Report of using Fast-ICA on Hyperspectral Image Classification
He Yang

1.1 The definition of ICA

Independent component analysis is a method to find a **linear representation of nongaussian** data so that the components are **statistically independent**, or as independent as possible.

1.2 The general problem model

For n given observation signals $x_i(t), i=1,2,\ldots n$, we assume each of the observation signals is a linear combination of k independent signals $s_j(t)$, which could express as the below linear equation:

$$x_i(t) = a_{i1}s_1(t) + a_{i2}s_2(t) + \ldots + a_{ik}s_k(t) = \sum_{j=1}^{k} a_{ij}s_j(t)$$

The above equation can be expressed use the Matrix form

$$\mathbf{x} = \mathbf{As}$$

Where a_{ij} is the element of $\mathbf{A}$ and $s_j(t)$ is the element of column vector $\mathbf{s}$, we call the matrix $\mathbf{A}$ the mixing matrix. Then the problem is we want infer $\mathbf{s}$ and $\mathbf{A}$ only form the observation $\mathbf{x}$. We estimating the $\mathbf{A}$, and then use the $\mathbf{W}$ which is the inverse of the mixing matrix $\mathbf{A}$ to obtain the independent component $\mathbf{s}$ by

$$\mathbf{s} = \mathbf{Wx}$$

1.3 The basis of ICA

As we only know the observation $\mathbf{x}$, the way we can estimate both unknown components is we apply some restriction on one of the components. The **statistically independent** component $\mathbf{s}$ is the choice. From the Central Limit Theorem, sum of a large number of independent and identically-distributed random variables will be approximately following a Gaussian distribution, so we have reason to guess that the sum of two independent random variables usually has a distribution that closer to Gaussian than any of the two original independent random variables. That's why the ICA model assumes the independent data should nongaussian.

Let a linear combination $y = \mathbf{w}^T\mathbf{x}$, where $\mathbf{w}^T$ is one of the rows of the inverse of $\mathbf{A}$. Then we have $y = \mathbf{w}^T\mathbf{x} = \mathbf{w}^T\mathbf{As} = (\mathbf{A}^T\mathbf{w})^T\mathbf{s} = \mathbf{z}^T\mathbf{s}$, then the y becomes a linear combination of $\mathbf{s}$, for those $\mathbf{w}^T$ maximizes the nongaussianity of the $y = \mathbf{w}^T\mathbf{x}$, the $\mathbf{z}$ only has one nonzero component. Then the y only have one independent component of $\mathbf{s}$.

2 The Fast-ICA Algorithm

2.1 The derivation of Fast-ICA Algorithm

The estimation problem is changed into find $\mathbf{w}^T$ which maximize the nongaussianity of the $y = \mathbf{w}^T\mathbf{x}$. We need have a quantitative measure of nongaussianity of a random variable y .For simplicity, we assume the y has zero mean and unit-variance. Measure to compute the nongaussianity include Kurtosis and negentropy. Here we just discuss about the method used in the Fast-ICA algorithm. The Fast-ICA algorithm try to maximize the negentropy of the $y = \mathbf{w}^T\mathbf{x}$, It use a approximations of the negentropy based on maximum-entropy principle:

$$J(y) \propto [E\{G(y)\} - E\{G(v)\}]^2$$

Where v is a Gaussian variable of zero-mean and unit variance and $G(\bullet)$ is a non-quadratic function.

For simplicity, we assume the $\mathbf{x}$ is already preprocessing which is centering (has zero mean)and whitening $E\{\mathbf{x}\mathbf{x}^T\} = \mathbf{I}$, As the $E\{G(v)\}$ is a constant for a given $G(\bullet)$, so the maxima of the approximation of the negentropy of the $y = \mathbf{w}^T\mathbf{x}$ can be get at the optima of $E\{G(y)\} = E\{G(\mathbf{w}^T\mathbf{x})\}$, under the constraint $E\{(\mathbf{w}^T\mathbf{x})^2\} = \|\mathbf{w}\|^2 = 1$

