

FCCNs: Fully Complex-valued Convolutional Networks using Complex-valued Color Model and Loss Function

Saurabh Yadav, Koteswar Rao Jerripothula
Indraprastha Institute of Information Technology Delhi (IIIT-Delhi)
New Delhi, India
{saurabhy,koteswar}@iiitd.ac.in

Abstract

Although complex-valued convolutional neural networks (iCNNs) have existed for a while, they lack proper complex-valued image inputs and loss functions. In addition, all their operations are not complex-valued as they have both complex-valued convolutional layers and real-valued fully-connected layers. As a result, they lack an end-to-end flow of complex-valued information, making them inconsistent w.r.t. the claimed operating domain, i.e., complex numbers. Considering these inconsistencies, we propose a complex-valued color model and loss function and turn fully-connected layers into convolutional layers. All these contributions culminate in what we call FCCNs (Fully Complex-valued Convolutional Networks), which take complex-valued images as inputs, perform only complex-valued operations, and have a complex-valued loss function. Thus, our proposed FCCNs have an end-to-end flow of complex-valued information, which lacks in existing iCNNs. Our extensive experiments on five image classification benchmark datasets show that FCCNs consistently perform better than existing iCNNs. Code is available at <https://github.com/saurabhy/FCCNs>.

1. Introduction

In the ever-evolving deep learning era, complex-valued neural networks (iCNNs) show great promise, given their ability to learn complex-valued representations. Nevertheless, iCNNs have some inconsistencies which should be addressed: (i) Input images are real-valued. (ii) They are not fully complex-valued. (iii) They have real-valued losses. Because of these inconsistencies, we don't witness an end-to-end flow of complex-valued information in these networks. As a result, it's difficult to declare them as truly complex-valued. This paper aims to weed out these inconsistencies and uncover their true potential in computer vision research.

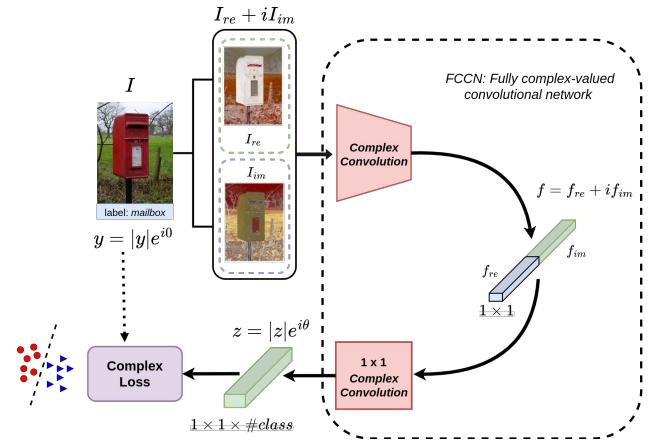


Figure 1. We introduce Fully Complex-valued Convolutional Networks (FCCNs), which leverage (i) complex-valued color inputs using our novel iHSV color model, (ii) complex-valued 1x1 convolutions to turn even fully-connected layers into convolutional layers, and (iii) our novel complex-valued loss function to tackle the deficit between $|y|e^{j\theta}$ (complex-valued labels) and $|z|e^{j\theta}$ (complex-valued outputs). Hence, FCCNs operate entirely in complex-domain.

However, several challenges are involved: (i) No complex-valued color model is readily available to allow us to have real and imaginary components of an image, as shown in Fig. 1. (ii) No prior study exists on how fully-connected layers can be made complex-valued to have an end-to-end flow of complex-valued information. (iii) There are no well-tested complex-valued loss functions for deep learning approaches.

Despite these challenges, what motivates us to explore complex-valued representation for color images is the sheer potential complex-valued networks have shown in other domains such as MRI [7, 40], radar signals [10, 11] and audio signals [12, 16], where complex-valued inputs are readily available. As per the recent studies conducted in [1, 8, 19, 27], iCNNs have performed better than their real-valued counterparts. Moreover, they are biologically moti-

vated [31] and possess greater generalization capacity [15], which inspires us to explore HSV color space for such inputs since it aligns with how human vision perceives color-making attributes.

In the past, in [37], authors have tried image classification using iCNNs inputting RGB as the real part and learned features from RGB as the imaginary part of an image. However, we argue that components of complex numbers are supposed to be orthogonal, which means they are mutually independent. So, it should not be possible to derive one part from another. In [34], authors look for orthogonality and propose to use $\{L^* + i0, a + ib\}$ and $\{R + iG, G + iB\}$ as possible complex-valued inputs. Note that these are more like complex-valued encodings than a complex-valued representation of an image. That is because we cannot decompose an image based on them into real and imaginary parts (as shown in Fig. 1) that look like images by themselves. Moreover, HSV color space is not explored yet for orthogonality and complex-valued representation.

There have also been attempts to preserve complex-valued information in fully-connected layers through manifold learning [2]. Nevertheless, the complex-valued representation is sacrificed, and that was done due to the lack of complex-valued losses; otherwise, imaginary or phase information would be required to be left out at the end. Thus, there is a need for complex-valued loss functions to preserve complex-valued representations throughout, and we can derive complex-valued loss function from prevalent cross-entropy loss itself. Moreover, since fully-connected layers can be implemented through convolution operations and complex-valued convolutions already exist, it must be possible to have fully complex-valued networks.

More specifically, we develop FCCNs (Fully Complex-valued Convolutional Networks), which take complex-valued images as inputs, perform only complex-valued operations, and have a complex-valued loss function. For inputs, we observe that any color in the cylindrical HSV color model has intersecting color planes. For e.g. (H_0, S_0, V_0) color will have $H = H_0$, $S = S_0$ and $V = V_0$ planes. If we consider these planes as argand planes, we can obtain complex numbers using the locations of that color on them. We use these complex numbers to represent any color. We call this color model iHSV. Following this, we can generate a complex-valued representation of any image $I \rightarrow I_{re} + iI_{im}$, as shown in Fig. 1, where we can separate an image into real and imaginary parts (images). When images are supplied in this form to our FCCNs, obtained by implementing fully-connected layers as convolutional layers, we get a complex-valued output, which is then matched against the label (in complex form) by our complex-valued loss function to help train the network, as shown in Fig. 1.

