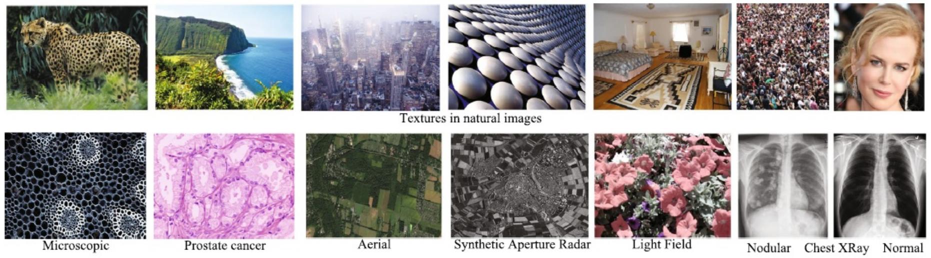
COMP 3411/9414 Artificial Intelligence

AI in Computer Vision – Part 2

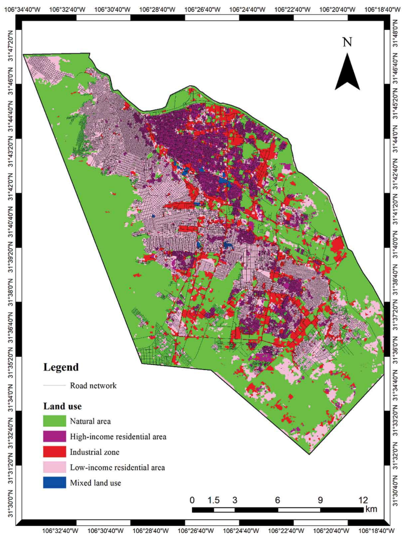
Texture Classification

- Objective of study:
  – Classifying/recognising different texture patterns automatically
  – It's been a fundamental computer vision problem with active research and industry development for multiple decades.
  – Real life applications in many different fields



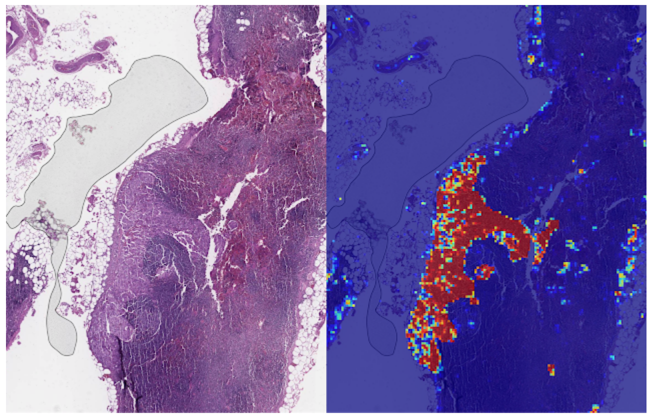
Source: L. Liu et al. *From BoW to CNN: two decades of texture representation for texture classification.* International Journal of Computer Vision, 2019.

Example – urban planning



Source: I. Hernandez & W. Shi. *A random forests classification method for urban land-use mapping integrating spatial metrics and texture analysis.* International Journal of Remote Sensing, 2018.

Example – cancer detection



Source: https://ai.googleblog.com/2018/10/applying-deep-learning-to-metastatic.html

Example – cancer detection

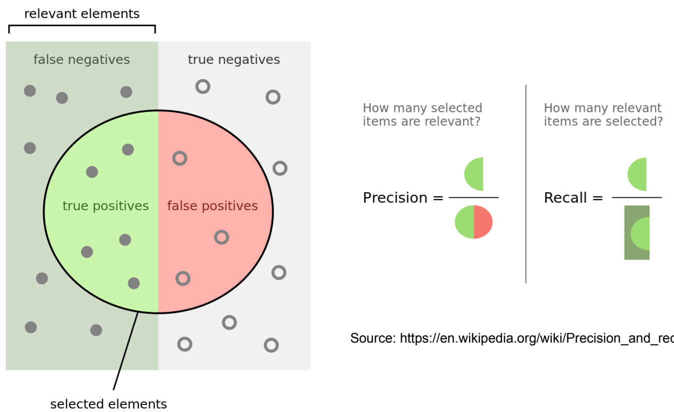
Source: https://ai.googleblog.com/2018/10/applying-deep-learning-to-metastatic.html

Texture Classification

- How to solve this problem with artificial intelligence?
  - Pre-processing?
  - Feature representation?
  - Classifier?
  - Training data?
  - Evaluation metrics?

