# Tropical Cyclone Intensity Estimation Using a Deep Convolutional Neural Network

Ritesh Pradhan, Ramazan S. Aygun, *Senior Member, IEEE*, Manil Maskey, *Member, IEEE*, Rahul Ramachandran, *Senior Member, IEEE*, and Daniel J. Cecil

***Abstract*— Tropical cyclone intensity estimation is a challenging task as it required domain knowledge while extracting features, significant pre-processing, various sets of parameters obtained from satellites, and human intervention for analysis. The inconsistency of results, significant pre-processing of data, complexity of the problem domain, and problems on generalizability are some of the issues related to intensity estimation. In this study, we design a deep convolutional neural network architecture for categorizing hurricanes based on intensity using graphics processing unit. Our model has achieved better accuracy and lower root-mean-square error by just using satellite images than 'state-of-the-art' techniques. Visualizations of learned features at various layers and their deconvolutions are also presented for understanding the learning process.**

***Index Terms*— Deep learning, image processing, convolutional neural networks, tropical cyclone category and intensity estimation.**

## I. Introduction

DEEP learning uses a deep architecture of multiple processing layers composed of linear or nonlinear transformations [1]–[6] while replacing handcrafted features with automated feature learning and hierarchical feature extraction [7]. Convolutional Neural Networks (CNNs) can be used to model spatial correlation with translation invariance making them suitable for image recognition [8], [9]. This study proposes a deep CNN architecture for estimating the *hurricane*[1] intensity by learning features.

### A. Motivation

Since hurricanes (or tropical cyclones) possess substantial threats and cause significant damage to lives and properties, studying the stages of a hurricane is important to determine its impact. From a scientific perspective, determining an accurate TC intensity helps i) better initialization in forecast models, leading to more accurate forecasts, ii) more accurate historical records of TCs, especially if a technique can be consistently applied to older satellite imagery (i.e., intensity reanalysis), and iii) providing consistent intensity estimates as current intensity estimates are made via a subjective algorithm (Dvorak technique) that is applied inconsistently in different forecast areas. Initial errors are too high, especially for weak and storms that are transitioning in structure.

In this study, we use Saffir-Simpson Hurricane Wind Scale (SSHWS) (provided in Table I) along with intensity categorization for *tropical storm* and *tropical depression* as tropical cyclone (TC) intensity categories. Since TC intensity is based on *maximum wind speeds* (MWS), estimating the TC intensity by just using image content is a challenging problem. There are a number of techniques that utilize satellite imagery for estimating tropical cyclone intensity using Dvorak [10], [11] and deviation-angle variance technique (DAVT) [12] techniques.

The main assumption of the *Dvorak* method is that cyclones with similar intensity tend to have a similar pattern.

TABLE I

Saffir-Simpson Hurricane Wind Scale and Related Classifications

| Category | Symbol | Wind speeds | Damage |
|---|---|---|---|
| Five | H5 | $\geq$ 137 knots | Catastrophic |
| Four | H4 | 113–136 knots | Catastrophic |
| Three | H3 | 96–112 knots | Devastating |
| Two | H2 | 83–95 knots | Extensive |
| One | H1 | 64–82 knots | Significant |
| Tropical storm | TS | 34–63 knots | Significant |
| Tropical depression | TD | 20–33 knots | Small |
| No Category | NC | $\leq$ 20 knots | - |

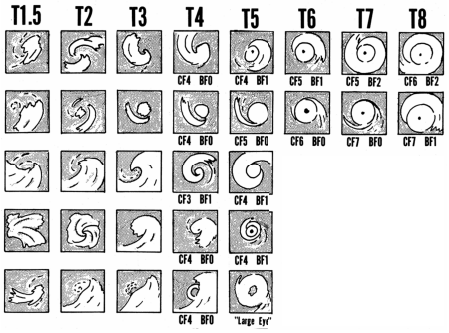


Fig. 1. Illustration of common development patterns and their intensities according to the Dvorak technique [13].

Manuscript received October 20, 2017; accepted October 20, 2017. Date of publication October 25, 2017; date of current version November 14, 2017. This work was supported by the NASA Earth Science Technology Office and its Advanced Information Systems Technology Program under Grant NNM11AA01A. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Adrienne Fisher. *(Corresponding author: Ritesh Pradhan.)*

R. Pradhan and R. S. Aygun are with The University of Alabama in Huntsville, AL 35899 USA (e-mail: ritesh.pradhan@uah.edu; aygunr@uah.edu).

M. Maskey, R. Ramachandran, and D. J. Cecil are with the NASA Marshall Space Flight Center Huntsville, AL 35812 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2017.2766358

[1]We use tropical cyclone, TC, cyclone, and hurricane interchangeably in this paper.

Figure 1 [13] shows some development patterns used by the Dvorak technique. Once a pattern is detected over a 24-hour period, the features such as length and banding from the storm are further analyzed to reach a particular T-number [14]. This relates tropical cloud structures to storm intensity. Nevertheless, this technique is not perfect and still suffers from subjective biases. Due to inherent limitations of the empirical method used, it cannot determine subtropical cyclone intensity. Today, with successful application of the Dvorak technique for more than 30 years along with some modifications and improvements, it is used worldwide for TC intensity estimation. The Advanced Dvorak Technique [15] provides a nearly instantaneous estimate of TC intensity in an objective manner. It removes a large amount of the subjectivity inherent in the process and produces errors similar to a human in most cases.

On the other hand, the *deviation angle variance technique (DAVT)* technique quantifies the axis symmetry of tropical cyclones in infrared (IR) satellite imagery and performs a type of directional gradient statistical analysis of the brightness of IR images. While initial versions assumed that the center is available as a reference for computations, later the center finding was automated [16]. DAVT techniques use different models and parameters for different regions of tropical cyclones. The DAVT technique was described and applied to the North Atlantic region by Pineros *et al.* (2008, 2011), [17], [18] and Ritchie *et al.* (2012) [16] and to the North Pacific region by Ritchie *et al.* (2014) [12]. Ritchie *et al.* (2014) [12] proposed two different fit models for *eastern North pacific region* and *western North Pacific region*. However, using best-track centers at 6-hour intervals is not reliable as cyclones change its form and path continuously and frequently.

In this study, we focus on obtaining higher accuracy and lower root mean squared (RMS) intensity error than previously used techniques such as DAVT [12] and Dvorak techniques [10], [11]. The efficiency of those techniques [10]–[12] depends on the knowledge of a person using the technique. Although TC intensity estimation has been greatly studied and many techniques have been devised [10]–[12], [18], determining the intensity of TCs with high confidence and accuracy is still challenging. The major issues with previous approaches are *inconsistency, significant pre-processing, complexity,* and *generalizability.*

*1) Inconsistency:* Previous techniques provide different root-mean-square intensity error (RMSE) values for different regions. The deviations in these RMSE values for different regions make the results inconsistent.

*2) Significant Pre-Processing:* Dvorak technique requires a 24-hour change in pattern before the pre-estimation of the intensity. Similarly, DAVT requires hurricane images with marked hurricane centers. So, evaluation and estimation require significant pre-processing steps like measurement of the cloud system, eye, curve, shear and 24-hour changes in the pattern.

*3) Complexity:* Dvorak technique depends on different geophysical properties and DAVT relies on finding the eye (or center) of the cyclone. The number of constraints in these techniques makes it hard to implement. Usually, domain knowledge is required for these techniques.

*4) Generalizability:* Different models for hurricanes in different regions are used in techniques like Dvorak and DAVT. Dvorak works in tropical regions but not in subtropical regions. Similarly, DAVT [12] proposed two different fitting models for western and eastern North Pacific regions. Developing a method or model that works for all regions or types of hurricanes is important for generalizability.