Our contributions in this paper are three-fold: (i) We propose iHSV, a novel complex-valued color model. (ii) We de-

rive a complex-valued loss function from the cross-entropy loss. (iii) We turn iCNNs into FCCNs by implementing fully-connected layers as convolutional layers.

2. Related Work

Complex-valued Learning: In the realm of deep learning, the use of complex numbers as weights presents new opportunities for two-dimensional spectra exploration [14, 38]. Furthermore, [31] and [17] provide evidence for the importance of phase information in the firing rate of neurons, highlighting the potential benefits of using complex-valued representations in neural networks. Specifically, synchronized neurons with similar phases fire together, while asynchronous neurons with different phases interfere. This behavior aligns more closely with the functioning of biological neurons, and the passage of synchronized inputs through neurons can be likened to the gating mechanism in both deep feedforward and recurrent neural networks [20, 35, 39].

In recent years, complex-valued networks have been shown to have greater generalization capacity, as demonstrated in previous works [15, 19]. Complex-valued autoencoders have also been shown to outperform slot-attention in object-centric learning [27]. Additionally, complex-valued networks have been utilized to enhance saliency prediction [17, 18] and iris recognition [29]. The work presented in [4] further demonstrated that using a complex-valued vector can be helpful for learning to solve multiple tasks.

Complex-valued Representation: Incorporating the phase component in deep learning models is also motivated from a signal-processing perspective. [30] demonstrate the importance of phase in speech signals and images. Similarly, [5] use complex-valued U-net to improve speech enhancement by incorporating phase information. According to [30], humans rely more on the phase component for perception. Recently, [3] developed a new augmentation technique by recombining phase and magnitude, resulting in increased robustness of CNNs by focusing more on the phase information. Although their method uses Fourier transform to generate phase, it highlights the potential benefits of using complex representations in learning approaches. Complex-valued networks can directly process complex-valued inputs in the spatial domain, but as images are real-valued, a complex-valued color model is required. This research gap is addressed in this study through the proposed iHSV color model.

In the complex-valued convolutional neural network (iCNN) proposed by [37], the complex-valued activations from the convolutional layers are flattened and fed into real-valued fully connected layers by concatenating their real and imaginary parts. As a result, the complex-valued information is lost during propagation through fully connected layers. To address this issue, SurReal [2] used manifold

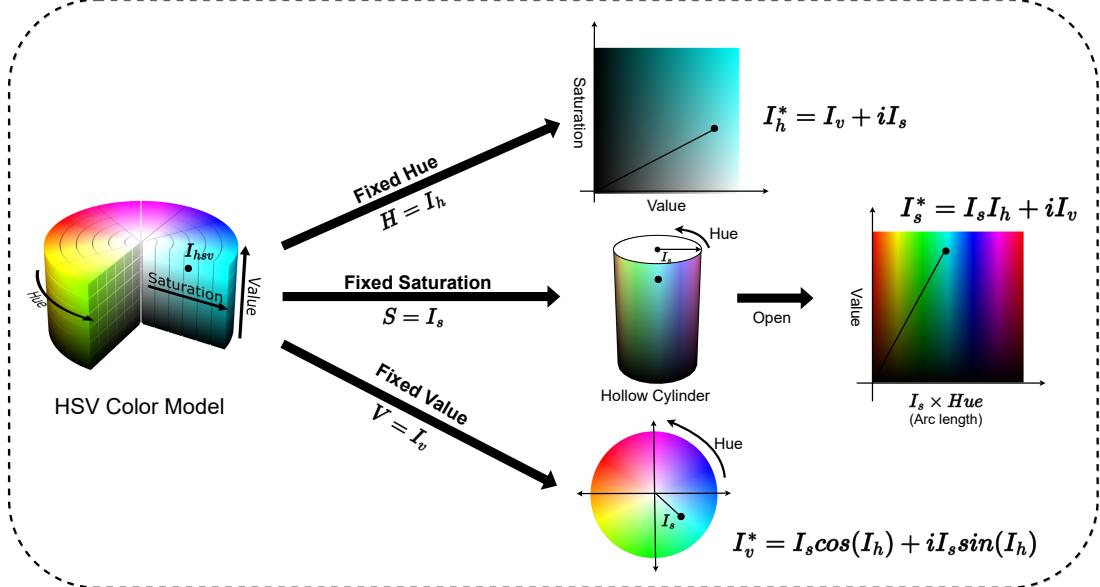


Figure 2. Proposed iHSV color model: Given an HSV color $I_{hsv}(p) = \{I_h(p), I_s(p), I_v(p)\}$ for a pixel p , we keep each of its elements fixed one at a time and derive three argand planes from the HSV color model. The locations of the color on these planes provide us the required complex-valued color representation $\{I_h^*(p), I_s^*(p), I_v^*(p)\}$. Note that (p) has been omitted in the symbols present in the figure for clarity. *Best viewed in color.* HSV Color model figure was taken from [Wikimedia Commons](#) (file: HSV_color_solid_cylinder.png).

learning, but this approach still resulted in real-valued propagation. To preserve complex-valued propagation in fully connected layers, we draw inspiration from [26, 33], which showed that fully connected layers could be implemented as 1x1 convolutions. Therefore, we decided to adopt this approach and continue with complex-valued convolution operations until the end. Since complex-valued convolution operations have already been defined in [37], we simply have to continue with them until the end.

3. Proposed Method

We propose FCCN, a complex-valued fully convolutional network that ensures complex-valued information flow throughout the network. For generating complex-valued inputs, we propose a novel complex color model, iHSV. We also propose a complex-valued loss function to tackle the deficit between complex-valued outputs and labels. In addition, we implement fully connected layers as convolution layers, thus preserving complex-valued information flow.

