

DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture

(DeepInsight法による非画像データへのCNN適用)

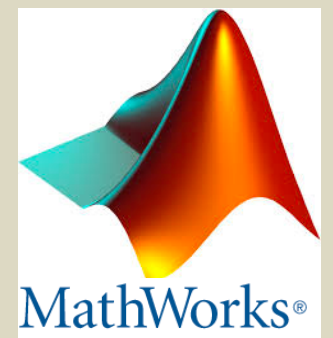


Alok Sharma

RIKEN Center for Integrative Medical Sciences, Japan

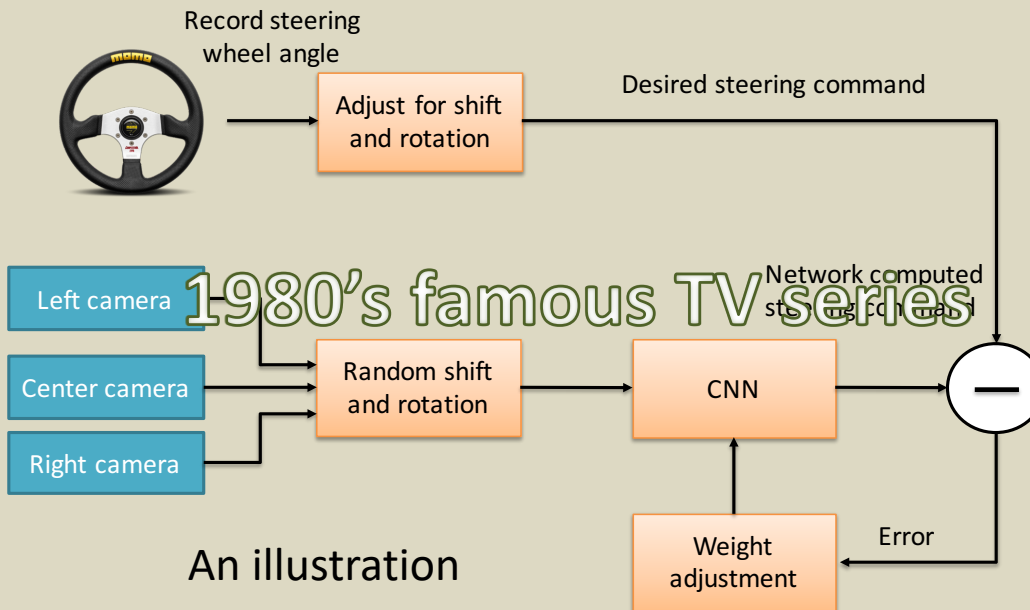
Griffith University, Australia

University of the South Pacific, Fiji



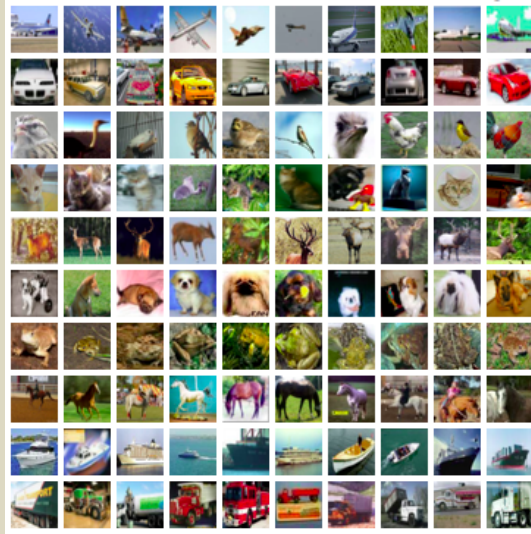
Convolutional Neural Network (CNN) Applications

- Driverless cars are the future of automobile industry.
- It uses AI models to guide vehicles.
- Convolution Neural Network (CNN) is a popular choice due to high performance



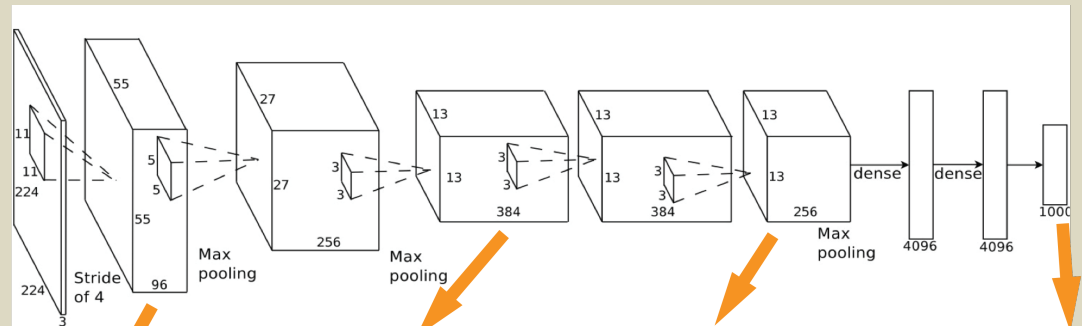
CNN Model

airplane
automobile
bird
cat
deer
dog
frog
horse
ship

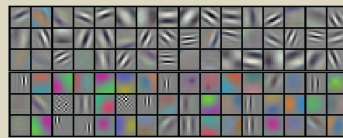
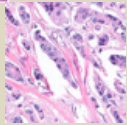
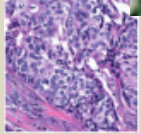
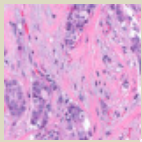


dog

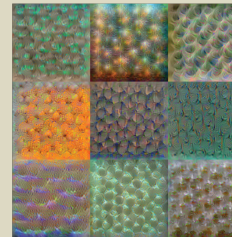
AI models/CNN



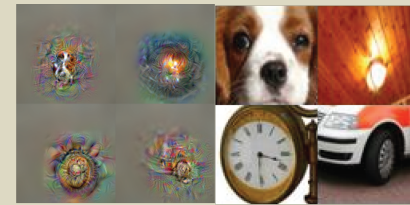
Classify



Conv 1: Edge+Blob



Conv 3: Texture



Conv 5: Object Parts



Fc8: Object Classes

Histopathological images
(cancer dataset)

- AI models can efficiently classify objects or cancers
- AI models can also guide vehicles (driverless cars).
- Convolution Neural Network (CNN) is a popular choice due to high performance.
- CNNs deals with only images

DeepInsight Project

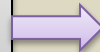
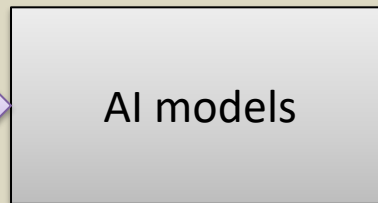
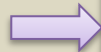
Broader Objectives

- To use AI models (like CNN) for non-image samples.
- To analyze patient information.
- To improve detection rate of diseases/tumors.
- To uncover novel phenotypes.
- To predict drugs using patient information.

