Detection of Overshooting Tops in Deep Convective Clouds using Deep Learning

1041045

Green Templeton College

University of Oxford
A thesis presented for the degree of
MSc in Computer Science

Trinity 2022

Abstract

Overshooting cloud tops are a symptom of violent updrafts in deep convective clouds. These violent updrafts often occur alongside 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 (CNNs) are well suited to this challenge as they perform excellently in image detection. This study uses two types of convolutional neural network to detect overshooting tops in deep convective clouds in GOES-16 satellite imagery from NOAAs Geostationary Operational Environmental Satellite. Visible and infrared images of GOES-16 satellite imagery are the primary source of input data. These images are divided into patches of approximately 15km wide. The model implemented takes in each patch and outputs its classification for that patch, whether it contains an overshooting tops or no overshooting top. One of the main challenges faced in training this model was the unequal class sizes, overshooting tops are very rare and therefore therefore 99.9% of patches do not contain one. To fix this, non-trivial sampling techniques and cost-sensitive learning were implemented. The final results of this model were very good, on par with the best detection models available with a probability of detection 81.2%, false alarm ratio 86.8, and critical success index 0.050. It was realised that one of the most important aspects of overshooting top detection was the availability and quality of the labelling of the images, and in this way this thesis also made improvements. A method of automatically labelling overshooting top regions the number of training instances can be greatly increased. 63% of the manually identified overshooting tops were within these regions so the regions serve as an appropriate guide for where overshooting tops are most likely to occur.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

OT	Overshooting Tops
GOES-16	Geostationary Operational Environmental Satellite
DCC	Deep Convective Clouds
ABI	Advanced Baseline Imager
VIIRS	Visible Infrared Imaging Radiometer Suite
IR	Infra-red
CNN	Convolutional Neural Network
GLM	Geostationary Lightning Mapper
NEXRAD	Next-generation Radar
WVD	Water-vapour difference
AMV	Atmospheric motion vectors

1 | Introduction

1.1 Background

This thesis will present a methodology to detect the presence of a particular type of weather phenomenon (overshooting tops) that occurs in thunderstorms. Firstly, I shall describe briefly the atmospheric conditions in a thunderstorm, this is important context for understanding how overshooting tops form and how we can use computer vision to detect them.

Thunderstorms are simply defined as storms with the presence of lightning and therefore thunder. Thunderstorms always occur in a cumulonimbus cloud, but a cumulonimbus cloud does not necessarily always contain lightning. Thunderstorms are frequently accompanied by severe weather: strong winds and heavy rain.

1.1.1 Formation of a Thunderstorm

Thunderstorms are most common around the tropics, where there is a great difference between the air temperatures of the warm, moist air from the tropical latitudes and the cooler air from polar latitudes.

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 objec-

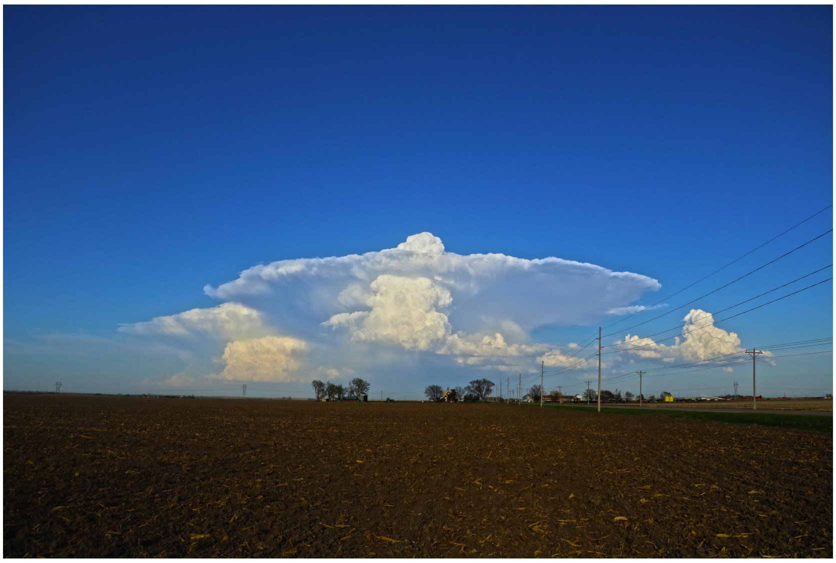

Figure 1.1: Image of an Overshooting top occurring above an anvil cloud taken from the ground

tives 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.

Overshooting tops (OTs), also called the penetrating tops, is a dome-like protrusions forming above a cumulonimbus anvil (K. M. Bedka & 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. 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 & Kumjian, 2015).

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 ((K. Bedka et al., 2010); (Homeyer & Kumjian, 2015); (K. M. Bedka & Khlopenkov, 2016); (Kim et al., 2017), (Kim, Lee, & Im, 2018)). Detecting OTs can be useful to forecasters for warning decision making as OTs can precede severe weather events 2 by 30 minutes or more ((?, ?)). OTs can produce turbulences at vast distance as they interact and penetrate through the tropopause, which can be a significant hazard for the aircraft ((K. Bedka et al., 2010)). Because of the hazards associated with OTs, various OT detection models have been developed ((K. Bedka et al., 2010); (K. M. Bedka & Khlopenkov, 2016); (Kim et al., 2017), (Kim et al., 2018)). Satellite data is commonly used for detecting OTs. Several studies have used images of visible and infrared channels to detect OTs ((K. M. Bedka & Khlopenkov, 2016), (Kim et al., 2017), (Kim et al., 2018)). OTs can penetrate more than 2 km above the surrounding anvil cloud ((Homeyer & Kumjian, 2015); (K. M. Bedka & 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.

However, not all OTs are significantly colder than the surrounding anvil cloud. A good outcome of the thesis would eb iOne of the goals of this thesis will be to produce a model which does not rely on the infrared channel (measuring temperature) to detect OTs. Forgoing the infrared channel as an input will allow for much earlier detection of OTs and detection of smaller OTs, creating in effect a detector for ’strong storm convection’ rather than for OTs.

Previous studies have shown that OTs can be detected using visible channel and infrared channel imagery ((Setvák et al., 2010); (K. M. Bedka & Khlopenkov, 2016), (K. Bedka, Murillo, Homeyer, Scarino, & Mersiovsky, 2018), (Kim et al., 2017), (Kim et al., 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 $7-9Kkm^{-1}$ as they ascend into the lower stratosphere ((Negri & 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. 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, Rabin, & Wang, 2007)) and InfraRed Window texture (IRW-texture) ((K. 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%. (K. M. Bedka & 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%.

1.2 Research Objectives

The specific objectives of this thesis are the following:

- To prepare GOES-16 imagery data to be input to the convolutional neural network (CNN) model using interpolation, image patching, and normalization techniques
- To prepare labelled image data to ensure that the CNN is simultaneously appropriately trained whilst also being relatively free from human bias
- To develop a convolutional neural network model to detect the OTs in the given input data.
- To validate the model performance on the test data
- To inspect and analyst different approaches to improve the performance of the model

2 | Data

Three main sources of data are used throughout this thesis. Both the visible and infrared (IR) imagery from the Advanced Baseline Imager (ABI) aboard the Geostationary Operational Environmental Satellite (GOES)-16 series of satellites is are used for the detection of OTs. Additionally observations from the geostationary lightning mapper (GLM) also aboard GOES-16 are used for validation.

2.1 Advanced Baseline Imager

The Advanced Baseline Imager (ABI) is a visible and IR radiometer aboard the GOES-16 weather satellite. GOES-16, sometimes called GOES-East, is located at $75.2°W$ above the equator, providing a field of view covering all of South America and the majority of North America. ABI has 16 channels operating at frequencies from visible light to thermal-IR. Whilst the majority of the channels have a resolution of 2km, over the continental United States this increases to 3km due to the azimuth angle. The ABI take images every five minutes.

2.1.1 Previous Generations of GOES

The ABI's refresh rate, spectral range, significantly improved signal to noise ratio and resolution makes it a huge improvement on previous generations of geostationary weather satellites. This observation is key when comparing models that were trained on GOES-14 or earlier to the models trained here, it is very difficult to disentangle performance of the model to improvements of the data without replicating the whole model and training on newly available data.

There is a much larger choice of channels with GOES-16 than previous generations of satellites. GOES-14 had 5 channels to choose from, GOES-16 has 16 channels as shown in Table (2.1)

Table 2.1: GOES-16 channels, their properties and names. The central wavelength is used because the actual wavelength is a range. The visible bands are useful indicators of OTs but only provide information in the daytime, in order to provide 24-hour detection capability, infrared channels must be used.

Band	Central Wavelength (μm)	Pixel Spacing (km)	Nickname	Classification
1	0.47	1	Blue	Visible
2	0.64	0.5	Red	Visible
3	0.865	1	Veggie	Near-infrared
4	1.378	2	Cirrus	Near-infrared
5	1.61	1	Snow/Ice	Near-infrared
6	2.25	2	Cloud Particle Size	Near-infrared
7	3.90	2	Shortwave Window	Infrared
8	6.19	2	Upper-level Tropospheric Water Vapour	Infrared
9	6.95	2	Mid-level Tropospheric Water Vapour	Near-infrared
10	7.34	2	Lower-level Tropospheric Water Vapour	Near-infrared
11	8.5	2	Cloud-Top Phase	Near-infrared
12	9.61	2	Ozone	Infrared
13	10.35	2	Clean Infrared Longwave Window	Infrared
14	11.2	2	Infrared Longwave Window	Infrared
15	12.3	2	Dirty Infrared Longwave Window	Infrared
16	13.3	2	CO_2 Longwave Infrared	Infrared

In this thesis the models were trained on GOES-16 data, on channels

2.1.2 Himawari-8 Imagery

Analagous to GOES-16 over the continental United States, there is also a geostationary satellite centred over Japan called the Advanced Himawari Imager (AHI) onboard Himawari-8. This satellite sensor has similar capabilities to GOES-16 and is the data source used by (Kim et al., 2017) and (Kim et al., 2018) for detecting overshooting tops. The area covered by Himawari-8 is conducive to single-cell thunderstorms so sees many OTs, it is therefore a good training data set for any OT detection model. GOES-16 will be used in this thesis for two reasons; the easier access to the data, and two, the presense of the Geostationary Lightning Mapper, which can provide useful validation.

