Cloud Structure Classification using Convolutional Neural Networks

Vikas Hanasoge Nataraja
viha4393@colorado.edu

Submitted in partial fulfillment of the requirements of

ECEN 5244
Environmental Signal Processing

Department of Electrical and Computer Engineering
University of Colorado Boulder

December 9, 2019

ABSTRACT

Cloud features have become a very important parameter in climate models. Understanding the organizational structure and different aspects of a cloud such as shape, density and thickness is vital because of the information they carry. Specifically, the organization of so called “shallow” clouds plays a crucial role in these models because of the fact that they reflect solar radiation which would otherwise heat the atmosphere. In a warming climate, it therefore becomes important to understand how these shallow clouds will change. The biggest problem in modeling these clouds their thinness. Resolving all the important features becomes hard due to the low thickness and density. Traditional rule-based methods like edge detection are often inaccurate because the boundaries between these organizations are often unclear.

This project aims to use deep learning to understand these cloud organizations and features. By employing deep convolutional neural networks with semantic segmentation to classify the clouds into distinctly different categories. The proposed model is based on concepts from GoogLeNet Inception v3 [1] and DeepLab v3 [2], two neural network architectures which classify images using fundamental changes to computer vision techniques for robust feature extraction. The former uses factorized convolutional layers to reduce the number of parameters while the latter introduces the use of atrous convolutions to prevent downsampling and preserve the field of view. The results from the proposed model are significantly better than several other deep learning approaches such as ResNet and UNet and also current traditional rule-based and threshold-based methods.

1.0 INTRODUCTION

Climate models are crucial in our understanding of the changes in Earth's climate features such as the interaction between energy and matter, ocean and sea levels, precipitation levels and so on. Most climate models can be categorized as general circulation models or GCMs which use mathematical and physics equations to characterize different parameters including sea surface temperatures which, in turn, reflect the changes in climate. [3] and [4] explain how climate models are constructed and evaluated along with a literature survey of past and present models with emphasis on the advances made over the past 20 years. One feature which plays a critical role in modeling is cloud structure. Studying clouds and their changes over time can serve as indicators of the overall direction of climatology. [5] explains how analysis of cloud feedback mechanisms, which model top-of-atmosphere radiative flux changing as a cause of cloud response to warming, can help improve global climate models.

In particular, studying so called "shallow" clouds can provide a better understanding of climate change due to their inherent thinness and guide improvements to the representation of shallow convection[1]. [6] discusses the role and impact that boundary layer and shallow cumulus clouds have on the forecast of a large-scale weather system. Figure 1 shows an example of a shallow cloud. The ability of the shallow clouds to reflect sunlight and irradiance from the sun has inevitably changed over the course of the past century and analyzing the effects of their changes in the future will be valuable to climatology. Current models which largely employ so called "rules-based" methods do not agree on whether shallow clouds will increase or decrease with climate change[2]. One reason that affects these calculations is the fact that these clouds are not the result of global circulation in the atmosphere and tend to arrange in seemingly random patterns. Another reason is that the structures in shallow clouds are broadly defined and the boundary between a pair of categories cannot be easily identified. [7] explains the meso-scale organization of shallow clouds in trade wind regions and subtropics and also bins them into categories which are used in this paper. Shallow cumulus clouds, also called fair-weather cumuli are most often characterized by their small size, relatively weak convection, and no precipitation, which is distinctly different from deep convection clouds[3].

To address the issue of modeling these clouds, this paper proposes the use of deep learning. Specifically, the use of deep convolutional neural networks or CNNs which has taken off in recent years. The biggest advantage with the use of deep learning is the automation of feature extraction, classification and segmentation all of which need to be hard coded in rules-based methods. Image classification in particular has taken big strides in the past 10 years. The ImageNet competition[4] (ILSVRC) is a platform to test new object detection and classification algorithms based on CNNs and the model used in this paper is based on two winners of the competition. Details of the model are discussed in section 3.