### *B. Our Contribution*

Successful applications of convolutional neural networks for image recognition with high accuracy [9], [19]–[23], motivated us using convolutional neural network for hurricane intensity analysis. To the best of our knowledge, deep convolutional neural networks have not been studied for hurricane intensity analysis. In this study, we present a deep convolutional network architecture for estimating tropical cyclone intensity without relying on domain knowledge while extracting features, signiï'cant pre-processing, various sets of parameters obtained from satellites, and human intervention for analysis. The critical features such as the curvature, bend, eye, color intensity, pattern, *etc* that are required to estimate the intensity of TC are automatically extracted from a series of feature maps generated in each step of convolution. In addition, the convolutional technique is so fast that *lenet* [9] can be trained with a huge dataset in a matter of hours. We can harness the computation power of Graphics Processing Unit (GPU) that drastically reduces the computation time despite a number of sequential layers. The applicability of our model for all regions (Atlantic and Pacific) validates its generalizability. This model corrects itself by backpropagation in case of misclassifications. We also present visualization of feature maps from each layer of convolution using *deep visualization toolbox* [24]. Our model, using *caffe* framework [25] with GPU, can estimate the intensity of any new cyclone in less than a second. Not only it obtained better accuracy but it also achieved lower RMS intensity errors value than other recent and old techniques. Our model analyzes images and then provides the TC category.

This paper is organized as follows. The following section provides information about deep CNNs. Section 3 explains our layers and architecture of our deep CNN. The experimental results, visualizations, and performance are provided in Section 4. The discussion on problematic cases and limitations are covered in Section 5. The last section concludes our paper.

## II. Deep Convolutional Neural Network

Convolutional Neural Networks (CNNs) have been successfully applied for processing two-dimensional visual data [2], [9], [26], [27]. Since CNNs are inspired by the organization of animal visual cortex in biological processes, many neurally-inspired models can be found in the literature (e.g., LeCun *et al.* [27]). The overlapping sub-regions of the visual field, called receptive fields, are obtained through the collection of small neurons across multiple layers in a CNN. This overlapping mechanism allows CNNs to tolerate the translation of an input image.

CNNs are variants of multilayer perceptrons [9], [27]. A convolutional network consists of various combinations

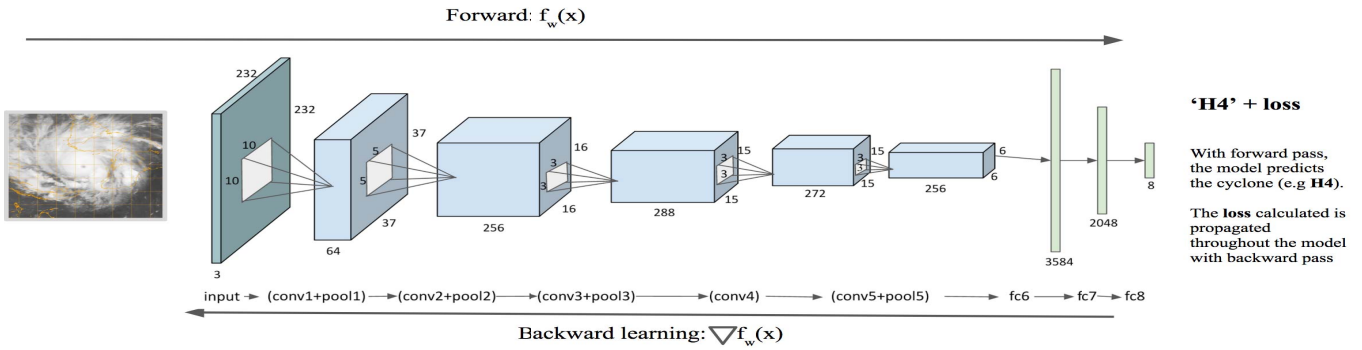


Fig. 2. Network architecture for hurricane intensity estimation showing different steps of convolution and pooling.

TABLE II
CONFIGURATION OF OUR CONVOLUTIONAL NETWORK

| Layer | Shape | Output Size | Parameter Shape | Parameters |
|---|---|---|---|---|
| Input | 3@232x232 | | | |
| conv1 | 64@10x10, s=3, p=0 | 75x75 | (64, 3, 10, 10) | 19,264 |
| pool1 | 3x3, s=2, p=0 | 37x37 | | |
| conv2 | 256@5x5, s=1, p=0 | 33x33 | (256, 64, 5, 5) | 409,856 |
| pool2 | 3x3, s=2, p=0 | 16x16 | | |
| conv3 | 288@3x3, s=1, p=1 | 16x16 | (288, 256, 3, 3) | 663,840 |
| pool3 | 2x2, s=1, p=0 | 15x15 | | |
| conv4 | 272@3x3, s=1, p=1 | 15x15 | (272, 288, 3, 3) | 705,296 |
| conv5 | 256@3x3, s=1, p=0 | 13x13 | (256, 272, 3, 3) | 528,984 |
| pool5 | 3x3, s=2, p=0 | 6x6 | | |
| fc6 | 3584 | | (3584, 9216) | 27,872,768 |
| fc7 | 2048 | | (2048, 3584) | 7,342,080 |
| fc8 | 8 | | (8, 2048) | 16,392 |
| | | | | **37,558,480** |

of convolutional and fully connected layers. The number of feature maps generated by a convolution layer is equal to the number of its kernels [9]. The outputs (feature maps) of a series of convolutional layers are fed into fully connected layers of a neural network for classification. The loss layer then penalizes the deviation between actual and classified labels. Generally, a deep neural network is designed as a feed-forward network and can be trained with the backpropagation algorithm. As the errors are backpropagated in a network, the network is optimized by updating the weights and biases to minimize the loss function. Use of shared parameters (weight and bias) with sparse connectivity reduces the number of free parameters to be learned, thus improves the computational performance [9], [27]. This also helps address overfitting problem.

Pooling layer is an important layer in a deep CNN and typically follows the convolutional layer. Max pooling has been used frequently in the literature [9], [22], [28]. Max pooling reduces the number of features by selecting the feature that has the maximum value among a set of features. Pooling layer also improves the robustness of the network.

Various methods [3], [5], [22], [29] have successfully used backpropagation method of Kelley [30]. Today many pattern recognition tasks are achieved through a backpropagation-trained neural network. For the high-level abstraction of features and to learn complicated functions, deep architectures are widely used nowadays [1]–[6], [26]. Various deep learning architectures have produced state-of-the-art results on various computer vision tasks [4]. For example, the *LeNet-5* network has been successfully applied to hand-written number recognition. Deep learning can successfully unfold high-level abstractions of features and select useful features for learning [3], [5].

The complexity of a classification task and the depth of the network along with limited computing resources makes training deep CNNs difficult and lengthy. Since training a deep CNN may take significant time, GPUs have also been used for training convolutional neural networks for computer vision problems [23], [31]. Alex *et al.* (2012) [22] used GPU for training millions of high-resolution images and won ImageNet classification contest. Later, the model of Zeiler and Fergus (2013) [28] outperformed the model by Alex *et al.* [22].

## III. OUR DEEP CNN LAYERS AND ARCHITECTURE

The architecture of our deep CNN is shown in Figure 2. Details of the input shape, filter shape, stride, padding, output size and parameters are tabulated in Table II. It clearly provides the number of kernels used in each layer along with corresponding stride and padding. For example, 3-channel input of size 232×232 is convolved with *conv1* (64 kernels of size 10×10, with stride=3 and padding=0).This produces 64 feature maps of size 75×75. 19,264 parameters are learned

in this process. Applying *pool1* on this output produces maps of size 37×37 and so on. Around 37.5 million parameters are learned throughout the network.

