Basics of Machine Learning

Assignments

Dmitry Ryabokon, github.com/dryabokon

Assignments

Scoring

A	90%
B	85%
C	75%
D	65%
E	60%
FX	<60%

Assignments

Scoring

Assignment 1	Assignment 2	Tech Talk
Classic Machine Learning	Computer Vision and Deep Learning	
50%	50%	Extra +30%

Classic Machine Learning

- Supervised Learning
- Naive Bayesian Classifier
- Gaussian Classification Model
- Linear Discriminant Functions
- Decision strategies
- Benchmarking of Supervised Learning Algorithms
- Unsupervised Learning

Supervised learning

Description

- Get train/test data sample from open sources or generate it by yourself. Focus on low-dimension data, preferably 2 dimensions, 2 classes

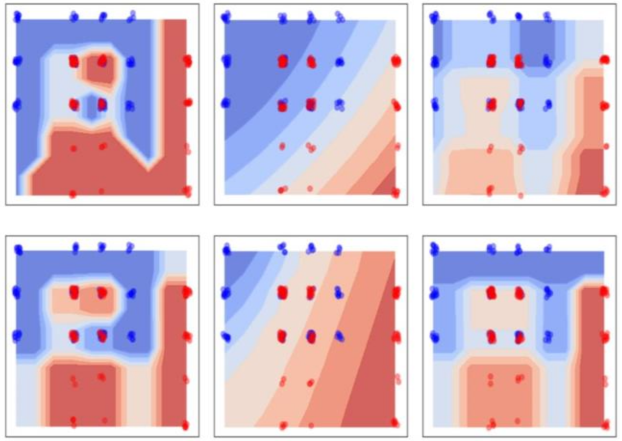
- Visualize the distribution of sample data

- Estimate accuracy of data classification for 3-4 classifiers studied over lecture course

- Build ROC curves and hit maps

Assessment criteria

○	In-house classifier:	40%
○	Out-of-box classifier:	20%
○	Data visualization charts:	20%
○	ROC curve charts:	20%
*	Extra: multi class championship	30%



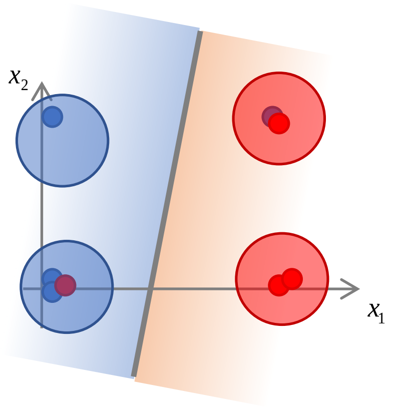
Naive Bayesian Classifier

Description

- Get train/test data sample from open sources, e.g. GBP/USD index. Use daily change (↑, ↓) as a target.
- Visualize the distribution of sample data
- Estimate accuracy of data classification
- Build ROC curves and calculate AUC

Assessment criteria

○ In-house classifier:	40%
○ Out-of-box classifier:	20%
○ Data visualization charts:	20%
○ ROC curve charts:	20%
* Extra: multi class championship	30%



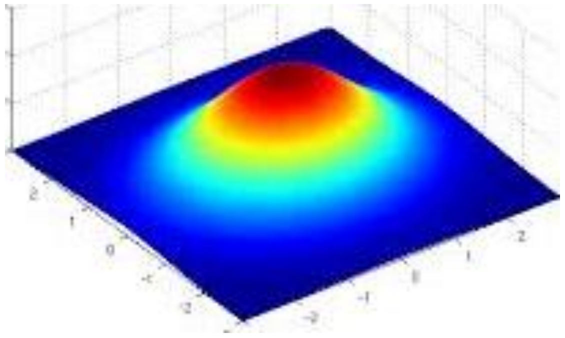
Gaussian Classification Model

Description

- Get train/test data sample from open sources, e.g. GBP/USD index. Use daily change (↑, ↓) as a target.
- Visualize the distribution of sample data
- Estimate accuracy of data classification
- Build ROC curves and calculate AUC

Assessment criteria

○	In-house classifier:	40%
○	Out-of-box classifier:	20%
○	Data visualization charts:	20%
○	ROC curve charts:	20%
*	Extra: multi class championship	30%