-towards personalized medicine

Data

(patient information, RNA seq data, methylation, mutation, clinical data, drug data, non-image samples)



- 1) Accurate tumor detection.
- 2) Discover subsets of genes.
- 3) Find drugs using models.
- 4) Classification

**Problem: CNN doesn't work on non-image data
(many patient data are non-image)**

DeepInsight: A concept

Background: several techniques like SVM, Random Forest, etc. exist, however, their performance on biological data are still limited.

Proposal: Can we transform patient data (e.g. RNA seq, clinical etc.) into image data? If so, then power of AI tools (like CNN) can be applied.

This could be a breakthrough as samples (which were not possible to envisage as an image) can be transformed to images and processed by CNNs (see Fig. 1).

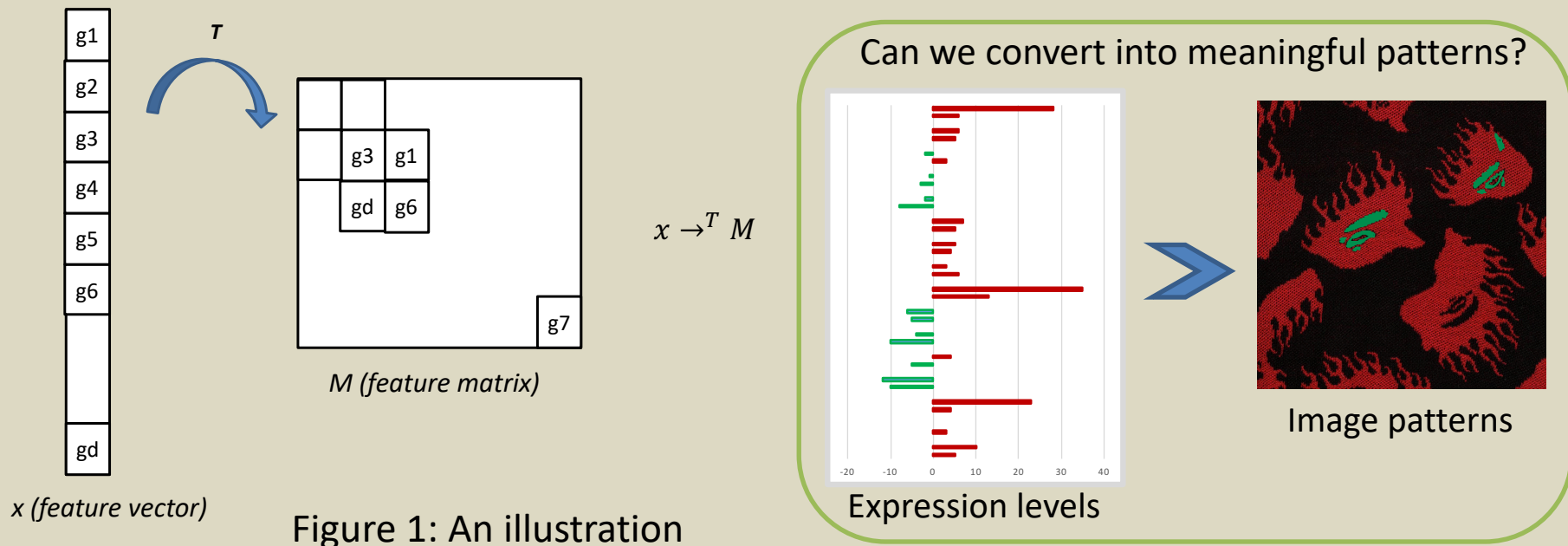


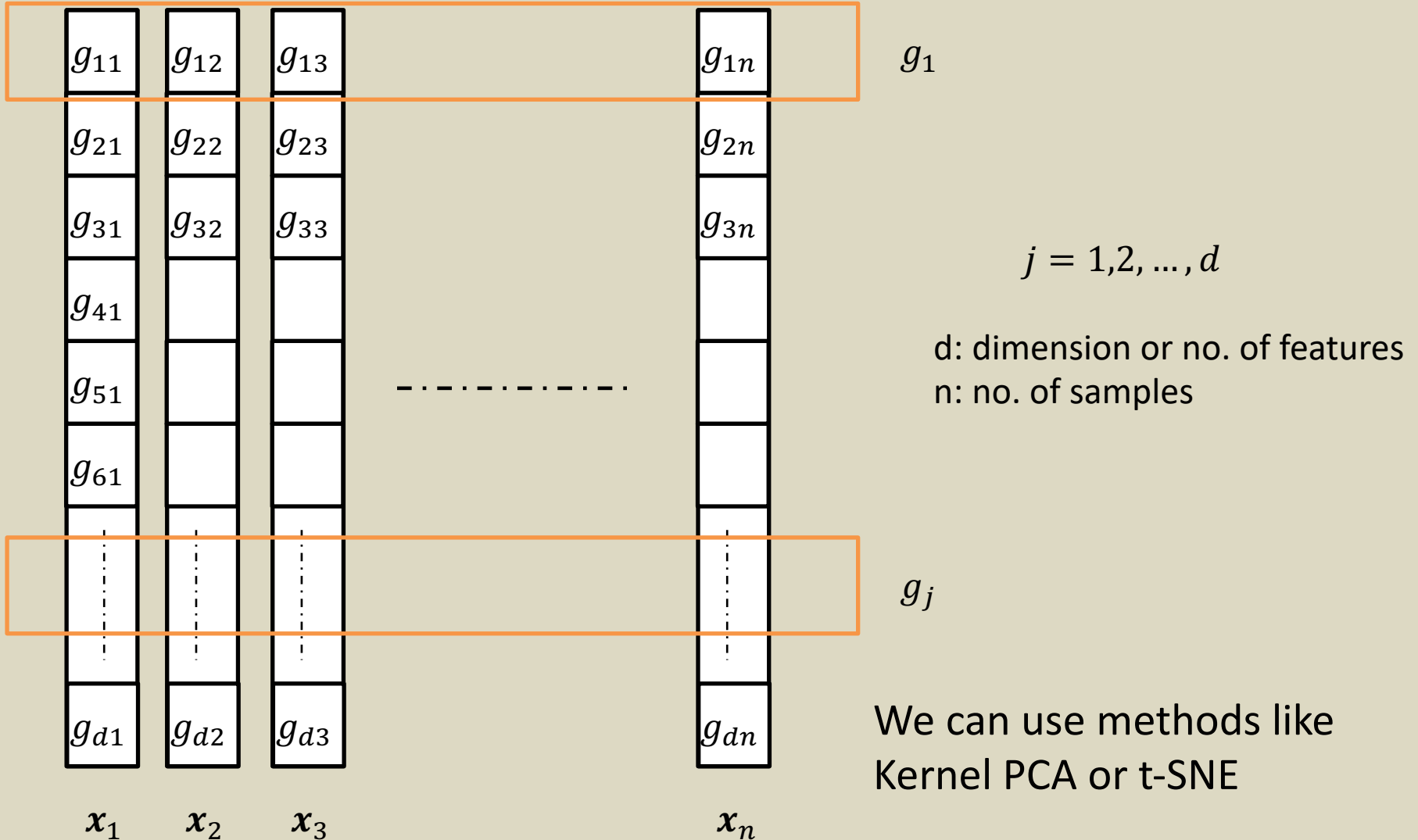
Figure 1: An illustration

We want to find a transform T , which can transform a feature vector x to an image M

Problems