### A. Overview of Layers

The weights of filters are initialized using a gaussian noise and learned through training. Convolutional layer convolves input with 3-dimensional filters and applies ReLU for non-linearity. Similarly, fully connected layer calculates the inner product and applies ReLU for non-linearity:

$$f(x) = max(0, x) \tag{1}$$

The convolutional output at layer $l$ is given in Equation 2, where $x_{ij}$ is the output unit at position $(i, j)$, $K * K$ is the size of filter, $w_{ab}$ is the value of weighted kernel at position $(a, b)$, $y^{l-1}_{(i+a)(j+b)}$ is a receptive field at position $(i+a)$, $(j+b)$ from layer $l-1$, and $B^l$ is the bias for layer $l$.

$$x^l_{ij} = \sum_{a=0}^{K-1} \sum_{b=0}^{K-1} w_{ab} y^{l-1}_{(i+a)(j+b)} + B^l \tag{2}$$

The hyperparameters such as the size of filters may vary at different layers. The spatial size of the output volume ($W_o$) is computed from these hyperparameters as in Equation 3, where $W_i$ is the input volume size, $k$ is the size of kernel applied with stride $s$ and padding $p$.

$$W_o = \frac{W_i - k + 2 * p}{s} + 1 \tag{3}$$

Pooling provides translation invariance, reduces the number of parameters through down-sampling [23], and helps avoid overfitting. In our deep CNN, max pooling layer follows the first, second, third and fifth convolutional layers.

We use the local response normalization (LRN) across channels. This helps achieve *lateral inhibition* by normalizing over local input regions [32]. Normalization [33] is done by dividing each input by:

$$\left(1 + \frac{\alpha}{n} \sum_i x_i^2\right)^{\beta}$$

where $n$ is the size of local region and $\alpha$ and $\beta$ are basic parameters. The units in the same position but from different channels are normalized in this way.

The last layer in Figure 2 consists of 8 units (equal to the number of distinct classes). We use *softmax* loss layer for computing multinomial logistic loss [25] to update the weight parameters that minimizes the loss function and to determine a single class out of K mutually exclusive classes as follows:

$$L = -\frac{1}{N} \sum_{n=1}^{N} log\left(\frac{\exp f_{y_n}}{\sum_{k=1}^{K} \exp f_k}\right) \tag{4}$$

where $y_n$ is the actual label for input $x_n$, $f_k$ is the $k^{th}$ element of the vector of class scores, and $f_{y_n}$ is the class score for $x_n$ corresponding to ${y_n}^{th}$ column.

Softmax takes input from a fully connected layer and produces a probability value per class. The loss is averaged over entire mini-batch computed from classified labels and actual labels. Then the final loss is computed by summing up total weighted loss over the network. Loss and gradient with respect to loss are computed with forward and backward passes, respectively.

We used stochastic gradient descent (SGD) technique to find the minimum cost iteratively. For mini-batch of $N$ dataset, the optimization is computed as the average loss over the mini-batch as:

$$L(W) \approx \frac{1}{N} \sum_{i}^{N} f_w(X^i) + \lambda r(W) \tag{5}$$

where $L(W)$ is the stochastic approximation of objective, $f_W(X^i)$ is the loss on data instance $X^i$, $r(W)$ is the regularization term, and $\lambda$ is the weight decay for the regularization term. SGD updates the weights by combining previous weights and the negative gradient of loss [22].

$$V_{t+1} = \mu V_t - \alpha \nabla L(W_t) \tag{6}$$

$$W_{t+1} = W_t + V_{t+1} \tag{7}$$

In Equation 6, the learning rate ($\alpha$) is the weight of the negative gradient, and the momentum ($\mu$) is the weight of its *previous update value* ($V_t$). In Equation 7, $W_{t+1}$ is the new updated weight using the previous weight ($W_t$) and the new updated value ($V_{t+1}$). These hyperparameters are used in our work as the basis of "Rule of Thumb" [34]. We use $\alpha = 0.001$ in the beginning and gradually decrease it by constant factor of 10 ($\gamma = 0.1$). The use of momentum smooths the weight updates across iterations and makes SGD stable and faster [25]. We used momentum value as $\mu = 0.9$. As given in Figure 2, the model computes $f_W$ in forward pass and the gradient $\nabla f_W$ in backward pass.

### B. Optimization

*1) Hyperparameters:* We tend to use larger convolution filter size for larger input and decrease the filter size gradually for higher layers. Layers near the input have fewer filters than that of the higher layers. However, the number of filters depends on the capacity of the network and the complexity of the task. In addition to convolution filters, we also need to choose the appropriate size of pooling filters. Large pooling filter drastically reduces the parameters. While large pooling filter may lead to a substantial loss in information, suitable filter size helps to mitigate overfitting. Determining the shape, size, and number of filters is always challenging. It is important to use the right level of granularity for the dataset considering the task complexity.

*2) Regularizations:* Regularization is a technique to prevent overfitting in machine learning by penalizing higher order features to smooth out the learning curve [22]. In our experiments, we have used the model obtained at around 90% validation accuracy for early stopping. Then we test our test dataset with this model. Sometimes, early stopping [1] may cause underfitting. Dropout method [2], [35] prevents overfitting and improves performance. We used a general dropout of $p = 0.5$ in our model.

TABLE III
CYCLONES USED FOR DATASET CREATION

| Region | Year | Cyclones |
|---|---|---|
| Atlantic | 1998 | Mitch |
| | 2003 | Isabel |
| | 2004 | Ivan |
| | 2005 | Emily, Katrina, Rita, Wilma |
| | 2007 | Dean, Felix |
| | 2010 | Alex, Bonnie, Colin, Danielle, Earl, Fiona, Five, Gaston, Igor, Julia, Karl, Lisa, Matthew, Nilcole, Otto, Paula, Richard, Shary, Tomas, Two |
| | 2011 | Arlene, Bret, Cindy, Don, Emily, Franklin, Gert, Harvey, Irene, Jose, Katia, Lee, Maria, Nate, Ophelia, Philippe, Rina, Sean, Ten |
| | 2012 | Alberto, Beryl, Chris, Debby, Ernesto, Florence, Gordon, Helene, Isaac, Joyce, Kirk, Leslie, Michael, Nadine, Oscar, Patty, Rafael, Sandy, Tony |
| | 2014 | Edouard |
| Pacific | 2002 | Elida, Fausto, Hernan, Kenna |
| | 2005 | Jova, Kenneth |
| | 2006 | Bud, Daniel, Ioke, John, Lane |
| | 2007 | Flossie |
| | 2008 | Hernan, Norbert |
| | 2009 | Felicia, Guillermo, Jimena, Rick |
| | 2010 | Celia, Darby |
| | 2011 | Adrian, Dora, Eugene, Hilary, Jova, Kenneth |
| | 2012 | Bud, Emilia, Miriam, Paul |

## IV. EXPERIMENTS

In this section, we explain our dataset, training and testing, visualization of features, and performance analysis.

### A. Dataset

Our dataset has two components: i) infrared (IR) hurricane images and ii) data for hurricanes. We formed our dataset by i) collecting information from different resources that have varying sampling rate, ii) fusing data into a single dataset, iii) interpolating hurricane data for images, and iv) augmenting additional images by transformations. We obtained *hurricane images* from tropical cyclone repository of the Marine Meteorology Division of U.S. Naval Research Laboratory (http://www.nrlmry.navy.mil). These satellite infrared (IR) images are captured around fifteen minutes apart and have additional information such as year, date, time and name of the hurricane.

We used HURDAT2 data (http://www.nhc.noaa.gov/data/#hurdat) to label images. This *hurdat2* is *Tropical Cyclone Best Track Reanalysis data*.[2] We also collected a different recon-only test dataset (http://www.nhc.noaa.gov/recon.php) for evaluating our model. This test set was totally based on the recon-informed hurricane date and time. This dataset was not used for training.