---

[1] https://www.pnnl.gov/science/highlights/highlight.asp?id=4982 Article about shallow clouds
[2] https://towardsdatascience.com/sugar-flower-fish-or-gravel-now-a-kaggle-competition-8d2b6b3b118 Article explaining the dataset in this paper and the reasoning behind it
[3] https://phys.org/news/2017-05-intensive-shallow-cumulus-clouds-mongolia.html Shallow cloud measurements
[4] http://www.image-net.org/challenges/LSVRC/ ImageNet Competition

Figure 1. Shallow clouds scattered over land. Some structures are not separated by a distinct boundary or a gap which makes it difficult to use thresholded methods (*Image credit: Stony Brook University*[5]*)*.

2.0 GENERAL DESIGN PRINCIPLES

This section details a few design principles based on large-scale experimentation with various architectural choices with convolutional neural networks. Additionally, this section briefly explains common terminologies in the design of neural network architectures and the specific techniques used in the proposed model in this paper.

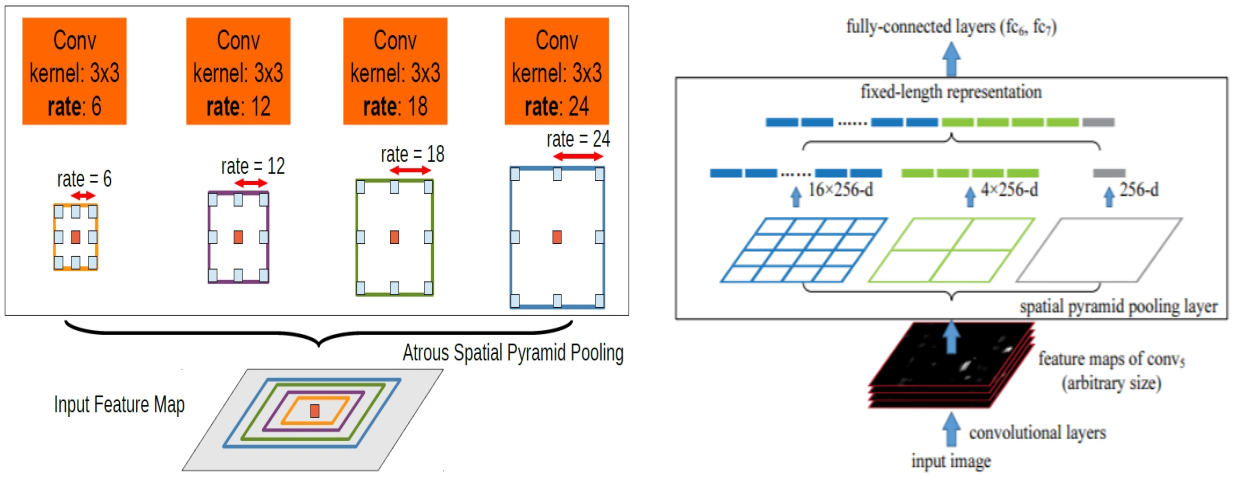
1. *Atrous convolution* is a concept that was first widely introduced in the application of deep learning by DeepLab v3 [2]. Up until then, most CNN models used spatial convolution operations for the convolutional layer followed by a form of pooling. However, this results in a downsampled product and the network will most often need to go deeper by adding more convolutional layers to extract all the features. In order to avoid severe degradation of the input images and preserve field of view, atrous convolution is used. This method pads zeros to the feature map produced by the convolution operation thus delivering a wider field of view. This is most commonly performed in parallel with different rates (or strides) and is then called *atrous spatial pyramid pooling,* an idea first introduced by He et al. [8]. Liu et al. [9] designed ParseNet, a CNN used primarily to look wider in images. Figure 2 shows an illustration of the operation and Figure 3 depicts the use of the technique in a CNN.

2. *Dropout* is a technique that is commonly used to prevent overfitting the model. It was introduced in 2015 by Srivastava et al. [10] and the key idea behind it is to drop random units and their connections during training so as to force the network to learn from fewer

5 https://www.somas.stonybrook.edu/2016/12/13/somas-study-uses-radar-to-better-predict-shallow-cloud-coverage/ Research article by Stony Brook University's School of Marine Atmospheric Sciences

features and become less codependent on units. Today, most CNNs use this technique almost universally. Each layer in a neural network can make use of this technique with a high dropout probability given to the input layer to make the learning process more robust. [10] also proved that the use of dropout increases the performance in tasks such as vision, speech recognition and document recognition (optical character recognition or OCR).

