UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

DETECTION OF OVERSHOOTING CLOUD TOPS WITH CONVOLUTIONAL NEURAL NETWORKS

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

BY

GOWTHAM TEJA KANNEGANTI
Norman, Oklahoma
2020

DETECTION OF OVERSHOOTING CLOUD TOPS WITH CONVOLUTIONAL NEURAL NETWORKS

A MASTER'S THESIS APPROVED FOR THE GALLOGLY COLLEGE OF ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Dean Hougen, Chair

Dr. Cameron Homeyer

Dr. Charles Nicholson

Acknowledgments

I would like to express my appreciation and gratitude to my advisor, Dr. Dean Hougen for his continuous guidance, valuable comments and feedback at various stages of this thesis. I would also thank him for believing in me and involving me in different research projects.

I would also like to thank Dr. Cameron Homeyer and Dr. Charles Nicholson for their willingness to serve on my thesis committee. I would specifically thank Dr. Cameron Homeyer for sharing his domain expertise and providing tropopause temperature data. I would also thank Dr. Philip Chilson for introducing and motivating me to this thesis topic.

I would like to specifically thank my family members for their support during critical times. Thanks to my mother and brother, Davuluri Padmavathi and Sai Teja Kanneganti for always believing in me.

I would like to acknowledge the Data Science and Analytics Directors Dr. Sridhar Radhakrishnan and Dr. Randa Shehab, faculty, and staff for their support during my graduate studies.

I would like to thank OU supercomputing center for education and research (OSCER) team for providing computation resources for this thesis. I would also thank Kristopher Bedka at NASA Langley Research Center for providing the overshooting top center data necessary for my research.

Abstract

Overshooting cloud tops can cause severe weather conditions, such as aviation turbulence, lightning, strong winds, heavy rainfall, hail, and tornadoes. Due to hazards caused by overshooting tops, several methods have been developed to detect them. Convolutional neural networks are an approach to machine learning that performs well on image-based tasks such as object detection. This study uses convolutional neural networks to detect overshooting tops in GOES-14 satellite imagery from NOAA's Geostationary Operational Environmental Satellite.

Visible and infrared images of GOES-14 satellite imagery are the primary source of input data. These images are divided into patches of size 31×31. The model takes in each patch and outputs its classification for that patch, whether it contains an overshooting top or not. The distribution of patches containing overshooting tops and those do not contain overshooting tops in the data is imbalanced. In this study, we first implement data sampling and cost-sensitive learning techniques to deal with imbalanced data. Next, we investigate approaches using tropopause temperature data to increase the performance of the model. The method we propose using tropopause temperature data in the preprocessing step performs better overall than other methods with a probability of detection 79.31%, false alarm ratio 90.94%, and critical success index 0.088. Most of the false alarms are located close to overshooting occurrence, and using a prepro-

cessing step decreases the testing time significantly. This thesis compares the performance of different imbalanced learning techniques on satellite images.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Overshooting tops (OTs) are important weather phenomenon because of the hazards associated with them. Having a system to automatically detect them would be helpful for weather forecasters. Overshooting tops have characteristics that can be observed in visible and infrared images of satellite data. Convolutional neural networks (CNNs) seem like an appropriate basis for developing a automatic overshooting top detection system. Consequently, this thesis uses CNNs for detecting OTs. This chapter gives a brief introduction to OTs, outlines the research objectives of this thesis, overviews important results, discusses the contributions of this thesis to the research community, then outlines the organization of the rest of the thesis.

1.1 Background

Overshooting tops (OTs), also called the penetrating tops, is a dome-like protrusions forming above a cumulonimbus anvil (Bedka and Khlopenkov, 2016). They form when an air parcel in a deep convective cloud protrudes its equilibrium

level near the tropopause due to momentum from a thunderstorm's updraft. In meteorology, equilibrium level is the height at which rising parcel of air is at the same temperature of its surroundings (Wikipedia contributors, 2019). The tropopause is the boundary between Earth's atmosphere between troposphere and stratosphere. Overshooting tops with strong ascending force can penetrate the tropopause and extend into the lower stratosphere. These OTs can transport gases and cloud ice into the lower stratosphere, which has a direct impact on global climate change (Homeyer and Kumjian, 2015).

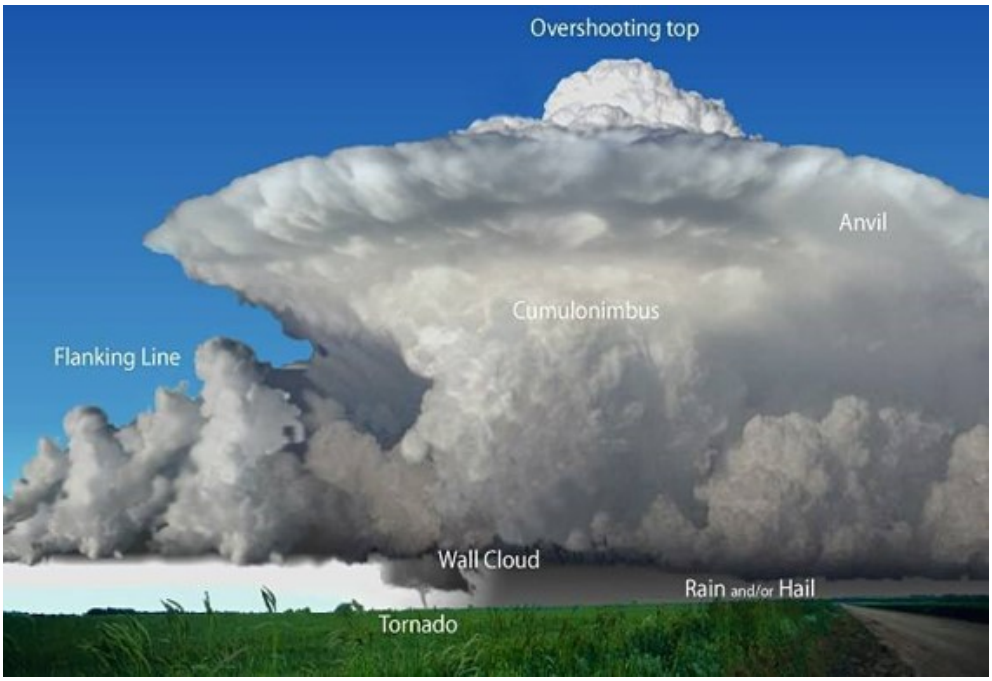


Figure 1.1: Image of an OT forming above anvil cloud (weather events, tumblr).

Thunderstorms with OTs can cause hazardous weather conditions such as heavy rainfall, large hail, and tornadoes, and these hazards are typically concentrated near OT regions (Bedka et al., 2010; Homeyer and Kumjian, 2015; Bedka and Khlopenkov, 2016; Kim et al., 2017, 2018). Detecting OTs can be useful to forecasters for warning decision making as OTs can precede severe weather events

by 30 minutes or more (Bedka and Khlopenkov, 2016). OTs can produce turbulences at vast distance as they interact and penetrate through the tropopause, which can be a significant hazard for the aircraft (Bedka et al., 2007, 2010). Because of the hazards associated with OTs, various OT detection models have been developed (Bedka et al., 2010; Bedka and Khlopenkov, 2016; Kim et al., 2017, 2018).

Satellite data is commonly used for detecting OTs. Several studies have used images of visible and infrared channels to detect OTs (Bedka and Khlopenkov, 2016; Kim et al., 2017, 2018). OTs can penetrate more than 2 km above the surrounding anvil cloud (Homeyer and Kumjian, 2015; Bedka and Khlopenkov, 2016), having a cauliflower-like structure in the visible image and producing shadows on the surrounding anvil clouds. Most OTs appear as a small cluster of cold infrared window (IRW) brightness temperatures (BTs) surrounded by a warmer anvil cloud.

1.2 Research Objectives

The specific objectives of this research are as follows:

1. To prepare GOES-14 imagery data to be input to the convolutional neural network (CNN) model using interpolation, image patching, and normalization techniques.

2. To develop a convolutional neural network model to detect the OTs in the given input data.

3. To validate the model performance on the test data.

4. To analyze different approaches to increase the performance of the model.

1.3 Overview of the Results

We also investigate six approaches to increase the performance of the CNN model on GOES-14 satellite data. This study compares the performance of these different approaches in terms of POD, FAR, and CSI. Approach 1 has a better POD of 97.29%, while Approach 2 has the least POD of 72.66%. Approach 2 has a low FAR of 90.44%, and Approach 4 has a higher FAR of 99.4%. CSI values for these approaches are within range of 0.005 to 0.092. Approach 2 has the best CSI value 0.092. However, Approach 7 has the second-best CSI value 0.088, and has more advantages. Approach 7 has better POD compared to Approach 2, and false positives generated using this approach are closer to OT occurrences compared to Approach 2.

1.4 Contributions to the Research Community

In this thesis, we looked at how convolutional neural networks approach work on the detection of overshooting cloud tops in satellite data. The first approach mentioned in this thesis is like the method mentioned in Kim et al. (2018) but is distinguished in terms of data used and data preparation steps. This research shows how imbalanced machine learning techniques, data preparation or preprocessing techniques work on satellite imagery. The results in this thesis shows that the CNN model is sensitive to the overshooting top features and is detecting overshooting top regions before and after they are detected by people.

1.5 Organization of the Thesis

This thesis is organized into seven chapters. Chapter 1 discusses the background of this study and the research objectives of the thesis. Chapter 2 reviews OT detection methods that researchers have previously developed on satellite images. Chapter 3 explains in detail the GOES-14 satellite data and tropopause temperature data used in the study, followed by data preprocessing steps and how data is calibrated as train and test data. Chapter 4 introduces neural networks, discusses the components of the CNN, the architecture of the CNN used in this study, and the modeling approach. Chapter 5 explain the approaches that are implemented in this study to detect OTs along with the results of each approach. Chapter 6 discusses the analysis made on the test data. Finally, Chapter 7 concludes the thesis and presents future work based on the results.

Chapter 2

Literature Overview

Previous studies have shown that OTs can be detected using visible channel and infrared channel imagery (Setvák et al., 2010; Bedka and Khlopenkov, 2016; Bedka et al., 2018; Kim et al., 2017, 2018). An OT can be observed in a daylight reflectance image (visible) as it protrudes above a cloud forming a dome/cauliflower-like structure and shadows by sunlight. Infrared images are used to get brightness temperatures. OTs continue to cool at a rate of $7-9$ Kkm^{-1} as they ascend into the lower stratosphere (Negri and Adler, 1981; Negri, 1982) and are usually colder than surrounding regions. So, OTs are often isolated regions of cold infrared window (IRW) brightness temperatures (BTs) relative to warmer surrounding anvil clouds. In this thesis, tropopause temperature data is also used because anvil clouds have temperatures at or near to that of tropopause temperature (Adler et al., 1985).

Previous studies have proposed various OT detection methods using visible and/or infrared images. Infrared images are widely used because infrared images can be utilized irrespective of procuring time, while visible images are only available during daytime. Dual Channel Difference (Setvák et al., 2007) and InfraRed

Window texture (IRW-texture) (Bedka et al., 2010) are the two most widely used methods to detect OTs with infrared images. The Dual Channel Difference or Water Vapor-InfraRed Window channel Brightness Temperature Difference (WV-IRW BTD) approach uses the brightness temperature difference between water vapor and window channels to detect OTs. However, the threshold used in this method varies with the characteristics of satellite data used.

The IRW-texture algorithm was developed to identify OTs as characteristics of groups of low IRW BT pixels within surrounding anvil clouds. However, the thresholds used in this approach are not sufficient to cover all the characteristics of OTs. This method have a POD 35.1% and FAR 24.9%. Bedka and Khlopenkov (2016) developed a probabilistic OT detection approach, that the gives the probability of occurrence of OTs using logistic regression. This method uses both infrared and visible image data. Although this approach has better results than above mentioned approaches this approach involves pattern recognition analysis and several rating approaches that are time consuming. This method has POD 69% and FAR 18.4%.

Two machine learning OT detection methods are developed recently using Himawari-8 satellite data. One of the methods uses multiple channels in himawari-8 satellite images and spatial texture information to extract 15 input variables to the model (Kim et al., 2017). In this approach, authors used three machine learning techniques—random forest, extremely randomized trees, and logistic regression to get the probability of OT/non-OT occurrence. This method has POD 77.06% and FAR 36.13%. The other method (Kim et al., 2018) uses a deep learning approach to detect OTs. This method uses visible and infrared image channels in Himawari-8 satellite images as input to the deep learning model. This method has POD 79.68% and FAR 9.78%.

Chapter 3

Data Processing

3.1 Overview

This chapter is divided into three sections. Section 3.2 describes the data used in this study. The data used in this study includes GOES-14 satellite images and tropopause temperature data. Section 3.3 of this chapter discusses the data preprocessing methods that are used in this study. The preprocessing steps include:

- Downscaling infrared images to 1 km resolution
- Downscaling infrared and visible images to 500 m resolution
- Normalization of both infrared and visible images
- Construction of OT center references on satellite imagery

Section 3.4 presents the data constructions steps for both training and testing. The data construction steps include:

- Dividing images into patches of size 31 $\times$ 31
- Calibrating training data for the CNN model

- Calibrating test data

3.2 Data

There are two basic types of data that are used in this thesis. They are GOES-14 imagery and tropopause temperature data. This section has two subsections. The two subsections 3.2.1 and 3.2.2 will explain those two data types in greater detail.

3.2.1 GOES-14 Imagery

Geostationary Operational Environmental Satellite (GOES)-14 imager is operated by National Oceanic and Atmospheric Administration (NOAA). The GOES-14 imager has five channels with a spatial resolution ranging from 1 km to 8 km. This study uses both visible and infrared images. Visible images of Channel 1 (wavelength $0.55 - 0.75$ um) have a spatial resolution of 1 km. The infrared image of Channel 4 (wavelength $10.20 - 11.20$ um) has a spatial resolution of 4 km. The infrared images are used to get the brightness temperature. OTs are usually colder than surrounding regions (Setvák et al., 2008, 2010). So, OTs are often isolated regions of cold IRW brightness temperatures (BTs) relative to a warmer surrounding anvil cloud. GOES-14 imagery is collected with a temporal resolution of 1 min for the infrared and visible images used in this thesis.

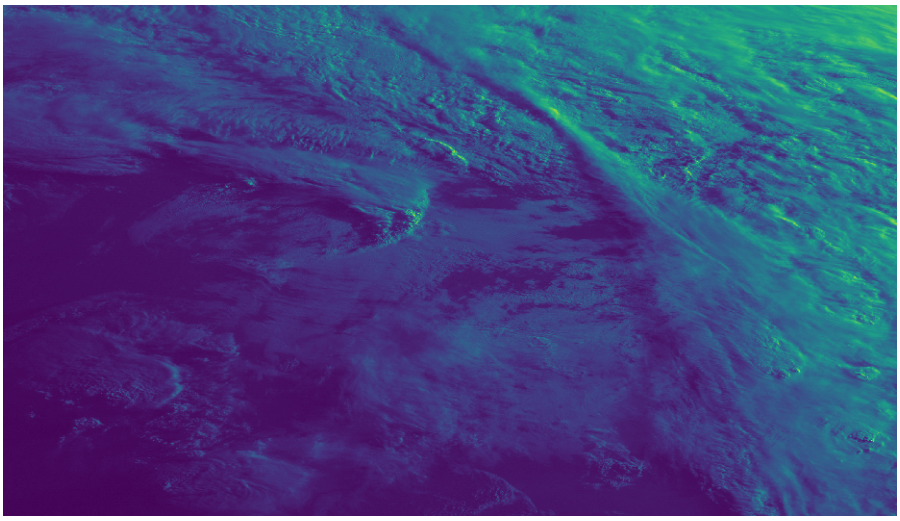


Figure 3.1: Raw visible image.

