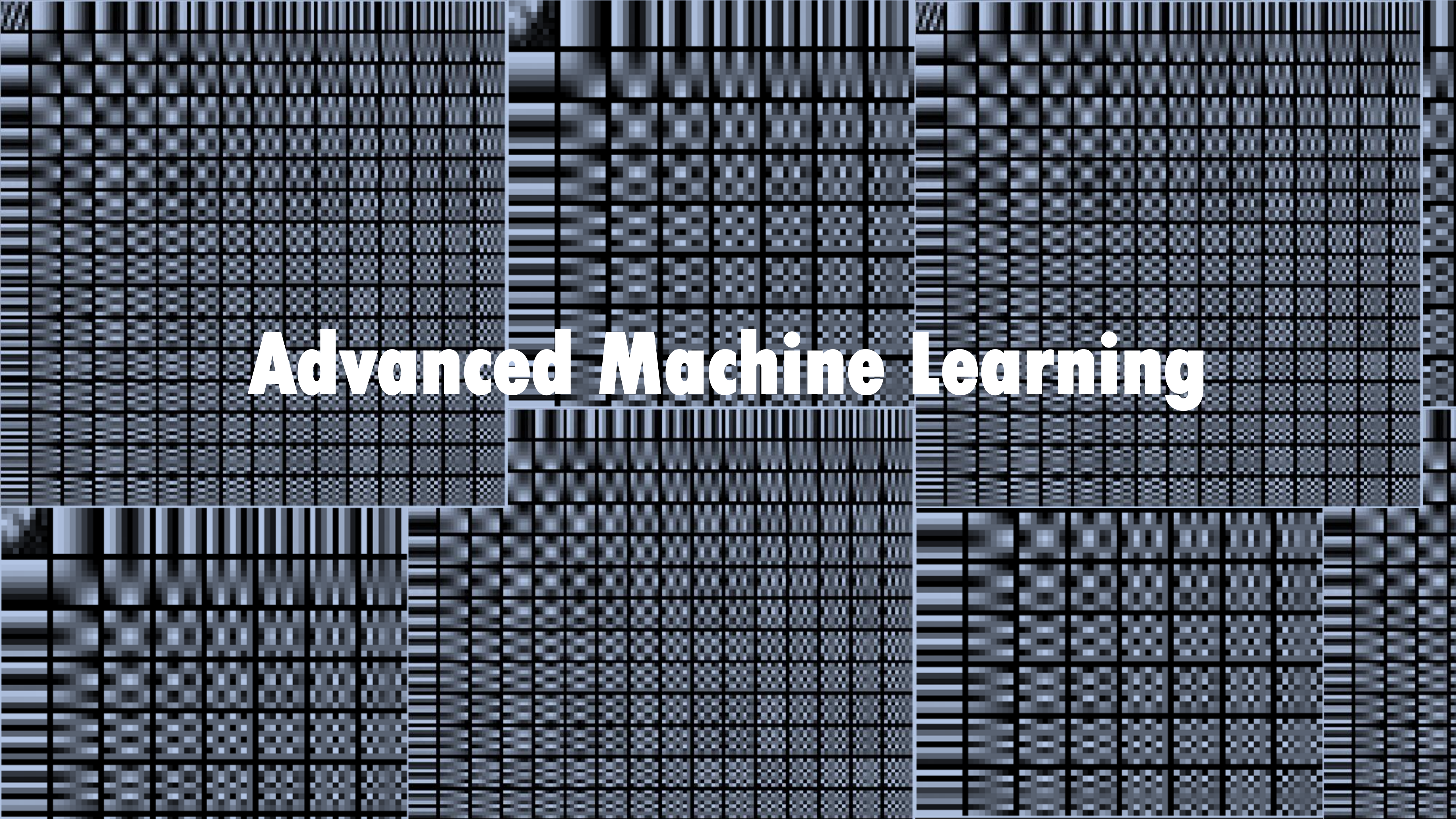
Advanced Machine Learning

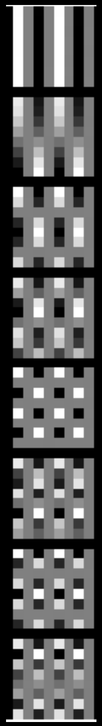
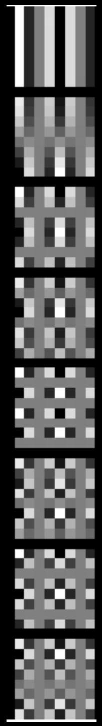


Noisy Image

De-noised

How do you go from noisy to de-noised?

Dictionary and Sparse Codes

 =  +  + 

$$\begin{Bmatrix} x_1 \\ 0 \\ 0 \\ 0 \\ x_5 \\ 0 \\ x_7 \\ 0 \end{Bmatrix}$$

Sparse Coding Techniques for Dictionary Learning in Context of Image De-noising

Sarah Panda (sp3206)

