4
Symmetric Dot Pattern based Modulation Classification

4.1 INTRODUCTION

In chapter 3, an analytic approach for modulation classification has been explained which has less complexity but also have marginal classification efficiency. Now a days, we have sufficient resources to execute highly complex algorithms to improve the classification efficiency. In this chapter, modulation classification based on the Symmetric Dot Pattern (SDP) representation of Radio Frequency (RF) signal is explained. Snowflake images generated by the SDP technique are used to train the Deep Convolution Neural Network (DCNN). DCNN had great success in the domain of modulation classification due to its capability to classify highly noisy images. It is well suited in the communication domain due to the easy availability of large datasets. DCNN has an advantage over conventional Machine learning (ML) methods in sense of feature selection. DCNN chooses features autonomously while ML methods select features manually. In this chapter, two DCNN models viz. ResNet-50 and Inception ResNet V2, both concatenated by 8 fully connected layers are implemented and trained using data generated through LabVIEW. The density of points in the SDP pattern is used to generate a grayscale image. The Adaptive Local Power Law Transform (ALPLT) method has been used for color image generation through a grayscale image. A hierarchical model of eight stages with each stage doing a binary classification using DCNN is formed. All modulation schemes considered are classified accurately above 9 dB SNR.

4.2 SYSTEM MODEL

In this chapter, it is assumed that the receiver has the information of carrier frequency (f_c) and symbol rate (R_s) has been calculated using the method defined in [Jajoo *et al.*, 2017]. The flow chart of the proposed method is given in Figure 4.1. The received RF signal is downconverted to an intermediate frequency ($f' = \beta \times R_s$; β is a constant and considered to be 10). The downconverted signal is used to generate SDP density format which is transformed into a gray-scale image. ALPLT method is used to create three channels (RGB) for color image generation. A set of hierarchical decision units are employed for classification, each of which contains a DCNN model trained with the SDP color images. Every decision unit has a binary classifier equipped with ResNet-50 or Inception ResNet V2 based DCNN model. Optimum values of SDP parameters are estimated at each binary classifier unit for SDP formation by maximization of the SDP density difference of two classes.

4.3 SYMMETRIC DOT PATTERN

Symmetric Dot Pattern transforms the time-domain signal into a polar domain pattern to make it more expressive in visual perception. SDP is mainly used for the detection of variation in amplitude and frequency, which is satisfied in communication signals. The produced dot pattern has a shape similar to symmetrical snowflakes. The shape of the SDP image is unique for a particular modulation scheme, which has been used as a feature for the classification of communication signals .

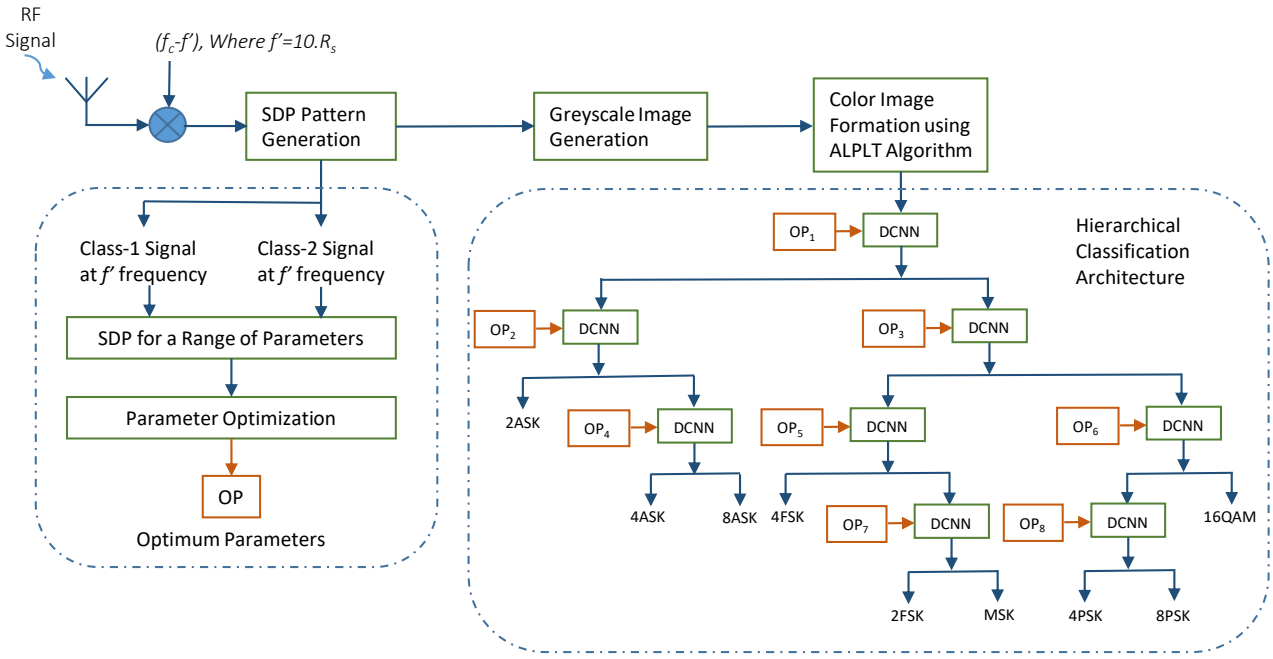


Figure 4.1 : Block diagram of complete system.

4.3.1 SDP formation

In the sampled sequence of the received signal, amplitude of the sample at time i is considered as $r_d(i)$ and the amplitude at time $i+\mathscr{L}$ is $r_d(i+\mathscr{L})$. By using these values in Equations (4.1), 4.2, and 4.3, the sample sequence is transformed into a polar coordinate system $P(\varkappa,\Psi)$. For one sample $r_d(i)$, two points are mapped at position $(\varkappa(i),\Psi(i))$ and $(\varkappa(i),\Psi'(i))$, where $\varkappa(i)$ is the radius of the polar coordinates, $\Psi(i)$ and $\Psi'(i)$ are counterclockwise and clockwise angle of point, respectively.

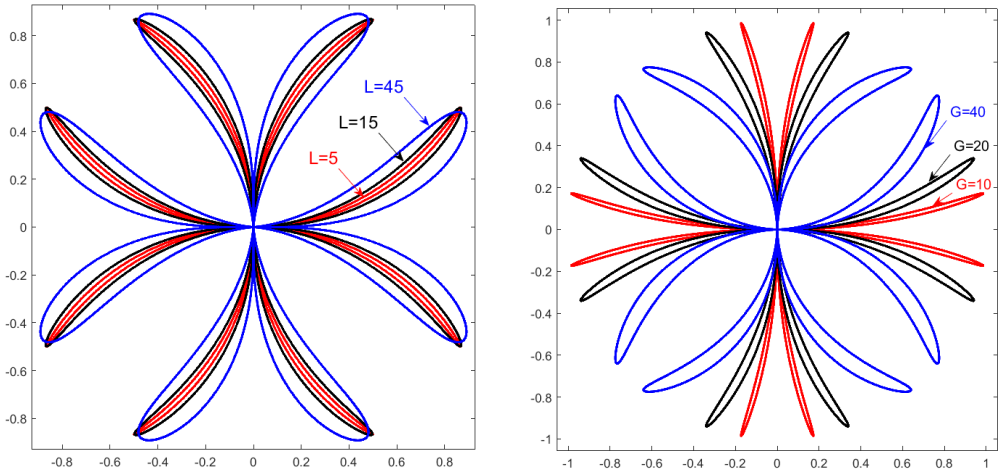
$$\varkappa(i) = \frac{r_d(i) - r_{d,\min}}{r_{d,\max} - r_{d,\min}} \tag{4.1}$$

$$\Psi(i) = \Psi_0 + \frac{r_d(i+\mathscr{L}) - r_{d,\min}}{r_{d,\max} - r_{d,\min}}\mathscr{G} \tag{4.2}$$

$$\Psi'(i) = \Psi_0 - \frac{r_d(i+\mathscr{L}) - r_{d,\min}}{r_{d,\max} - r_{d,\min}}\mathscr{G} \tag{4.3}$$

where, $r_{d,\max}$ and $r_{d,\min}$ are the maximum and the minimum amplitude of the r_d signal. $\mathscr{L}$ is the time delay between two consecutive data samples, Ψ_0 depicts the angle for the line of symmetry for two symmetrical snowflakes, and $\mathscr{G}$ is the amplification factor.

For one signal $r_d(i)$, $\Psi(i)$ and $\Psi'(i)$ composedly creates two symmetrical snowflakes around the line of symmetry at an angle Ψ_0. SDP image of a sine wave for 100 Hz frequency, and the effect of variation in $\mathscr{L}$ and $\mathscr{G}$ on the SDP image is shown in Figure 4.2. Figure 4.2(a) shows the effect of varying $\mathscr{L}$ on SDP pattern for $\mathscr{G}$=30, and Figure 4.2(b) shows the effect of varying $\mathscr{G}$ on SDP pattern for $\mathscr{L}$=15.



(a) G=30 (b) L=15

Figure 4.2 : SDP for sine wave at frequency of 100 Hz. (a) Effect of variation in $\mathscr{L}$ on SDP image for $\mathscr{G}$=30. (b) Effect of variation in $\mathscr{G}$ on SDP image for $\mathscr{L}$=15.

Table 4.1 : Candidate modulation schemes considered for classification.

