A Few Words On Interesting Concepts in Applied Machine Learning

RightHand Robotics
Neil Joshi – 07/15/16

Overview

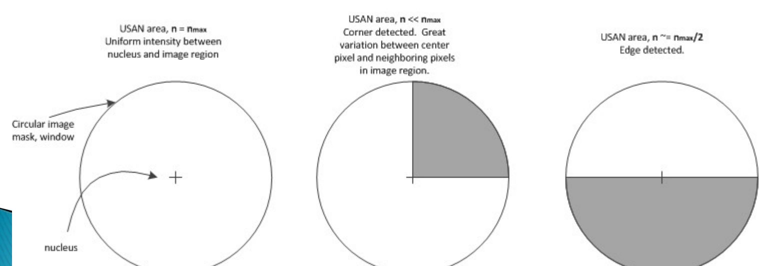
- 2 concepts in applied ML
  - **Feature identification and extraction**
    - Computer Vision application – effective corner detection using vector frame/image processing
  - **Automatic feature extraction with Learned Representations**
    - Unsupervised learning clustering application – Deep Learning

Effective Corner Detection – vector frame / image processing

- Features are regularities or attributes in data useful for describing or explaining the data
- Corners – corner detection, corners are useful features for visual data
- Effective corner detectors – SUSAN
- Works well with grayscale frames/images
- What of color visual data?
- Color (RGB) pixel encoded 24bits in contrast to grayscale 8bits

Effective corner detection : color visual data

- Use extra information encoded in color images to enhance corner detection
- Take SUSAN example:
  - evaluates intensities, I, between each pixel in image to neighboring pixels within a specified radius (windowing); formulates USAN area



Corner detection – USAN

- Define USAN area as, *n*, r_o as pixel center (nucleus), r_i pixel value of pixel *i*, then

$$c\left(r_i, r_0\right) = e^{-\left(\frac{I(r_i) - I(r_0)}{t}\right)^6}$$

$$n\left(r_0\right) = \sum_i c\left(r_i, r_0\right)$$

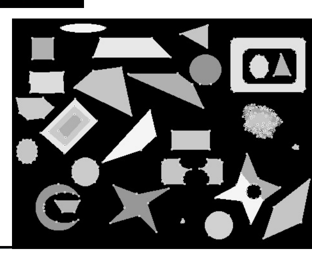
- Now, for color, use information in all channels, vector image processing (eg. Color image processing using ordered statistics)
- Does it work?

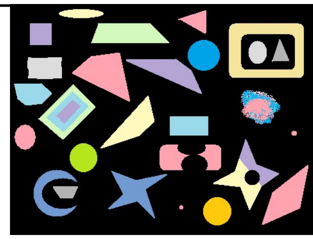
Corner detection – using vector image processing

- Test images on shapes, standard USAN corner detector (a), vector corner detector (b)

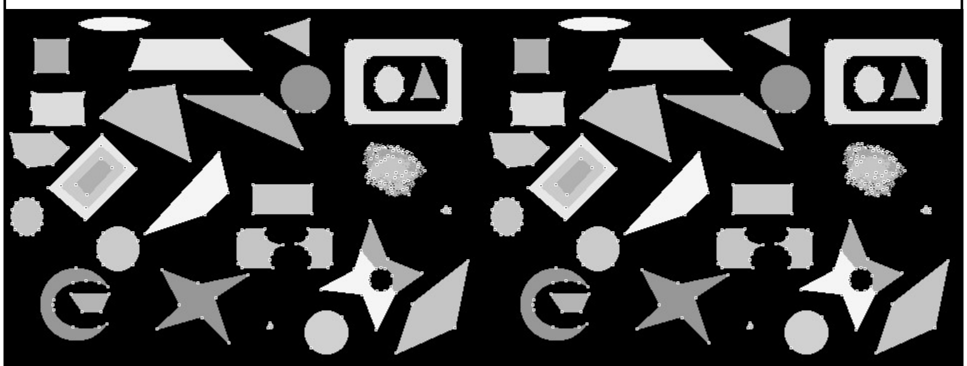
(a)

(b)





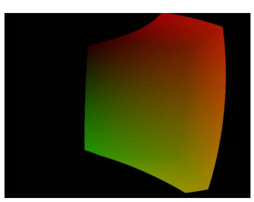
(a) (b)