1. How to define a d -dimensional feature vector to a 2-dimensional space?
2. How to find a rectangle encompassing all the data (for data framing)?
3. How to adjust rectangle and data horizontally/vertically?
4. How to find number of rows and columns (A,B)?
5. How to transform from cartesian coordinates to pixel frames?

1. Dimensionality reduction to 2D-space (gene wise NOT sample wise)

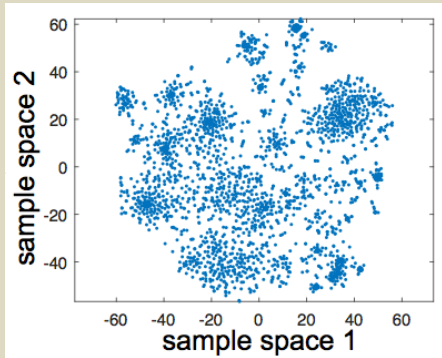


g_1
 g_2
 g_3

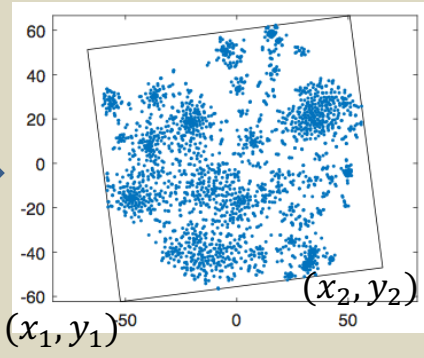
 g_d

genes/
 features

tSNE/
 kPCA

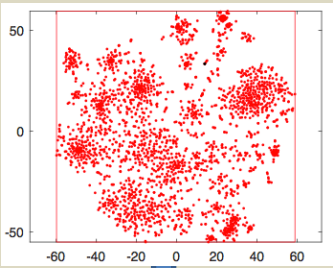


Convex Hull
 Algorithm



Smallest rectangle
 consisting all the data

Rotation



Framing
 (pixels)

Pixel coordinates

Gradient of two corner
 points of a rectangle $G = \frac{y_2 - y_1}{x_2 - x_1}$

$$\theta = \tan^{-1}(G)$$

Rotation matrix $R = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$

Horizontal axis $A_c = |x_2 - x_1|$

Vertical axis $B_c = |y_3 - y_2|$

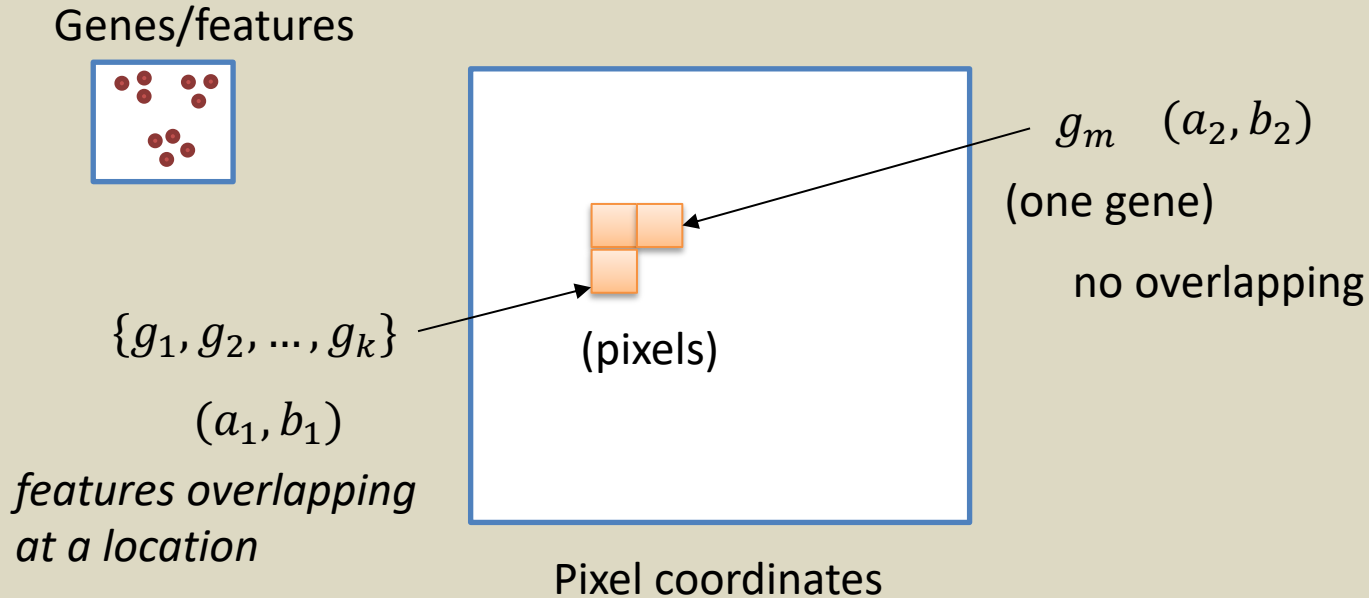
$$A_p = \text{ceil}(A_c \times \frac{\text{Precision}}{d_{min}})$$