Domain	Modulation Schemes
Phase-based modulations	QPSK, 8PSK
Phase and amplitude-based modulations	16QAM
Amplitude-based modulations	2ASK, 4ASK, 8ASK
Frequency-based modulations	2FSK, 4FSK, MSK

4.3.2 Gray-scale image formation

The received RF time-series signal is downconverted at f' frequency and processed using the SDP technique to build an SDP density structure in the Comma-Separated Values (CSV) format. SDP CSV files for all modulation schemes are generated using Equations (4.1), (4.2), and (4.3). The value of $\varkappa$ for any point mapped on a complex plane lies between 0 to 1 and the overall SDP pattern has a dimension of 2×2. The considered CNN models require a three-channel image with each channel dimension of 224×224. The SDP pattern is divided into 224×224 grid sections considering each section as a pixel and the number of points in each grid section is termed as pixel density. The pixel density in 224×224 dimensions is scaled in the range [0,255] and a gray-scale image is formed.

4.3.3 Color image generation using ALPLT

Most of the deep learning models require three-channel images for classification. The formed grayscale image is converted in three channels using Adaptive Local Power Law Transform (ALPLT) as shown in Figure 4.3. Equation (4.4), (4.5) and (4.6) are used for generation of three channels of the color image with ρ=1.7, ρ=2.2 and ρ=2.7, respectively. The ρ values are chosen which are appreciated by numerical analysis in [Tsai, 2013]. The pixel density of the three channels has different values which are combined for color image generation.

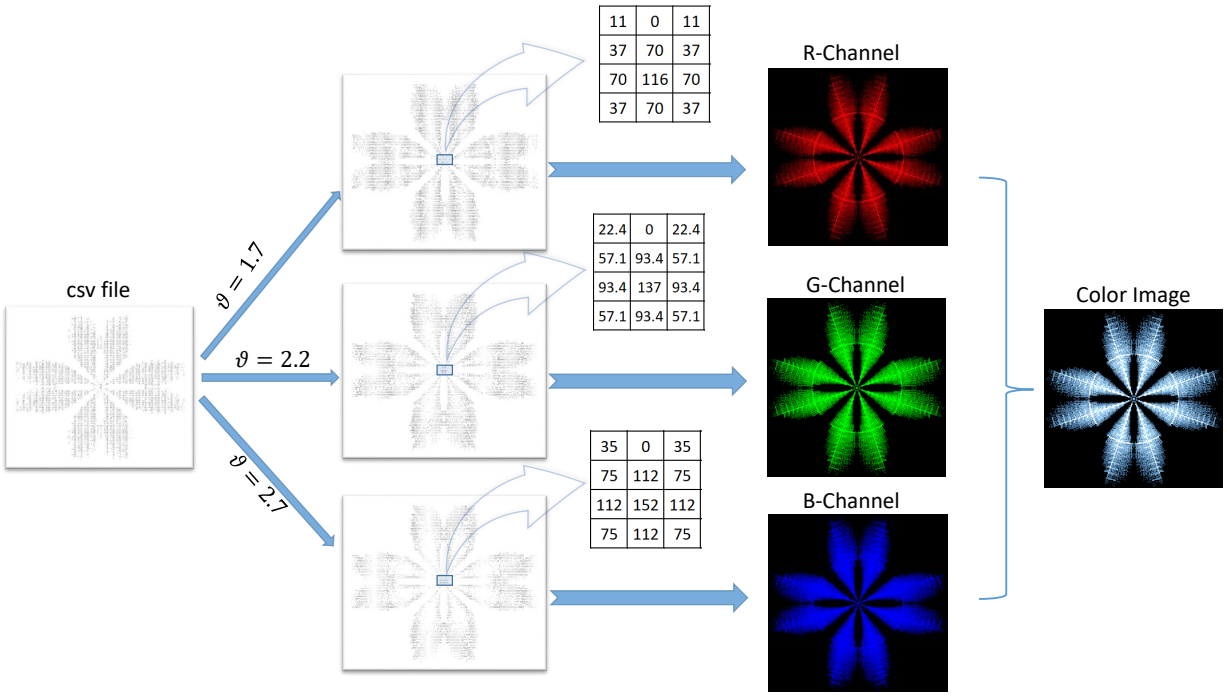


Figure 4.3 : Formation of color image using ALPLT method.

$$I(i,j) = 255\left(\frac{d(i,j)}{255}\right)^{\lambda} \tag{4.4}$$

$$\lambda = \left(\frac{1}{\rho}\right)^{\left(\frac{128-\mu(i,j)}{128}\right)} \tag{4.5}$$

$$\mu(i,j) = \frac{1}{9}\left[\frac{1}{9}\sum_{k=i-1}^{i+1}\sum_{\substack{l=j-1 \\ (k\neq i)\&(l\neq j)}}^{j+1} d(k,l) + d(i,j)\right] \tag{4.6}$$

Here $d(i,j)$ and $I(i,j)$ represent pixel density of gray-scale image and the intensity value of one of the channels of a color image, respectively.

In the same way data of all the considered modulation schemes have been generated. Color images of all modulations at 5 dB and 15 dB SNR for particular $\mathscr{L}$ and $\mathscr{G}$ are shown in Figure 4.4.

4.3.4 Selection of parameters (Ψ_0, $\mathscr{L}$, $\mathscr{G}$)

In the polar coordinate system, the selection of parameters Ψ_0, $\mathscr{L}$, and $\mathscr{G}$ decides the orientation and shape of the snowflakes. The considered four values of Ψ_0 are 0°, 90°, 180°, and 270° which creates four pairs of snowflakes. If the difference between two adjacent Ψ_0 values is

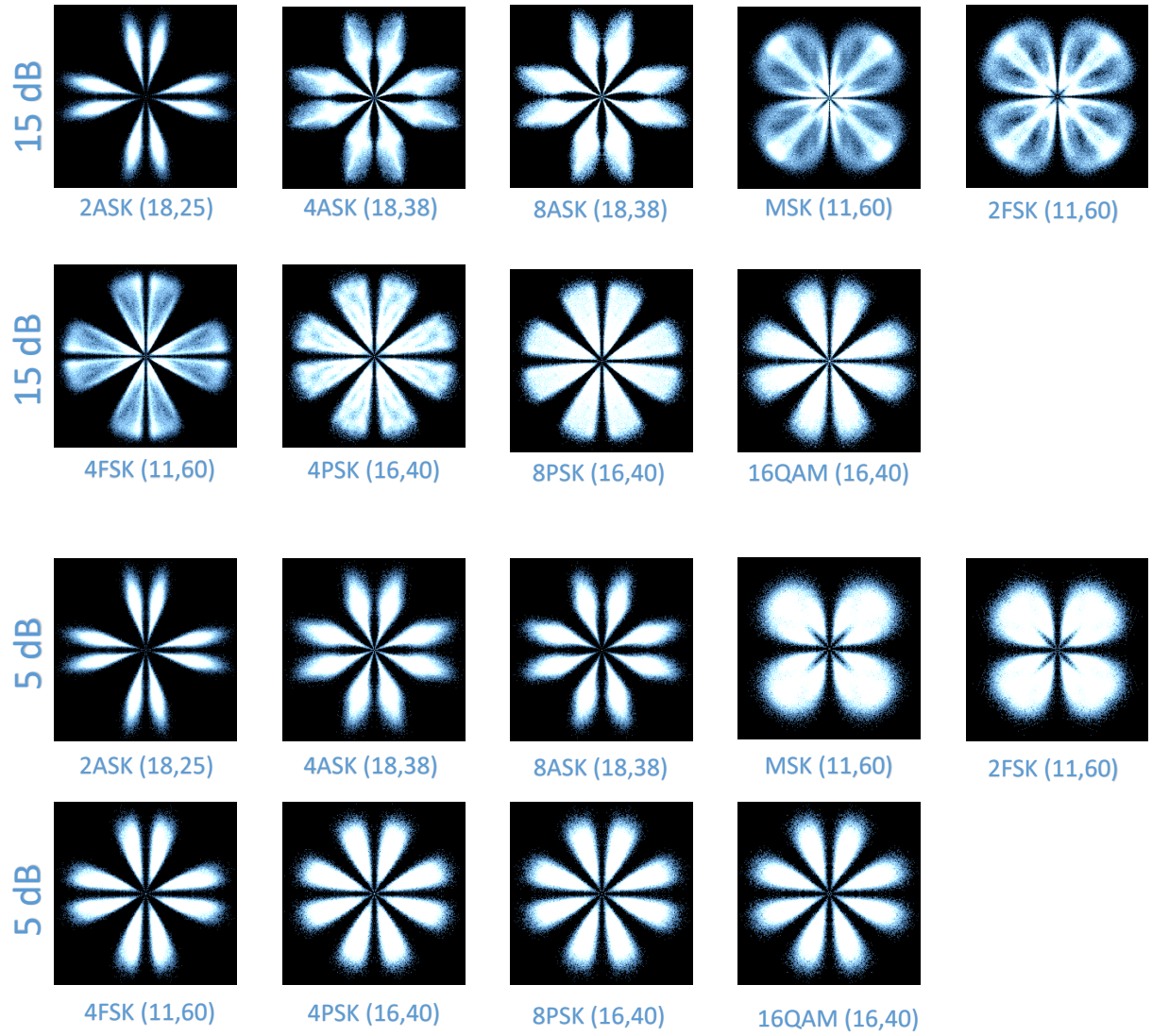


Figure 4.4 : SDP color images for all nine modulation schemes at 5dB and 15 dB SNR for particular ($\mathscr{L}$, $\mathscr{G}$) value.

small, the overlapping of snowflakes increases, which creates a disturbance in the identification of the pattern. The shape of the image depends on $\mathscr{L}$ and $\mathscr{G}$. $\mathscr{L}$ gives a correlation between the points of time series and $\mathscr{G}$ increases the angular width of each leaf of the polar plot. The optimum value of $\mathscr{L}$ and $\mathscr{G}$ improves the resolution of the image, which helps in signal classification. The values of $\mathscr{L}$ and $\mathscr{G}$ can vary from 1 to 20 and 1° to 90°, respectively. The estimation of the optimum value of $\mathscr{L}$ and $\mathscr{G}$ is discussed in the next section.