Satellite imagery used in this study is for nine hours duration from 1200–2059 UTC, and then 2245 on 25 May 2015. We collected GOES Imager data from NOAA comprehensive large array-data stewardship system (CLASS) website (NOAA CLASS). There are a total of 459 GOES–14 images in the time interval mentioned above. We divided these 459 images into training and validation sets. Each visible and infrared image is downloaded in netCDF format and contains information about the data at each pixel, latitude and longitude for each pixel, timestamp, other data (not relevant to this thesis). Visible images have 1 km geographic field of view (i.e., resolution) at nadir. This means that the data value at each pixel is for 1km. Each visible image is of the size 1387 $\times$ 2428. An example visible image can be seen in Figure 3.1. Infrared images have 4 km geographic field of view (i.e., resolution) at nadir. This means that for the infrared image, we have only one reading for 4 km, while the visible image has one reading for every 1 km. Infrared image is of size 346 $\times$ 608. An example of infrared image can be seen in Figure 3.2.

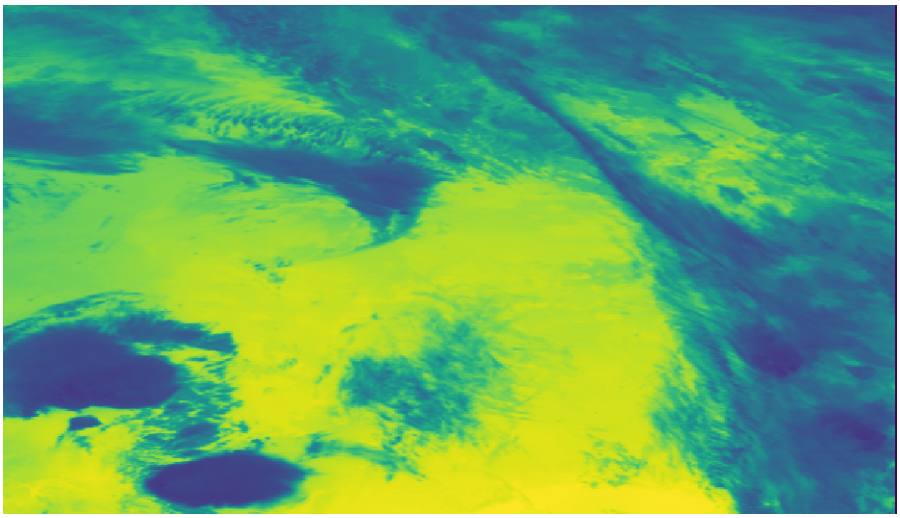


Figure 3.2: Raw infrared image.

3.2.2 Tropopause Temperature

In this study, the tropopause temperature is used to reduce the number of false positives. The Modern-Era Retrospective Analysis for Research and Applications, version 2 (MERRA-2), is the latest atmospheric reanalysis of the modern satellite era produced by NASA's Global Modeling and Assimilation Office (GMAO) (Gelaro et al., 2017). MERRA-2 reanalysis fields were used to calculate the tropopause temperature. MERRA-2 has a 0.5°× 0.625°grid spacing and provides hourly surface analyses and 3-hourly vertical profiles at 72 vertical levels. The MERRA-2 data to compute tropopause temperatures was downloaded from the Global Modeling and Assimilation Office (GMAO) (2015). The tropopause temperature data has readings for every 1° latitude and 1° longitude with a temporal resolution of 1 minute. This tropopause temperature data is used to interpolate to the coordinates of the GOES-14 imagery at 500 m resolution.

3.3 Data Preprocessing

Infrared and visible images are at different spatial resolutions. To be able to use these images together, we first downscale infrared images to 1 km resolution in the first data preprocessing step. In the later steps, we downscale both infrared and visible images to 500m resolution, normalize values in these images in the range 0–1, and the construct OT center references on satellite images. Section 3.3 discuses the data preprocessing steps mentioned above.

3.3.1 Downscaling Infrared Images to 1 km Resolution

Infrared images have a spatial resolution of 4 km, while visible images have a 1 km spatial resolution. To use both visible and infrared images together as a two-channel image, both the images should have the same resolution. To achieve this, we downscale infrared images to 1 km resolution. There are various methods to downscale the image or increase in the size of the image. After studying several image processing methods, interpolation methods seem to work well for remote sensing data. Previous studies (Han, 2013/03; Titus and Geroge, 2013) on the comparison of different interpolation techniques have shown that bicubic interpolation is the best interpolation technique for satellite images.

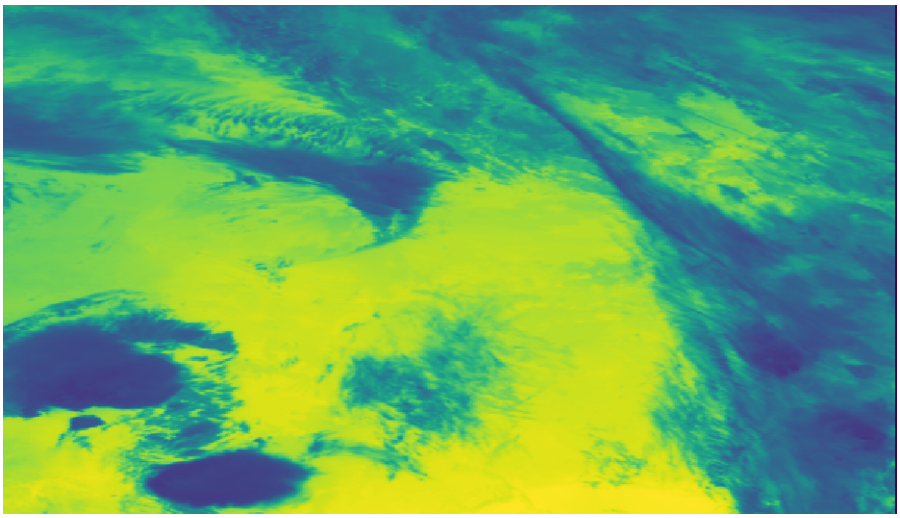


Figure 3.3: Infrared image before interpolation.

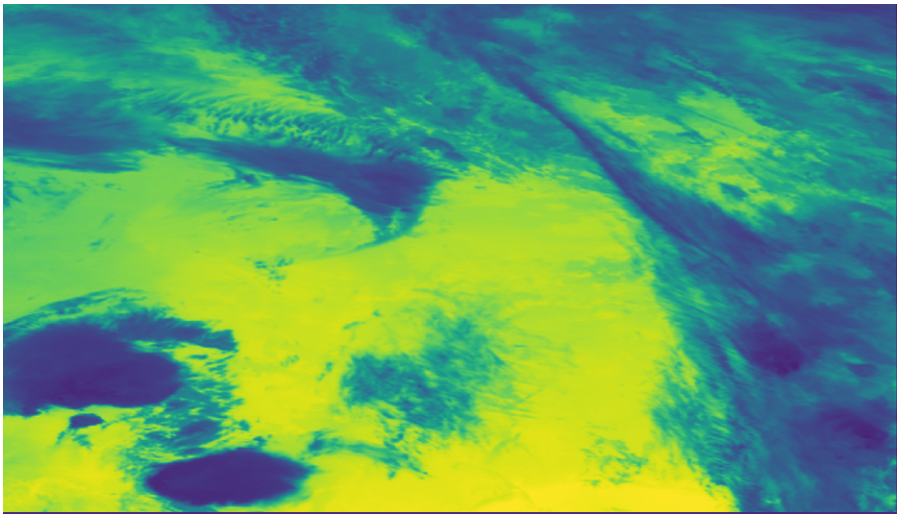


Figure 3.4: Infrared image after interpolation.

In downscaling, bicubic interpolation interpolates the value of unknown pixel p in an amplified image based on the 16 adjacent pixels (4×4) and their distance to p. In Figure 3.5, intermediate pixel $\hat{F}(p', q')$ is created by interpolating nearest 4×4 pixels from $F(p-1, q-1)$ to $F(p+2, q+2)$. The coordinates of the amplified image must be provided to get the magnified image. Latitude and longitude of the infrared image are taken as the coordinates of the original image, and the

latitude and longitude of the visible image are used as the coordinates of the interpolated image. Previous studies have shown that the bicubic interpolation technique gives a smooth, amplified image. On comparing the original Figure 3.3 and interpolated Figure 3.4 both the images looked similar. The advantage of the interpolation method other than providing a smooth image is that the values in the interpolated image and the original image are on the same scale.

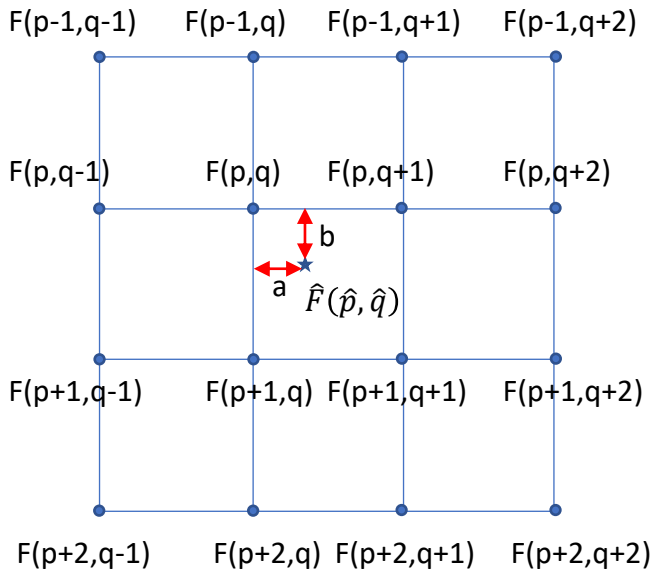


Figure 3.5: Bicubic interpolation logic (Santhosh G, 2011).

3.3.2 Downscaling Infrared and Visible Images to 500 m Resolution

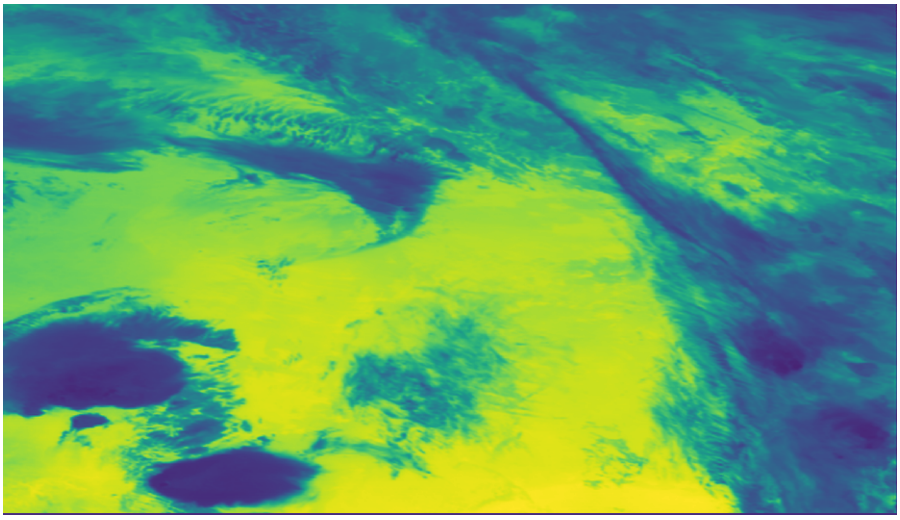


Figure 3.6: Infrared image at 1 km spatial resolution.

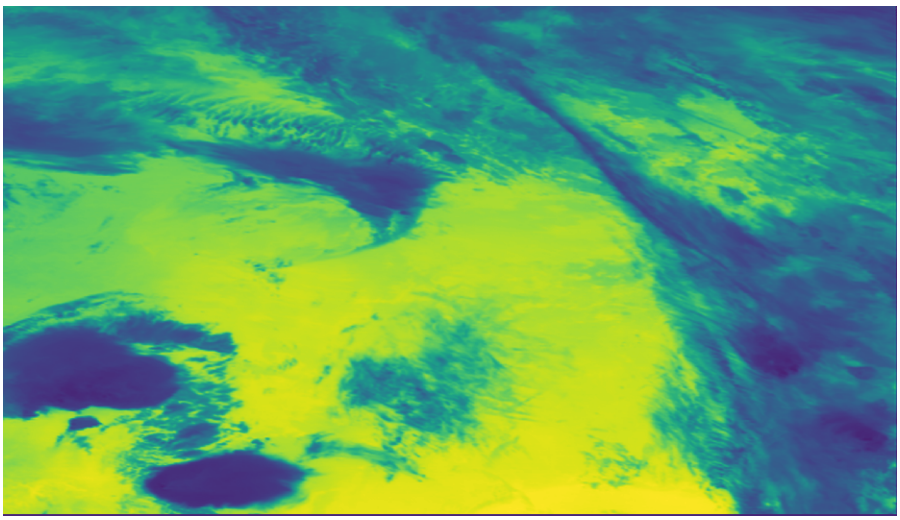


Figure 3.7: Infrared image at 500 m spatial resolution.

CNN models are widely used for image recognition tasks (LeCun et al., 2010; Sermanet et al., 2013; Girshick et al., 2014; Ren et al., 2015; Redmon et al., 2016; Krizhevsky et al., 2017; Redmon and Farhadi, 2019). Consequently, we used the CNN model for this thesis. Generally, in convolutional neural networks, as we go

deeper into layers, the size of output for the layer decreases. The input to the CNN model should be large enough so that the output layers are not very small. Infrared and visible images are downscaled to 500 m resolution to increase the size of input to the model.

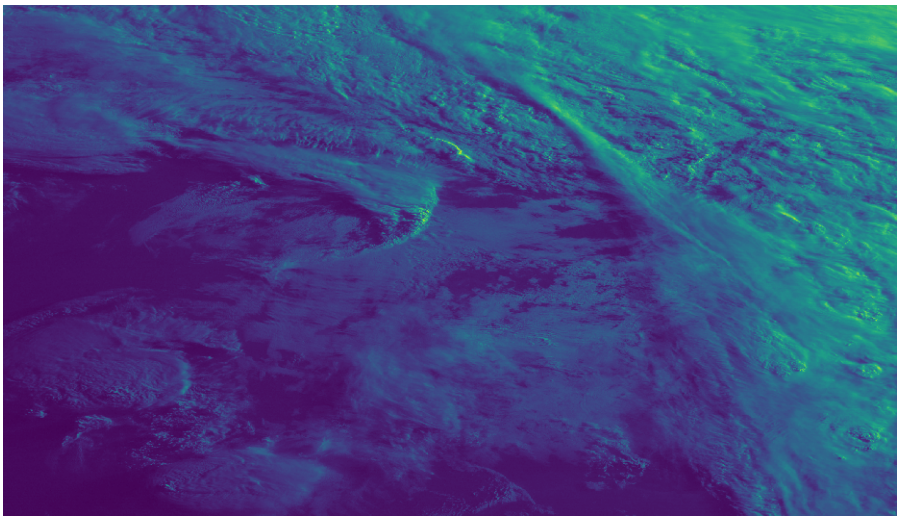


Figure 3.8: Visible image at 1 km spatial resolution.

