WEEK 1 & 2
COUNTING IN DENSE CROWDS USING DEEP LEARNING

Student: Karunya Tota

Mentor: Haroon Idrees

Group Member: Logan Lebanoff

WEEK 1

- Matlab Review
- Computer Vision Lectures
- Guest Speakers
- Topics: SIFT, Adaboost, Bag of Features

WEEK 2: SELECTION OF PROJECT

Counting in Dense Crowds using Deep Learning





EXAMPLES

- Concerts
- Political Speeches
- Rallies
- Stadiums

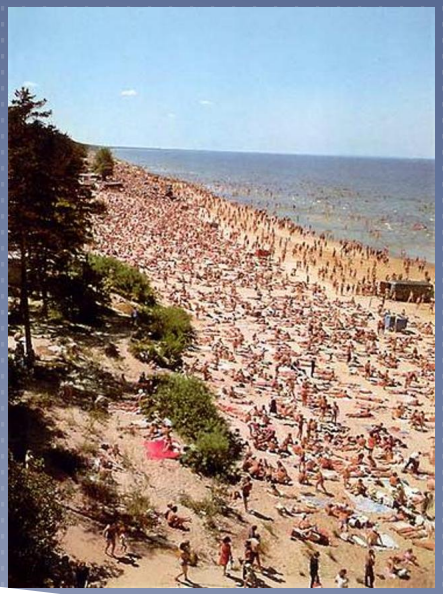
REAL WORLD APPLICATIONS

- Crowd Management
- Safety
- Surveillance
- Design Analysis for Buildings

DIFFICULTIES

- Occlusion
- Clutter
- Low resolution
- Foreshortening
- Irregular Arrangement







RESEARCH GOALS

- Perform deep learning via Convolutional Neural Networks (CNNs) to obtain counts of people in images
- Train the network and replace softmax loss function (used for classification task) with Mean Squared Error(MSE)
- Extract dense SIFT descriptors from the images and encode these descriptors using fisher vectors and then train a support vector machine.

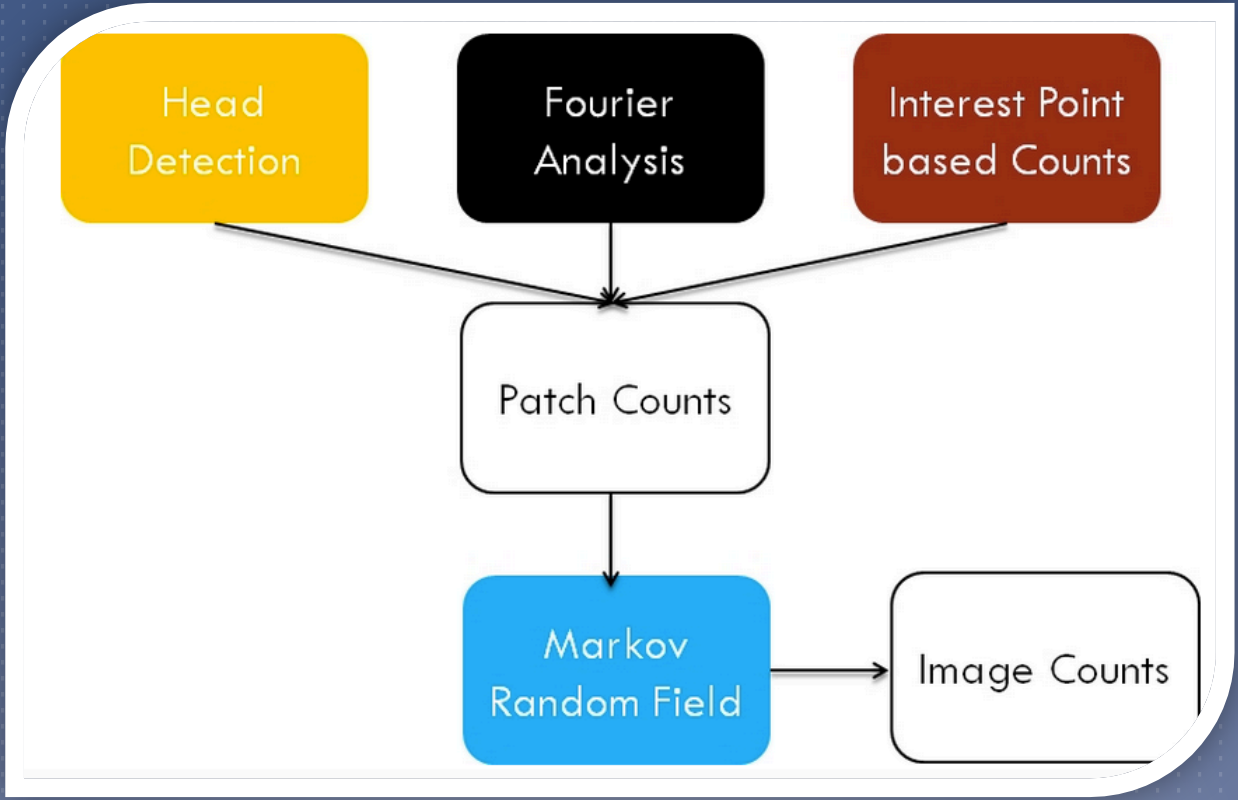
RESEARCH GOALS (CONT.)