2.1.3 Selection of ABI Channels and Channel Combinations

In order to have equal performance during both day and nighttime[1], a selection of longwave IR ABI channels are used for the detection and tracking of OTs. These channels consist of the LW clean and dirty windows at $10.8\mu m$ and $12.3\mu m$ respectively, and the upper and lower troposphere water vapour channels at $6.2\mu m$ and $7.3\mu m$ respectively. Across much of the literature, LW window IR brightness temperature is used for the detection of OTs, but even within the DCC this channel varies considerably with meteorology and latitude so does not generalise well, so it will not be used in this thesis to detect OT. However it is used for the optical flow calculation of the cloud motion field, which is used for the detection of the cloud anvil boundary.

---

[1] Visible channels do not work at night

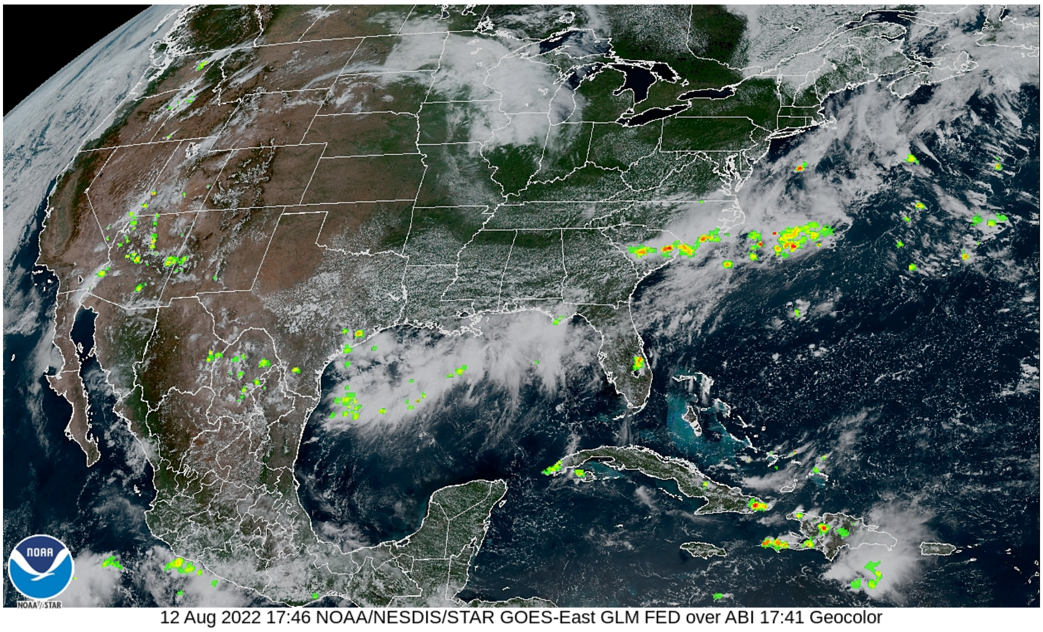
NOAA
NOAA/STAR
12 Aug 2022 17:46 NOAA/NESDIS/STAR GOES-East GLM FED over ABI 17:41 Geocolor

Figure 2.1: Geostationary Lightning Mapper, showing the lightning flashes across the continental United State.

2.1.4 Geostationary Lightning Mapper

The Geostationary Lightning Mapper (GLM) is another instrument on GOES-16. It is a single channel, optical transient detector that detects all forms of lightning day and night, with a high spatial resolution and detection efficiency.

In this thesis the GLM is used to validate the Deep Convective Cloud (DCC) detector, to enable the

The OT and non-OT reference data were constructed by manual inspection of the visible channel by human experts. OT can be identified in the visible channel by a couple of characteristic features; they have a 'cauliflower like' texture and produce shadows upon the anvil when illuminated at off-nadir angles. Visible identification is then potentially fraught with biases: OTs can be less obvious at solar noon then they can in low azimuth angle sunlight, and only the most pronounced OTs will protrude enough to show a cauliflower like shape.

Any OTs found by a human should therefore be considered to be *firm positives*. It is hoped that more OTs can be found by a deep learning solution compared to human labelling.

2.1.5 Troposphere Temperature

Troposphere temperature is a very useful measure that will be used in this thesis to eliminate some false positive detections. The process of converting the infrared radiance values into temperature is described by (Weinreb & Han, 2011) and written out below:

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

   $$R = \frac{X - b}{m} \tag{2.1}$$

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

2. Radiance is converted to effective temperature using the function

   $$T_{eff} = \frac{c_2 v}{\log\left(1 + \frac{c_1 + v^3}{R}\right)} \tag{2.2}$$

   where $c_1 = 1.191066 \times 10^{-5}$, $c_2 = 1.438833$ and $v = 936.20$.

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

   $$T = \alpha + \beta T_{eff} \tag{2.3}$$

   where $\alpha = -0.2875616$ and $\beta = 1.001258$

3 | Methodology

3.1 Different Available Architectures

The problem of object detection can lead to many different solutions proposed. Whether the end goal is to locate the OT exactly, or just identify its presence is one such design decision that must be made.

In the literature for detecting OT several architectures have been proposed. They all share many common features and only differ on the details. In this section I want to thoroughly review the architectures proposed by the academic community and investigate whether there are any alternative architectures available that

So far, methods proposed have focused around three main architectures:

1. Detection of OT using image processing (K. M. Bedka & Khlopenkov, 2016). This approach centres around the key observation that an OT cools down when it emerges above the tropopause. This technique only works for clear, protruding OTs in developed cumulonimbus anvils, this is only a subset of all OTs. Many OTs don't cool down sufficiently to be noticed by these methods and would go undetected.

2. Detection of OT using Convolutional Neural Networks. Particularly, two architectures proposed by (Kim et al., 2017) This technique has seen a couple of applications in recent years but still relies on the aforementioned temperature drop of the OT penetrating into the troposphere.

3. Detection of OT using a U-net. The U-net is really an extensions of the downsampling CNN, with the addition of an upsampling network to create high resolution outputs for the purpose of image segmentation. This ap-

proach has the benefit of being able to produce OT regions rather than point estimates.

Two satellite infrared-based overshooting convective cloud-top (OT) detection methods have recently been described in the literature: 1) the $11\ m$ infrared window channel texture (IRW texture) method, which uses IRW channel brightness temperature (BT) spatial gradients and thresholds, and 2) the water vapor minus IRW BT difference (WV-IRW BTD). Both of these methods have the problem that they are not capable of detecting OTs before they appear above the tropopause, and therefore the air parcel experiences the cooling that is visible in the IRW channel and the protrusion that is obvious in the visible spectrum. How can you detect something that has not happened yet? Overshooting tops do not occur in isolation. Surrounding the top is a divergent flow field and the presence of this divergent flow is evidence of the overshooting top that we seek to detect.

3.2 Convolutional Neural Network Theory

A CNN typically consists of a convolutional layer, pooling layer, and fully connected layer, and its performance depends on how those layers are arranged and how the network is trained. The convolutional layer and pooling layer are located at the front of the CNN and serve to extract image features. The convolutional layer uses a filter (kernel) to generate a convolved feature of an input image; the feature extraction is done as a kernel matrix (in this case, 3×3), which sweeps through the input image, multiplies the matrix corresponding to each element of the image, and then adds them together. This process transforms the image into a feature map. CNNs repeat this process to produce several features of the input image and learns based on these diverse feature maps rather than using only one dataset. Then, the pooling layer subsamples the result of the convolution operation by extracting the maximum stimulus (i.e., the maximum value) of a feature map. One of the greatest strengths of a CNN is its ability to extract features without prior knowledge or effort from the user. Using the extracted features after the convolutional and pooling layers, classification or regression proceeds in the fully connected layer.

3.2.1 Components of CNNs

3.2.1.1 Convolution Layers

Convolution leverages three important ideas that can help improve a machine learning system: sparse interactions, parameter sharing and equivariant representations. Sparse interactions is achieved by making the kernel smaller than the input. The kernel can be interpreted as learning the smaller, meaningful features such as edges that only occupy tens of pixels. Parameter sharing is a result of using the same kernel for the whole input image. In contrast to a dense network, where each parameter is used only once, in CNNs the same parameter is used at all input locations. This has the dual benefit of ensuring that the layer has a property called equivariance to translation and the network is quicker and easier to train.