*1) Cyclones and Images in Dataset:* To build a single model for estimating intensity, we used cyclone images from 68 Atlantic cyclones and 30 Pacific cyclones from 1999 to 2014 (http://www.nrlmry.navy.mil/tcdat/), which are provided in Table III. To avoid the side-effect of unbalanced distribution of TC categories while training our deep CNN, we tried to balance the distribution by using storms that reach at least H3 category (there could be some exceptions). We collected 8,138 images for every 2 hours from 98 cyclones in Table III. Since hurdat2 data was available every 6 hours, we interpolated 6-hour hurdat2 data to obtain maximum wind speed at every two hours. This provided us labels (hurricane category) for images every two hours. For validation purposes, we provide a sample image difference at two hours apart and their Structural Similarity Index Measure (SSIM)[3] and Root Mean Squared Error (RMSE) values in Figure 3. With the help of best track data, all images were properly labeled on the basis of their respective maximum wind speed using Table I. Then we applied 5 image transformations (horizontal and vertical flips, and rotations of 90°, 180°, and 270° as shown in Figure 4) to increase the number of images to 48,828 images. We used the same hurricane data of the original image for these transformed images. Moreoever, separate 2,646 images were collected for testing recon-only dataset.

*2) Pre-Processing:* First, we cropped unnecessary text from those images. Then, we formed the maximal square images by removing the longer region from these rectangular image.

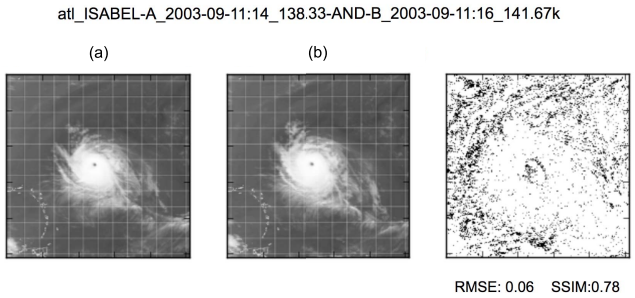


Fig. 3. RMSE, SSIM and pixelwise difference plot for images captured two hours apart. Hurricane Isabel from the Atlantic region (a) 2003-09-11:14 (138.33 kt) (b) 2003-09-11:16 (141.67 kt).

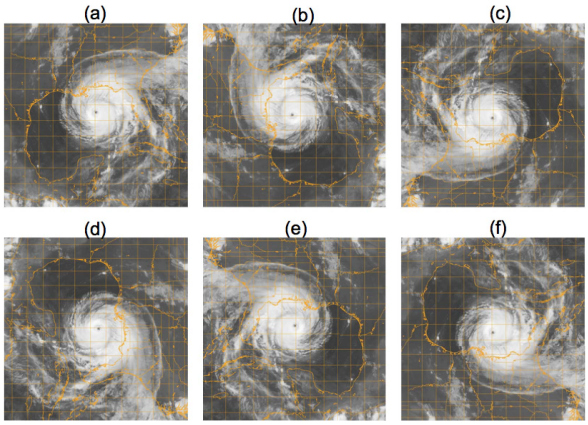


Fig. 4. Illustration of various transformations used (*hurricane IVAN from 2014-09-15*). (a) original (b) 90° rotation (c) 180° rotation (d) 270° rotation (e) horizontal flip, and (f) vertical flip.

[2]Best track data consist of the positions and intensities during the life cycle of a tropical cyclone.

[3]Structural Similarity Index Measure is the index to measure the similarity between two images. While SSIM=1 indicates perfect similarity, SSIM=0 indicates no similarity.

TABLE IV
TRAINING, VALIDATION AND TEST DATASETS

| Hurricane Category | Train | Validation | Test | Total |
|---|---|---|---|---|
| H1 | 3314 | 1104 | 1816 | 6234 |
| H2 | 1860 | 620 | 994 | 3474 |
| H3 | 1848 | 616 | 992 | 3456 |
| H4 | 1886 | 628 | 1032 | 3546 |
| H5 | 603 | 201 | 306 | 1110 |
| NC | 126 | 42 | 54 | 222 |
| TD | 6363 | 2121 | 3576 | 12060 |
| TS | 9863 | 3288 | 5575 | 18726 |
| **Total** | **25863** | **8620** | **14345** | **48828** |

After resizing each into a 256×256 image, a 232×232 region is cropped randomly to input to our model.

### *B. Training and Testing*

We split our dataset of 48,828 images into training, test and validation sets mutually exclusive as shown in Table IV. Each transformed image is maintained in the same set as its original image. In other words, if an original image goes into the training set, its transformed images are also assigned to the training set. However, there may be a few transformed images separated from its original image due to split ratio (percentage) between these sets. However, this should not have any major impact on the training as well as overall accuracy.

For recon-only test dataset, we used recon-informed hurricane date, time and speed. We have carefully chosen instances which have correspondence in our hurdat2 test set. This helps us use an available untrained image for that instance. Moreover, this provides the intensity value somewhat independent of the Dvorak technique [36] and helps compare RMSE values of our HURDAT2 test dataset with those of recon-only dataset.

We generated a mean image of images in our training dataset. All training images were subtracted from the mean image. So basically we trained our network on the centered (0-mean) raw RGB values of pixels [22]. This makes our model more robust to the change of contrast in images.

Our network was trained on GRID K520 4GB GPU. It took around 8 hours to complete 65 epochs of training. We stopped training at around 90% validation accuracy to prevent overfitting. Using GPU of 4GB memory restricted the maximum size of networks that can be trained. Therefore, we implemented a mini-batch system for training. A single epoch of training involves running all mini-batches to cover the training dataset. We trained our model using caffe framework (in C++), which supports CUDA.

Figure 5 shows the graph of the validation accuracy, validation loss, and training loss for each training iteration. As the number of epochs increases, the model learns better. This can be observed by the gradual increase in accuracy and decrease in loss after each epoch. The slope of the accuracy curve becomes close to 0 with a high epoch number. This indicates convergence to the best model and it is a good indicator of stopping training. Stepwise learning rate ($\alpha$) is reduced by a factor of 10 in our study.

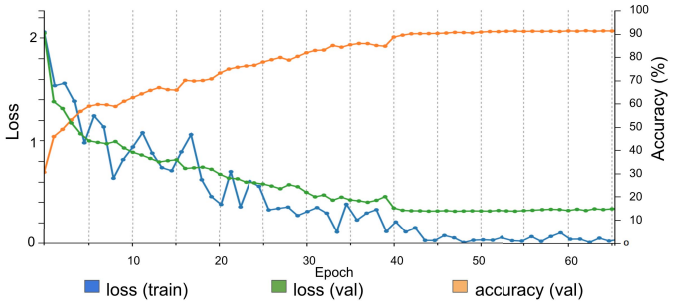


Fig. 5. Accuracy and loss plots in the training process.

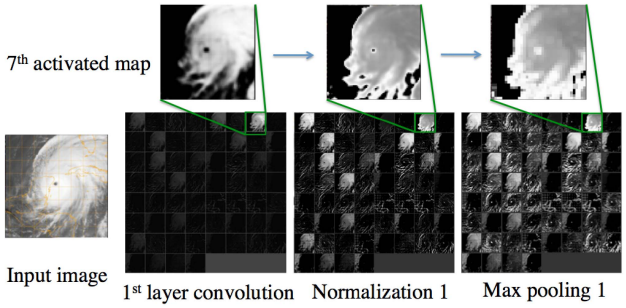


Fig. 6. Feature maps generated from first convolutional layer of our network.

The model obtained at around 90% validation accuracy was used for testing. We tested our model against the collection of images from both the Atlantic and Pacific regions. This will help us observe the generalizability of our model to classify tropical cyclones from both regions. We analyzed the top (top-1) and the second best (top-2) classification for each image in the dataset. The probabilities from softmax function are used in classification. The category is assigned the TC class with the highest probability.

### *C. Visualization*

Figure 6 displays the visualizations at the first convolution layer using *deep visualization toolbox* [24]. Input image along with the feature maps from the first convolution, normalization and pooling are shown sequentially. Each filter produces a different map. The $7^{th}$ feature map is presented by zooming. Activated images from the first convolution are easy to interpret. Visualizations for other higher layers are displayed in Figure 7. It is hard to analyze the cause of activations in those feature maps.