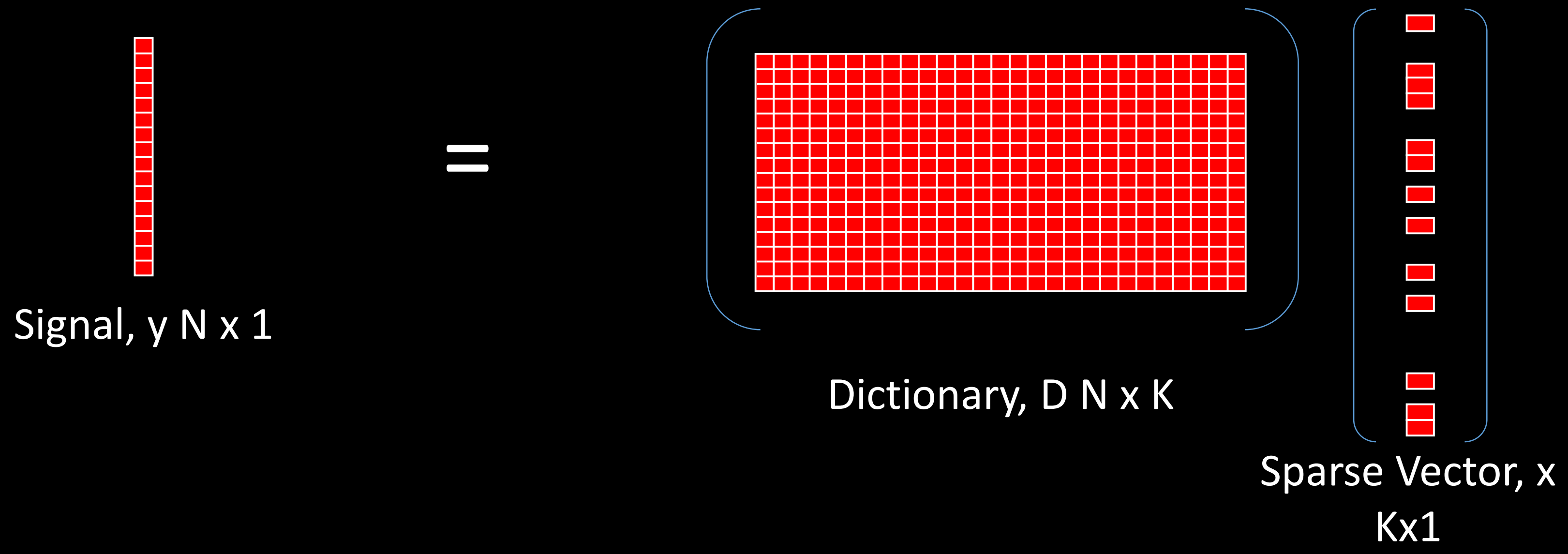
Dhaivat Shah (ds3267)

Gaurav Ahuja (ga2371)

Dictionary and Sparse Codes

- A dictionary, D is an N X k matrix that contains k prototype signals called ATOMS as columns, such that any given signal y can be represented as a linear combination of a small number of these atoms.
- For any signal y and dictionary D, there must exist a vector x in R^K, such that,

$$y = D*x$$



OK, where do we get the dictionary from?

- Use pre-calculated dictionaries built from a huge dataset of natural images over time.

OR

- Learn a dictionary from the given set of training signals : Has proven to dramatically improve signal reconstruction.

- So, here comes the problem of **Dictionary Learning!!**

Dictionary Learning

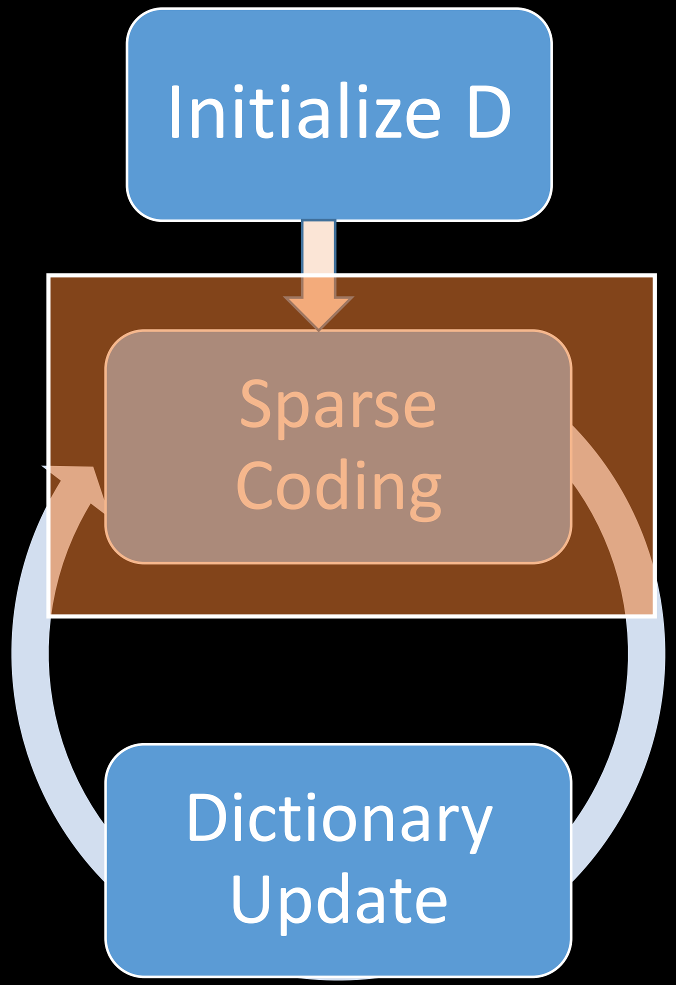
- Classical dictionary learning techniques try to optimize the following empirical cost function, for a training set of signals Y = [y1, y2,…., yn]

$$L(D) \quad = \frac{1}{n}\sum_{i=1}^{n} l(y, D)$$

- $l(y, D)$ is a loss function that assumes a low value if the dictionary D is good at representing signal y

K-SVD

- Sounds Good?
- But x and D are unkown…
- Dictionary update Step uses standard K-SVD
- We focus our attention on Sparse Coding



Sparsity

$$\min \|x\|_0$$

Subject to y = D*x

- This is the exact representation of the problem.
- But as it turns out, solving the above equation is NP-Hard.

Relaxation

$$\min \|x\|_0$$
$$\text{Subject to } \|y - Dx\|^2 < \varepsilon$$

- But still solving this is hard.
- So what do we do?
- Get Greedy and more Relaxation

Greedy Approach (Matching Pursuit)

$$\min \|x\|_0$$
$$\text{Subject to } \|y - Dx\|^2 < \varepsilon$$

- Sequentially selects best Dictionary atoms
- Example Solvers: MP, OMP

Basis Pursuit

$$\min_{x} ||y - Dx||^2 \; + \lambda ||x||_1$$

The original equation can be converted to a convex form by relaxing the l_0 norm with l_1 norm.