$$x_p = \text{round}(1 + \frac{(x_c - x_{min}) * A_p}{x_{max} - x_{min}})$$

$$y_p = \text{round}(1 - B_p \frac{(y_c - y_{min})}{y_{max} - y_{min}})$$

Feature Mapping

- Mapping of genes/features on pixel locations

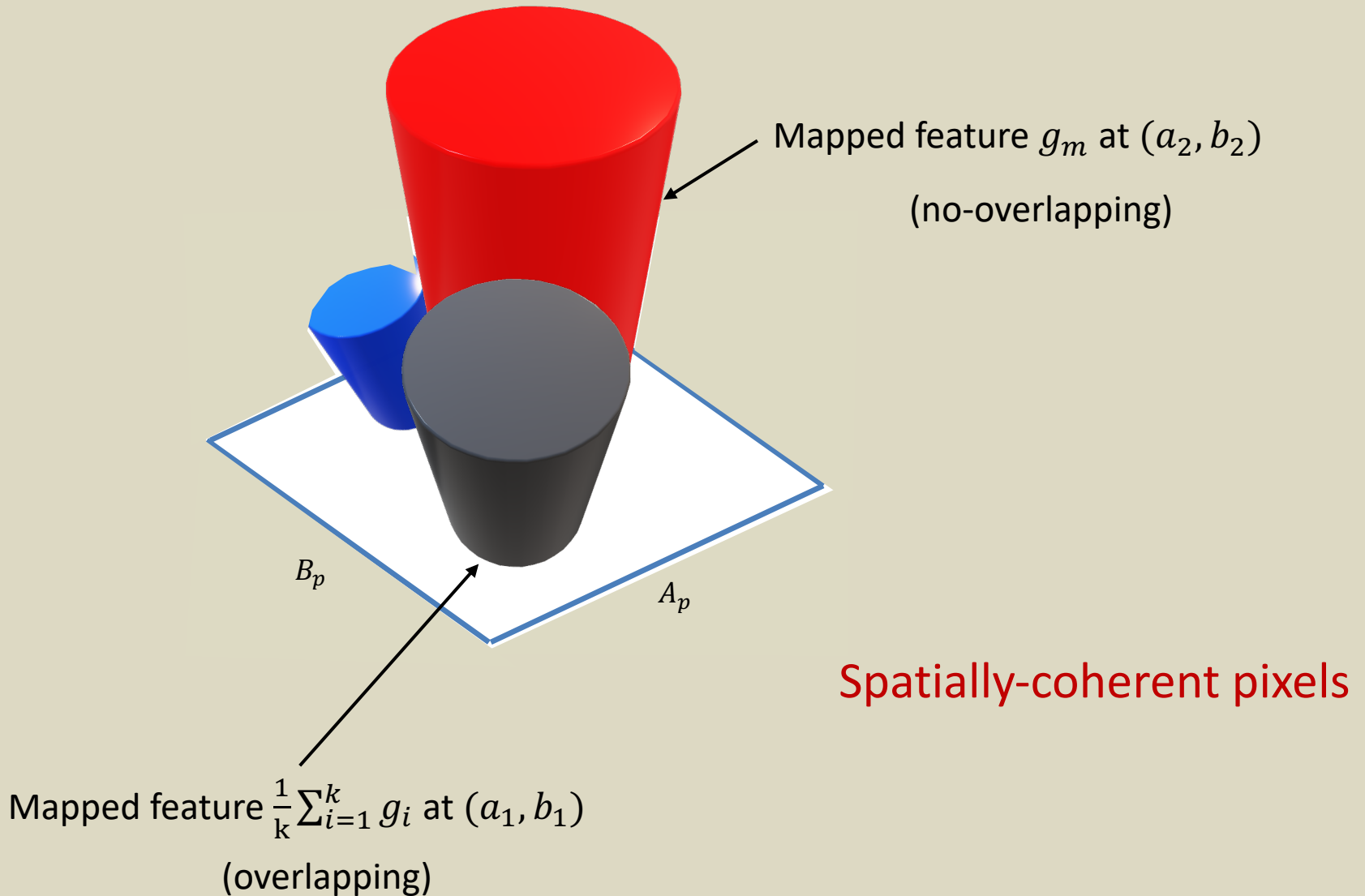


- Once the feature locations are defined using the training set, the next step is to map feature values to these locations.
- If two or more than two features occupy the same location then their averaged values are used, leading to lossy compression of features.

