

```
In [22]: # Libraries import
import numpy as np
import sklearn
import scipy.io as sio
from sklearn.decomposition import PCA
from sklearn import preprocessing
import matplotlib.pyplot as plt
```

In [23]: *# import the datasets*

```
indian_pines = sio.loadmat(r"C:\Users\DELL\Downloads\archive (6)\Indian_pines_cor
print(indian_pines)
```

```
{'__header__': b'MATLAB 5.0 MAT-file, Platform: GLNXA64, Created on: Fri May 20
18:47:44 2011', '__version__': '1.0', '__globals__': [], 'indian_pines_correcte
d': array([[[3172, 4142, 4506, ..., 1057, 1020, 1020],
           [2580, 4266, 4502, ..., 1064, 1029, 1020],
           [3687, 4266, 4421, ..., 1061, 1030, 1016],
           ...,
           [2570, 3890, 4320, ..., 1042, 1021, 1015],
           [3170, 4130, 4320, ..., 1054, 1024, 1020],
           [3172, 3890, 4316, ..., 1043, 1034, 1016]],

           [[2576, 4388, 4334, ..., 1047, 1030, 1006],
           [2747, 4264, 4592, ..., 1055, 1039, 1015],
           [2750, 4268, 4423, ..., 1047, 1026, 1015],
           ...,
           [3859, 4512, 4605, ..., 1056, 1035, 1015],
           [3686, 4264, 4690, ..., 1051, 1012, 1020],
           [2744, 4268, 4597, ..., 1047, 1019, 1016]],

           [[2744, 4146, 4416, ..., 1055, 1029, 1025],
           [2576, 4389, 4416, ..., 1051, 1021, 1011],
           [2744, 4273, 4420, ..., 1068, 1033, 1010],
           ...,
           [2570, 4266, 4509, ..., 1051, 1025, 1010],
           [2576, 4262, 4496, ..., 1047, 1029, 1020],
           [2742, 4142, 4230, ..., 1042, 1025, 1011]],

           ...,

           [[3324, 3728, 4002, ..., 1003, 1004, 1004],
           [2983, 3604, 3829, ..., 1011, 1013, 1008],
           [2988, 3612, 3913, ..., 1012, 1001, 1004],
           ...,
           [2564, 4115, 4103, ..., 1003, 1005, 1013],
           [2730, 4111, 4103, ..., 1015, 1013, 1004],
           [3156, 3991, 4103, ..., 1017, 1014, 1000]],

           [[3161, 3731, 3834, ..., 1002, 1000, 1000],
           [2727, 3742, 4011, ..., 999, 991, 1003],
           [2988, 4114, 4011, ..., 1006, 1008, 1013],
           ...,
           [3156, 3858, 4016, ..., 1011, 1004, 1003],
           [3159, 3858, 4100, ..., 1016, 1000, 1000],
           [2561, 3866, 4003, ..., 1008, 1008, 1000]],

           [[2979, 3728, 3732, ..., 1006, 1004, 1000],
           [2977, 3728, 3741, ..., 1007, 1009, 990],
           [2814, 3728, 3914, ..., 999, 1009, 1003],
           ...,
           [3153, 3864, 4282, ..., 1003, 1008, 1000],
           [3155, 4104, 4106, ..., 1011, 1005, 1003],
           [3323, 3860, 4197, ..., 1007, 1004, 1000]]], dtype=uint16))
```

```
In [24]: indian_pines_key=list(indian_pines.keys())
indian_pines_IN = (indian_pines[indian_pines_key[3]])
print(indian_pines_IN)
```

```
[[[3172 4142 4506 ... 1057 1020 1020]
  [2580 4266 4502 ... 1064 1029 1020]
  [3687 4266 4421 ... 1061 1030 1016]
  ...
  [2570 3890 4320 ... 1042 1021 1015]
  [3170 4130 4320 ... 1054 1024 1020]
  [3172 3890 4316 ... 1043 1034 1016]]

[[[2576 4388 4334 ... 1047 1030 1006]
  [2747 4264 4592 ... 1055 1039 1015]
  [2750 4268 4423 ... 1047 1026 1015]
  ...
  [3859 4512 4605 ... 1056 1035 1015]
  [3686 4264 4690 ... 1051 1012 1020]
  [2744 4268 4597 ... 1047 1019 1016]]

[[[2744 4146 4416 ... 1055 1029 1025]
  [2576 4389 4416 ... 1051 1021 1011]
  [2744 4273 4420 ... 1068 1033 1010]
  ...
