

# ON COMPLEX-DOMAIN CNN REPRESENTATIONS FOR CLASSIFYING REAL/COMPLEX-VALUED DATA

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

This paper is about complex-valued CNNs (CV-CNNs) for computer vision that use representations that are complex-valued instead of real-valued. We divide input data into three categories: inherently real-valued, inherently complex-valued, and complex-valued obtained by transforming real-valued. We study the question whether complex-valued representation of CV-CNNs offers any advantages over the commonly used real-valued CNNs (RV-CNNs). For concreteness, we focus on the classification task. The existing literature offers contradictory answers to our question. We find that this is mainly because (a) they seldom employ a common performance measure (e.g., CV-CNN compared against RV-CNN with similar network structure vs similar number of parameters) (b) diversity of evaluation datasets used are limited (e.g., datasets in which magnitude information is more, less or as important as phase information) (c) less effort has been devoted to reduce the randomness in training between CV-CNN and RV-CNN. Towards this, we propose performance measures based on similar network structure, number of parameters and number of MAC operations. Also, we consider diverse datasets with varying magnitude/phase information, and deal with the randomness in training. As a result, we expect that any observed performance differences will be independent of the above disparities, and arise from the use of real vs complex representations. Theoretically, we show that, unlike RV-CNNs, CV-CNNs can preserve magnitude and phase through intermediate stages of processing. Our main experimental findings are the following. (i) As network depth decreases – the performance of CV-CNNs improves with respect to similar network structure; the performances of CV-CNN and RV-CNN having a similar number of parameters become more comparable; and the performance of RV-CNNs improves with respect to similar number of MAC operations; (ii) The above performance differences diminish as the network depth increases. (iii) With respect to data diversity, performance depends on whether the dataset has dominant magnitude or phase, i.e., whether reconstruction error is lower using only magnitude or only phase. If a complex-valued data has dominant magnitude, instead of providing real and imaginary parts as input, providing the magnitude part produces significant performance gain, whereas if the data has dominant phase, providing both real and imaginary parts is important. This holds true for different network depths.

## 1 INTRODUCTION

The types of data most in use in computer vision and image processing include three types: (a) inherently real-valued data such as image and audio signals, (b) inherently complex-valued data such as SAR, SONAR, and MRI data, and (c) (pseudo)complex-valued data obtained by transforming real-valued data such as FFT and STFT outputs. Some practical applications of the inherent complex-valued data include MRI signal processing (Virtue et al., 2017), wind prediction (Sepasi et al., 2017), radio modulation and SAR image classification (Chakraborty et al., 2019), whereas pseudo complex-valued data are used for speech enhancement (Tsuzuki et al., 2013), texture modelling (Ghanem & Ahuja, 2007) and filter design (Freeman et al., 1991). The relative significances of the magnitude and phase parts of a complex dataset may vary with problems, e.g., spectral phase carries more information than spectral magnitude in natural images (Pearlman & Gray, 1978) while spectral magnitude is more significant than spectral phase in speech signals (Li et al., 2022). Thus,

it makes more sense to employ a complex-valued model for complex-valued data to capture the relative roles of magnitude and phase Lee et al. (2022).

The simultaneous use of magnitude and phase, i.e., complex-valued representations has received limited attention in the learning literature, in neural networks in general, as well as in deep learning. Hirose (2012; 2009) discusses the magnitude-vs-phase importance for designing complex-valued neural networks for complex-valued signals. In deep learning also, for example for the common problem of classification, most work has focused on real-valued CNNs (RV-CNNs). There is limited work on complex-valued CNNs (CV-CNNs) Lee et al. (2022). CV-CNNs have been developed as an extension of RV-CNN, by employing complex-domain versions of RV-CNN components (Trabelsi et al., 2018). Lee et al. (2022); Bassey et al. (2021) present a comprehensive survey of CV-CNN and its historical development. Several works have investigated the question of whether CV-CNNs add any value to RV-CNNs at all. Interestingly, the conclusions of these papers often contradict each other. For example, Drude et al. (2016) claims that the use of complex-valued representations can hardly be justified. Trabelsi et al. (2018) show that CV-CNN performance is only comparable to that of RV-CNN. Huang et al. (2020) shows that complex-valued input in CV-CNNs performs inferior to that when magnitude is input to RV-CNNs. On the contrary, Yao et al. (2020); Gu & Ding (2018); Popa (2017); Popa & Cernăzanu-Glăvan (2018) claim that CV-CNNs perform significantly better than RV-CNNs. We find that these contradictions mainly arise due to the following reasons:

1. Different works derive *general* conclusions based on different performance measures, e.g., CV-CNN is compared against an RV-CNN having similar network structure Yao et al. (2020); Gu & Ding (2018); Popa & Cernăzanu-Glăvan (2018), similar number of real-valued parameters Drude et al. (2016); Trabelsi et al. (2018); Huang et al. (2020). Albeit for MLPs, (Barrachina et al. (2021); Hirose (2009) consider similar number of MAC operations.
2. For MLP, Barrachina et al. (2021) show that performance-difference depends on complex-valued data properties. However, less attention has been devoted in this aspect for CV-CNN, e.g., Gu & Ding (2018); Popa (2017); Trabelsi et al. (2018) confine to real-valued data, Yao et al. (2020); Huang et al. (2020) confine to inherent complex-valued data, whereas Popa & Cernăzanu-Glăvan (2018) confine to FFT of real-valued data.
3. Less effort has been devoted to reduce the randomness in training between CV-CNN and its corresponding RV-CNN baseline(s). Main factors are random initialization, data augmentation, data shuffling, and stochastic layers (e.g., dropout) (Zhuang et al., 2022).

As these methods employ different performance measures and disparate network architectures and/or datasets, it is difficult to consolidate results of these works. The use of the extra information known to be captured in complex-valued representations, combined with the contradictions seen in the conclusions to date motivate a more conclusive analysis, and have motivated the work reported in this paper. Specifically, we study the merits of complex-valued representation of CV-CNNs for classification task (factoring in the Points 1-3). Our main contributions can be summarized as:

- We consider all three performance measures to evaluate CV-CNN (i.e., similar network structure, number of parameters and number of MAC operations). For each measure, theoretically we reveal the merits/demerits of CV-CNN’s layers in comparison with RV-CNN. We show that CV-CNNs, unlike RV-CNNs, can preserve magnitude-and-phase through ReLU and Max-pooling layers (as RV-CNNs work for real-valued inputs).
- Our comprehensive empirical study, based on all three performance measures and several networks while accounting for the randomness in training, to find the merits of complex-valued representations of CV-CNNs for classification task leads to definitive answers.
- Our empirical study with different datasets (real-valued: CIFAR-10, CIFAR-100; complex-valued: MSTAR and several transformations of CIFAR-10 and CIFAR-100 with varying magnitude/phase information) reveals the dependency of CV-CNN to the magnitude/phase information in data. This advocates suitable choice of network for different datasets.
- Our work resolves various contradictions in the current literature regarding the merits of complex-valued representations. Our findings corroborate and add good value to several existing works that are marred by the contradictions to date.

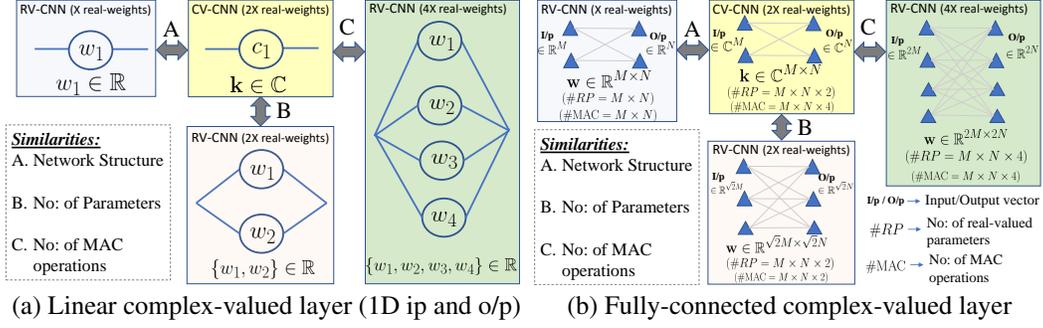


Figure 1: (a) Different performance measures with which a CV-CNN can be compared against RV-CNNs. Case A compares based on similar network structure (or similar graph), Case B compares based on similar number of real-valued training parameters, and Case C compares based on similar number of MAC operations. Figs. (a-b) consider two complex-valued layers: (a) Complex-valued linear layer with one-dimensional complex-valued input and output. (b) Complex-valued linear layer with  $M$  dimensional complex-valued input and  $N$  dimensional complex-valued output.