Evaluation Metrics

- Classification is often evaluated using *Precision* and *Recall*.



Source: https://en.wikipedia.org/wiki/Precision_and_recall

Classifier

- K-nearest neighbour
- Linear regression
- Decision trees
- Artificial Neural Networks
- Support vector machine
- …

- A good introduction: https://www.dezyre.com/article/top-10-machine-learning-algorithms/202

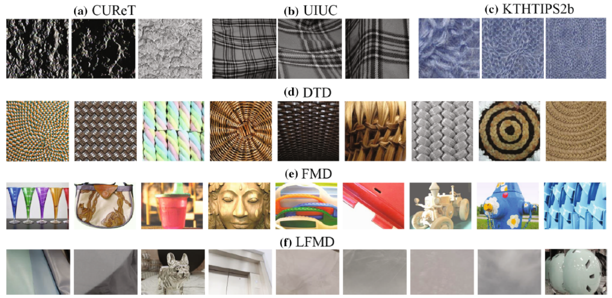
Training Data

- Six well-known public datasets are commonly used for benchmarking

Name	# classes	# images
CUReT	92	5612
UIUC	25	1000
KTHTIPS2b	11	4752
FMD	10	1000
LFMD	12	1200
DTD	47	5640

Feature Representation

- How to encode the texture information effectively so that the images can be classified accurately?
- Critical problems:
  - intra-class variation and inter-class ambiguity



Feature Representation

- Feature representation methods evolve over time:
  - Since 1970s: morphological feature descriptors (Haralick, Gabor)
  - Since 2000: high-dimensional texture descriptors (LBP, HOG, SIFT)
  - Also since 2000: feature encoding (BoW, Texton, FV)
  - Since 2012: deep learning (AlexNet, VGG, ResNet)
- All methods are widely used in current studies.
- The latter two are learning-based approaches.
- Still a difficult problem with low classification accuracies.

Feature Descriptors

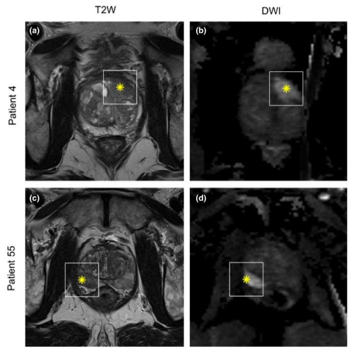
- Haralick features:
  - R. Haralick and I. Dinstein. *Textural features for image classification*. IEEE Trans Syst Man Cybern, 1973.
  - A set of morphological features, to describe the texture features with meaningful concepts (e.g. contrast, variance, entropy)

Angular Second Moment	$\sum_i \sum_j p(i,j)^2$
Contrast	$\sum_{n=0}^{N_g-1} n^2 \{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j)\}, \lvert i-j \rvert = n$
Correlation	$\frac{\sum_i \sum_j (ij)p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}$ where μ_x, μ_y, σ_x, and σ_y are the means and std. deviations of p_x and p_y, the partial probability density functions
Sum of Squares: Variance	$\sum_i \sum_j (i-\mu)^2 p(i,j)$
Inverse Difference Moment	$\sum_i \sum_j \frac{1}{1+(i-j)^2} p(i,j)$
Sum Average	$\sum_{i=2}^{2N_g} i p_{x+y}(i)$ where x and y are the coordinates (row and column) of an entry in the co-occurrence matrix, and $p_{x+y}(i)$ is the probability of co-occurrence matrix coordinates summing to $x+y$