$113^{th}$ and $39^{th}$ feature maps generated from conv2 are shown in Figure 8. *Feature map 39* is activated with the upper curvature of hurricane structure whereas *feature map 113* is activated with the overall curved shape of the input hurricane image. This shows that each feature map learns different structures and features from the same input.

Synthetic images of activation maps generated using *deep visualization toolbox* [24] are shown in Figure 9 to visualize high activation as a result of regularized optimization. Each image corresponds to a unit representing a category in the fc8 layer. Circular motion for categories of H1, H2, and H3,

TABLE V
COMPARISON OF RESULTS ON DIFFERENT NETWORK CONFIGURATIONS. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

| Model | | | | Top-1 Accuracy | Top-2 Accuracy | f1-score | RMSE (kt) | MAE (kt) |
|---|---|---|---|---|---|---|---|---|
| ID | CNNs | FCs | Parameters | | | | | |
| M1 | 5 | 3 | C60@10, P1, C256@5, P2, C256@3, P3, C192@3, C216@3, P5, FC3072, FC1536, dropout=0.6, $\alpha$=0.001, $\mu$=0.9 | 77.85% | 95.04% | 0.78 | 11.05 | 7.65 |
| M2 | 5 | 3 | C56@11, P1, C216@5, P2, C256@3, P3, C192@3, P4, C216@3, P5, FC2048, FC1024, dropout=0.6, $\alpha$=0.001, $\mu$=0.9 | 76.58% | 94.29% | 0.77 | 11.05 | 7.88 |
| M3 | 5 | 3 | C60@10, P1, C256@5, P2, C256@3, P3, C192@3, C216@3, FC3072, FC1536, dropout=0.5, $\alpha$=0.001, $\mu$=0.95 | 78.45% | 95.10% | 0.78 | 10.87 | 7.50 |
| M4 | 6 | 3 | C72@12, P1, C216@5, P2, C216@3, C256@3, C256@3, C216@2, P6, FC4096, FC3072, dropout=0.5, $\alpha$=0.001, $\mu$=0.9 | 69.28% | 90.76% | 0.69 | 13.38 | 9.20 |
| M5 | 5 | 3 | C64@10, P1, C256@5, P2, C288@3, P3, C272@3, C256@3, P5, FC3584, FC2048, dropout=0.5, $\alpha$=0.001, $\mu$=0.9 | **80.66%** | **95.47%** | **0.80** | **10.18** | **7.28** |

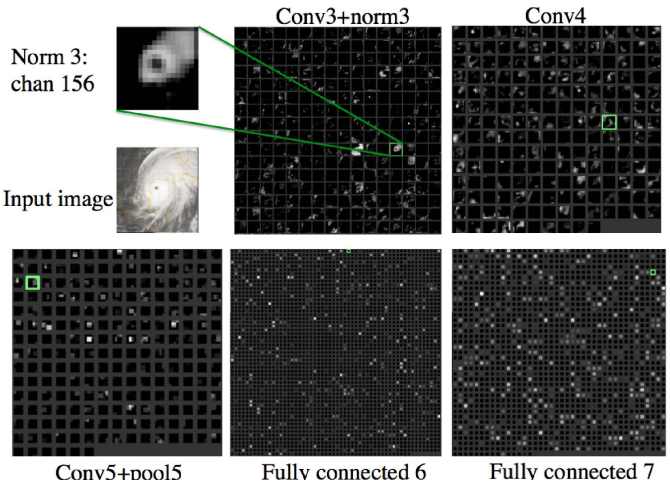


Fig. 7. Visualization for layers from convolution 3 to fully connected 7.

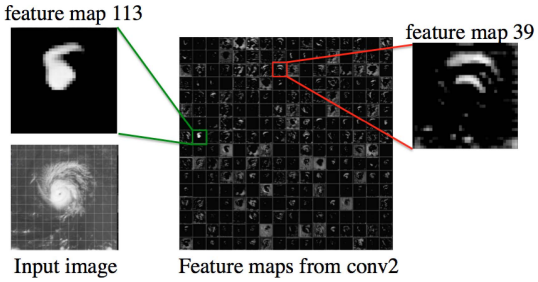


Fig. 8. Feature maps from second convolutional layer.

and random structure for NC, TD, and TS can be observed. Synthetic H4 and H5 images have smooth texture with the prominent eye of hurricane located nearly at the center.

### D. Performance Analysis

In this section, experimental results are presented, and the performance of our model is compared with the performance of previous techniques for intensity estimation.

*1) Models:* Once there is a somewhat acceptable architecture, its hyperparameters could be adjusted based on the validation dataset. In our case, we have not always got consistent response while trying our architecture with different hyperparameters.

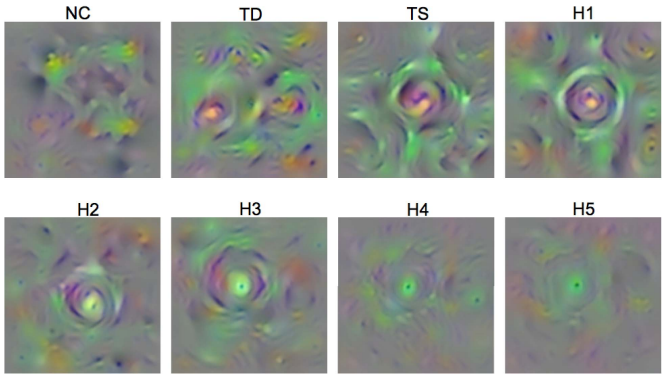


Fig. 9. Synthetic images of 8 units of fc8 layer. Each unit corresponds to single class label.

For example, consider models M1 and M3 in Table V. Despite missing the fifth pooling layer (P5) that M1 had, the accuracy of M3 is slightly higher than the accuracy of M1. Compared to M1, M2 had the additional fourth pooling layer (P4). The number of kernels used in the first and second convolutional layer is slightly less in M2. Despite using P4 in M2, its accuracy is even lesser than the accuracy of M1. The pooling layer P4 did not improve the performance of M4. If models M3 and M4 are compared, M4 has one more convolutional layer, while the number of initial kernels in M4 is more than that of M3, the others are slightly different. After the second pooling layer (P2), M4 has only pooling layer after the sixth convolutional layer (P6). This generated the lowest accuracy among models in Table V. The number of parameters learned in M4 is significantly higher compared to M3. Model M5 is a type of fusion of other models. Its number of kernels for the first convolutional layer is less than that of M4 (with the lowest accuracy) but higher than others. It has the additional fifth pooling layer (P5) that M3 (the second highest accuracy) did not have. Its parameters for fully connected layers is less than M4 but higher than others. Model M5 has generated the best results in our experiments.

The model used for the rest was the final snapshot taken at the end of $56,095^{th}$ iteration (equivalent to 65 epochs) that obtained around 90% validation accuracy for the best model. 4GB RAM and 4GB GPU memory were enough to train our network in about eight hours. After classifying 14,345 hurdat2 and 2,646 recon-only test images, we analyzed the performance of our model.