Recall that hue (H), saturation (S), and value (V) represent the cylindrical HSV color model's angular, radial, and vertical axes. A point in this cylinder represents a color denoted by a set of three elements: H , S , and V values of that point. Let $I_{rgb} = \{I_r, I_g, I_b\}$ denote an RGB image, and $I_{hsv} = \{I_h, I_s, I_v\}$ its HSV equivalent. The location of HSV color of a pixel p in this cylinder can be denoted as $\{I_h(p), I_s(p), I_v(p)\}$. This point in the cylinder is the intersection point of three color planes: $H = I_h(p)$, $S = I_s(p)$,

and $V = I_v(p)$. As shown in Fig. 2, each color plane is obtained by fixing one of the color elements as the corresponding value in color $\{I_h(p), I_s(p), I_v(p)\}$.

Let us understand how the above discussion helps us generate the complex-valued color channels that are fed to FCCNs as inputs, our FCCN architecture, and the complex-valued loss function we develop.

3.1. Fixed Hue Based Channel (I_h^*)

We slice a rectangular plane by fixing H as $I_h(p)$. While we have fixed the value of H , the other two axes S and V still lie on that plane and are orthogonal to each other, as shown in Fig. 2. Let us treat it as an argand plane with V as a real axis and S as an imaginary axis. Upon doing that, we can represent any 2-D location on that plane as a complex number, i.e., (V, S) can be represented as $V + iS$. Therefore, we can derive the following complex number from the location of $I_{hsv}(p)$ on $H = I_h(p)$ color plane:

$$I_h^*(p) = I_v(p) + iI_s(p) \quad (1)$$

where we simply substitute the color values and denote the derived complex number with $I_h^*(p)$, since it's obtained by fixing $H = I_h(p)$. Thus, we now have the first element of the complex representation of color $I_{hsv}(p)$ in the form of $I_h^*(p)$.

Algorithm 1: iHSV Color Model

Input: An image I

```
/* convert RGB image to HSV image */
```

Pre-processing: $I_{rgb} \rightarrow I_{hsv}$

Process: iHSV Generation

```
/* convert each pixel from HSV → iHSV */
```

foreach $p \in I_{hsv}$ **do**

$$\begin{cases} I_h^*(p) = I_v(p) + iI_s(p) \\ I_s^*(p) = I_s(p)I_h(p) + iI_v(p) \\ I_v^*(p) = I_s(p)\cos(I_h(p)) + iI_s(p)\sin(I_h(p)) \end{cases}$$

$$I_{hsv}^* = \{I_h^*, I_s^*, I_v^*\}$$

return I_{hsv}^*

3.2. Fixed Saturation Based Channel (I_s^*)

We can similarly slice out a hollow cylinder when we fix S to $I_s(p)$. While we have fixed the value of S , the other two axes, H and V , still play a role in locating a point on that hollow cylinder. We open up this hollow cylinder to form a rectangular plane, as shown in Fig. 2. On this plane lie two orthogonal axes, $I_s(p)H$ and V . Note here that the axis H is scaled by the factor of $I_s(p)$ (the radius of the cylinder) to represent arc length instead of the angle. We can treat this plane also as an argand plane with $I_s(p)H$ as the real axis and V as the imaginary axis. Upon doing that, again, we can represent any 2-D location on that plane as a complex, i.e., the location $(I_s(p)H, V)$ leads to $I_s(p)H + iV$. Therefore, the complex number derived from the location of $I_{hsv}(p)$ on the color plane $S = I_s(p)$ is as follows:

$$I_s^*(p) = I_s(p)I_h(p) + iI_v(p) \quad (2)$$

where we substitute the color values and denote the derived complex number with $I_s^*(p)$, since it's obtained by fixing $S = I_s(p)$. Thus, $I_s^*(p)$ becomes the second element we require for complex representation of color $I_{hsv}(p)$.

3.3. Fixed Value Based Channel (I_v^*)

As shown in Fig. 2, we can also slice out a circular plane when we fix V as $I_v(p)$. While we have fixed the value of V , the other two axes, H and S , can still be used to find the polar coordinates of a point on that circular plane. Given that, we can have a cartesian coordinate system with $S\cos(H)$ and $S\sin(H)$ as the two orthogonal axes. Let us treat this plane also as an argand plane with $S\cos(H)$ as a real axis and $S\sin(H)$ as an imaginary axis. Upon doing that, $S\cos(H) + iS\sin(H)$ can now represent location $(S\cos(H), S\sin(H))$. Therefore, the complex number derived from the location of $I_{hsv}(p)$ on the color plane $V = I_v(p)$ is as follows:

$$I_v^*(p) = I_s(p)\cos(I_h(p)) + iI_s(p)\sin(I_h(p)) \quad (3)$$

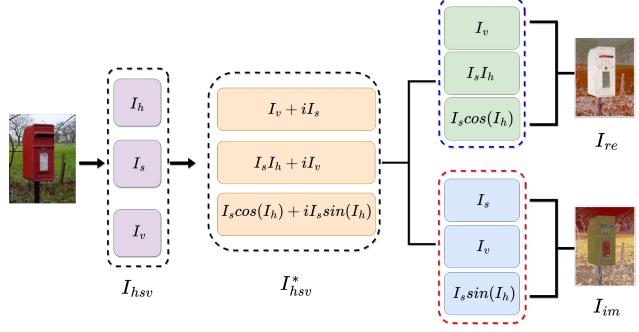


Figure 3. An illustration of how, using iHSV, an RGB image can be converted into complex-valued inputs required by an iCNN.

where we substitute the color values and denote the derived complex number with $I_v^*(p)$, since it's obtained by fixing $V = I_v(p)$. Thus, we now have the third element of our complex representation of color $I_{hsv}(p)$ as $I_v^*(p)$.

3.4. iHSV Color

Now that we have the three complex-valued elements of color, we can combine them to form the set required for the complex representation of color $I_{hsv}^*(p)$ as the following:

$$I_{hsv}^*(p) = \{I_h^*(p), I_s^*(p), I_v^*(p)\} \quad (4)$$

which we represent as $I_{hsv}^*(p)$. This way, any colored pixel can get a complex-valued color representation, as we do in Algorithm 1 to generate complex image I_{hsv}^* from an RGB image I . The constituents of I_{hsv}^* are obtained using Eqns.(1)-(3), for which we need to first convert $I_{rgb} \rightarrow I_{hsv}$ through the pre-processing step.

