Volcano Detection on Venus —- CS 684 Final Project Report

Si Ao Chen
University of Waterloo

Qi Mai
University of Waterloo

Abstract NASA's Magellan spacecraft launched in 1989 returned a number of photos of the surface of Venus. Researchers were interested in the volcanoes on the surface and their features. Our tasks are to determine whether there are volcanoes in an image and if existence is determined by researchers, the number of volcanoes and the radius are of interest. Since the data is imbalance, we perform oversampling and class weighting in our convolutional neural network.

Keywords: object detection, classification, imbalance data, convolutional neural network

1. Introduction

Spending approximately 4 years traveling around Venus, the spacecraft named Magellan captured a large amount of data of the planet's surface. The mission was to acquire the topological relief of Venus by mapping the surface using synthetic aperture radar (Dheeru and Karra Taniskidou, 2017). Each image in the dataset has a view of less than $100m^2$ on the ground. Researchers labelled each image with the following four tags: volcano detection (binary, 1 for presence and 0 for no volcanoes), uncertainty level of detection (4 levels, Figure 1), the number of volcanoes and the radius of volcanoes (Figure 2). Due to the ambiguity of the data, the four labels are not guaranteed to be absolute ground truths.

Given the set of images, we trained models to make predictions on the four labels. We conducted classification on the first three labels and regression for radius estimation. In classification tasks, we realized that the dataset is imbalance – that is, some classes have a large number of images but some classes have only a few. To tackle the imbalance issue, we could either oversample the images in the minor classes or put weights on classes to balance the dataset. In our experiments, the CNN models we use are generally not stable due to imbalance data. Adding batch normalization layers significantly increases the model stability.

The rest of our report is organized as follows. Section 2 introduces the dataset properties and describes data preprocessing methods. Section 3 introduces different classification methods on each feature of interest and the corresponding prediction results. Finally, Section 4 provides the discussion of our discoveries, as well as the suggestions on future work.

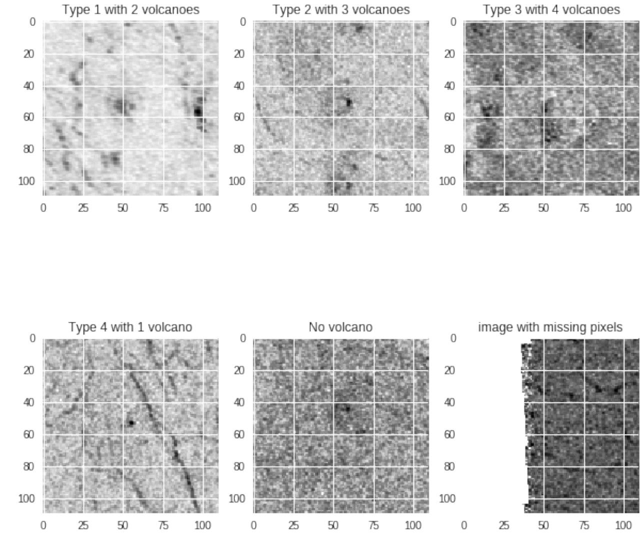

Figure 1. Examples of images in the training set

2. Data preprocessing

2.1 Data Overview

The dataset we use in this project was downloaded on Kaggle. There are 7000 images in the training set and 2734 images in the test set. Each image has size 110x110 pixels with values in [0,255]. An image may have volcanoes or do not have volcanoes. Researchers also labelled their uncertainty of volcano detection as 4 types:

- Type 1: definitely has volcanoes,
- Type 2: probably has volcanoes,
- Type 3: possibly has volcanoes,
- Type 4: only a pit is visible.

There are much more images without volcanoes than images with volcanoes. Among the images with positive volcano detection, the majority of discovered volcanoes are type

3 or 4, which means that the uncertainty on detected volcanoes is generally very low (Figure 2). Also, some images are corrupted possibly resulted from the gaps in communication processes, with blank regions as shown in the last image in Figure 1. We dropped these the corrupted images from the training set and eventually, we had 6741 images for training. As shown in Figure 3, most images with volcanoes contain only one volcano, and the majority of volcanoes detected have small radius.

