Detection of Volcanoes on Venus using Anomaly Detection

Group Number: 32
Team members:
Ruchitha Mekala
Lahari Pokala
Harish Paul Thavisi
Nagavenkata Jagan Chandanada

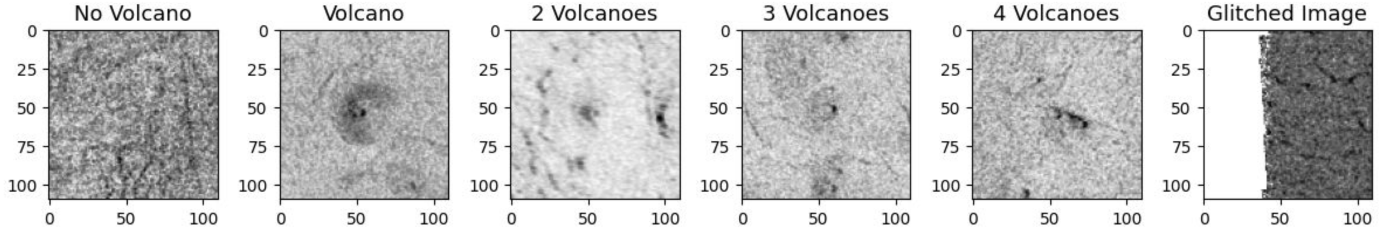
Motivation

- Geological evidence of recent volcanic activity found on Venus surface, discovered by studying old radar images from NASA's Magellan mission.
- Rockets can be launched to send probes and rovers to Venus, to study the geology and atmosphere. Example: NASA is working on VERITAS that will assist scientists obtain a better understanding of Venus.
- Venus has a large number of volcanoes, which makes identifying safe areas for exploration difficult and time-consuming.
- Given an image of Venus' land surface, our goal is to predict if there are any volcanoes present in the image or not.

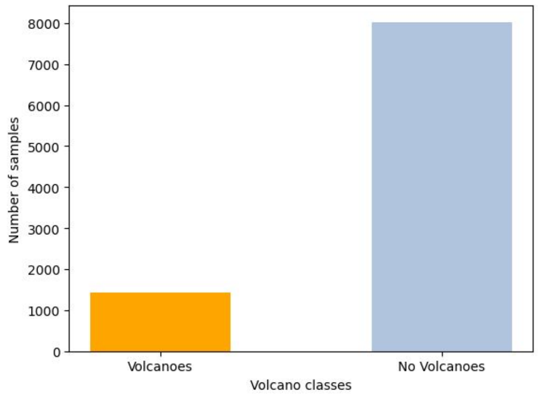
Related Work

Paper	Key techniques	Results
Đuranović et.al [1]	Data Augmentation technique ➔ Mosaic Augmentation + classical Augmentation techniques Image Detection Algorithm - YoloV5	Precision = 0.896 Recall = 0.769 mAP@0.5 = 0.835 F1-Score = 0.826
Burl et.al [2]	Three step process involved in the Automated system 1. FOA(Focus of Attention) 2. PCA 3. Classification technique - Quadratic Classifier	Experimental results show that the system approaches human performance on Homogenous image sets but performs relatively poorly on Heterogenous image sets.
Scofield et.al [3]	Data Augmentation technique ➔ Increased the data size by shifting the volcano position in the image. Feature Extraction Algorithms ➔ Frozen Dictionary Learning with sparse encoding Classification Algorithms ➔ Feed Forward NN	Precision = 0.0173 Recall = 0.9456

Dataset



- The dataset we used to identify volcanoes on Venus were obtained from Magellan imaging which was gathered by the Magellan spacecraft over a four-year period, from 1990 to 1994
- The images are 110X110 pixels with pixel values are in the range [0,255]
- Labels indicate some subjective ambiguity (1 = definitely a volcano, 2 = probably, 3 = potentially, 4 = simply a pit is seen)
- There is no absolute ground truth for this collection of data
- There is high Imbalance in the data