4.3.5 Optimum $\mathscr{L}$ and $\mathscr{G}$ estimation ($\mathscr{L}_{opt}$, $\mathscr{G}_{opt}$)

$\mathscr{L}_{opt}$ and $\mathscr{G}_{opt}$ are obtained by using the method explained in Algorithm 1. Ideal signal's SDP of two classes are generated for a range of values of $\mathscr{L}$ and $\mathscr{G}$. Matrix of m×n, $S_D(\mathscr{L},\mathscr{G})$ is formed by taking the sum of difference between pixel densities of two classes for every value of $\mathscr{L}$ and $\mathscr{G}$. For selection of $\mathscr{L}_{opt}$, $max(S_D(\mathscr{L},\mathscr{G}))$ function is used, which has two output vectors: *value* and *index*. *value* is a row vector containing the maximum value of each column, and *index* vector holds indices of rows containing those maximum values from each particular column. The

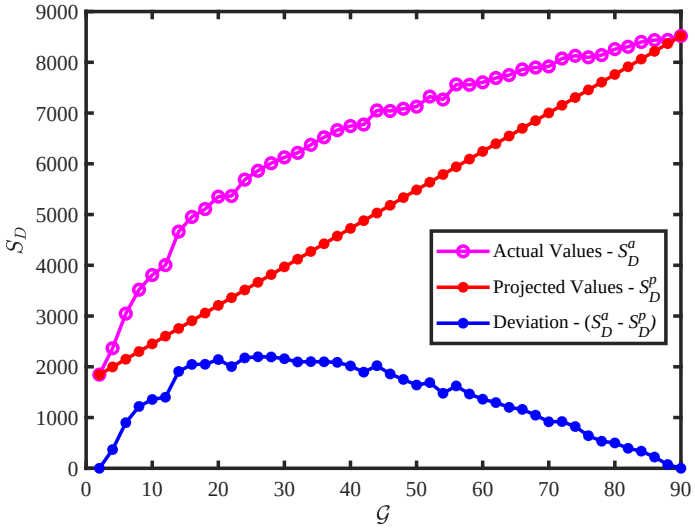


Figure 4.5 : Estimation of optimum value of $\mathscr{G}$ by maximizing natural and actual statistical difference between two classes.

mode value of *index* vector is taken as $\mathscr{L}_{opt}$. Once $\mathscr{L}_{opt}$ is selected, row vector of $S_D(\mathscr{L}_{opt},\mathscr{G})$ is used for selection of $\mathscr{G}_{opt}$. Value of $\mathscr{G}$ for which $S_D(\mathscr{L}_{opt},\mathscr{G})$ shows maximum deviation from its general characteristics (line joining initial and final value of $S_D(\mathscr{L}_{opt},\mathscr{G})$) is chosen as $\mathscr{G}_{opt}$. S_D^p shows projected values from general characteristics and S_D^a have the actual value of $S_D(\mathscr{L},\mathscr{G})$ as shown in Figure 4.5. The maximum considered values for $\mathscr{L}$ and $\mathscr{G}$ are m=20 and n=90 respectively.

Algorithm 1: $\mathscr{L}_{opt}$ and $\mathscr{G}_{opt}$ Estimation.

$Input : d_{class1}(\mathscr{L},\mathscr{G}), d_{class2}(\mathscr{L},\mathscr{G})$
$Output : \mathscr{L}_{opt}, \mathscr{G}_{opt}$
for $\mathscr{L} = 1 : m$ **do**
  for $\mathscr{G} = 1 : n$ **do**
    $D(\mathscr{L},\mathscr{G}) = |d_{class1}(\mathscr{L},\mathscr{G}) - d_{class2}(\mathscr{L},\mathscr{G})|$
    $S_D(\mathscr{L},\mathscr{G}) = Sum[D(\mathscr{L},\mathscr{G})]$
  end for
end for
$[value, index] = max(S_D(\mathscr{L},\mathscr{G}))$
$\mathscr{L}_{opt} = mode(index)$
$S_D^p[\mathscr{L}_{opt},\mathscr{G}] = S_D^a[\mathscr{L}_{opt},\mathscr{G}(1)] + \frac{S_D^a[\mathscr{L}_{opt},\mathscr{G}(90)] - S_D^a[\mathscr{L}_{opt},\mathscr{G}(1)]}{\mathscr{G}(90) - \mathscr{G}(1)}[\mathscr{G} - \mathscr{G}(1)]$
$\mathscr{G}_{opt} = \arg(\max_{\forall \mathscr{G}}(S_D^a[\mathscr{L}_{opt},\mathscr{G}] - S_D^p[\mathscr{L}_{opt},\mathscr{G}]))$

4.4 MODEL ARCHITECTURES OF CLASSIFICATION METHODS

A significant amount of work has been done using deep learning models for modulation classification [Peng *et al.*, 2018; Hiremath *et al.*, 2019; Zhang *et al.*, 2019]. The scope of performance improvement in classification using DCNN is still available and researchers are working towards optimization of architecture design. In the field of image processing, a large-scale image classification problem is solved with improved performance by incorporating more layers in deep network [He *et al.*, 2016]. In this work, ResNet-50 and Inception ResNet V2 models are used to

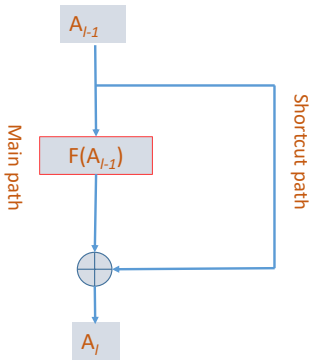


Figure 4.6 : Residual block structure.

build a modulation classifier. There are two types of problems in deep networks [He *et al.*, 2016]. One is accuracy saturation, which increases training and testing errors and the second is vanishing gradient, which affects the update of earlier layers parameters of the network. The vanishing gradient problem is handled by choosing the initial parameters of the model carefully and batch normalization before giving input to hidden layers. Accuracy saturation and gradient vanishing are addressed by employing residual blocks in deep networks. A shortcut connection is introduced and network layers are reformulated with a new residual function concerning the layer inputs.

4.4.1 Deep residual network

A residual block is a combination of the main path, consisting of some stacked nonlinear layers, and a shortcut path. A deep network is formed with multiple residual blocks. The residual block structure is shown in Figure 4.6. Consider that, each residual block is approximated by mapping of a function $H(x)$, where x is input to the function. For l^{th} layer having input and output A_{l-1} and A_l, the function relation between these two is given by

$$A_l = H(A_{l-1}) \tag{4.7}$$

In the main path, rather than directly mapping the function $H(x)$, we approximated the residual mapping by function $F(x) = H(x) - x$. Extraction of the actual function can be done by using relation $H(x) = F(x) + x$. So from (4.7),

$$A_l = F(A_{l-1}) + A_{l-1}. \tag{4.8}$$

In most cases, it is difficult to approximate the identity function using an array of nonlinear layers, which causes performance degradation. This has been resolved by residual mapping simply by making all the parameters of the main path to zero and the identity function is implemented by a shortcut path. In the proposed work, residual mapping has been done by using two types of blocks, one is convolution block (Conv_Block) and another is the identity block (Identity_Block) as shown in Figure 4.8(a) and 4.8(b). The main path of both the blocks contains two 1×1 and one 3×3 convolutions, three batch normalization, and three Rectified Linear Unit (ReLU) activation

layers. The shortcut path of the convolution block contains one 1×1 convolution and one batch normalization layer, whereas the identity block is directly passing the input.

From Equation (4.8), it is shown that learning of the residual block is done by element-wise addition of output of the main path and shortcut path, which is only possible for equal dimension data. The dimension of the shortcut path needs to be equal to the dimension of the main path for element-wise addition, which is done by linear projection [He *et al.*, 2016] as given by

$$A_l = F(A_{l-1}) + W_s A_{l-1} \tag{4.9}$$

Here, W_s is a linear projection matrix.

4.4.2 ResNet-50

The ResNet-50 model is shown in Figure 4.7. It comprises of two parts. The first is a pre-defined ResNet-50 model and the second consists of eight fully connected layers. ResNet-50 is a well-defined model, mostly used for image classification. The ResNet-50 model is a stacked configuration of convolution, batch normalization, activation, padding, and pooling layers. The convolution layer learns local features of the input image with a lower number of parameters as compared to the fully connected layer. Details of the complete model are given in further subsections. In the second part, a network of eight fully connected layers is concatenated to the output of the pre-defined ResNet-50 for extraction of the significant features.

