error prone. Therefore, we discuss the effects that we obtained based on all parameters combined. The best performing model is the 2D-CNN model trained with disharmony between BN and DO techniques using 10-fold CV with data augmentation. 10-fold CV is generally considered to be more representative of the real-world scenarios due to its large training set (fewer validation set samples) as compared to its 5-fold counterpart. Augmentation helped in this case because, as the number of examples increases, the generalization ability of the model improves. The second best model, which studies the disharmony between BN and DO techniques trained using 5-fold CV with augmentation, is more representative of the real-world scenarios performs slightly worse due to more validation data. Next comes, the 2D-CNN model trained using 10-fold CV with augmentation along with BN. This model is performing better than the CNN model trained using 5-fold CV with augmentation along with BN layer, which can be explained by its larger training set size. The performance of CNN models trained using 10-fold CV without augmentation is lower due to their inability to model real-world statistics adequately due to their smaller validation set sizes and also lack of augmentation. We can see that the model with augmentation is performing better than its non-augmentation counterparts. The model using disharmony between BN and DO techniques trained with 10-fold CV performed better than the model using disharmony between BN and DO techniques trained with 5-fold CV or other 2D-CNN models without disharmony between BN and DO techniques. We can also see that if we will not keep the DO rate smaller, then the disharmony between BN and DO deteriorates the performance of the models and results in the loss of generalization abilities. Hence, it would be a better idea to cater to this effect. The models without BN are not able to generalize at all, which highlights the importance of the fair use of BN in training the deep learning models for the multiclass digital modulated signals classification.

The work presented has important contributions in the real settings. First, the deployment of higher-order data dimension such as 2D allows for better exploitation of information present in the data. Secondly, the use of medical image datasets such as COVID-19 instead of normal images is more beneficial in practice to the people. Thirdly, the use of deep learning architectures such as CNNs allows for a more powerful representation of the data for classification tasks. Most of the time, data affected by noise or fading effects is hard to classify. The proposed methods deploy deep learning architectures thus exploiting the structure of data effectively leading towards robustness and cost-effectiveness while also being time efficient.

## 6. Conclusion

In this article, we presented the problem of the detection of digital modulated images in the presence of channel noise using CNNs. The best performing model is the 2D-CNN model using disharmony between BN and DO techniques trained using 10-fold CV with augmentation. This study can be further enhanced in a number of ways. 3D-CNN architectures can be deployed instead of 2D with variations in the types of augmentation methods deployed in the network. In addition, more digital modulation schemes and channel types can be considered to build a robust AMC system to cater for the real world environments.

## Data Availability

Data are available on request.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Acknowledgments

This work has been supported by National Natural Science Foundation of China. This research was funded by the National Natural Science Foundation of China under grant 62031014.

## References

[1] A. Iversen, N. K. Taylor, K. E. Brown, and J. Karstad, *Classification of Communication Signals and Detection of Unknown Formats Using Artificial Neural Networks*, Heriot-Watt Univ Edinburgh, United Kingdom, 2006.

[2] J.-P. Linnartz, "Exact analysis of the outage probability in multiple-user mobile radio," *IEEE Transactions on Communications*, vol. 40, no. 1, pp. 20–23, 1992.

[3] M. Austin and G. Stuber, "Exact cochannel interference analysis for log-normal shadowed Rician fading channels," *Electronics Letters*, vol. 30, no. 10, pp. 748-749, 1994.

[4] D. L. Kumari and M. G. Prasad, "Error rate performance of OFDMA and MIMO technology over Rayleigh fading channel in 4G networks," *International Journal of Applied Engineering Research*, vol. 13, no. 12, pp. 10687–10689, 2018.

[5] C. E. Shannon, "A mathematical theory of communication," *ACM Sigmobile mobile computing and communications review*, vol. 5, no. 1, pp. 3–55, 2001.

[6] X. Li, F. Dong, S. Zhang, and W. Guo, "A survey on deep learning techniques in wireless signal recognition," *Wireless Communications and Mobile Computing*, vol. 2019, 12 pages, 2019.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[8] F. Clement, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013.