Rest of this paper is organized as follows: Section 2 theoretically differentiate, based on each performance measure, intermediate layers of CV-CNNs and RV-CNNs. The main objective is to reason why one network may perform better than the other. In addition, here we discuss the magnitude-and-phase preservation property inherent to CV-CNN. Section 3 presents our empirical study. We consider experiments on inherent real-valued datasets, inherent and pseudo complex-valued datasets; discuss our results in Section 3.1 and conclude in Section 4.

## 2 COMPLEX-VALUED CNN AND ITS RELATION TO REAL-VALUED CNN

In this section, based on each performance measure, we study how CV-CNN’s intermediate layers differ from that of corresponding RV-CNN’s layers. For comparing CV-CNN and RV-CNN, we consider identical ordering of layers in both networks (which is a standard practice (Guberman, 2016; Trabelsi et al., 2018; Gu & Ding, 2018; Popa, 2017)). This enables a divide and conquer analysis scheme for an otherwise complicated end-to-end mapping. Then we show that complex-valued ReLUs and Max-pooling are required to preserve the magnitude-and-phase of complex-valued input (as similar to ReLU and Max-pooling operate in RV-CNN for real-valued inputs).

We discussed three performance measures to compare CV-CNN and RV-CNN, i.e., based on similar network structure, similar number of real-valued parameters, and similar number of real-valued MAC operations. First, we study these different measures using two different layers: (a) A simplistic complex-valued linear layer with one-dimensional input and output; (b) A complex-valued fully-connected layer with multi-dimensional input and output (Fig. 1).

The complex-valued linear layer performs a scalar multiplication of a complex-valued input  $x_C$  with a trainable complex-valued parameter  $k_C$  (see the yellow box of Fig. 1(a)). The input  $x_C \in \mathbb{C}$  can be equivalently represented by an ordered pair of real numbers in  $\mathbb{R}^2$  (Hirose, 2009)

$$x_C \equiv \begin{Bmatrix} x_r \\ x_i \end{Bmatrix}, \quad (1)$$

where subscript  $C$  indicates complex values, and subscripts  $r$  and  $i$  indicate the real and imaginary parts of complex number, respectively. With this convention, the complex-valued linear layer transforms the complex-valued scalar input  $x_C$  as

$$y_C = k_C \cdot x_C = \begin{Bmatrix} k_r x_r - k_i x_i \\ k_i x_r + k_r x_i \end{Bmatrix} \triangleq \begin{Bmatrix} k_r & -k_i \\ k_i & k_r \end{Bmatrix} \begin{Bmatrix} x_r \\ x_i \end{Bmatrix}, \quad (2)$$

where  $k_r$  and  $k_i$  are the real and image parts of the trainable parameter  $k_C$ . Using equation 2, we can see that this layer has two real-valued parameters ( $k_r$  and  $k_i$ ), and has four real-valued MAC operations ( $k_r \cdot x_r + 0$ ,  $-k_i \cdot x_i + 0$ ,  $k_i \cdot x_r + 0$ , and  $k_r \cdot x_i + 0$ ).

Next we focus on the Complex-valued fully-connected layer (FCL). It maps a  $M$  dimensional complex-valued input to a  $N$  dimensional complex-valued output using a learnable complex-valued weight matrix of dimension  $M \times N$  (see the yellow box in Fig. 1(b)). Thus it has  $M \times N \times 2$  real-valued parameters. Because this layer has  $M \times N$  complex-valued multiplication and each complex multiplication has 4 MAC operations (equation 2), it has  $M \times N \times 4$  MAC operations. Note that convolution layer is a special case of FCL with locally connected shared weight.

In what follows, we consider each performance measure one by one, and study the difference of the layers discussed above with corresponding layers of RV-CNN. The main objective is to gather some insights regarding which network may perform well over the other (rather than empirical study).

## 2.1 SIMILAR NETWORK STRUCTURE

The network graph of complex-valued linear layer in Fig. 1(a) is a single input node and a single output node, with a scaling in between. Therefore, a real-valued layer with the similar structure is of the form  $y = w_1 * x$ ,  $\{y, x, w_1\} \in \mathbb{R}$  (see Case A in Fig. 1(a)). This equation is a constrained case of equation 2 ( $k_i = x_i = 0$ ). This implies that the complex-valued layer has better representation capability as compared to the real-valued layer (e.g.,  $k_i \neq 0$  and/or  $x_i \neq 0$ ).

The complex-valued FCL maps a  $M$  dimensional complex-valued input to a  $N$  dimensional complex-valued output using a complex-valued matrix of dimension  $M \times N$ . Therefore based on similar structure, real-valued FCL maps a  $M$  dimensional real-valued input to a  $N$  dimensional real-valued output using a real-valued matrix of dimension  $M \times N$ . From Case A in Fig. 1(b), real-valued FCL is a constrained case of complex-valued FCL via imaginary parts of input and weight constrained to zero. As a result, intuitively, *based on similar network structure, complex-valued network may perform better than real-valued network due to its high representation capability.*<sup>1</sup>

## 2.2 SIMILAR NUMBER OF PARAMETERS

The complex-valued linear network has two real-valued parameters ( $k_r$  and  $k_i$  in equation 2). A real-valued network with two parameters that resembles a multiplication layer is Hirose (2009)

$$\begin{Bmatrix} y_r \\ y_i \end{Bmatrix} = \begin{Bmatrix} k_r x_r \\ k_i x_i \end{Bmatrix} \triangleq \begin{Bmatrix} k_r & 0 \\ 0 & k_i \end{Bmatrix} \begin{Bmatrix} x_r \\ x_i \end{Bmatrix}. \quad (3)$$

Comparing equation 2 and equation 3, it is clear that complex-valued linear layer cannot represent the input-to-output mappings of the real-valued linear layer and vice-versa. First we introduce two properties. Property 1 is that the real/imaginary part of output in Equation 2 is *dependent on both* the real and imaginary part of input (i.e.,  $y_r = k_r x_r - k_i x_i$  and  $y_i = k_i x_r + k_r x_i$ ). Property 2 is that the real/imaginary part of output is independent of imaginary/real part (e.g.,  $y_r = a x_r$  and  $y_i = b x_i$ , where  $a \neq b$ ). Note that equation 2 satisfies Property 1, but cannot always satisfy Property 2. In contrast, equation 3 satisfies Property 2, but cannot always satisfy Property 1. Thus there exist complementary qualities for the two layers.

For FCL too, there exist complimentary qualities. Complex-valued FCL has  $M \times N \times 2$  real-valued parameters. Real-valued FCL with similar number of parameters, with a fixed input-and-output aspect ratio, is a mapping from  $\sqrt{2}M$  to  $\sqrt{2}N$  dimensional real-valued vectors (so that number of parameters becomes  $M \times N \times 2$ ). Based on the real-valued representation of complex values in equation 1, complex-valued FCL can be considered as a mapping from  $2M$  to  $2N$  dimensional real-valued vector (see Case B in Fig. 1(b)). Thus complex-valued FCL exhibits a high-dimensional input-output mapping than the real-valued FCL ( $2M \rightarrow 2N$  versus  $\sqrt{2}M \rightarrow \sqrt{2}N$ ). But the downside is that the mapping is constrained with half the input dimension ( $= M$ ) due to complex multiplication (Property 1). In contrast, real-valued FCL exhibits an unconstrained mapping (Property 2) from a higher input dimension ( $\sqrt{2}M$  versus  $M$ ). *Due to the complimentary nature, the performance differences of CV-CNN and RV-CNN may be comparable if the complimentary qualities have similar effect, or the performances may differ if one quality dominates over the other.*

<sup>1</sup>In Secs. 2.1-2.3, we assume that the network training does not get stuck in some undesirable local minima.