Layer details

The proposed deep residual network, shown in Figure 4.7 contains 4 Conv_Block and 12 Identity_Block. The operational flow graph of Conv_Block and Identity_Block is shown in Figure 4.8. Each Conv_Block consists of 4 convolution layers, where two filters of size 1×1 and one of 3×3 in the main path are engaged to make a bottleneck structure to reduce the computational burden. The first 1×1 filter reduces the data dimensions by reducing the number of channels and the second increases it to the required dimensions by increasing the number of channels, whereas the 3×3 filter behaves as a bottleneck of the Conv_Block and Identity_Block. Due to the nonlinear behavior of the activation function, the implementation of an identity function is difficult. Identity_Block has been employed to overcome this problem. During training, the identity function is created by assigning the zero value to the main path parameters of Identity_Block, and the shortcut path serves the purpose. Comparison of computation cost of one Conv_Block using bottleneck structure and the plain network is shown in Figure 4.9(a) and similarly, the computation cost of Identity_Block using bottleneck and the plain network is shown in Figure 4.9(b). To convert 56×56×256 data into 28×28×512, bottleneck structure and plain network requires 295,436,288 and 1,849,688,064 multiplications respectively. Similarly, Identity_Block requires 192,675,840 and 2,066,448,384 multiplications in bottleneck structure and plain network respectively to convert 28×28×256 into 28×28×512. The same type of processing is done through all residual blocks and the output of the ResNet-50 network is flatten to 1000 values to give input to the fully connected network.

Pooling

For a fixed area of the image, one representative is chosen by taking mean or max. The pooling of features is needed to reduce the distortion effect along with computational complexity. In the proposed ResNet-50 model, after the first convolution layer of 7×7 filter, max-pooling is done to extract significant features. The maximum value is chosen in each 3×3 matrices with the stride of 2. A stride is a step size by which the filter moves in one direction. After the last convolution

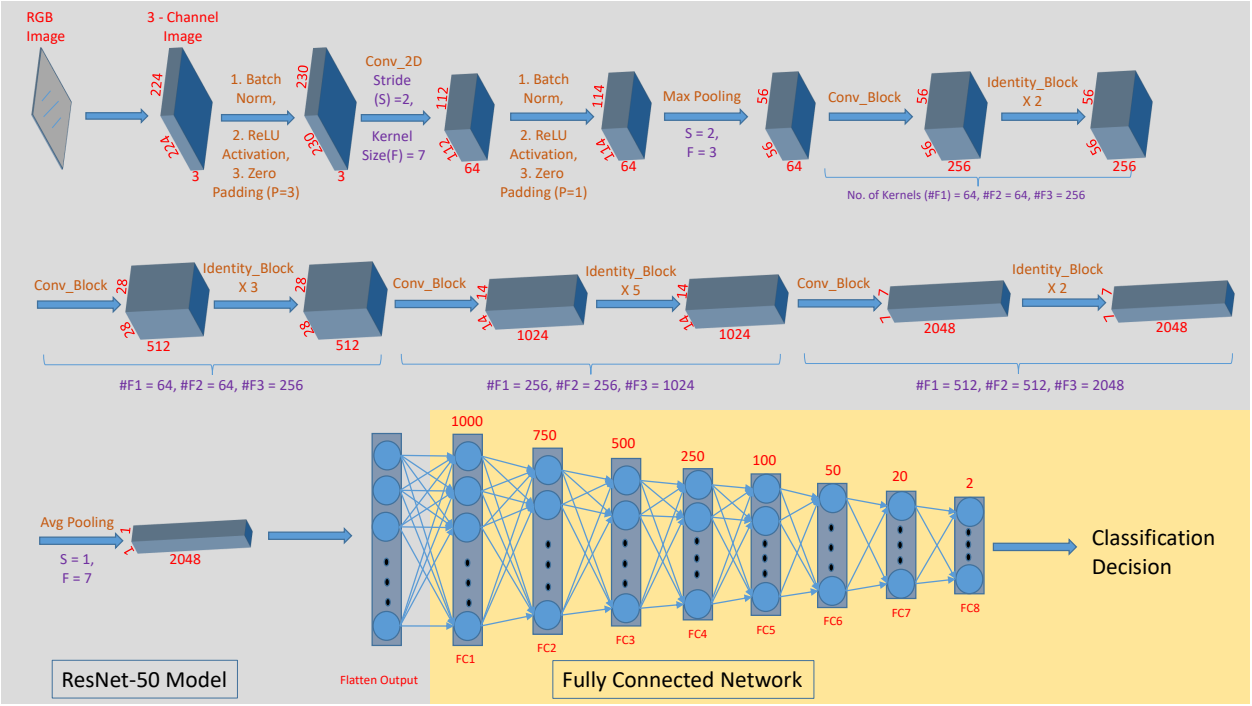


Figure 4.7 : DCNN model architecture. The first part of the Figure is the ResNet-50 [He *et al.*, 2016] model used for the extraction of basic features. Each Conv_Block and Identity_Block has formed a bottleneck structure to reduce computational complexity. The second part is a fully connected network, which takes an array of 1000 outputs from the pre-defined ResNet-50 model to extract significant features.

operation, average pooling is done as all the features are equally important for decision.

Batch normalization

Usually, network training is complicated due to the change in nature of the distribution of the current layer's input during training, as parameters of the previous layer change. A wide range of input to the activation function and a lower value of learning rate cause a slower training process. To overcome this situation, batch normalization has been taken into account. The input data to a layer is first normalized in mini-batches and then scaled and shifted before activation. Training data to the deep network is given in mini-batches of size N. Input mini-batch to each hidden layer before activation is normalized using equations as given below.

$$\mu = \frac{1}{N}\sum_{i=1}^{N} Z^{(i)} \tag{4.10}$$

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N} (Z^{(i)} - \mu)^2 \tag{4.11}$$

$$Z_{norm}^{(i)} = \frac{Z^{(i)} - \mu}{\sqrt{\sigma^2 + \delta}} \tag{4.12}$$

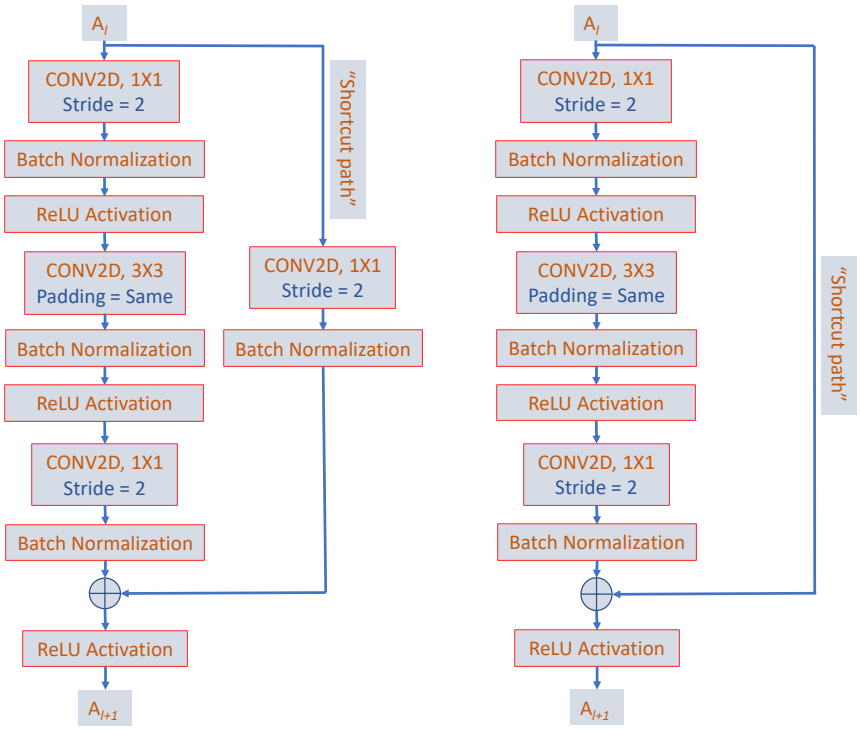


Figure 4.8 : Structure of (a) Conv_Block [He *et al.*, 2016] (b) Identity_Block [He *et al.*, 2016].

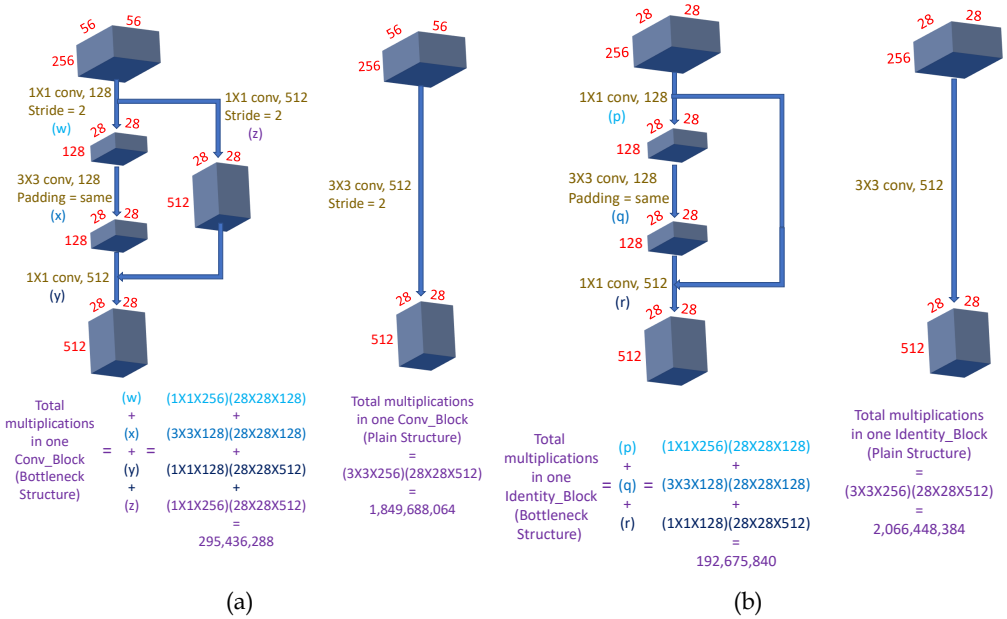


Figure 4.9 : (a) Comparison of multiplications required in the bottleneck structure of Conv_Block and equivalent plain network. (b) Comparison of multiplications required in bottleneck structure of Identity_Block and equivalent plain network.

