

Pràctica 2. Neteja i anàlisi de dades. Universitat Oberta de Catalunya.

DrCyZ: Techniques for analyzing and extracting useful information from CyZ.

J. de Curtò i DíAz.

I. de Zarzà i Cubero.

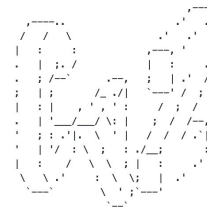
04-01-2022.

c@decurto.be

z@dezarza.be



DrCyZ: Techniques for analyzing and extracting useful information from CyZ.



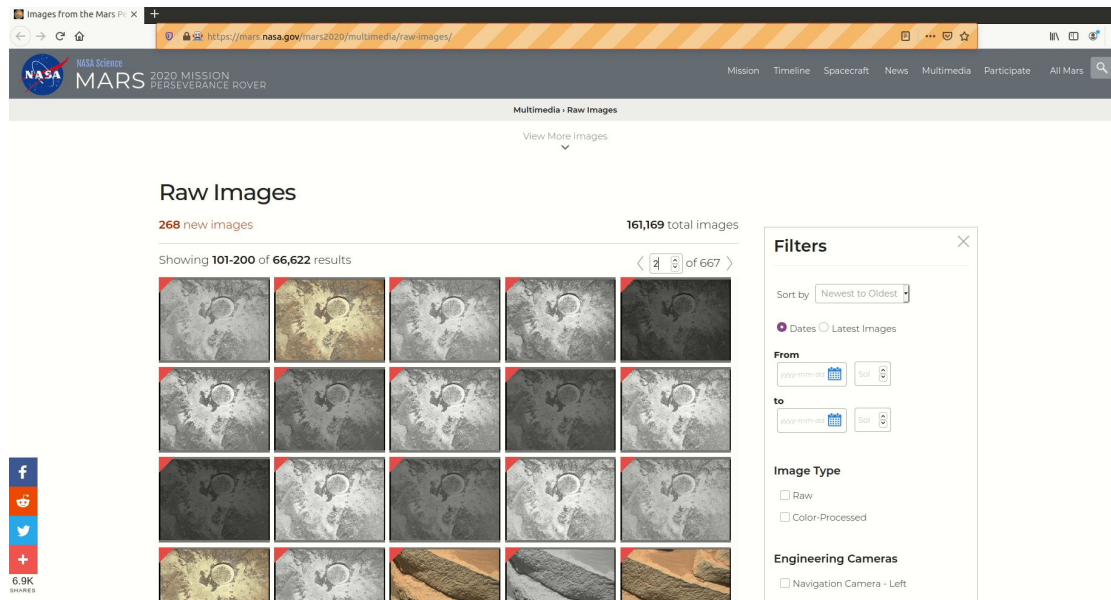
De Curtò i DíAz.
De Zarzà i Cubero.

<https://github.com/decurtoidiaz/cyz>

<https://github.com/decurtoidiaz/drcyz>

c@decurto.be

z@dezarza.be



DrCyZ: Techniques for analyzing and extracting useful information from CyZ.



De Curtò i DíAz.
De Zarzà i Cubero.

K-means Clustering with PCA(2)

```
from sklearn.decomposition import PCA

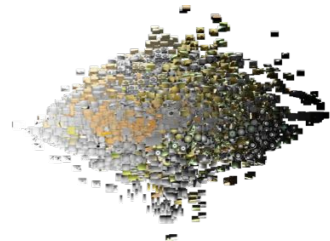
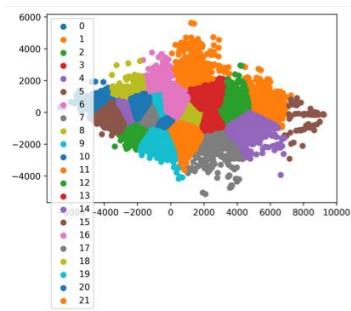
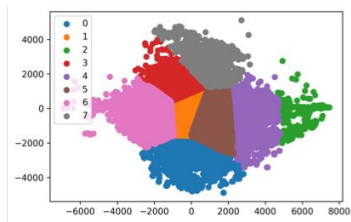
features = np.array(features)
pca = PCA(2)

#Transform the data
df = pca.fit_transform(features)

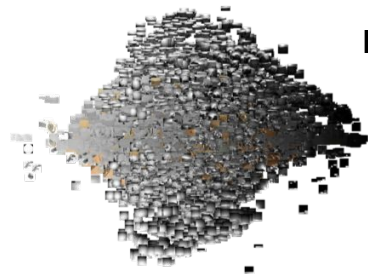
n_cams = 8
kmeans = MiniBatchKMeans(n_clusters=n_cams)

#predict the labels of clusters.
label = kmeans.fit_predict(df)

#Getting unique labels
u_labels = np.unique(label)
```



Curiosity



Perseverance

c@decurto.be
z@dezarza.be

DrCyZ: Techniques for analyzing and extracting useful information from CyZ.