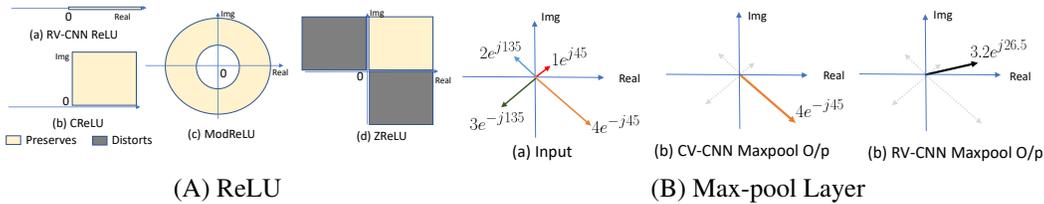


Figure 2: Magnitude-and-Phase Preservation in CV-CNN Non-Linearities (A) RV-CNN ReLU, CReLU and ModReLU preserves magnitude-and-phase information of a set of values, while suppressing other values to zero. But ZReLU distorts magnitude-and-phase in values present in second and third quadrant. (B) Comparison of CV-CNN and RV-CNN max-pooling layer. The former preserves the magnitude-and-phase information whereas the latter severely distorts it.

### 2.3 SIMILAR NUMBER OF MAC OPERATIONS

The complex-valued linear network has four MAC operations (equation 2). A real-valued network with four MAC operations that resembles a multiplication layer is Hirose (2009)

$$\begin{Bmatrix} y_r \\ y_i \end{Bmatrix} = \begin{Bmatrix} a & b \\ c & d \end{Bmatrix} \begin{Bmatrix} x_r \\ x_i \end{Bmatrix}, \quad (4)$$

where  $\{a, b, c, d\} \in \mathbb{R}$  are trainable real parameters. Note that the equation 2 of complex-valued network is a constrained case of the above equation ( $a = d = k_r$  and  $-b = c = k_i$ ), which means real-valued analogue has better representation capability (e.g.,  $a \neq d$  and/or  $-b \neq c$ ).

Complex-valued FCL has  $M \times N$  complex-valued multiplication, and hence possess  $M \times N \times 4$  MAC operations. Real-valued FCL with similar number of MAC operations, with a fixed input-and-output aspect ratio, is a mapping from  $2M$  to  $2N$  real-valued dimensions (so that number of MAC operations become  $M \times N \times 4$ ). In Sec. 2.2, we have discussed that complex-valued FCL exhibits a high-dimensional mapping from  $2M$  to  $2N$  real-valued vector, but the drawback is that this mapping is constrained due to complex multiplication. In contrast, here real-valued FCL exhibits the same high dimensional mapping ( $2M$  to  $2N$ ), and in addition there exists no such constraints or drawback. Compare this with the case of similar number of parameters (Sec. 2.2), real-valued FCL in itself has both the complementary qualities. As a result, intuitively, *based on similar network structure, RV-CNN may perform better than CV-CNN due to its high representation capability*. Note that this is in stark contrast with the case of similar structure measure (Sec. 2.1) where CV-CNN has better representation capability, and hence may perform better.

Based on the measure of similar number of MAC operations, we have seen that complex-valued linear layer and fully connected layer can be represented by respective real-valued layers. Please refer to Table2 in our supplementary material that summarizes the representations of these two layers, as well as other standard layers such as convolution, max-pooling, ReLUs, etc. One important observation from the Table is that there exists no equivalent RV-CNN representations for several non-linearities of CV-CNN, which the next section addresses.

### 2.4 NON-LINEARITIES IN COMPLEX-VALUED CNN VERSUS REAL-VALUED CNN

Here we consider ReLUs and Max-pooling, that add non-linearity into a neural network. ReLU is typically the preferred activation function because it is easier to train and achieves good performance (Szandafa, 2021). The ReLU activation of RV-CNN is an element-wise function (i.e., it acts on each element independently), which is

$$\text{ReLU}(x) = \max(0, x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}. \quad (5)$$

One main property of the above RV-CNN ReLU, and that is satisfied by most of the existing complex-ReLUs Trabelsi et al. (2018), is the preservation of input magnitude and phase for a set of non-zero inputs ( $\mathbb{P}$ ) and the suppression of input to zero for the complement of that set ( $\mathbb{P}^C$ ). For

example, as shown in Fig. 2(a), RV-CNN ReLU preserves the positive axis, CReLU preserves the first quadrant, and ModReLU preserves a circular segments. Note that  $\mathbb{P}$  must contain atleast one non-zero element while not including all non-zero values of  $\mathbb{C}$  so as to have a non-linearity property. In this regard, we present the following remark:

**Remark 1:** *Any real-valued non-linear activation cannot preserve magnitude and phase of complex-valued inputs in a set  $\mathbb{P} \subset \mathbb{C} \setminus \{0\}$ , while suppressing the inputs to zero for the set  $\mathbb{P}^C$  (In words, complex-valued activations are necessary for magnitude-and-phase preserving nonlinearity.)*

Before providing a formal proof, we attempt to provide an example using the RV-CNN ReLU. As equation 5 is an element-wise function, its complex equivalent form (equation 1) is

$$\begin{cases} out_r \\ out_i \end{cases} = \begin{cases} \text{ReLU}(x_r) \\ \text{ReLU}(x_i) \end{cases}. \quad (6)$$

This mapping is shown as zReLU in Fig. 2(a), where it preserves the magnitude-and-phase of inputs in the first quadrant and suppresses the fourth quadrant inputs to zero; but it also produces output with *non-zero* magnitude for second/ forth quadrant inputs but with its imaginary/real parts as zero, thereby distorting their magnitude-and-phase.

*Proof:* We prove using proof by contradiction. Assume that there exists such a real-valued activation  $f(\cdot)$ . The preservation of magnitude-and-phase of input implies  $\tan^{-1}(\frac{f(x_i)}{f(x_r)}) = \tan^{-1}(\frac{x_i}{x_r})$  and  $f(x_i) = x_i; f(x_r) = x_r$ . Combining these expressions, we get

$$\frac{f(x_i)}{x_i} = \frac{f(x_r)}{x_r} = 1, \forall \begin{cases} x_r \\ x_i \end{cases} \in \mathbb{P}. \quad (7)$$

Suppression of input to zero implies

$$f(x_i) = f(x_r) = 0, \forall \begin{cases} x_r \\ x_i \end{cases} \in \mathbb{P}^C. \quad (8)$$

Set  $\mathbb{P}$  can be equivalently represented as  $\begin{cases} x_r \in \mathbb{P}_1 \subseteq \mathbb{R} \\ x_i \in \mathbb{P}_2 \subseteq \mathbb{R} \end{cases}$ . The consideration that  $\mathbb{P}$  must contain atleast one non-zero element while not including all non-zero values of  $\mathbb{C}$  implies that  $\mathbb{P}_1$  or  $\mathbb{P}_2$  contains atleast one non-zero element while not including all non-zero values of  $\mathbb{R}$ . As  $x_i$  and  $x_r$  are interchangeable in equation 7,  $\mathbb{P}_1$  must be equal to  $\mathbb{P}_2$ . Hence,  $\mathbb{P}_1$  and  $\mathbb{P}_1^C$  must contain atleast one non-zero element. Consider a nonzero element  $x'_r \in \mathbb{P}_1$  and  $x'_i \in \mathbb{P}_1^C$ . Clearly, the above element is not in  $\mathbb{P}$  because its imaginary part  $x'_i$  is not an element of  $\mathbb{P}_2$  (as  $\mathbb{P}_1 = \mathbb{P}_2$ ). As  $x'_r + jx'_i \in \mathbb{P}^C$ , from equation 8,  $f(x'_r) = f(x'_i) = 0$ . But as  $x'_r \in \mathbb{P}_1$ , equation 7 implies  $f(x'_r) = x'_r \neq 0$ , which is a contradiction. ■

Next we focus on max-pooling. It is used in classification task to aggregate dominant feature-maps from a local region in order to present a larger receptive field for subsequent layers. It selects the maximum value from a non-intersecting spatial window  $W$  of size  $K \times K$ , i.e.,

$$\mathbf{Y} = \max(\mathbf{X})|_W, \quad (9)$$

where  $|_W$  indicates the sliding window. For  $\mathbf{X} \in \mathbb{R}^{B \times M \times N}$ , the output  $\mathbf{Y} \in \mathbb{R}^{B \times M/K \times N/K}$ .