Where, μ, σ^2 are the mean and variance of one mini-batch. Equation (4.12) is used to normalize the current batch and this is followed for all the batches. δ is used for numerical stability of the equation, when σ^2 becomes zero. The output of the hidden layers can be given by $Z_f^{(i)}$

$$Z_f^{(i)} = \xi Z_{norm}^{(i)} + \gamma \tag{4.13}$$

Here γ and ξ are mean and standard deviation, which are additional learning parameters of the model [Ioffe and Szegedy, 2015].

Batch normalization adds noise to the output as it has not been computed for the complete set of input data. This gives a slight regularization effect to the model. The increase in batch size reduces noise and regularization, but gives an advantage by reducing time to calculate the values of the parameters.

Activation functions

Sigmoid, hyperbolic tangent (tanh), rectified linear unit (ReLU), leaky ReLU are some of the activation functions used in neural networks as per the application's requirement. Sigmoid and tanh activation functions are not preferred in the deep neural network because of the gradient vanishing problem which drags the parameter update process into a large iterative loop. This problem occurs in these activations because they have an S shape relation between input and output with the output range of [0,1] and [-1,1], respectively. Both the functions show good sensitivity in the mid-range i.e. 0.5 and zero for sigmoid and tanh, respectively. Very small and very large values of input drag the output to the extent of the function. This saturation of the functions makes it very difficult to train the model even for the high-end GPU. This problem has been resolved using ReLU activation function by using linear dependency between input and output for input greater than zero ($f(x) = max(0,x)$). For both ResNet-50 and Inception ResNet V2 networks, ReLU activation has been used in all hidden layers, and softmax activation is used in the decision layer of the fully connected network.

4.4.3 Inception ResNet V2

Inception networks have very deep architecture and require more time to train. The training time of these networks has been considerably reduced by introducing residual blocks at concatenation stages. Refer to Figure 4.10 for full model architecture, and Figure 4.11, 4.12 and 4.13 for architecture components. The stem is the first component in the model, which contains a series of Conv2D_BN and MaxPooling blocks, as shown in Figure 4.11(a). Other components are Reduction_A and Reduction_B shown in Figure 4.11(b) and 4.11(c) used to reduce the size of output data to 12×12 and 5×5, respectively. The number of filters, filter size, stride (S), and padding are specified with each operational block.

The diagram shown in Figure 4.12(a) is a generalized schema for the inception_Resnet_X (X=A/B/C) blocks given in Figure 4.13. Where inception block is replaced by Conv2D_BN, which is a subnetwork of traditional Conv2D layers followed by batch normalization and activation. While performing a residual summation of input and inception block output, the depth of both should be the same, for that purpose scaling is used to compensate for the dimensionality reduction induced by the Inception block. The scaling block consists of (1X1) Conv2D layer, and the depth of each Conv2D layer is shown in Figure 4.13. Also, to reduce memory consumption, normalization is done only after traditional layers, not after residual summation. 'Valid' padding is used to get reduced dimension and 'Same' padding is used to get the output dimension the same as the input.

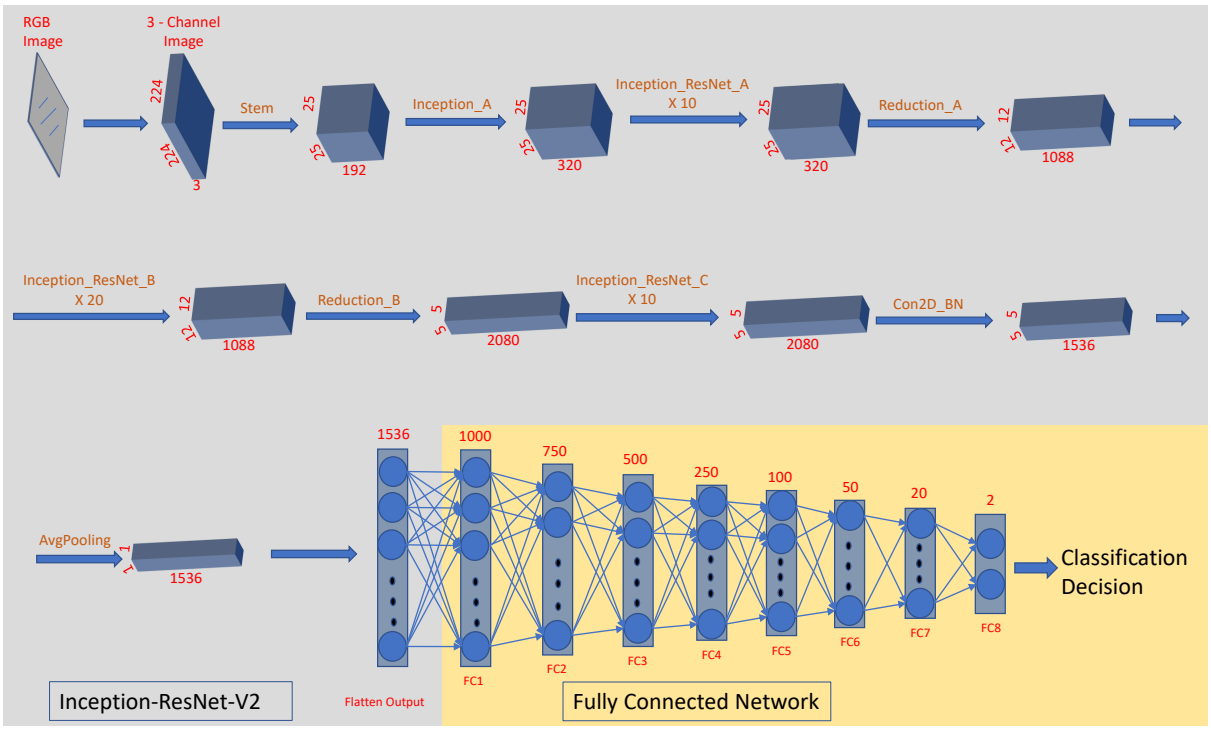


Figure 4.10 : Inception ResNet V2 architecture [Szegedy *et al.*, 2016].

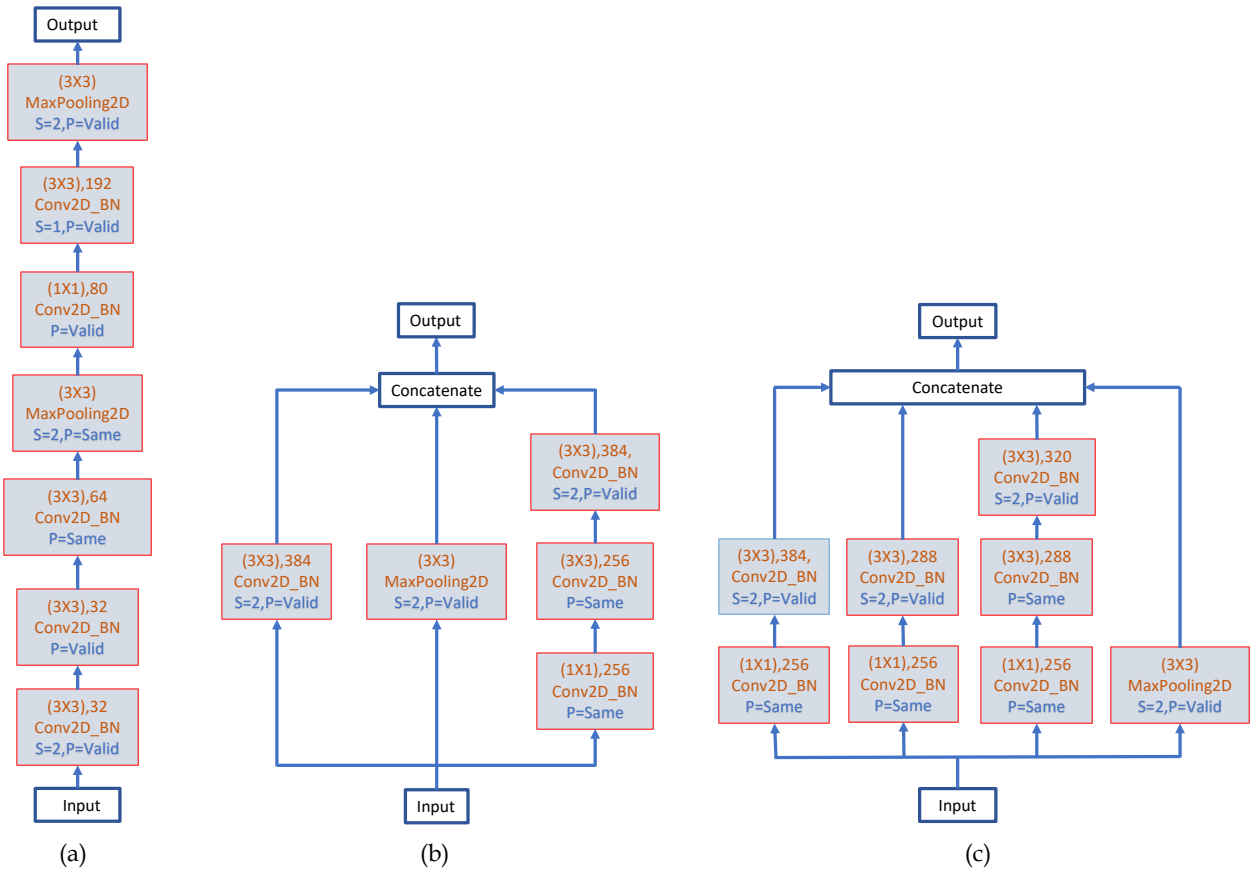


Figure 4.11 : Building blocks of Inception ResNet V2 [Szegedy *et al.*, 2016] (a) Stem, (b) Reduction_A and (c) Reduction_B.