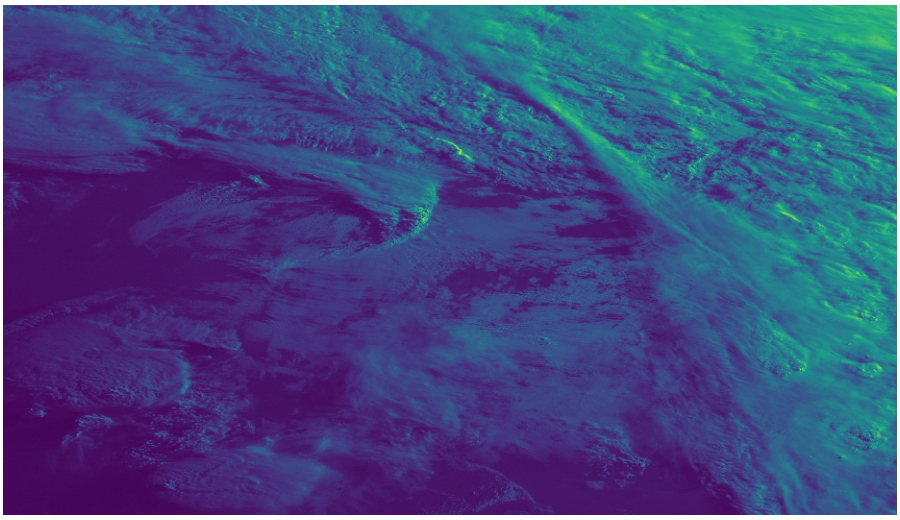


Figure 3.9: Visible image at 500 m spatial resolution.

Latitude and longitude coordinates for the 500 m resolution are calculated by taking the mean values between every pair of adjacent rows and, correspondingly

every pair of adjacent columns. Newly generated coordinates are approximately 500 m apart. Bicubic interpolation is used to generate infrared and visible images at 500 m spatial resolution. The images at 500 m resolution are of size 2773×4855.

3.3.3 Normalization of both Infrared and Visible Images

The data values for both infrared and visible images have a range in thousands. In neural networks, especially for gradient-based methods, if the input values are on different scales, model performance can be degraded as the learning speed becomes slow, and it will be difficult for the learning algorithm to converge (Ioffe and Szegedy, 2015; Kim et al., 2018). The data values of the infrared and visible images are normalized using a min-max approach. For each value at pixel X, the normalized value is $(X - X_{min})/(X_{max} - X_{min})$, where X_{min} and X_{max} are the minimum and maximum values of an input image respectively.

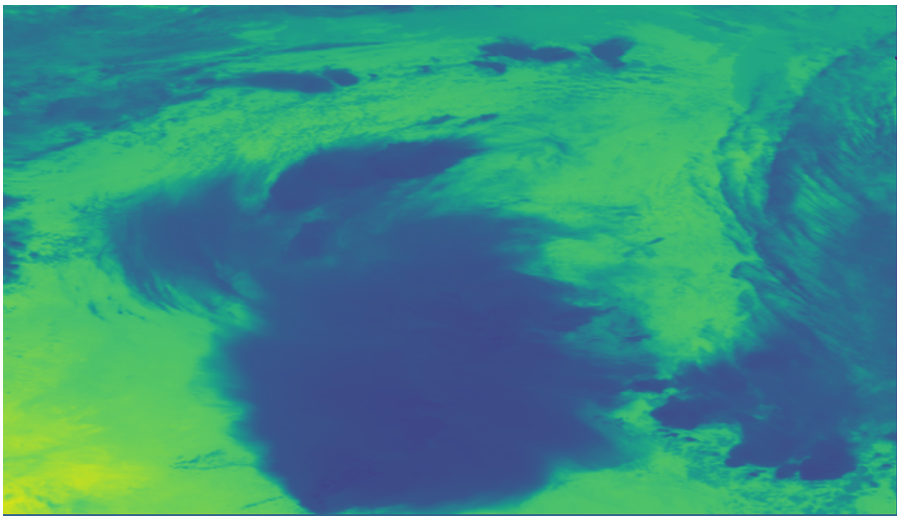


Figure 3.10: Infrared image before normalization.

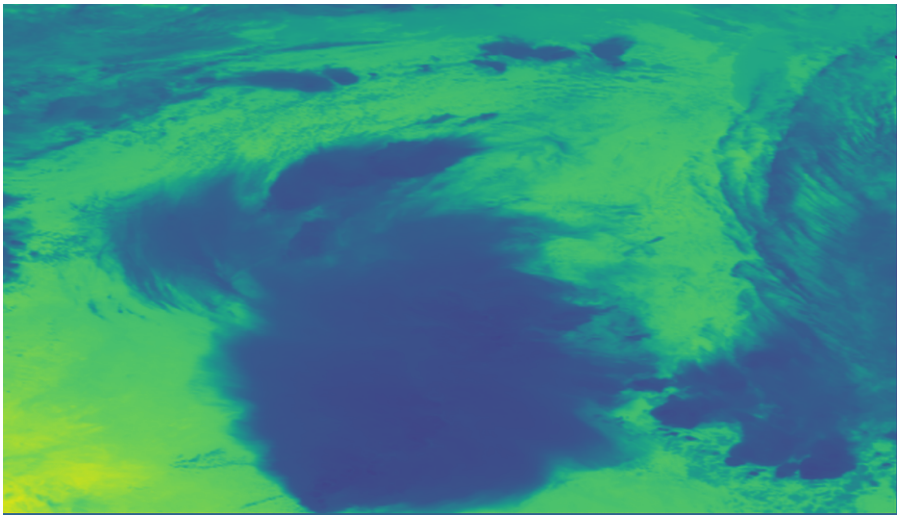


Figure 3.11: Infrared image after normalization.

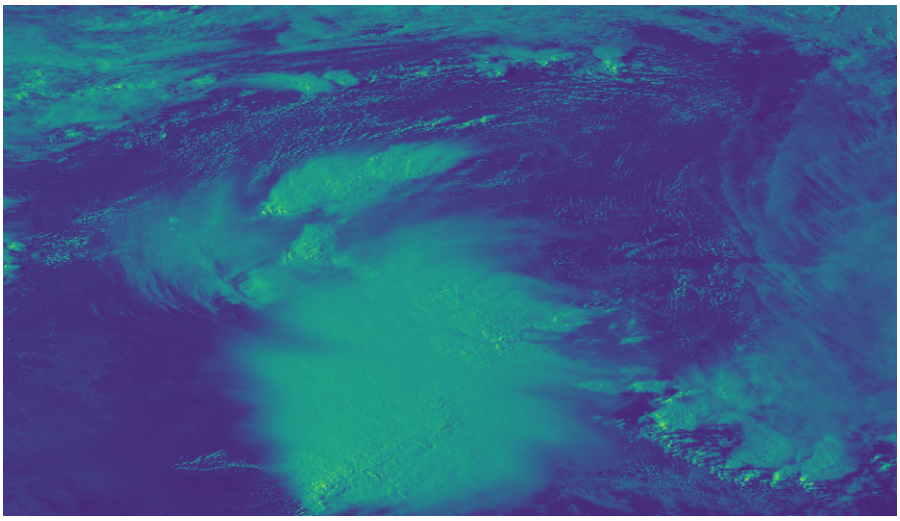


Figure 3.12: Visible image before normalization.

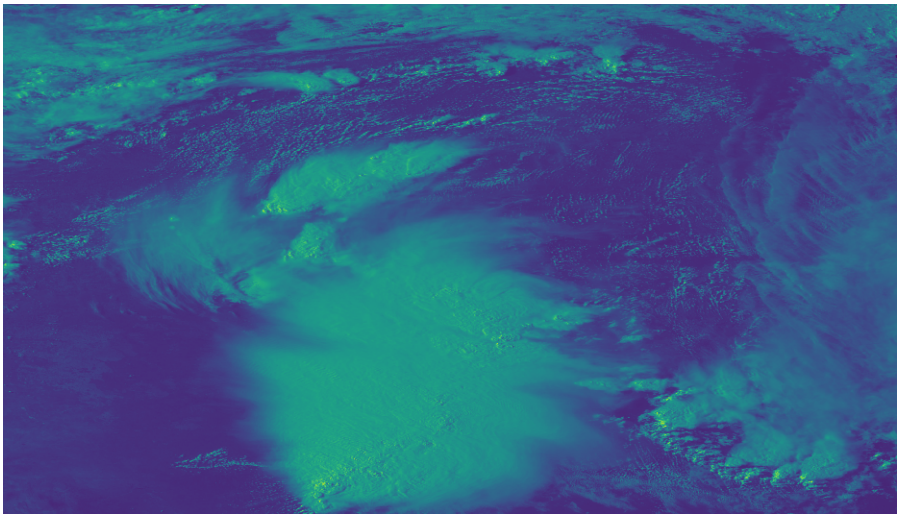


Figure 3.13: Visible image after normalization.

3.3.4 Construction of Overshooting Top Center Reference on Satellite Imagery

The OT database used for this study has the timestamp at which each OT is observed, and the co-ordinates (latitude and longitude) of the overshooting top centers occurring at that timestamp. OT centers are marked by a high school

intern at NASA and are provided for this study by Kristopher Bedka at NASA Langley Research Center. The database has information for every image analyzed from 1200-2059 UTC, and then 2245 UTC for 25 May 2015 for a long-lived mesoscale convective system (MCS) across Texas. Each OT center pixel is identified by calculating the distance between the coordinates of the OT center and the coordinates of each pixel. The pixel with the coordinates closest to the OT center is marked as the OT center. These marked pixel values are used to extract patches from the image while training and label them as OT and non-OT occurrence.

3.4 Data Construction for Training and Testing

Section 3.4 presents the steps involved in constructing the training and testing data. First, the whole image is divided into patches of size 31 $\times$ 31. Since OTs are rare events and are small in size compared to whole image, the number of patches having OT occurrences is far less than the number of patches not having OT occurrence. For training, a data augmentation technique is used to oversample OT occurring patches and randomly undersampled non-OT patches. For testing, each image is divided into patches of size 31 $\times$ 31 and tested on the model. While testing, an image patch is marked as OT occurrence only if that patch contains OT center pixel. The reasons for choosing the patch size, data augmentation techniques, and the test data construction is discussed in detail in this section.

3.4.1 Dividing Images into Patches of Size 31 × 31

After all the preprocessing steps mentioned above, both infrared and visible images are at 500 m resolution, and the size of each image is 2773 × 4855. There are several steps to prepare data for training. First, the infrared and visible images are chopped into images of size 31 × 31. Figure 3.14 shows part of an image divided into patches. The maximum size to which an OT can grow is 20 km (Bedka et al., 2012; Setvák et al., 2013; Mikuš and Mahović, 2013). We took 31 × 31 as the patch size to make sure that OTs can be fully distinguished in the patch, and the patch size is not too large for the CNN model so as to be difficult to find the localized OT regions. If the patch size is small, the distinguishable characteristics of OTs from the surrounding anvil clouds may not be captured. In infrared images, OTs are cold brightness temperature compared to the surrounding anvil clouds. In visible images, OTs can be identified as cauliflower-shape clouds forming shadows on the surrounding clouds. So, the patch size should be large enough to capture the distinguishable characteristics of OTs.

Conversely, if the patch size is large, it is difficult for the CNN model to find the localized OT regions because the features may not be clearly distinguishable in the large patch. Although OTs can grow to a maximum size of 20 km, most of the OTs have a diameter in the range of 5 km to 10 km. We used a patch size of 31 × 31 as in Kim et al. (2018) to balance between reducing the risk of misclassification and finding local OT regions. The distance between two OTs occurring together is typically more than 16 km. By using a 31 × 31 patch, we will not have two overshooting top centers in the same image patch.

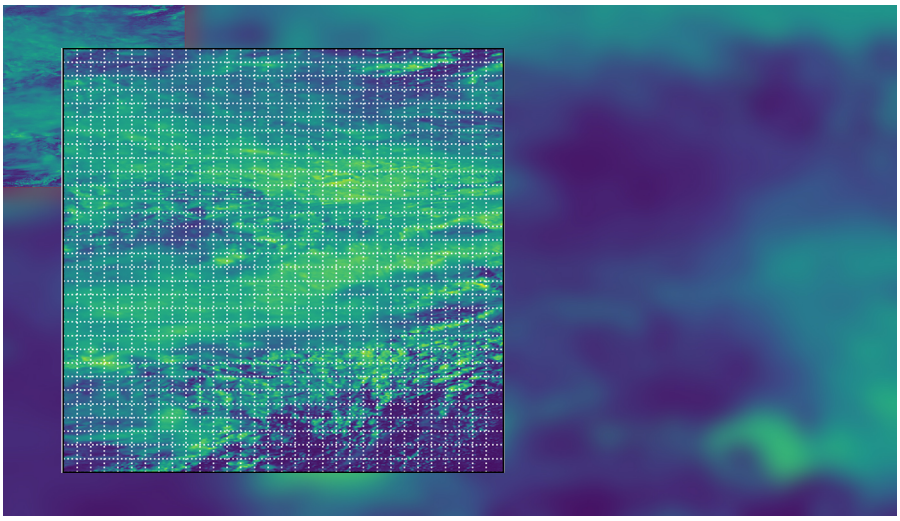


Figure 3.14: Image partially divided into 31 × 31 patches.

3.4.2 Calibrating Training Data for CNN Model

Each image is divided into 31 × 31 patches with no overlap. A full image generates 13884 patches. The total number of OTs for all the 459 images in our dataset is close to 3400. So the total number of patches for 459 images is 6372756 and the percentage of patches having OT center is 0.05%. This means that most of the patches are clear sky (do not contain OT). In machine learning models, if one of the classes is much larger than the other class, the model will be biased towards the class with a larger sample size. We used sampling methods on the training data to get a balanced distribution.

Previous studies (Yu et al., 2017; Perez and Wang, 2017; Inoue, 2018) have proposed several data augmentation techniques to overcome limited samples or create diverse data set with variants. In this thesis, we used data augmentation technique to oversample the minority class (OT occurrence) data. For training data, we extracted three different categories of patches for each OT occurrence. The first category has an OT center as the center or close to center of the 31 × 31

patch. The second category is offset images with the center of OT slightly offset from the center of the 31 × 31 patch. The third category is peripheral images with the center of OT within the 2.5 km edge of the 31 × 31 window. The schematic diagram of the 31 × 31 patch showing the OT center location in 3 different locations is shown in Figure 3.15. This data augmentation process increases the number of OT patches by three times. This method of data augmentation may also be helpful in predicting on the test data. The center of OT may not always be at the center of the patch when the whole image is chopped into 31 × 31 patches with no overlap while testing.