As the complex numbers does not belong to an ordered field, magnitude values are employed for the maximum, while preserving the phase Trabelsi et al. (2018), i.e.,

$$\mathbf{Y}_C = \left( \max(\|\mathbf{X}_C\|)|_W \right) e^{j\theta_{\mathbf{X}_C}}. \quad (10)$$

Employing the standard max-pooling layer of RV-CNN to the complex equivalent form becomes

$$\begin{cases} \mathbf{out}_r \\ \mathbf{out}_i \end{cases} = \begin{cases} \max(\|\mathbf{X}_r\|)|_W \\ \max(\|\mathbf{X}_i\|)|_W \end{cases}. \quad (11)$$

Clearly, there exists *no* equivalence between max-pooling in CV-CNN and RV-CNN, and more important, the latter seriously distorts the amplitude and phase information of the input. That is, unlike equation 10, equation 11 admits a one-to-one correspondence from the output to input (following the notion of max-pool of selecting important feature-map values). Further, magnitude-and-phase of output via Eq. equation 11 can have *no* relation with the input (this is illustrated in Fig. 2(B)).

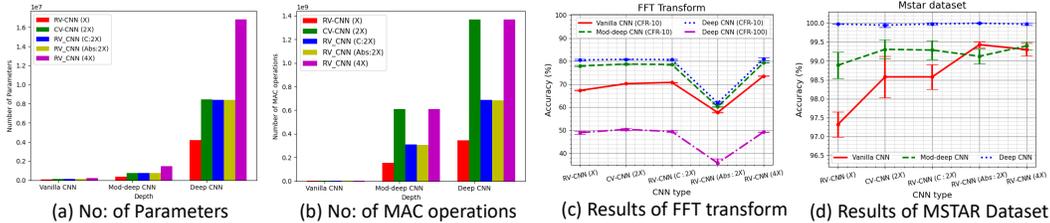


Figure 3: (a-b) Comparison of different performance measures based on number of parameters and MAC operations. Note that similarity based on one performance measure creates a dissimilarity in the other ones. (c-d) Results of complex-valued datasets: FFT and MSTAR. Note that the performance of CV-CNN with RV-CNN depends on different performance measures. Further, for the FFT case, the magnitude input to RV-CNN (RV-CNN (Abs:2X)) severely degrades the performance.

### 3 EXPERIMENTS

In this Section, we empirically study the performance difference of CV-CNN based on different performance measures (network structure, number of parameters, number of MAC operations). We consider diverse datasets, viz., inherent real-valued datasets, complex-valued dataset, and pseudo complex-valued datasets in which magnitude information is more, less or as important as phase information. For each measure and dataset, we consider networks of different depth to study the effect of depth. We limit the randomness in training between RV-CNN and CV-CNN in the following ways: (a) Data augmentation and shuffling are made consistent across RV-CNN and CV-CNN (via fixed algorithmic seed) (b) The effect of random initialization and stochastic layers are reduced by considering statistical measure of multiple independent runs (via multiple seeds). We consider three different seeds. The above scheme enables reproducibility of our results.

**Baselines:** For a given CV-CNN (denoted as CV-CNN(2X)), we consider four different RV-CNN baselines.

1. **RV-CNN (X):** RV-CNN based on similar network structure as that of CV-CNN. It is derived from replacing each layer of CV-CNN with respective RV-CNN layer. The network input is the concatenated real and imaginary parts of the dataset considered (Sec. 2.1).
2. **RV-CNN (C:2X):** RV-CNN based on similar number of parameters. It is derived by multiplying the number of output maps of each CV-CNN convolution and fully-connected layer by  $\sqrt{2}$ . The network input is same as that of RV-CNN (X) (Sec. 2.2).
3. **RV-CNN (Abs:2X):** This is same as that of RV-CNN (C:2X), but with input as only the magnitude part of dataset (Chen et al. (2016); Wilmanski et al. (2016); Morgan (2015)).
4. **RV-CNN (4X):** RV-CNN based on similar number of MAC operations. This is similar to RV-CNN (C:2X) with same input, but multiplication factor is 2 instead of  $\sqrt{2}$  (Sec. 2.3).

Figures 3(a-b) compare these baselines based on the number of parameters and MAC operations. Note that a similarity based on one performance measure creates a dissimilarity based on the others.

**Datasets:** We evaluate CV-CNN against the RV-CNN baselines on 11 different datasets.

- (i) **Inherent real-valued data:** We consider CIFAR-10 and CIFAR-100 datasets. The CIFAR10 contains 60000  $32 \times 32$  colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The CIFAR100 dataset is similar to CIFAR-10, except it has 100 classes containing 600 images each. There are 500 training images and 100 testing images per class.
- (ii) **Inherent complex-valued data:** Following (Singhal et al., 2022), we consider the publicly available MSTAR dataset (Keydel et al., 1996) that has 10K complex-valued SAR images of size  $128 \times 128$  distributed across 10 classes. We used 80% data for training.
- (iii) **Pseudo complex-valued data:** We transform CIFAR-10 and CIFAR-100 datasets such that the resultant data characterize diverse magnitude-phase content. Figure 4 first row illustrates our transformation of real-valued grayscale image-intensities in the range  $[0, 1]$  (i.e.,

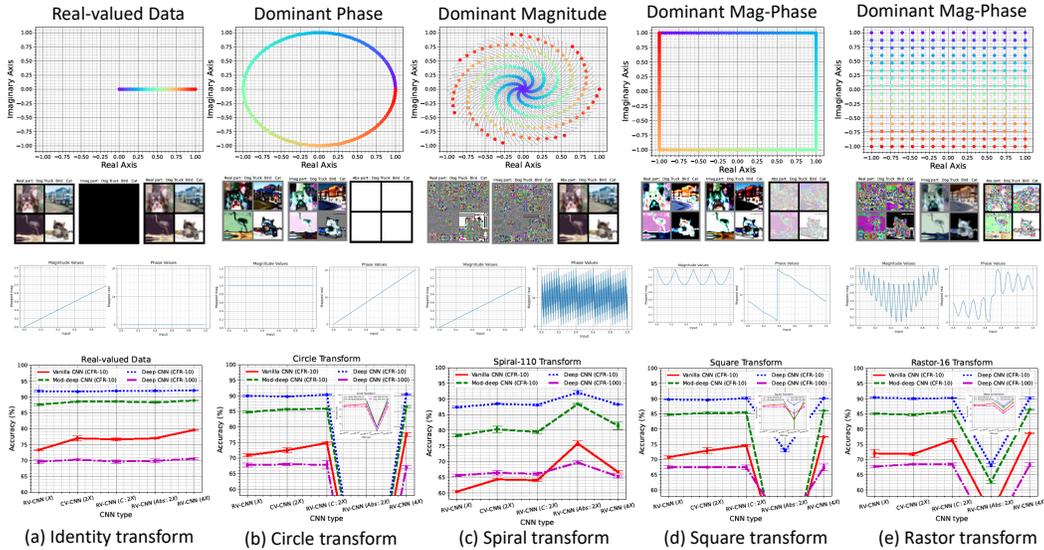


Figure 4: Pseudo Complex-valued Data: Figs. (b-e) represent different complex-valued transforms. The first row depicts element-wise transformation of real-valued data (left-most figure) using VIBGYOR colors. The second row illustrates how the real, imaginary, and magnitude parts of RGB images are modified due to a transform. The third row depicts the magnitude and phase for each image-intensity  $[0, 1]$ . The fourth row compares the performance of different methods. Note that the performance difference of CV-CNN with RV-CNN depends on the performance measures.