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

In a convolutional layer, the network learns filters that extract specific features related to the spatial positions in input images. Each filter moves across the width and height of the input and produces a 2-dimensional feature map (a convolved

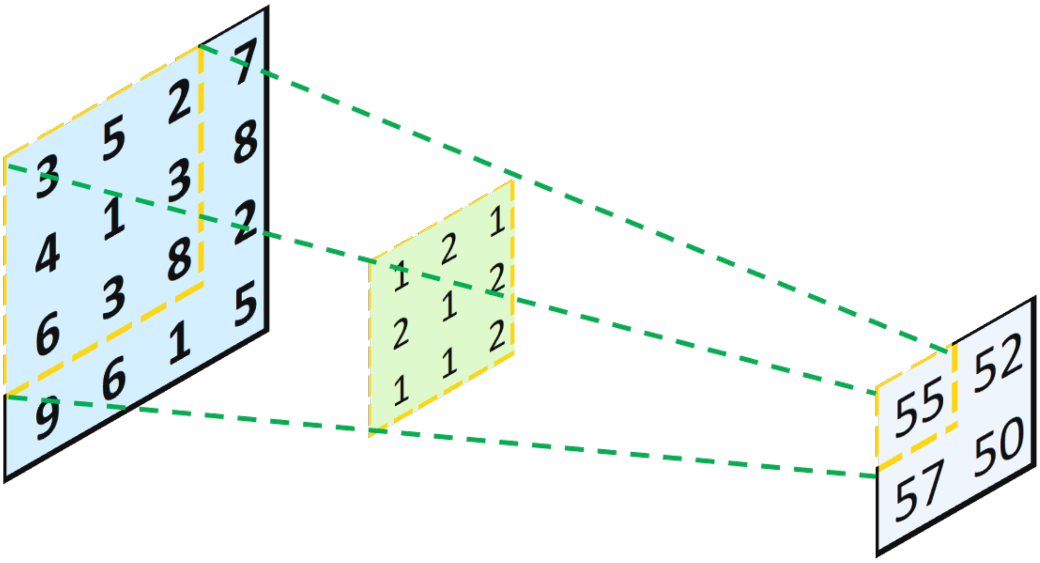

Figure 3.1: A convolutional layer uses a filter (kernel) to generate a convolved feature of the input image.

image) of that filter by computing the dot product between the entries of the filter and the input and passing an activation function . This process is the same as the way a neuron process information only for its receptive field. Under the assumption that the input image and the filter are both squares, this takes the following mathematical form:

$$h_{uv} = f((X * X)_{uv} + b) \tag{3.1}$$

$$(X * W)_{uv} = \sum_{m=0}^{T} \sum_{n=0}^{T} x_{u+m,v+n} w_{mn} \tag{3.2}$$

where $u, v = 1, ..., D$

$$D = \frac{I - T}{S} + 1 \tag{3.3}$$

where h_{uv} is a component of the u-th row and v-th column of feature h. I and T are the size of an input image X and a filter W, respectively that consists of x and w. The convolution operation, the bias term and the activation function are denoted by $*$, b and $f(\dot{})$ respectively. The generated feature map H has the size D that depends on the step size of the filter denoted by S. When another convolutional layer is added to the previous convolutional layer, the feature map in a new convolutional layer can be calculated as

$$H_j^n = f\Big(\sum_{k=1}^{K} H_k^{n-1} * W_{kj}^n + b_{kj}^n\Big) \tag{3.4}$$

where H_k^{n-1} and H_j^n are the k th input and j th output feature maps. K is the number of feature maps generated in he $(n-1)$ th layer. W_{kj}^n is the filter of the k th feature map in the $(n-1)$ th layer for the j th feature map in the n th layer, and b_{kj}^n is the corresponding bias term. Every entry in a produced feature map can be interpreted as a neuron that recognises a specific region in the input. Neurons in the same feature map share parameters because each feature map is created by one filter. The output of this layer is various feature maps generated from various filters. In this process, the number of convolutional filters, the size of these filters, and an activation function should be prefixed. The purpose of using an active function in a CNN is basically to introduce non-linearity int he network that calculates the linear operation (simply element-wise multiplication and summing) in the convolutional layer.

3.2.1.2 Pooling Layers

Pooling layers make the representation approximately invariant to small translations of the input. Invariance to translation means that if the input is translated by a small amount, the values of most of the pooled outputs are unchanged. This is a useful property in feature detection as it is much more important whether the object being detected is present then where it is.

A convolutional neural network can be imagined as similar to a fully connected network, but with an infinitely strong prior over its weights. An infinitely strong prior says that the weights for one hidden unit must be identical to the weights of its neighbour except shifted in space. The prior also says the weight must be zero, except for in the small, spatially contiguous receptive field assigned to that hidden unit. Overall, we can think of the use of convolution as introducing an infinitely strong prior probability distribution over the parameters of the layer. This prior says that the function the layer should learn *only* contains local interactions and is equivalent to translation.

By this reasoning, it is clear that convolution and pooling can result in underfitting. Like any prior, convolution and pooling are only useful when the assumptions made by the prior are reasonably accurate. If a task involves incorporating information from very distant locations in the input, then the prior imposed by convolution may be inappropriate.

This is why CNNs are appropriate for this task: overshooting tops are a local phenomenon, the only information needed to detect them is the immediate area around the OT, as this is the area that will give the informationt that is critical to the detection: the temperature difference, the 'cauliflower like' shape in the visible spectrum, and the divergent flow field

3.2.1.3 Activation Function

The purpose of the activation function is to make the neural network learn nonlinear 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. There are many activation functions available, with differing benefits to each.

The ReLU activation is a good choice because it does not saturate for positive values and is fast to compute, but it suffers from a problem known as *dying* ReLUs: during training, some neurons stop outputting anything other than 0, effectively dying. Using *leaky* ReLUs ensure that neurons never die, by providing a small gradient for $z < 0$ they can stay stagnant for a while but they always have a chance to eventually wake up, as there is always a non-zero gradient upon which to act.

$$\text{LeakyReLU}_{\alpha}(z) = \max(\alpha z, z) \tag{3.5}$$

The hyperparameter α defines how much the function leaks, typical values range from 0.01 to as high as 0.2. There are variants where α is randomised during training and then fixed at an average value for testing, this is known as randomised leaky ReLU or another variant where α is learned during training called the parametric leaky ReLU. Both of these variants show some improvement, especially over large datasets. It would be remiss to not mention the state-of-the-art in activation functions: the exponential linear unit (ELU) and Scaled ELU (SELU) activation functions. Both these outperform leaky ReLU.

However, when it comes to choosing an activation function it is not all about performance - ReLU is the most used activation function by far because of its simplicity. This leads to the dominant machine learning libraries having outstanding optimisation for ReLU and therefore despite the theoretical drawbacks of ReLU it still has very good practical performance.

3.2.1.4 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, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 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.

3.2.1.5 Faster Optimisers

Training deep neural networks can be very slow. A huge speed boost can be gained by using a faster optimiser than the regular Gradient Descent optimiser. Whilst many have been proposed: momentum optimiser, Nestorov Accelerated Gradient, AdaGrad, RMSProp. In this thesis the state-of-the-art shall be used, this is the Adam Optimiser. Adam, which stands for adaptive momentum estimation, combines the idea of two of its predecessors momentum optimisation and RMSProp: just like momentum optimisation, it keeps track of an exponentially decaying average of past gradients; and just like RMSProp, it keeps track of an exponentially decaying average of past squared gradients.

$$\mathbf{m} \leftarrow \beta_1 \mathbf{m} - (1 - \beta_1) \nabla_\theta J(\theta) \tag{3.6}$$

$$\mathbf{s} \leftarrow \beta_2 \mathbf{s} - (1 - \beta_2) \nabla_\theta J(\theta) \otimes \nabla_\theta J(\theta) \tag{3.7}$$

$$\hat{\mathbf{m}} \leftarrow \frac{\mathbf{m}}{1 - \beta_1^T} \tag{3.8}$$

$$\hat{\mathbf{s}} \leftarrow \frac{\mathbf{s}}{1 - \beta_1^T} \tag{3.9}$$

$$\theta \leftarrow \theta + \eta \hat{\mathbf{m}} \oslash \sqrt{\hat{\mathbf{s}} + \epsilon} \tag{3.10}$$

3.2.1.6 Contrast Normalisation

There is significant variation in the contrast of the images: this is the difference in the magnitude between the bright and dark pixels. Global contrast normalisation aims to prevent images from having varying amounts of contrast by subtracting the mean from each image, then rescaling it so that the standard deviation across its pixels is equal to some constant s. The problem with global contrast normalisation is it does not highlight edges and changes in contrast within an area, so in this particular scenario the minimum and maximum of the contrast will be dominated by clear sky and cloud tops respectively for most channels. This motivates local contrast normalisation, where contrast is normalised over each small window, rather than the image as a whole.

3.2.1.7 Batch Normalisation

Batch normalisation helps further reduce the danger of vanishing gradients. It involves adding an operation in the model just before or after the activation function of each hidden layer. The operation simply zero-centres and normalises each input, then scales and shifts the result using two new parameter vectors per layer: one for scaling, the other for shifting. This essentially lets the model learn the optimal scale and mean for each of the layer's inputs. Whilst most authors working in the field of detecting OTs do not use batch normalisation in their network architectures, it was hypothesised that in this case it might be a good tool to enable the training of deeper networks quicker.

$$\mu_B = \frac{1}{m_B} \sum_{i=1}^{m_B} x^{(i)} \tag{3.11}$$

$$\sigma_B^2 = \frac{1}{m_B} \sum_{i=1}^{m_B} (x^{(i)-\mu_B})^2 \tag{3.12}$$

$$\hat{x}^{(i)} = \frac{x^{(i)} - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{3.13}$$

$$z^{(i)} = \gamma \otimes \hat{x}^{(i)} + \beta \tag{3.14}$$

3.3 Processing of Data

3.3.1 Pre-processing

The details of the GOES-16 imager and the data it produces have already been outlined in Table (2.1) This study uses both visible and infrared images. Visible images of Channel 1 have a spatial resolution of 500m. The infrared image of Channel 9 has a spatial resolution of 2 km. The infrared images are used to get the brightness temperature, which is the temperature at the cloud top. OTs are usually colder than surrounding regions (Setvák et al., 2010). So, OTs are often isolated regions of cold brightness temperatures (BTs) relative to a warmer surrounding anvil cloud, this will be a key feature that is used in distinguishing them. GOES-16 imagery is collected every minute for the infrared and visible images.

The data used in this study for training is all taken on 25th May 2015, with a total of 259 GOES-16 images available at this time. Each image has a size of 1300×2400, with infrared images being half the size of visible images. All images are taken at nadir.