Sum Variance	$\sum_{i=2}^{2N_g} (i - f_8)^2 p_{x+y}(i)$
Sum Entropy	$-\sum_{i=2}^{2N_g} p_{x+y}(i) \log\{p_{x+y}(i)\} = f_8$
Entropy	$-\sum_i \sum_j p(i,j) log(p(i,j))$
Difference Variance	$\sum_{i=0}^{N_g-1} i^2 p_{x-y}(i)$
Difference Entropy	$-\sum_{i=0}^{N_g-1} p_{x-y}(i) \log\{p_{x-y}(i)\}$
Info. Measure of Correlation 1	$\frac{HXY-HXY1}{\max\{HX,HY\}}$
Info. Measure of Correlation 2	$(1-\exp[-2(HXY2-HXY)])^{\frac{1}{2}}$ where $HXY = -\sum_i \sum_j p(i,j) \log(p(i,j))$, HX, HY are the entropies of p_x and p_y, $HXY1 = -\sum_i \sum_j p(i,j) \log\{p_x(i) p_y(j)\} HXY2 = -\sum_i \sum_j p_x(i) p_y(j) \log\{p_x(i) p_y(j)\}$
Max. Correlation Coeff.	Square root of the second largest eigenvalue of $\mathbf{Q}$ where $\mathbf{Q}(i,j) = \sum_k \frac{p(i,k)p(j,k)}{p_x(i)p_y(k)}$

Feature Descriptors

- Haralick features:
  - Commonly used nowadays in medical imaging studies due to its simplicity and interpretability

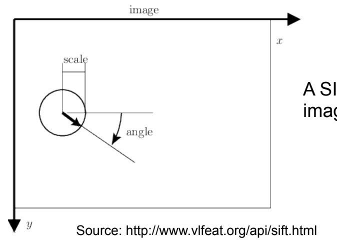


C. Jensen et al. *Assessment of prostate cancer prognostic Gleason grade group using zonal-specific features extracted from biparametric MRI using a KNN classifier*. Journal of Applied Clinical Medical Physics, 2019.

1. Pre-processing
2. Extract Haralick, run-length and histogram features from the region of interest
3. Feature selection
4. Classification using kNN

Feature Descriptors

- Scale-invariant feature transform (SIFT):
  - D. Lowe. *Distinctive image features from scale-invariant keypoints*. International Journal of Computer Vision, 2004.
  - Probably the most popular texture feature descriptor prior to the success of deep learning
  - Moved away from meaningful descriptions, but more numerical, abstract and high-dimensional



A SIFT keypoint is a circular image region with an orientation.

Source: http://www.vlfeat.org/api/sift.html

Feature Descriptors

- Scale-invariant feature transform (SIFT):
  - D. Lowe. *Distinctive image features from scale-invariant keypoints*. International Journal of Computer Vision, 2004.
  - Probably the most popular texture feature descriptor prior to the success of deep learning

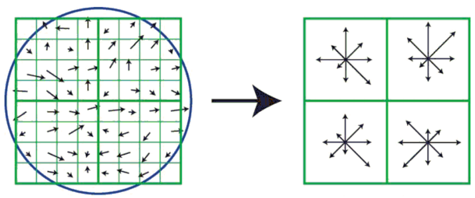


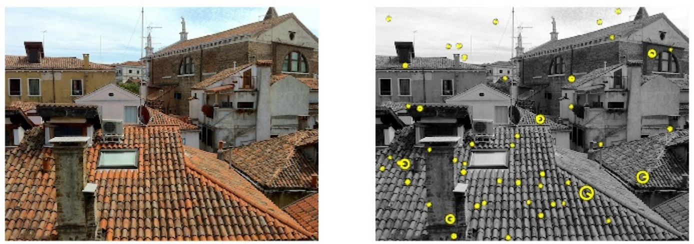
Image gradients

Keypoint descriptor

A keypoint descriptor is 128-dimensional (i.e. it's a vector of 128 elements).