left most figure) to complex-valued data in the range  $[-1, 1] + j[-1, 1]$ . Element-wise transformation is depicted via VIBGYOR color. Figures 4(b-d) present four transformations: circle, spiral, square and rastor, which are one-to-one mappings from  $\mathbb{R} \rightarrow \mathbb{C}$ . The second row of Fig. 4 provides real, imaginary, and magnitude parts of four transformed images. The third row side-by-side indicates how the magnitude and phase is transformed. The higher the cardinality of many-to-one mapping(s) in magnitude/phase (i.e., graphically, number of intersections made by a horizontal line), higher the information loss in magnitude/phase. E.g., circle transform in Fig. 4(b) third row exhibits the highest intersections in the magnitude plot and hence contains the least information in the magnitude part (e.g., images’ magnitude part in the second row is null). Hence, circle transform has dominant phase and likewise, spiral transform has dominant magnitude. Note that the square and rastor transform have information loss in both magnitude-and-phase, and because those mappings are one-to-one, both magnitude and phase are required to represent the data (i.e., dominant magnitude-and-phase). We also employ FFT transform that has dominant phase (whether a complex-valued dataset has dominant magnitude/phase can be found similarly by the cardinality of many-to-one mapping of its elements from *complex* plane to magnitude/phase). Our supplementary material contains three more transformations.

**Networks:** For classifying CIFAR10, MSTAR and eight transformed CIFAR10 datasets, we consider three CNN architectures with increasing depth and complexity. Figures 3(a-b) compare these networks based on the number of parameters and MAC operations. The different networks are

- (A) Vanilla CNN: A basic feedforward CNN with five-layers (two convolutional and three fully-connected layers).
- (B) Moderately Deep CNN: A residual CNN with five layers (four convolutional layer with one residual block and one fully-connected layer) and dropout (Resnet5).
- (C) Deep CNN: A residual CNN with eight layers (seven convolutional layers with two residual blocks and one fully-connected layer). Here, batch normalization is also added (Resnet8).

For CIFAR100 and eight transformed CIFAR100 datasets, we consider Resnet18 (seventeen convolutional layers with eight residual blocks, and one fully-connected layer with batch normalization).

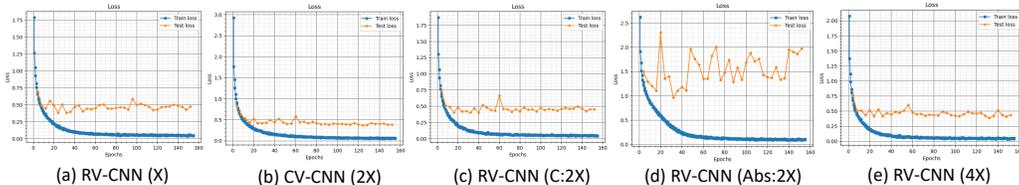


Figure 5: Convergence during learning based on different performance measures (for the case of Rastor transform in Fig. 4e). Note that CV-CNN converges well as compared to different RV-CNNs.

Table 1: Result Summary (C and R denote performances of CV-CNN and RV-CNN, respectively).

Network Depth	Similarity Measures (RV-CNNs with Real-Imaginary input)			Data Diversity (RV-CNNs with Magnitude input)		
	Network Structure	No: of Parameters	No: of MAC Ops	Dominant Magnitude	Dominant Phase	Similar Mag-Phase
↓ (decr)	C > R	C ≈ R	C < R	C < R	C >> R	C >> R
↑ (incr)	C ≈ R	C ≈ R	C ≈ R	C < R	C >> R	C >> R

### 3.1 RESULTS AND DISCUSSIONS

We provide different results in Figs. 3(c-d) and Fig. 4 last row. For brevity, we divide them into two cases: (A) RV-CNN input consist of the real and imaginary parts (RV-CNN (X), RV-CNN (C:2X), RV-CNN (4X)), (B) RV-CNN input consist of only the magnitude part (RV-CNN (Abs:2X)).

**Case A:** Note that as the network depth decreases, the overall performances of: CV-CNN (2X) is better than RV-CNN (X); CV-CNN (2X) and RV-CNN (C:2X) are comparable; and RV-CNN (4X) is better than CV-CNN (2X). This means that answer to the question whether CV-CNN is better/worse than RV-CNN depends on the performance measures (as we inferred in Secs. 2.1-2.3). If this is accounted for, it eliminates the contradictions in prior works as follows: CV-CNNs perform better based on similar network structure (Gu & Ding, 2018; Popa & Cernăzanu-Glăvan, 2018; Yao et al., 2020), performances of CV-CNNs and RV-CNNs are comparable based on similar number of parameters (Guberman, 2016; Trabelsi et al., 2018), and RV-CNNs perform better based on similar number of MACs (Barrachina et al., 2021). Note that these performance differences diminish as the network depth increases. This can be possibly attributed to the over-parameterization of all networks (or representation capabilities in Secs. 2.1-2.3 saturate). Fig. 5 shows that CV-CNN converges well. **Case B** Note in Fig. 4 that magnitude input in an RV-CNN (RV-CNN (Abs:2X)) performs better for spiral transform, whereas magnitude input significantly degrades the performances of FFT, circle, rastor, and square transforms that has phase or magnitude-and-phase as the dominant part. Thus we conclude that magnitude input to a RV-CNN is beneficial for datasets with dominant magnitude (as in Huang et al. (2020)), whereas it degrades the performance when data has dominant phase or both magnitude-and-phase. This holds true for different network depths. But this has no effect for classifying real-valued data (Fig. 4(a)). Our main experimental findings are summarized in Table 3.1.

(Please refer to our supplementary material for additional results that reinforce our findings. There, we further relate CV-CNN layers with RV-CNN layers, and provide informative tables.)

## 4 CONCLUSION

In this paper, we addressed the contradictory answers present in the literature for the following question: CV-CNN performs better or worse than RV-CNN for classification task? To this end, we analysed the relation between different layers of CV-CNN and RV-CNN based on different performance measures. We showed that CV-CNN, unlike RV-CNN, can preserve magnitude-and-phase in ReLU and maxpooling layers. Our analysis and comprehensive empirical study led to the conclusion that an answer to the above question cannot be generalized, but depends on the performance measure under consideration. Our findings corroborated several prior works that had contradictory answers and placed them in a strong footing. We believe that this encourages more research in CV-CNNs.