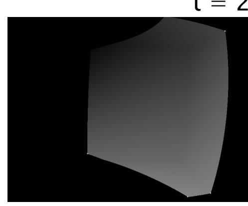
Corner detection – using vector image processing

- Test images on shapes, standard USAN corner detector (a), vector corner detector (b)

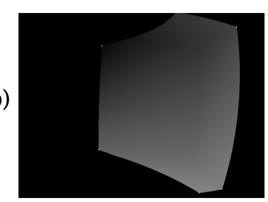
t = 20



(a)

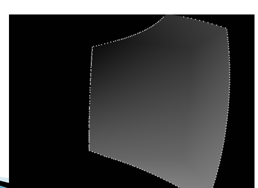


(b)

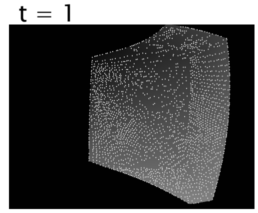


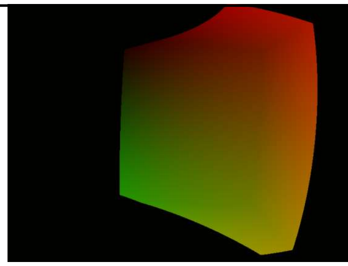
t = 1

(a)

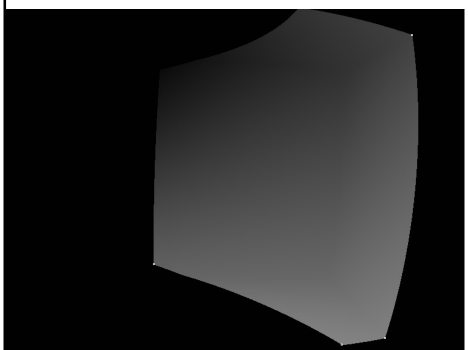


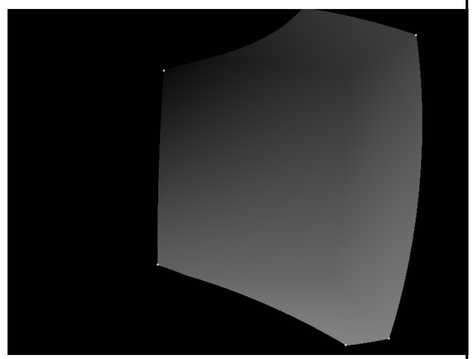
(b)

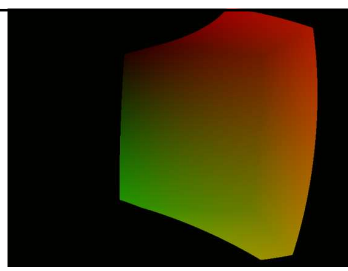




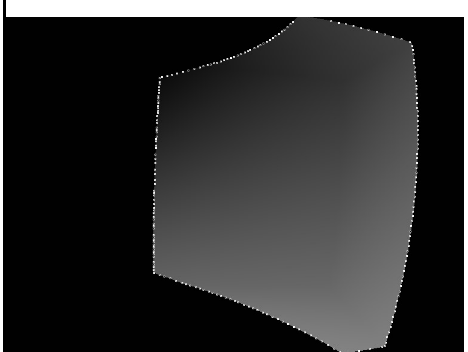
(a) t = 20 (b)

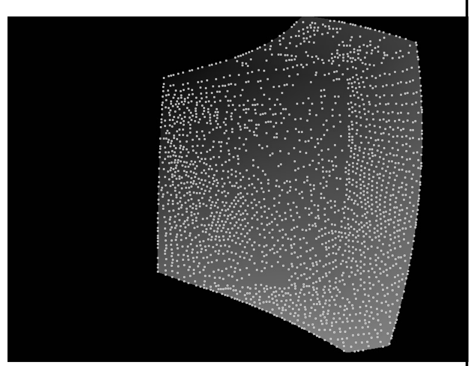






(a) t = 1 (b)





Corner detection – using vector image processing

- Test images on shapes, standard USAN corner detector (a), vector corner detector (b)

(a)

(b)