3. *Optimizer* is an algorithm used to converge to the global minimum of the cost function and guided by the loss function. Essentially, optimizers tie together the loss function and the model parameters by updating the model weights in response to the output and the loss function. One of the most commonly used optimizers is *Adam* or *Adaptive Momentum Estimation* which uses past gradients to update the current gradients. It has been known to converge faster than most other optimizers and this paper also uses the same.



(a) Atrous convolution with kernel 3x3 (b) Spatial pyramid pooling layer

Figure 2. (a) shows atrous convolution operation with a 3x3 kernel and different atrous rates. Higher rates yield greater field-of-view [2]. (b) shows a spatial pyramid pooling layer employed in a network, 256 is the filter number for the $conv_5$ layer.

3.0 DATASET

The dataset used in this paper comes from NASA Worldview and the Max Planck Institute for Meteorology in Hamburg, Germany, explained in Rasp et al. [7]. Over 9,000 images are made available for the scope of this paper. Of those images, over 5,500 images are used for training the model. Each training image has a ground truth value associated with it. The class label falls into one of 4 categories - Fish, Flower, Sugar and Gravel and an example is shown in Figure 3. Along with the label, another parameter called Encoded Pixels is given which when decoded produces a mask over a portion of the image that has the particular label associated with it. Figure 4 shows an example of images overlaid with their masks. An image can have multiple labels and masks associated and some more statistics about the dataset are shown in Appendix A.

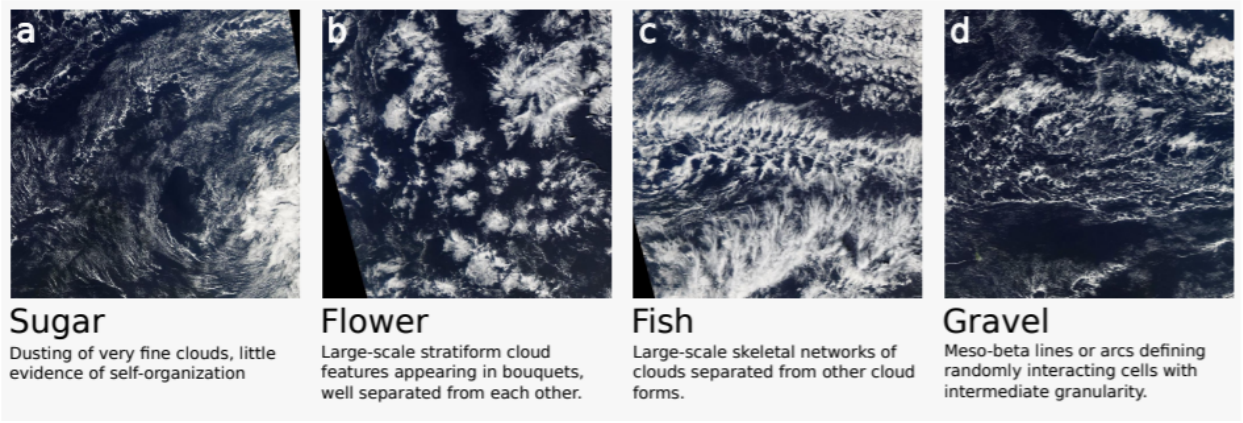


Figure 3.(a)-(d) show the 4 class label examples of the 4 cloud categories with corresponding descriptions. Each label was selected subjectively by Rasp et al. [7] and broadly classified.

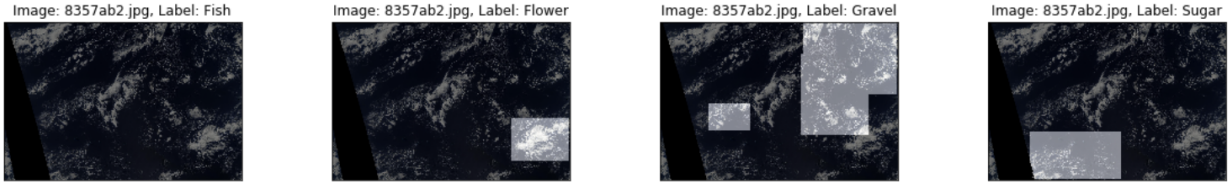


Figure 4. Image 8357ab2.jpg overlaid with respective masks for corresponding labels in gray. Not all images have all four class labels.