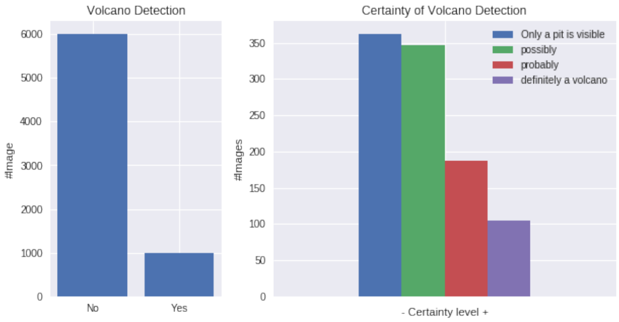

Figure 2. The uncertainty of Volcano Detection

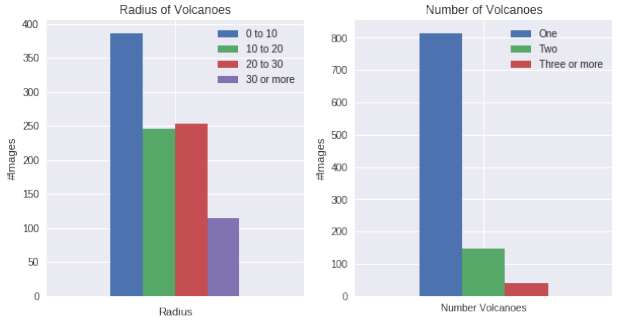

Figure 3. The number of volcanoes and the radius of volcanoes

2.2 Imbalance Data

Even though volcanic features dominate the surface of Venus, there is still a relatively small number of pictures actually contain volcanoes. The image set has much less items containing volcanoes than the ones do not. To tackle an imbalance classification problem, the common ways are resampling data (undersampling the majority or oversampling the minority), weighting classes (light weight on the majority and heavy weight on the minority) and using different performance metrics instead of accuracy, such as precision, recall, F1-score and confusion matrix. Accuracy is a misleading metric to evaluate the performance of a classifier on imbalance data. For example, suppose the dominating class occupies 80 percent of the whole training set, predicting every instance to be in this class gives 80 percent accuracy but such classifier is trivial.

To tackle this problem, we can perform oversampling on the underrepresented class or put class weights in our model. The ratio between the majority class and the minority class is about 6:1. We perform oversampling using SMOTE, which generates synthetic images to achieve balance among classes (Chawla, Bowyer, Hall, and Kegelmeyer, 2002). Class weights are computed using the ratio of samples in each class; dominating classes receives small weights and minor classes receives large weights. In the following experiments, we apply both methods and compare the corresponding results.

3. CNN models and results

3.1 Volcano Detection

We adopted the model created by Heming Zhen (2018) and modified it as shown in Table 1. Stochastic Gradient Descent is chosen as the optimizer for our model, with learning rate 0.001, decay 1e-6 and Nesterov momentum 0.95. At the beginning, we used dropout with rate 0.5 after max pooling layers and no batch normalization, but the model always predicts only one single class, either the dominating class or the minor class, with accuracy 84.13% and 15.87% respectively. We tried different optimizers with various learning rates, added more layers and inherited the VGG16 model (Simonyan and Zisserman, 2014) with a customized dense layer, but none of these approaches solved the issue. With batch normalization after each convolution layer and the fully connected layer, our model and VGG16 model both successfully learned from two classes and made meaningful predictions. One possible reason for this issue is that the data is imbalance in each batch and the model got stuck at the local minimums. The class weights are unable to equalize the amount of information to be learned from each class. In our experiment, our model works only a few times without batch normalization and these are the cases that the model gets good bias and kernel initialization. Such successful cases are not generally reproducible. Batch normalization increases the stability of our model and gives pretty good accuracy and F1 score.

We adopted the VGG16 model trained on ImageNet and modified the final fully connected layers to fit our problem. A batch normalization layer is also added for stabilization. However, we trained this model several times and found that VGG16 sometimes has significant fluctuations in both accuracy and loss, while the learning process of our model is stable. One possible explanation could be no batch normalization in VGG16 model. When class weights are assigned, transfer learning with VGG16 gives better results than our CNN model, with higher accuracy and F1-score.