Table 1 presents the iHSV values of several well-known colors to give readers a sense of the types of complex values we obtain. Notably, when the saturation I_s is zero, all three complex channels depend only on I_v . This behavior is expected since the iHSV color model is based on the HSV model, where, in the case of zero saturation, we are left with only the axis of the cylinder. Therefore, the value remains the only variable in this scenario.

3.5. Complex-valued Inputs

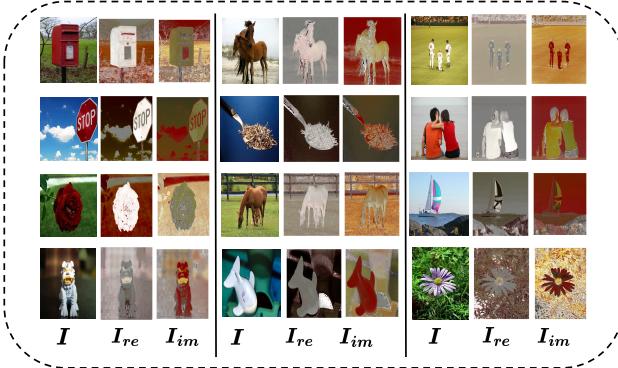
iCNNs require a real image and an imaginary image as inputs. To obtain the images, we stack the real components of our color elements to form the real image and the imaginary components for the imaginary image. Hence, the resulting complex image I can be written as:

$$I = I_{re} + iI_{im} \quad (5)$$

where, the real component of an image is $I_{re} = \{\Re(I_h^*), \Re(I_s^*), \Re(I_v^*)\}$, and the imaginary component is $I_{im} = \{\Im(I_h^*), \Im(I_s^*), \Im(I_v^*)\}$. In Fig. 3, given an image, we pictorially illustrate how one can obtain these components to form the two inputs required for iCNNs. Fig. 4

Table 1. RGB → HSV → iHSV conversion of basic colors

Color	RGB	HSV	iHSV			
			I_h^*	I_s^*	I_v^*	
Red		(255, 0, 0)	(0.00, 1.00, 1.00)	$1.00 + i1.00$	$0.00 + i1.00$	$1.00 + i0.00$
Green		(0, 255, 0)	(2.09, 1.00, 1.00)	$1.00 + i1.00$	$2.09 + i1.00$	$-0.49 + i0.86$
Blue		(0, 0, 255)	(4.19, 1.00, 1.00)	$1.00 + i1.00$	$4.19 + i1.00$	$-0.50 - i0.86$
Cyan		(0, 255, 255)	(3.14, 1.00, 1.00)	$1.00 + i1.00$	$3.14 + i1.00$	$-1.00 + i1.22$
Yellow		(255, 255, 0)	(1.04, 1.00, 1.00)	$1.00 + i1.00$	$1.04 + i1.00$	$0.50 + i0.86$
Magenta		(255, 0, 255)	(5.23, 1.00, 1.00)	$1.00 + i1.00$	$5.23 + i1.00$	$0.50 - i0.86$
Black		(0, 0, 0)	(0.00, 0.00, 0.00)	$0.00 + i0.00$	$0.00 + i0.00$	$0.00 + i0.00$
Gray		(128, 128, 128)	(0.00, 0.00, 0.50)	$0.50 + i0.00$	$0.00 + i0.50$	$0.00 + i0.00$
White		(255, 255, 255)	(0.00, 0.00, 1.00)	$1.00 + i0.00$	$0.00 + i1.00$	$0.00 + i0.00$


 Figure 4. Real (I_{re}) and imaginary (I_{im}) components of sample images (I).

shows image visualization of both the components for sample images.

It is worth noting that the proposed conversion function, $I_{hsv} \rightarrow I_{hsv}^*$, is a one-to-one, invertible function. This can be proven by demonstrating that the original $I_{hsv}(p)$ values can be obtained from a given I_{hsv}^* , as shown below:

$$I_h = \frac{\Re(I_s^*)}{|I_v^*|} \Rightarrow \frac{\Re(I_s I_h + i I_v)}{|I_s \cos(I_h) + i I_s \sin(I_h)|} \Rightarrow \frac{I_s I_h}{I_s} \Rightarrow I_h \quad (6)$$

$$I_s = |I_v^*| \Rightarrow |I_s \cos(I_h) + i I_s \sin(I_h)| \Rightarrow I_s \quad (7)$$

$$I_v = \Re(I_h^*) \Rightarrow \Re(I_v + i I_s) \Rightarrow I_v \quad (8)$$

Note that the last step of Eqn.(6) assumes $I_s \neq 0$. When $I_s = 0$, the undefined case of 0/0 arises, exactly what we expect from HSV color model, since hue is undefined when saturation turns zero.

3.6. FCCN Architecture

Generating probabilities and calculating losses in image classification tasks requires real values, which poses a challenge for complex-valued features. Current methods

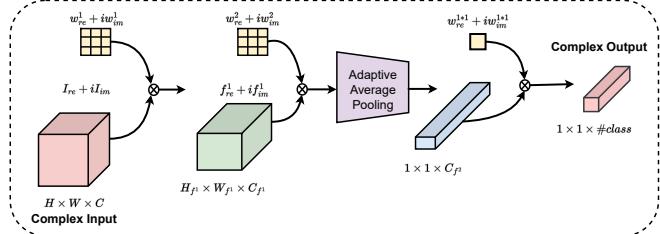


Figure 5. Blueprint of FCCNs: First, complex-valued convolutions are performed to obtain complex-valued feature maps. Then, adaptive average pooling is done to reduce the spatial dimensions to 1×1 . At the end, 1×1 complex-valued convolutions are performed to reduce the channel size to the number of classes.

address this challenge by converting complex-valued features to real-valued ones using different strategies. Some methods, such as those based on DCN [37], treat the real and imaginary components of complex-valued features as two separate real-valued features. Other methods, such as SurReal [2], employ manifold distance metrics. However, both of these strategies discard valuable phase information and undermine the end-to-end complex-valued information flow. To preserve complex-valued properties, we propose using convolutional layers to replace fully-connected ones. In order to accomplish this, we use complex convolution layers to obtain feature maps. Then, adaptive average pooling is done to reduce the spatial dimension of features to size 1×1 ($H \times W \times C \rightarrow 1 \times 1 \times C_f$, where $C = \#channels$, $H = height$, and $W = width$). Since 1×1 convolution is equivalent to fully connected operation, we can use 1×1 convolutions to replace them [26]. This way, we can preserve the complex-valued flow and eventually reduce the number of channels to the number of classes ($1 \times 1 \times \#class$). We denote this 1×1 convolution part as f_{conv} . In Fig. 5, we give a blueprint of this approach to better understand how FCCNs are fully complex-valued and convolutional in a true sense.