3.3.1.1 Bicubic Downscaling

In order to ensure that the inputs are the same size the infrared image is bicubically upsampled to be the same dimension as the visible image. This involves interpolating every second pixel to generate a higher quality image. Obviously the information contained in the image is preserved by this approach.

Now both the visible and the infrared images have the same dimension, so both in effect have 1km pixels. The best input to a deep neural network is a high dimensional one, as the image dimension is constantly reduced by the process of pooling and convolution, therefore in order to ensure that a high dimensional image is feed through the network, another downscaling operation is performed on both images again by a factor of two this time to 500m resolution. The interpolation performed here was bicubic interpolation (Sermanet et al., 2013) based on the 16 (4×4 grid) pixels surrounding the target pixel to interpolate. This leads to a smoother image.

3.3.1.2 Contrast Normalisation

The images were then globally normalised to a min-max scale so that all the images had intensities in the range $0 - 1$.

3.3.1.3 Splitting into Patches

Both images, which is now of resolution 2600×4800, is split into patches of size 31×31. This patch size is chosen as it optimally represents the distance between overshooting tops whilst containing all the information required to detect one. OTs are not points and have a sometimes significant spatial extent. They can

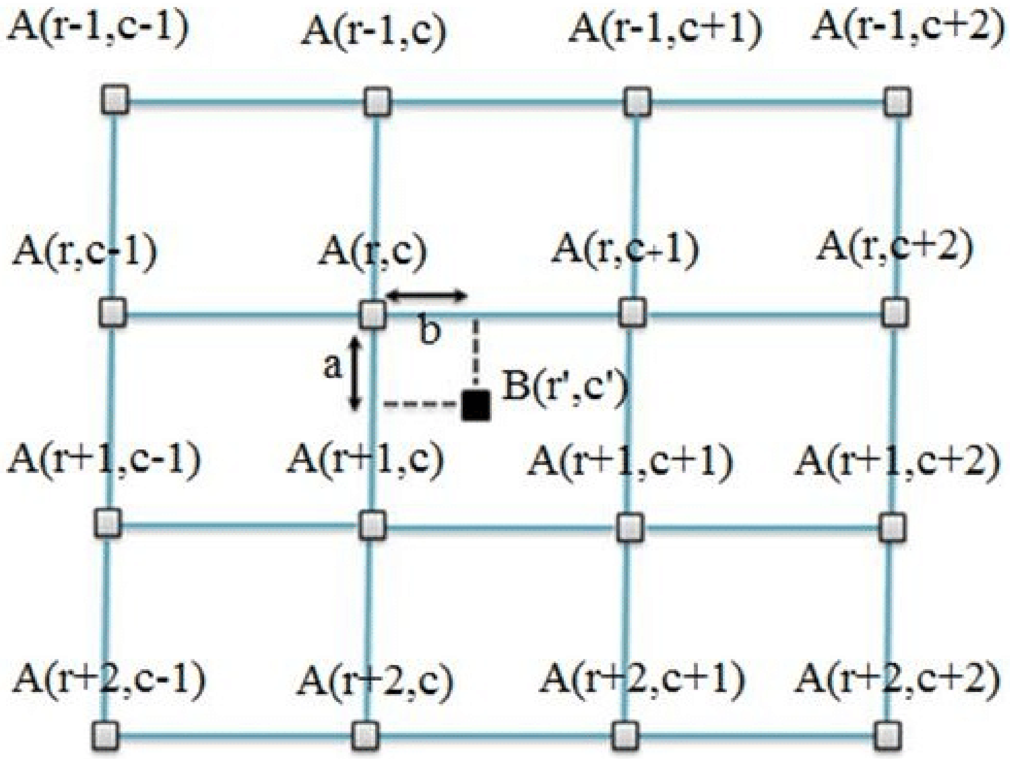

Figure 3.2: Bicubic interpolation of the nearest 16 pixels.

be anywhere between 5-10km in diameter (K. Bedka et al., 2010). A patch size of 31×31 is approximately 15.5km in width and height [1], this means that even the biggest OTs are fully contained within a patch. Why not make the patch size bigger? This would be useful to contain more context around the OT and potentially help detection, after all, it is the distinctive cauliflower-shape in the visible spectrum and the temperature difference in the infrared spectrum that make the OT detectable in the first place. Both these features require the background to make them features, you cannot detect an OT based on temperature without comparing the temperature to the anvil cloud. Because we are not hoping to determine where in the patch the OT is, only that it is present. Therefore you can effectively treat these 31×31 patches as our 'pixels' they are the smallest resolution to which we locate the OTs. In addition, OTs do not occur closer than 16km, so there is a guarantee that there will not be more than one OT per patch.

The stride of these patches is 31, it is important not to confuse these patch sizes with the kernel size later on in the thesis.

3.3.2 Post-processing

Two post-processing steps were included in the CNN-based model to improve OT detection. First, patches detected as OTs with outside of the Deep Convective Cloud (DCC) anvil were eliminated to reduce potential misclassifications. A methodology using the algorithm outlined in (Jones, Christensen, & Stier, 2022) was used to identify the cloud anvil and eliminate these misclassifcations. This is because in a patch caught by cloud edges, a spatial difference between the anvil and the cloudless sky surrounding it could be misinterpreted as the same kind of difference as between an OT and the surrounding anvil. Therefore, OT detection

---

[1] On average, as the patch size changes with longitude, patches closer to the equator are smaller than those further north.

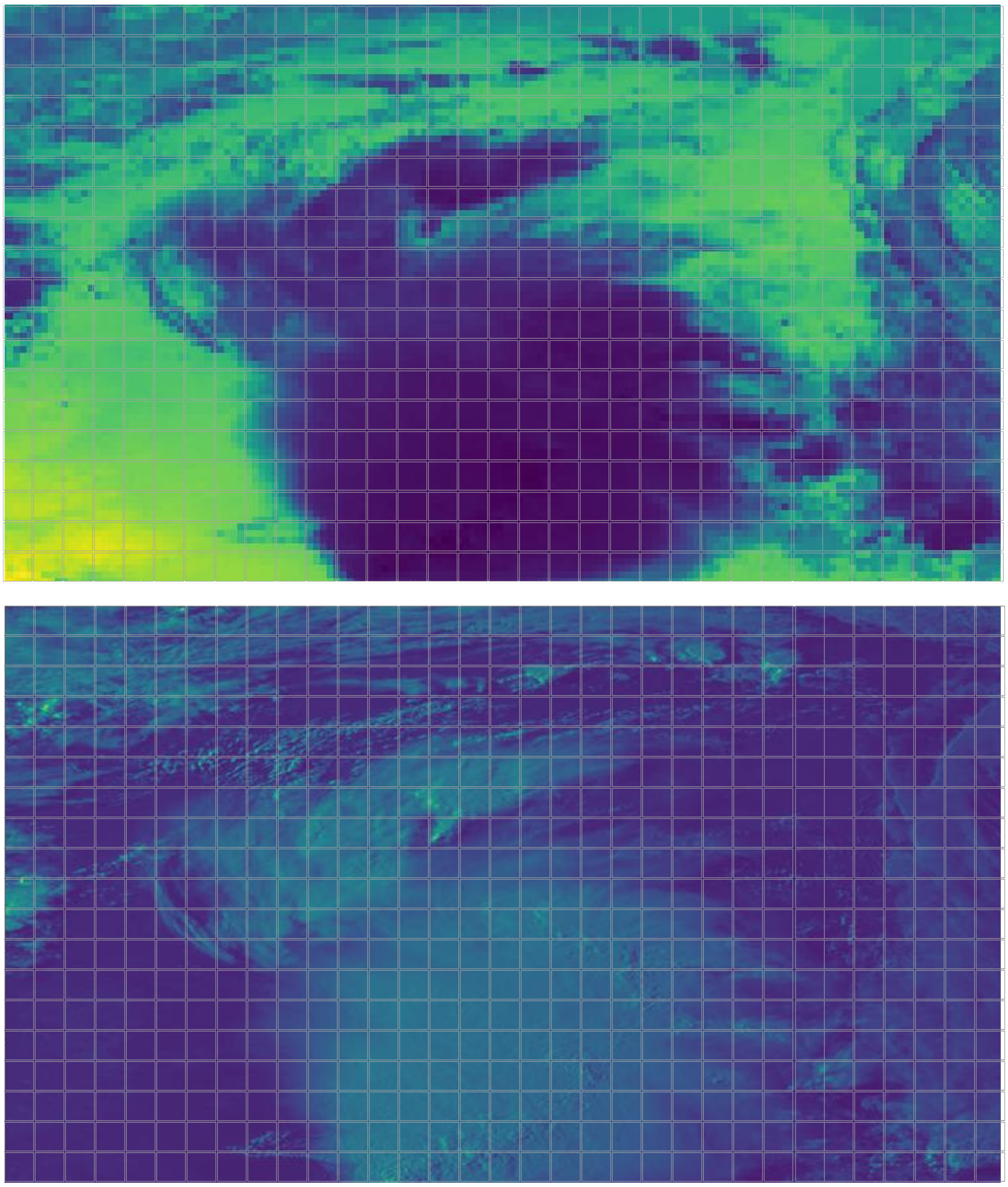

Figure 3.3: The visible and IR images split into patches. The actual size of the grid is much much smaller, each image generates 13884 patches.

results can be bettered by ensuring that detections are only valid if they occur in the anvil. Different thesholds of detection could be played around with here, but it was decided in the interest of time to use the thresholds proposed by Jones.

In the second post-processing step, to clearly show OT regions within patches, the pixels with relatively low brightness temperatures in OT patches were extracted. Adjacent patches were bundled together at first. Then, pixels with brightness temperatures that were lower than the sum of the minimum value (min) of brightness temperatures in bundled patches plus the standard deviation (σ) of brightness temperatures in bundled patches times the scaling factor (λ) (i.e., pixels $< \text{min} + \sigma\lambda$) were found. The optimal value of the scaling factor was determined using calibration data.