4.0 MODEL ARCHITECTURE

To automate the process of feature extraction, detection and classification, this paper proposes using a convolutional neural network based on concepts taken from [1], [2] and section 2. The backbone of the proposed model is Inception v3 and the complete model is shown in Figure 5. The standard convolutions are replaced by *atrous convolutions* and any convolution referenced further points to atrous convolution. The original Inception v3 model is 48 layers deep while the proposed model is 51 layers deep with a series of *convolutional*, *pooling and concatenation layers* and rounded out by a *fully connected layer* and softmax function as the output layer to produce the classification results. The inception module consists of parallel convolutional layers, each with different kernel sizes and strides (*factorized convolution)* and this module is repeated several times to increase the depth of the network. Inception Module A has 4 parallel convolutions of which two are 1x1 convolutions, one 3x3 convolution and one 5x5 convolution with this structure being repeated 5 times. Module B has a similar architecture except that the inception layer is repeated 4 times. Module C has 6 parallel layers, each having convolutions of varying kernel sizes (3x3, 1x1, 5x5) and two of the layers have a concatenation layer to account for dimensional compatibility.

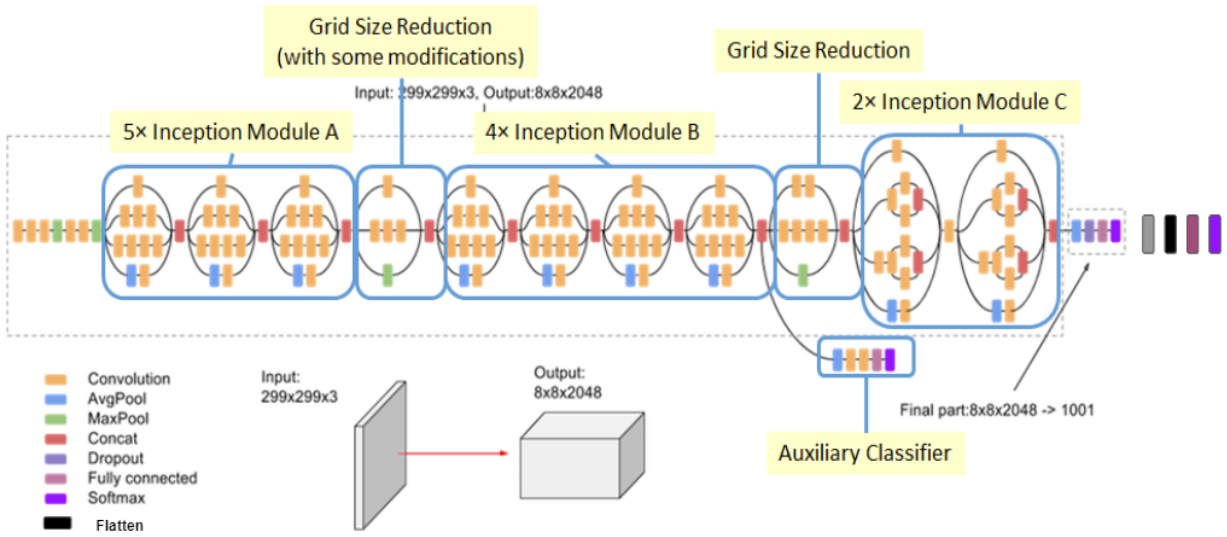


Figure 5. Model architecture proposed in this paper. The layers are each color-coded with details of the layers being explained in section 4.

The input image is fed to the first 7 layers which are a series of convolution and average pooling layers. Inception module A is then employed followed by a grid-size reduction module. This is fed to Inception module B which is also followed by a grid-size reduction module that is deeper than the previous. Finally, Inception module C is used followed by the auxiliary classifier which employs dropout, fully connected, pooling and softmax[6] layers. The final 3 layers - Dropout, Flatten and a fully connected layer are added at the end. The output layer is a softmax function.

5.0 METHODOLOGY

This section discusses the implementation of the model to classify the cloud structures into one or more categories. Each sub-section explains a different task or methodology used before and during the training process.