TABLE VI
CONFUSION MATRIX

| | | hurdat2 | | | | | | | | | Recon | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **NC** | **TD** | **TS** | **H1** | **H2** | **H3** | **H4** | **H5** | **Total** | **NC** | **TD** | **TS** | **H1** | **H2** | **H3** | **H4** | **H5** | **Total** |
| **Actual Category** | **NC** | 32 | 20 | 2 | 0 | 0 | 0 | 0 | 0 | **54** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **0** |
| | **TD** | 9 | 3174 | 393 | 0 | 0 | 0 | 0 | 0 | **3576** | 0 | 71 | 31 | 0 | 0 | 0 | 0 | 0 | **102** |
| | **TS** | 1 | 488 | 4838 | 208 | 25 | 10 | 3 | 2 | **5575** | 0 | 87 | 849 | 44 | 3 | 1 | 0 | 0 | **984** |
| | **H1** | 0 | 16 | 423 | 1235 | 115 | 20 | 7 | 0 | **1816** | 0 | 1 | 38 | 322 | 36 | 7 | 4 | 0 | **408** |
| | **H2** | 0 | 0 | 70 | 193 | 614 | 98 | 19 | 0 | **994** | 0 | 0 | 13 | 54 | 181 | 39 | 13 | 0 | **300** |
| | **H3** | 0 | 0 | 35 | 37 | 156 | 657 | 106 | 1 | **992** | 0 | 0 | 15 | 27 | 32 | 195 | 48 | 1 | **318** |
| | **H4** | 0 | 0 | 14 | 4 | 24 | 117 | 816 | 57 | **1032** | 0 | 0 | 7 | 3 | 8 | 25 | 319 | 28 | **390** |
| | **H5** | 0 | 0 | 0 | 0 | 1 | 14 | 86 | 205 | **306** | 0 | 0 | 0 | 0 | 1 | 3 | 42 | 98 | **144** |
| | **Total** | **42** | **3698** | **5775** | **1677** | **935** | **916** | **1037** | **265** | **14345** | **0** | **159** | **953** | **450** | **261** | **270** | **426** | **127** | **2646** |

TABLE VII
CLASSIFICATION REPORT

| | hurdat2 | | | | Recon | | | |
|---|---|---|---|---|---|---|---|---|
| **Cat** | **P** | **R** | **F1** | **Total** | **P** | **R** | **F1** | **Total** |
| NC | 0.76 | 0.59 | 0.67 | 54 | - | - | - | - |
| TD | 0.86 | 0.89 | 0.87 | 3576 | 0.45 | 0.70 | 0.54 | 102 |
| TS | 0.84 | 0.87 | 0.85 | 5575 | 0.89 | 0.86 | 0.88 | 984 |
| H1 | 0.74 | 0.68 | 0.71 | 1816 | 0.72 | 0.79 | 0.75 | 408 |
| H2 | 0.66 | 0.62 | 0.64 | 994 | 0.69 | 0.60 | 0.65 | 300 |
| H3 | 0.72 | 0.66 | 0.69 | 992 | 0.72 | 0.61 | 0.66 | 318 |
| H4 | 0.79 | 0.79 | 0.79 | 1032 | 0.75 | 0.82 | 0.78 | 390 |
| H5 | 0.77 | 0.67 | 0.72 | 306 | 0.77 | 0.68 | 0.72 | 144 |
| **avg** | **0.80** | **0.81** | **0.80** | **14345** | **0.78** | **0.77** | **0.77** | **2646** |

TABLE VIII
TOP-1 AND TOP-2 HITS

| **Category** | **Total** | **Top-1** | **$2^{nd}$ hit** | **Top-2** |
|---|---|---|---|---|
| NC | 54 | 32 | 15 | 47 |
| TD | 3576 | 3174 | 364 | 3538 |
| TS | 5575 | 4838 | 665 | 5503 |
| H1 | 1816 | 1235 | 432 | 1667 |
| H2 | 994 | 614 | 215 | 829 |
| H3 | 992 | 657 | 212 | 869 |
| H4 | 1032 | 816 | 148 | 964 |
| H5 | 306 | 205 | 73 | 278 |
| **Total** | **14345** | **11571** | **2124** | **13695** |

TABLE IX
COMPARISON OF RMSE VALUES IN KT WITH RESULTS FROM DAVT TECHNIQUES [12] AND [16]

| **Cat** | **Hurdat2** | **Recon** | **Western North Pacific [12]** | **Eastern North Pacific [12]** | **North Atlantic** [16] |
|---|---|---|---|---|---|
| NC | 9.41 | - | - | - | - |
| TD | 6.52 | 9.61 | 11.0 | 10.0 | - |
| TS | 9.81 | 9.18 | 13.8 | 11.5 | 11.0 |
| H1 | 11.69 | 9.94 | 19.5 | 16.6 | 12.5 |
| H2 | 12.50 | 11.62 | | | 12.5 |
| H3 | 13.78 | 15.97 | 19.5 | 26.1 | 12.6 |
| H4 | 11.88 | 13.21 | | | 17.7 |
| H5 | 13.45 | 11.47 | | | 32.4 |
| **Avg** | **10.18** | **11.36** | | | |

*2) Confusion Matrix:* We present the confusion matrix of hurricane classifications for both hurdat2 and recon-only datasets in Table VI. The number of correctly classified images for any category are the numbers along the diagonal.

In Table VI, we can see that a hurricane is more likely to be misclassified to a closer class (low intensity difference) rather than to a far class (with high intensity difference). Some near misses (*i.e.*, misclassified by a single class) are observed. For example, TD is mostly misclassified as TS, H1 as TS, H5 as H4, and so on. So, our model is mainly confused with cyclones of similar intensity.

*3) Classification Performance:* Precision, recall (probability of detection), and f1-score are used to evaluate the performance of our model. Precision(P) is the ratio of the number of true positive class values to the total number of positive classifications. Recall(R) is the ratio of the number of true positive classifications to the the number of positive class values in the test data. F1-score(F1) is the harmonic mean of recall and precision used to measure tests accuracy. Table VII presents the precision, recall, and f1-score to be around 0.8. This shows that our model is robust and is not biased towards recall or precision. For recon-only dataset, the values of these measures are around 0.73.

We also analyze the performance of our model by providing top-1 and top-2 accuracies. Table VIII presents whether the corresponding category is classified as the top-class or the second-class. *Exact-hit* is the correct classification of hurricane with the highest confidence. *2nd-hit* is the correct classification with the second highest confidence. Top-1 measure is the number of *exact-hit*s whereas top-2 is the sum of *exact-hit*s and *2nd-hit*s. For example, if any image with intensity H1 has classification probabilities as $H1 : 73.3\%$, $H2 : 15.5\%$, $TD : 5.1\%$, then it is counted in both top-1 and top-2 (*exact-hit*). However, if H2 image has the same classification probabilities then it contributes to the top-2 accuracy (*2nd-hit*).

For hurdat2 dataset the top-1 accuracy obtained is around 80.66%. The top-2 accuracy is 95.47% and suggests that if misclassification between neighboring (confusing) classes is reduced, very high accuracy can be obtained. Similarly, the accuracies of top-1 and top-2 for recon-only dataset are around 76.91% and 92.55% respectively.

*4) RMS Intensity Errors:* Table IX provides root-mean-square intensity error (RMSE) values measured in knots. For categories TD through H4, the estimated speed is determined as the weighted average of two highest categories with respect to their probabilities. Otherwise, the mean speed of the category that has the highest probability is used. We also limit

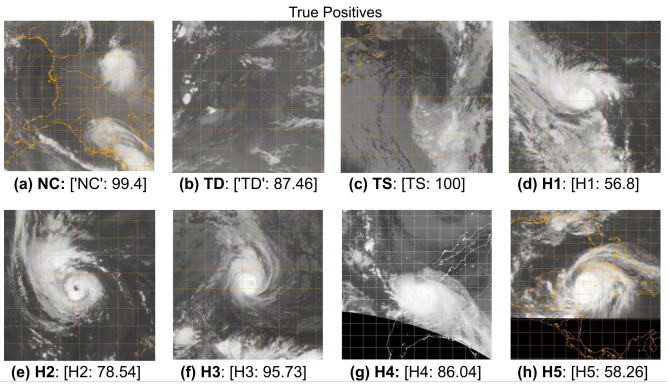


Fig. 10. Correct classification with highest confidence.