Moreover, putting weights on classes gives higher accuracy and F1-score than oversampling by SMOTE in our

Table 1

CNN Architecture

Layer	Depth	Kernel size
Conv2D	32	3×3
Batch Normalization		
Max pooling		2×2
Conv2D	64	3×3
Batch Normalization		
Max pooling		2×2
Conv2D	128	3×3
Batch Normalization		
Max pooling		2×2
Conv2D	64	3×3
Batch Normalization		
Max pooling		2×2
Dense	512	
Batch Normalization		

Table 2

Volcano Detection (Binary Classification) Results

Model	Balance Method	Test Accuracy	F1 Score (0, 1)
Our model	class weights	97.92	(0.99, 0.93)
	oversampling	96.96	(0.98, 0.88)
VGG16	class weights	98.61	(0.99, 0.96)
	oversampling	94.95	(0.97,0.84)

model and VGG16. However, our model learns faster using balance data than putting weights on imbalance data since we have more sample data in the balance set. It took only 3 epochs to reach 96% accuracy in the test set, while our model with class weights took 7 epochs. When balance data is fed in VGG16, the accuracy on the test set fluctuates more frequently than using class weights. We conclude that the VGG16 model might overfit the balance data.

3.2 Uncertainty Prediction

To predict the uncertainty of volcano detection, we used 4 types of uncertainty plus "no volcanoes" as the 5th label. The classes are imbalance and the prediction for minor classes, i.e. volcanoes in images with certainty, is extremely inaccurate. Our model gives 90.78% accuracy but poor f1-score in the classes of certainty. For the comparison of class weights and oversampling method, again, our model learns faster with balance data, which takes our model 2 epochs to reach about 90% accuracy in the test set, while imbalance data with class weights takes about 15 epochs to achieve such accuracy. Overall, assigning class weights gives slightly better result.

Alternatively, a more intuitive way to predict the uncertainty is to threshold the probabilities in the output of the very last dense layer, where the sigmoid activation function is applied, of the volcano detection model. However, this approach does not give better result than throwing the data in a five-class classification model stated as above. We omit the result of this method here.

Table 3

Uncertainty prediction results

Our model	F1-score	
Class	oversampling	class weights
No volcanoes	0.97	0.98
Only a pit visible	0.68	0.75
Possibly	0.39	0.49
Probably	0.27	0.24
Definitely	0.25	0.21
Overall Accuracy	90.16	90.82

3.3 Number of Volcanoes Prediction

The number of volcanoes in an image is also of interest. We trained our model with the number of volcanoes (including zero volcanoes) as labels and obtain 94.95% accuracy in the test set. However, for the images with volcanoes only, i.e. we excluded the case of zero volcanoes, the accuracy is about 76%.

Furthermore, we also tried to ensemble different models to make predictions. Our well trained classification model has already extracted a lot of representative features of the surface of Venus. In the following experiment, we extracted the output before the very last dense layer and take each row as the features of an image. Then, we fitted the features using the following classifiers to predict the number of volcanoes from the images that have positive volcano detection:

- Support Vector Machine(SVM), a popular supervised learning model that is able to perform excellent multi-class classification with a correct selection of kernel.
- Random Forest, an ensemble of independent and randomized decision trees that can performs both regression and classification problems
- XGBoost, a updated implementation of Gradient Boosting that ameliorate the speed and performance of Gradient Boosting.

In the task of predicting the number of volcanoes with positive detection, using the extracted features from the binary classification model, the above three classifiers have similar performance. Meanwhile, these three classifiers also yield higher accuracy than 76%, which is the accuracy produced by the our CNN model for 1-to-3 volcanoes prediction for the images with positive volcano detection, as mentioned in the beginning of section 3.3.