Automatic feature extraction with Learned Representations

- Premise to learn good features, regularities in data that explain the data
- Previous example, corner detector, is a tailored feature
- Transform the input space so regularities are evident?
  - Spectral methods, PCA, spectral graph cluster

$$\Sigma = U \lambda U^T \; , \quad X^{'} = XU \; ,$$

$$min \frac{x^T L x}{x^T x}$$

Learned Representations

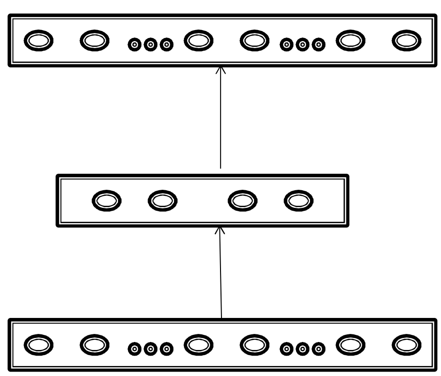
- Infer representation from data – minimize KL divergence between true input distribution and encoded (transformed) distribution

$$KL = \sum p\left(x\right)\, log\left(\frac{p(x)}{q(x)}\right)$$

$$H\left(p,q\right) = -\sum p\left(x\right)\, log\, q(x)$$

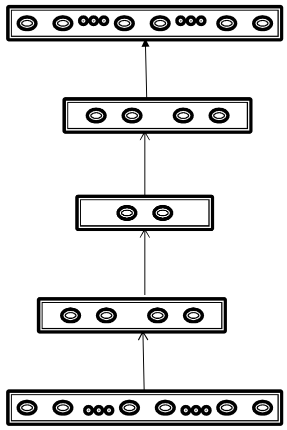
Learned Representations – Autoencoder

- Realized with the Autoencoder topology



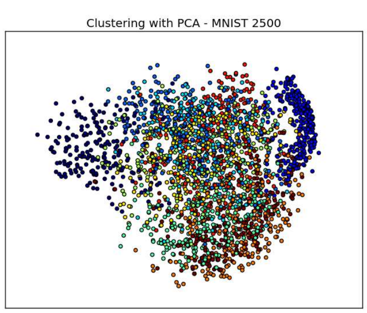
Learned Representation – Deep Autoencoder

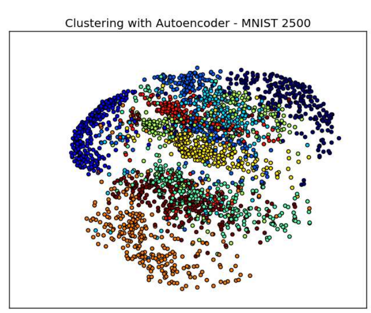
- More meaningful features extracted with stacked layers ; aids in training too.



Deep Autoencoder – Clustering application example

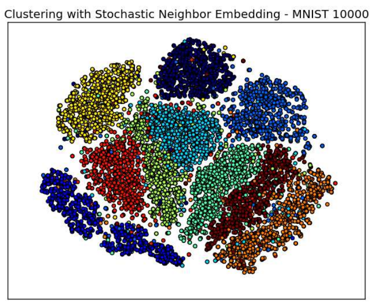
- MNIST – 2500 examples; digits 0-9
- Comparison between data encoded by PCA and Autoencoder with 784-500-2000-2-2000-500-784 topology:

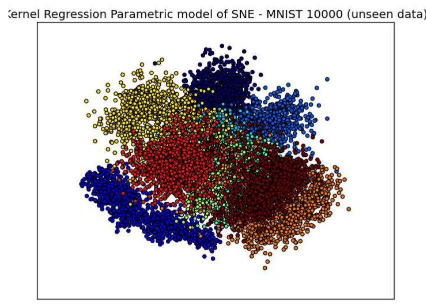




Learned Representations – MNIST Clustering Data Visualization

- Stochastic Neighbor Embedding is very effective for data visualization of commonalities within data
- This nonparametric method does not lend to unseen data – Soln: use kernel non linear reg.





Conclusions

- Use of vector computer vision image processing enhances corner detection. (example implemented with OpenCV)
- Learned representations can be formed from Deep Architectures. They are very effective for inferring meaningful features, extracting features from data. Non parametric data visualization clustering SNE method can be useful for unseen data using parametric model inferred using kernel non linear regression

Conclusions

Thank You!