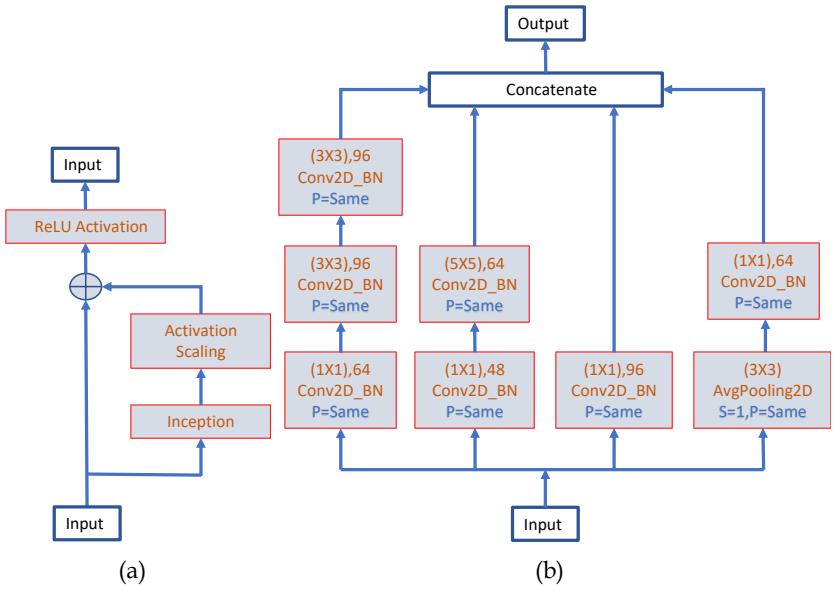


Figure 4.12 : (a) General schema for Inception_ResNet blocks [Szegedy *et al.*, 2016] and (b) Inception_A block [Szegedy *et al.*, 2016].

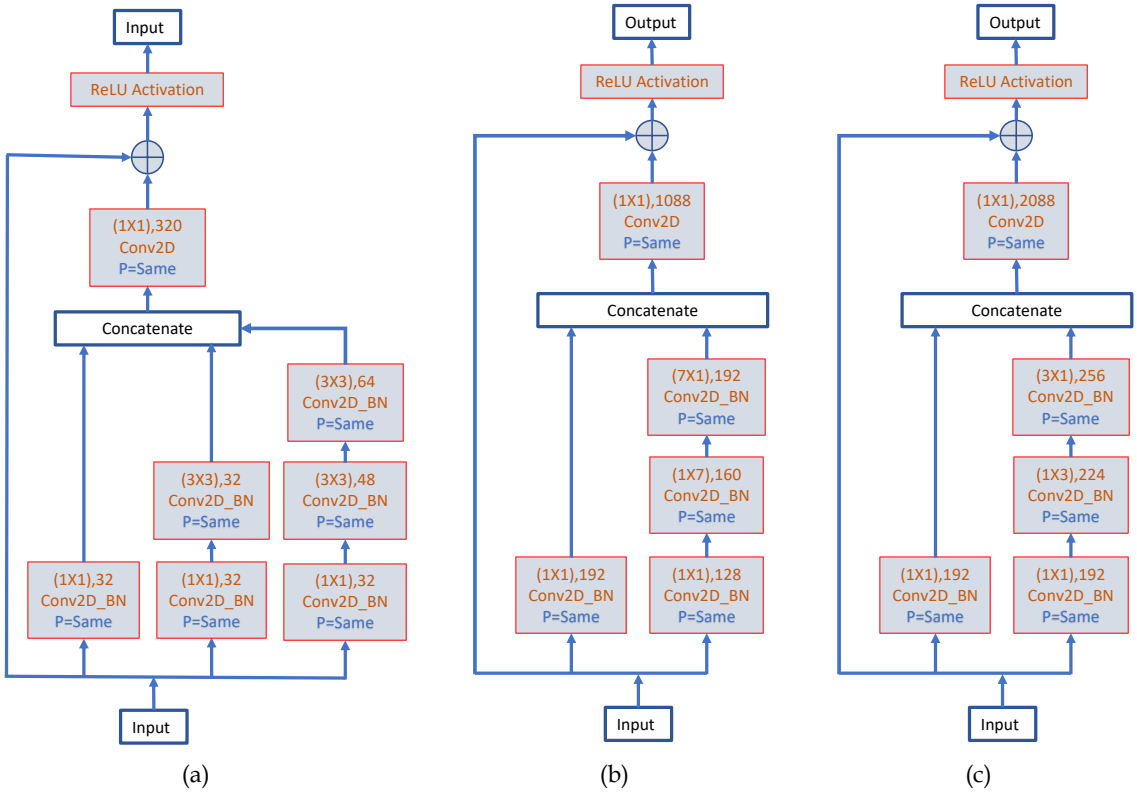


Figure 4.13 : Building blocks of Inception ResNet V2 [Szegedy *et al.*, 2016] (a) Inception_ResNet_A, (b) Inception_ResNet_B, and (c) Inception_ResNet_C.

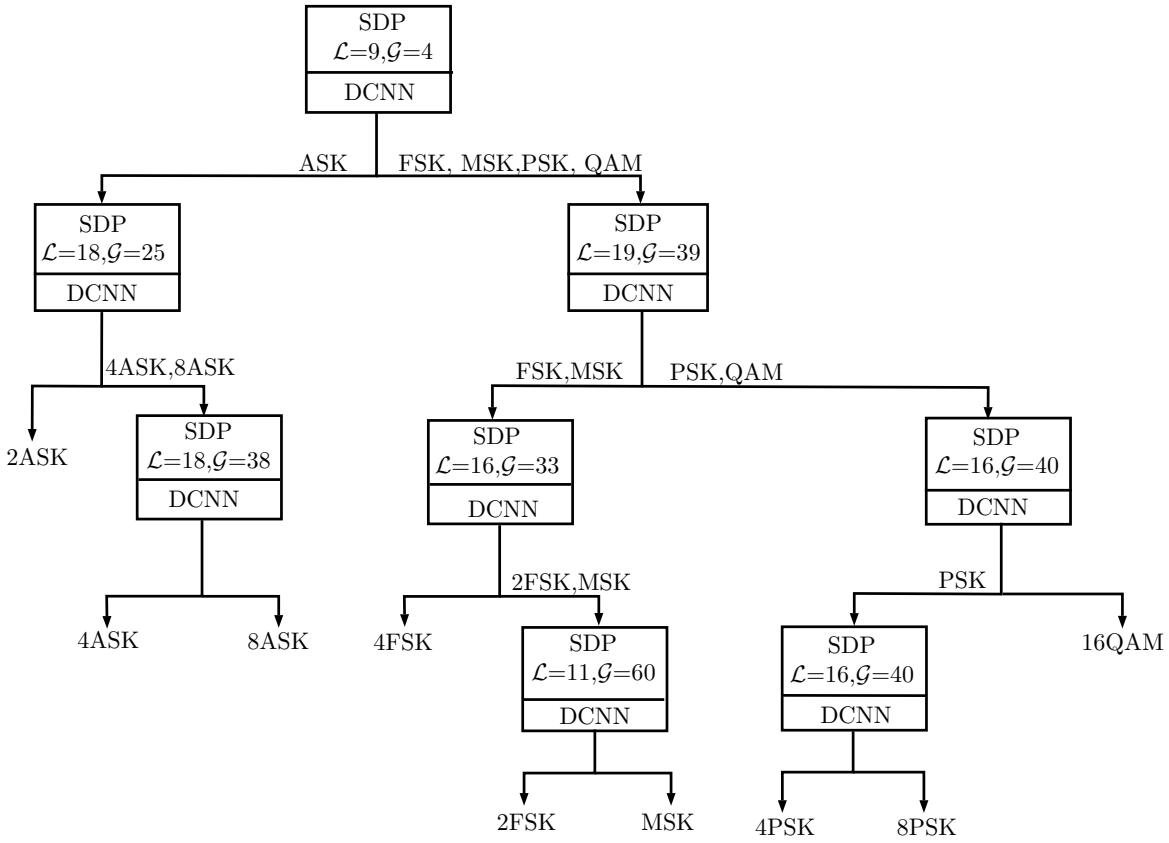


Figure 4.14 : Hierarchy followed for modulation classification.

4.4.4 Training of network

The proposed DCNN modulation classification architecture is trained in the hierarchical manner defined in Figure 4.14. The Hierarchy follows binary classification at each step. The optimum parameters ($\mathscr{L}_{opt}$ and $\mathscr{G}_{opt}$) have been calculated for each individual step using the method defined in section 4.3.5. The description of the training dataset and model parameters is given in table 4.2. For each SNR, 30 SDP images are formed for training. From 0 to 30 dB SNR, 930 training images are formed with a step of 1 dB, and from 35 to 70 dB with a step of 5 dB, 240 images are formed (1170 in total for a single modulation). These images are generated for each modulation scheme for a particular $\mathscr{L}$ and $\mathscr{G}$ value to train the model. At the first level of the hierarchy, the classification between two groups ASK and (FSK, MSK, PSK, and QAM) is done by training the DCNN model with 10,530 (1170 of each modulation scheme in both groups) SDP images generated with $\mathscr{L}$=9 and $\mathscr{G}$=4. The order of ASK is classified in two levels of training for specified $\mathscr{L}$, and $\mathscr{G}$, and similarly, all modulations are classified by the following hierarchy as shown in Figure 4.14. For each Resnet-50 and Inception ResNet V2 model, a total of eight DCNN networks have been trained for the overall classification process.