Sparse Coding Techniques Used

- Matching Pursuit (MP)
- Orthogonal Matching Pursuit (OMP)
- FISTA
- Augmented Langrangian Method (ALM)
- Feature Sign
- Truncated Newton Interior Point Method (L1LS)

Matching Pursuit

$$y = \sum_i x_i d_i$$

Input: Signal – y, dictionary D

Output: List of Coeff: $(x\ )$

Init: R_1 = y, $x\ = 0$,

Repeat:

  find $d_i \in$ D with max inner product $|\langle R_n, d_i \rangle|$

  $x_i = x_i + \langle R_n, d_i \rangle$

  $R_{n+1} = R_n - x_i d_i$

Until Stop condition: $\|R_n\| <$ threshold;

Orthogonal Matching Pursuit

- Extension to MP.
- Orthogonal Projection of signal onto already selected set of atoms

Input: y, D_0
Init: $\Gamma \leftarrow \phi$, r = y
while $\|r\|$ > threshold
    $i \leftarrow \mathrm{argmax}_{\{p=1,\ldots,k\}} \left| d_p^T r \right|$
    $\Gamma \leftarrow \Gamma \cup \{i\}$
    Solve for z: $(d_i' * d_i)\, z_i = (D' * r)[i]\ \forall i \in \Gamma$
    $x_i = x_i + \mathrm{z_i}\ \forall\, i \in \Gamma$
    $r = y\ - D(:, \Gamma)' * x(\Gamma)$

Feature Sign

$$\min_{x,D} \sum_{i=1}^{n} \left(\left\| y^i - \sum_{j=1}^{k} x_j^i \, D_j \right\|^2 + \lambda \sum_{j=1}^{k} |x_j^i| \right)$$

Input: Signal- Y = [$y_{1,}y_2$,…, y_n], dictionary D

Output: List of Coeff: (x_j)

Init: x := 0, θ=0, active set ={};

Repeat:

Activate a co-efficient x_i from non-zero co-efficient of x , if it locally improved the objective($\frac{\partial \|y-D*x\|^2}{\partial x_i}$)

Repeat:

Let D be the submatrix of D that include columns corresponding to active set

Given a and θ,

$x_{new} = (D^T D)^{-1} (D^T y - \lambda \frac{\theta}{2})$ ….(unconstrained optimization problem, solved analytically)

Perform a discrete line search from x to x_{new} and check for points where co-effient changes sign.

Update a with the point with the least objective value.

Until $\frac{\partial \|y-D*x\|^2}{\partial x_i} + \lambda sign(x_i)$=0, $\forall x_i = 0$, not satisfied

Until $\left|\frac{\partial \|y-D*x\|^2}{\partial x_i}\right| \leq \lambda, \forall x_i \neq 0$ not satisfied

Primal ALM

$$\mathcal{L}_{\xi}(x, \theta) = g(x) + \frac{\xi}{2}\|h(x)\|_2^2 + \theta^T h(x),$$

Algorithm 1 Primal Augmented Lagrangian Method (PALM) for CAB

Input: $\boldsymbol{b} \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$, $\boldsymbol{x}_1 = \boldsymbol{0}$, $\boldsymbol{e}_1 = \boldsymbol{b}$, $\boldsymbol{\theta}_1 = \boldsymbol{0}$.
while not converged $(k = 1, 2, \ldots)$ **do**
  $\boldsymbol{e}_{k+1} \leftarrow \text{shrink}(\boldsymbol{b} - A\boldsymbol{x}_k + \frac{1}{\xi}\boldsymbol{\theta}_k, \frac{1}{\xi})$;
  $t_1 \leftarrow 1$, $\boldsymbol{z}_1 \leftarrow \boldsymbol{x}_k$, $\boldsymbol{w}_1 \leftarrow \boldsymbol{x}_k$;
  while not converged $(l = 1, 2, \ldots)$ **do**
    $\boldsymbol{w}_{l+1} \leftarrow \text{shrink}(\boldsymbol{z}_l + \frac{1}{L}A^T(\boldsymbol{b} - A\boldsymbol{z}_l - \boldsymbol{e}_{k+1} + \frac{1}{\xi}\boldsymbol{\theta}_k), \frac{1}{\xi L})$;
    $t_{l+1} \leftarrow \frac{1}{2}(1 + \sqrt{1 + 4t_l^2})$;
    $\boldsymbol{z}_{l+1} \leftarrow \boldsymbol{w}_{l+1} + \frac{t_l - 1}{t_{l+1}}(\boldsymbol{w}_{l+1} - \boldsymbol{w}_l)$;
  end while
  $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{w}_l$, $\quad \boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k + \xi(\boldsymbol{b} - A\boldsymbol{x}_{k+1} - \boldsymbol{e}_{k+1})$;
end while
Output: $\boldsymbol{x}^* \leftarrow \boldsymbol{x}_k, \boldsymbol{e}^* \leftarrow \boldsymbol{e}_k$.

$$\begin{aligned} e_{k+1} &= \arg\min_e \mathcal{L}_{\xi}(x_k, e, \theta_k) \\ x_{k+1} &= \arg\min_x \mathcal{L}_{\xi}(x, e_{k+1}, \theta_k) \\ \theta_{k+1} &= \theta_k + \xi(b - Ax_{k+1} - e_{k+1}) \end{aligned},$$

Dual ALM

$$\min_{y,x} \; -b^T y - x^T(z - A^T y) + \frac{\beta}{2}\|z - A^T y\|_2^2$$
$$\text{subj. to} \quad z \in B_1^\infty.$$

Algorithm 2 Dual Augmented Lagrangian Method (DALM) for CAB

1: **Input:** $b \in \mathbb{R}^m$, $B = [A, \; I] \in \mathbb{R}^{m \times (n+m)}$, $w_1 = 0$, $y_1 = 0$.
2: **while** not converged ($k = 1, 2, \ldots$) **do**
3: $\quad z_{k+1} = \mathcal{P}_{B_1^\infty}(B^T y_k + w_k/\beta)$;
4: $\quad y_{k+1} = (BB^T)^{-1}(Az_{k+1} - (Bw_k - b)/\beta)$;
5: $\quad w_{k+1} = w_k - \beta(z_{k+1} - A^T y_{k+1})$;
6: **end while**
7: **Output:** $x^* \leftarrow w_k[1:n], e^* \leftarrow w_k[n+1:n+m], y^* \leftarrow y_k$.