3.3.3 Validation

As a performance measure, the probability of detection (POD) and false alarm ratio (FAR) were used. Where TP is True positive, FP is false positive and FN is false negatives,

$$POD = \frac{TP}{TP + FN} \tag{3.15}$$

$$FAR = \frac{FP}{TP + FP} \tag{3.16}$$

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.

3.4 Labelling of Data

3.4.1 Manually Labelled

The OT and non-OT reference data were constructed by manual inspection of the visible channel by human experts. OT can be identified in the visible channel by a couple of characteristic features; they have a 'cauliflower like' texture and produce shadows upon the anvil when illuminated at off-nadir angles. Visible identification is then potentially fraught with biases: OTs can be less obvious at solar noon then they can in low azimuth angle sunlight, and only the most pronounced OTs will protrude enough to show a cauliflower like shape.

The images are chopped into patches of size $31x31$. 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.

Of the 459 images that have OT labels in Bedka's labelled dataset, 40 were randomly selected for testing. These images would then go through the same process as the training images in being chopped into 31×31 patches. They are given a binary label for whether or not they contain an OT.

3.4.2 Algorithmically Labelled

Given the relative scarcity of labelled OT data, a good way to vastly increase the number of labelled instances is by following a process based approach to labelling, one that can be performed automatically.

Another problem with the labelled dataset above is that it treats OT as points. In reality every OT has a spatial extent and therefore a more accurate model would treat the OTs as occupying an region, therefore the images should be labelled with OT regions rather than OT points. Points would be a reasonable approximation if distance from the OT is a good metric for success, however if the model detects an OT within the cloud anvil, that is a lot less wrong than an OT being detected outside of the cloud anvil. Equally due to the way in which Bedka's dataset was created, it does not encourage the algorithm to learn what an early stage OT looks like, it only allows the algorithm to learn what a human-detectable mature OT looks like. By learning that

This process works on the Brightness Temperature of the cloud top, and follows the following process, as outlined by (Zha et al., 2020):

1. Segmenting the infrared window brightness temperature (IRW BT) cloud image into 31×31 patches (same patch size as before)
2. Finding the minimum brightness temperature T^i_{min} of the ith patch.
3. Determining the initial candidate OT pixels $T^i_{H_{OT(n)}}$ which satisfy both $T^i_{H_{OT(n)}} < T^i_{\text{min}} + 4$ and $T^i_{H_{OT(n)}} < 215$, where $n = 1, 2 ... N$.
4. Determine the anvil-shaped cloud pixels $T^i_{Z_{OT(n)}}$. Here we differ from the method proposed by (Zha et al., 2020) and use the methodology to detect Deep Convective Clouds (DCCs) produced by (Jones et al., 2022), with this algorithm demonstrated in Fig. (3.4).
5. With each candidate OT pixel, check that it is within a DCC as detected by Jones' algorithm. If it is not within the DCC, then it must be in clear sky and therefore is a false alarm. If it is within a DCC, then it is likely an OT.
6. Repeat the steps above to find all the candidate OT pixels, and merge this

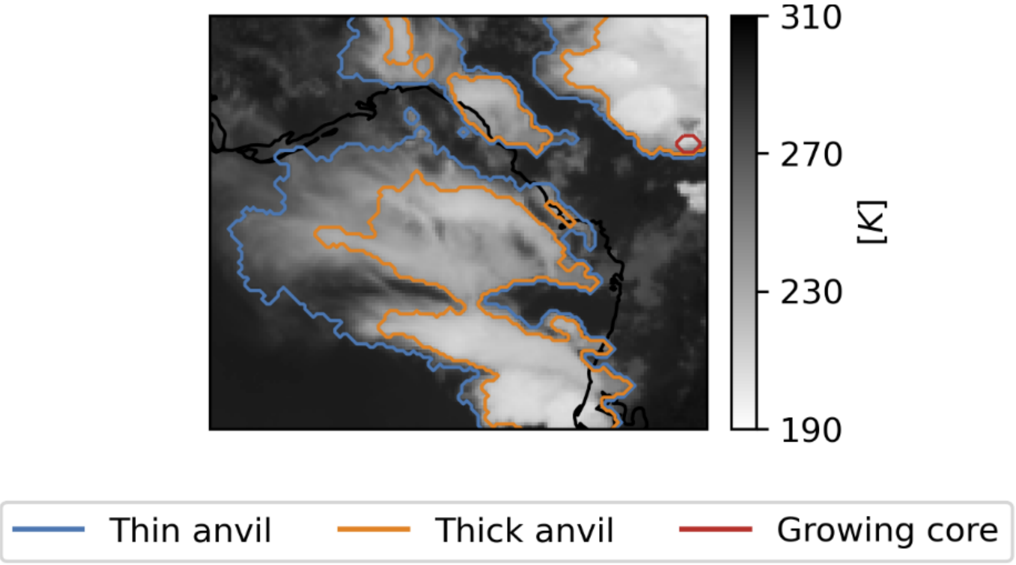

Figure 3.4: Example output from (Jones et al., 2022) algorithm to detect the growing and developed anvil of a Deep Convective Cloud (DCC).

pixels if they are adjacent to form OT regions.

7. Ideally every OT region would then be confirmed by a human expert, however due to time constraints this was not possible. Adding this step in would greatly decrease the number of false alarm OTs but would also add in a human bias to the labelling and limit the number of labelled instances that would be able to be produced.

The purpose of the algorithmic labelling is another angle to look at whether the learned CNN really is learning the ground truth, and whether the OTs identified in Bedka's dataset are complete.

3.5 Convolutional Neural Network Architecture

The architecture used for this CNN is below, this architecture has been used before by (Kim et al., 2018), but with less advanced filtering of clear sky detections.

Table 3.1: Model Architecture

Layer	Type	Maps	Size	Kernel Size	Stride	Padding	Activation
In	Input	3	31×31	-	-	-	-
C1	Convolution	32	29×29	3×3	1	same	Leaky ReLU
C2	Convolution	32	27×27	3×3	1	same	ReLU
S3	Max Pooling	32	13×13	3×3	1	same	-
C4	Convolution	64	11×11	3×3	1	same	ReLU
C5	Convolution	64	9×9	3×3	1	same	ReLU
S6	Max Pooling	64	4×4	3×3	1	same	-
F7	Fully connected	-	1024	-	-	-	-
F8	Fully connected	-	256	-	-	-	-
Out	Fully connected	-	2	-	-	-	Softmax

3.5.1 Data Augmentation

Previous studies (Yu, Wu, Luo, & Ren, 2017), (Perez & Wang, 2017) and (Inoue, 2018) have proposed several data augmentation techniques to overcome limited samples or create diverse data set with variants. For each OT example we have in the training set, we created three copies of it with the following characteristics essentially oversampling the minority class. For training data, we extracted three different categories of patches for each OT occurrence, they are listed below:

1. An OT center as the center or close to center of the 31×31 patch
2. Offset images with the centre of the OT slightly shifted from the geometric centre of the 31×31 patch
3. Peripheral images with the centre of the OT within 2km of the edge of the 31×31 window.

This approach has the twofold advantage of increasing the number of positive OT examples in the training data and also increasing the robustness of the model as in the test data there is no guarantee that a chosen patch will contain an OT near the

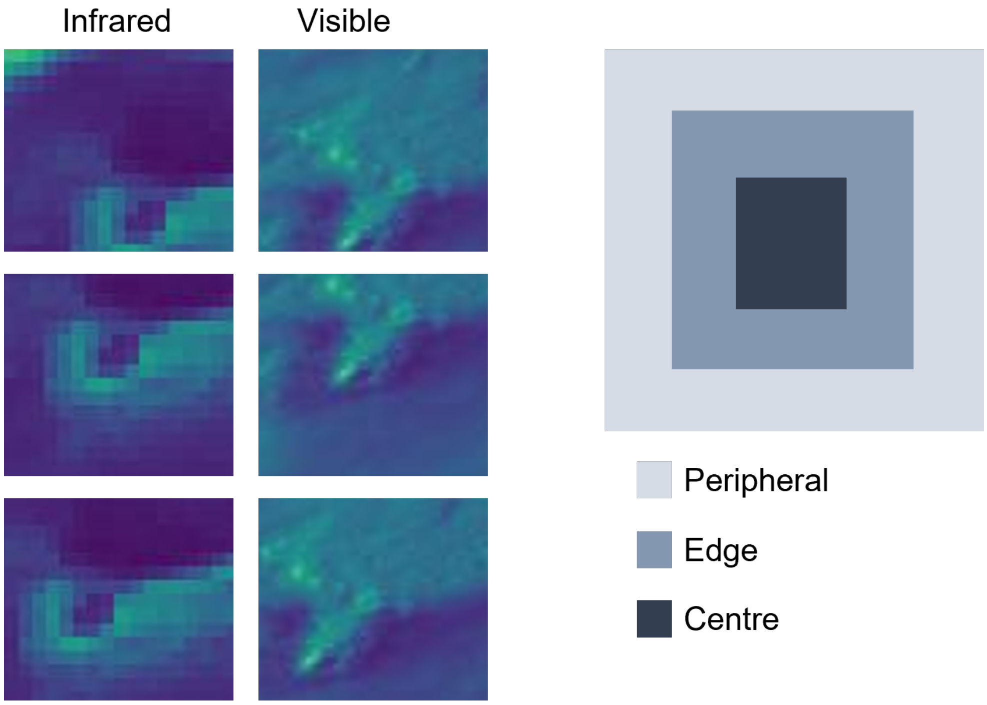

Figure 3.5: Data Augmentation was performed on the manually labelled data to increase the number of positive OT examples in the training data by approximately threefold. This was done by translating each patch in which an OT occurs to be either in the centre, peripheral or edge. This also increases the robustness of the model in testing to off-centre OTs.

centre of the image.