**t-SNE with
PCA (explaining 99%
of variance)**

c@decurto.be
z@dezarza.be

```
from sklearn.decomposition import PCA

features = np.array(features)
pca = PCA(n_components=0.99, svd_solver='full')
pca.fit(features)
pca_features = pca.transform(features)

print(pca.explained_variance_)
print(pca.explained_variance_ratio_)
print(pca.explained_variance_ratio_.cumsum())
print(pca.n_components_)

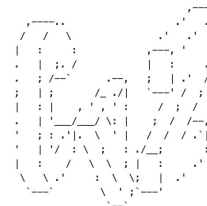
[3.32881889e+06 1.68059055e+06 1.08131421e+06 ... 1.01072405e+02
 1.01004887e+02 1.08862950e+02]
[3.33581389e-01 1.68412206e-01 1.08358643e-01 ... 1.01284794e-05
 1.01217103e-05 1.01074898e-05]
[0.33358139 0.5019936 0.61035224 ... 0.98998276 0.98999289 0.99000299]
1919
```

```
from sklearn.decomposition import PCA

features = np.array(features)
pca = PCA(n_components=0.99, svd_solver='full')
pca.fit(features)
pca_features = pca.transform(features)

print(pca.explained_variance_)
print(pca.explained_variance_ratio_)
print(pca.explained_variance_ratio_.cumsum())
print(pca.n_components_)

[7.16016904e+06 9.08974590e+05 4.84503908e+05 ... 1.23337673e+02
 1.23188237e+02 1.23049858e+02]
[5.87204089e-01 7.45448316e-02 3.97340725e-02 ... 1.01148989e-05
 1.01026436e-05 1.00912952e-05]
[0.58720409 0.66174892 0.70148299 ... 0.98998802 0.98999812 0.99000821]
2031
```



De Curtò i DíAz.
De Zarzà i Cubero.

Curiosity

Perseverance

DrCyZ: Techniques for analyzing and extracting useful information from CyZ.



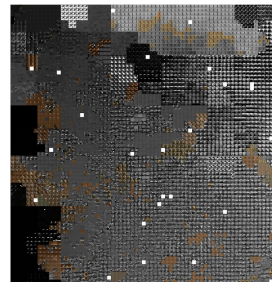
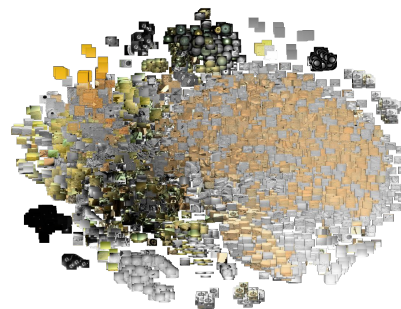
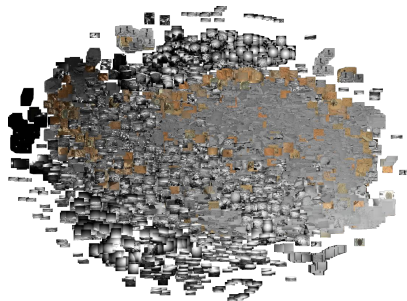
De Curtò i DíAz.
De Zarzà i Cubero.

t-SNE with PCA (explaining 99% of variance)

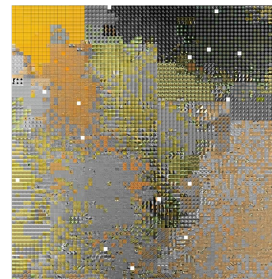
```
num_images_to_plot = 22717

if len(images) > num_images_to_plot:
    sort_order = sorted(random.sample(range(len(images)), num_images_to_plot))
    images = [images[i] for i in sort_order]
    pca_features = [pca_features[i] for i in sort_order]

[ ] X = np.array(pca_features)
tsne = TSNE(n_components=2, learning_rate=150, perplexity=30, angle=0.2, verbose=2).fit_transform(X)
```



Curiosity



Perseverance

c@decurto.be
z@dezarza.be

DrCyZ: Techniques for analyzing and extracting useful information from CyZ.



De Curtò i DíAz.
De Zarzà i Cubero.