Methodology

1. The class imbalance in the data stimulated the idea of detecting volcanoes using Anomaly Detection approach that incorporated reconstruction error method.
2. PCA and Autoencoders have been implemented.
3. Data can be interpreted as ➔
    - Non volcanic Data - Normal Data
    - Volcanic Data - Anomalous Data
4. For this method, the dataset was split into the following manner ➔
    - Training Data - Non volcanic images (6608 images)
    - Testing Data - Non volcanic images (1419 images) + Volcanic images (1419 images)

Methodology (Cont.)

- PCA and Autoencoders have been fit to training data (i.e Non-volcanic Data)
- The Autoencoder architecture (for encoding dimensions - 3) is shown in the figure on right.

Layer (type)	Output Shape	Param #
input_4 (InputLayer)	[(None, 12100)]	0
dense_18 (Dense)	(None, 54)	653454
dense_19 (Dense)	(None, 28)	1540
dense_20 (Dense)	(None, 3)	87
dense_21 (Dense)	(None, 28)	112
dense_22 (Dense)	(None, 54)	1566
dense_23 (Dense)	(None, 12100)	665500

Total params: 1,322,259
Trainable params: 1,322,259
Non-trainable params: 0

Architecture of Autoencoder

Methodology (Cont.)

- The testing data has been transformed to lower dimensions using the fitted model and has been reconstructed back into the original data.
- The squared error has been calculated for the reconstructed image and the original image.
- An error threshold has been set to flag a particular image as an anomaly or not aka Volcanic or Non-volcanic ➔ An image has been flagged as
    - Anomaly (Volcanic image) if the reconstruction error is > threshold
    - Normal (Non-volcanic image) if the reconstruction error is < threshold

Methodology (Cont.)

- CNN model has also been implemented and is considered as a base model.
- The architecture of CNN is shown in the figure
- Model implementation details -
    - Epochs - 10
    - Loss function - Categorical Cross Entropy
    - Optimizer - Adam Optimizer

Layer (type)	Output Shape	Param #
conv2d_10 (Conv2D)	(None, 110, 110, 8)	208
max_pooling2d_10 (MaxPooling2D)	(None, 55, 55, 8)	0
dropout_8 (Dropout)	(None, 55, 55, 8)	0
conv2d_11 (Conv2D)	(None, 55, 55, 16)	1168
max_pooling2d_11 (MaxPooling2D)	(None, 27, 27, 16)	0
dropout_9 (Dropout)	(None, 27, 27, 16)	0
flatten_5 (Flatten)	(None, 11664)	0
dense_4 (Dense)	(None, 2)	23330

Total params: 24,706
Trainable params: 24,706
Non-trainable params: 0

Architecture of the CNN model

Results

- The performance of the models have been evaluated using Training accuracy, Precision, Recall and F1 Score.
- PCA and Autoencoders have been evaluated individually.
- The lower dimensions components in PCA and Autoencoders (Encoding dimensions) have been varied to see the trend and record the results

Results (Cont.) - Grid resulted from varying PCA components and Error Threshold value