3.7. Complex-valued Cross-entropy Loss (\mathcal{L}_{comp})

Let us consider our FCCN as a classifier function $f_{comp} : \mathbb{C}^{H \times W \times 3} \rightarrow \mathbb{C}^c$, solving a c -class classification problem. Let α denote the set of parameters of the classifier, and let y_{jk} denote k^{th} element of one-hot encoded label of j^{th} training sample. We cannot directly use the cross-entropy loss for complex-valued classifier f_{comp} , which has a complex-valued output. So, we modify cross entropy by decomposing $z_j = f_{comp}(x_j; \alpha)$, where z_j is the complex-valued vector output of f_{comp} . We now define our new complex loss function as \mathcal{L}_{comp} and derive it as follows:

$$\begin{aligned}\mathcal{L}_{comp} &= -\frac{1}{n} \sum_{j=1}^n \sum_{k=1}^c y_{jk} \log z_{jk} \\ &= -\frac{1}{n} \sum_{j=1}^n \sum_{k=1}^c y_{jk} \log(|z_{jk}| e^{i\theta_{jk}})\end{aligned}\quad (9)$$

where we express our z in polar form. This can now be rewritten as follows by considering $|z|$ s as the required probabilities for the classification task.

$$\mathcal{L}_{comp} = \mathcal{L}_{CE} - i \frac{1}{n} \sum_{j=1}^n \sum_{k=1}^c y_{jk} \theta_{jk} \quad (10)$$

where \mathcal{L}_{CE} is the usual real-valued cross-entropy loss, and the other term serves as imaginary loss.

It is worth noting that the magnitude of the complex-valued output vector z_j is constrained by $\sum_{k=1}^c |z_{jk}| = 1$ and $|z_{jk}| \geq 0 \forall j, k$. To handle this constraint and ensure that the probability distribution of the output is valid, we first separate the magnitude and phase components of the complex vector. Next, we use softmax to create a probability distribution for the magnitude while leaving the phase information unchanged. Once we obtain the probability distribution in the form of magnitudes, we can apply the standard cross-entropy loss to the real part of \mathcal{L}_{comp} . To ensure that our proposed loss function is differentiable, it is necessary to demonstrate that it adheres to the Cauchy-Riemann equations [36]. These equations pertain to a complex function $f(x, y) = u(x, y) + iv(x, y)$, where x and y are components of a complex number $z = x + iy$. Specifically, the equations require that:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \quad \text{and} \quad \frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y}$$

We provide complete proof of the differentiability of our proposed loss function in the supplementary material of this paper.

In complex domain, label vector y_j can be represented as $y_j e^{i0}$, suggesting that our loss should minimize the deviation between $y_j e^{i0}$ and z_j . It is clear that real part minimizes

deviation between their magnitudes through \mathcal{L}_{CE} . On the other hand, when $y_{jk} = 1$, the imaginary part forces θ_{jk} to be 2π , which is equivalent to 0 because $e^{i(2\pi+\theta)} = e^{i\theta}$. That is when the minimum imaginary loss (-2π) can occur, so \mathcal{L}_{comp} also minimizes the deficit between phases. Notably, our proposed loss function differs from traditional approaches that discard phase information in favor of simpler representations. Instead, by explicitly minimizing the imaginary loss, our model can effectively leverage phase information in the complex domain.

Optimization: There are two types of gradients at the last layer: one for $|z|$ s and another for θ s. In our loss function, since the real part entirely relies on $|z|$ s and the imaginary part entirely relies on θ s, we assign the gradients of real part to $|z|$ s and the gradients of the imaginary part to θ s.

4. Experiments

In this section, we present the evaluation of our proposed Fully Convolutional Complex-valued Networks (FCCNs) and compare their performance with other methods on both real-valued and complex-valued datasets.

4.1. Datasets & Model Architectures

We employ a total of five diverse image classification datasets to evaluate the performance of our proposed method: CIFAR-10 [23], CIFAR-100 [24], STL-10 [6], SVHN [28], and Tiny-ImageNet [25]. CIFAR-10 comprises ten categories, each with 5k training and 1k testing images. SVHN also has ten categories, but it contains a larger number of images: 73.2k for training and 26k for testing in total. Similarly, STL-10 contains ten categories, with each category having 500 images for training and 800 for testing. On the other hand, CIFAR-100 has 100 categories, with each category having 500 training and 100 testing images. Finally, Tiny-ImageNet, a subset of the Original ImageNet, contains 200 categories. In addition, we conduct experiments on the MSTAR [32] dataset, which is a complex-valued SAR dataset consisting of 10 different categories, each with images of size 128×128 .

In our experiments, we adopt the Adam optimizer [21] with a learning rate of 10^{-3} and weight decay of 0.1 for training all models to 100 epochs, unless otherwise stated. We use CReLU activation [37] and complex batch normalization in our architectures. To demonstrate the effectiveness of our proposed approach, we evaluate our method on both smaller and larger networks and compare our results with existing approaches. For smaller networks, we experiment with different versions of CIFARnet (a small CNN with 3 convolutional layers and stride=2). Similarly, following [34], for larger networks, we use the CDS-large as the baseline for comparison.

Table 2. Comparing classification accuracies of different approaches using CIFARnet [34] while varying the input strategy on five benchmark real-valued datasets. The values in **blue** are best, and the ones in **red** are second best. Our FCCN approach with iHSV inputs obtains the best results on every dataset. The second best results are achieved using our FCCN approach with sliding [34] input strategy on 3/5 datasets.