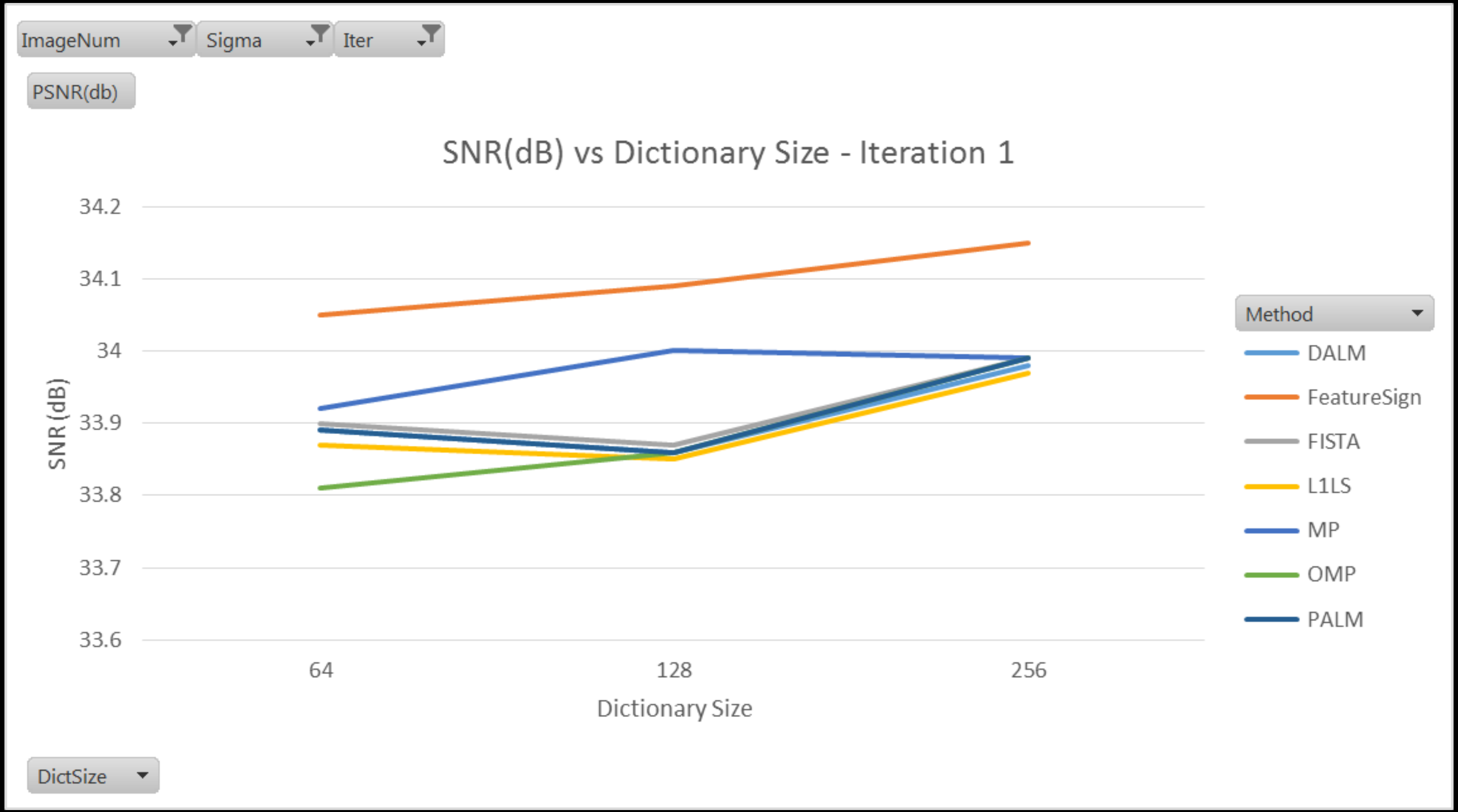
$$z_{k+1} = \mathcal{P}_{B_1^\infty}(A^T y_k + x_k/\beta),$$