[9] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," *Advances in Neural Information Processing Systems*, vol. 2, no. 1, pp. 1799–1807, 2014.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] A. Noor, Y. Zhao, R. Khan, L. Wu, and F. Y. O. Abdalla, "Median filters combined with denoising convolutional neural

network for Gaussian and impulse noises," *Multimedia Tools and Applications*, vol. 79, no. 25-26, pp. 18553–18568, 2020.

[12] A. Noor, Y. Zhao, K. Anis, L. Wu, R. Khan, and F. Y. O. Abdalla, "Automated sheep facial expression classification using deep transfer learning," *Computers and Electronics in Agriculture*, vol. 175, p. 105528, 2020.

[13] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[14] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward convolutional blind denoising of real photographs," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1712–1722, Long Beach, CA, USA, 2019.

[15] Q. Zhang, Q. Yuan, J. Li, X. Liu, H. Shen, and L. Zhang, "Hybrid noise removal in hyperspectral imagery with a spatial–spectral gradient network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 10, pp. 7317–7329, 2019.

[16] Q. Zhang, Q. Yuan, J. Li, F. Sun, and L. Zhang, "Deep spatio-spectral Bayesian posterior for hyperspectral image non-i.i.d. noise removal," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 164, pp. 125–137, 2020.

[17] H. Zhang, Y. Wang, L. Xu, T. Aaron Gulliver, and C. Cao, "Automatic modulation classification using a deep multi-stream neural network," *IEEE Access*, vol. 8, pp. 43888–43897, 2020.

[18] K. Bu, Y. He, X. Jing, and J. Han, "Adversarial transfer learning for deep learning based automatic modulation classification," *IEEE Signal Processing Letters*, vol. 27, pp. 880–884, 2020.

[19] R. Yilmaz and A. E. Pusane, "Deep learning based automatic modulation classification in the case of carrier phase shift," in *43rd International Conference on Telecommunications and Signal Processing (TSP)*, pp. 354–357, Milan, Italy, 2020.

[20] Y. Wu, X. Li, and J. Fang, "A deep learning approach for modulation recognition via exploiting temporal correlations," in *19th IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, Kalamata, Greece, 2018.

[21] J. Sun, G. Wang, Z. Lin, S. G. Razul, and X. Lai, "Automatic modulation classification of cochannel signals using deep learning," in *23rd IEEE International Conference on Digital Signal Processing (DSP)*, pp. 1–5, Shanghai, China, 2018.

[22] X. Li, S. Chen, X. Hu, and J. Yang, "Understanding the disharmony between dropout and batch normalization by variance shift," in *Proceedings of the IEEE/CVPR Conference on Computer Vision and Pattern Recognition*, pp. 2682–2690, Long Beach, CA, USA, 2019.

[23] K. Lenc and A. Vedaldi, "Understanding image representations by measuring their equivariance and equivalence," *International Journal of Computer Vision*, vol. 127, no. 5, pp. 456–476, 2019.

[24] C. Xiao, J.-Y. Zhu, B. Li, W. He, M. Liu, and D. Song, "Spatially transformed adversarial examples," in *Proceedings of 6th International Conference on Learning Representations, (ICLR)*, Vancouver, Canada, 2018.

[25] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Harmonic networks: deep translation and rotation equivariance," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7168–7177, Honolulu, HI, USA, 2017.

[26] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proceedings of International Conference on Learning Representations (ICLR)*, San Diego, California, United States, 2015.

[27] A. Azulay and Y. Weiss, "Why do deep convolutional networks generalize so poorly to small image transformations?," *Journal of Machine Learning Research*, vol. 20, no. 184, pp. 1–25, 2019.

[28] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell, "Understanding data augmentation for classification: when to warp?," in *Proceedings of 2016 international conference on digital image computing: techniques and applications (DICTA)*, pp. 1–6, Gold Coast, QLD, Australia, 2016.