  [2570 4266 4509 ... 1051 1025 1010]
  [2576 4262 4496 ... 1047 1029 1020]
  [2742 4142 4230 ... 1042 1025 1011]]

...

[[[3324 3728 4002 ... 1003 1004 1004]
  [2983 3604 3829 ... 1011 1013 1008]
  [2988 3612 3913 ... 1012 1001 1004]
  ...
  [2564 4115 4103 ... 1003 1005 1013]
  [2730 4111 4103 ... 1015 1013 1004]
  [3156 3991 4103 ... 1017 1014 1000]]

[[[3161 3731 3834 ... 1002 1000 1000]
  [2727 3742 4011 ... 999 991 1003]
  [2988 4114 4011 ... 1006 1008 1013]
  ...
  [3156 3858 4016 ... 1011 1004 1003]
  [3159 3858 4100 ... 1016 1000 1000]
  [2561 3866 4003 ... 1008 1008 1000]]

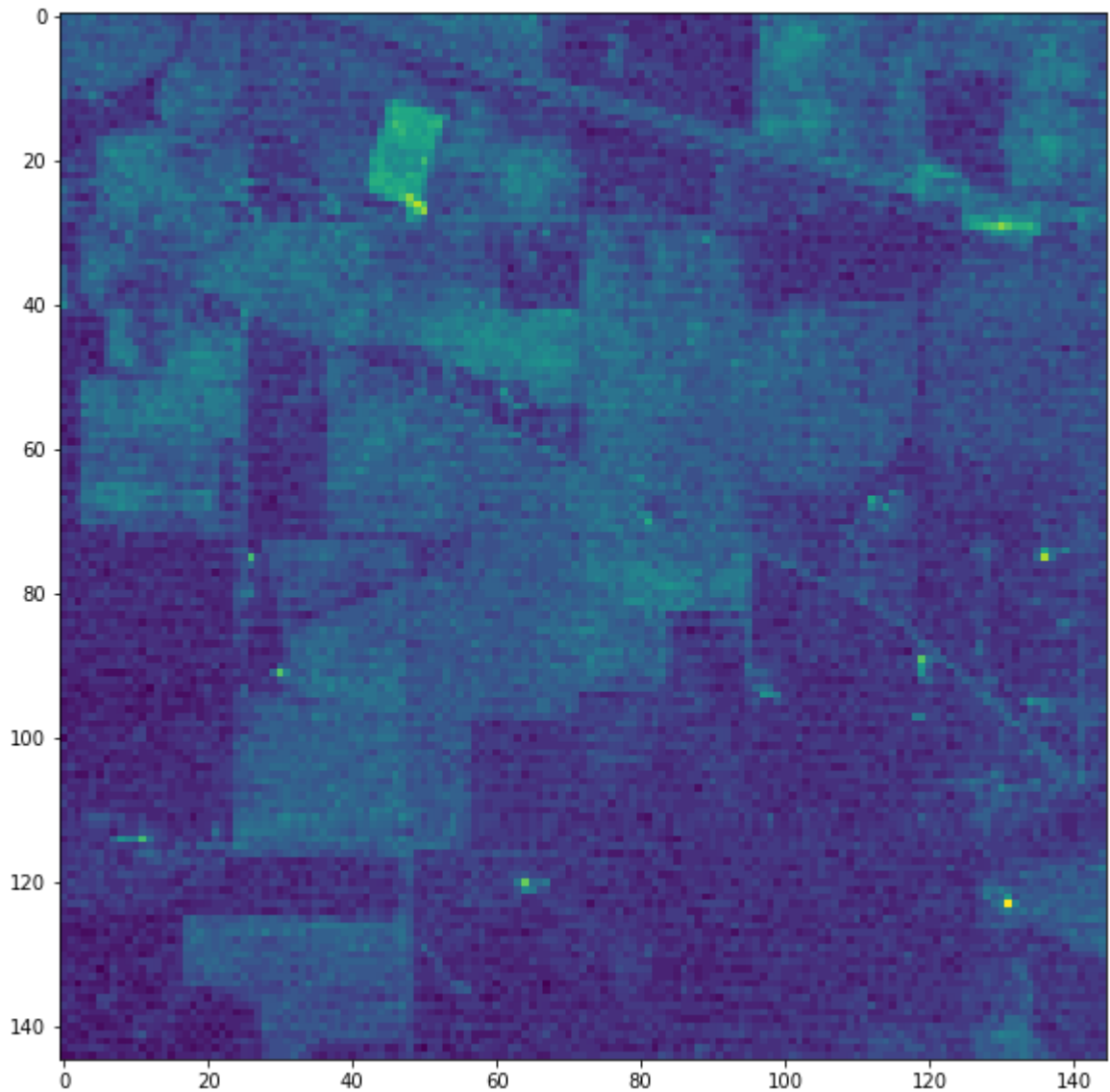
[[[2979 3728 3732 ... 1006 1004 1000]
  [2977 3728 3741 ... 1007 1009 990]
  [2814 3728 3914 ... 999 1009 1003]
  ...