$$\beta A A^T y = \beta A z_{k+1} - (A x_k - b).$$

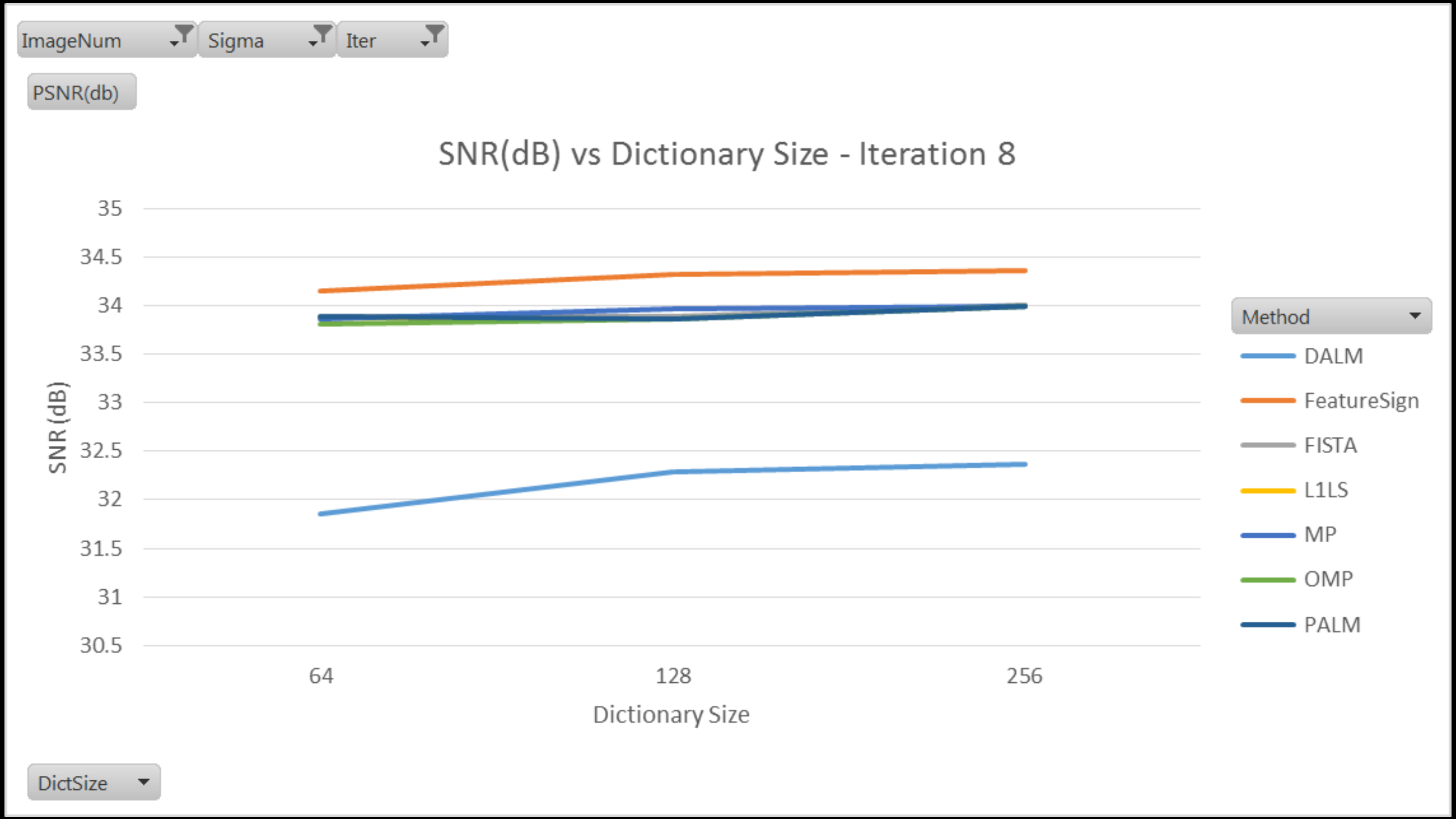
Experimental Setting

- K-SVD for Dictionary Learning
- Parameters experimented with
  - Image Noise Level
  - Dictionary Size
  - Sparse Coding Techniques
- Evaluation metric
  - Execution Time
  - SNR of de-noised image

Observations – Variations With Dictionary Size



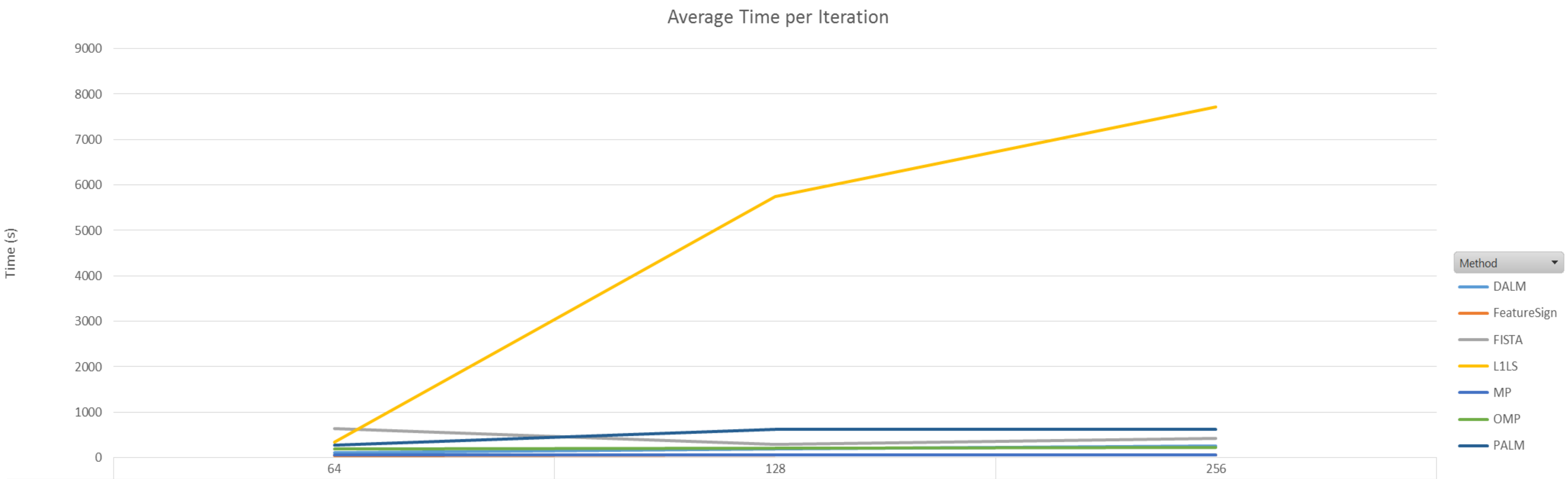
Observations – Variations With Dictionary Size