The above equation can be solved use the Kuhn-Tucker conditions (It can be found in the Wikipedia) the optima can be obtained at the points where

$$\begin{aligned}
&F(\mathbf{w}) = \nabla E\{G(\mathbf{w}^T\mathbf{x})\} - \frac{\beta}{2}\nabla E\{\mathbf{w}^T\mathbf{w}\} = 0\\
&as\\
&\nabla E\{G(\mathbf{w}^T\mathbf{x})\} = E\{\frac{\nabla \mathbf{w}^T\mathbf{x}}{\nabla \mathbf{w}} g(\mathbf{w}^T\mathbf{x})\} = E\{\mathbf{x}g(\mathbf{w}^T\mathbf{x})\}\\
&\nabla E\{\mathbf{w}^T\mathbf{w}\} = \nabla E\{\mathbf{w}^T\mathbf{I}\mathbf{w}\} = E\{2\mathbf{I}\mathbf{w}\} = 2E\{\mathbf{w}\}\\
&so\\
&F(\mathbf{w}) = \nabla E\{G(\mathbf{w}^T\mathbf{x})\} - \frac{\beta}{2}\nabla E\{\mathbf{w}^T\mathbf{w}\} = E\{\mathbf{x}g(\mathbf{w}^T\mathbf{x})\} - \beta\mathbf{w} = 0
\end{aligned}$$

The $g(\bullet)$ is the derivation of the function $G(\bullet)$. Then solve the above equation use the Newton's method. We can find the methods to compute the derivative of vector and matrix in the appendix of the textbook. The Newton's method for nonlinear systems is the stated below:

For $F(\mathbf{x}) = 0$ the solution $\mathbf{x}$ can be approached by iteration

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} - J^{-1}(\mathbf{x}^{(k-1)})F(\mathbf{x}^{(k-1)})$$

Where $J(\mathbf{x})$ is the Jacobian matrix of the $F(\mathbf{x})$

For the above problem we can get the Jacobian matrix of $F(\mathbf{w})$ the

$$J(\mathbf{w}) = E\{\mathbf{x}\mathbf{x}^T g'(\mathbf{w}^T\mathbf{x})\} - \beta$$

Then from the Newton's method we get the $\mathbf{w}$ by

$$\mathbf{w}^+ = \mathbf{w} - \frac{E\{\mathbf{x}g(\mathbf{w}^T\mathbf{x})\} - \beta\mathbf{w}}{E\{\mathbf{x}\mathbf{x}^T g'(\mathbf{w}^T\mathbf{x})\} - \beta}$$

where the $\mathbf{w}^+$ means the update of $\mathbf{w}$. This is the Fast-ICA's iteration. As the data are centered and whitened. We can use the approximation

$$E\{\mathbf{x}\mathbf{x}^T g'(\mathbf{w}^T\mathbf{x})\} \approx E\{\mathbf{x}\mathbf{x}^T\}E\{g'(\mathbf{w}^T\mathbf{x})\} = E\{g'(\mathbf{w}^T\mathbf{x})\}I$$

We use the $\mathbf{w}^T E\{\mathbf{x}g(\mathbf{w}^T\mathbf{x})\} - \mathbf{w}^T\beta\mathbf{w} = E\{\mathbf{w}^T\mathbf{x}g(\mathbf{w}^T\mathbf{x})\} - \beta = 0$

For the reason of use the normalization to get stability, we can just multiply the right side of the above iterations by $\beta - E\{g'(\mathbf{w}^T\mathbf{x})\}$ to further simplify the iteration.

$$\mathbf{w}^+ = E\{\mathbf{x}g(\mathbf{w}^T\mathbf{x})\} - E\{g'(\mathbf{w}^T\mathbf{x})\}\mathbf{w}$$

$$\mathbf{w}^* = \frac{\mathbf{w}^+}{\left\|\mathbf{w}^+\right\|}$$

Then the $\mathbf{w}^*$ can be used for next time iterations until it achieve converge. For multiple independent components we need use a Gram-Schmidt-like decorrelation method. We estimate the independent components one-by-one, and each iteration we subtract the currently get vector's projection to all previously get vector. It can be expressed use the below methods:

$$\mathbf{w}_{j+1} = \mathbf{w}_{j+1} - \sum_{i=1}^{j} \mathbf{w}_{j+1}\mathbf{w}_j\mathbf{w}_j$$