Table 4
Number of volcanoes prediction results (0-3 volcanoes)

Our model	F1-score	
#Volcanoes	oversampling	class weights
0	0.98	0.98
1	0.80	0.81
2	0.08	0.21
3	0.00	0.07
Overall Accuracy	93.93	94.95
1-3 Volcanoes Acc	63.36	73.76

Table 5
Number of volcanoes prediction results (1-3 volcanoes, using weighted CNN model)

#Volcanoes	F1-score
1	0.87
2	0.20
3	0.10
Overall Accuracy	76.27

Table 6
Number of volcanoes prediction results (1-3 volcanoes, using extracted features)

Models	Precision	Recall	F1-score	Test Accuracy
SVM	1.00	0.83	0.91	82.72%
Random Forest	0.97	0.82	0.89	82.49%
XGBoost	0.98	0.82	0.89	81.80%

3.4 Radius of Volcanoes

We found that estimating the radius of volcanoes is the most difficult task in mapping Venus's surface. First, we conducted regression to predict radius with various regressors, such as Gradient Boost and multilayer perceptron model. We tried to use the original images or the output before the last dense layer of the binary classification model as the input to the regressors, but we did not obtain good results. The models overfit the training data severely. Parameter tuning still gives us no satisfactory results. Then, we relaxed the regression problem to a classification problem – dividing the data into 4 groups by thresholding the radius. However, such relaxation still does not give a desired solution as expected. More investigations would be needed to find a good model for the radius prediction.

4. Discussion and Further Work

In this project, we used CNN model to classify the images of Venus surface. Even though our CNN model is relatively small and shallow, it completes the detection and classification tasks well with respect to different target labels. We also explored the methods for imbalance data, assigning class weights and oversampling with SMOTE. Through our experiments, we noticed that batch normalization is crucial for training this imbalance dataset. A good prediction of our model is barely reproducible without batch normalization, which stabilizes the model and decreases the computational cost.

For the imbalance issue, even though the overall accuracy of all our predictions is satisfactory, the f1 scores for minority classes still have a large space for improvement. Accuracy metric could be misleading for imbalance data, while precision, recall or f1 score are better for interpreting the model performance. In other words, the high accuracy is mainly due to correct prediction for the majority class, while the prediction result for minority classes is not ideal. Thus, we may want to try some other CNN models and other oversampling methods to get higher f1 scores for the minority classes.

Moreover, since there are not many images containing volcanoes, we tried to do data augmentation. However, when we tried to fit the augmented data, we observed a huge fluctuation in the accuracy of the validation set no matter how we tuned the parameters (as shown in our Jupyter notebook). If more time is given, we may investigate the reason behind this inconsistency.

Furthermore, we experimented the VGG16 model on volcano detection, uncertainty prediction and the number of volcanoes prediction. The results and the plots in our Jupyter notebook shows that applying VGG16 model in these three prediction tasks has similar behaviors to the results discussed above for data augmentation – the test accuracy fluctuates during training. Possible reasons could be overfitting or no batch normalization in VGG16. Even though the prediction shown in our code is high, similar prediction accuracy is not easily reproducible due to unstable validation accuracy and loss. Training for more epochs may solve this problem.

Also, the images we dealt with are not of high quality. The image size is relatively small in this dataset, while the real astronomical images are actually big. Meanwhile, most of them are noisy and of low contrast. Denoising and contrast enhancement or other image processing methods can be done before images being input into models. But this potentially causes the loss of information. Instead, these methods can be used for data augmentation. We may expect that neural network is able to extract more good features and perform better with respect to different labels when the image quality is higher.

In conclusion, we are very pleased with our results given by our CNN model. The fact that current technologies can achieve such amazing performance motivates us to create better model architectures for future generations.

References

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority oversampling technique. *Journal of artificial intelligence research*, *16*, 321–357.

Dheeru, D. & Karra Taniskidou, E. (2017). UCI machine learning repository. Retrieved from http://archive.ics.uci.edu/ml/datasets/volcanoes+on+venus+-+jartool+experiment

Simonyan, K. & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Zhen, H. (2018). Deep cnn 0.97 accuracy with >0.9 recall. Retrieved from https://www.kaggle.com/hotrank/deep-cnn-0-97-accuracy-with-0-9-recall/notebook