  [3153 3864 4282 ... 1003 1008 1000]
  [3155 4104 4106 ... 1011 1005 1003]
  [3323 3860 4197 ... 1007 1004 1000]]]
```

```
In [25]: print(indian_pines_IN.shape)
```

```
(145, 145, 200)
```

```
In [26]: # plotting the data sets
```

```
fig = plt.figure(figsize = (10,10))  
plt.imshow(indian_pines_IN[:, :, 2], interpolation='nearest')  
plt.show()
```



```
In [27]: indian_pine_data = indian_pines_IN.reshape(np.prod(indian_pines_IN.shape[:2]), np.
```

```
# New shape of the data is  
print(indian_pine_data.shape)
```

```
(21025, 200)
```

```
In [28]: print(indian_pine_data)
```

```
[[3172 4142 4506 ... 1057 1020 1020]
 [2580 4266 4502 ... 1064 1029 1020]
 [3687 4266 4421 ... 1061 1030 1016]
 ...
 [3153 3864 4282 ... 1003 1008 1000]
 [3155 4104 4106 ... 1011 1005 1003]
 [3323 3860 4197 ... 1007 1004 1000]]
```

```
In [29]: from sklearn.preprocessing import StandardScaler
indian_pine_data = StandardScaler().fit_transform(indian_pine_data)
```

```
In [30]: print(indian_pine_data.shape)
```

```
(21025, 200)
```

```
In [31]: print(indian_pine_data)
```

```
[[ 0.6047628  0.21997468  0.8862633 ... 1.28472014  0.38894136
  1.62910551]
 [-1.06326446  0.75820528  0.87074869 ... 1.62021647  1.17937664
  1.62910551]
 [ 2.05583381  0.75820528  0.55657785 ... 1.47643233  1.26720278
  1.06179023]
 ...
 [ 0.55122814 -0.98670358  0.01744517 ... -1.30339441 -0.66497235
 -1.20747086]
 [ 0.55686337  0.05503305 -0.66519764 ... -0.91997003 -0.92845077