Table 4.2 : Dataset and system parameters used for DCNN training

Parameters	Values
Amount of Training Data	8424
Amount of Validation Data	2106
SNR Range (dB)	0 to 70
Batch Size	16
Learning Rate	0.0001

4.5 EXPERIMENTS AND RESULTS

4.5.1 System implementation

RF signals are simulated in LabVIEW. Signals are pre-processed to downconvert at an intermediate frequency mentioned in Figure 4.1. These signals are further used to form SDP color images. DCNN model is implemented in python with the Keras libraries. GPU Quadro K2200 and CPU Intel Xeon E5-2640 have been used to train the model for all modulation schemes. All the layers are made trainable. The description of parameters and dataset is given in table 4.2. The model is compiled to create a computation flow graph with Adam optimizer of the learning rate 0.0001. Batch size is taken 16 while training. 80% of data is used for training and 20% is used for validation of the created model in every epoch of training. The model is trained for 10 epochs and model weights are saved when validation accuracy is maximum in between. The initialization of model weights has been done using the fine-tuned model for the ImageNet dataset. The classification process includes two stages: training and testing. The training process requires hours to complete for both the models while testing needs few milliseconds to complete. Fortunately, training needs to be done once for a large dataset. The computation time to get the classification result at each stage of the hierarchy for one signal is found to be 21.5 msec and 28.2 msec for ResNet-50 and Inception ResNet V2, respectively. The computation time is calculated for 1000 signal realizations and averaged for precision.

4.5.2 Performance of classification method

We employed DCNN for the classification of nine modulation schemes listed in table 4.1. Classification results for all modulation schemes are shown in Figure 4.15, 4.16, and 4.17. Classification results for 2ASK, 4ASK, and 8ASK modulation schemes using ResNet-50 and Inception ResNet V2 models are shown in Figure 4.15. Both the networks achieve 100% accuracy above 5 dB SNR. Figure 4.16 shows results for QPSK, 8PSK, and 16QAM. Accuracy for 2FSK, 4FSK, and MSK schemes are shown in Figure 4.17. Results reveal that all the classes are reliably separated above 9 dB SNR for both models. Classification results are given by the cumulative decision of multiple stages. The final probability of classification is obtained by multiplying probabilities at individual stages. As per the hierarchy followed for classification, ASK is separated from other domain modulation schemes in first step of classification and the order-wise classification is done in further steps. The SDP patterns for ASK schemes are different from the others as shown in Figure 4.14, which gives good classification accuracy as given in Figure 4.15. While in further steps, QPSK and 8PSK looks equivalent and 2FSK and 4FSK also looks equivalent in terms of features which cause the accuracy degradation in the last step of classification.

4.5.3 Comparison with existing work

In this section, the accuracy of the proposed classifier is compared with classifiers based on ANN [Popoola and van Olst, 2011], stacked autoencoders [Ali *et al.*, 2017], clustering [Jajoo *et al.*, 2017], and decision-theoretic approach [Nandi and Azzouz, 1998], shown in table 4.3. In [Popoola and van Olst, 2011], classification has been done based on statistical features of the signal, which have been given to the artificial neural network (ANN) of two hidden layers. This method works

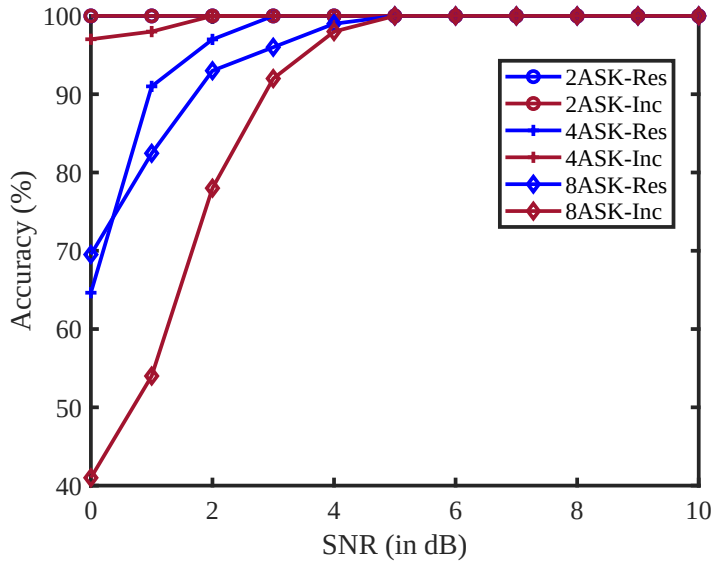


Figure 4.15 : Accuracy versus SNR graph of 2ASK, 4ASK and 8ASK for ResNet-50 (Res) and Inception ResNet V2 (Inc) models.

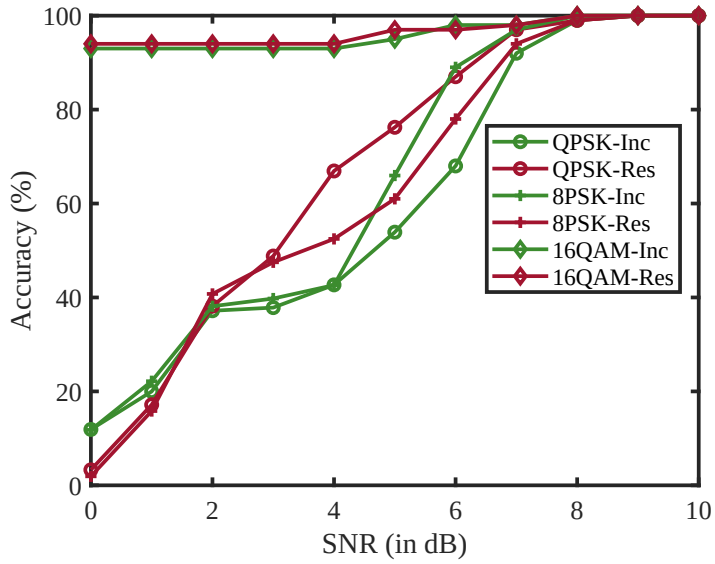


Figure 4.16 : Accuracy versus SNR graph of QPSK, 8PSK and 16QAM for ResNet-50 (Res) and Inception ResNet V2 (Inc) models.

well for lower SNR values, whereas proposed DCNN based classifiers perform better for 9 dB SNR and above. For BPSK, QPSK, and 16QAM, the proposed method achieves better classification accuracy than the method based on stacked autoencoders [Ali *et al.*, 2017]. In [Jajoo *et al.*, 2017], the authors have used constellation to detect modulation. OPTICS (Ordering Points To Investigate the Clustering Structure) has been used for estimation of the order of modulation and k-means clustering method for domain identification. This method got 96% classification accuracy at 10 dB SNR for BPSK, QPSK, and 16QAM. For BPSK, QPSK, 8PSK, 4ASK, 8ASK, and 16 QAM average

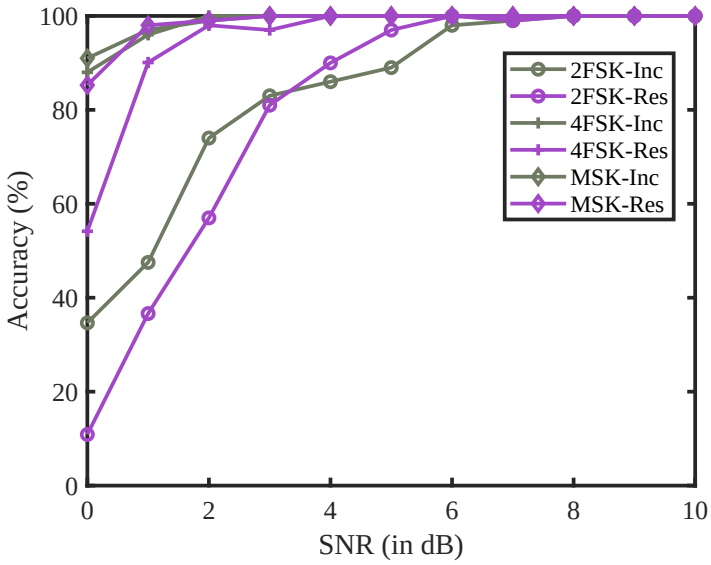


Figure 4.17 : Accuracy versus SNR graph of 2FSK, 4FSK and MSK for ResNet-50 (Res) and Inception ResNet V2 (Inc) models.