Low-resolution – lossy compression

High-resolution – lossless compression but large image size

Feature Mapping



Plan: An overview of the execution plan is shown in the Fig. below

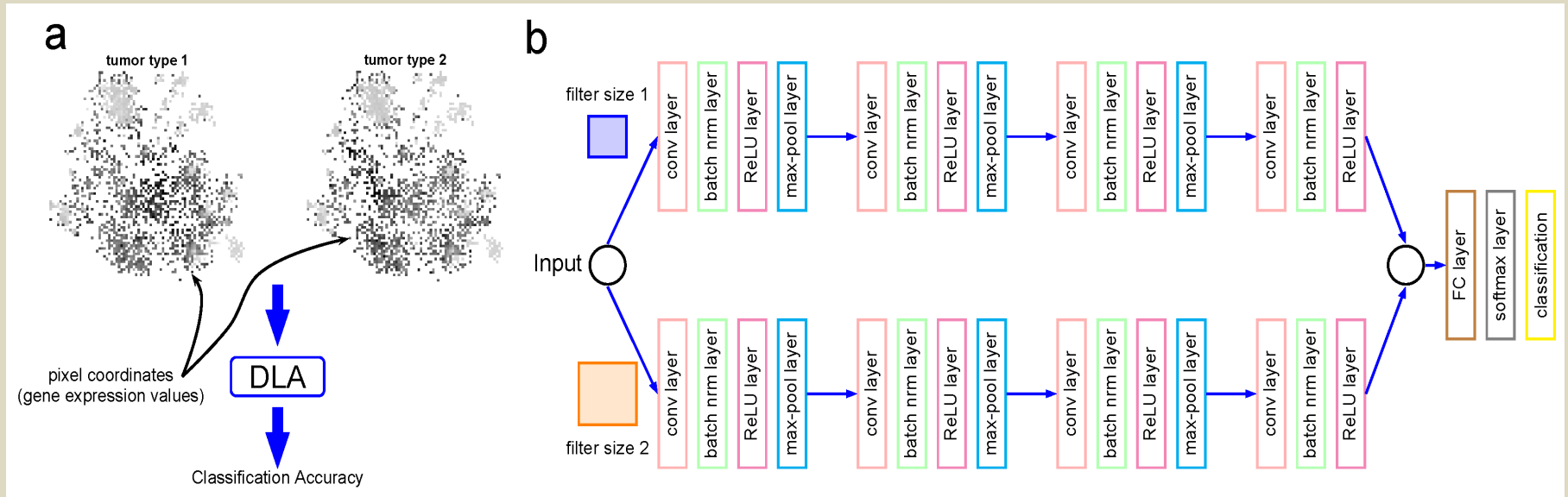


Figure 2 a) An illustration of two types of tumors using the *DeepInsight* method.

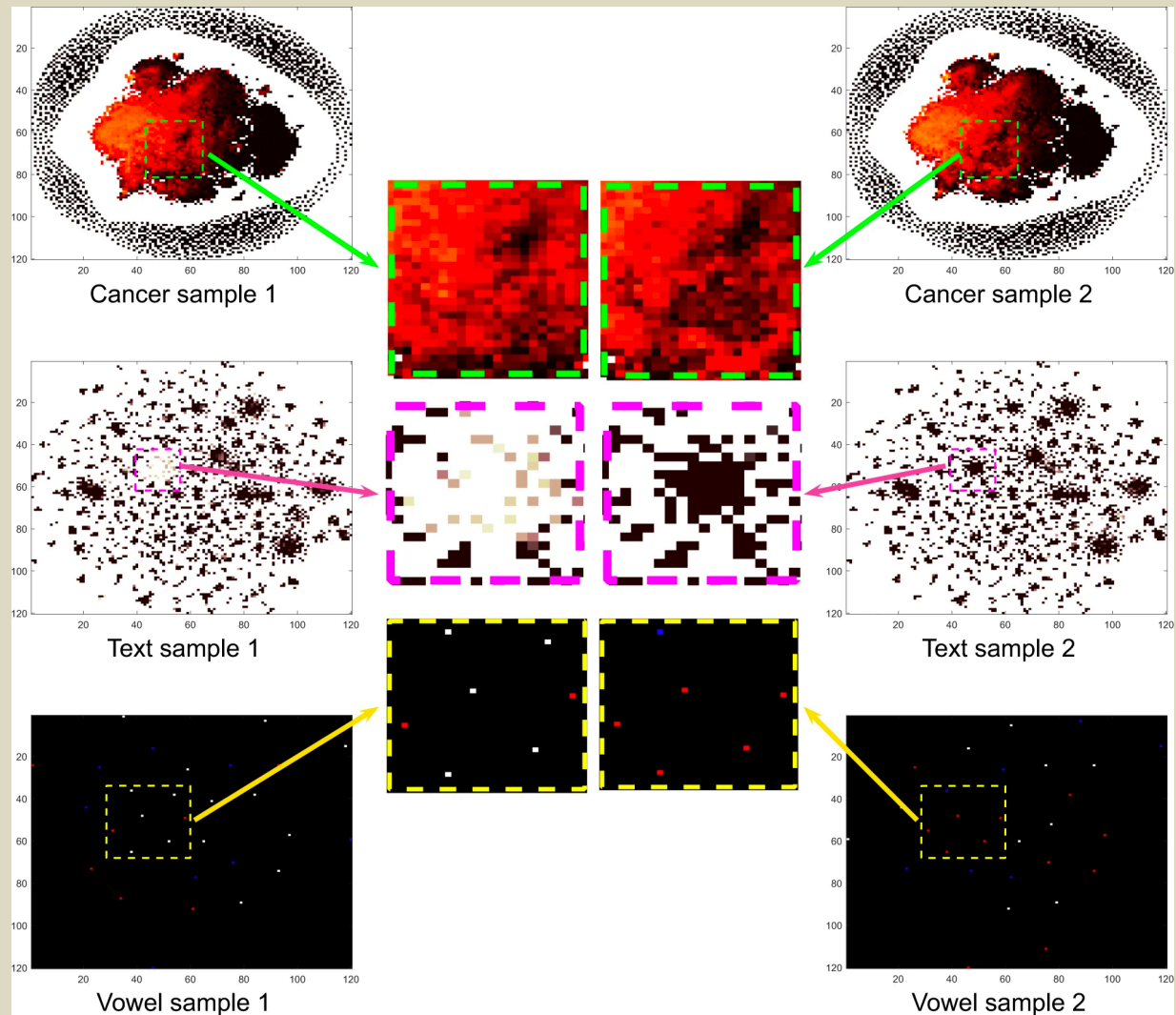
2b) Parallel CNN architecture of the *DeepInsight* method **(a breakthrough method)**
non-image → *image (wide applicability)*

Results: Four different kind of datasets are used.