Observations – Execution Time

ImageNum | Sigma | Iter

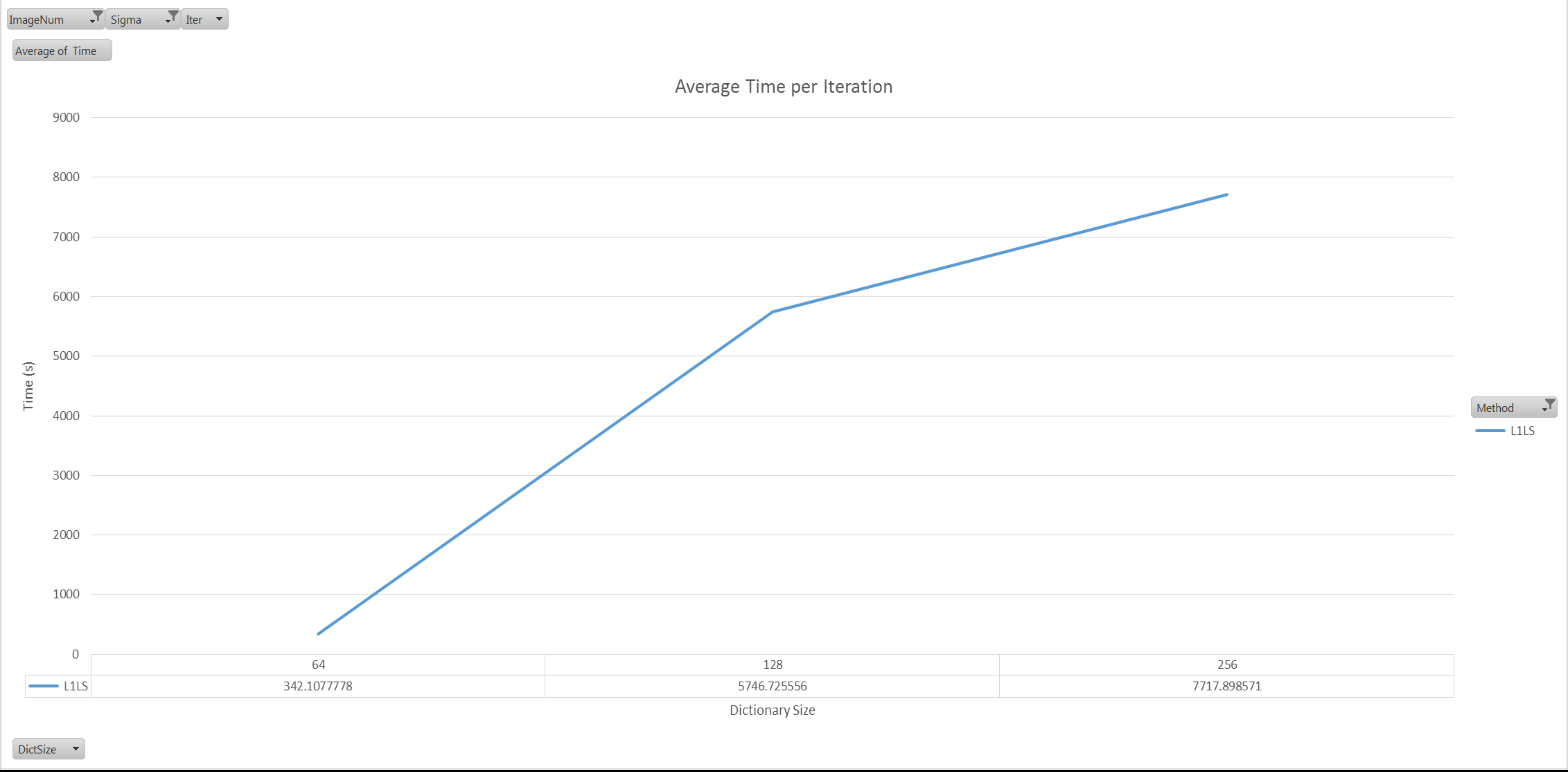
Average of Time

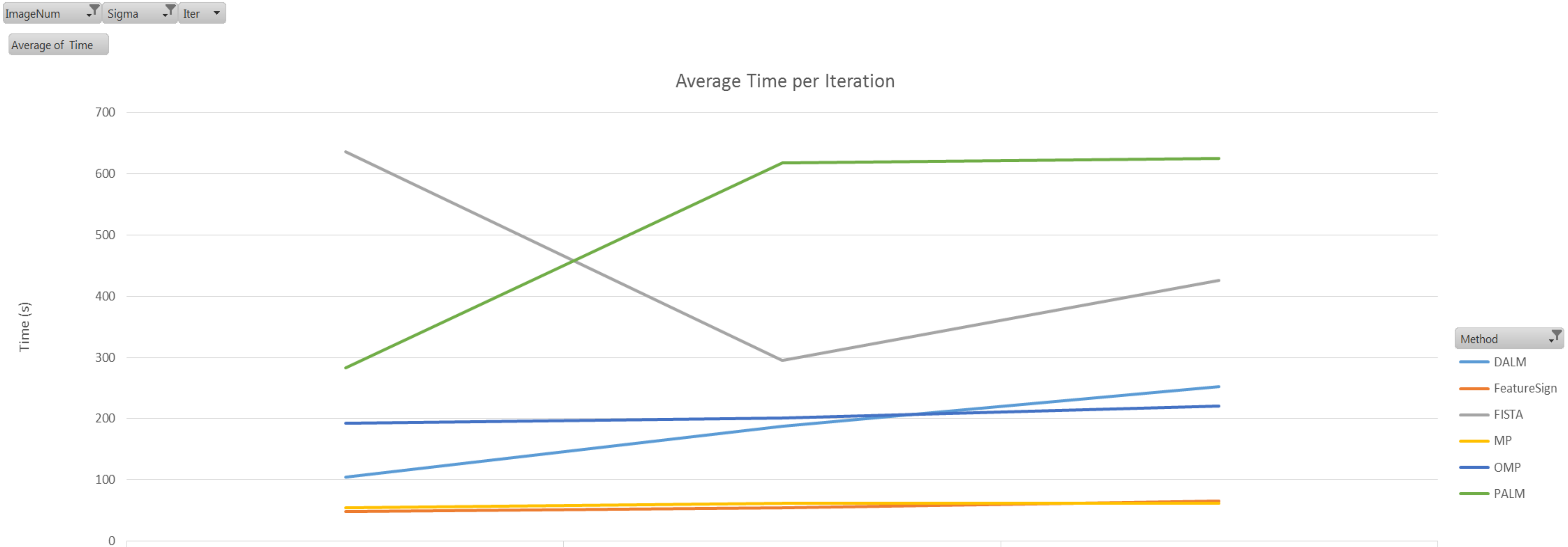


	64	128	256
DALM	104.9044444	187.4333333	251.7722222
FeatureSign	48.40333333	54.28888889	65.26
FISTA	635.6233333	295.1633333	426.4233333
L1LS	342.1077778	5746.725556	7717.898571
MP	54.76888889	61.43666667	61.54222222
OMP	193.1055556	201.6588889	221.12
PALM	282.6133333	618.2755556	624.8477778