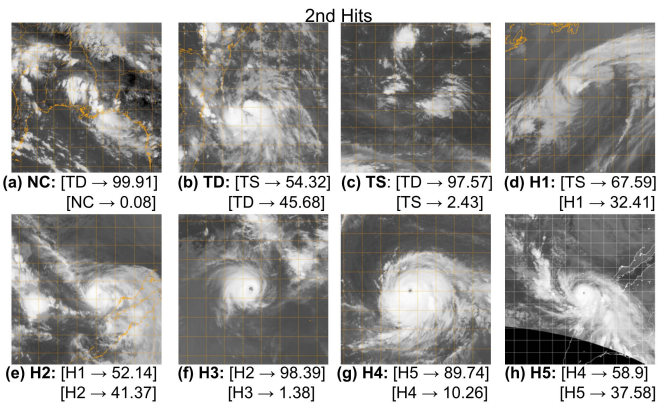


Fig. 11. Images with 2nd-hits and their confidence.

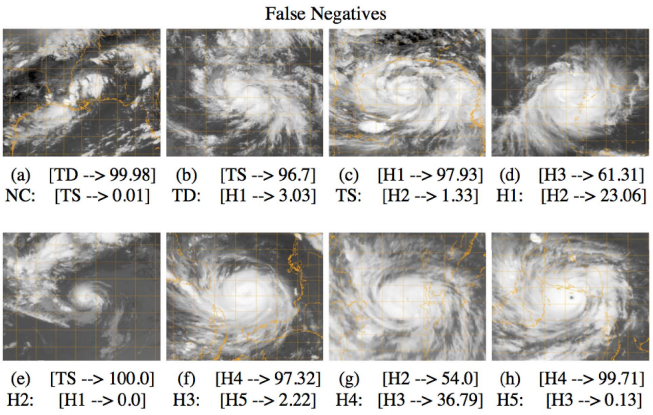


Fig. 12. Misclassifications.

the weighted average of speed within the speed interval of the category. The difference between the estimated speed and the actual speed is used for RMSE calculation.

The root-mean-square errors (RMSEs) of hurdat2 dataset and recon-only dataset are *10.18 kt* and *11.36 kt*, respectively, for the Atlantic and Pacific regions. Root-mean-square error (RMSE) value obtained by Pineros *et al.* (2011) for estimating the tropical cyclone intensity for the North Atlantic Basin was *14.7 kt* [18]. This was further improved by Ritchie *et al.* (2012) using improved deviation angle technique [16] to *12.9 kt*. In the North Pacific, using the same deviation angle technique, Ritchie *et al.* [12] achieved the RMSE of *14.3 kt*. They also presented RMSE for different categories such as the tropical depression, tropical storm, etc. Our results indicate significant improvements with respect to previous techniques [12], [16] for both the Atlantic and Pacific regions. Not only overall RMSE values improved, but our RMSE values are also better for each category. Table IX shows the comparison of our method in terms of RMSE (kt) with recent DAVT techniques [12], [16] across the Atlantic and Pacific regions per category. This table also validates the improvements of our method.

## V. Discussion

### A. Misclassifications

Correctly classified hurricane images for all categories are displayed in Figure 10. For example, in Figure 10 (h), H5 hurricane image is correctly classified H5 with 58.26% confidence. Figure 11 provides the images having the second highest confidence with the correct label (*2nd-hits*). For example, H2 hurricane in Figure 11 (e) is classified as H2 with the second-highest confidence (41.37%).

Furthermore, Figure 12 shows misclassifications for each category. These images do not fall under either top-1 or top-2 hits. Our model was unable to provide good classifications for these images. Reaching conclusions based on these sample images might be misleading. These results should be analyzed along with Table VI that shows misclassifications into various categories. It is always possible to misclassify with a close category. The most important problem is why images are classified into distant categories. For example, 14 H4 category images are classified as TS. We do not use any temporal information about categories, and the use of temporal information could improve the accuracy. For example, we did not use the classification of a storm in previous 6 hours for the analysis. Use of such information would be helpful. Adjusting deep CNN based on misclassifications is very challenging as it is hard to interpret its model.

### B. Limitations

*1) Quality of Dataset:* The main problem with training was the quality of dataset. Images contain colored latitude-longitude grids and coastlines. These grids and coastlines in images act as noise and may complicate training. In addition, the color of grids is not uniform throughout the dataset. Hurricane *MITCH* in 1998 contains red grids whereas others have yellow grids. This mismatch in color can affect the training of the model. Additionally, we have lots of images containing black patches (*e.g.*, Figure 10 (g) and 10 (h)). These patches do not carry any information about the hurricane and may complicate the learning process.

*2) Size of Dataset:* The size of a dataset is a major concern in any deep learning technique. We need a very large training dataset to avoid overfitting and for better generalizability of the model. We did interpolations and various transformations to reach up to 25,863 images (Table IV) for training, but this number is still not high for a deep learning process. It was hard to collect a large number of images because best track data (hurdat2) had information about images for every 6 hours.

This massively restricted the number of images that can be collected each day. There is also a substantial difference in the number of images for each category. Hurricanes with the intensity TS and TD are more likely to occur than hurricanes of H4 and H5 leading to the skewness in the dataset. These biases in the dataset may affect the accuracy of classification.

*3) Hyperparameters:* The performance of the model depends on the architecture of the network. Parameters like the number of convolutional layers, pooling layers, fully connected layers, and the number of filters (kernels) used in each layer affect training as well as testing. Due to the limitation of hardware resources, we have used 5 convolutional layers and 3 fully connected layers. Various minor adjustments to the learning rate, the size of filters, stride, padding, *etc.* may improve the accuracy. Refinement in regularization and normalization are yet to be tested with different combinations in this network. Hyperparameters from previous CNN techniques [22], [25] are taken into consideration while building our network. Future work should test with a different number of layers and parameters. However, our current results are very promising.

## VI. Conclusion

In this paper, we presented a reliable and robust technique for estimating the intensity of tropical cyclones using a deep convolutional neural network. Deep network with multiple convolutional and fully connected layers using regularization techniques make the complex feature extraction task from hurricane images effective. Estimating the intensity category of new hurricane sample can be done in seconds with very little human effort. Our model shows significant improvement in both RMSE values and generalizability.

There are various small tasks that can improve the accuracy and lower the RMSE intensity value in our method. First, the colored grids and coastlines could be removed from the training and test sets. It is also possible to use images without grids and coastlines. In addition to that, removing black patched images from our dataset might increase the overall accuracy of our findings. It is always a good idea to use a large training set for deep learning. Brightness and contrast could be used in augmenting the dataset. Finally, as a deep convolutional neural network is governed by several hyperparameters, increasing the number of convolutional and fully connected layers, tweaking the parameters like regularization and learning rate for further optimization might improve the accuracy.

## Acknowledgment

The authors would like to thank the Naval Research Laboratory for images and National Hurricane Center for HURDAT data. They also thank the Information Technology and Systems Center (ITSC) at the University of Alabama in Huntsville for supporting and providing resources for this research.

## References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[2] C. E. Perez. (2016). *Design Patterns for Deep Learning Architecture*. [Online]. Available: http://www.deeplearningpatterns.com/doku.php

[3] J. Schmidhuber, "Deep learning," *Scholarpedia*, vol. 10, no. 11, p. 32832, 2015.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[5] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[6] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[7] H. A. Song and S.-Y. Lee, "Hierarchical representation using NMF," in *Proc. 20th ICONIP*, Daegu, South Korea, 2013, pp. 466–473.

[8] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time-series," in *The Handbook of Brain Theory and Neural Networks*, M. A. Arbib, Ed. Cambridge, MA, USA: MIT Press, 1998.

[9] LISA Lab. *Convolutional Neural Networks (LeNet)—DeepLearning 0.1*. Accessed: Feb. 2016. [Online]. Available: http://deeplearning.net/tutorial/lenet.html

[10] V. F. Dvorak, "Tropical cyclone intensity analysis and forecasting from satellite imagery," *Monthly Weather Rev.*, vol. 103, pp. 420–430, May 1975.