Datasets	Decision Tree	Ada-Boost	Random Forest	DeepInsight
RNA-seq (gene expression)	0.94	0.29	0.93	0.98~0.99
Vowels (speech)	0.79	0.21	0.89	0.97
Text	0.81	0.78	0.85	0.86~0.88
Artificial (Madelon)	0.67	0.58	0.59	0.83~0.85
Artificial (Ringnorm-DELVE)	0.92	0.78	0.95	0.98~0.99

Further extensions: 1) Multi-omics data can be analyzed as colored image with 3 layers.
 2) Different kind of data can be normalized and analyzed in a single layer.

An illustration with different kinds of data



Revealing patterns Different patterns achieved by DeepInsight on gene-expression (different kind of cancers), text (two types of text) and vowels (two types of vowels). Each plot shows a transformed sample, the difference between samples can now be noticed straightforwardly.

DeepInsight model tasks

It performs 3 the actions

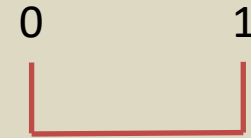
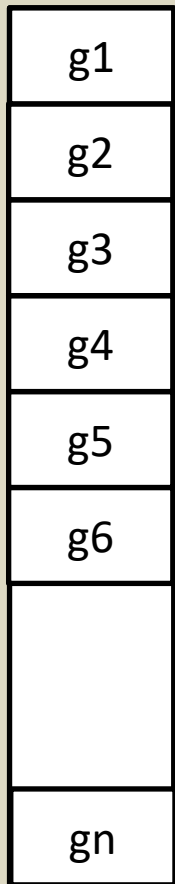
- **Element arrangement** (via mapping)
 - **Feature extraction** and **Classification** (via CNN)
-

Element arrangement

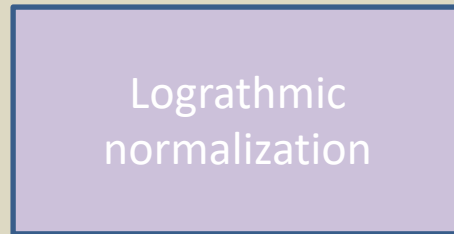
- Dimensionality reduction techniques (DRTs), like tSNE and Kernel PCA, can map data from high-dimensional space to 2D space in a non-linear fashion.
- The samples with similarity, map close to each other, and with dissimilarity mapped apart.
- Many linear DRTs map data to a 2D plane, however, mapped samples are highly convoluted, and it becomes very challenging for clustering algorithms to find a reasonable level of groupings.
- On the other hand, tSNE has the potential to map very high-dimensional data to a 2D plane while keeping the data topology (i.e., with minimum error).
- The tSNE algorithm uses Euclidean distance to compute probabilities. However, in DeepInsight, cosine distance was used.
- The processing of tSNE can be prolonged. For faster processing, Barneshut algorithm is used to approximate joint distributions instead of the exact distribution.

Data Normalization

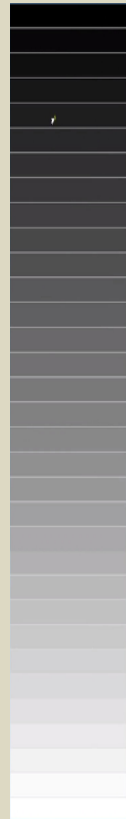
The normalization will bring a feature value between 0 and 1.



$$\begin{aligned} \text{Max}_j &= \max_{\text{samples}} X_{tr}(j, :) \\ \text{Min}_j &= \min_{\text{samples}} X_{tr}(j, :) \\ X_{tr}(j, :) &= \frac{X_{tr}(j, :) - \text{Min}_j}{\text{Max}_j - \text{Min}_j} \\ X_{val}(j, :) &= \frac{X_{val}(j, :) - \text{Min}_j}{\text{Max}_j - \text{Min}_j} \\ X_{test}(j, :) &= \frac{X_{test}(j, :) - \text{Min}_j}{\text{Max}_j - \text{Min}_j} \end{aligned}$$