Feature Descriptors

- Scale-invariant feature transform (SIFT):
  - Often used to perform image matching



Source: http://www.vlfeat.org/overview/sift.html

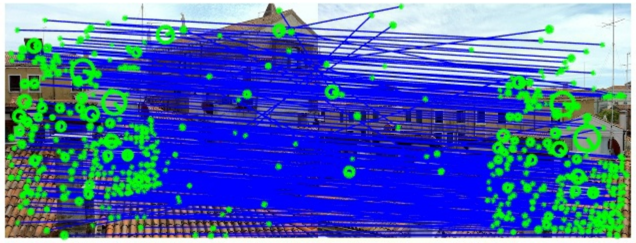
Feature Descriptors

- Scale-invariant feature transform (SIFT):
  - Often used to perform image matching

Source: http://www.vlfeat.org/overview/sift.html

Feature Descriptors

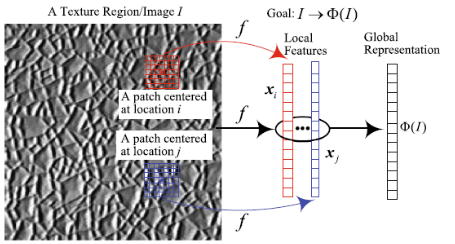
- Scale-invariant feature transform (SIFT):
  - Often used to perform image matching



Source: http://www.vlfeat.org/overview/sift.html

Feature Descriptors

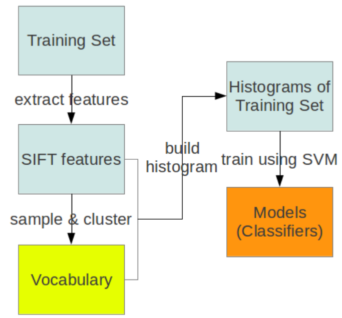
- Scale-invariant feature transform (SIFT):
  - How to use SIFT for texture classification, with so many keypoints?
  - Answer: Feature Encoding
    - Integrate the local features (i.e. SIFT keypoint descriptors) of an image into a global vector to represent the whole image



Source: L. Liu et al. *From BoW to CNN: two decades of texture representation for texture classification.* International Journal of Computer Vision, 2019.

Feature Encoding

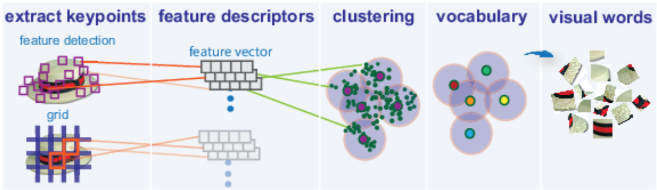
- The most popular method: Bag-of-Words (BoW)
- Local image features are encoded into a histogram to represent the overall image characteristics.



Source: http://cs.brown.edu/courses/cs143/2011/results/proj3/hangsu/

Feature Encoding

- Bag-of-Words (BoW)
  - Step 1: create the vocabulary from the set of local descriptors (e.g. SIFT keypoint descriptors) extracted from the training data
  - This vocabulary represents the categories of local descriptors.



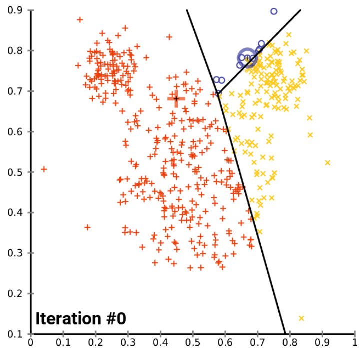
Source: https://au.mathworks.com/help/vision/ug/image-classification-with-bag-of-visual-words.html

Feature Encoding

- Bag-of-Words (BoW)
  - Main technique used to create the vocabulary: *k-means clustering*
  - K-means clustering is one of the simplest and most popular unsupervised learning approaches that perform automatic clustering (i.e. partitioning) of the training data into multiple categories

Feature Encoding

- Bag-of-Words (BoW)
  - K-means clustering: the data are iteratively grouped into different clusters based on their distances from the cluster centres.



Source: https://en.wikipedia.org/wiki/K-means_clustering

Feature Encoding

- Bag-of-Words (BoW)
  - The cluster centres become the "vocabularies", i.e. a set of "visual words".
  - Individual local feature descriptor (e.g. SIFT keypoint descriptor) is assigned to one visual word with the smallest distance.

