# COMP 3411/9414 Artificial Intelligence

AI in Computer Vision



# Outline

- Introduction
  - Applications of AI in computer vision
- Overall framework
  - Handwritten digit recognition as a case study
- Advanced topics
  - Texture classification
  - AI in biomedical imaging
- Reference readings



# Introduction

- AI has many applications:
  - e.g. medicine, robotics, finance, transportation, marketing, arts
- Many new trends of AI technologies are developed in Computer Vision applications:
  - e.g. face recognition, surveillance, remote sensing, medical imaging, self-driving cars
  - A highly active area in both industry and research



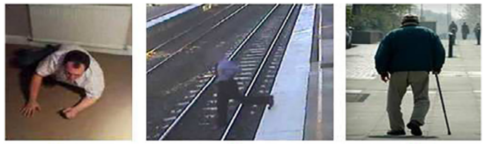
# Introduction

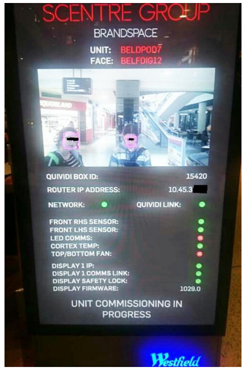
- Application – Face recognition
  - One of the most well-studied areas in computer vision
  - Popular algorithms: eigenface, distance metric learning
  - Industrial applications in public security, police operation, access control, advertising

Source: O. Parkhi et al. *Deep face recognition*. BMVC, 2015.

# Introduction

- Application – Surveillance
  – Many different applications: object tracking, event detection, summarisation, human behaviour analysis, vehicle re-identification
  – A vast range of image/video analysis methods have been developed.



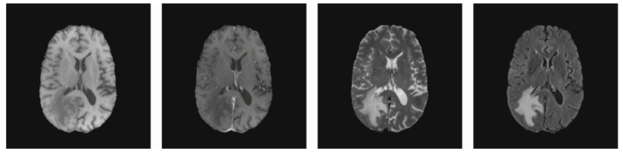



Source: A. Mabrouk and E. Zagrouba. *Abnormal behaviour recognition for intelligence video surveillance systems: a review.* Expert Systems with Applications, 2018.
https://www.news.com.au/finance/business/retail/westfield-is-using-facial-detection-software-to-watch-how-you-shop/news-story/7d0653eb21fe1b07be51d508bfe46262



# Introduction

- Application – Medical imaging
  – Computer vision/machine learning techniques are becoming increasingly important to medical diagnosis and treatment planning.
  – Applications in lesion segmentation, tumour classification, survival prediction, drug development, nanotechnology



(a) T1 (b) T1ce (c) T2 (d) FLAIR

Source: D. Liu et al. *3D large kernel anisotropic network for brain tumor segmentation.* ICONIP, 2018.



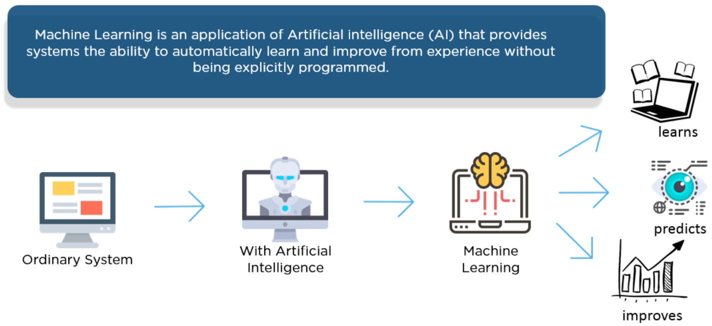
# Key Learning Outcomes

- Main components of a computer vision framework
  – **Machine learning** – learning-based classification, designing a machine learning algorithm to classify the input data
  – Feature extraction – represent the images by feature vectors
  – Evaluation – ways to evaluate the performance of the system



# Machine Learning

- Machine Learning, as a subfield of AI, is one of the main components in Computer Vision studies.




Source: https://www.quora.com/How-is-machine-learning-related-with-artificial-intelligence

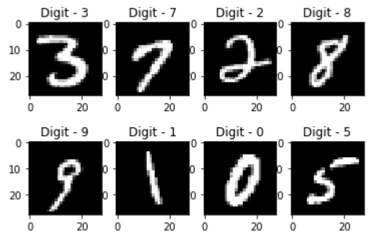
# Machine Learning

- Use **handwritten digit recognition** as a case study
  - A classical machine learning problem
  - Can be solved without the feature extraction component
- Key points:
  - Typical way of implementing a machine learning method
  - Overview of popular classifier models
  - Resources for further study



# Handwritten Digit Recognition

- Objective of study:
  - Recognising handwritten digits 0-9 automatically
  - Example: https://neuralnetwork--ieatpython.repl.co/
  - Applications in the industry: read postcode, bank cheques