3.5.2 Training

The computational work was done by the servers based in (CUDA, n.d.), the data was stored on Google Cloud and downloaded from the NOAA Class website.

3.5.3 Testing

Of the 459 images that have OT labels in Bedka's labelled dataset, 40 were randomly selected for testing. These images would then go through the same process as the training images in being chopped into 31×31 patches. They are given a binary label for whether or not they contain an OT.

4 | Results

4.1 Building the Labelled Dataset

There is an existing labelled dataset available, one produced by Christopher Bedka. This dataset is available on satellite imagery from the 25th May 2015, a day where over the continental United States there were plenty of overshooting tops to identify. In total there are 3407 OTs detected in the dataset, across 459 images. For a particular snapshot of what one of the images in the labelled dataset looks like, see Fig. (4.1).

Before even beginning training the CNN on this dataset, the same images were applied to the DCC tracker to draw the outlines of the DCCs. The DCC detector was run independently of the OT region detector, and also independently of Bedka's labelled OT point dataset then the three were brought together as can be seen in Fig (4.2). This figure is really interesting because three independent methods show remarkable agreement on both OT locations and how these OT are constrained within DCCs (the latter conclusion is less of a surprise). By demonstrating alignment in particular between Bedka's labelled dataset and the OT region - 63% of Bedka's OT points were within the regions, the methodology of the OT regional assignment has been validated. Moreover, just 11% of the area contained within the anvil cloud of the DCC was given a OT region label, indicating that the algorithm was not succeeding to contain OT points by occupying as large an area as possible. Just 0.85% of the labelled OT points were outside of a DCC outline.

Investigating the cases where the OT points were outside of the DCC outline further it was found that these often occurred on the edge of the thunderstorm, where the three channels used to calculate the DCC outline were poorly correlated. Of-

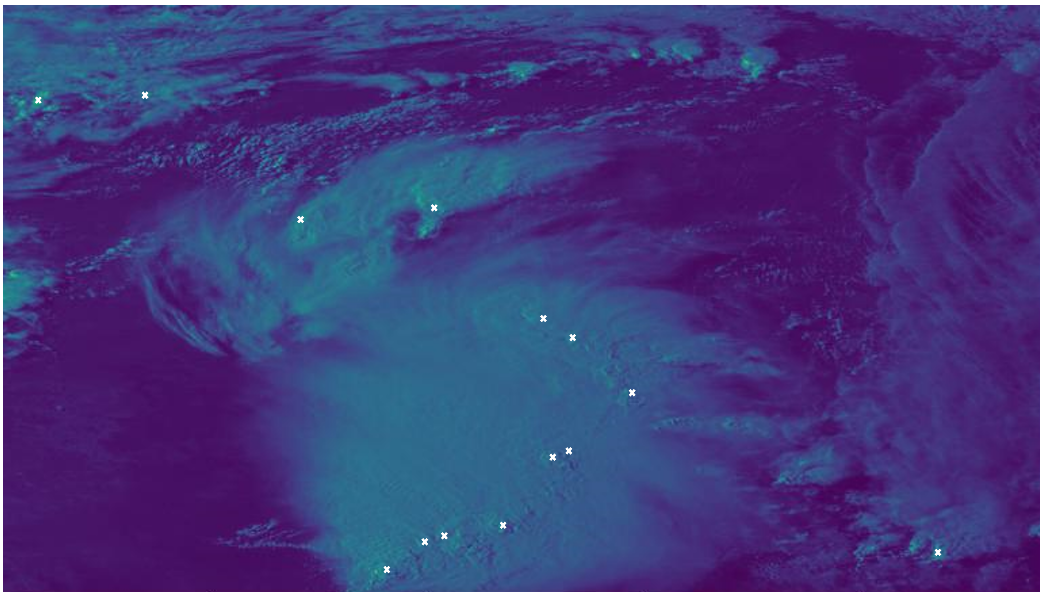

Figure 4.1: Manually labelled OT, using Kristopher Bedka's dataset.

ten, one channel can show a change in pixel intensity whilst the others can remain constant. This appears to happen particularly during the growing stage of the DCC. However, this is a computer science thesis and not a physics one, so this problem was not explored further.

Having assembled the OT regions, this could then be used to validate directly the OT candidates produced by the CNN, rather than comparing directly with the OT points. The way this works is by a simple point-in-polygon approach, which is somewhat unusual in computer vision, as normally it's matching point to point or area to area. The area to area comparison would be a suitable extension but would require a completely different architecture to resolve, involving a U-net which would produce a high resolution regional outputs.

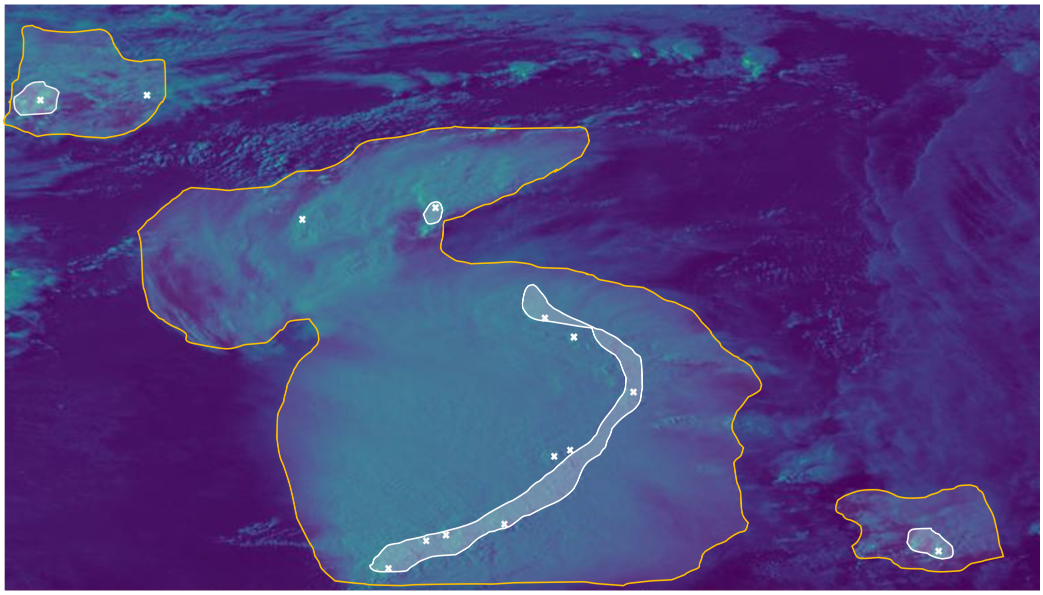

Figure 4.2: Using the DCC outline detector, the automatic labelling of OT regions was made possible, particularly by discounting areas that were not within the DCC.

4.1.1 Plotting the Polygons

A note on how these polygons were plotted and how the closed shapes are produced. The algorithm that detects the edge of DCCs does exactly that, detects the edge. Therefore there is no guarantee that a closed shape will be produced. Similarly, for the algorithm producing the OT region there is also no guarantee of a closed shape. Ordinarily in image segmentation this is not an issue as a closed shape is produced through the appending of pixels, such that the image segment is accreted. This works well when every pixel satisfies the condition for being part of the image segment, but not so well when only the edges of an image segment satistfy the conditions as is the case here. Therefore if a closed shape was not formed by the line segments produced by the algorithms for detecting DCCs and OT regions, then the lines were buffered until the smallest enclosed shape could be found.

The buffering and then un-buffering, essentially simplifying the polygons, also helps in smoothing out the shapes.

Additionally, in order to ensure that polygons did not become open because they reach the image edge, when a line segment reaches the edge of an image the line segment is connected to another line segment that results in the smallest possible enclosed area. The principle motivation being that the area covered by DCCs is much smaller than the area covered by other cloud types and clear sky. Therefore completing polygons using an area minimisation approach is valid.

4.2 Training the CNN

The CNN, with the architecture described in the methodology was trained on the labelled OT dataset. All the experiments were performed on an Intel Core i7-7700 CPU at 3.5 GHz with 64-GB RAM. The weights were initialized with random values, the batch size was to 5, the learning rate was set to 0.0001 and 2000 epochs were trained on 11G NVIDIA GeForce GTX 1080Ti GPU, with Stochastic Gradient Descent (SGD) as optimizer.