5.1 Image Pre-Processing

The available images are RGB and of size 2100 x 1400. Before the training process begins, a number of steps are taken to pre-process the image and tune the hyperparameters. In the pre-processing step is important and common in most models. The following techniques were used in this step:

[6] https://medium.com/data-science-bootcamp/understand-the-softmax-function-in-minutes-f3a59641e86d Article on softmax function

- *Image resizing* - The input image is resized to 400 x 400 because the Inception v3 architecture was originally trained and tested on 299 x 299 images but 400 x 400 was sufficient.
- *Contrast increase* - The dataset has distinct color separations between white and blue/black. Increasing the contrast is a way of nuding the network towards looking for higher contrast differences which can be done by increasing the original contrast.

- *Image normalization* - All images are normalized to pixel values between 0 and 1. This is primarily done because most CNNs expect images to be normalized and also because normalization is generally accepted method to set the image values.

- *Data augmentation* - Augmentation is a popular technique used to add to the existing dataset with different permutational manipulations to images. Mainly used to augment small datasets, it also serves to make the model more robust to different images. The data augmentation techniques used in this paper are horizontal and vertical flips, rotation and image crops.

It is important to note that the validation set does not undergo this step. This is because the model does not learn from the validation set but rather uses it as a check on the training set.

5.2 Data Generation

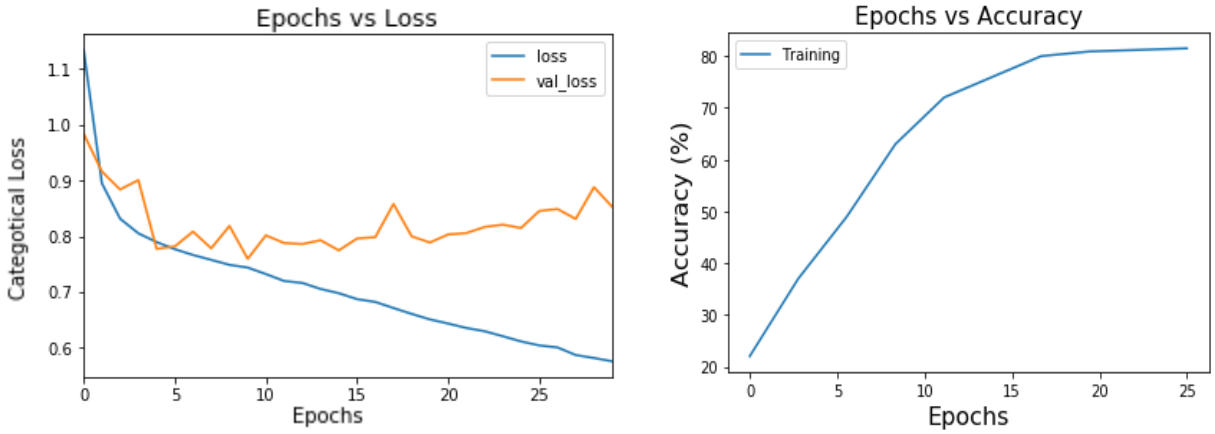
Once pre-processing is done, the output labels are one-hot encoded. This is to ensure the softmax function can easily activate the neurons. The training set was also split 80:20 into a training set and a validation set. The latter is a commonly used practice mainly used as a sanity check during the training process. Both the training and validation sets are fed by a generator which produces images in the set batch size with the various pre-processing steps included. It is important to note that the network does not learn from the validation set i.e., there is no backpropagation used. It is only used to serve as a check on the training process.

5.3 Hyperparameter Tuning

The training process works through various hyperparameters that can be changed or tuned during successive runs. The *batch size* is set to 64 for the batch normalization and data generation steps. Higher batch sizes usually increase training time but also increases accuracy metrics. The number of *epochs* is set to 25 with 25 *steps per epoch*. This is the recommended preset by Keras, the framework used for this model. The *train-validation set* split, which is set at 80:20 can also be tuned, so too can the *resizing dimensions*. *Dropout probability* is set to 0.5 which means there will be a 50% chance of a node or a connection being dropped in the hidden layer. The *learning rate* for the Adam optimizer is set to 0.0001 which increases convergence time but improves loss and accuracy measurement metrics.