Dictionary Size

DictSize


ImageNum
Sigma
Iter
Average of Time
Average Time per Iteration
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
Time (s)
64
128
256
L1LS
342.1077778
5746.725556
7717.898571
Dictionary Size
Method
L1LS
DictSize

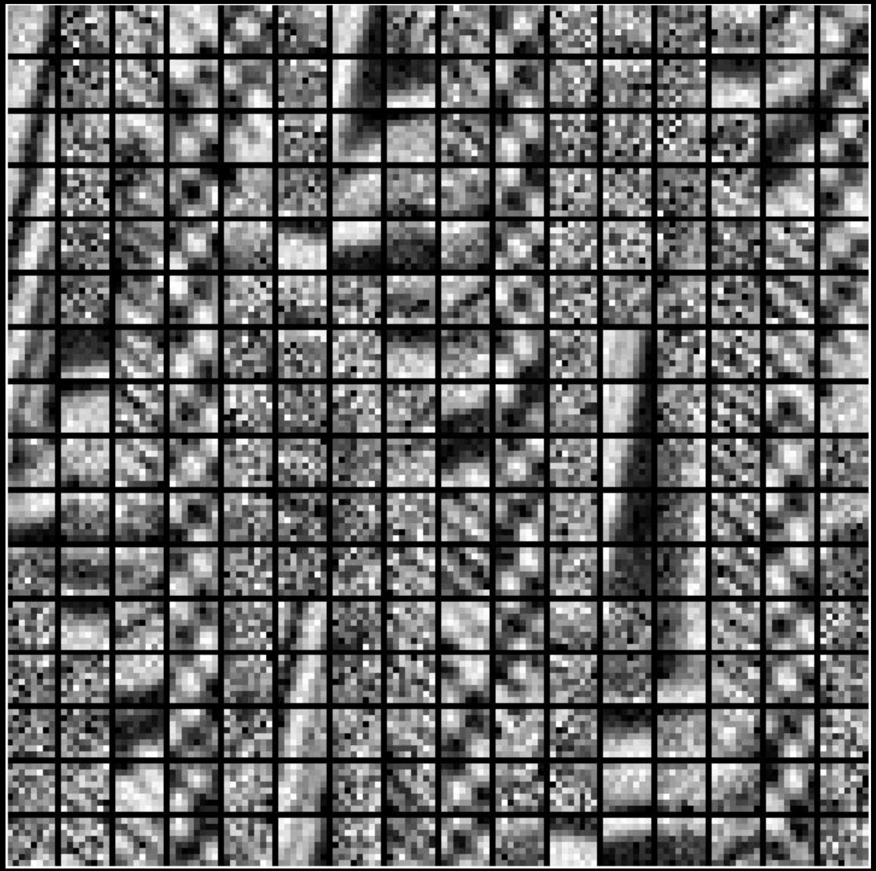


	64	128	256
DALM	104.9044444	187.4333333	251.7722222
FeatureSign	48.40333333	54.28888889	65.26
FISTA	635.6233333	295.1633333	426.4233333
MP	54.76888889	61.43666667	61.54222222
OMP	193.1055556	201.6588889	221.12
PALM	282.6133333	618.2755556	624.8477778