## REFERENCES

- J. A. Barrachina, C. Ren, C. Morisseau, G. Vieillard, and J.-P. Ovarlez. Complex-valued vs. real-valued neural networks for classification perspectives: An example on non-circular data. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2990–2994, 2021. doi: 10.1109/ICASSP39728.2021.9413814.
- Joshua Basse, Lijun Qian, and Xianfang Li. A survey of complex-valued neural networks. *arXiv preprint arXiv:2101.12249*, 2021.
- Rudrasis Chakraborty, Jiayun Wang, and Stella X Yu. Sur-real: Frechet mean and distance transform for complex-valued deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.
- Sizhe Chen, Haipeng Wang, Feng Xu, and Ya-Qiu Jin. Target classification using the deep convolutional networks for sar images. *IEEE transactions on geoscience and remote sensing*, 54(8): 4806–4817, 2016.
- Lukas Drude, Bhiksha Raj, and Reinhold Haeb-Umbach. On the appropriateness of complex-valued neural networks for speech enhancement. In *Interspeech*, pp. 1745–1749, 2016.
- William T Freeman, Edward H Adelson, et al. The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence*, 13(9):891–906, 1991.
- Bernard Ghanem and Narendra Ahuja. Phase based modelling of dynamic textures. In *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8, 2007. doi: 10.1109/ICCV.2007.4409094.
- Shenshen Gu and Lu Ding. A complex-valued vgg network based deep learning algorithm for image recognition. In *2018 Ninth International Conference on Intelligent Control and Information Processing (ICICIP)*, pp. 340–343. IEEE, 2018.
- Nitzan Guberman. On complex valued convolutional neural networks. *arXiv preprint arXiv:1602.09046*, 2016.
- Akira Hirose. Complex-valued neural networks: The merits and their origins. In *2009 International joint conference on neural networks*, pp. 1237–1244. IEEE, 2009.
- Akira Hirose. *Complex-valued neural networks*, volume 400. Springer Science & Business Media, 2012.
- Zhongling Huang, Mihai Datcu, Zongxu Pan, and Bin Lei. Deep sar-net: Learning objects from signals. *ISPRS Journal of Photogrammetry and Remote Sensing*, 161:179–193, 2020.
- Eric R Keydel, Shung Wu Lee, and John T Moore. Mstar extended operating conditions: A tutorial. *Algorithms for Synthetic Aperture Radar Imagery III*, 2757:228–242, 1996.
- ChiYan Lee, Hideyuki Hasegawa, and Shangce Gao. Complex-valued neural networks: A comprehensive survey. *IEEE/CAA Journal of Automatica Sinica*, 9(8):1406–1426, 2022. doi: 10.1109/JAS.2022.105743.
- Jinyu Li et al. Recent advances in end-to-end automatic speech recognition. *APSIPA Transactions on Signal and Information Processing*, 11(1), 2022.
- Nils Mönning and Suresh Manandhar. Evaluation of complex-valued neural networks on real-valued classification tasks. *arXiv preprint arXiv:1811.12351*, 2018.
- David AE Morgan. Deep convolutional neural networks for atr from sar imagery. In *Algorithms for Synthetic Aperture Radar Imagery XXII*, volume 9475, pp. 116–128. SPIE, 2015.
- W. Pearlman and R. Gray. Source coding of the discrete fourier transform. *IEEE Transactions on Information Theory*, 24(6):683–692, 1978. doi: 10.1109/TIT.1978.1055950.

- Călin-Adrian Popa and Cosmin Cernăzanu-Glăvan. Fourier transform-based image classification using complex-valued convolutional neural networks. In *International Symposium on Neural Networks*, pp. 300–309. Springer, 2018.
- Călin-Adrian Popa. Complex-valued convolutional neural networks for real-valued image classification. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 816–822, 2017. doi: 10.1109/IJCNN.2017.7965936.
- Saeed Sepasi, Ehsan Reihani, Abdul M Howlader, Leon R Roose, and Marc M Matsuura. Very short term load forecasting of a distribution system with high pv penetration. *Renewable energy*, 106: 142–148, 2017.
- Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- Utkarsh Singhal, Yifei Xing, and Stella X Yu. Co-domain symmetry for complex-valued deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 681–690, 2022.
- Tomasz Szandała. Review and comparison of commonly used activation functions for deep neural networks. In *Bio-inspired Neurocomputing*, pp. 203–224. Springer, 2021.
- Chiheb Trabelsi, Olexa Bilaniuk, Ying Zhang, Dmitriy Serdyuk, Sandeep Subramanian, Joao Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J Pal. Deep complex networks. In *Proceedings of the International Conference on Learning Representations*, 2018.
- Hirofumi Tsuzuki, Mauricio Kugler, Susumu Kuroyanagi, and Akira Iwata. An approach for sound source localization by complex-valued neural network. *IEICE TRANSACTIONS on Information and Systems*, 96(10):2257–2265, 2013.
- Mark Tygert, Joan Bruna, Soumith Chintala, Yann LeCun, Serkan Piantino, and Arthur Szlam. A mathematical motivation for complex-valued convolutional networks. *Neural computation*, 28(5):815–825, 2016.
- Patrick Virtue, X Yu Stella, and Michael Lustig. Better than real: Complex-valued neural nets for mri fingerprinting. In *2017 IEEE international conference on image processing (ICIP)*, pp. 3953–3957. IEEE, 2017.
- Michael Wilmanski, Chris Kreucher, and Jim Lauer. Modern approaches in deep learning for sar atr. In *Algorithms for synthetic aperture radar imagery XXIII*, volume 9843, pp. 195–204. SPIE, 2016.
- Xin Yao, Xiaoran Shi, and Feng Zhou. Human activities classification based on complex-value convolutional neural network. *IEEE Sensors Journal*, 20(13):7169–7180, 2020.
- Donglin Zhuang, Xingyao Zhang, Shuaiwen Song, and Sara Hooker. Randomness in neural network training: Characterizing the impact of tooling. *Proceedings of Machine Learning and Systems*, 4: 316–336, 2022.

## A SUPPLEMENTARY MATERIAL

1. Relating convolutional layer in CV-CNN to that of RV-CNN (Sec. A.1).
2. Additional Experiments (Fig. 7)
  - (a) Semicircle transform (dominant phase)
  - (b) Spiral transform with 44 rotations (dominant magnitude)
  - (c) Rastor transform with four rows (dominant magnitude-and-phase)
3. Different Tables
  - (a) Standard building blocks of CV-CNN and their relation to RV-CNN. (Table 2)
  - (b) Contradicting answers in the literature regarding the merits of CV-CNN (Table 3)
  - (c) A sample of all data used for comparison for a real-valued classification task (Table 4)

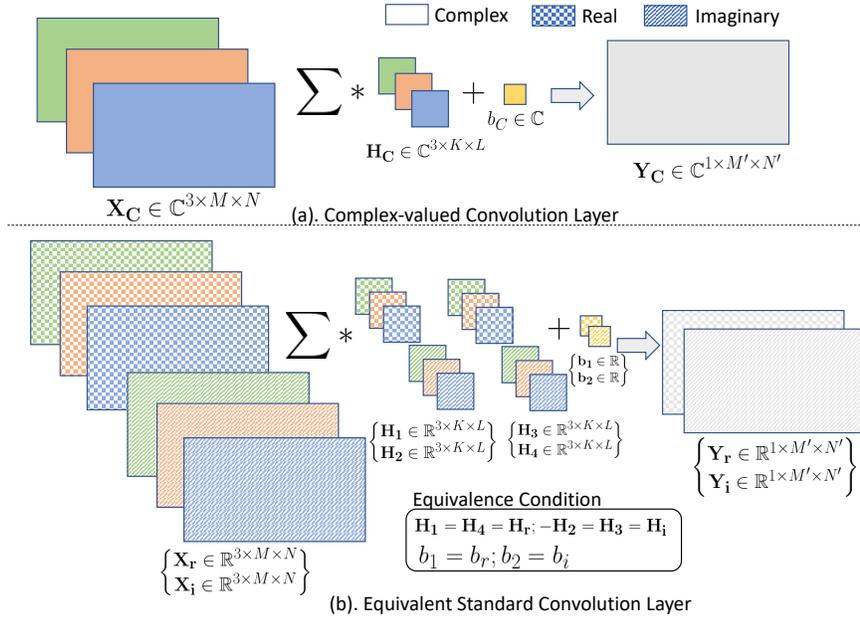


Figure 6: Equivalence of Convolutional Layer.