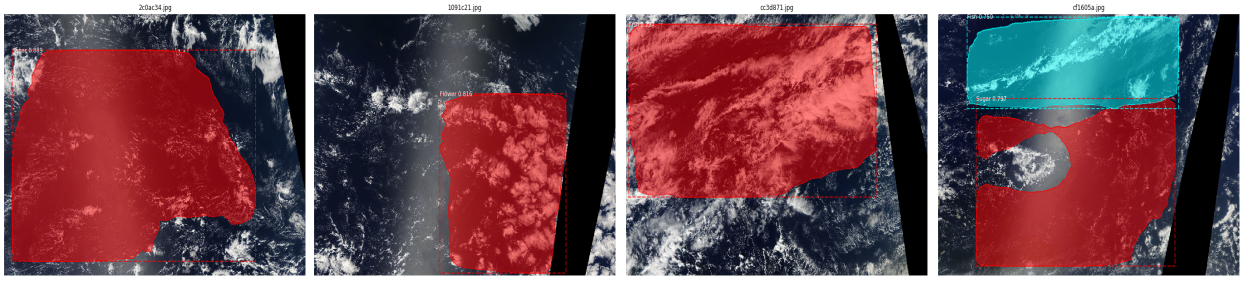
6.0 RESULTS

The model was run with categorical cross entropy as the loss function and accuracy was measured at each epoch. Figure 6(a) shows the progress of the loss for both the training and validation sets. It clearly shows that loss decreases over epochs and plateaus after about 25 epochs. This plot combined with 6(b) which plots the accuracy for the training set paints a clear picture of the network learning the features and making correct predictions. The highest training accuracy is 81.5% while the peak for testing is 66.9%. The training loss steadily decreases over time. The validation loss, which was not used in the learning process by the model suffers and has a noisy decrease. Since this was a Kaggle competition, the ground truth labels for the test set are not available, hence no plots. There are, however, some predictions that the model is capable of making and the mask predictions along with the label are shown in Figure 7.



(a) Plot of loss over epochs (b) Plot of accuracy over epochs

Figure 6. (a) shows the decreasing loss over the epochs for the training set and validation set and (b) shows the plot of accuracy over epochs for the training set.



(a) Sugar (b) Flower (c) Fish (d) Fish and Sugar

Figure 7. (a) shows the mask prediction for the class label "Sugar", (b) shows "Flower" (c) shows "Fish" and (d) shows an instance with 2 masks - "Fish" in teal, "Sugar" in red

Table 1 shows a comparison with the leaderboard of the Kaggle competition and the proposed model ends up in position 6 of 1,538 participants. It beats other state-of-the-art models including

ResNet and VGG-16 models used by other teams by considerable margins. It outperforms EfficientNet b2 by nearly 5 percentage points.

Position	Model	Accuracy score
1	Unknown model	0.67175
2	ResNet - 38	0.67167
3	ResNet - 50	0.67136
4	Unknown model	0.67082
5	UNet	0.67080
6	**Proposed Model**	**0.66920**

Table 1. Leaderboard of the Kaggle competition. Shows the expected finish of the proposed model at 6th place.

7.0 CONCLUSIONS

This paper and project were researched and tested over the course of over a month and from the results, some conclusions can be drawn and are detailed in this section.

The network learned to cope with cropped and irregular shapes of data. This can mainly be attributed to the data augmentation done at the pre-processing step. The model did not overfit due to the appropriate use of dropout and data augmentation. The validation set did however, suffer which can be because of insufficient data in the dataset itself. Another reason might be that the set did not undergo the pre-processing step. Fine tuning the *batch size* hyperparameter proved to be crucial to achieving better accuracies and overall improved performance more than any other hyperparameters.

8.0 FUTURE WORK

The current model can be expanded in multiple ways and following are some of the suggested improvements that can be made in the future:

1. Increased dataset and better validation set. CNNs typically require thousands of images to train and test and increasing the size of the dataset will undoubtedly be beneficial. Supplementing other forms of data augmentation should also be a part of any such endeavors.

2. Change accuracy and loss metrics with regularization. The proposed model used categorical cross-entropy as the loss function and standard accuracy to measure the model's performance. This can instead be replaced with the *dice coefficient* and the *dice loss* which have been known to give good results in segmentation models for image classification.