- Also, extract features from the network and train a support vector machine afterward.
- Possibly explore other options to improve the performance of the counting algorithm (i.e. horizontal line estimation, using size of humans to infer density)

RESOURCES TO READ

- [1] Haroon Idrees, Imran Saleemi, Cody Seibert, Mubarak Shah, Multi-Source Multi-Scale Counting in Extremely Dense Crowd Images, *Computer Vision and Pattern Recognition (CVPR), Portland, Oregon, 2013*.
- [2] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." In Advances in neural information processing systems, pp. 1097-1105. 2012.

MULTI-SOURCE MULTI-SCALE COUNTING IN EXTREMELY DENSE CROWD IMAGES



Code

- Downloaded MatConvNet – Matlab toolbox implementing CNNs and worked with pre-trained models

```
% setup MtConvNet in MATLAB
run matlab/vl_setupnn

% download a pre-trained CNN from the web
urlwrite('http://www.vlfeat.org/sandbox-matconvne
  'imagenet-vgg-f.mat') ;
net = load('imagenet-vgg-f.mat') ;

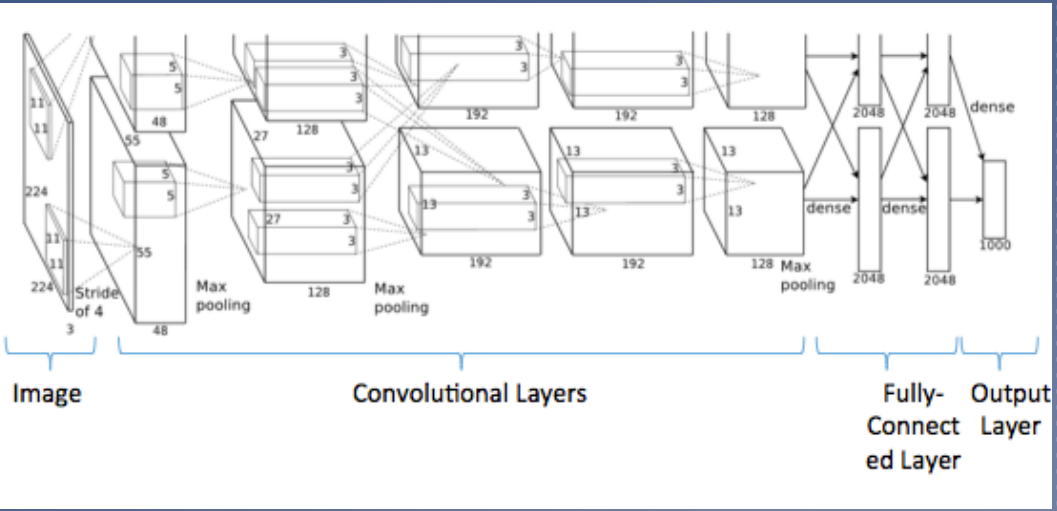
% obtain and preprocess an image
im = imread('peppers.png') ;
im_ = single(im) ; % note: 255 range
im_ = imresize(im_, net.normalization.imageSize(1
im_ = im_ - net.normalization.averageImage ;

% run the CNN
res = vl_simplenn(net, im_) ;

% show the classification result
scores = squeeze(gather(res(end).x)) ;
[bestScore, best] = max(scores) ;
figure(1) ; clf ; imagesc(im) ;
title(sprintf('%s (%d), score %.3f',...
net.classes.description{best}, best, bestScore))
```