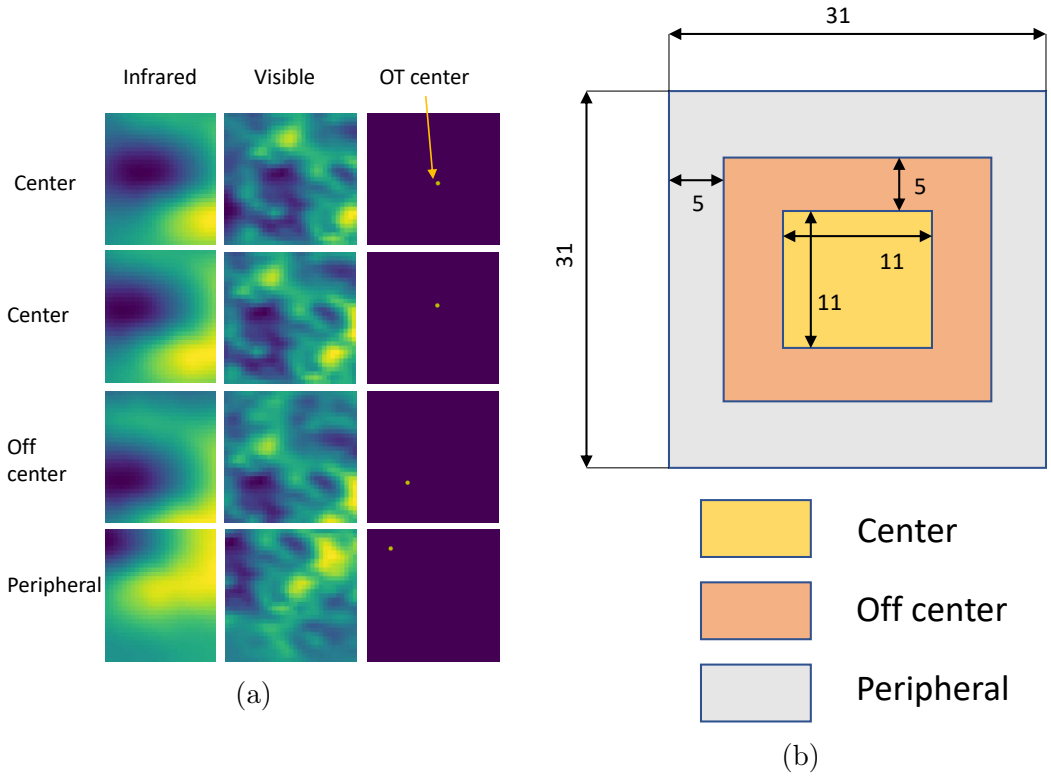


Figure 3.15: (a) Example of an image patch with OT centered at different positions. (b) OT center is placed at center, slightly off center and edge centered.

The patches having OT center at the center or offset from the center or at the peripheral are labeled as OT occurrences. Non-OT occurrence patches are

randomly generated from each image to get the same number of OT occurrence patches. We chose the center of these non-OT occurrences at random. The distance of these randomly generated OT centers are calculated and only nine patches out of eight thousand nine hundred and thirty have OT center.

3.4.3 Calibrating Test Data

Out of the 459 images, 50 images are taken for testing. Images in test data are not included in the training data. For testing, the CNN-based model is tested on the whole satellite image. The complete image is chopped into patches of size 31×31 with no overlap. We labeled each patch as "OT" or "non-OT" depending on whether the patch has an OT center or not. Then, we applied the CNN model on the patches.

Chapter 4

Convolutional Neural Networks

Artificial Neural Networks (ANNs) are composed of artificial neurons that retain the biological concept of neurons, which receive input, combine the input with their internal state, and produce output using an output function. ANNs generally consists of a set of layers, where each layer is a set of neurons. The first layer of a layered ANN is called the *input layer*, the last layer is called the *output layer*, and the layers in between are called *hidden layers*. Neurons in one layer are connected to the following and/or previous layers. Each connection has a strength, referred to as a *weights*.

The first section of this chapter discuss the components of the convolutional neural networks (CNN) which are convolutional layer, pooling layer, fully connected layer, and activation functions. The second and third sections of the chapter discuss the architecture of the CNN and the modeling approach implemented in this thesis.

4.1 Components of CNNs

This section explains how convolutional neural networks can be used to extract features from images. The main components of convolutional neural networks are the convolutional layer, pooling layer, fully connected layer, and activation function. Convolutional and pooling layers are used to extract features from the image, and fully connected layers use these features to give the output based on the training set. This section, also discusses dropout.

4.1.1 Convolutional Layer

Convolution is a mathematical way of combining two functions to produce third function. A *convolutional layer* uses a filter (kernel) to generate a convolved feature of the input image. *Stride* is the amount of movement between applications of filter to the input image. In the convolutional layer, there will be a kernel matrix that sweeps through the input image with a given stride and outputs the value that is generally the sum of the product of corresponding elements in an image. This process extracts features from the image called a *feature map*.

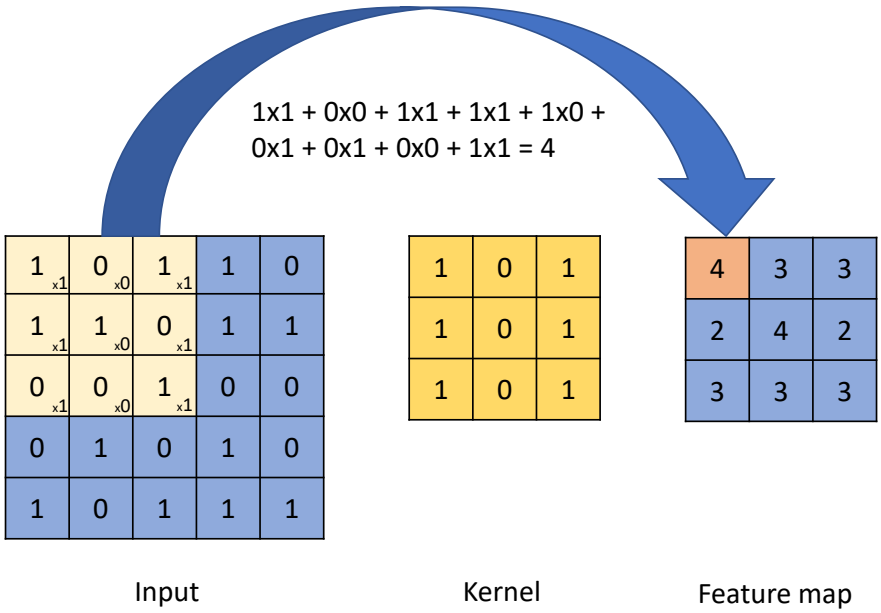


Figure 4.1: An example of 2D convolutional layer with kernel size 3×3 and stride $= 1$. The input is 5×5 and we have shown the values of the kernel in input layer that gives the value of highlighted value in the feature map.

4.1.2 Pooling Layer

A *pooling layer* replaces the output of the network at a certain location with a summary statistic of the nearby outputs (Goodfellow et al., 2016). Typical pooling functions include max pooling, average pooling, and L_2-norm pooling that respectively output maximum, average, and the square root of the sum of squares (Scherer et al., 2010). A pooling layer is usually applied after a convolutional layer. Pooling is a way to effectively reduce the amount of computation because it sub-samples the features in the feature map. The down-sampling or sub-sampling of the pooling layer also reduces the resolution of the feature map and achieves spatial invariance (Scherer et al., 2010).

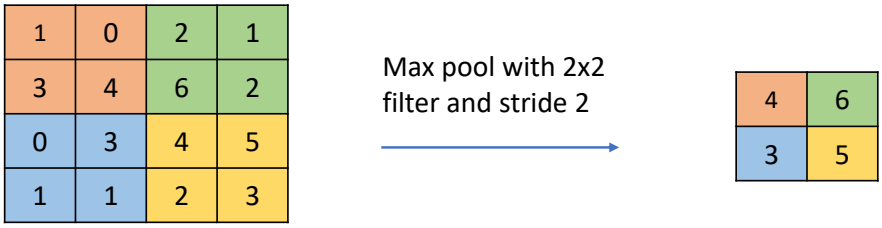


Figure 4.2: An example of pooling layer with filter size 2×2 and stride $= 2$.

4.1.3 Fully Connected Layer

The fully connected layer is same as a traditional ANN. The term *fully connected* implies that every neuron in the previous layer is connected to every neuron in the next layer. Convolutional and pooling layers extract high-level features from the input image. The purpose of the fully connected layers is to take in these features and provide output based on the training dataset. Sigmoid or softmax activation functions are commonly used as activation functions for fully connected layers.

4.1.4 Activation Function

The purpose of the activation function is to make the neural network learn non-linear functional mappings between the inputs and the output variable. The output of a neuron is generally the activation function applied to the sum of products of its inputs and their corresponding weights. The most popular activation functions are sigmoid, tanh, softmax, and ReLU functions. The following subsections discusses these activation functions, their mathematical representations, and graphical representations.

Sigmoid

The sigmoid activation function is also referred to as the logistic function. It is an S-shaped curve. The output of a sigmoid activation function is between 0 and 1. Therefore, it is especially used for models where we have to predict the probability as an output. Since probability of anything exists only between the range of 0 and 1, sigmoid is usually used in the output layer of a binary classifier. The mathematical expression for the sigmoid function is

$$f(x) = \frac{1}{1 + e^{-x}} \tag{4.1}$$

where x is the net value computed based on inputs, weights, and bias of that neuron.

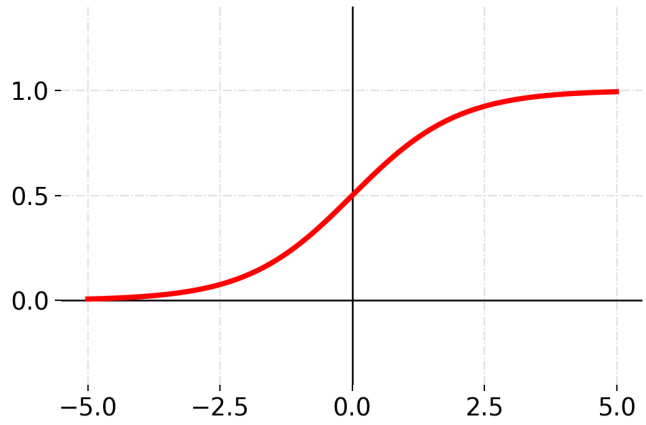


Figure 4.3: Graphical representation of sigmoid activation function.

Hyperbolic Tangent Function: Tanh

The hyperbolic tangent function, known as *tanh*, is similar to the sigmoid function. Tanh is a smooth zero-centered function. The range of tanh function is between -1 and 1. The mathematical expression for the tanh function is

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{4.2}$$

where x is the net value computed based on inputs, weights, and bias of that neuron.

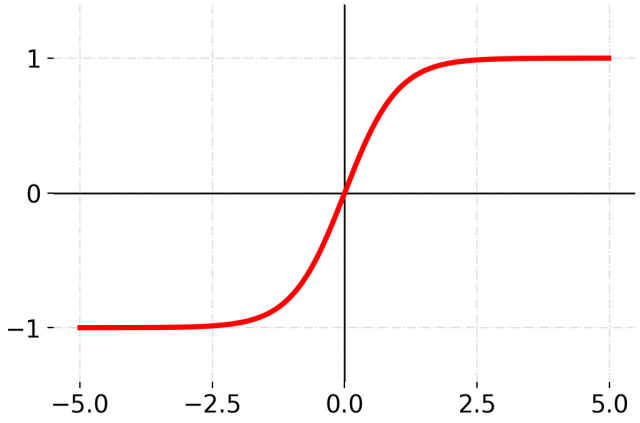


Figure 4.4: Graphical representation of tanh activation function.

Softmax

The *softmax* computes the exponent of the individual input divided by the sum of exponents of all the inputs. The output of a softmax function is a vector of length equal to number of classes, and range of values lies between 0 and 1. The sum of these vector values is equal to 1. The softmax function is preferred in the output layer of the multi-class models (Nwankpa et al., 2018). The mathematical expression of the softmax function is

$$f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{4.3}$$

where x_i is the real number of a input vector.

Rectified Linear Unit (ReLU) Function

The *Rectified Linear Unit* (ReLU) activation function (Nair and Hinton, 2010) is widely used for deep learning applications. Sigmoid and tanh activation functions have gradients (derivative of the function) that are smaller than one. If multiple layers are used, the product of these gradients decreases exponentially and results in vanishing gradient problem. Vanishing gradient slows the learning process in deep learning applications. ReLU activation has gradient of 1 for positive values and zero for any value less than zero. The ReLU activation function is given by

$$f(x) = max(0, x) \tag{4.4}$$

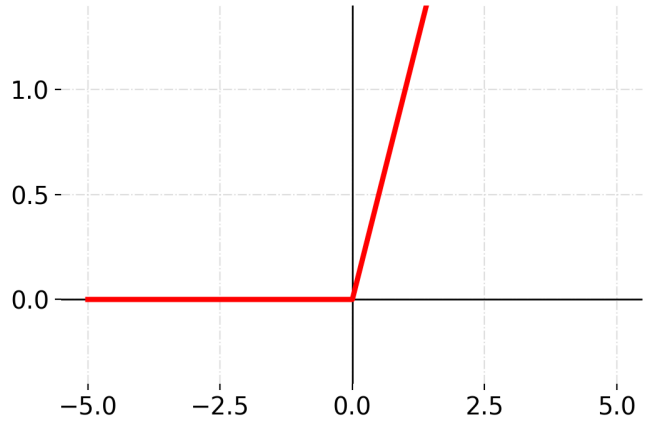


Figure 4.5: Graphical representation of ReLU activation function.

Leaky ReLU

While training a neural network with ReLU activation function, if the output of a unit is zero the gradient for that unit is zero and the unit *dies*. *Leaky ReLU* (Maas et al., 2013) was introduced to avoid the dying ReLU problem. For $x \leq 0$, LReLU has a small slope α. The LReLU activation function is given by

$$f(x) = \begin{cases} x, & if\ x>0 \\ \alpha x, & if\ x \leq 0. \end{cases} \tag{4.5}$$

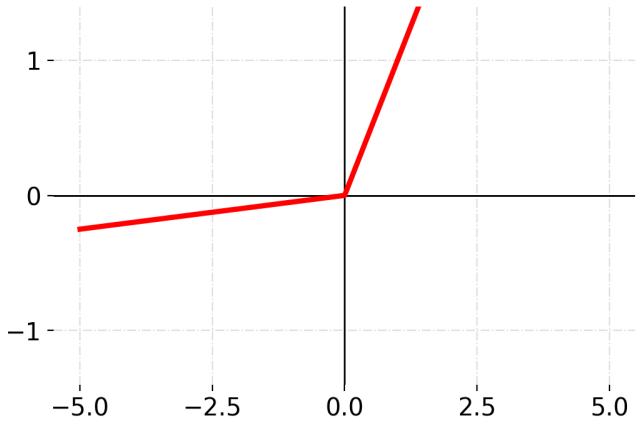


Figure 4.6: Graphical representation of LReLU activation function.

4.1.5 Dropout

In neural nets, increasing the number of hidden layers improves the network's ability to learn complicated relations between their inputs and outputs. However, with limited training data, these models may learn the noise in the training data, which may not exist in the test data. This leads to *overfitting* in such networks. Large networks are slow to learn and require more training data to get favorable results. To address this problem, dropout was introduced (Srivastava et al., 2014). *Dropout* refers to randomly removing neurons and its weights at the training phase. This prevents units from co-adapting too much. In the simplest case, each neuron is retained with a probability p independent of other neurons, where p can be picked using a validation set or simply set to 0.5.

4.2 Architecture of CNN

The schematic of the CNN used in this study is shown in Figure 4.7. The architecture of the CNN used in this study is from Kim et al. (2018). The input to the CNN layer is $2 \times 31 \times 31$ image. The first channel is the visible image and the second channel is the infrared image. The first two layers are convolutional layers with filter size 3×3. The next layer is a max pooling layer, followed by two convolutional layers and another max pooling layer. The first layer has a filter size of $2 \times 3 \times 3$; and the number of filters is 32 with a stride of 1. This will extract 32 feature maps of size 29×29. The second layer has a filter size of $2 \times 3 \times 3$, and the number of filters is 32 with a stride of 1. This layer will extract feature maps of size 27×27. After two convolutional layers, a max pooling layer is used with a filter size of 2×2, and a stride of 2. The output of this layer will be 32 feature maps with a size of 13×13. The third and fourth layers are convolutional layers with 64 filters, followed by the second max pooling layer. The output of the last pooling layer is flattened (i.e., converting data into 1-dimensional array for inputting it to the next layer) and sent as an input to the fully connected layer with 256 neurons. The last layer is the output layer with two neurons for classification. The predicted class is the class corresponding to the highest value in the output vector.