Threshold

PCA		100	110	120	130	140	150	160	170	180	190	200
2	P	0.521	0.525	0.525	0.526	0.515	0.499	0.480	0.462	0.454	0.455	0.449
	R	0.973	0.934	0.869	0.779	0.672	0.576	0.483	0.408	0.347	0.293	0.246
	F1	0.678	0.672	0.659	0.628	0.583	0.535	0.482	0.433	0.393	0.356	0.318
	T_A	0.089	0.145	0.208	0.280	0.352	0.415	0.475	0.534	0.587	0.632	0.678
3	P	0.522	0.527	0.532	0.528	0.509	0.494	0.477	0.457	0.457	0.448	0.446
	R	0.966	0.927	0.858	0.761	0.645	0.550	0.462	0.382	0.325	0.271	0.224
	F1	0.678	0.672	0.657	0.623	0.569	0.520	0.470	0.416	0.380	0.337	0.298
	T_A	0.101	0.157	0.224	0.302	0.369	0.434	0.495	0.553	0.608	0.655	0.704
4	P	0.522	0.526	0.532	0.521	0.503	0.490	0.469	0.456	0.450	0.442	0.436
	R	0.962	0.914	0.847	0.736	0.622	0.531	0.440	0.369	0.307	0.257	0.206
	F1	0.677	0.668	0.653	0.611	0.556	0.510	0.454	0.408	0.365	0.325	0.280
	T_A	0.106	0.163	0.230	0.309	0.376	0.443	0.504	0.563	0.622	0.665	0.719
5	P	0.522	0.527	0.530	0.522	0.501	0.489	0.467	0.453	0.446	0.442	0.430
	R	0.959	0.911	0.835	0.727	0.608	0.521	0.425	0.356	0.295	0.250	0.197
	F1	0.959	0.911	0.835	0.727	0.608	0.521	0.425	0.356	0.295	0.250	0.197
	T_A	0.109	0.167	0.238	0.317	0.384	0.454	0.515	0.576	0.631	0.677	0.729
6	P	0.523	0.527	0.530	0.520	0.499	0.485	0.465	0.454	0.441	0.444	0.429
	R	0.956	0.904	0.828	0.715	0.599	0.510	0.412	0.345	0.284	0.242	0.188
	F1	0.676	0.666	0.646	0.602	0.544	0.497	0.437	0.392	0.345	0.313	0.262
	T_A	0.113	0.173	0.245	0.323	0.393	0.460	0.529	0.586	0.638	0.688	0.738

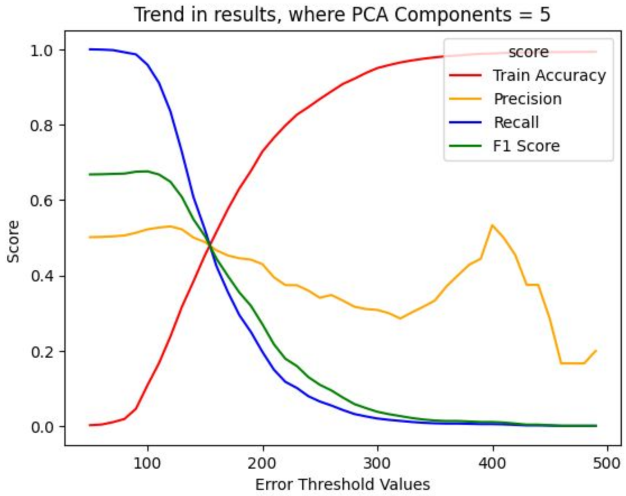
Results (Cont.) - Grid resulted from varying Autoencoder Dimensions and Error Threshold value