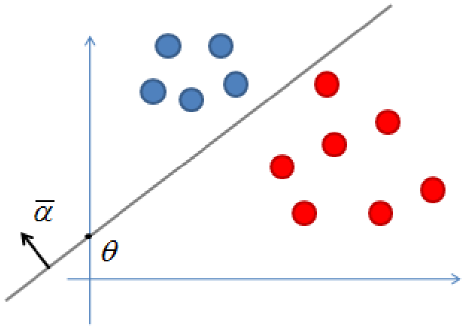
Linear Discrimination with SVM

Description

- Get train/test data sample from open sources, e.g. GBP/USD index. Use daily change (↑, ↓) as a target.
- Visualize the distribution of sample data
- Estimate accuracy of data classification
- Build ROC curves and calculate AUC

Assessment criteria

○ In-house classifier:	40%
○ Out-of-box classifier:	20%
○ Data visualization charts:	20%
○ ROC curve charts:	20%
* Extra: multi class championship	30%



Decision strategies

Description

- Get bet data from open sources
- Visualize the distribution of sample data
- Estimate accuracy of random, deterministic (host, visitor, challenger) and Bayesian strategies
- Build ROC curves and calculate AUC
- Engineer the predictive features and estimate their predictive power

Assessment criteria

○	Deterministic strategy:	20%
○	Random Strategy:	20%
○	Bayesian Strategy:	40%
○	ROC curve charts:	20%
*	Show predictive features:	30%

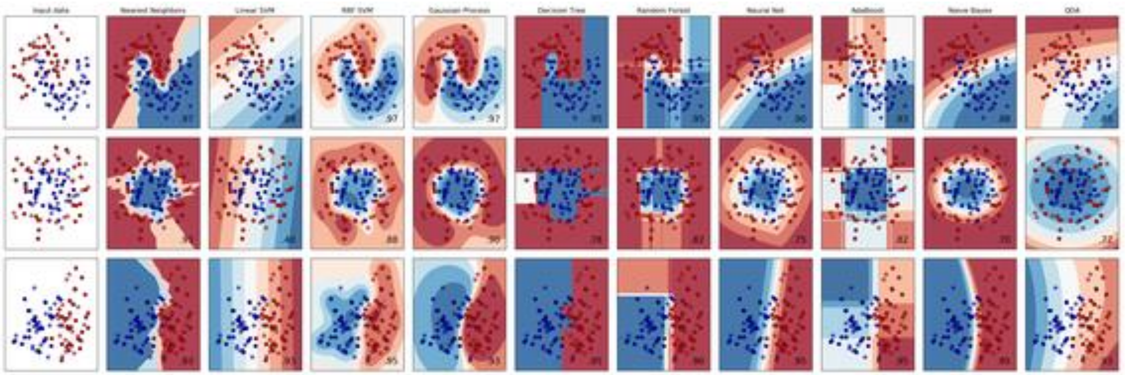
Benchmarking the Supervised Learning Algorithms

Description

- Get data from open sources
- Visualize the distribution of sample data
- Estimate accuracy of different classifiers
- Build ROC curves and calculate AUC
- Engineer the predictive features and estimate their predictive power

Assessment criteria

○	Linear discriminant:	20%
○	Gaussian model:	20%
○	Other models:	20%
○	Data visualization charts:	20%
○	ROC curve charts:	20%
*	Extra: multi class championship	30%



Unsupervised learning

Description

- Get train/test data sample from open sources or generate it by yourself. Focus on low-dimension data, preferably 2 dimensions, 3-5 classes
- Visualize the distribution of sample data
- Visualize iteration of EM/K-Means algorithms
- Estimate the accuracy of classification

Assessment criteria

○ In-house unsupervised classifier:	40%
○ Out-of-box classifier:	20%
○ Data visualization charts:	20%
○ Visualize the learning iterations:	20%
* Benchmark results with some clustering algorithms	30%

Deep Learning

- Feature extraction
- Transfer learning
- Fine-tuning
- Object Detection
- Texture Segmentation
- Stereo Vision
- OCR
- Feature Visualization

Feature extraction

Description

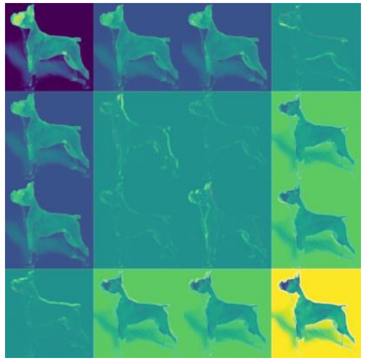
- Get train/test data sample from open sources. Try both scenarios when classless are well separable and are not.