#### A.1 CONVOLUTIONAL LAYER IN CV-CNN AND RV-CNN

Convolutional layer is a main component of CNN, that maps an input tensor,  $\mathbf{X} \in \mathbb{R}^{B \times M \times N}$ , by an affine transformation to an output tensor  $\mathbf{Z} \in \mathbb{R}^{B' \times M' \times N'}$ . A feature-map of output  $Z$  (say,  $\mathbf{Y} \in \mathbb{R}^{1 \times M' \times N'}$ ) is obtained as

$$\mathbf{Y} = \left( \sum_{m=1}^B \mathbf{X}(m) * \mathbf{H}(m) \right) + b, \quad (12)$$

where  $*$  denotes the convolution operation and  $\mathbf{X}(m)$  is the  $i$ th layer of  $\mathbf{X}$  ( $1 \leq m \leq B$ ). The learnable parameters of convolutional layer is the kernel  $\mathbf{H} \in \mathbb{R}^{B \times K \times L}$  and the bias  $b \in \mathbb{R}$ . Note that each layer of the output tensor  $\mathbf{Z}_C$  has an individual kernel and bias as in equation 12.

In contrast, convolutional layer in CV-CNN has complex inputs and weights which is given by Trabelsi et al. (2018); Gu & Ding (2018)

$$\mathbf{Y}_C = \left( \sum_{m=1}^B \mathbf{X}_C(m) * \mathbf{H}_C(m) \right) + b_{C.}, \quad (13)$$

which can be equivalently written as

$$\begin{aligned} \mathbf{Y}_r &= \left( \sum_{m=1}^B \mathbf{X}_r(m) * \mathbf{H}_r(m) - \mathbf{X}_i(m) * \mathbf{H}_i(m) \right) + b_r, \\ \mathbf{Y}_i &= \left( \sum_{m=1}^B \mathbf{X}_r(m) * \mathbf{H}_i(m) + \mathbf{X}_i(m) * \mathbf{H}_r(m) \right) + b_i. \end{aligned} \quad (14)$$

Using equation 14, an equivalent form using normal CNN can be designed as shown in Fig. 6A(b), i.e.,

$$\begin{Bmatrix} \text{out}_r \\ \text{out}_i \end{Bmatrix} = \begin{Bmatrix} \mathbf{H1} & \mathbf{H2} \\ \mathbf{H3} & \mathbf{H4} \end{Bmatrix} * \begin{Bmatrix} \mathbf{X}_r \\ \mathbf{X}_i \end{Bmatrix} + \begin{Bmatrix} b_1 \\ b_2 \end{Bmatrix}, \quad (15)$$

where we consider  $B = 1$  (wlog) for brevity, and  $*$  distributes as per matrix multiplication rule and all parameters are real. Comparing equation 15 and equation 14, as in linear complex layer,

Table 2: Standard Building Blocks of Complex-valued CNN. (Note: Subscript  $C$ ,  $r$  and  $i$  indicate complex-valued, real and imaginary parts, respectively.)

Modules	RV-CNN with Similar Network Structure	Complex-valued CNN	RV-CNN with Similar Number of MAC operations
<b>Affine transformations</b>			
Scalar Multiplication	$k \cdot \mathbf{X}$ for $k \in \mathbb{R}$	$k_C \cdot \mathbf{X}_C$ for $k_C \in \mathbb{C}$	$\begin{Bmatrix} k_1 & k_2 \\ k_3 & k_4 \end{Bmatrix} \cdot \begin{Bmatrix} \mathbf{X}_r \\ \mathbf{X}_i \end{Bmatrix}$ , where $k_1 = k_4 = k_r$ , $-k_2 = k_3 = k_i$ , and $k_1, k_2, k_3, k_4 \in \mathbb{R}$
Fully-connected	$\mathbf{A} \cdot \mathbf{x}$ , where $\mathbf{A}$ is a real weight matrix	$\mathbf{A}_C \cdot \mathbf{x}_C = (\mathbf{A}_r \cdot \mathbf{x}_r - \mathbf{A}_i \cdot \mathbf{x}_i) + j(\mathbf{A}_i \cdot \mathbf{x}_r - \mathbf{A}_r \cdot \mathbf{x}_i)$	$\begin{Bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{Bmatrix} \cdot \begin{Bmatrix} \mathbf{X}_r \\ \mathbf{X}_i \end{Bmatrix}$ , where $\mathbf{A}_1 = \mathbf{A}_4 = \mathbf{A}_r$ , $-\mathbf{A}_2 = \mathbf{A}_3 = \mathbf{A}_i$ .
Convolution	$\mathbf{X} * \mathbf{H}$ , where $\mathbf{H}$ is a real kernel	$\mathbf{X}_C * \mathbf{H}_C = (\mathbf{X}_r * \mathbf{H}_r - \mathbf{X}_i * \mathbf{H}_i) + j(\mathbf{X}_r * \mathbf{H}_i + \mathbf{X}_i * \mathbf{H}_r)$	$\begin{Bmatrix} \mathbf{H}_1 & \mathbf{H}_2 \\ \mathbf{H}_3 & \mathbf{H}_4 \end{Bmatrix} * \begin{Bmatrix} \mathbf{X}_r \\ \mathbf{X}_i \end{Bmatrix}$ , where $\mathbf{H}_1 = \mathbf{H}_4 = \mathbf{H}_r$ , $-\mathbf{H}_2 = \mathbf{H}_3 = \mathbf{H}_i$ .
Residual block	$F(\mathbf{X}, \{\mathbf{W}\}) + \mathbf{X}$ , for real parameters $\{\mathbf{W}\}$ .	$F(\mathbf{X}_C, \{\mathbf{W}_C\}) + \mathbf{X}_C$	$\begin{Bmatrix} F(\mathbf{X}_C, \{\mathbf{W}_C\})_r \\ F(\mathbf{X}_C, \{\mathbf{W}_C\})_i \end{Bmatrix} + \begin{Bmatrix} \mathbf{X}_r \\ \mathbf{X}_i \end{Bmatrix}$
Average Pooling	$(\text{mean}(\mathbf{X}) _W) \downarrow_M$ , where $W$ is a window of size $M \times M$ .	$(\text{mean}(\mathbf{X}_C) _W) \downarrow_M$	$\begin{Bmatrix} (\text{mean}(\mathbf{X}_r) _W) \downarrow_M \\ (\text{mean}(\mathbf{X}_i) _W) \downarrow_M \end{Bmatrix}$
<b>Non-linear Transformations</b>			
ReLU	$\text{ReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$	$\text{zReLU}(x) = \begin{cases} x & \text{if } \theta_x \in [0, \frac{\pi}{2}] \\ 0 & \text{otherwise} \end{cases}$	$\mathbf{x}$
		$\text{CReLU}(x_c) = \text{ReLU}(x_r) + j\text{ReLU}(x_i)$	$\begin{Bmatrix} \text{ReLU}(x_r) \\ \text{ReLU}(x_i) \end{Bmatrix}$
		$\text{MODReLU}(x_C) = \text{ReLU}( x_C  - b)e^{j\theta_x}$	$\mathbf{x}$
		$\text{EqReLU}(x) = \begin{cases} x & \text{if } \theta_x \in [-\frac{\pi}{2}, \frac{\pi}{2}] \\ 0 & \text{otherwise} \end{cases}$	$\mathbf{x}$
Max Pooling	$(\max(\mathbf{X}) _W) \downarrow_M$ , where $W$ is a window of size $M \times M$ .	$(\max(\ \mathbf{X}\ ) _W) \downarrow_M e^{j\theta_x}$	$\mathbf{x}$

for equivalence the convolution filters in CNN require twice the number of real parameters as that of CV-CNN, and hence there exist many degenerate mappings other than the desired mapping (i.e.,  $\mathbf{H}_1 = \mathbf{H}_4 = \mathbf{H}_r$ ,  $-\mathbf{H}_2 = \mathbf{H}_3 = \mathbf{H}_i$ ). But there is no increase in the real parameters for bias  $b_C^2$ .