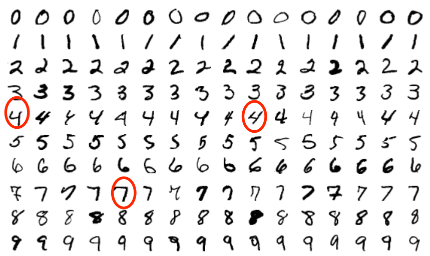





# Handwritten Digit Recognition

- An intuitive solution: K-Nearest Neighbour (kNN)

**2**



1. Find *K* samples that are *most similar** to the test image
2. Determine which class the majority of these *K* samples belong to
3. Assign the test image to this class

** Similarity is measured by certain distance functions between vectors, e.g. Euclidean distance* $||x_1-x_2||$

If using the image pixels as input directly with Euclidean distance, *K*=3 nearest neighbours give a 5% error rate.



# Handwritten Digit Recognition

- Methodology:

  Step 1: Construct a database for training

  Step 2: Image pre-processing

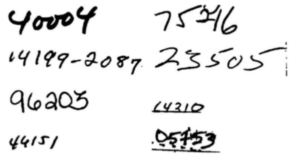
  Step 3: Create a machine learning / classifier model

  Step 4: Apply the machine learning on a test image

# Handwritten Digit Recognition

- Handwritten digit recognition with a back-propagation network (Y. Le Cun et al. 1990)

  Step 1 – Create dataset
  - Construct a dataset of 9298 segmented numerals digitised from handwritten postcodes and 3349 printed digits of 35 different fonts
  - The training set contains 7291 handwritten digits and 2549 printed digits.
  - The test set contains 2007 handwritten digits and 700 printed digits.
  - Important: both sets contain numerous examples that are ambiguous, unclassifiable, or even misclassified.





# Handwritten Digit Recognition

- Handwritten digit recognition with a back-propagation network (Y. Le Cun et al. 1990)

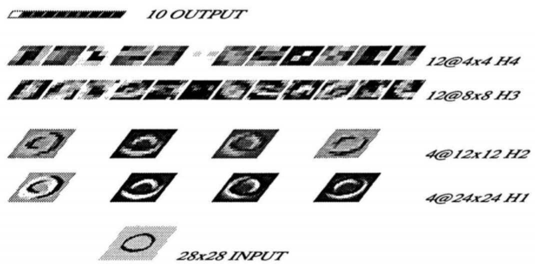
  Step 2 – Image pre-processing
  - Thresholding, segmentation of individual digits
  - Normalisation of digit sizes



# Handwritten Digit Recognition

- Handwritten digit recognition with a back-propagation network (Y. Le Cun et al. 1990)

  Step 3 – Create a multi-layer neural network (LeNet-1)
  - Input: normalised digit image
  - Output: digit category (0-9)






# Handwritten Digit Recognition

- Handwritten digit recognition with a back-propagation network (Y. Le Cun et al. 1990)

  Step 4 – Apply on the test image
  - Pre-processing
  - Then apply the trained neuron network to obtain the digit category
  - Evaluation using error rate (3.4%) and mean square error (0.024)

# Key Point 1

- To design a machine learning system:
  - Need to construct a training set and a testing set
  - Training produces a machine learning / classifier model, which is then applied to the test image to generate the classification output
  - Some image pre-processing might be needed before the training and testing processes



# Handwritten Digit Recognition

- Dataset is important
  - Creation of the MNIST database, 1998
    - Training set has 60,000 samples; Test set has 10,000 samples
    - Pre-processed with size normalisation (28x28 pixels) and image centering
  - The large size of this database enables many subsequent studies of more advanced methodologies



# Handwritten Digit Recognition

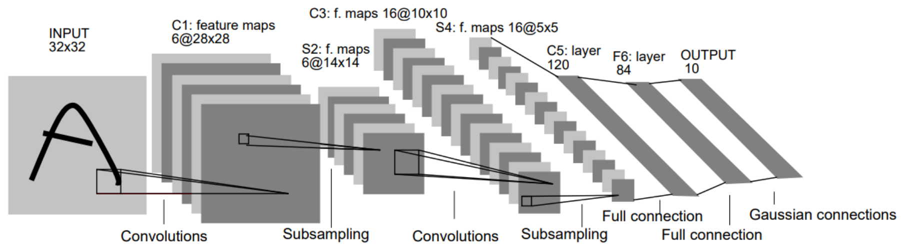
- Design of machine learning / classifier models is important

| Classifier used | Best performance achieved / lowest error rate (%) |
|---|---|
| Linear classifier | 7.6 (LeCun et al. 1998) |
| Non-linear classifier | 3.3 (LeCun et al. 1998) |
| K-Nearest Neighbour | 0.52 (Keysers et al. IEEE TPAMI 2007) |
| Support Vector Machine | 0.56 (DeCoste and Scholkopf, MLJ 2002) |
| Convolutional Neural Network | 0.21 (Wan et al. ICML 2013) |