Method	STL-10				CIFAR-10				CIFAR-100				SVHN				Tiny-ImageNet			
	RGB	HSV	LAB	Sliding	iHSV	RGB	HSV	LAB	Sliding	iHSV	RGB	HSV	LAB	Sliding	iHSV	RGB	HSV	LAB	Sliding	iHSV
DCN [37]	31.80	31.25	32.20	32.42	33.81	65.17	64.37	58.64	63.83	66.71	32.52	31.46	27.36	28.87	32.94	85.26	85.31	84.43	87.44	87.59
SurReal [2]	-	-	-	-	50.68	-	53.02	54.61	-	23.57	-	25.97	26.66	-	80.51	-	53.48	80.79	-	-
CNN [22]	45.14	44.03	46.66	40.47	41.45	64.43	63.82	63.00	63.43	64.80	31.57	30.16	31.72	31.93	33.28	87.47	87.05	84.93	87.37	88.49
CDS [34]	41.23	40.32	46.54	47.38	45.60	69.23	66.74	67.58	69.19	68.47	41.83	38.31	39.52	42.08	39.24	89.39	87.64	88.86	90.25	89.41
FCCN	59.60	53.67	54.53	58.84	60.61	77.85	74.23	76.76	78.01	78.23	43.40	41.59	42.03	44.19	47.34	89.88	88.17	89.16	90.81	90.95

Table 3. Comparing classification accuracies(%) of different approaches using CDS-large [34], a large network architecture, on CIFAR-10 dataset. Our proposed FCCN outperforms all other approaches (DCN [37], CNN [22], and CDS [34]).

Method	DCN [37]	CNN [22]	CDS [34]	FCCN
Accuracy (%)	92.8	93.0	93.7	94.3

We aim to demonstrate the scalability and applicability of our proposed method FCCN. Hence, we also conduct experiments on large-scale ImageNet [9] dataset and MSTAR [32] (SAR dataset). The ImageNet dataset comprises 1,281,167 training images and 50,000 validation images with 1000 classes, while the MSTAR dataset consists of naturally occurring complex images.

4.2. Results on Benchmark Datasets

We compare our proposed method with difference approaches (DCN [37], SurReal [2], CNN [22], and CDS [34]) based on CIFARnet architecture, as shown in Table 2. For detailed analysis, we provide five different inputs. RGB and HSV are regular color models, LAB input for networks is obtained as $\{L + i0, a + ib\}$, while sliding input is created as $\{R + iG, G + iB\}$ following CDS [34], and iHSV is our proposed color model. On CIFAR-10, CIFAR-100, STL-10, SVHN, and Tiny-ImageNet, for all inputs, our FCCN obtains higher accuracy than other methods. Our FCCN approach, along with our iHSV inputs, obtains the best results on every dataset. The second best results are achieved using our FCCN approach with sliding [34] input strategy on 3/5 datasets and with RGB input on the remaining datasets. We also compare FCCN with other methods using CDS-Large [34] architecture on CIFAR-10 dataset in Table 3. We again obtain higher accuracy than all other methods.

4.3. Experiments on ImageNet

We are the first to perform a large-scale experiment using complex-valued networks for color images. Top-1 and Top-5 classification accuracies are reported in Table 4. We evaluate the performance of our proposed FCCN on large architectures: CDS-large, ResNet18, ResNet-50, and ResNet-152. Using CDS-large architecture, we compare with CDS [34] and DCN [37]. Using the rest of the architectures, we

compare with DCN [37] and CNN [13]. The experimental results demonstrate that our proposed FCCN consistently outperforms competing methods.

We use NVIDIA A100 GPU in these experiments. All the models are trained with a batch size of 256 using SGD optimizer, with 0.9 momentum. The input size is 224×224 for all images. We used random crop, random horizontal flip, and random rotation augmentations. The network was randomly initialized and trained for up to a total of 50×10^4 iterations, and the learning rate starts from 0.1 and is divided by 10 when the error plateaus.

Table 4. Comparing Top-1 and Top-5 classification accuracies of our approach with different approaches (CDS [34], DCN [37], CNN [13]) using multiple large network architectures on large Imagenet-1k dataset. FCCN performs best again regardless of the network architecture used.

CDS-large [34]	ResNet18 [13]		ResNet50 [13]		ResNet152 [13]	
	DCN	CDS	FCCN	CNN	DCN	FCCN
Top-1	49.76	53.25	64.92	72.64	68.54	73.41
Top-5	62.38	76.59	85.24	90.73	87.61	91.95

4.4. Results on MSTAR

The MSTAR (Moving and Stationary Target Acquisition and Recognition) [32] is a widely used benchmark dataset comprising SAR images, which are complex-valued. We follow the same experimental setup as that of [34] to make comparisons on this dataset. In Table 5, we report the results of different approaches, both real-valued (CNN [22]) and complex-valued (SurReal [2], DCN [37], CDS [34] and our FCCN). We use CDS-large [34] architecture for this experiment. It is clear from the table that our FCCN performs the best, indicating our enhancements (1×1 complex-valued convolutions and complex-valued loss function) are indeed significant contributions to iCNN literature.

4.5. Ablation Study

We conduct four ablation studies to highlight our contributions. First three ablation studies are conducted on CIFAR-10 dataset (following [37]), and the fourth study is conducted on five different datasets. In all four ablation studies, we used standard architectures: AlexNet, VGG-16, VGG-19, ResNet-18, and ResNet-50.

Table 5. Comparing classification accuracies of different approaches on MSTAR, a complex-valued benchmark dataset.

Method	CNN [22]	SurReal [2]	DCN [37]	CDS [34]	FCCN
Accuracy (%)	66.9	94.9	89.1	96.1	97.4

First, we study the contributions of different complex-valued channels of our iHSV color space. Second, we study how the two components, real and imaginary, of images contribute. Third, we try to highlight the three contributions we have made in this paper. Finally, we study the overall effect of improvements we make over baseline DCN [37] through experiments on different datasets using different standard architectures.

