Deep Learning 19/20: Understanding Clouds

Naomi Tzabari, Shai Aharon

naomilau699@gmail.com
shai.aharon1@msmail.ariel.ac.il
Ariel University

Abstract

Climate change has been at the top of our minds for many years. Shallow clouds play a huge role in determining the Earth's climate. Sadly, they are also difficult to understand and to represent in climate models.
By classifying different types of cloud organization, researchers at "Max Planck" institute hope to improve our physical understanding of these clouds, which in turn will help us build better climate models.There are many ways in which clouds can be organize, but the boundaries between different forms of organization are murky. This makes it challenging to build traditional rule-based algorithms to separate cloud features. The human eye, however, is really good at detecting features—such as clouds that resemble flowers. In the past decade there has been a wide success in the field of Deep Learning in classification problems, from simple objects to complex structures. This paper tries to approach the cloud formation classification problem using Deep Neural Networks in general, and particular Convolutional Neural Networks combined with auxiliary loss method, which have shown to be very powerful in image classification tasks.

1. Introduction

The classification of objects in images is a well studied topic in the field of computer vision, it began in a summer project in 1966 at MIT[1], and not surprisingly still goes on to this day. Although a lot has been accomplished in the field, most of the work of classification is centered around common objects, like cars, people, animals etc. Some work relies on classic machine learning techniques[2] and others (most) use deep neural networks[3].
For rigid objects, classical machine learning techniques would mostly suffice, even hand crafted features

[1] The Summer Vision Project
[2] Histograms of Oriented Gradients for Human Detection
[3] ImageNet Classification with Deep Convolutional Neural Networks

such as SIFT[4] would achieve good results, but would fail when it comes to deformable objects such as humans. A significant advancement was achieved using Deformable Parts Model[5] which gave the HOG model much more spatial flexibility, pale in comparison to the CNN based solutions.

1.1 Related Work

In this paper we discuss cloud formation classification which does not share the same characteristics as the above objects due to the abstract shape of the clouds, and in some aspects are more related to texture classification. To the extent of our knowledge, texture recognition work focused on small recurring textures[6]. This description does not apply to cloud formation as the are partially recurrent at best.

Zhuo et-al.[7] classified clouds using texture feature extraction and a Multi-Class SVM classifier. They separated the RGB image to 3 seperate channels using a CCT (*color census transform*) function and extracted features from each channel independently. For each channel they built a 4 level spatial pyramid and extracted histograms for each level. Their work showed remarkable results, but was very limited as they detected only clouds types and not formations, meaning they limited their work to clouds who have a very well defined shape. Another issue is that the CCT is tailored to the task as it not learned

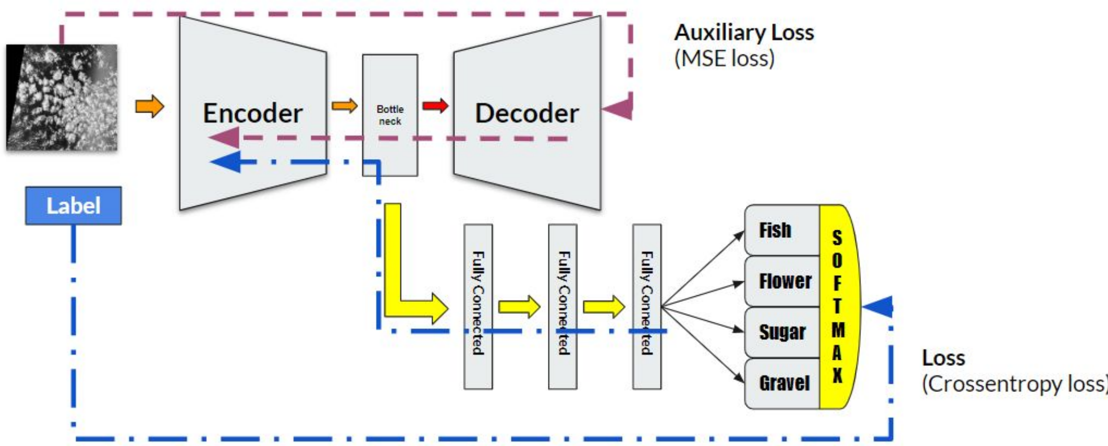

Figure 1. The network architecture.