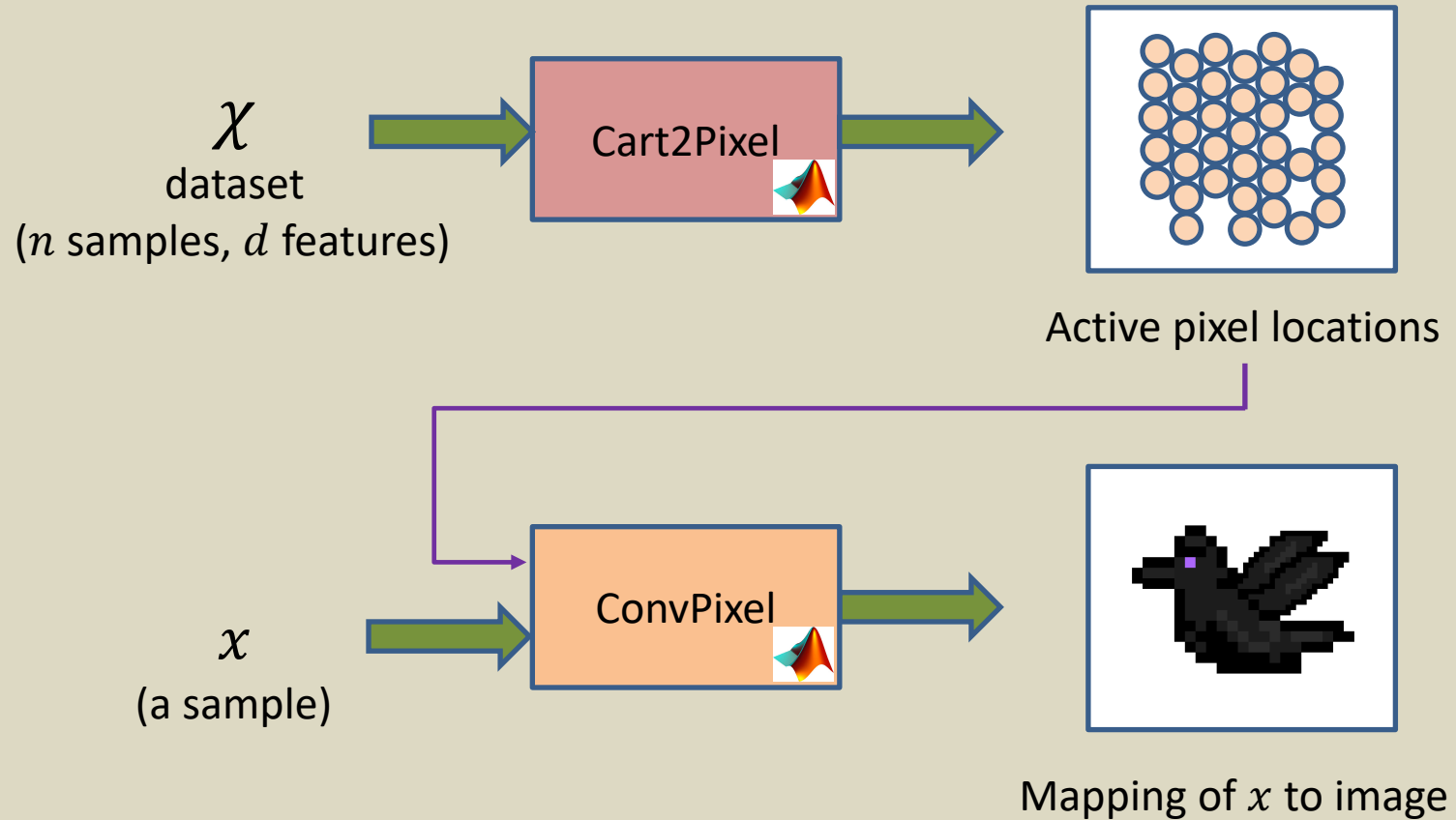


$$\begin{aligned} \text{Min}_j &= \min_{\text{samples}} X_{tr}(j, :) \\ X_{tr}(j, :) &\leftarrow \log(X_{tr}(j, :) + |\text{Min}_j| + 1) \\ \text{Max} &= \max(X_{tr}) \\ X_{tr}(j, :) &\leftarrow \frac{X_{tr}(j, :)}{\text{Max}} \end{aligned}$$

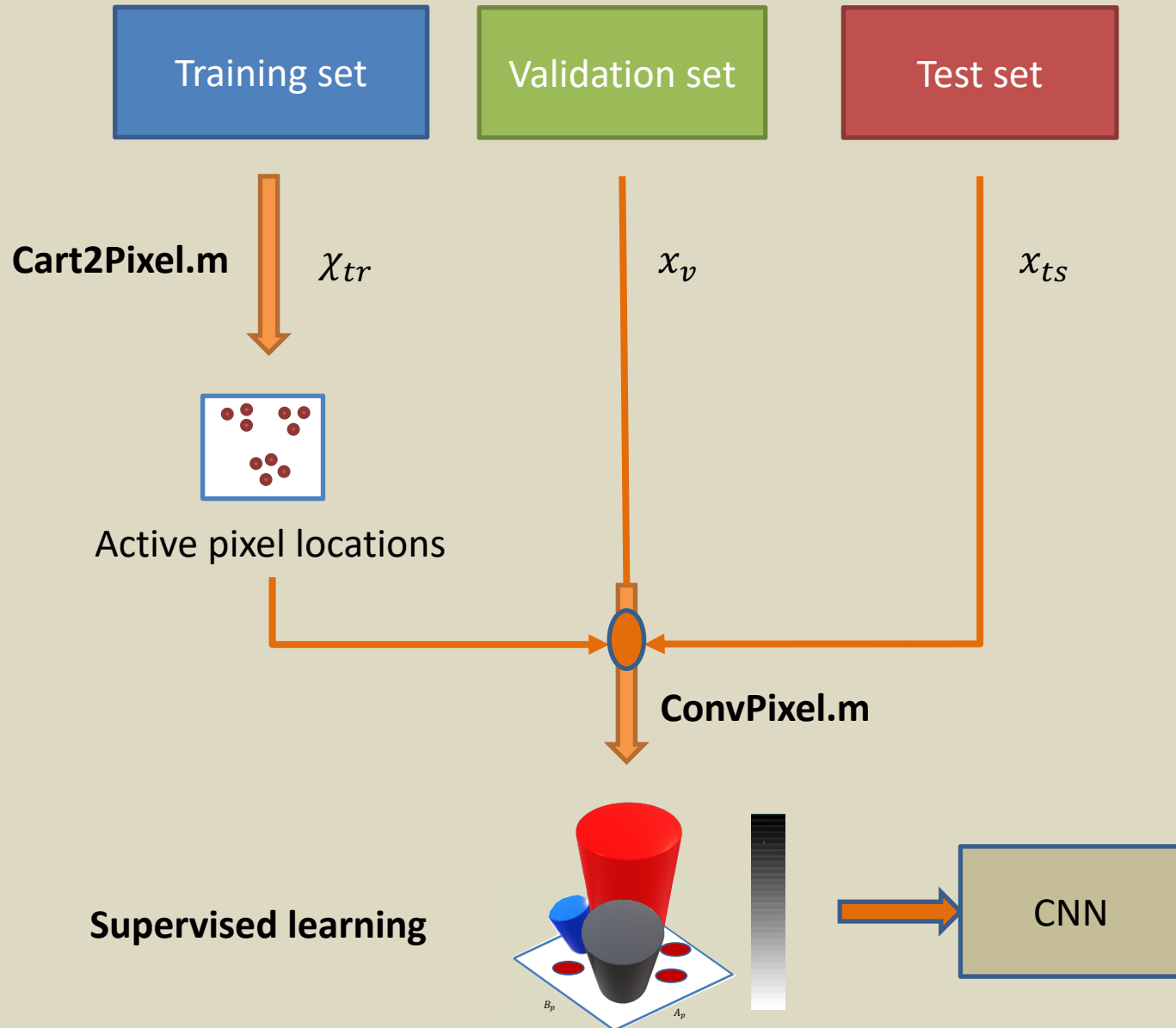


255 shades of grey

Non-image to Image transformation (coding aspect)



Dataset arrangement



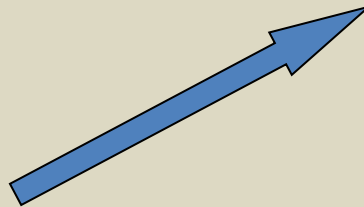
Supervised Learning

What is this?