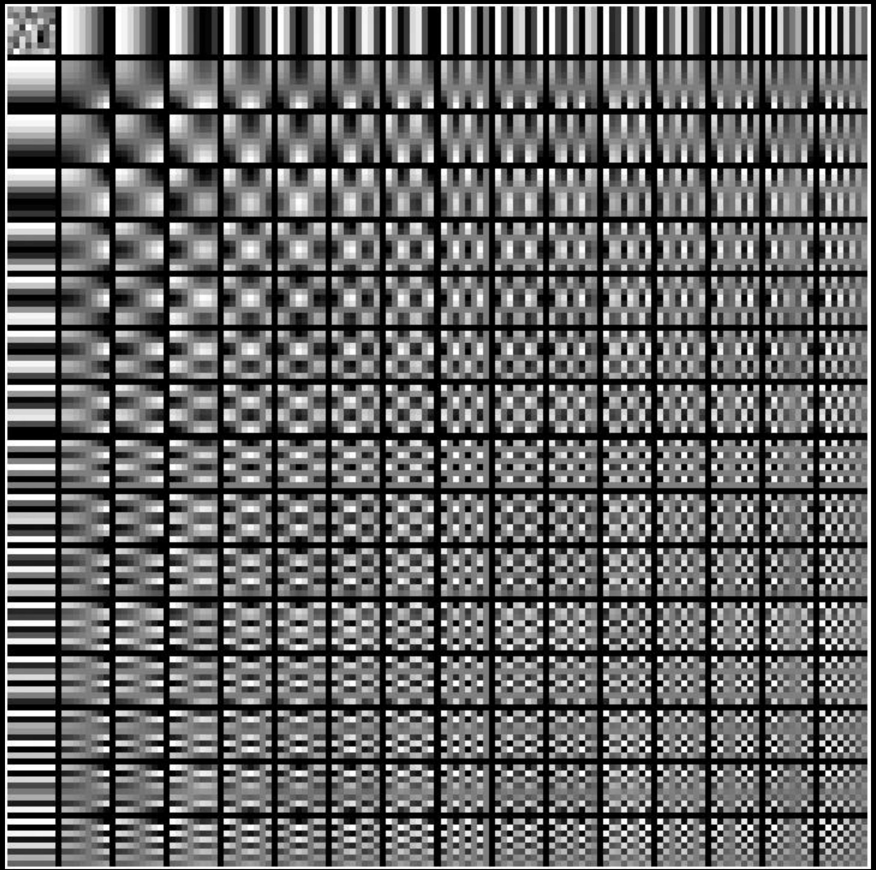
Dictionary Size

DictSize

Trained Dictionaries

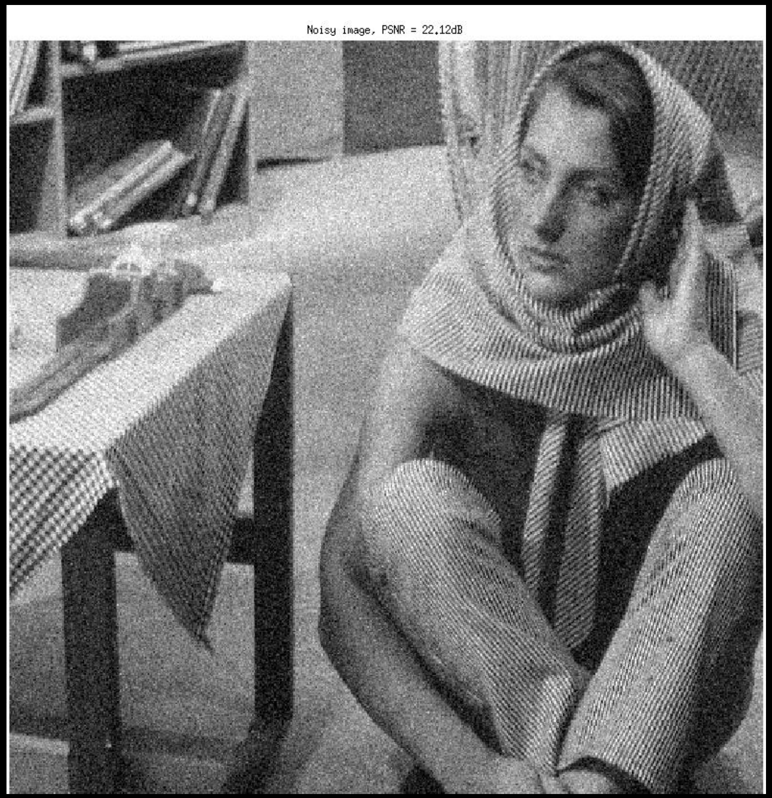


DALM – Sigma -10
De-noised Image PSNR = 32.36 (dB)



PALM– Sigma -10
De-noised Image PSNR = 33.99 (dB)

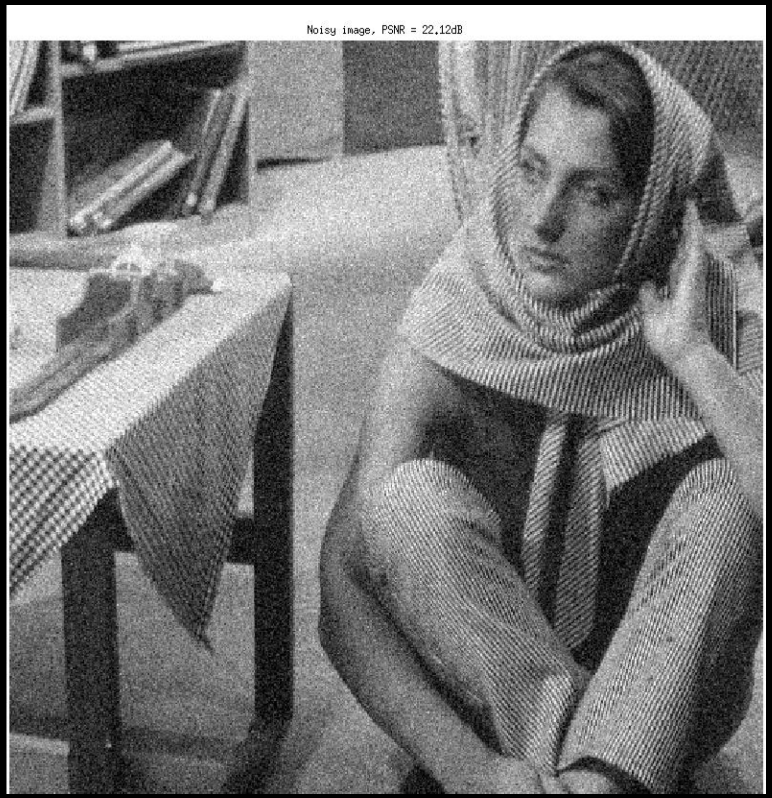
De-noised Images –DALM, Sigma=20



PSNR = 22.12 dB

PSNR = 27.94 dB

De-noised Images –Feature Sign, Sigma=20



PSNR = 22.12 dB

PSNR = 30.50 dB

Conclusion

- Performs better as dictionary size increases
- Feature-sign gives best results and converges quicker
- L1LS takes the maximum time per iteration
- DALM learns a poor dictionary

For more results

- https://github.com/dkdfirefly/aml

References

- Yang, Allen Y., et al. "Fast l1-minimization algorithms for robust face recognition." *arXiv preprint arXiv:1007.3753* (2010).
- Lee, Honglak, et al. "Efficient sparse coding algorithms." *Advances in neural information processing systems* 19 (2007): 801.
- Mallat, Stéphane G., and Zhifeng Zhang. "Matching pursuits with time-frequency dictionaries." *Signal Processing, IEEE Transactions on* 41.12 (1993): 3397-3415.
- Pati, Yagyensh Chandra, Ramin Rezaiifar, and P. S. Krishnaprasad. "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition." *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*. IEEE, 1993.
- Aharon, Michal, Michael Elad, and Alfred Bruckstein. "-svd: An algorithm for designing overcomplete dictionaries for sparse representation." *Signal Processing, IEEE Transactions on* 54.11 (2006): 4311-4322.

Thank You