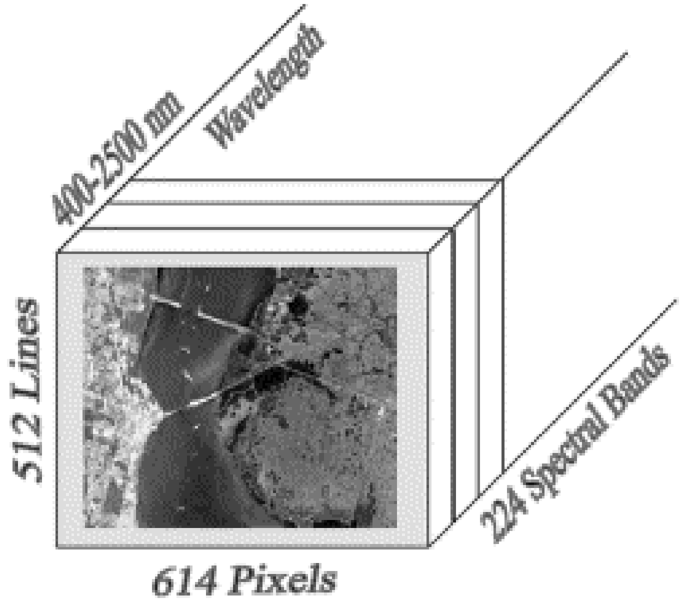
$$\mathbf{w}_{j+1} = \frac{\mathbf{w}_{j+1}}{\left\|\mathbf{w}_{j+1}\right\|}$$

After we get $\mathbf{W}$, we can use it to get the independent components $\mathbf{s}$.

3 Use Fast-ICA algorithm for Hyperspectral Image Classification.

3.1 Experiments

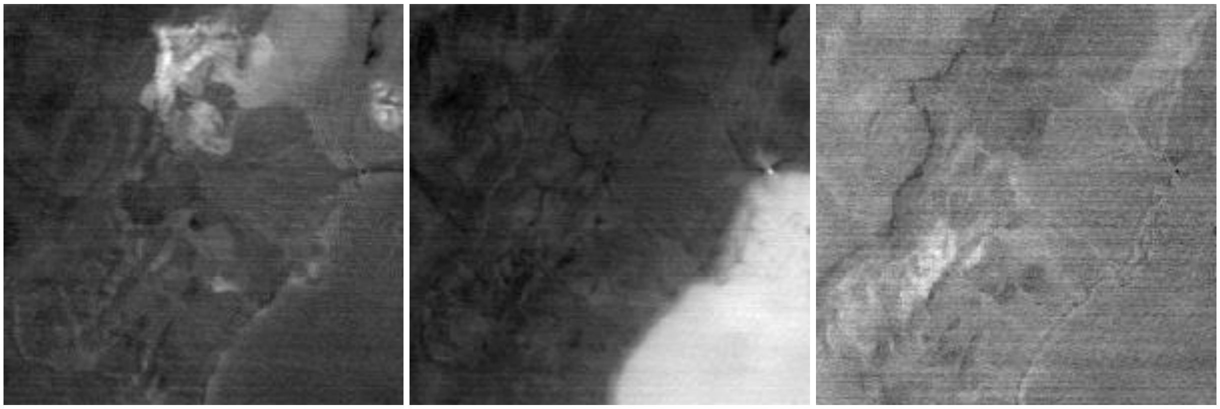
Hyperspectral image is a $l \times m \times n$ dimensional data cube, where the l is the number of bands in the hyperspectral image, and $m \times n$ is the number of pixels.