CNN

VGG Convolutional Neural Networks Practical



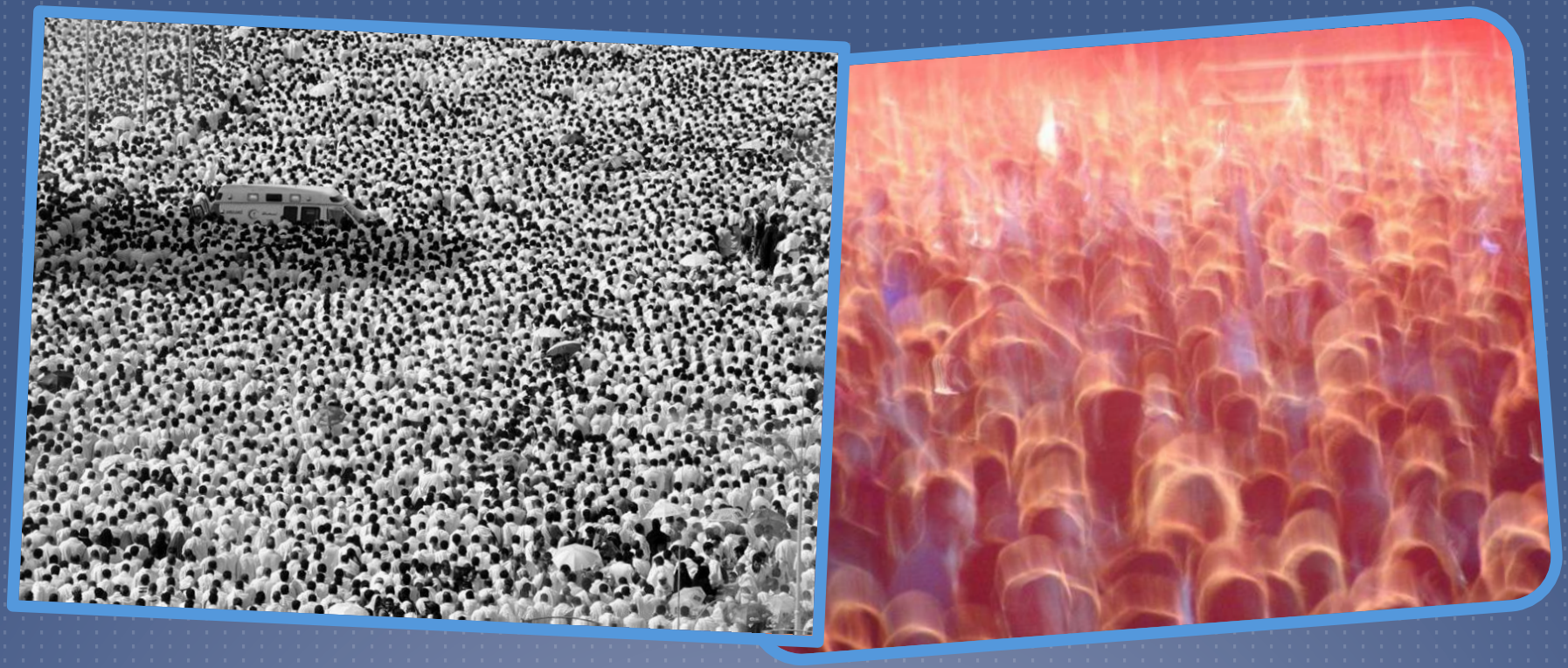

```
% Read an example image
x = imread('peppers.png') ;

% Convert to single format
x = im2single(x) ;

% Visualize the input x
figure(1) ; clf ; imagesc(x)

% Create a bank of linear filters
w = randn(5,5,3,10,'single') ;

% Apply the convolution operator
y = vl_nnconv(x, w, []) ;
```



DATASET

- 50 crowd images with counts ranging from 94 to 4543
- Will have to annotate the images in the future

NEXT STEPS

- Finish reviewing relevant papers
- Test code from the 'Multi Source Multi Counting in Dense Crowds' paper
- Learn more information about encoding via Fisher Vectors