[29] S. Dieleman, K. W. Willett, and J. Dambre, "Rotation-invariant convolutional neural networks for galaxy morphology prediction," *Monthly Notices of the Royal Astronomical Society*, vol. 450, no. 2, pp. 1441–1459, 2015.

[30] Q. S. Zhang and S. C. Zhu, "Visual interpretability for deep learning: a survey," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 1, pp. 27–39, 2018.

[31] L. Sifre and S. Mallat, "Rotation, scaling and deformation invariant scattering for texture discrimination," in *Proceedings of 2013 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 1233–1240, Portland, OR, USA, 2013.

[32] R. Zhang, "Making convolutional networks shift-invariant again," in *International Conference on Machine Learning (ICML)*, pp. 7324–7334, Long Beach, California, USA, 2019.

[33] I. Goodfellow, H. Lee, Q. V. Le, A. Saxe, and A. Y. Ng, "Measuring invariances in deep networks," *Advances in neural information processing systems*, vol. 22, pp. 646–654, 2009.

[34] Y. Xu, T. Xiao, J. Zhang, K. Yang, and Z. Zhang, "Scale-invariant convolutional neural networks," 2014, https://arxiv.org/abs/1411.6369.

[35] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *Proceedings of CVPR*, pp. 1521–1528, Colorado Springs, CO, USA, 2011.

[36] J. Donahue, Y. Jia, O. Vinyals et al., "A deep convolutional activation feature for generic visual recognition," 2013, https://arxiv.org/abs/1310.1531.

[37] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1872–1886, 2013.

[38] K. Sohn and H. Lee, "Learning invariant representations with local transformations," in *Proceedings of the 29 th International Conference on Machine Learning (ICML)*, pp. 1339–1346, Edinburgh, Scotland, UK, 2012.

[39] A. Ruderman, N. C. Rabinowitz, A. S. Morcos, and D. Zoran, "Pooling is neither necessary nor sufficient for appropriate deformation stability in CNNs," 2018, https://arxiv.org/abs/1804.04438.

[40] J. Bruna, A. Szlam, and Y. LeCun, "Learning stable group invariant representations with convolutional networks," in *Ist International Conference on Learning Representations (ICLR)*, Scottsdale, Arizona, USA, 2013.

[41] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[42] T. S. Cohen and M. Welling, "Transformation properties of learned visual representations," in *International Conference*

*on Learning Representations (ICLR)*, pp. 1–11, San Diego, California, United States, 2015.

[43] J. Ngiam, Z. Chen, S. A. Bhaskar, P. W. Koh, and A. Y. Ng, "Sparse filtering," *Advances in neural information processing systems*, vol. 24, pp. 1125–1133, 2011.

[44] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012, https://arxiv.org/abs/1207.0580.

[45] A. Labach, H. Salehinejad, and S. Valaee, "Survey of dropout methods for deep neural networks," 2019, https://arxiv.org/abs/1904.13310.

[46] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning (ICML)*, pp. 448–456, Lille, France, 2015.

[47] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–13, 2020.

[48] D. Hong, L. Gao, N. Yokoya et al., "More diverse means better: multimodal deep learning meets remote-sensing imagery classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 5, pp. 4340–4354, 2021.

[49] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "An augmented linear mixing model to address spectral variability for hyperspectral unmixing," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1923–1938, 2019.

[50] B. Rasti, D. Hong, R. Hang et al., "Feature extraction for hyperspectral imagery: the evolution from shallow to deep: overview and toolbox," *IEEE Geoscience and Remote Sensing Magazine*, vol. 8, no. 4, pp. 60–88, 2020.

[51] D. Hong, N. Yokoya, G.-S. Xia, J. Chanussot, and X. X. Zhu, "X-ModalNet: a semi-supervised deep cross-modal network for classification of remote sensing data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 167, pp. 12–23, 2020.

[52] A. Vehtari, A. Gelman, and J. Gabry, "Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC," *Statistics and Computing*, vol. 27, no. 5, pp. 1413–1432, 2017.

[53] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, pp. 1–15, San Diego, California, United States, 2015.