Source: https://au.mathworks.com/help/vision/ug/image-classification-with-bag-of-visual-words.html

Source: https://en.wikipedia.org/wiki/K-means_clustering

Feature Encoding

- Bag-of-Words (BoW)
  - For an image, the number of local descriptors assigned to each visual word is computed.
  - The numbers are concatenated into a vector => the BoW representation of the image

Source: https://au.mathworks.com/help/vision/ug/image-classification-with-bag-of-visual-words.html

Source: https://en.wikipedia.org/wiki/K-means_clustering

Feature Encoding

- Bag-of-Words (BoW)
  - BoW is often considered as a histogram representation since it counts the occurrence frequencies of visual words.
  - The BoW feature vectors are derived for all training images, and used to train a classifier to perform the texture classification.

Source: https://au.mathworks.com/help/vision/ug/image-classification-with-bag-of-visual-words.html

Source: https://en.wikipedia.org/wiki/K-means_clustering

Feature Encoding

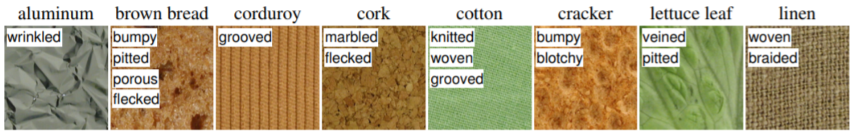
- BoW recap:
  - Extract a set of local feature descriptors
  - K-means clustering and generate visual words
  - Assign local descriptors to one nearest visual word
  - Count the number of occurrences of the visual words and obtain the histogram feature representation
- What are the main limitations with BoW?

Feature Encoding

- Go beyond BoW
  - The very powerful model: *Fisher Vector (FV) Encoding*
- Main differences from BoW:
  - Use Gaussian Mixture Models instead of k-means clustering
  - Soft assignment instead of assigning to the nearest visual word
  - Computation of first and second-order distances
  - Resulting in a very high-dimensional feature representation (e.g. 64K elements in the final feature vector)

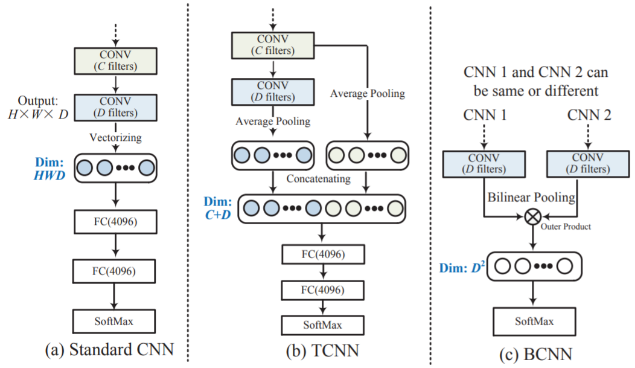
Feature Encoding

- Fisher Vector Encoding:
  - Works best with the Support Vector Machine (SVM) classifier, due to the high-dimensionality of Fisher Vectors.
  - Achieved the highest texture classification performance prior to deep learning approaches
  - Further reading:
    - M. Cimpoi et al. *Describing textures in the wild*. CVPR, 2014.



Deep Learning

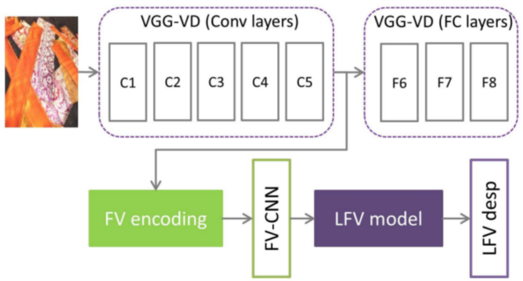
- Standard CNN models have been customised for better texture classification results.



Source: L. Liu et al. *From BoW to CNN: two decades of texture representation for texture classification*. International Journal of Computer Vision, 2019.