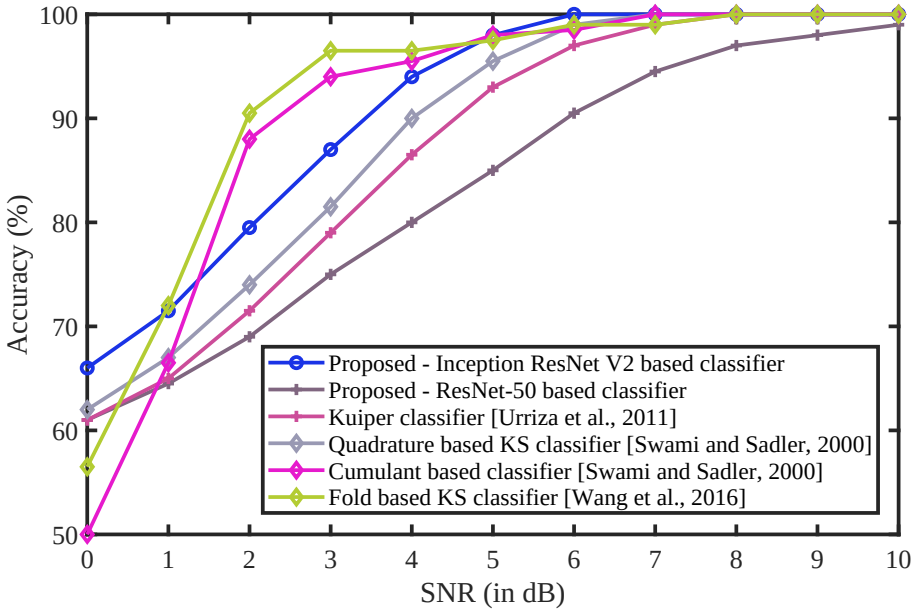


Figure 4.18 : Average classification accuracy of QPSK and 16QAM.

accuracy at 20 dB SNR is 81.33%, whereas the proposed method got better results than both the combinations of modulation schemes.

In [Nandi and Azzouz, 1998], classification has been done with two methods, decision-theoretic approach, and ANN-based. Classification results of their work show that ANN is better than the decision-theoretic approach, whereas our proposed DCNN models work better

Table 4.3 : Comparison of results with existing methods

Modulation Schemes	Method	SNR (dB)				
		0	5	10	15	20
2ASK, 4ASK, 2FSK, QPSK	[Popoola and van Olst, 2011]	**99.75**	**99.91**	99.87	-	99.94
	Proposed - ResNet-50	77.50	99.25	**100**	**100**	**100**
	Proposed - Inception ResNet V2	82.25	96.75	**100**	**100**	**100**
BPSK, QPSK, 16QAM	[Ali *et al.*, 2017]	46.67	89	99.93	**100**	**100**
	[Jajoo *et al.*, 2017]	-	66.33	96	**100**	**100**
	Proposed - ResNet-50	68.67	**98.67**	**100**	**100**	**100**
	Proposed - Inception ResNet V2	**72.67**	98.33	**100**	**100**	**100**
2ASK, 4ASK	Decision theoretic [Nandi and Azzouz, 1998]	-	-	-	99.05	98.5
	Proposed - ResNet-50	68.67	**98.67**	**100**	**100**	**100**
	Proposed - Inception ResNet V2	**72.67**	98.33	**100**	**100**	**100**
2FSK, 4FSK	Decision theoretic [Nandi and Azzouz, 1998]	-	-	-	99.65	98.75
	Proposed - ResNet-50	68.67	**98.67**	**100**	**100**	**100**
	Proposed - Inception ResNet V2	**72.67**	98.33	**100**	**100**	**100**
BPSK, QPSK, 8PSK, 4ASK, 8ASK, 16QAM	[Jajoo *et al.*, 2017]	-	48.33	81.33	88.33	100
	Proposed - ResNet-50	**64.24**	89.04	**100**	**100**	**100**
	Proposed - Inception ResNet V2	59.21	**89.33**	**100**	**100**	**100**

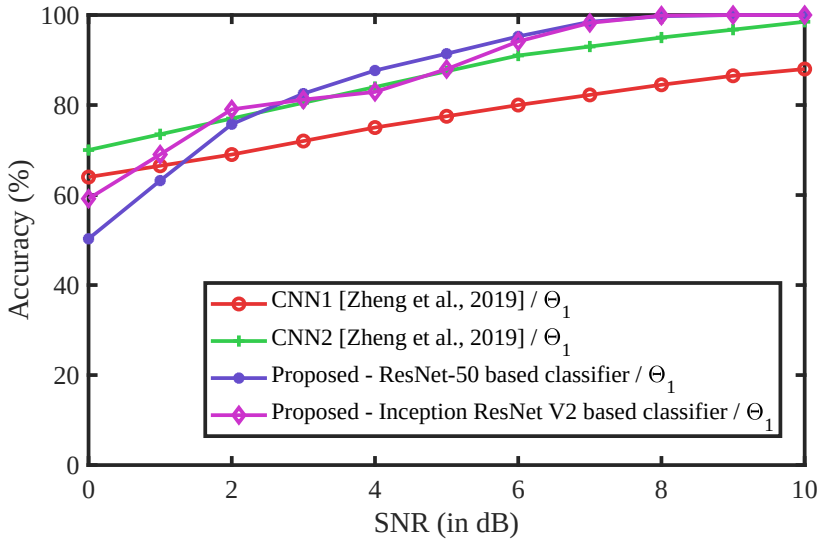


Figure 4.19 : Comparison of average classification accuracy of proposed methods with CNN1 and CNN2.

than these for selected modulation schemes. Results for (2ASK, 4ASK) and (2FSK, 4FSK) are compared with the proposed method at 15 dB and 20 dB SNR, given in table 4.3. Comparison of average classification accuracy of QPSK and 16QAM for present research and four previous research works is reported in Figure 4.18.

In [Aslam *et al.*, 2010], the authors have developed the Genetic Programming in combination with K - Nearest Neighbor (GP-KNN) method, where new features were generated using genetic programming and classification has been done using KNN. The proposed method provides better classification accuracy than GP-KNN for BPSK, QPSK, and 16QAM, for an equal number of symbols and results are compared in table 4.4. In [Hiremath *et al.*, 2019], 10 modulation

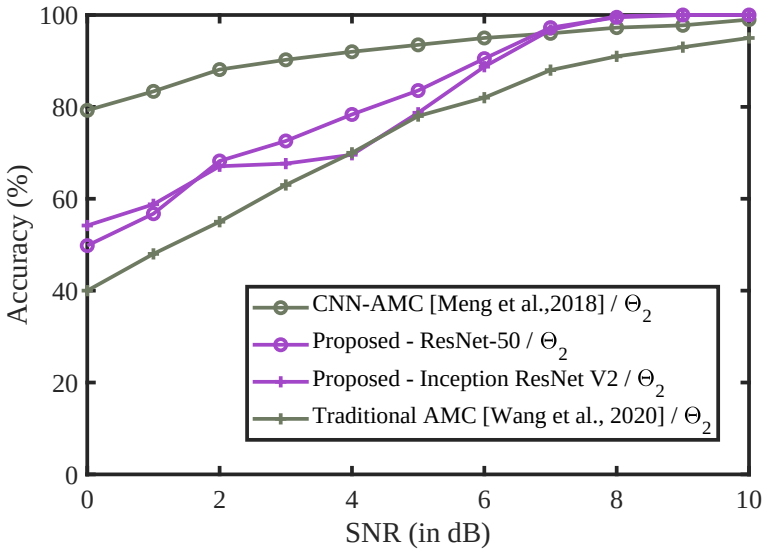


Figure 4.20 : Comparison of average classification accuracy of proposed methods with CNN-AMC and traditional AMC methods.

Table 4.4 : Comparison of results with GP-KNN

Modulation Schemes	Method	SNR (dB)		
		4	12	20
BPSK, QPSK, 16QAM	GP-KNN [Aslam *et al.*, 2010]	92.72	99.90	99.99
	Proposed - ResNet-50	**100**	**100**	**100**
	Proposed - Inception ResNet V2	**100**	**100**	**100**

schemes are classified using Stockwell transform where the DL model has been proposed and trained with extended labels of 3-channel images to prepare the model for varying SNR. The authors have given training to their proposed model for the range of SNR (-8 dB, 8 dB). For the SNR of 8 dB, their proposed model doesn't provide 100% accuracy, which concludes that possibly it won't be able to provide reliable classification accuracy for a higher range of SNR. While our proposed model has been given training for 0 dB to 70 dB SNR and it provides 100% average classification accuracy above 9 dB SNR. Hierarchical ANN has been employed in [Louis and Sehier, 1994] for classification between 10 modulation schemes and gives classification accuracy of 90% for 50 dB SNR. The proposed methods outperform for the same set of modulation schemes.

In DCNN models, mostly constant length of the signal is required when IQ data is used for modulation detection. When the signal length is more than the required input length, authors in [Zheng *et al.*, 2019] have suggested three fusion methods viz. voting-based fusion, confidence-based fusion, and feature-based fusion to fully utilize the signal length for improvement in the system performance. Two DCNN models are given in [Zheng *et al.*, 2019], CNN1, and CNN2 to classify a set of modulation schemes. In Figure 4.19, performance of proposed models is compared with CNN1 and CNN2 for average classification accuracy of three proposed fusion methods, where considered modulation schemes are $\{\Theta_1 \in BPSK, QPSK, 8PSK, 2FSK, 4FSK, 4ASK, 8ASK, 16QAM\}$. In [Meng *et al.*, 2018], a DCNN model has been proposed for modulation classification by considering raw IQ data and estimated SNR of the received signal as an input. Also, two-step training including pre-training and fine-tuning is performed to reduce the

training time. The average classification accuracy of CNN-AMC, proposed in [Meng *et al.*, 2018] is compared with our proposed methods in Figure 4.20 for $\{\Theta_2 \in BPSK, QPSK, 8PSK, 16QAM\}$ modulation schemes. The classification accuracy of the developed method in [Wang *et al.*, 2020b] is compared with the traditional AMC method. The traditional AMC method takes higher-order cumulants (HOC) as input and SVM for classification. The classification results of the traditional method are compared with our proposed methods in Figure 4.20.

…