[11] C. Velden *et al.*, "The Dvorak tropical cyclone intensity estimation technique: A satellite-based method that has endured for over 30 years," *Bull. Amer. Meteorol. Soc.*, vol. 87, no. 9, pp. 1195–1210, 2006.

[12] E. A. Ritchie, K. M. Wood, O. G. Rodriguez-Herrera, M. F. Pineros, and J. S. Tyo, "Satellite-derived tropical cyclone intensity in the north pacific ocean using the deviation-angle variance technique," vol. 29, no. 3, pp. 505–516, Jun. 2014.

[13] V. F. Dvorak. (Feb. 1973). *A Technique for Analysis and Forecasting of Tropical Cyclones Intensities From Satellite Pictures*. [Online]. Available: https://commons.wikimedia.org/wiki/File:DvorakCDP1973.png

[14] NOAA Satellite and Information Service. *Dvorak Current Intensity Chart*. Accessed: Aug. 2016. [Online]. Available: http://www.ssd.noaa.gov/PS/TROP/CI-chart.html

[15] C. Velden, T. Olander, D. Herndon, and J. P. Kossin, "Reprocessing the most intense historical tropical cyclones in the satellite era using the advanced Dvorak technique," *Monthly Weather Rev.*, vol. 145, no. 3, pp. 971–983, 2017. [Online]. Available: https://doi.org/10.1175/MWR-D-16-0312.1

[16] E. A. Ritchie, G. Valliere-Kelley, M. F. Pineros, and J. S. Tyo, "Tropical cyclone intensity estimation in the north atlantic basin using an improved deviation angle variance technique," *Weather Forecast.*, vol. 27, pp. 1264–1277, Oct. 2012.

[17] M. F. Pineros, E. A. Ritchie, and J. S. Tyo, "Objective measures of tropical cyclone structure and intensity change from remotely sensed infrared image data," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 11, pp. 3574–3580, Nov. 2008.

[18] M. F. Piñeros, E. A. Ritchie, and J. S. Tyo, "Estimating tropical cyclone intensity from infrared image data," *Weather Forecast.*, vol. 26, pp. 690–698, Oct. 2011.

[19] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proc. Int. Symp. Circuits Syst. (ISCAS)*, 2010, pp. 253–256.

[20] K. Kavukcuoglu, P. Sermanet, Y. Boureau, K. Gregor, M. Mathieu, and Y. LeCun, "Learning convolutional feature hierarchies for visual recognition," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 23. 2010, pp. 1090–1098.

[21] C. Szegedy *et al.*, "Going deeper with convolutions," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, 2015, pp. 1–9.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran & Associates Inc., 2012, pp. 1097–1105.

[23] D. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance concolutional neural networks for image classification," in *Proc. 20th Int. Joint Conf. Artif. Intell. (IJCAI)*, vol. 2. Barcelona, Spain, 2011.

[24] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. (2015). "Understanding neural networks through deep visualization." [Online]. Available: https://arxiv.org/abs/1506.06579

[25] Y. Jia *et al.* (2014). "Caffe: Convolutional architecture for fast feature embedding." [Online]. Available: https://arxiv.org/abs/1408.5093

[26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[27] Y. LeCun, B. Boser, J. S. Denker, and D. Henderson, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.
[28] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision*, D. Fleet, T. Pajdla, B. Schiele, and T. Schiele, Eds. Cham, Switzerland: Springer, 2014, pp. 818–833. [Online]. Available: https://doi.org/10.1007/978-3-319-10590-1_53.
[29] E. Mizutani, S. E. Dreyfus, and K. Nishio, "On derivation of MLP backpropagation from the Kelley–Bryson optimal-control gradient formula and its application," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Como, Italy, Jul. 2000, pp. 167–172.
[30] H. J. Kelley, "Gradient theory of optimal flight paths," *ARS J.*, vol. 30, no. 10, pp. 947–954, 1960.
[31] K. Chellapilla, S. Puri, and P. Simard, "High performance convolutional neural networks for document processing," in *Proc. 10th Int. Workshop Frontiers Handwriting Recognit.*, La Baule, France, 2006.
[32] Y. Jia and E. Shelhamer. *Local Response Normalization (LRN)*. Accessed: Feb. 2016. [Online]. Available: http://caffe.berkeleyvision.org/tutorial/layers.html
[33] A. Krizhevsky. *cuda-convnet*. [Online]. Available: https://code.google.com/p/cuda-convnet/wiki/LayerParams
[34] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade*, vol. 7700, G. Montavon, G. B. Orr, and K. R. Müller, Eds. Berlin, Germany: Springer, Jan. 2012, pp. 421–436.
[35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, Jun. 2014.
[36] J. A. Knaff, D. P. Brown, J. Courtney, and G. M. Gallina, "An evaluation of Dvorak technique–based tropical cyclone intensity estimates," *Weather Forecast.*, vol. 25, no. 5, pp. 1362–1379, Oct. 2010.

**Ritesh Pradhan** received the B.E. degree in computer engineering from the Institute of Engineering, Tribhuvan University, Nepal, and the M.S. degree in computer science from The University of Alabama, Huntsville. He is currently a Senior Software Engineer with the Innovation and Custom Engineering-Data Science Team, FireEye Inc.

**Ramazan S. Aygun** received the B.S. degree in computer science/engineering from Bilkent University in 1996, the M.S. degree in computer engineering from Middle East Technical University, Turkey, in 1998, and the Ph.D. degree in computer Science and engineering from University at Buffalo, The State University of New York, in 2003. He is currently an Associate Professor with the Computer Science Department, The University of Alabama in Huntsville.

He has authored or presented around 90 refereed international journal and conference papers in the areas of spatio-temporal modeling and retrieval, video panorama generation, computer vision, video and image processing, multimedia information retrieval, multimedia systems, data analytics for protein crystallization, and bioinformatics.

Dr. Aygun served as the Program Co-Chair of a leading conference in multimedia systems (the IEEE International Symposium on Multimedia 2012), a Guest Editor for a special issue of *International Journal of Semantic Computing*, an Editorial Review Board of the *International Journal of Multimedia Data Engineering and Management*, and an Organizer for workshop on video panorama.

He has served as the Demo Co-Chair for the IEEE ISM (2006, 2008, 2011, and 2013–2017), the IEEE Semantic Computing 2008, and the IEEE Big Multimedia 2015; the Publicity Chair for the IEEE ISM 2009 and VLDB 2012; the special sessions Co-Chair for the IEEE Semantic Computing 2007; the workshop and special sessions Co-Chair for the IEEE Semantic Computing 2016, and the systems and applications technical area Co-Chair for the IEEE Semantic Computing 2008.

**Manil Maskey** is currently pursuing the Ph.D. degree in computer science with The University of Alabama in Huntsville, Huntsville, AL. He is a Research Scientist with the Data Science and Informatics Group, NASA Marshall Space Flight Center. He contributes to DSIG research activities in the area of computer vision, visualization, and data analytics.

**Rahul Ramachandran** received the Ph.D. degree in atmospheric science from The University of Alabama, Huntsville. He leads the Data Science and Informatics Group, NASA Marshall Space Flight Center, and serves as the Manager for the Global Hydrology Resource Center. His research focuses on novel application of computational methods and information technology to the acquisition, storage, processing, interchange, analysis and visualization of earth science data and information.

**Daniel J. Cecil** received the B.S. degree in meteorology from Saint Louis University in 1994, the M.S. degree in meteorology from Texas A&M University in 1997, and the Ph.D. degree in atmospheric science with Texas A&M University in 2000. He is a Research Scientist with the NASA Marshall Space Flight Center, Huntsville, AL. He has been active in the fields of thunderstorm and hurricane research. He is the Principal Investigator for the Hurricane Imaging Radiometer instrument.