[4] Improved SIFT-Features Matching for Object Recognition

[5] Object Detection with Discriminatively Trained Part Based Models

[6] BRINT: Binary Rotation Invariant and Noise Tolerant Texture Classification

[7] Cloud Classification of Ground-Based Images Using Texture–Structure Features

thus cannot be easily generalized to other domains as in image from different viewpoints (satellite images etc..),
In this paper we present a novel approach to classify cloud formation from images taken from satellites using a combination of a *CNN AutoEncoder* and *Auxiliary loss*.

2 Proposed Method

The task of **C**loud **F**ormation **C**lassification (CFC) combines the properties of texture recognition and object recognition. In this section we propose a novel network architecture to solve the CFC problem. As illustrated in above. The network is composed of two main blocks:
I) CNN AutoEncoder
II) ANN
Each has its own loss, block **I** is trained compared to the input image itself with MSE loss while block **II** is trained compared to the images label.
The ANNs input is the autoencoders bottleneck. In test time, the network only used the encoder part of block **I** and the ANN.
The goal of block **I** is to reconstruct the input image as best as it can, while block **II**'s goal is to classify the clouds formation.
The idea behind the current structure is that although the autoencoder will not be able to fully recreate the cloud image, it will, how ever succeed in recreating the outline shape of the cloud formation, and that information is embedded in the "bottleneck" of the autoencoder. Since the input for the ANN is from the "bottleneck", it gets a "cleaner" (*skeleton*) input, since the autoencoder filtered out the noise (random, shapeless margins of the clouds etc..) and more (see **fig.2**), and can learn much more efficiently. Although, in some cases sun flair on the image caused the reconstruction to be "add" a cloud as shown in **fig 5**.

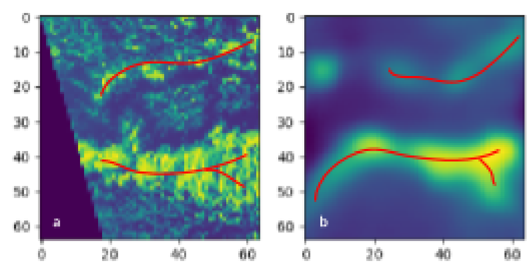

Figure 2. (a) Input image, (b) AutoEncoder output. In red the "*Cloud Skeleton*"

The two blocks are trained together and not one after the other. This is done in order to force the autoencoder to embed in a way that will not only be beneficial for reconstruction, but in a way that will benefit the classification, It does reduce the performance of the reconstruction, but that is neglectable due to the fact that is still aids the classification.

3 Experiments and Results

The AutoEncoder layers are a series of convolutional layers as follows

(C_s^k -2D convolutional layer with a kernel size of *k* and a stride of *s*, and a negative s means its a transposed convolutional layer
FC_n - Fully connected layer with n neurons
DO_r - Dropout layer with a ration of *r*):
Encoder:
$C_1^5 - C_1^5 - C_1^5 - C_2^5 - C_1^5 - C_1^5 - C_1^5 - C_2^5 - C_1^5 - C_1^5 - C_1^5$
Bottleneck: FC_{256}
Decoder:
$C_1^5 - C_1^5 - C_{-2}^5 - C_1^5 - C_1^5 - C_{-2}^5 - C_1^5 - C_1^5 - C_1^5$
Transposed convolutional layers were used instead of the traditional MaxPool layer due to the information lost using them which made the degraded the reconstruction process.
ANN:
$FC_{32} - FC_{32} - FC_{32} - DO_{0.4} - FC_4$
The network was trained for 80 epochs with a learning rate of 10^{-4} and a decay of 0.1 every 5 epochs, and achieved a test accuracy of **71.4%** (see **fig 3**).
The training was performed on a laptop with NVIDIA GeForce GTX 1650 GPU and took 57 minutes.