<sup>2</sup>The inferences derived here for Convolutional layer hold good for fully-connected layer with bias as well, the main difference being that the convolution in equation 15 has to be replaced by multiplication

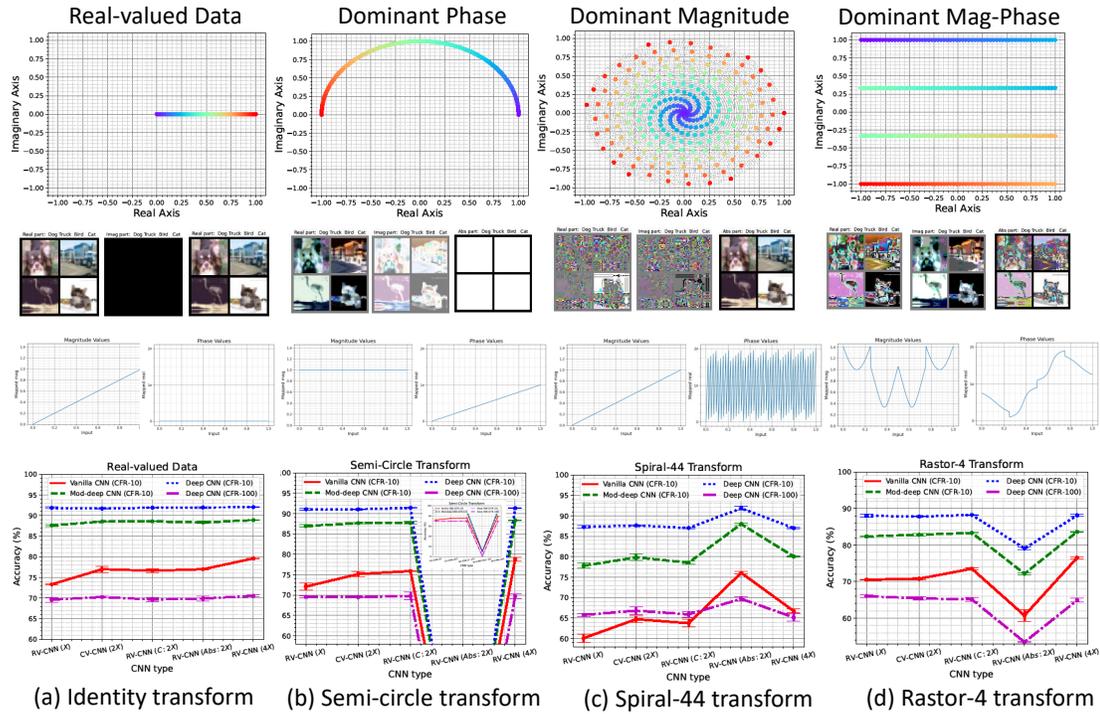


Figure 7: Pseudo Complex-valued Data: Figs. (b-d) represent different complex-valued transforms. The first row depicts element-wise transformation of real-valued data (left-most figure) using VIB-GYOR colors. The second row illustrates how the real, imaginary, and magnitude parts of RGB images are modified due to a transform. The third row depicts the magnitude and phase for each image-intensity ( $[0, 1]$ ). The fourth row compares the performance of different methods. Note that the performance difference of CV-CNN with RV-CNN depends on the performance measures.

Table 3: Contradictions on the performances between CV-CNN and RV-CNN. Note that different works highlight *only* the inferences (last column), while seldom relating them with other details (columns 2-4). Our results (see Table 3.1) resolve these contradictions to a large extent).

Works	Model	Performance Measures	Datasets Employed	Inferences for CV-CNN (against RV-CNN)
<b>Not in favor of complex-domain representations in CV-CNNs</b>				
Guberman (2016)	Vanilla CNN ( <i>without</i> Resblocks, dropout and batch norm)	Similar No: of parameters	Cell Classification (binary class)	No improvement
Trabelsi et al. (2018)	CNN (with Resblocks, dropout and batch norm)	Similar No: of parameters	CIFAR10, CIFAR100, Truncated SVHN	Slightly inferior in two out of three cases
Mönning & Manandhar (2018)	Multilayer Perceptron	Similar No: of parameters	MNIST, CIFAR10, CIFAR100, Reuters	No Improvement
Tygart et al. (2016)	Blocks of: Conv + absolute + pooling	(Not clear from the paper)	CIFAR10	No Improvement
Huang et al. (2020)	Parallel CNNs with spatial and spectral data, with fusion at end.	Similar No: of parameters	SAR datasets	Inferior w.r.t magnitude input
<b>In favor of complex-domain representations in CV-CNNs</b>				
Gu & Ding (2018)	Complex-valued VGG Network Simonyan & Zisserman (2014)	Similar Network Structure	CIFAR10	Superior
Popa (2017)	Vanilla CNN ( <i>without</i> Resblocks, dropout and batch norm)	Similar No: of parameters	MNIST, CIFAR10 (Employ custom gradient for training)	Superior in both cases
Barrachina et al. (2021)	Multilayer Perceptron	Similar No: of MAC operations	Custom data (non-circular 2D random numbers)	Superior
Popa & Cernăzanu-Glăvan (2018)	Vanilla CNN ( <i>without</i> Resblocks, dropout and batch norm)	Similar Network Structure	FFT of MNIST, CIFAR10, SVHN	Superior in all cases
Yao et al. (2020)	Vanilla CNN ( <i>without</i> Resblocks, dropout and batch norm)	Similar Network structure	Simulated Human Radar Echo dataset	Superior

Table 4: Sample Results employed to create the plots for real-valued data.

	RV-CNN (X) (Real Input)	CV-CNN (2X) (Complex Input)	RV-CNN (Abs:2X) (Real Input)	RV-CNN (C:2X) (Complex Input)	RV-CNN (4X) (Complex Input)
<b>Vanilla CNN (<i>without</i> Resblocks, Dropout, and Batch norm) on CIFAR-10</b>					
No: of RV Params	62,006	1,24,012	1,24,700	1,24,605	2,45,862
Accuracy (3 seeds)	73.23/73.51/73.23	76.91/77.79/76.29	77.21/76.97/76.77	76.48/77.08/76.37	79.72/79.61/79.51
Mean & Stdev Acc.	73.32 (0.16)	77.00 (0.75)	76.98 (0.22)	76.64 (0.38)	79.61 (0.10)
<b>Moderately Deep Network (Resnet5 <i>with</i> Resblocks and Dropout) on CIFAR-10</b>					
No: of RV Params	3,75,946	7,51,892	7,55,674	7,58,131	14,92,618
Accuracy (3 seeds)	87.73/87.35/87.73	88.66/88.59/88.39	88.66/88.00/88.33	88.61/88.70/88.47	89.04/88.81/88.81
Mean & Stdev Acc.	87.60 (0.22)	88.55 (0.14)	88.33 (0.33)	88.59 (0.12)	88.89 (0.13)
<b>Deep Network (Resnet8 <i>with</i> Resblocks, Dropout, and Batch norm) on CIFAR-10</b>					
No: of RV Params	42,14,538	84,30,804	84,21,050	84,23,507	1,68,37,514
Accuracy (3 seeds)	91.62/91.60/91.82	92.09/91.73/91.68	91.80/91.86/92.20	91.91/91.87/91.89	91.98/92.13/91.95
Mean & Stdev Acc.	91.68 (0.12)	91.83 (0.23)	91.95 (0.22)	91.89 (0.02)	92.02 (0.10)
<b>Deep Network (Resnet18 <i>with</i> Resblocks and Batch norm) on CIFAR-100</b>					
No: of RV Params	1,12,20,132	2,24,45,064	2,26,46,451	2,26,48,908	4,47,58,628
Accuracy (3 seeds)	70.2/69.07/69.60	70.24/70.04/70.48	70.47/69.4/69.81	70.27/69.2/69.53	70.72/70.61/70.32
Mean & Stdev Acc.	69.62 (0.57)	70.25 (0.22)	69.89 (0.54)	69.67 (0.55)	70.55 (0.21)