Deep Learning

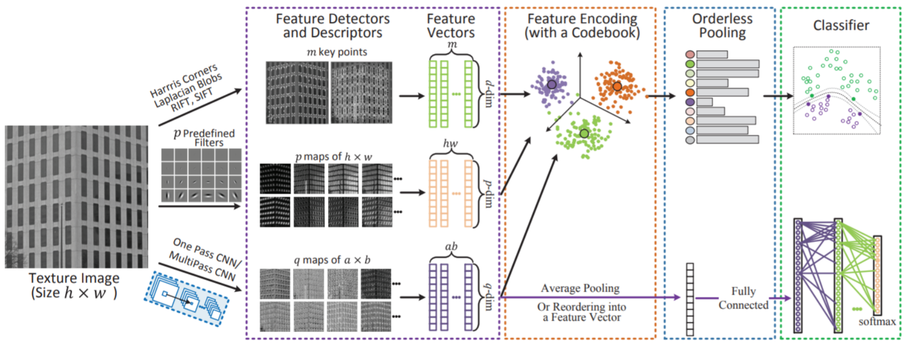
- CNN models have been integrated with Fisher Vector encoding to provide state-of-the-art texture classification performance.



Source: Y. Song et al. *Locally-transferred Fisher Vectors for texture classification*. ICCV, 2017.

Feature Encoding vs. Deep Learning

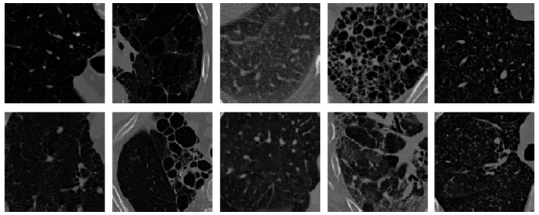
- A unified model:



Source: L. Liu et al. *From BoW to CNN: two decades of texture representation for texture classification*. International Journal of Computer Vision, 2019.

Application

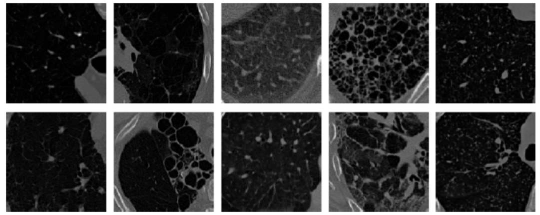
- Classification of interstitial lung disease (ILD):
  - ILD represents a group of more than 150 disorders of the lung parenchyma.
  - Highly challenging with high intra-class variation and inter-class ambiguity



Normal Emphysema Ground glass Fibrosis Micronodules

Application

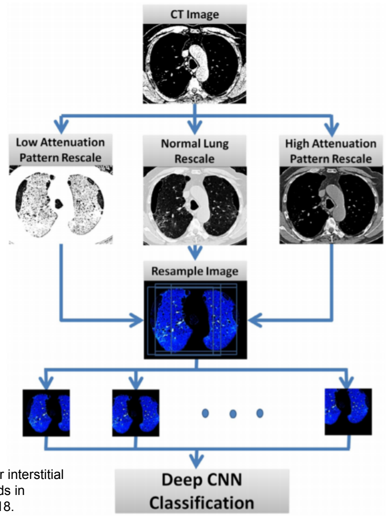
- Classification of ILD:
  - Many approaches have been developed over the past a few years.
  - Mainly two categories of approaches:
    - Texture feature encoding + classification
    - Deep learning



Normal Emphysema Ground glass Fibrosis Micronodules

Application

- Classification of ILD:
  - Integrating domain knowledge with deep learning has shown highly promising results on ILD classification
  - This is an important trend in AI and deep learning research.



Source: M. Gao et al. Holistic classification of CT attenuation patterns for interstitial lung diseases via deep convolutional neural networks. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 2018.

Summary

- Texture classification is an important and fundamental problem in many applications.
- The methods have evolved from morphological representation to deep learning.
- Classification accuracy is still unsatisfactory in many cases.
- Major design choices to make:
  - Pre-processing?
  - Feature representation?
  - Classifier?
  - Training data?
  - Evaluation metrics?

Further Reading

- L. Liu et al. *From BoW to CNN: two decades of texture representation for texture classification*. International Journal of Computer Vision, 2019.
- A very useful computer vision toolkit with concise tutorials. http://www.vlfeat.org/
- A list of popular classifier models. https://www.dezyre.com/article/top-10-machine-learning-algorithms/202

Further Reading

- K. Simonyan et al. *Fisher vector faces in the wild*. BMVC, 2013.