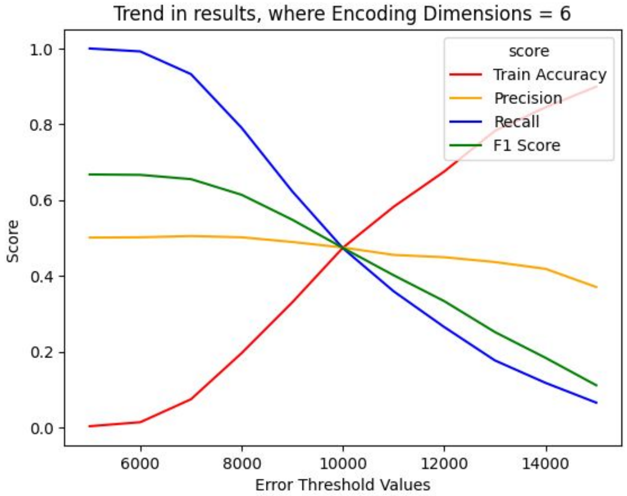
Threshold

Dimensions		5000	6000	7000	8000	9000	1000	11000	12000	13000	14000	15000
2	P	0.501	0.501	0.503	0.501	0.49	0.478	0.465	0.453	0.436	0.434	0.403
	R	1	0.992	0.933	0.791	0.624	0.473	0.36	0.266	0.177	0.118	0.066
	F1	0.667	0.666	0.654	0.613	0.549	0.475	0.406	0.335	0.252	0.186	0.114
	Train_Acc	0.003	0.014	0.076	0.197	0.328	0.473	0.579	0.673	0.783	0.843	0.896
3	P	0.501	0.502	0.503	0.501	0.484	0.475	0.474	0.462	0.468	0.443	0.416
	R	1	0.992	0.932	0.791	0.623	0.473	0.36	0.265	0.177	0.118	0.066
	F1	0.668	0.667	0.654	0.613	0.545	0.474	0.409	0.337	0.257	0.186	0.114
	Train_Acc	0.003	0.014	0.076	0.197	0.332	0.474	0.576	0.671	0.777	0.841	0.895
4	P	0.501	0.501	0.505	0.502	0.484	0.47	0.46	0.447	0.451	0.436	0.399
	R	1	0.992	0.932	0.79	0.622	0.473	0.359	0.264	0.177	0.118	0.066
	F1	0.667	0.666	0.655	0.614	0.544	0.471	0.403	0.332	0.254	0.185	0.113
	Train_Acc	0.004	0.015	0.076	0.197	0.334	0.478	0.582	0.677	0.781	0.843	0.897
5	P	0.501	0.501	0.504	0.494	0.476	0.462	0.448	0.426	0.422	0.404	0.371
	R	1	0.992	0.932	0.79	0.622	0.473	0.359	0.264	0.177	0.118	0.066
	F1	0.668	0.666	0.654	0.608	0.54	0.468	0.398	0.326	0.249	0.182	0.111
	Train_Acc	0.003	0.014	0.077	0.202	0.338	0.481	0.587	0.684	0.786	0.847	0.9
6	P	0.501	0.502	0.505	0.502	0.489	0.475	0.455	0.449	0.437	0.419	0.371
	R	1	0.992	0.932	0.791	0.623	0.473	0.359	0.265	0.177	0.118	0.066
	F1	0.668	0.667	0.655	0.614	0.548	0.474	0.402	0.333	0.252	0.184	0.111
	Train_Acc	0.003	0.014	0.074	0.196	0.33	0.474	0.582	0.675	0.783	0.845	0.899

Results (Cont.)

- The scores have been plotted against the threshold values for one particular pca component (PCA) and encoding dimension (Autoencoders)
- It is evident from the graphs that if training accuracy is maximized, test accuracy is being compromised and vice versa.
- The trade off can be chosen based on the requirement.





Results (Cont.)

Model	Precision	Recall	F1 Score
CNN (Baseline)	0.81	0.54	0.648
Anomaly Detection - PCA	0.51	0.608	0.61
Anomaly Detection - Autoencoders	0.48	0.623	0.54

Results (Cont.)