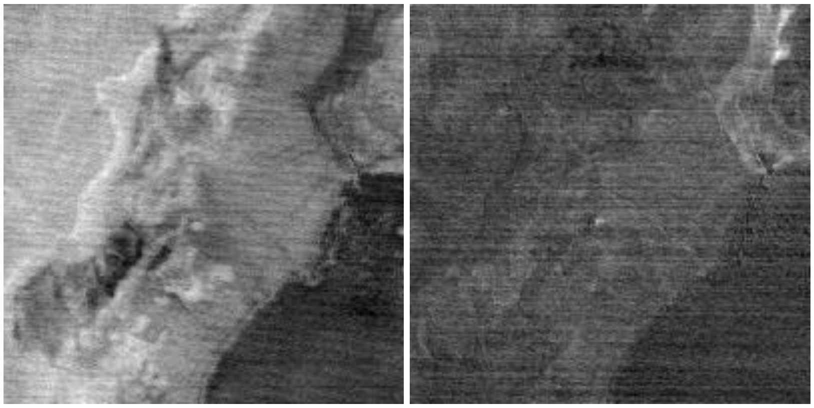


We can take each band as an observation of the mixed signal, and assume different classes in each pixel are statistically independent. Then we can reduce the 3D data cubic into a 2D matrix. Then we can use each row of the matrix as the observed signal. And the Fast-ICA algorithm will computed the independents component for each pixel, which we assume is the class in each pixel.

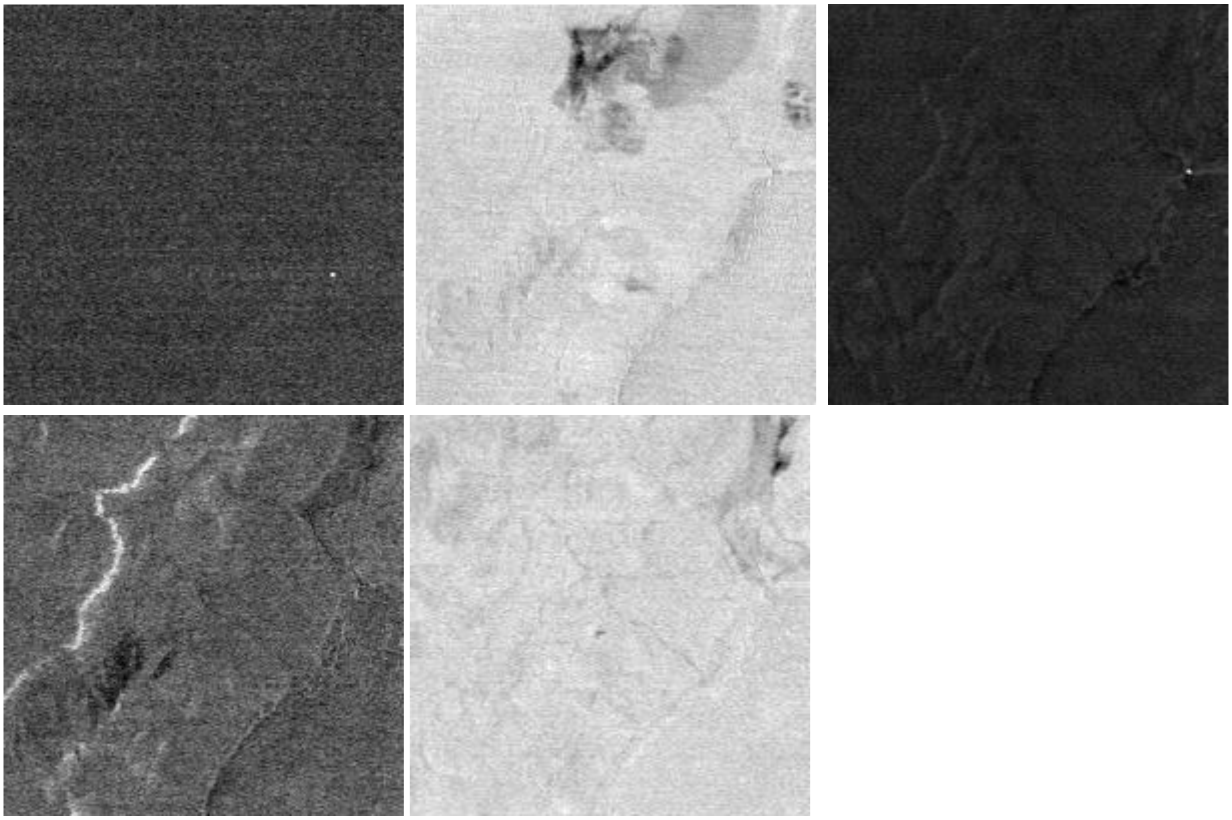
Below we will use the Fast-ICA algorithm to do unsupervised classification to the “Lunar Lake” hyperspectral image, it has 158 bands and size is 200x200 that is each bands has 40000 pixels. We will compare its results with the supervised classification method’s results (OSP) which we take it as a kind of ground truth. For the Fast-ICA algorithm, we use the matlab package we get from the internet (http://www.cis.hut.fi/projects/ica/fastica/).