Effects of different channels: In this ablation study, we analyze the contribution of each complex-valued channel in I_{hsv}^* by inputting only one or a pair of them to FCCN. Table 6 summarizes the results. It is observed that I_h^* and I_s^* perform better individually than I_v^* , but still require the assistance of the other complex-valued channels to achieve optimal performance. Moreover, different pair-wise combinations of channels peak with different architectures, and once again, the third channel’s contribution is found to be crucial for the best performance. These results suggest that all three complex-valued channels play an essential role in improving the overall performance of FCCN.

Effects of I_{re} and I_{im} : We also examined the individual contribution of the real and imaginary components of the I_{hsv}^* image representation. We input only one component at a time to FCCN and evaluated their impact on the overall performance. The results, as summarized in Table 6, indicate that both real and imaginary components are necessary for best performance and make almost equal contributions.

Effects of $iHSV$, f_{conv} and \mathcal{L}_{comp} : In this study, we add each of our main contributions one-by-one to assess their significance. Our main contributions are (i) complex-valued color model ($iHSV$), (ii) 1×1 convolution (f_{conv}), and (iii) complex-valued loss function (\mathcal{L}_{comp}). When $iHSV$ is absent, complex input has only RGB as real part, and imaginary part of input is made zero. When f_{conv} is absent, we use the same strategy as DCN [37]. In the absence of \mathcal{L}_{comp} , cross-entropy loss is used, and we ensure real-valued outputs by having 1×1 convolution in the end. The results in Table 7 demonstrate that adding our contributions one-by-one steadily increases the performance.

Extensive comparisons with baseline: In this study, we extensively compare with our baseline DCN [37] on five different datasets using five standard architectures. As evident from the results in Table 8, we can conclude that our improvements over baseline DCN [37] are indeed helpful.

Table 6. Results of our ablation study that highlights the value of different parts of our $iHSV$ color representation, which can be divided in two ways: channel-wise $\{I_h^*, I_s^*, I_v^*\}$ and component-wise $\{I_{re}, I_{im}\}$. We conduct this study using a wide-range of standard architectures on CIFAR-10 dataset.

I_h^*	I_s^*	I_v^*	I_{re}	I_{im}	AlexNet	VGG-16	VGG-19	ResNet-18	ResNet-50
✓					81.66	90.07	91.80	89.81	89.24
	✓				81.12	90.97	91.49	88.73	88.86
		✓			71.15	81.17	80.85	78.23	79.22
			✓		86.75	88.58	91.16	91.41	88.39
			✓	✓	85.08	90.21	91.36	90.91	88.45
			✓	✓	82.79	90.06	91.03	90.18	90.53
				✓	79.33	91.64	91.81	90.45	91.84
				✓	79.38	91.36	91.31	91.54	91.69
✓	✓	✓	✓	✓	89.36	92.71	93.25	93.35	93.58

Table 7. Results of our ablation study that highlights the value of different components of our proposed method: $iHSV$, f_{conv} and \mathcal{L}_{comp} . We conduct this study using a wide-range of standard architectures on CIFAR-10 dataset.

$iHSV$	f_{conv}	\mathcal{L}_{comp}	AlexNet	VGG-16	VGG-19	ResNet-18	ResNet-50
			79.71	90.26	90.35	90.54	90.17
✓			80.54	90.61	90.29	90.01	90.87
✓	✓		82.33	91.54	91.05	91.17	91.79
✓	✓	✓	89.36	92.71	93.25	93.35	93.58

Table 8. All our improvements are over DCN [37], our baseline method. Here, we study the overall affect of our improvements using a wide-range of standard architectures on different benchmark datasets.

Dataset	Method	AlexNet	VGG-16	VGG-19	ResNet-18	ResNet-50
CIFAR-10	DCN [37]	80.55	91.74	92.26	86.89	88.61
	FCCN	89.36	92.71	93.25	93.35	93.58
STL-10	DCN [37]	66.97	72.84	70.49	65.31	66.32
	FCCN	71.03	76.42	76.62	76.48	76.22
SVHN	DCN [37]	92.16	95.51	95.34	94.58	95.41
	FCCN	92.54	94.76	95.12	96.31	96.38
CIFAR-100	DCN [37]	58.14	67.18	66.13	68.45	68.63
	FCCN	65.17	69.31	71.77	73.63	74.96
Tiny-ImageNet	DCN [37]	46.09	50.82	52.33	54.26	55.28
	FCCN	48.93	55.38	55.57	59.84	60.70

4.6. Model performance Analysis

Scalability: As evident from the discussions above, it can be observed that we have conducted experiments ranging from small models with small datasets (CIFARnet with STL-10) to large models with large datasets (ResNet-152 with ImageNet). In all these experiments, we observed superior results for our proposed approach. Thus, our proposed approach is definitely scalable. We also observe that on small-scale datasets, such as STL-10, where the number of training examples is less, our approach achieves considerable performance improvements (9-13%) using small (see Table 2) and large models (see Table 8), demonstrating the effectiveness of FCCN in low-resource scenarios.

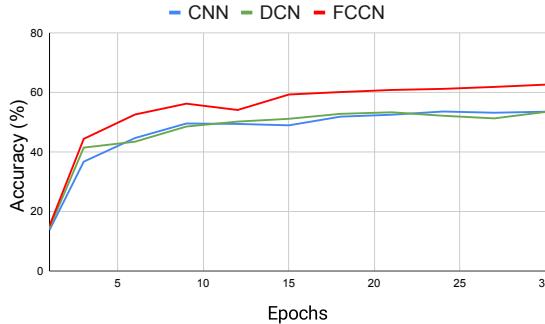


Figure 6. Accuracy curves obtained using DCN, CNN, and FCCN approaches on ImageNet-1k, revealing the faster learning capability of FCCN.

Faster Learning: From the accuracy plots of CNN, DCN, and FCCN approaches shown in Fig. 6, it’s clear that our FCCN approach exhibits faster learning capability compared to the other two. Here, we kept the learning rate constant (0.1) while training ResNet-50 models using these approaches for 30 epochs on the ImageNet-1K dataset.