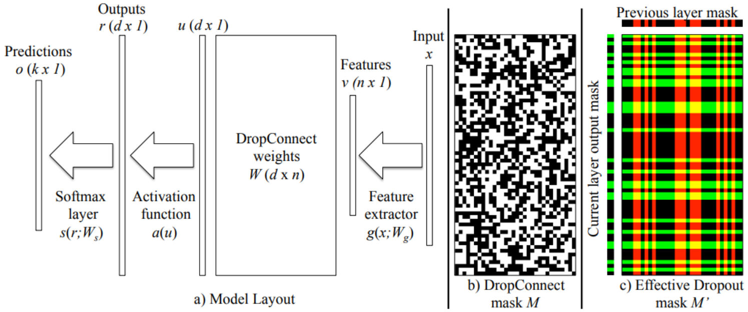
# Handwritten Digit Recognition

- Convolutional Neural Network was proposed in 1998
  - Y. LeCun et al. *Gradient-based learning applied to document recognition.* Proc. Of the IEEE, 1998.
  - The CNN network was named “LeNet-5”, giving an error rate of 0.7%.
  - Used to read 10-20% of all bank cheques in US



# Handwritten Digit Recognition

- State-of-the-art performance (0.21% error rate) achieved in 2013 by introducing *drop-out* into LeNet-5
  - L. Wan et al. *Regularization of neural networks using DropConnect*. ICML, 2013.






# Key Point 2

- To design a machine learning system:
  - Size and quality of the training dataset are critical
  - There are many different choices of classifiers
  - Different classifiers give different performances



# Summary

- AI is critical in computer vision applications
- Modern computer vision development often requires machine or deep learning methodologies
- The same principles apply to other application domains:
  - Financial data processing
  - Autonomous driving
  - Mass media
  - Social media
  - Politics
  - Bioinformatics



# Summary

- Major design choices to make:
  - **Pre-processing?**
  - Feature representation?
  - **Classifier?**
  - **Training data?**
  - Evaluation metrics?
- Next:
  - Feature representation
  - Evaluation metrics

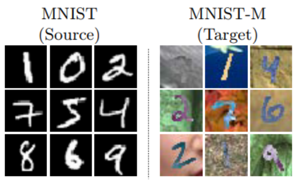
# Further Reading

- Y. LeCun et al. *Gradient-based learning applied to document recognition.* Proc. Of the IEEE, 1998.
- A brief summary of well-known CNN models [https://medium.com/@sidereal/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5](https://medium.com/@sidereal/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5)



# Further Reading

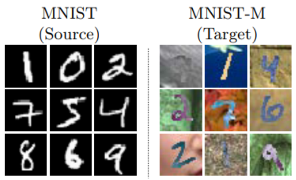
- Go beyond MNIST:
  - MNIST-M dataset was created in 2015 by blending digits from MNIST with colour photos from the BSDS500 dataset
  - The recognition problem becomes much harder
  - The focus of study shifts to *unsupervised domain adaptation*:
    - How to use knowledge learned from MNIST to classify images in MNIST-M?

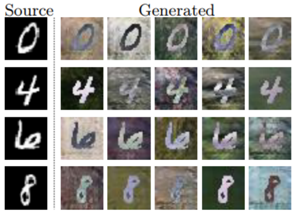





# Further Reading

- Unsupervised domain adaptation:
  - H. Lee et al. *Diverse image-to-image translation via disentangled representations*. ECCV, 2018.
    - Training with source images only: 56.6% accuracy
    - Training with target images only: 96.5% accuracy
    - Training on generated target images: 91.54% accuracy

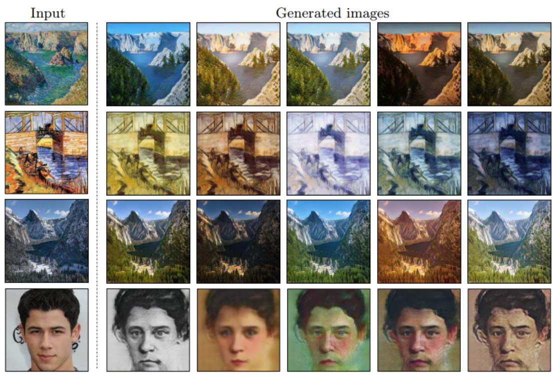





# Further Reading

- H. Lee et al. *Diverse image-to-image translation via disentangled representations*. ECCV, 2018.




Monet => Photo

van Gogh => Monet

winter => summer

photo => portrait