The results of OSP method: The below classification results are **Cinder (C), Playa Lake (P), Rhyolite(R), Shade (S), Vegetation (V).**



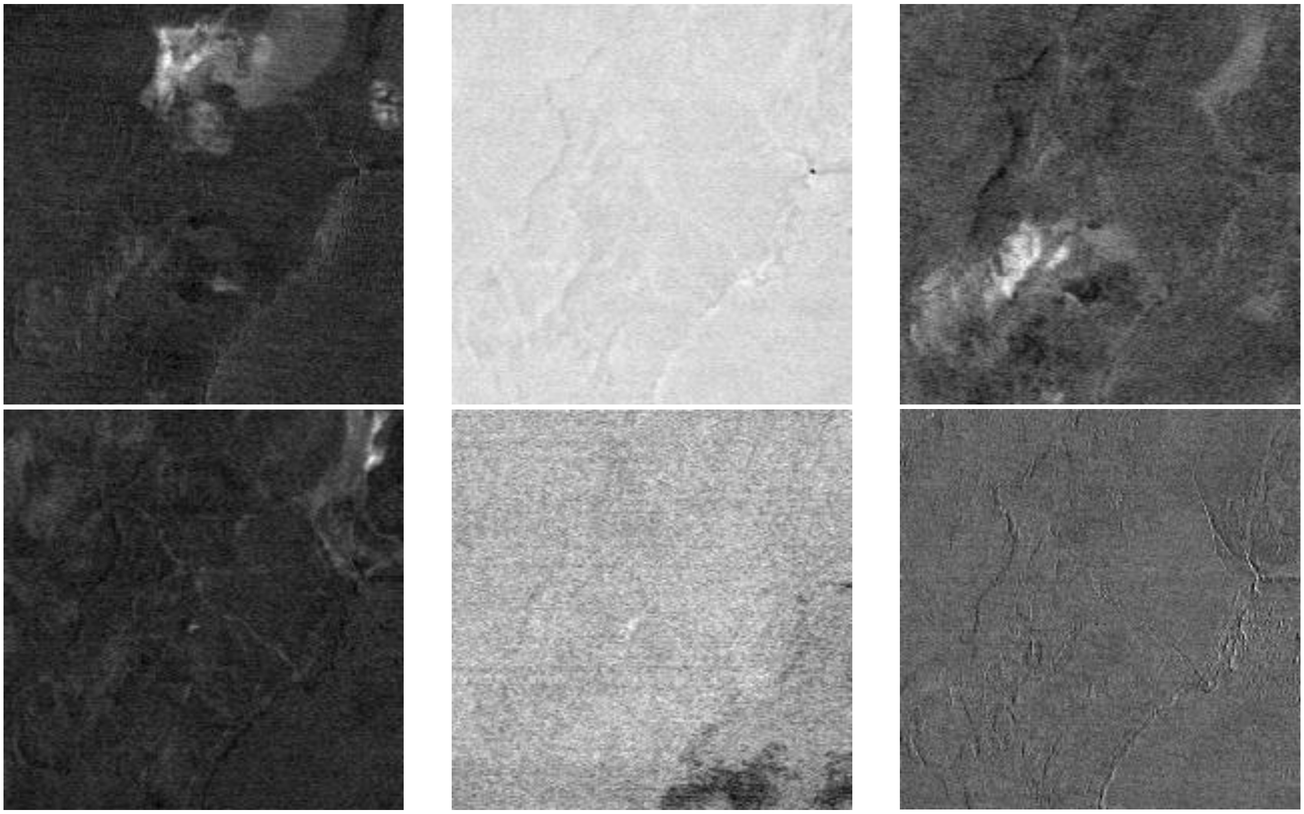


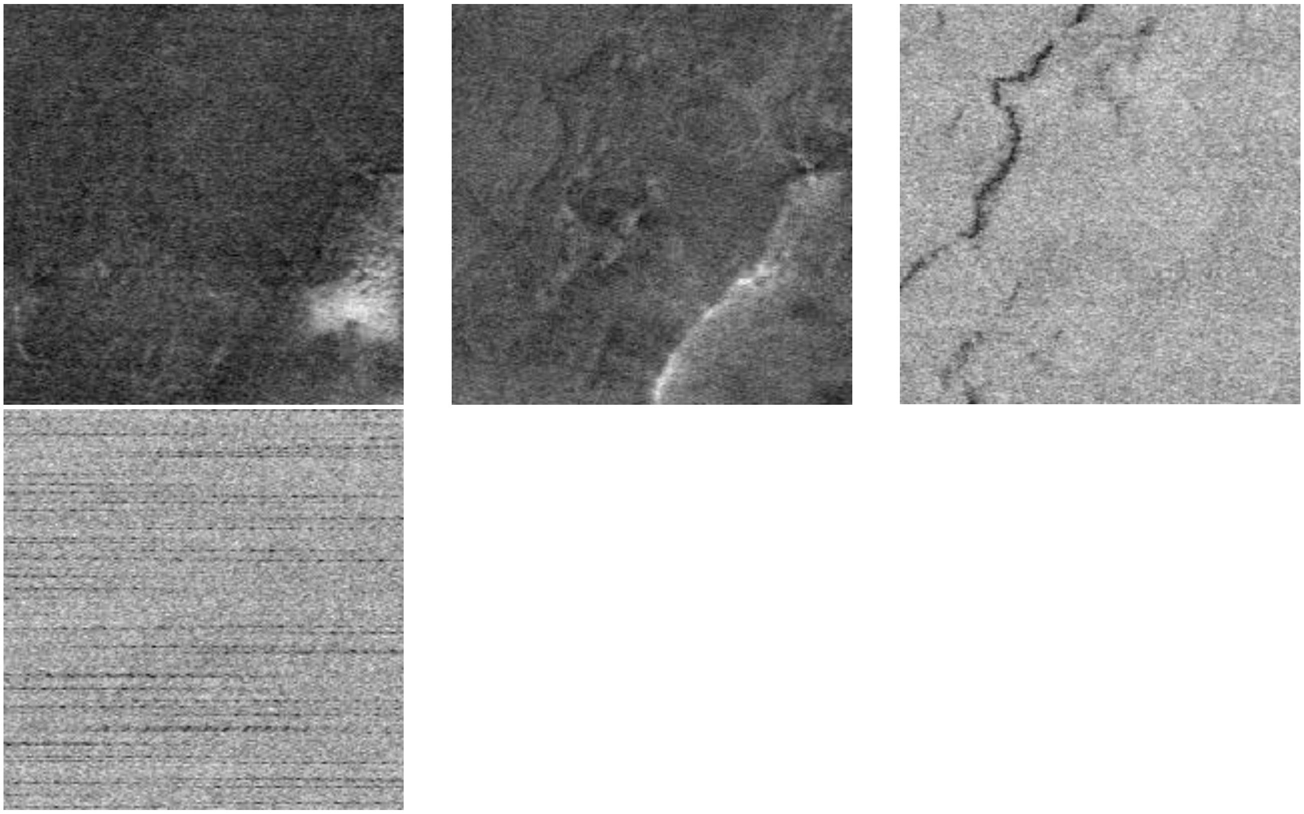
As OSP is a supervised classification method, it use the already known spectral signature to classification the hyperspectral image which means the number of interested class is known. However, in Fast-ICA algorithm, we don't know the number of classes, so we try to change the number of independents components to see the results. First we try the Fast-ICA algorithm with different number of independents components with the defaults function $g(x)=x^{3}$, which is the derivation $G(x)$ the non-quadratic function. We first start with independent components number equal to 5. Below are the classification results:



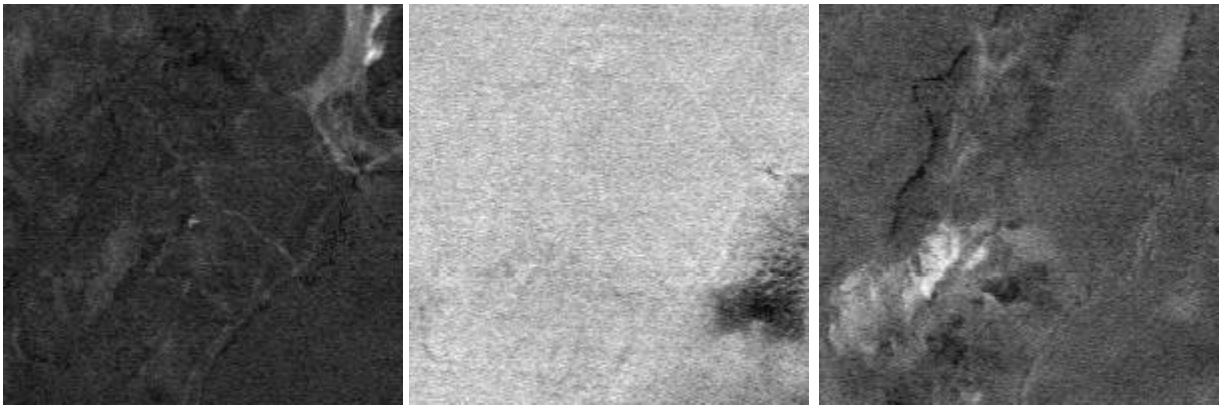
It can be clear seen that, the classification results is not good, but still we can see the second image show the class of cinder and the fifth image is the vegetable.

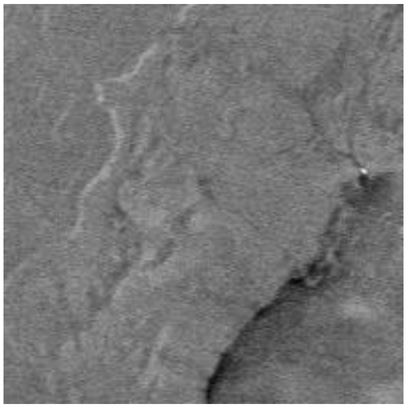
Then we increase the number of independents components number to 10 and keep the same non-quadratic function. Then we got below results. It can be seen from the results, the algorithm correctly classified the **Cinder (C) (Pic.1), Rhyolite(R) (Pic.3), Vegetation (V) (Pic.4).** Partly correct of **Playa Lake (P) (Pic.7), Shade (S),.**

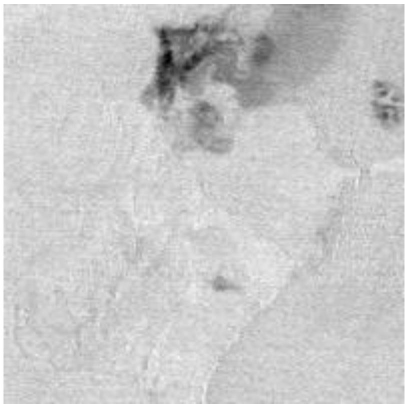




Then we change the Fast-ICA’s non-quadratic function, use the original paper suggests optimal function: $g(x) = \tanh(a_1 x)$, and use 5 independent components.







It can be seen that the above picture have better results: **Cinder (C) (Pic.5), Rhyolite(R) (Pic.3), Vegetation (V) (Pic.1).**Partly correct of **Playa Lake (P) (Pic.2).**

3.2 Conclusion

From the above experiments, we can see the Fast-ICA algorithm works on the unsupervised classification. Even the assumption that each class is independent on each pixel may not true. However we can see there are some restrictions in classification use the ICA method. First the results show that the classification results depends on the number of independents components we choose. If we only use the number of class we interested as a basis, it usually not given a good results. We can increase the number of independent components, but we need discard some results we are not interested in. Second the results also depends on the non-linear function we use, a better function can have a better results than other function. Third, we can't decide the order of class as the ICA algorithm can't decide which independent's components has high priority than others.

4 Reference:

[1] A. Hyvärinen and E. Oja. **Independent Component Analysis: Algorithms and Applications.** *Neural Networks*, 13(4-5):411-430, 2000.

[2] A. Hyvärinen. **Fast and Robust Fixed-Point Algorithms for Independent Component Analysis**. *IEEE Transactions on Neural Networks* 10(3):626-634, 1999.

[3] A. Hyvärinen. **New Approximations of Differential Entropy for Independent Component Analysis and Projection Pursuit.** In *Advances in Neural Information Processing Systems 10* (NIPS*97), pp. 273-279, MIT Press, 1998