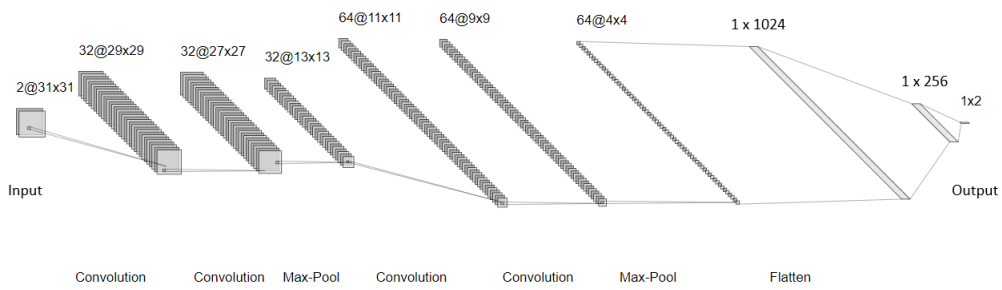


Figure 4.7: Schematic diagram of CNN.

Leaky ReLU with a slope (α) of 0.01 is used for the first convolutional layer. The softmax activation function is used in the output layer. The ReLU activation function is used on second, third, and fourth convolutional layers. As the neural networks become deeper, they are likely to fall into overfitting. To solve this problem, dropout (Srivastava et al., 2014) is used after two max pooling layers with p value 0.5.

From the model summary in Table 4.1, the CNN consisted of 328194 parameters. In the first convolutional layer, the shape of the filter is $2 \times 3 \times 3$ and there is a bias term with 32 filters. Therefore, number of parameters in the first convolutional layer is $(2 \times 3 \times 3 + 1) \times 32 = 608$. Similarly, the number of parameters in the second convolutional layer is $(32 \times 3 \times 3 + 1) \times 32 = 9248$, the number of parameters in the third convolutional layer is $(32 \times 3 \times 3 + 1) \times 64 = 18496$, the number of parameters in the fourth convolutional layer is $(64 \times 3 \times 3 + 1) \times 64 = 36928$, the number of parameters in the first fully connected layer is $(1024 + 1) \times 256 = 262400$ and the number of parameters in the second fully connected layer is $(256 + 1) \times 2 = 514$. Pooling layers do not have any parameters. This gives a total of 328,194 parameters.

Layer (type)	**Output Shape**	**Param #**
conv2d_1 (Conv2D)	(32, 29, 29)	608
conv2d_2 (Conv2D)	(32, 27, 27)	9248
max_pooling2d_1	(32, 13, 13)	0
dropout_1 (Dropoout)	(32, 13, 13)	0
conv2d_3 (Conv2D)	(64, 11, 11)	18496
conv2d_4 (Conv2D)	(64, 9, 9)	36928
max_pooling2d_2	(64, 4, 4)	0
dropout_2 (Dropoout)	(64, 4, 4)	0
flatten_1 (Flatten)	(1024)	0
dropout_3 (Dropoout)	(1024)	0
dense_1 (Dense)	(256)	262400
dense_2 (Dense)	(2)	514
Total Parameters:		328194

Table 4.1: Summary of the CNN model.

4.3 Modeling

While training feedforward neural networks for supervised learning, the output is calculated using forward propagation. Output is calculated by multiplying inputs with weights and adding bias at various layers. Loss is calculated using the true label and predicted value. *Backpropagation* is a widely used algorithm in feedforward neural networks. During backpropagation, the gradient of the loss function is calculated with respect to parameters (weights and bias) of a neural network. Parameters of a neural network are updated using the above gradients. In most feedforward neural networks, parameters are learned by forward propagation and backpropagation. Cross-entropy loss or log loss is used as a loss function in this thesis. *Cross-entropy* loss measures the performance of a classification model whose output is a value between 0 and 1. Cross-entropy loss increases as predicted values diverge from the actual labels. An ideal model has a cross-entropy loss of zero. Below is an equation used to compute cross-entropy loss for binary

classifier.

$$L(y, \hat{y}) = -y \log \hat{y} + (1 - y) \log(1 - \hat{y}) \tag{4.6}$$

where y is a vector of true labels and $\hat{y}$ is a vector of estimated values.

Gradient descent (Ruder, 2017) is an optimization algorithm used to minimize or maximize a function. Here we used gradient descent to minimize cross-entropy loss. Thus, gradient descent is used to update parameters of our model. There are various types of gradient descent, depending upon the amount of data used to calculate the gradient of the cost function during gradient descent. The first type is *batch gradient descent*, in which the complete dataset is used to compute the gradient of the cost function and perform one update. Batch gradient descent may take a long time to converge as parameters get updated only once for each forward and backpropagation on entire data. The second type is *stochastic gradient descent* (SGD), in which only one training example is used in every iteration to compute the gradient of a cost function. SGD is faster for large datasets but may not achieve accuracy. SGD is faster because parameters get updated for each observation, i.e., parameters get updated for forward and backpropagation on each observation. It may not achieve better accuracy as the parameters are always updated for each observation, thereby SGD updates parameters locally to that observation. The third type is *mini-batch gradient descent*, in which m training examples are used as a batch to compute the gradient of the cost function. Thus mini-batch gradient descent is a tradeoff between batch gradient descent and SGD and has advantages of both batch gradient descent and SGD. Therefore, mini-batch gradient descent is accurate and fast to implement. In this thesis, we used mini-batch gradient descent with a batch size m of 128.

In most cases, gradient descent with *momentum* works faster than standard gradient descent. The primary idea is to compute exponentially weighted average of gradients and then use this gradient to update parameters. Another variation of gradient descent is *RMSprop* (Root Mean Square prop). In search space, RMSprop helps to make progress (learns faster) in the direction of optimal solution and decrease oscillations in other directions. Thereby, RMSprop helps to converge to optimal parameters quickly. *Adaptive moment estimation* (Adam) optimizer is a combination of both gradient descent with momentum and RMSprop (Kingma and Ba, 2017). So, it has advantages of both gradient descent with momentum and RMSprop. Recommended parameters in Kingma and Ba (2017) are used in this thesis. *Epoch* is one forward propagation and backpropagation on entire training data. Learning algorithms like feedforward neural networks use multiple epochs during their training. We trained the CNN model with 100 epochs on the training data.

Chapter 5

Approaches and Results

This chapter discusses approaches to detect OTs along with the results of these approaches. The CNN architecture in all these methods is almost the same. In one of the methods, we used tropopause temperature data as the third channel in the input. The performance of the models was measured using probability of detection (POD), false alarm ratio (FAR) and critical success index (CSI).

	Actual positive	Actual negative
Predict positive	True positive (TP)	False Positive (FP)
Predict negative	False negative (FN)	True negative (TN)

Table 5.1: An example of cost matrix for binary classification.

$$POD = \frac{\text{Number of OTs detected correctly}}{\text{Total number of OT reference points}}$$
$$= \frac{TP}{TP + FN}$$

$$FAR = \frac{\text{Number of OTs falsely detected by the CNN model}}{\text{Number of OTs detected by the CNN}}$$
$$= \frac{FP}{TP + FP}$$

$$CSI = \frac{\text{Number of OTs detected correctly}}{\text{Total number of OT reference points} + \text{Number of OTs falsely detected}}$$
$$= \frac{TP}{TP + FP + FN}$$

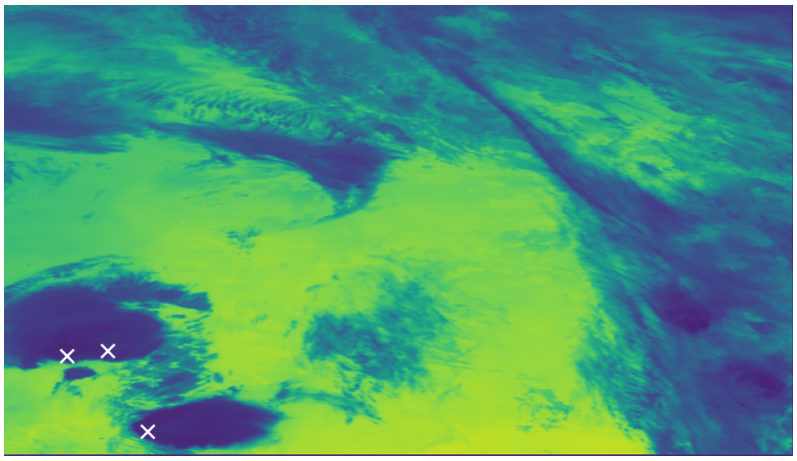
The performance of each approach is evaluated using these metrics. It is better to have a model with high probability of detection (POD) and low false alarm ratio (FAR). Fifty full images at different timestamps are used for validation. The OT centers on satellite images that are labelled by people are used as reference. The test images are chopped into patches of size 31 $\times$ 31. Each patch is labelled as "OT" or "non-OT" depending on whether that patch has OT center or not. There are total 406 OT occurrence patches, and 693794 non-OT patches in test data. So, the ratio of OT and non-OT patches in test data is 1:1709.

5.1 Approach 1: Random Undersampling

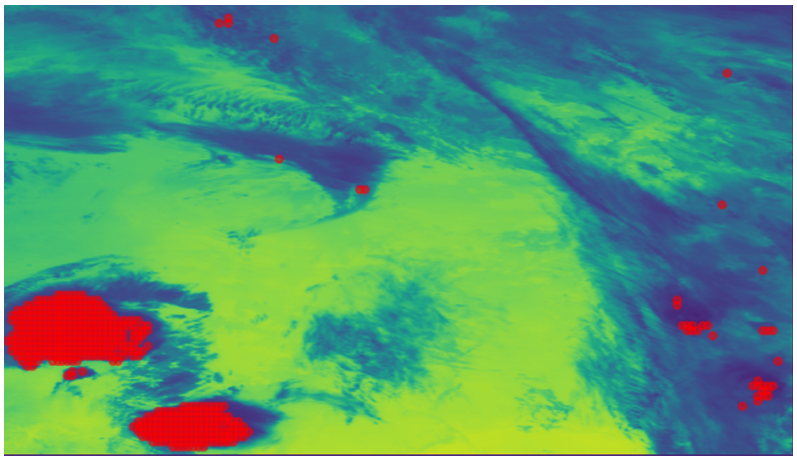
In this approach, training data is prepared from infrared and visible images as explained in Section 3.4. The image patches labeled as OT occurrences have OT centers either at the center, or slightly offset from the center, or close to the

edge of the 31 $\times$ 31 patch. Non-OT occurrence images are taken using a random undersampling technique such that the number of OT occurrences and non-OT occurrences is in the ratio 1:1.

The model is tested on full images in the test data. The model has a POD of 97.3% , FAR of 97.92% and CSI 0.02. It is observed that most of the patches having colder pixels are detected by the model as OT occurrence. In Figure 5.1, the above image is an infrared image showing the actual OT occurrences and the below image is an infrared image showing model detected OT regions in red circle. Based on the results, the CNN model is identifying the difference between cold temperature and warmer temperature pixels instead of the actual difference between OT occurrence and non-OT occurrence. This model generated a lot of false positives because there are many colder regions which do not have OT occurrences. On comparing both the images in Figure 5.1a and 5.1b, we can say that the model classifies most of the cold regions as OT regions. This can be because random undersampling of majority class (non-OT) can increase the variance of the classifier by discarding important samples.



(a) An infrared image with actual OT center marked as white ×.



(b) An infrared image with predicted OT center marked in red circle.

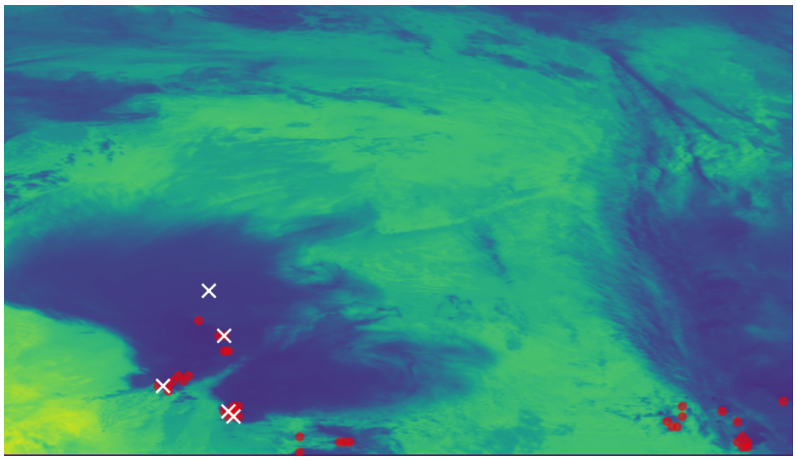
Figure 5.1: A figure showing the actual and predicted OT centers with Approach 1. The model is predicting most of the cold regions as OT occurrence.

5.2 Approach 2: Targeted Undersampling

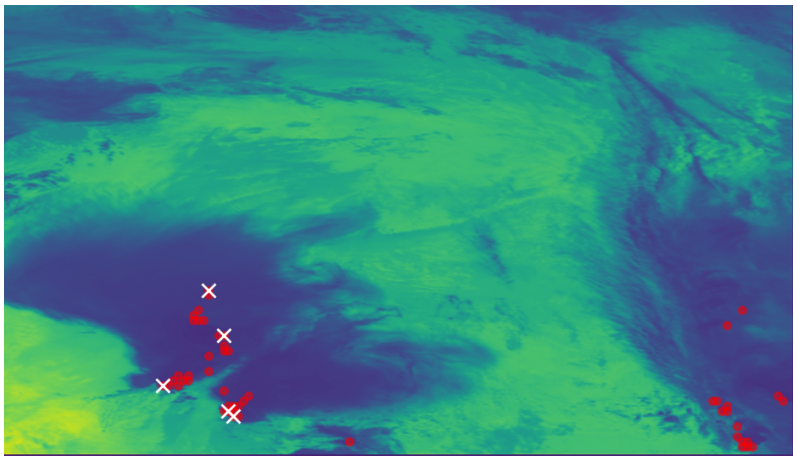
In the above approach, the model classifies most patches having colder pixels as OT occurrence and other images as non-OT occurrence. We hypothesized that adding additional targeted non-OT occurrence patches that are colder to the training data would help the model to learn the difference between OT and

non-OT regions rather than cold and warm regions. All the training data used in random undersampling is used for this approach. Additional non-OT occurrence patches that are close to an OT center but do not contain an OT center are added to the training data. The centers of these targeted samples are chosen at a distance between 12.5–20 km far from the OT center.

The POD, FAR, and CSI for this model are 72.66%, 90.44% and 0.092 respectively. Figure 5.2a shows the actual and predicted OT centers using Approach 2. It is clear from the image that this model is not detecting most cold regions as OT occurrences. Training using the new targeted undersampling data helped the model in learning the difference between cold patches that actually have OT occurrence vs the cold patches that do not have OT occurrence. Although the number of false positives in the model dropped significantly and the skill (CSI) increased, the probability of detection is reduced.



(a) An infrared image with actual and predicted OT centers with Approach 2.



(b) An infrared image with actual and predicted OT centers with Approach 3.