 -0.7819844 ]
 [ 1.03022246 -1.00406586 -0.31224028 ... -1.11168222 -1.01627691
 -1.20747086]]
```

```
In [32]: # used PCA as a dimensionality reduction-
from sklearn.decomposition import PCA
pca_decompostn = PCA(n_components=40)
indian_pine_data_pca = pca_decompostn.fit_transform(indian_pine_data)
```

```
In [33]: print(pca_decompostn.explained_variance_ratio_)
```

```
[6.86181404e-01 1.91941055e-01 2.56119002e-02 1.68655181e-02
 1.04908555e-02 8.40367362e-03 4.70794054e-03 3.92647117e-03
 3.72137486e-03 3.46935850e-03 2.84682198e-03 2.45589437e-03
 2.14139842e-03 1.97505497e-03 1.85914986e-03 1.80135460e-03
 1.74202170e-03 1.58030126e-03 1.54109122e-03 1.50045539e-03
 1.31749238e-03 1.21411963e-03 1.10724560e-03 1.00049479e-03
 9.57957880e-04 8.79787759e-04 8.37197305e-04 7.97171993e-04
 7.58914310e-04 7.20055704e-04 6.97095361e-04 6.72021623e-04
 6.26064601e-04 6.13136729e-04 5.89741772e-04 5.81621904e-04
 5.18465931e-04 4.92740001e-04 4.58347813e-04 4.46770335e-04]
```

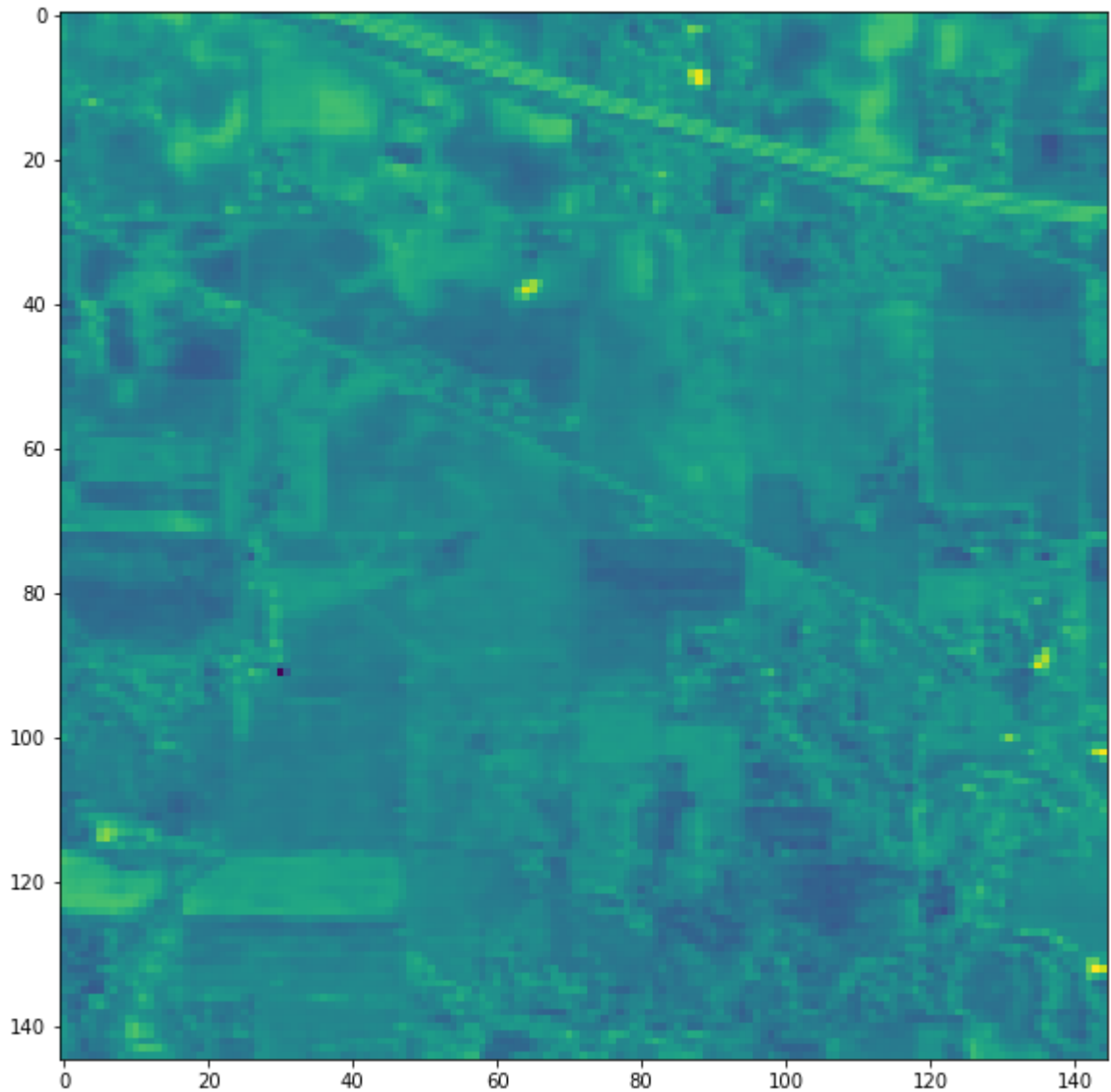
```
In [34]: print(indian_pine_data_pca.shape)
```

```
(21025, 40)
```

```
In [37]: indian_pine_data_pca_new = indian_pine_data_pca.reshape(145,145,40)
print(indian_pine_data_pca_new.shape)
```

```
(145, 145, 40)
```

```
In [38]: fig = plt.figure(figsize = (10,10))
plt.imshow(indian_pine_data_pca_new[:, :, 1], interpolation='nearest')
plt.show()
```



```
In [ ]:
```