4.2.1 First successful CNN iteration

Several months of coding, testing, failing, and then repeating finally led to some meaningful results, the first of which is shown in Fig. (4.3). This figure demonstrates a key problem with OT detection: there are very few OTs but many, many patches of clear sky and clear cloud. This means that the algorithm has a difficult time finding enough training examples to really understand what to look for in identifying an OT. Therefore to avoid being punished by a high error, it simply predicts many OTs, in the hope that some of them will be close to the ground truth, knowing that it is not punished relatively as much for false positives as it is

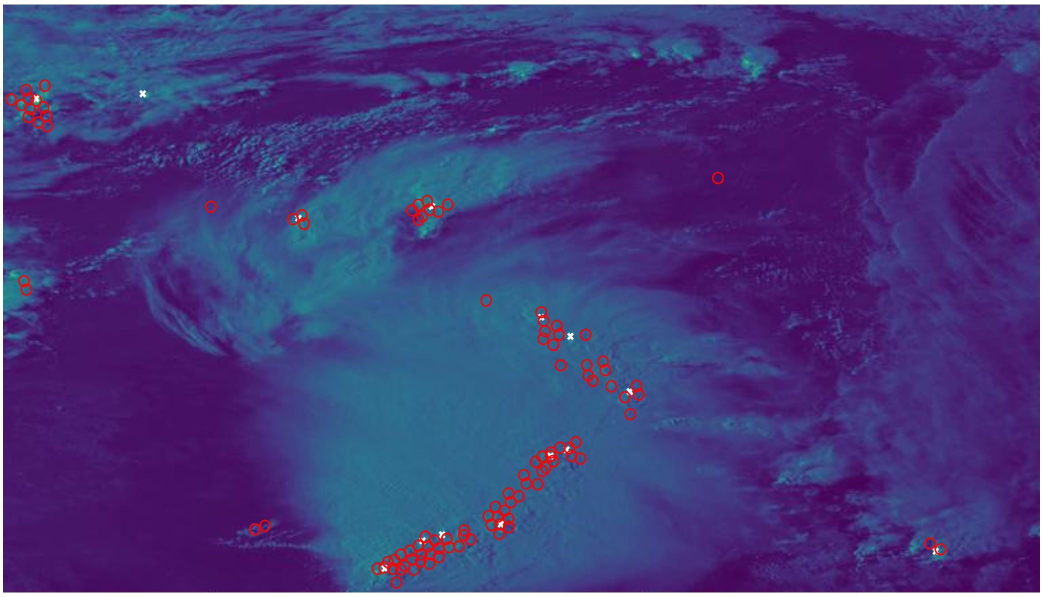

Figure 4.3: Comparison of the output of the first iteration of the model with the labelled (red circles) with the labelled ground truth (white crosses)

for true negatives. This approach is clearly not desirable. In this case whilst the probability of detection is quite high, at 70% the false alarm ratio is also unacceptably high at 90% showing that clearly the great majority of the proposed OT detections are mistakes.

Couple of approaches were implemented to address this and improve the performance of the model further:

4.2.1.1 Increasing the OT occurrences

Increasing the number of instances of the minority class (a positive OT detection), so that they appear in the ratio 1:2 in the training set. This is done by selectively sampling the positive OT detections much stronger than the negative OT detection. Obviously the test set remains unchanged, but by massively increasing the number of examples the model sees then it is able to better learn OT detection.

4.2.1.2 Increasing the near-OT occurrences

The majority of the false positives were actually near a real positive, indicating an inability of the model to distinguish being near an OT and being on an OT. Whilst this might be slightly symptomatic of the fact that OT have a spatial extent and are not points. For the purpose of the training it was decided to increase the number of patches that are near an OT but not contain an OT.

4.2.1.3 Increasing the cost of misclassifying

The model currently produces a high FAR because the cost of predicting incorrectly is not high enough, so as a trade off in order to get as many of the OTs as possible it predicts many OTs are occurring. In a simple binary classification process such as this the way to solve this is to make the error from predicting the majority class (OT) as the minority class (non-OT) higher than the inverse. By changing the cost function the weights will be updated through the whole model through backpropagation. The cost of misclassifying OT reference as non-OT is n times the inverse, this was a hyperparameter that was learned and after performing a grid search on ranges $2 - 20$ the optimum value was found to be 12.

4.2.1.4 Local Normalisation

Global normalisation was already performed on the images, but locally there was significant variations in intensity of signal. Due to the global normalisation the signal was almost binary: either inside a cloud or outside, which made it very difficult to get sufficient signal when inside the cloud to identify OT. Local normalisation is very simple, within each patch, perform the same max-min normalisation as was done before for the whole image. This increases the contrast between the OT and the rest of the anvil, improving performance.

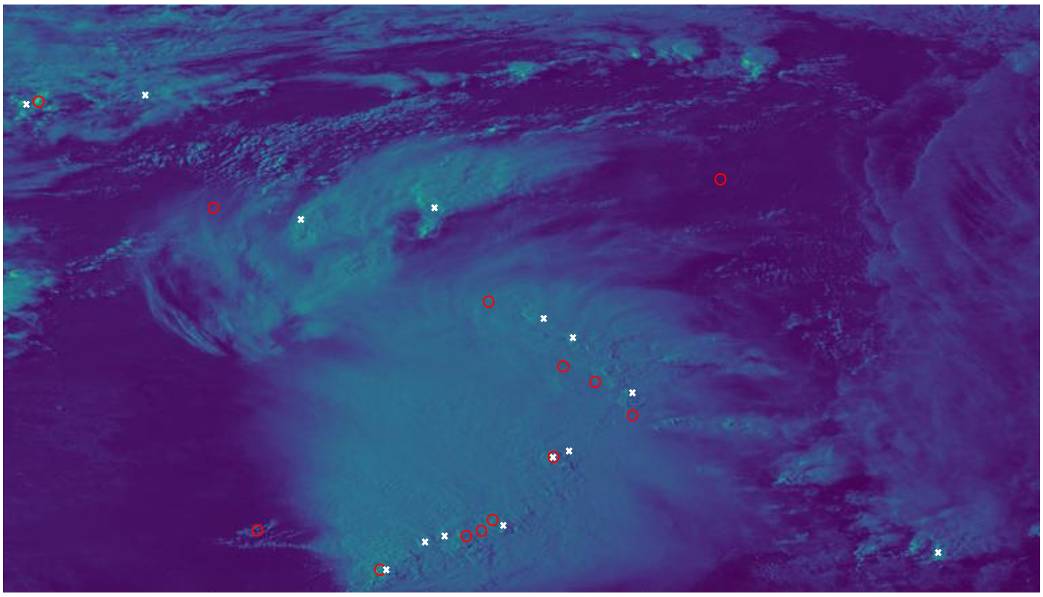

Figure 4.4: Comparison of the output of the second iteration of the model with the labelled (red circles) with the labelled ground truth (white crosses)

4.2.1.5 DCC clipping

Any detections that fell outside the bounds of the DCC were also removed. This reduced the tendency of the model to predict OTs on the edge of clouds or on non-convective clouds in clear sky, as these clouds have a similar infrared difference to an OT within the anvil cloud.

4.2.2 Improved CNN

These improvements were implemented and there was a substantial boost in the precision of the detections made by the CNN, see Fig. (4.4).

The improved CNN showed a similar performance at detecting the OT but with many fewer false positives, as was hoped by the improvements specified. The model appears to have really learnt the relevant features of an OT. Nonetheless, the false alarm rate of 86.8% might appear quite high, the reasons for this high

FAR will be analysed in the next chapter.

A lot of the false negative detections appear to be near-misses: adjacent to an OT patch. Therefore a more leniant criteria was developed that checks if the detection is within the OT region, as created earlier. Almost all (80%) of OT predictions were in the OT region, indicating that with slightly looser and more reasonable criteria for what qualifies as a match between prediction and observation, there is much greater agreement.

4.3 Overall Results

The total number of actual OT references in the test data is 406. Table (4.1) summarises the number the POD, FAR and CSI for the approach presented in this thesis and comparable approaches from other authors.

Table 4.1: Overall performances for the different models used and comparisons to models in existing literature

Num	Approach	POD (%)	FAR (%)
1	CNN with DCC clear sky elimination	81.2	86.8
2	IRW-texture candidates and detection criteria	35.1	24.9
3	OT probability	69.2	18.2
4	OT probability with visible rating detection	51.4	1.6
5	CNN with tropopause temperature processing	72.66	90.44
6	CNN with clear sky processing	79.68	9.78

5 | Discussion

The model fails to detect around 20% of overshooting tops, and in this chapter the reasons behind this are investigated.

5.1 The Problem with Patches

The patches are the greatest drawback the model has. The model has to learn to identify an overshooting top whether it is in the middle, near the edge or on the very edge of each patch, and it is only given the single patch to identify this. This is the equivalent of expecting a object classification model to be able to tell apart animals from half the face. It's hard.

It is very rare that an OT is in the centre of the patch, in fact over 50% of the OTs are located on the periphery of the patches. In the extreme case where the OT is just on the edge of the patch, the model essentially has to guess which of the two adjacent patches the OT falls into. Even with a perfect model, this leads to a naturally high false alarm rate.

It is found that when an OT is not detected (a false negative), the majority of the time a neighbouring patch was identified as having an OT. Demonstrating that, in fact the model is learning for the signs of OT, it just struggles with finding the exact centre.

Therefore the more lenient validation approach proposed using the OT region makes even more sense, if it is found that the image of an OT effectively bleeds across multiple patches, it seems unfair to penalise the model for identifying those neighbouring patches as having OT-like features.

5.2 Early Detection

In some cases the OT identified by the model, classified as false negatives, were in fact early detections of OTs. Early, in this context, means before the human identified them as an OT. This can be easily verified by checking through time and confirming that the same patch did eventually become an OT in a later image, within a time range of 10-15 minutes.

This was a minor contributor for the high FAR however as inevitably the model ended up adopting human standards for what an OT feature looks like, which means that early detection was somewhat stunted. If a Recurrent Neural Network were applied to the data then there is the possibility of enhancing the early detection aspect of the model.

6 | Conclusion

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.

6.1 Conclusions

Overshooting tops are an important weather phenomenon. They give an insight as to what is happening within a cloud that is completely opaque to satellite imagery. As the correlate so strongly with severe weather, the detection of overshooting tops is therefore a important mission and target for computer vision. Early, accurate detection could lead to warning systems being put in place in areas where weather radar is not established. This thesis shows that convolutional neural networks are a very appropriate tool for detecting overshooting tops. There are some unique challenges faced by this application. One of the most difficult is the problem of filtering out 'clear sky detections'. This is because the change in temperature between a cloud and clear sky is similar to that of the change in temperature between an overshooting top and its anvil. This makes the filtering out of these detections very important to ensuring accurate detections. In this thesis this was done by a deep convective cloud identification and segmentation algorithm that filtered out any detections made outside of the deep convective cloud, which is where all the overshooting tops occur. This led to a slight improvement on other convolutional neural network based models. The other insight was to adopt a region based approach to overshooting top labelling. This enabled a more reasonable validation of the performance of the model and it was not penalised as heavily for near miss detections.