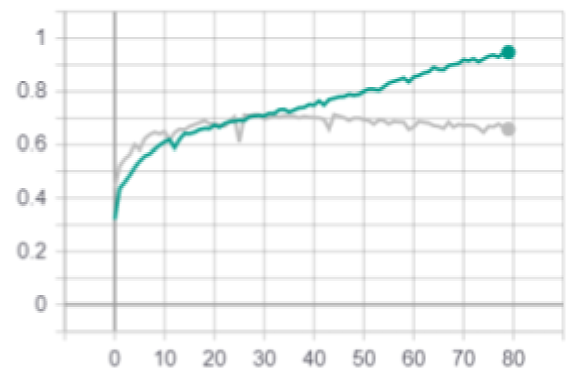

Figure 3. Training/Test accuracy during learning

4 Conclusions

In this paper, we demonstrated a method to classify highly non-rigid cloud formation using a convolutional neural network, Although reaching the accuracy that it did wasn't trivial, it's still, in our opinion, a low score. The fact that in training we were able to achieve overfitting on the train set gives us hope that the task is feasible. On the bright side, the inference time on the proposed network is remarkably fast, $\approx 176\,fps$,
We hope it will be useful in aiding the climate change research.

5 Previous Attempts

Before arriving to the final model, other simpler models were tested. First, a simple Logistic Regression was tried, the results were 50% accuracy on the test, and 54% accuracy on the training set. Since we did not get evan

close to overfitting on the training set, we concluded that LR does not contain enough weights and/or learning power to learn the task in hand.
Adding more layers to the previous network did not improve the test accuracy, but was sufficient to overfit on the training data.

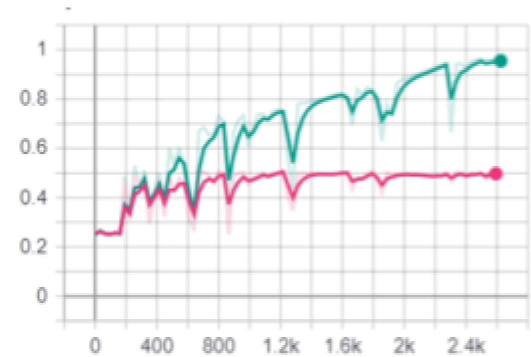

Figure 4. Train/Test accuracy during ANN learning

A CNN model was tested and yielded a high accuracy of **68%**. Many different architectures of CNN were tested, but none passed the above accuracy.
A new approach was proposed, training an *AutoEncoder* on the cloud images, with a narrow bottleneck, then build a *K-NearestNeighbor* model, while using the encoder as the feature extractor. Although the *autoencoder* was able to reconstruct the images quite well, the embedding into euclidean space and classifying using the *KNN* model did not yield good results and were $\approx 52\%$. Lastly we decided to combine the CNN with the AutoEncoder with an auxiliary loss as described in the paper above.

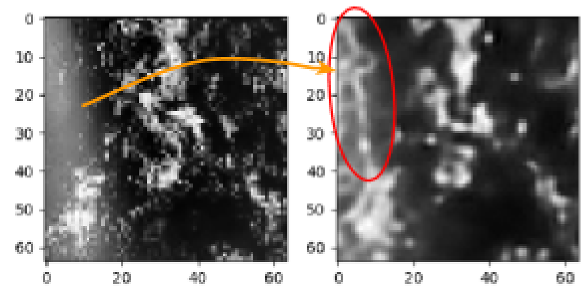

Figure 5. The AutoEncoder reconstruction (right). In red, the "Ghost" clouds caused by a sun flair.