5. Conclusion

In this work, we have presented fully complex-valued convolutional networks (FCCN), which operate entirely in the complex domain. It takes complex-valued image inputs using our novel complex-valued color model, iHSV, and is optimized using our novel complex-valued loss function. By maintaining such end-to-end consistency in complex-valued nature, our proposed approach outperforms previous works on complex-valued networks with significant margins on numerous datasets, including ImageNet-1k.

Acknowledgement

This work was supported in part by the Infosys Centre for Artificial Intelligence, IIIT-Delhi, and in part by Junior Research Fellowship (JRF) scheme of the University Grants Commission (UGC).

References

- [1] Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In *International Conference on Machine Learning*, pages 1120–1128. PMLR, 2016. 1
- [2] Rudrasis Chakraborty, Yifei Xing, and X Yu Stella. Surreal: complex-valued learning as principled transformations on a scaling and rotation manifold. *IEEE Transactions on Neural Networks and Learning Systems*, 2020. 2, 5, 7, 8
- [3] Guangyao Chen, Peixi Peng, Li Ma, Jia Li, Lin Du, and Yonghong Tian. Amplitude-phase recombination: Rethinking robustness of convolutional neural networks in frequency domain. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 458–467, 2021. 2
- [4] Brian Cheung, Alexander Terekhov, Yubei Chen, Pulkit Agrawal, and Bruno Olshausen. Superposition of many models into one. *Advances in neural information processing systems*, 32, 2019. 2
- [5] Hyeong-Seok Choi, Jang-Hyun Kim, Jaesung Huh, Adrian Kim, Jung-Woo Ha, and Kyogu Lee. Phase-aware speech enhancement with deep complex u-net. In *International Conference on Learning Representations*, 2018. 2
- [6] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 6
- [7] Elizabeth Cole, Joseph Cheng, John Pauly, and Shreyas Vasanawala. Analysis of deep complex-valued convolutional neural networks for mri reconstruction and phase-focused applications. *Magnetic resonance in medicine*, 86(2):1093–1109, 2021. 1
- [8] Ivo Danihelka, Greg Wayne, Benigno Uria, Nal Kalchbrenner, and Alex Graves. Associative long short-term memory. In *International Conference on Machine Learning*, pages 1986–1994. PMLR, 2016. 1
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 7
- [10] Jingkun Gao, Bin Deng, Yuliang Qin, Hongqiang Wang, and Xiang Li. Enhanced radar imaging using a complex-valued convolutional neural network. *IEEE Geoscience and Remote Sensing Letters*, 16(1):35–39, 2018. 1
- [11] George M Georgiou and Cris Koutsougeras. Complex domain backpropagation. *IEEE transactions on Circuits and systems II: analog and digital signal processing*, 39(5):330–334, 1992. 1
- [12] Daichi Hayakawa, Takashi Masuko, and Hiroshi Fujimura. Applying complex-valued neural networks to acoustic modeling for speech recognition. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1725–1731. IEEE, 2018. 1
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 7
- [14] Akira Hirose and Shotaro Yoshida. Comparison of complex- and real-valued feedforward neural networks in their generalization ability. In *International Conference on Neural Information Processing*, pages 526–531. Springer, 2011. 2
- [15] Akira Hirose and Shotaro Yoshida. Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence. *IEEE Transactions on Neural Networks and Learning Systems*, 23(4):541–551, 2012. 2
- [16] Yanxin Hu, Yun Liu, Shubo Lv, Mengtao Xing, Shimin Zhang, Yihui Fu, Jian Wu, Bihong Zhang, and Lei Xie. Dccrn: Deep complex convolution recurrent network for phase-aware speech enhancement. 2020. 1
- [17] Lai Jiang, Zhe Wang, Mai Xu, and Zulin Wang. Image saliency prediction in transformed domain: A deep complex

- neural network method. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8521–8528, 2019. 2
- [18] Lai Jiang, Mai Xu, Shanyi Zhang, and Leonid Sigal. Deepet: A novel deep complex-valued network with learnable transform for video saliency prediction. *Pattern Recognition*, 102:107234, 2020. 2
- [19] Mario Jojoa, Begonya Garcia-Zapirain, and Winston Percy-brooks. A fair performance comparison between complex-valued and real-valued neural networks for disease detection. *Diagnostics*, 12(8):1893, 2022. 1, 2
- [20] Taehwan Kim and Tülay Adalı. Approximation by fully complex multilayer perceptrons. *Neural computation*, 15(7):1641–1666, 2003. 2
- [21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 6
- [22] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 7, 8
- [23] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). 6
- [24] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-100 (canadian institute for advanced research). 6
- [25] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 6
- [26] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. 3, 5
- [27] Sindy Löwe, Phillip Lippe, Maja Rudolph, and Max Welling. Complex-valued autoencoders for object discovery. *Transactions on Machine Learning Research*. 1, 2
- [28] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 6
- [29] Kien Nguyen, Clinton Fookes, Sridha Sridharan, and Arun Ross. Complex-valued iris recognition network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):182–196, 2022. 2
- [30] Alan V Oppenheim and Jae S Lim. The importance of phase in signals. *Proceedings of the IEEE*, 69(5):529–541, 1981. 2
- [31] David P. Reichert and Thomas Serre. Neuronal synchrony in complex-valued deep networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. 2
- [32] Timothy D Ross, Steven W Worrell, Vincent J Velten, John C Mossing, and Michael Lee Bryant. Standard sar atr evaluation experiments using the mstar public release data set. In *Algorithms for Synthetic Aperture Radar Imagery V*, volume 3370, pages 566–573. SPIE, 1998. 6, 7
- [33] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. 3
- [34] 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*, pages 681–690, 2022. 2, 6, 7, 8
- [35] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015. 2
- [36] Elias M Stein and Rami Shakarchi. *Complex analysis*, volume 2. Princeton University Press, 2010. 6
- [37] 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 *International Conference on Learning Representations*, 2018. 2, 3, 5, 6, 7, 8
- [38] 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. 2
- [39] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016. 2
- [40] Bhavya Vasudeva, Puneesh Deora, Saumik Bhattacharya, and Pyari Mohan Pradhan. Compressed sensing mri reconstruction with co-vegan: Complex-valued generative adversarial network. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 672–681, 2022. 1