- apple
- orange

And what is this?



What do we understand from this example?

- A 3-year-old boy can adapt and react to different objects.
- We want to teach similar adaptation to a *Machine* to recognize and provide actions.

Unsupervised Learning

Bag of Fruits, can we arrange?



We want to arrange similar fruits together



We found 3 bags



This is called Clustering

Properties of DeepInsight

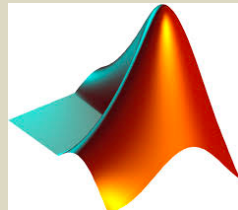
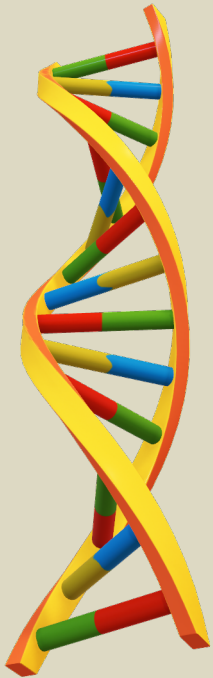
- No need to worry for feature extraction for non-image samples (as required in machine learning techniques).
- Enables the use of CNN which is becoming a promise for images in accuracy for industrial applications.
 - Advantages of CNN becomes an integral part of DeepInsight; i.e., the versatility of CNN is extended to non-image domain.

Tips

- It is not necessary to use parallel layers as done in DeepInsight.
- Convert a non-image sample to image sample by Cart2Pixel.m and ConvPixel.m Matlab codes.
- Build your own specific DNN or use public nets such as Alexnet, SqueezeNet, GoogleNet, Resnet etc. (which are available from Matlab)

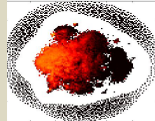
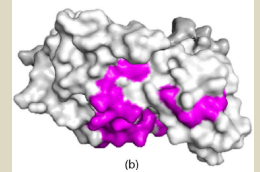
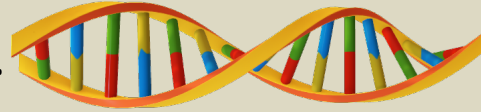
Importance of Matlab in Research

- Deep learning packages, statistics, machine learning, parallel computing, GPU computing and a vast range of toolboxes available.
 - Multithreading capabilities and multi-GPU execution environment.
 - Advance graphics support (2D image, 3D image, video).
 - Very easy to implement without worrying about language syntax.
 - Fast in concept development; such as lesser time required in coding which can promote research advancement in a rapid pace.
 - Very easy to debug via Matlab's editor.
 - Useful in many fields of research not limited to computer science, engineering, bioinformatics, and energy.
-
- Expensive
 - Pay for every toolbox, however, institutional licenses are available having all toolboxes.
 - Restricted use of License (e.g. subject to geographical location).



Possible industrial applications

- Genome data analysis and research.
- Brain-computer interface.
- Proteome data analysis (protein-peptide interactions, DNA binding proteins etc.).
- Data mining and knowledge discovery.
- Health care, personalized medicine & drug discovery (e.g. risk prediction for dementia, disease identification).
- Data visualization.
- Business and finance.



Recent examples

Buturivić and Miljković (2020), introduced an approach inspired by DeepInsight. They applied their approach to gene expression data derived from blood samples of patients with bacterial or viral infections and showed promising results.

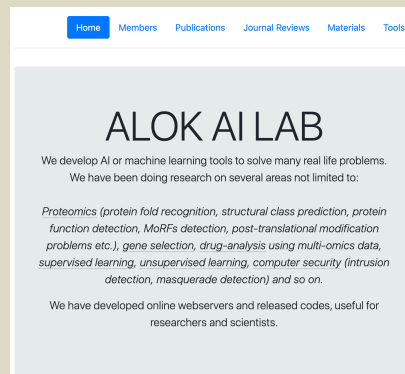


Similarly, Kanber (2020), implemented it using sparse data and showed that on MNIST database with 70k samples, DeepInsight, shown superior performance than a state-of-the-art machine learning (random forest) method.



Availability: Paper and Tools

- Sharma A*, Vans E, Shigemizu D, Boroevich KA, Tsunoda T*, DeepInsight: a methodology to transform a non-image data to an image for convolution neural network architecture, ***Scientific Reports***, 9:11399, pp. 1-7, 2019. <https://www.nature.com/articles/s41598-019-47765-6>
- Code/Package: <http://www.alok-ai-lab.com/tools.php> or : https://www.riken.jp/en/research/labs/ims/med_sci_math/



tSNE in brief

- Let a training set has N samples and d features.
- The conditional probability, $p_{j|i}$, is a measure of the probability that sample x_i will pick x_j as its neighbour under Gaussian distribution, where $x_i \in \mathbb{R}^d$ and $x_j \in \mathbb{R}^d$ (note, it is not d here). It can be defined as:

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)}$$

where σ_i is the variance of the Gaussian that is centered at x_i .

- Similarly, the conditional probability in 2D-plane, $q_{j|i}$, for mapped samples y_i and y_j (where $y \in \mathbb{R}^2$) can be given as

$$q_{j|i} = \frac{\exp(-\|y_i - y_j\|^2 / 2\sigma_{yi}^2)}{\sum_{k \neq i} \exp(-\|y_i - y_k\|^2 / 2\sigma_{yi}^2)}$$

where variance $\sigma_{yi} = 1/\sqrt{2}$; i.e., $2\sigma_{yi}^2 = 1$.

- If $p_{j|i} = q_{j|i}$ then it means sample points, y_i and y_j , correctly model the similarity between the higher dimensional samples, x_i and x_j . This is done by minimizing Kullback-Leibler divergence with respect to y_i , using a gradient descent method as

$$C = \sum_i KL(P_i || Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

Thank you
ありがとうございます

Q/A



Email: alok.fj@gmail.com