Figure 5.2: A figure showing the actual and predicted OT centers with Approach 2 and Approach 3. Actual OT centers are marked as white $\times$ and predicted OT centers are marked as red dot.

5.3 Approach 3: Cost-Sensitive Learning

Cost-sensitive learning addresses the imbalanced classification problem by using different costs for misclassifying any particular class. For binary classification, the cost or penalty of misclassifying the minority class as the majority class is

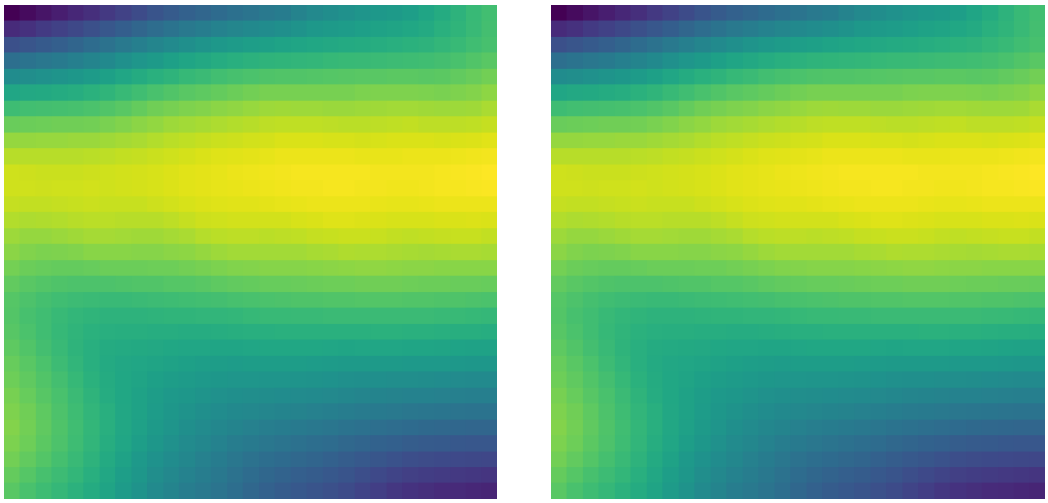
generally higher than misclassifying the majority class as a minority class. The backpropagation algorithm updates the weights of a neural network using the weighted misclassification cost. This approach makes neural networks pay more attention to the minority class. Let the cost of misclassifying OT reference as non-OT be n times the contrast case. Our hypothesis in this approach is that the cost-sensitive learning model will increase the probability of correctly detecting OT reference samples. After training and validating on test data, we took 10 as the optimal value of n. The approach uses the same training data used in targeted undersampling approach

The POD increased to 88.9% but FAR also increased to 96%. CSI value for this model is 0.039. Subfigure 5.2b shows the actual and predicted OT centers with this approach. This class-weighted model performed better in detecting the number of OT references as we hypothesized but the skill is reduced. By comparing Subfigure 5.2a and 5.2b, we can observe that the model using cost-sensitive learning along with targeted undersampling is detecting more OTs compared to Approach 2.

5.4 Approach 4: Local Normalization

For local normalization, the values of both infrared and visible images are normalized using a min-max approach. For other approaches, X_{min} and X_{max} values are calculated for the full image. The minimum and maximum values have a wide range, and when all pixels in an image are normalized based on these values, some important information may be lost. Mainly, this loses the difference in temperature between the overshooting top and its surrounding anvil clouds. To address this problem, the local normalization approach normalizes 31 $\times$ 31

infrared patches during training and testing. We hypothesized that using local normalization may help in clearly identifying OTs from their surroundings.



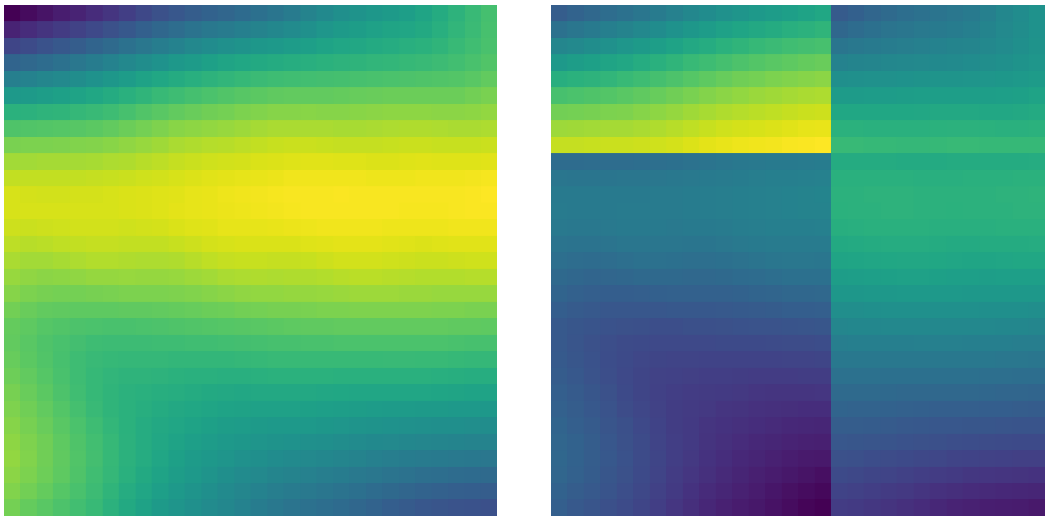
(a) Locally normalized image patch. (b) Normalization image patch.

Figure 5.3: Comparison of locally normalized and normalized image patches with OT at the center.

For training data, we took image patches similar to Approach 1 but brightness temperature patches are extracted from interpolated images at 500 m resolution before they are normalized. These patches are individually normalized using min-max approach, where X_{min} and X_{max} are the minimum and maximum values of the 31 $\times$ 31 patch respectively. While testing, the interpolated infrared images at 500 m are taken and the average value of the complete image is computed and subtracted from all pixels. Later, the image is chopped into patches of size 31 $\times$ 31. Each of these patches is normalized using the min-max approach mentioned above and tested on the trained model.

Using this approach the model has a POD of 90.39%, FAR of 99.47% and CSI value 0.005. Although locally normalizing each patch enhances the difference between each OT center and its surroundings in brightness temperature

image, discontinuity is created in the image visually. Subfigure 5.4 is the image patch with OT center at the center of the 31 $\times$ 31 patch. Subfigure 5.3b is the image patch from normalized image. Although these two patches look the same they differ in values at each pixel. While testing, dividing the full image to 31 $\times$ 31 patches and locally normalizing is causing discontinuity in patches and OT features are disrupted. Subfigure 5.4b shows part of an image with OT center at the center of patch while testing. It can be observed that OT features are discontinued in the individual patches.



(a) Locally normalized image patch using data construction method for training.

(b) Locally normalized image patch using data construction method for testing.

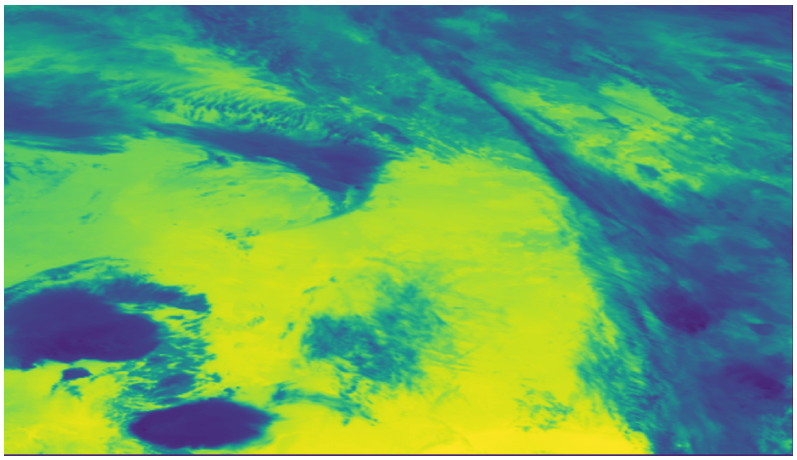
Figure 5.4: Comparison of locally normalized image patches constructed using training and testing methods.

5.5 Approach 5: 3-Channel Image

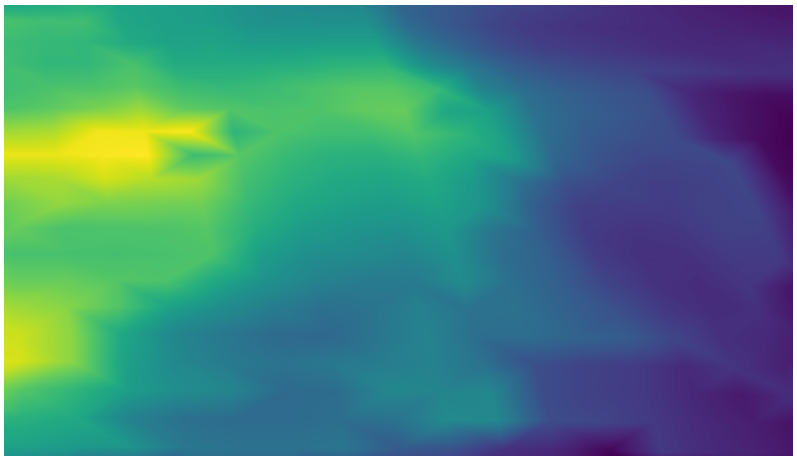
OTs can penetrate 2 km above the surrounding anvil clouds and have colder BTs compared to the anvil cloud temperature. These surrounding anvil clouds are said to have temperature at or near to that of tropopause level (Adler et al., 1985).

In the above mentioned four approaches, regions that are colder and do not have any OT region close to them are also identified as the OT occurrence. It is observed that the tropopause temperature in those regions is also colder. It can be observed from Figure 5.6 that both images have colder temperatures in the right regions. So, we used tropopause temperature data as a third channel in the input to the CNN model. Adding third channel image in the input increases the number of trainable parameters in first layer from 608 to 896 and the total trainable parameters are 328482. We hypothesized that including tropopause temperature as input would help the model to learn the relation between tropopause temperature and the infrared brightness temperature in classifying whether the given image has an OT or not. Training data in this approach is extracted similar to the targeted undersampling data and this approach also uses cost-sensitive learning method with the cost of misclassifying the OT as non-OT is ten times the contrast case.

This approach has a POD of 93.59%, FAR of 97.18%, and CSI of 0.028. This approach helped in detection of more OTs but the number of false positives are also increased.



(a) Infrared image.



(b) Image of tropopause temperature in Kelvin.

Figure 5.5: Infrared and tropopause temperature images.

5.6 Approach 6: Temperature Difference

Some of the false positives in the previous approaches are at places that are colder in general. After looking at the tropopause values, it is observed that the tropopause temperature in those regions is colder compared to other regions. In the temperature difference approach, infrared images are converted to Kelvin by first calculating the scene radiance, followed by the effective temperature, and

then the brightness temperature. The steps for converting infrared image values to temperature are described in National Oceanic and Atmospheric Administration (2011) as follows:

Step 1: Scene radiance R at a pixel with value X is calculated using the formula

$$R = \frac{X - b}{m} \tag{5.1}$$

where the value of b and m are 5.2285 and 15.6854, respectively.

Step 2: Radiance is converted to effective temperature using the function

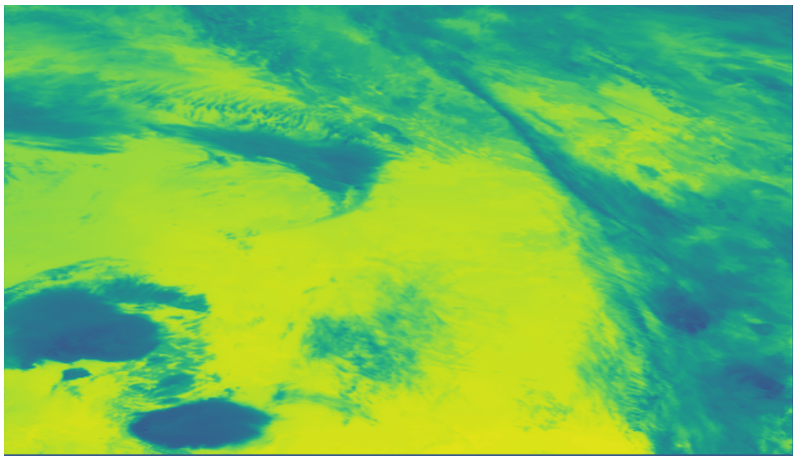
$$T_{\mathit{eff}} = \frac{c_2 v}{\log\left(1 + \dfrac{c_1 + v^3}{R}\right)} \tag{5.2}$$

where $c_1 = 1.191\,066 \times 10^{-5}$, $c_2 = 1.438833$, and $v = 936.20$.

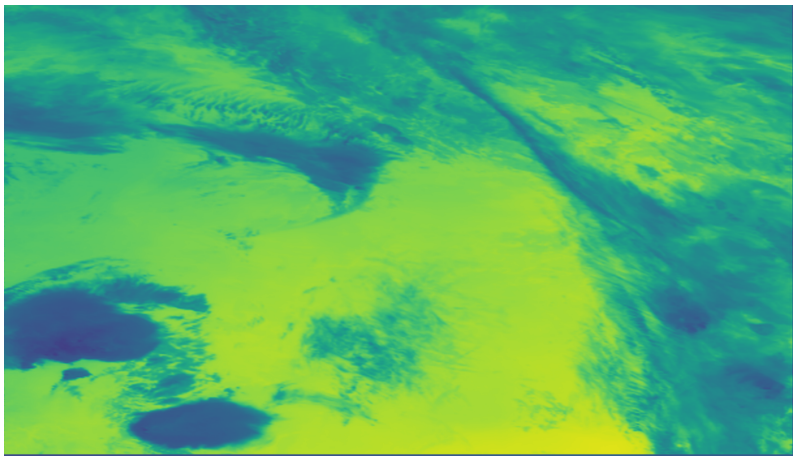
Step 3: Actual temperature T in Kelvin is computed from effective temperature T_{eff} using the following equation

$$T = \alpha + \beta T_{\mathit{eff}} \tag{5.3}$$

where $\alpha = -0.287\,561\,6$ and $\beta = 1.001\,258$.



(a) Infrared image.



(b) Temperature difference image.

Figure 5.6: Infrared image and temperature difference image generated by subtracting tropopause temperature from infrared image.

In infrared images, the value at each pixel is converted to actual temperatures using the steps mentioned above. New feature images are generated by subtracting tropopause temperature from actual temperature at each pixel. This results in negative values only at pixels that are colder than the tropopause temperature. These feature images are normalized using min-max approach. The new temperature image is similar to the tropopause temperature image and is shown in Figure 5.6b. For training, data is extracted similar to targeted sampling ap-

proach and cost-sensitive learning learning is used with 10 as value of n. POD, FAR, and CSI for this approach are 89.9%, 96.33%, and 0.036. The number of false positives reduced in this approach compared to the previous approach.

5.7 Approach 7: Tropopause Temperature Preprocessing

Most overshooting tops have temperatures colder than the tropopause temperature. In this approach, only patches having a minimum temperature colder than the minimum tropopause temperature in that patch are considered for training. In this step, all full images are divided into 31 $\times$ 31 size patches similar to how test images are divided in the previous approaches. If a patch has a minimum brightness temperature lower than the minimum tropopause temperature and contains an OT center, it is marked as an OT occurrence. Other patches that have a minimum temperature colder than the minimum tropopause temperature but do not contain an OT center are marked as a non-OT occurrence.

