

Assignment 1—Group:

Chen Chen (123123123), Dania(123123123), Joyce(123123123)

6th September 2018

Abstract

Abstract text goes here, justified and in italics. The abstract would normally be one paragraph long.

Contents

1	Introduction	1
2	Methods	2
3	Experiments	2
4	Conclusion	2
A	Appendix	2

1 Introduction

This template should be used as a starting point for your report.

2 Methods

12312323 Lee and Seung [2001] Guan et al. [2012]

3 Experiments

123

4 Conclusion

Your conclusion goes at the end, followed by References, which must follow the Vancouver Style (see: www.icmje.org/index.html). References begin below with a header that is centered. Only the first word of an article title is capitalized in the References.

A Appendix

Algorithm 1: The Levenberg–Marquardt algorithm iteratively finds optimal macroscale Robin boundary conditions.

```
1 %A=imread('C:\Users\chenc\OneDrive - UNSW\machine learning\  
assignment 1\data\CroppedYaleB\yaleB01\yaleB01_P00A+000E+00.  
pgm');  
2 red=3;  
3 k=40;  
4  
5 imagefiles = dir('data/ORL/*/*.pgm');  
6 imagefiles2=struct2cell(imagefiles);  
7 imagefiles=imagefiles((~endsWith(imagefiles2(1,:), 'Ambient.pgm'))  
    ');  
8 imagefiles2=struct2cell(imagefiles);  
9 A=imread(strcat(imagefiles(1).folder, '\', imagefiles(1).name));  
10 if size(A,1)==112  
11     A=A(1:111,1:90);
```

```

12 end
13 A_list=zeros(size(A,1)/red,size(A,2)/red,red);
14 nfiles = length(imagefiles); % Number of files found
15 matrix_image=zeros(prod(size(A))/red^2,nfiles);
16 temp=struct2cell(imagefiles);
17 names=temp(1,:);
18 for ii=1:nfiles
19     currentfilename = imagefiles(ii).name;
20     currentfilename=strcat(imagefiles(ii).folder,'\',
        currentfilename);
21     currentimage = imread(currentfilename);
22     if abs(size(A,1)-112)<=1
23         currentimage=currentimage(1:111,1:90);
24     end
25
26     for i=1:red
27         A_list(:,:,i)=currentimage(i:red:end,i:red:end);
28     end
29     A2=uint8(mean(A_list,3));
30     matrix_image(:,ii) = A2(:);
31 end
32
33 [w h]=NeNMF(matrix_image,k);
34 idx = kmeans(h',k)
35 Y_pred=zeros(size(matrix_image,2),1)
36 namess=str2mat(string(imagefiles2(2,:))')
37 namess=str2num(namess(:,end-1:end))
38 for ii=unique(idx)'
39     ind= (idx==ii);
40     Y_pred(ind)=mode(namess(ind,:));
41 end
42 nmi(Y_pred,namess)

```

References

Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 556–562. MIT Press, 2001. URL <http://papers.nips.cc/paper/1861-algorithms-for-non-negative-matrix-factorization.pdf>.

- N. Guan, D. Tao, Z. Luo, and B. Yuan. Nnmf: An optimal gradient method for nonnegative matrix factorization. *IEEE Transactions on Signal Processing*, 60(6):2882–2898, June 2012. ISSN 1053-587X. doi:[10.1109/TSP.2012.2190406](https://doi.org/10.1109/TSP.2012.2190406).