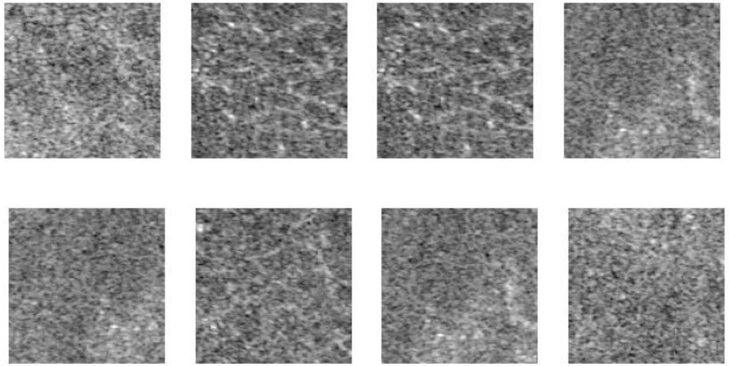


Fig 1 : Non-Volcanic Data

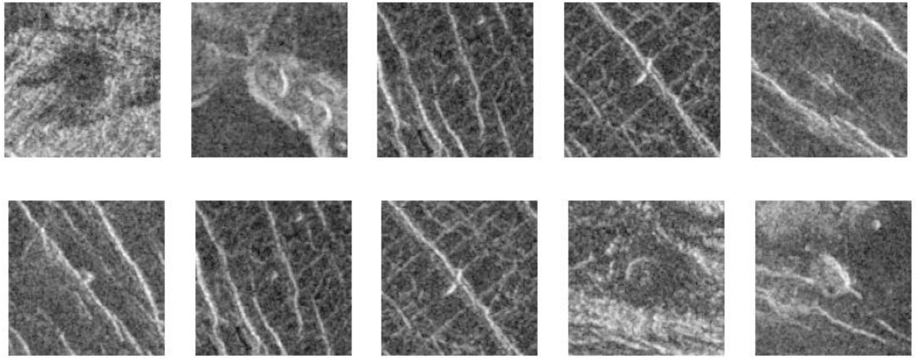


Fig 2 : Volcanic Data that were flagged as ‘**Anomaly**’

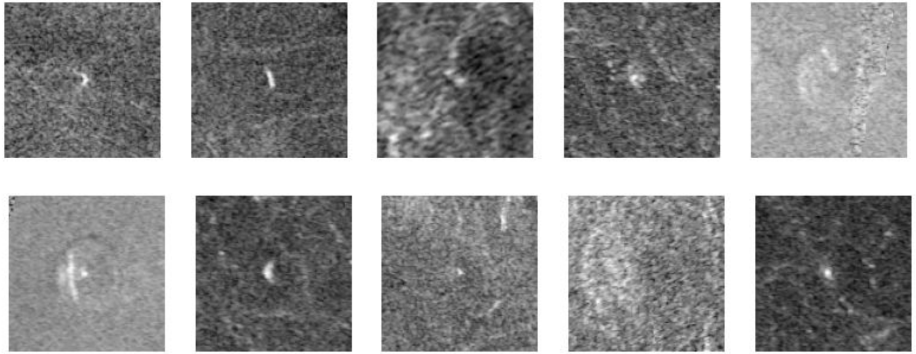


Fig 3 : Volcanic Data that were flagged as ‘**Normal**’

Insights & Conclusion

- Few images in volcano data (Fig 3), share the major portion of the image with training data - non volcanic data, hence the small speck which is labeled as volcano is not captured by the anomaly detector.
- The results haven’t met the expectations.
- Although there is an imbalance in the data, it can be concluded that it is challenging to implement anomaly detector for this classification task since the dataset isn’t appropriate to the objective of ‘Anomaly Detection’.

Future Work

- More sophisticated error metrics can be looked into.
- Although few corrupted images have been detected and eliminated, there seem to be still few corrupted images in the data. Need to figure out to completely get rid of those images.

References

[1] Daniel Đuranović, Sandi Baressi Šegota, Ivan Lorencin, and Zlatan Car. Localization and classification of venusian volcanoes using image detection algorithms. Sensors, 23(3), 2023

[2] Chael Burl, Lars Asker, Padhraic Smyth, Usama Fayyad, Pietro Perona, L. Crumpler, and Jayne Aubele. Learning to recognize volcanoes on venus. Machine Learning, 30:165–194, 02 1998.

[3] T. Scofield and B. M. Whitaker. "Machine learning pipeline for shift-invariant detection of volcanoes on venus". 2020 Intermountain engineering, Technology and Computing (IETC), pages 1–6, 2020.