For training, non-OT labeled patches are randomly sampled and cost-sensitive learning is used with 10 as a value of n. While testing, all images are chopped into patches of size 31 $\times$ 31. If these patches have a minimum temperature warmer than the tropopause temperature in that patch, then that patch is marked as non-OT occurrence. Otherwise, the image patch is tested using the trained model to find OT occurrences. In this preprocessing step, some of the OTs are by default considered to be non-OT references. In the test data, 12% of the actual OTs are considered as non-OTs in the preprocessing step. For testing, using the same preprocessing approach all the 31 $\times$ 31 patches with a minimum BT warmer than minimum tropopause temperature in that location are labelled as non-OT. The remaining patches are tested using the CNN model to get the classification for

that patch. On the complete test data this approach performed with a POD of 79.31%, FAR of 90.94%, and CSI value 0.088. In this approach although the FAR is high, the false positives are closely located around the OT occurrences. This approach is time efficient. Testing on a full image using this approach takes significantly less time compared to the previous approaches.

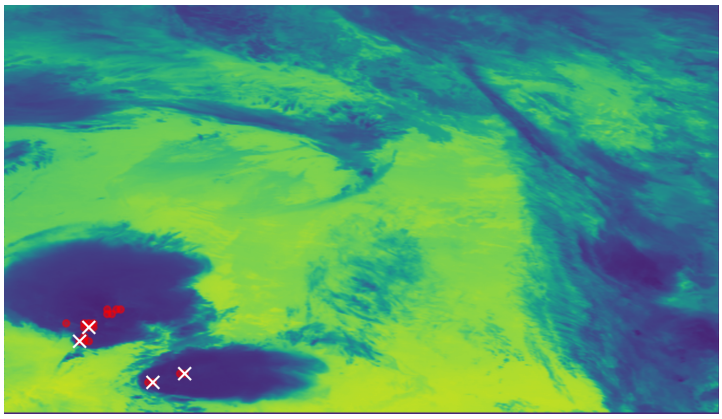


Figure 5.7: An infrared image with actual and predicted OT centers with Approach 7. Actual OT centers are marked as white × and predicted OT centers are marked as red dot.

5.8 Overall Results

The total number of actual OT references in the test data is 406. Table 5.2 summarises the number of true positive (TP), number of false positive (FP), POD, FAR, and CSI for all the approaches presented in this thesis. Approach 1 random undersamling has the best POD 97.29% with number of true positives 395. Approach 2 have lower FAR of 90.44% and number of false positives is 2794. Approach 2 also has the best CSI value of 0.092.

From the table 5.2, we can observe that Approach 2 Targeted Undersampling results in the best CSI value followed by Approach 7 Tropopause Temperature Preprocessing. Although the targeted undersampling approach results in slightly

Approach	true positives	false positives	POD (%)	FAR (%)	CSI
Approach 1: Random Undersampling	**395**	18649	**97.29**	97.92	0.021
Approach 2: Targeted Undersampling	295	**2794**	72.66	**90.44**	**0.092**
Approach 3: Cost-Sensitive Learning	361	8775	88.92	96.04	0.039
Approach 4: Local Normalization	367	69695	90.39	99.47	0.005
Approach 5: 3-Channel Image	380	13114	93.59	97.18	0.028
Approach 6: Temperature Difference	365	9598	89.90	96.33	0.036
Approach 7: Tropopause Temperature Preprocessing	322	3234	79.31	90.94	0.088

Table 5.2: Comparison of performances for different approaches.

better CSI value than the Approach 7, the model is predicting OTs that are not in the OT region. On the other hand, the tropopause temperature preprocessing method predicts most of its false positives close to actual OT occurrence. So, the tropopause temperature preprocessing method is more useful than other models.

Chapter 6

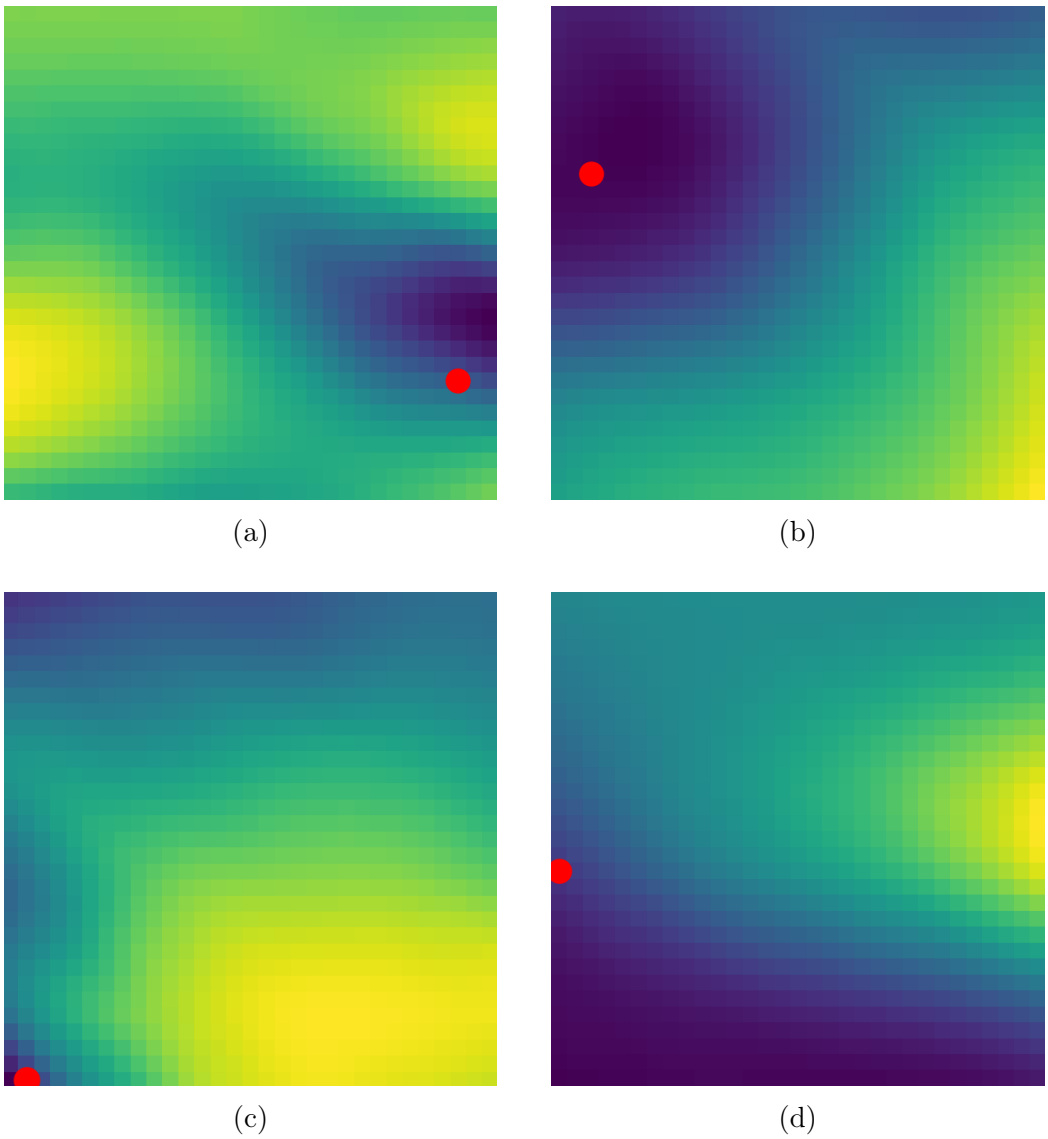
Discussion

This chapter discusses why the models are failing to detect some OTs and also the reasons for more false positives. After analysing the results of Approach 7 on each test image, we found three main points that are leading to low performance of the model. These three points will be discussed in this chapter.

6.1 Location of OT centers in Test Data

The results from the Approach 7 Tropopause temperature preprocessing show that the model is often detecting OT surrounding regions as OT occurrences even though they do not have the OT center. When the test images are randomly divided into patches, most of the OT centers are located close to edges. An OT occurrence is at center if the center of OT is located at the center 11 $\times$ 11 region of a 31 $\times$ 31 patch. A patch having OT center within 5 pixels offset from the center 11 $\times$ 11 region of a 31 $\times$ 31 patch is called offset patch. A peripheral patch is a 31 $\times$ 31 patch with the OT center within 5 pixels of the edge. Out of the total 406 OT occurrences, 61 OTs are located at center, 114 OTs are slightly

offset from the center, and 231 OTs are located at the periphery. The presence of OT centers near the edges not only generates false alarms in surrounding regions but also generates false negatives for the OT center patch itself.

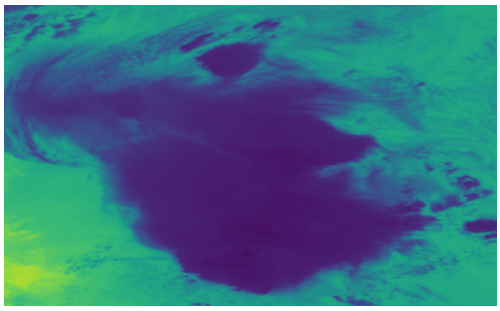


(a) (b)

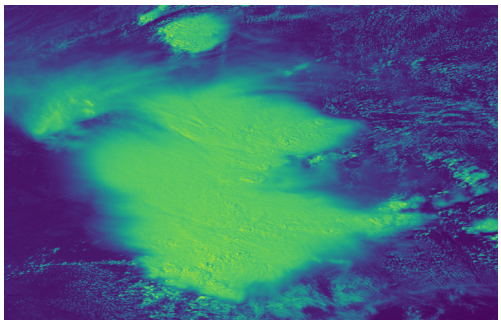
(c) (d)

Figure 6.1: Four patches of size 31 × 31 taken from test images with OT center pixel marked in red circle.

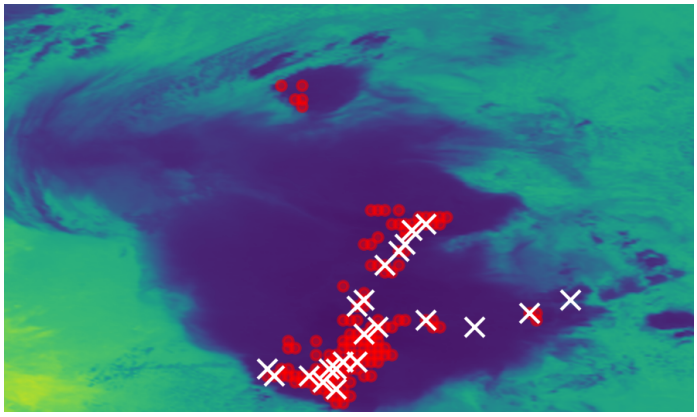
The two patches in Figure 6.1a, 6.1b are detected by the model with their surrounding regions falsely detected as OT occurrences. The two patches in



(a) Infrared image.



(b) Visible image.



(c) An infrared image with actual OT marked as white × and predicted OT center as red circle.

Figure 6.2: Region where OTs occur together.

Figure 6.1c, 6.1d have OT centers very close to the edge and they are not detected by the model. In case of Figure 6.1c, the three patches close to the OT center are incorrectly marked as positive.

6.2 Number of OTs occurring Together

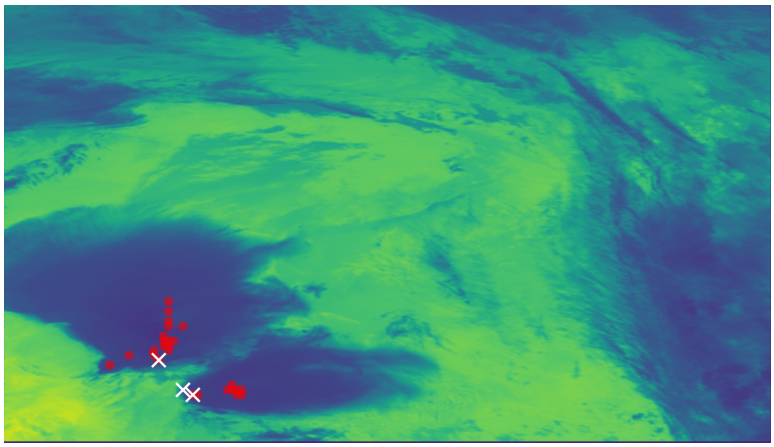
We have observed that the number of false alarms is high when there are multiple OTs occurring together. This may be because all the regions between these OTs have colder temperatures in their infrared image and have features of a dome when images are chopped into patches.

In Figure 6.2 it can be observed that when multiple OTs occur together, all the regions in between are also detected as OTs. This can be because these regions are cold and have dome-like structures.

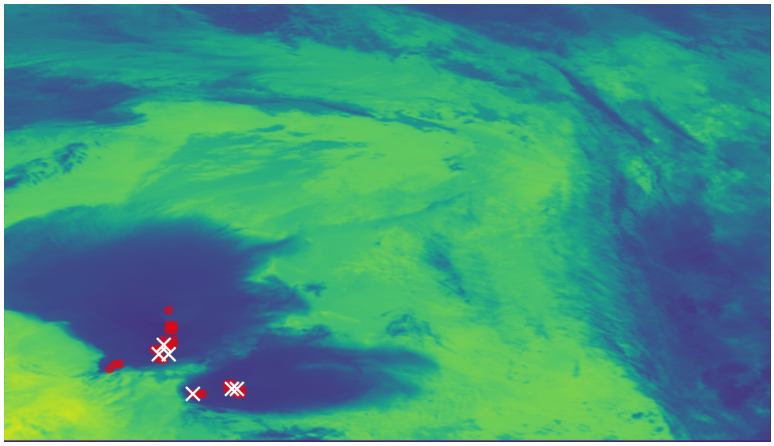
6.3 OT Detection Model Sensitivity

We observed the actual OT occurrences and the regions detected by the model as OTs across time on the test data. It is observed that the OT detection model is identifying regions as OT regions before and after there is an actual OT occurrence. The reason for this is that the model is sensitive to OT features. Either our model is detecting those features before and after OT occurrence or there is an actual OT occurrence that is not detected by the person while labeling data.

In Figure 6.3, we can observe that the model detected two regions as OT occurrence that are not marked as OT regions in the image at 15:25:37 UTC but are marked as OT occurrence in the image at 15:28:42 UTC. Similarly in Figure 6.4, we can observe that there are three regions where OTs are occurring at 15:41:05 UTC. The same three regions are detected as OT occurrence by model on image at 15:47:20 UTC but only one of those regions is marked as OT occurrence by the person assigned to label them.

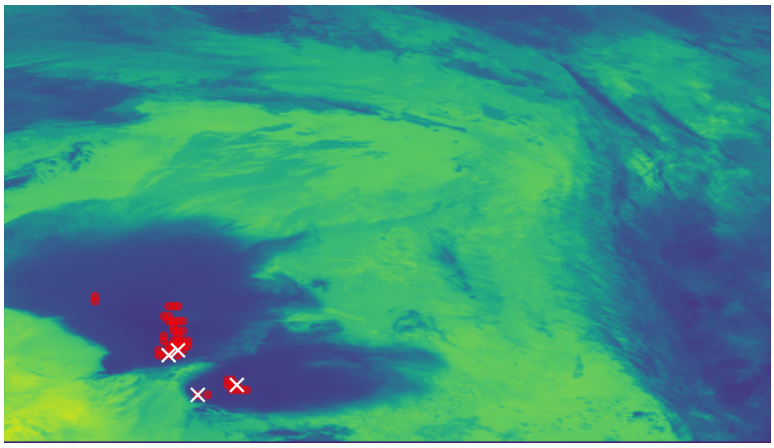


(a) An infrared image at 25 May 2015, 152537 UTC showing actual and predicted OT centers.