3. Scheduled learning rate. The proposed model used a constant learning rate of 0.0001 which can be modified to be scheduled to increase towards the end of training or after a certain number of epochs. This will improve convergence time as well as the model performance.

Any future work should focus on one or more of these suggestions to increase model performance. With the proposed architecture, several changes and modifications can be made in due time to improve overall accuracy of the classification.

9.0 REFERENCES

[1] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. "Rethinking the Inception architecture for computer vision". *Conf. on Computer Vision and Pattern Recognition,* December 2017

[2] Chen, L.C., Papandreou, G., Schroff, F., Adam, H. "Rethinking atrous convolution for semantic image segmentation". *Conf. on Computer Vision and Pattern Recognition,* December 2017

[3] Randall, D.A., R.A. Wood, S. Bony, R. Colman, T. Fichefet, J. Fyfe, V. Kattsov, A. Pitman, J. Shukla, J. Srinivasan, R.J. Stouffer, A. Sumi and K.E. Taylor: "Climate Models and Their Evaluation. In: Climate Change 2007: The Physical Science Basis". *Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change* Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2007

[4] Pincus, R., C. P. Batstone, R. J. P. Hofmann, K. E. Taylor, and P. J. Glecker. "Evaluating the present-day simulation of clouds, precipitation, and radiation in climate models". *J. Geophys. Res. Atmos.*, 113, D14209, July 2008

[5] Ceppi P, Brient F, Zelinka MD, Hatmann DL. "Cloud feedback mechanisms and their representation in global climate models". *Wiley Interdiscip. Rev. Clim. Change* 8:e465, December 2017

[6] Bélair,S., J. Mailhot, C. Girhard and P. Vaillancourt. "Boundary layer and shallow cumulus clouds in a medium-range forecast of a large-scale weather system". *Mon. Wea. Rev.* July 2005

[7] Rasp, Stephen., Hauke Schulz, Sandrine Bony, and Bjorn Stevens. "Combining crowd-sourcing and deep learning to understand meso-scale organization of shallow convection". *arXiv preprint arXiv:1906.01906*, June 2019.

[8] He, K., Zhang, X., Ren, S., & Su, J.." Spatial pyramid pooling in deep convolutional networks for visual recognition". *European Conf. on Computer Vision*, June 2014

[9] Liu, W., Rabinovich, A., Berg, A.C. "ParseNet: Looking wider to see better". *International Conf. on Learning Rep*., May 2016

[10] Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. “Dropout: A simple way to prevent neural networks from overfitting”. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014

APPENDIX A

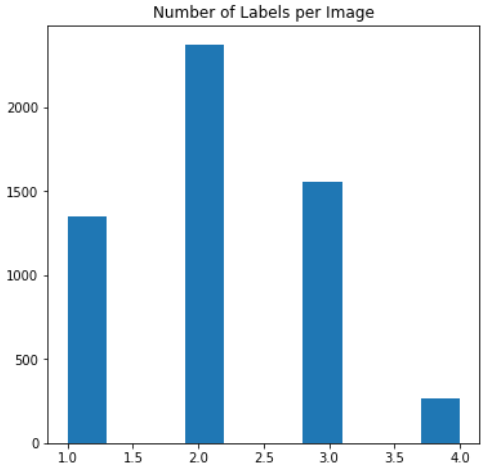


Figure 8. Distribution of the labels per image in the dataset. Most images have 2 labels while less than 300 images have all 4 labels.

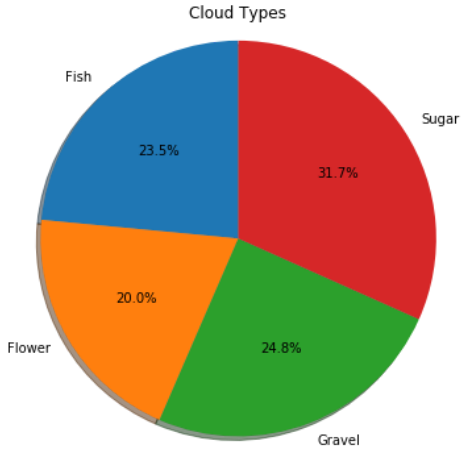


Figure 9. Pie chart showing the distribution of cloud types in the dataset.