**Stylegan2-ada training
with subset of samples
(drcyz - 5025 corresponding terrain
pictures to ease convergence)
from Perseverance.**

```
training_path = project_path / 'training' / dataset_name
if not training_path.is_dir():
    mkdir(training_path)

#how often should the model generate samples and a .pkl file
snapshot_count = 2
#should the images be mirrored left to right?
mirrored = True
#should the images be mirrored top to bottom?
mirroredY = False
#metrics?
metric_list = None
#augments
augs = 'bgr'

resume_from = 'noresume'

!python "{stylegan2_repo_path} / 'train.py'" --outdir="{training_path}" \
  --data="{local_dataset_path}" --resume="{resume_from}" \
  --snap={snapshot_count} --augs={augs} \
  --mirror={mirrored} --mirrorY={mirroredY} --cfg='{auto}' \
  --metrics={metric_list} #--dry-run
```

c@decurto.be
z@dezarza.be



**Grid of 100
synthetic samples.**

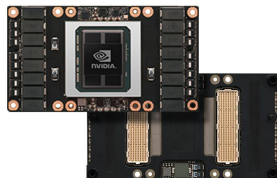
DrCyZ: Techniques for analyzing and extracting useful information from CyZ.



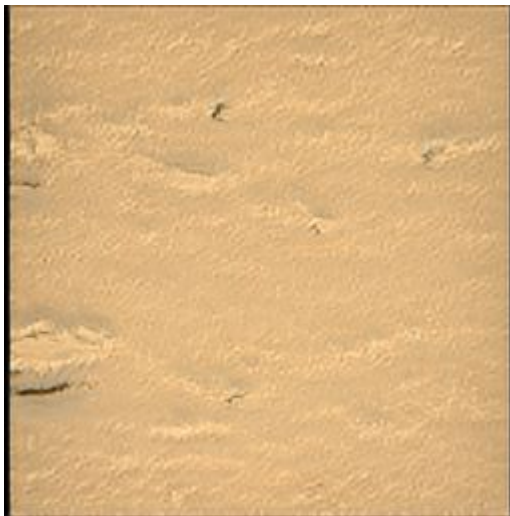
De Curtò i DíAz.
De Zarzà i Cubero.

**Stylegan2-ada training
with subset of samples
from Perseverance.**

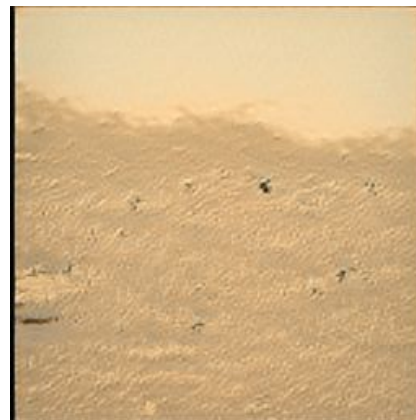
**1 x NVIDIA Tesla P-100
around 48h.**



c@decurto.be
z@dezarza.be



**Two frames exploring
z-sphere latent space.**



DrCyZ: Techniques for analyzing and extracting useful information from CyZ.



De Curtò i DíAz.
De Zarzà i Cubero.

Statistical comparison of the samples by mean intensity (rgb and grey scale).

```
# We propose to compute mean of pixel intensity for statistics
mean_features = []
for c in features:
    mean_features.append(c.mean())

mean_features2 = []
for c2 in features2:
    mean_features2.append(c2.mean())

print(mean_features)
print(mean_features2)
```

```
[136.9560546875, 153.189208984375, 132.62744140625, 109.865478515625, 151.930908203125, 147.08154296875, 158.233642578125, 175.22705078125, 135.7158203125, 163.289306640625, 182.438720703125, 174.16552734375, 164.75390625, 160.764892578125, 160.10498046875, 158.564697265625, 187.696533203125, 157.413330078125]
```

```
[ ] # Test homogeneity of variance
from scipy import stats

print(scipy.stats.levene(mean_features, mean_features2)) #Levene
print(scipy.stats.levene(mean_features, mean_features2, center='mean')) #Brown-Forsyth
print(scipy.stats.bartlett(mean_features, mean_features2)) #Bartlett

LeveneResult(statistic=156.12869754877337, pvalue=2.1729631188690332e-35)
LeveneResult(statistic=166.13819743144077, pvalue=1.611279115945387e-37)
BartlettResult(statistic=874.8317447750159, pvalue=2.905247248253299e-192)
```

```
# Test normality
print(scipy.stats.normaltest(mean_features)) #Normality of original samples
print(scipy.stats.normaltest(mean_features2)) #Normality of generated samples

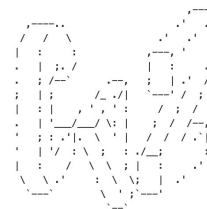
NormaltestResult(statistic=2571.783381290372, pvalue=0.0)
NormaltestResult(statistic=0.11406195050626958, pvalue=0.9445648107792636)
```

```
[ ] #Test goodness of fit (non-parametric)
print(scipy.stats.kruskal(mean_features, mean_features2)) #Kruskal-Wallis

KruskalResult(statistic=30.97421391554919, pvalue=2.6147941787875364e-08)
```