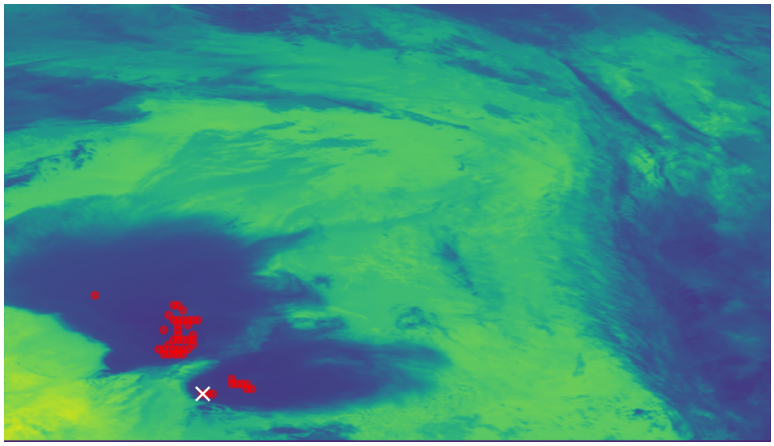


(b) An infrared image at 25 May 2015, 152842 UTC showing actual and predicted OT centers.

Figure 6.3: An example of the model detecting OTs in regions before they are no longer detected by the person assigned to label OTs. Actual OT centers are marked as white $\times$ and predicted OT centers are marked as red dot.



(a) An infrared image at 25 May 2015, 154105 UTC showing actual and predicted OT centers.



(b) An infrared image at 25 May 2015, 154720 UTC showing actual and predicted OT centers.

Figure 6.4: An example of the model detecting OTs in regions after they are no longer detected by people. Actual OT centers are marked as white $\times$ and predicted OT centers are marked as red circle.

Chapter 7

Conclusions and Future Work

This chapter is divided into two sections. The first section presents conclusions based on the analysis conducted for this research. The second section presents the recommendations of this thesis.

7.1 Conclusions

Detection of overshooting tops is important for weather forecasters, given the impact of storms producing OTs on weather conditions and global climate. In this thesis, GOES-14 satellite images are the primary sources of input to detect the OTs. The data collected for training and testing is for 9 hrs on 25 May 2015 for a long-lived mesoscale convective system (MCS) across Texas. Convolutional neural networks are used to extract features from the given images and output if there is an OT occurrence or not. Data undersampling and oversampling techniques are used to prepare data for training and tested on a test set. The performance of this model is increased by including data from regions closer to OTs that are non-OT occurrences and by implementing a cost sensitive learning

method. Tropopause temperature data is used in three different approaches. The final approach uses tropopause temperature data in the preprocessing step. Although this approach includes bias in classifying all OTs that are warmer than tropopause temperature as non-OTs, this approach reduces the number of false positives and computing time significantly. Although Approach 7 have higher FAR, most of the false positives are surrounding the OT reference. This may be due to the fact that most OT centers are close to the edges when the image is chopped into patches. Also, it is observed that the model is classifying an image patch as an OT before and after sometime of OT occurrence. From the results, we are able to show how imbalanced learning techniques can be used on satellite images.

7.2 Future Work

Based on the analysis conducted in this thesis, the following future work can be done:

1. One of the future work would be to combine the Approach 2 and Approach 7. In Approach 7, non-OT occurrence data is randomly sampled for training. If targeted sampling method can be implemented along with Approach 7, there is chance to decrease the number of false positives.

2. The detection models in this thesis find multiple detections of the same overshooting top. To detect each OT only once, we believe that the detection algorithm can be tested by using a smaller stride and then applying non-max suppression. Currently, we divide each complete test image into 31×31 patches with stride 31. If the stride is reduced the same pixels will

appear in more than one patch and there is more chance that an OT will be near the center of a patch in this case and this patch can have highest value for OT occurrence. To make sure the system detects each object only once a non-max suppression method can be used and only one patch is detected for each OT occurrence.

3. Application of postprocessing steps can increase the performance of the model. Previous studies have shown that applying postprocessing steps has reduced the false alarms significantly. One of the postprocessing steps can be used to first identify the coldest pixel in each OT detected patches and select only coldest pixel within any 15.5 km radius. The reason for suggesting 15.5 km radius is that minimum distance between any two OTs is 15.5 km. This will also make sure that each overshooting top is detected only once.

4. GOES-14 images are available with a temporal resolution of 1 minute. GOES-14 image data can be treated as a time-series data which can help in improving the performance of OT detection model.

Bibliography

Robert F. Adler, Michael J. Markus, and Douglas D. Fenn. Detection of severe midwest thunderstorms using geosynchronous satellite data. *Monthly Weather Review*, 113(5):769–781, 1985. doi: 10.1175/1520-0493(1985)113<0769:DOSMTU>2.0.CO;2. URL https://doi.org/10.1175/1520-0493(1985)113<0769:DOSMTU>2.0.CO;2.

KM Bedka, WF Feltz, JR Mecikalksi, RD Sharman, Annelise Lenz, and Jordan Gerth. Satellite signatures associated with significant convectively-induced turbulence events. In *Proc. Joint Meteorological Satellite and 15th Satellite Meteorology and Oceanography Conf*, 2007.

Kristopher Bedka, Jason Brunner, Richard Dworak, Wayne Feltz, Jason Otkin, and Thomas Greenwald. Objective Satellite-Based Detection of Overshooting Tops Using Infrared Window Channel Brightness Temperature Gradients. *Journal of Applied Meteorology and Climatology*, 49(2):181–202, February 2010. ISSN 1558-8424, 1558-8432. doi: 10.1175/2009JAMC2286.1. URL http://journals.ametsoc.org/doi/10.1175/2009JAMC2286.1.

Kristopher Bedka, Elisa M. Murillo, Cameron R. Homeyer, Benjamin Scarino, and Haiden Mersiovsky. The Above-Anvil Cirrus Plume: An Important Severe Weather Indicator in Visible and Infrared Satellite Imagery. *Weather and Forecasting*, 33(5):1159–1181, October 2018. ISSN 0882-8156, 1520-0434. doi: 10.1175/WAF-D-18-0040.1. URL http://journals.ametsoc.org/doi/10.1175/WAF-D-18-0040.1.

Kristopher M. Bedka and Konstantin Khlopenkov. A Probabilistic Multispectral Pattern Recognition Method for Detection of Overshooting Cloud Tops Using Passive Satellite Imager Observations. *Journal of Applied Meteorology and Climatology*, 55(9):1983–2005, September 2016. ISSN 1558-8424, 1558-8432. doi: 10.1175/JAMC-D-15-0249.1. URL http://journals.ametsoc.org/doi/10.1175/JAMC-D-15-0249.1.

Kristopher M. Bedka, Richard Dworak, Jason Brunner, and Wayne Feltz. Validation of satellite-based objective overshooting cloud-top detection methods using cloudsat cloud profiling radar observations. *Journal of Applied Meteorology and*

Climatology, 51(10):1811–1822, 2012. doi: 10.1175/JAMC-D-11-0131.1. URL https://doi.org/10.1175/JAMC-D-11-0131.1.

Ronald Gelaro, Will McCarty, Max J. Suárez, Ricardo Todling, Andrea Molod, Lawrence Takacs, Cynthia A. Randles, Anton Darmenov, Michael G. Bosilovich, Rolf Reichle, Krzysztof Wargan, Lawrence Coy, Richard Cullather, Clara Draper, Santha Akella, Virginie Buchard, Austin Conaty, Arlindo M. da Silva, Wei Gu, Gi-Kong Kim, Randal Koster, Robert Lucchesi, Dagmar Merkova, Jon Eric Nielsen, Gary Partyka, Steven Pawson, William Putman, Michele Rienecker, Siegfried D. Schubert, Meta Sienkiewicz, and Bin Zhao. The modern-era retrospective analysis for research and applications, version 2 (MERRA-2). *Journal of Climate*, 30(14):5419–5454, 2017. doi: 10.1175/JCLI-D-16-0758.1. URL https://doi.org/10.1175/JCLI-D-16-0758.1.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

Global Modeling and Assimilation Office (GMAO) (2015). Goddard Earth Sciences Data and Information Services Center (GES DISC). https://disc.gsfc.nasa.gov/datasets/M2I3NVASM_5.12.4/summary, Last accessed on 2020-02-15.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Dianyuan Han. Comparison of commonly used image interpolation methods. In *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering*, pages 1556–1559. Atlantis Press, 2013/03. ISBN 978-90-78677-61-1. doi: https://doi.org/10.2991/iccsee.2013.391. URL https://doi.org/10.2991/iccsee.2013.391.

Cameron R Homeyer and Matthew R Kumjian. Microphysical characteristics of overshooting convection from polarimetric radar observations. *Journal of the Atmospheric Sciences*, 72(2):870–891, 2015.

Hiroshi Inoue. Data Augmentation by Pairing Samples for Images Classification. *arXiv:1801.02929 [cs, stat]*, April 2018. URL http://arxiv.org/abs/1801.02929. arXiv: 1801.02929.

Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167 [cs]*, March 2015. URL http://arxiv.org/abs/1502.03167. arXiv: 1502.03167.

Miae Kim, Jungho Im, Haemi Park, Seonyoung Park, Myong-In Lee, and Myoung-Hwan Ahn. Detection of Tropical Overshooting Cloud Tops Using Himawari-8 Imagery. *Remote Sensing*, 9(7):685, July 2017. ISSN 2072-4292. doi: 10.3390/rs9070685. URL http://www.mdpi.com/2072-4292/9/7/685.

Miae Kim, Junghye Lee, and Jungho Im. Deep learning-based monitoring of overshooting cloud tops from geostationary satellite data. *GIScience & Remote Sensing*, 55(5):763–792, September 2018. ISSN 1548-1603, 1943-7226. doi: 10.1080/15481603.2018.1457201. URL https://www.tandfonline.com/doi/full/10.1080/15481603.2018.1457201.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, January 2017. URL http://arxiv.org/abs/1412.6980. arXiv: 1412.6980.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60 (6):84–90, May 2017. ISSN 00010782. doi: 10.1145/3065386. URL http://dl.acm.org/citation.cfm?doid=3098997.3065386.

Yann LeCun, Koray Kavukcuoglu, and Clement Farabet. Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 253–256, Paris, France, May 2010. IEEE. ISBN 978-1-4244-5308-5. doi: 10.1109/ISCAS.2010.5537907. URL http://ieeexplore.ieee.org/document/5537907/.

Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

Petra Mikuš and Nataša Strelec Mahović. Satellite-based overshooting top detection methods and an analysis of correlated weather conditions. *Atmospheric Research*, 123:268 – 280, 2013. ISSN 0169-8095. doi: https://doi.org/10.1016/j.atmosres.2012.09.001. URL http://www.sciencedirect.com/science/article/pii/S0169809512002980.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

National Oceanic and Atmospheric Administration. Conversion of gvar infrared data to scene radiance or temperature, 2011. https://www.ospo.noaa.gov/Operations/GOES/calibration/gvar-conversion.html, Last accessed on 2020-02-26.

Andrew J. Negri. Cloud-top structure of tornadic storms on 10 april 1979 from rapid scan and stereo satellite observations. *Bulletin of the American Meteorological Society*, 63(10):1151–1159, 1982. doi: 10.1175/1520-0477-63.10.1151. URL https://doi.org/10.1175/1520-0477-63.10.1151.

Andrew J. Negri and Robert F. Adler. Relation of satellite-based thunderstorm intensity to radar-estimated rainfall. *Journal of Applied Meteorology*, 20(3):288–300, 1981. doi: 10.1175/1520-0450(1981)020<0288:ROSBTI>2.0.CO;2. URL https://doi.org/10.1175/1520-0450(1981)020<0288:ROSBTI>2.0.CO;2.

NOAA CLASS. Comprehensive large array-data stewardship system (CLASS). https://www.bou.class.noaa.gov/saa/products/welcome, Last accessed on 2019-08-15.

Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation functions: Comparison of trends in practice and research for deep learning. *CoRR*, abs/1811.03378, 2018. URL http://arxiv.org/abs/1811.03378.

Luis Perez and Jason Wang. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *arXiv:1712.04621 [cs]*, December 2017. URL http://arxiv.org/abs/1712.04621. arXiv: 1712.04621.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. arxiv 2018. *arXiv preprint arXiv:1804.02767*, 2019.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv:1609.04747 [cs]*, June 2017. URL http://arxiv.org/abs/1609.04747. arXiv: 1609.04747.

Santhosh G. Zoom an image with different interpolation types, code project, 2011. https://www.codeproject.com/Articles/236394/Bi-Cubic-and-Bi-Linear-Interpolation-with-GLSL, Last accessed on 2020-05-07.

Dominik Scherer, Andreas Müller, and Sven Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In *International conference on artificial neural networks*, pages 92–101. Springer, 2010.

Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

Martin Setvák, Robert M Rabin, and Pao K Wang. Contribution of the modis instrument to observations of deep convective storms and stratospheric moisture detection in goes and msg imagery. *Atmospheric research*, 83(2-4):505–518, 2007.

Martin Setvák, Daniel T. Lindsey, Robert M. Rabin, Pao K. Wang, and Alžbeta Demeterová. Indication of water vapor transport into the lower stratosphere above midlatitude convective storms: Meteosat Second Generation satellite observations and radiative transfer model simulations. *Atmospheric Research*, 89(1-2):170–180, July 2008. ISSN 01698095. doi: 10.1016/j.atmosres.2007.11.031. URL https://linkinghub.elsevier.com/retrieve/pii/S0169809508000136.

Martin Setvák, Daniel T. Lindsey, Petr Novák, Pao K. Wang, Michaela Radová, Jochen Kerkmann, Louie Grasso, Shih-Hao Su, Robert M. Rabin, Jindřich Šťástka, and Zdeněk Charvát. Satellite-observed cold-ring-shaped features atop deep convective clouds. *Atmospheric Research*, 97(1-2):80–96, July 2010. ISSN 01698095. doi: 10.1016/j.atmosres.2010.03.009. URL https://linkinghub.elsevier.com/retrieve/pii/S016980951000058X.

Martin Setvák, Kristopher Bedka, Daniel T. Lindsey, Alois Sokol, Zdeněk Charvát, Jindřich Šťástka, and Pao K. Wang. A-Train observations of deep convective storm tops. *Atmospheric Research*, 123:229 – 248, 2013. ISSN 0169-8095. doi: https://doi.org/10.1016/j.atmosres.2012.06.020. URL http://www.sciencedirect.com/science/article/pii/S0169809512001950.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Jency Titus and Sebastian Geroge. A Comparison Study On Different Interpolation Methods Based On Satellite Images. *International Journal of Engineering Research*, 2(6):4, 2013.

weather events, tumblr. tumblr. https://weatherevents.tumblr.com/post/156825673024/some-of-the-features-to-be-found-in-a-supercell, Last accessed on 2020-05-07.

Wikipedia contributors. Equilibrium level — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Equilibrium_level&oldid=910098422, 2019. [Online; accessed 9-May-2020].

Xingrui Yu, Xiaomin Wu, Chunbo Luo, and Peng Ren. Deep learning in remote sensing scene classification: a data augmentation enhanced convolutional neural network framework. *GIScience & Remote Sensing*, 54(5): 741–758, September 2017. ISSN 1548-1603, 1943-7226. doi: 10.1080/15481603.2017.1323377. URL https://www.tandfonline.com/doi/full/10.1080/15481603.2017.1323377.