6.2 Future Work

This thesis leaves a number of questions unanswered and work remaining to be done, among these are:

1. It is possible to infer the physical flow field from the visible image and use this to detect for divergent flow around the overshooting tops, which would be a completely novel way of detecting them.
2. The use of visible images to detect OTs is effective because of the nature of the CNN, which can extract representative image features particularly well. However, since visible images are only available during daytime, this model is limited, in that it cannot work at night. In order to detect OTs during nighttime, it would be necessary to develop a CNN model that uses infrared images only.
3. GOES-16 images form a sequential run of data. In order to improve detection a sequence of images could be feed into a CNN, or a variant of the CNN the Recurrent Neural Network (RNN). This would allow the network to understand the temporal relations of the movement of OT and lead to applications of tracking, forecasting and earlier detection.
4. Weather radar is the gold standard of rainfall tracking. By using radar systems well located across Europe and the United States, an OT tracker could feasibly be trained by radar systems with only the visible and infrared channels as inputs. The reason this would work is because the majority of the false positives obtained by the OT detection algorithms are found at the edge of clouds, where the temperature change looks similar to the model as at the centre. However rainfall occurs within clouds and most strongly underneath OTs. An example of weather radar within clouds is shown in Fig. (6.1)

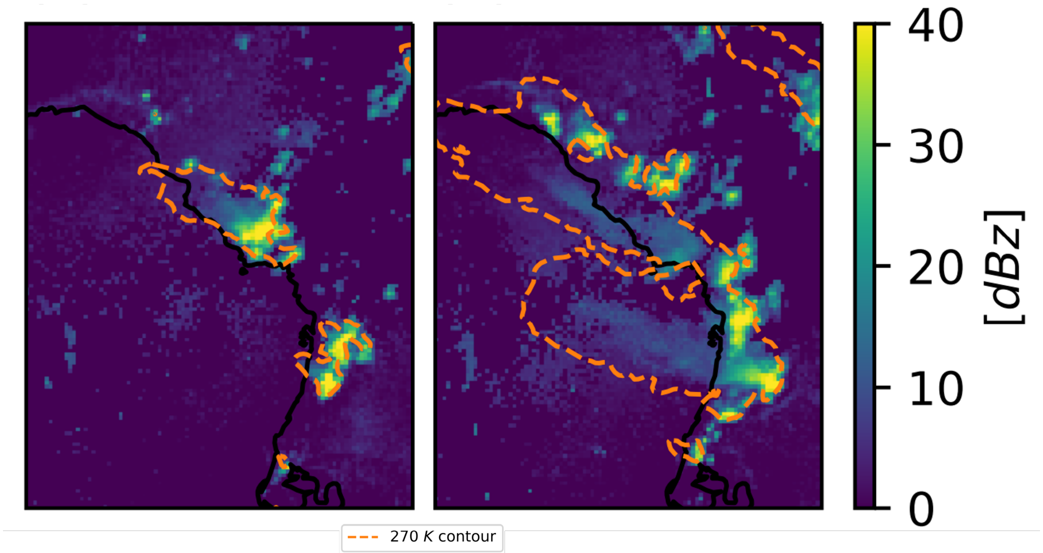

Figure 6.1: Weather radar, used to detect rainfall, is one feature that strongly correlates with where OTs occur. Therefore using radar systems to validate the detection of OTs would be a good avenue for future work (Jones et al., 2022). The 270K contour roughly indicates the edge of the cloud.

5. The work done on algorithmically labelling the OT region naturally leads to applications from high resolution output CNNs, such as U-nets. By outputting regions of detection at a pixel level rather than categorising large. This would have the other benefit of removing the need for patches, which is one the biggest limitations of the current model.

Bibliography

Bedka, K., Brunner, J., Dworak, R., Feltz, W., Otkin, J., & Greenwald, T. (2010). Objective satellite-based detection of overshooting tops using infrared window channel brightness temperature gradients. *Journal of Applied Meteorology and Climatology*, *49*, 181 - 202. Retrieved from `https://journals.ametsoc.org/view/journals/apme/49/2/2009jamc2286.1.xml` doi: 10.1175/2009JAMC2286.1

Bedka, K., Murillo, E. M., Homeyer, C. R., Scarino, B., & Mersiovsky, H. (2018). The above-anvil cirrus plume: An important severe weather indicator in visible and infrared satellite imagery. *Weather and Forecasting*, *33*, 1159 - 1181. Retrieved from `https://journals.ametsoc.org/view/journals/wefo/33/5/waf-d-18-0040_1.xml` doi: 10.1175/WAF-D-18-0040.1

Bedka, K. M., & Khlopenkov, K. (2016). A probabilistic multispectral pattern recognition method for detection of overshooting cloud tops using passive satellite imager observations. *Journal of Applied Meteorology and Climatology*, *55*, 1983 - 2005. Retrieved from `https://journals.ametsoc.org/view/journals/apme/55/9/jamc-d-15-0249.1.xml` doi: 10.1175/JAMC-D-15-0249.1

CUDA. (n.d.). *Jasmin: The uk's data analysis facility for environmental science.* `https://jasmin.ac.uk/about/`.

Homeyer, C., & Kumjian, M. (2015). Microphysical characteristics of overshooting convection from polarimetric radar observations. *Journal Of The Atmospheric Sciences*, *72*, 870-891.

Inoue, H. (2018). *Data augmentation by pairing samples for images clas-*

sification. arXiv. Retrieved from https://arxiv.org/abs/1801.02929 doi: 10.48550/ARXIV.1801.02929

Jones, W. K., Christensen, M. W., & Stier, P. (2022, February). *A semi-lagrangian method for detecting and tracking deep convective clouds in geostationary satellite observations.* doi: https://doi.org/10.5194/amt-2022-31

Kanneganti, G. T. (2020).

Kim, M., Im, J., Park, H., Park, S., Lee, M.-I., & Ahn, M.-H. (2017). Detection of tropical overshooting cloud tops using himawari-8 imagery. *Remote Sensing*, *9*. Retrieved from https://www.mdpi.com/2072-4292/9/7/685 doi: 10.3390/rs9070685

Kim, M., Lee, J., & Im, J. (2018). Deep learning-based monitoring of overshooting cloud tops from geostationary satellite data. *GIScience & Remote Sensing*, *55*, 763-792.

Negri, A. J. (1982). Cloud-top structure of tornadic storms on 10 april 1979 from rapid scan and stereo satellite observations. *Bulletin of the American Meteorological Society*, *63*, 1151 - 1159. Retrieved from https://journals.ametsoc.org/view/journals/bams/63/10/1520-0477-63_10_1151.xml doi: 10.1175/1520-0477-63.10.1151

Negri, A. J., & Adler, R. F. (1981). Relation of satellite-based thunderstorm intensity to radar-estimated rainfall. *Journal of Applied Meteorology and Climatology*, *20*, 288 - 300. Retrieved from https://journals.ametsoc.org/view/journals/apme/20/3/1520-0450_1981_020_0288_rosbti_2_0_co_2.xml doi: 10.1175/1520-0450(1981)020<0288:ROSBTI>2.0.CO;2

Perez, L., & Wang, J. (2017). *The effectiveness of data augmentation in image classification using deep learning.* arXiv. Retrieved from https://

arxiv.org/abs/1712.04621 doi: 10.48550/ARXIV.1712.04621

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). *Overfeat: Integrated recognition, localization and detection using convolutional networks.* arXiv. Retrieved from https://arxiv.org/abs/1312.6229 doi: 10.48550/ARXIV.1312.6229

Setvák, M., Lindsey, D. T., Novák, P., Wang, P. K., Radová, M., Kerkmann, J., … Rabin, R. M. (2010). Satellite-observed cold-ring-shaped features atop deep convective clouds. *Atmospheric Research*, *97*, 80-96. Retrieved from https://www.sciencedirect.com/science/article/pii/S016980951000058X doi: https://doi.org/10.1016/j.atmosres.2010.03.009

Setvák, M., Rabin, R., & Wang, P. (2007, February). Contribution of the modis instrument to observations of deep convective storms and stratospheric moisture detection in goes and msg imagery. *Journal de Recherches Atmospheriques*, *83*(2-4 SPEC. ISS.), 505–518. doi: 10.1016/j.atmosres.2005.09.015

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014, jan). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, *15*(1), 1929â1958.

Weinreb, M., & Han, D. (2011, August). *Conversion of gvar infrared data to scene radiance or temperature.* (Accessed August 8th, 2022. https://www.ospo.noaa.gov/Operations/GOES/calibration/gvar-conversion.html)

Yu, X., Wu, X., Luo, C., & Ren, P. (2017). Deep learning in remote sensing scene classification: a data augmentation enhanced convolutional neural network framework. *GIScience & Remote Sensing*, *54*, 741-758. Retrieved from https://doi.org/10.1080/15481603.2017.1323377 doi:

10.1080/15481603.2017.1323377

Zha, S., Jin, W., He, C., Chen, Z., Si, G., & Jin, Z. (2020). *Detecting of overshooting cloud tops via himawari-8 imagery using dual channel multi-scale deep network.*

A | Appendix

A.1 Alternative Detection Methods

Table A.1: The name, approach taken, data used and author of comparable methods

Num	Approach	Data	Author
1	CNN with DCC clear sky elimination	GOES-16	this thesis
2	IRW-texture candidates and detection criteria	GOES-14	Bedka (K. M. Bedka & Khlopenkov, 2016)
3	OT probability	GOES-14	Bedka (K. M. Bedka & Khlopenkov, 2016)
4	OT probability with visible rating detection	GOES-14	Bedka (K. M. Bedka & Khlopenkov, 2016)
5	CNN with tropopause temperature processing	GOES-14	(Kanneganti, 2020)
6	CNN with clear sky processing	Himawari-8	(Kim et al., 2018)