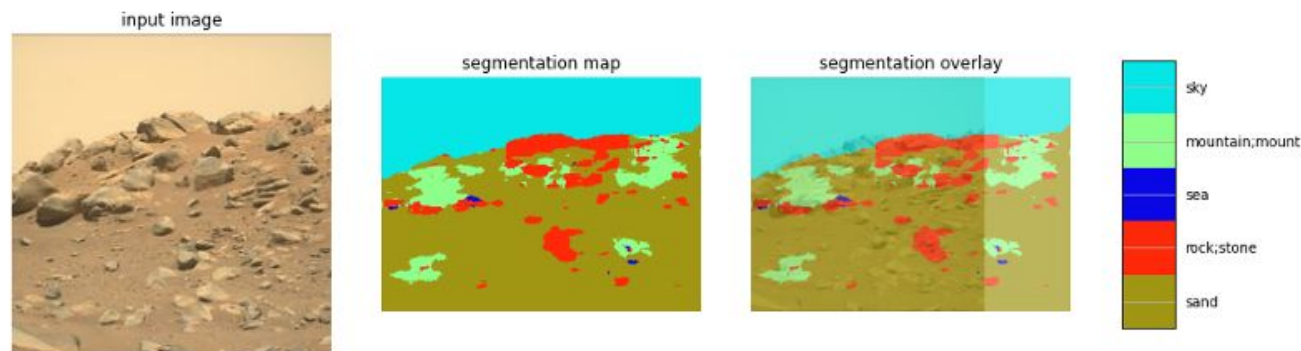
c@decurto.be
z@dezarza.be

DrCyZ: Techniques for analyzing and extracting useful information from CyZ.



De Curtò i DíAz.
De Zarzà i Cubero.

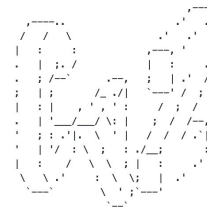
**Example of instance segmentation
using Deeplab on a sample from
Perseverance.**



c@decurto.be

z@dezarza.be

DrCyZ: Techniques for analyzing and extracting useful information from CyZ.



De Curtò i DíAz.
De Zarzà i Cubero.

<https://github.com/decurtoidiaz/cyz>

<https://github.com/decurtoidiaz/drcyz>

<https://doi.org/10.5281/zenodo.5655473>

DOI 10.5281/zenodo.5655473

c@decurto.be

z@dezarza.be

November 8, 2021

Dataset Open Access

CyZ: MARS Space Exploration Dataset.

de Curtò, J.; de Zarzà, I.

Images from NASA missions of the celestial body.

Repository: <https://github.com/decurtoidiaz/cyz>

Authors:

J. de Curtò c@decurto.be

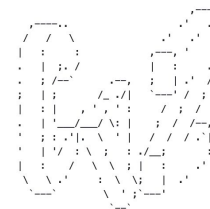
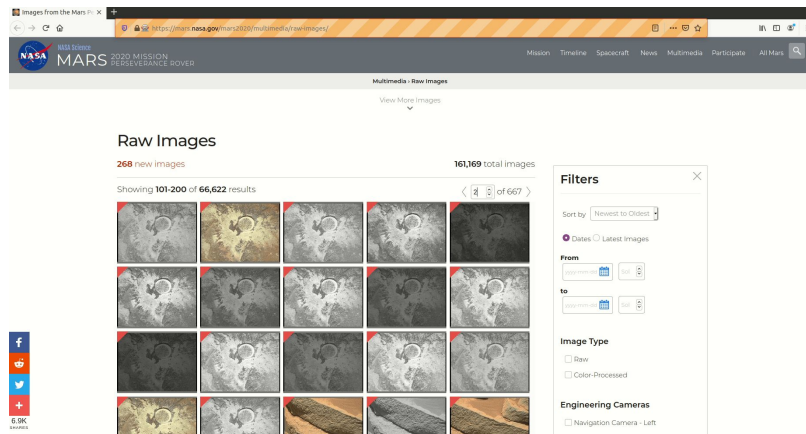
I. de Zarzà z@dezarza.be

File Information from CyZ-1.1

- Curiosity ([cyz/curiosity_cyz](#)).
 - [png \(cyz/curiosity_cyz/png\)](#).
 - PNG files for the corresponding cameras.
 - [csv \(cyz/curiosity_cyz/csv\)](#).
 - CSV files.
- Perseverance ([cyz/perseverance_cyz](#)).
 - [png \(cyz/perseverance_cyz/png\)](#).
 - PNG files for the corresponding cameras.
 - [csv \(cyz/perseverance_cyz/csv\)](#).
 - CSV files.

DrCyZ: Techniques for analyzing and extracting useful information from CyZ.

<https://github.com/decurtoidiaz/drcyz>



De Curtò i DíAz.
De Zarzà i Cubero.

c@decurto.be
z@dezarza.be