- Run data thru various feature extractors and visualize the results, estimate the quality of clustering

- Visualize the intermediate data of CNN: filters, layers

Assessment criteria

○	Build own extractor :	40%
○	Use existing feature extractors:	20%
○	PCA and TSNE charts:	20%
○	Benchmark quality of extractors:	20%
*	Illustrate de-convolutions:	30%



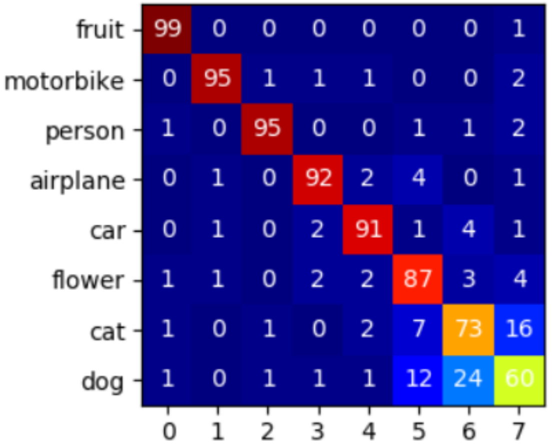
Transfer learning

Description

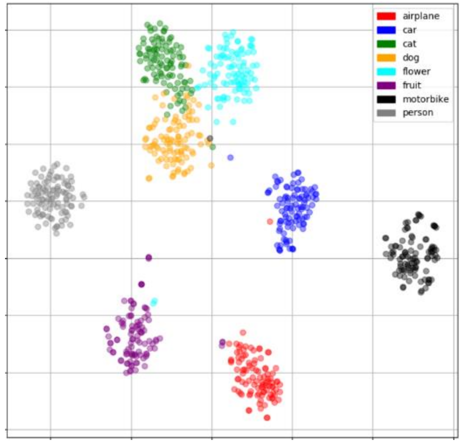
- Benchmark accuracy of multiple classifiers (SVM, RF, LM, etc) applied on top of several feature extractors (AlexNet, VGG, ImageNet etc)
- Research how accuracy very depending on the depth of the frozen layers, visualize this dependency
- Build/visualize confusion matrices
- Illustrate most typical classification errors

Assessment criteria

Build confusion matrices:	60%
Build the accuracy trend:	20%
Illustrate typical errors:	20%



	0	1	2	3	4	5	6	7
fruit	99	0	0	0	0	0	0	1
motorbike	0	95	1	1	1	0	0	2
person	1	0	95	0	0	1	1	2
airplane	0	1	0	92	2	4	0	1
car	0	1	0	2	91	1	4	1
flower	1	1	0	2	2	87	3	4
cat	1	0	1	0	2	7	73	16
dog	1	0	1	1	1	12	24	60



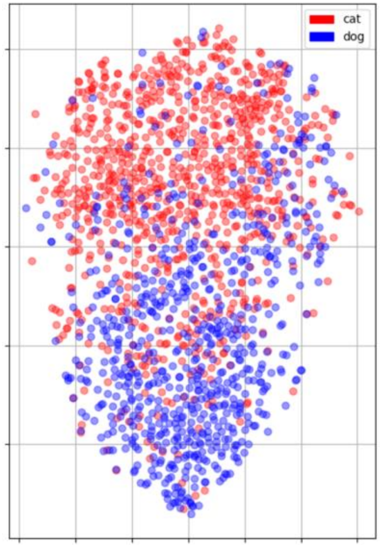
Fine-tuning

Description

- Benchmark accuracy of multiple CNNs finetuned on specific classes of images (e.g faces, cars)
- Illustrate how internal parameters were optimized to maximize the classification accuracy
- Build/visualize confusion matrices
- Illustrate most typical classification errors

Assessment criteria

○ Build confusion matrices:	60%
○ Illustrate internal CNN data:	20%
○ Illustrate typical errors:	20%



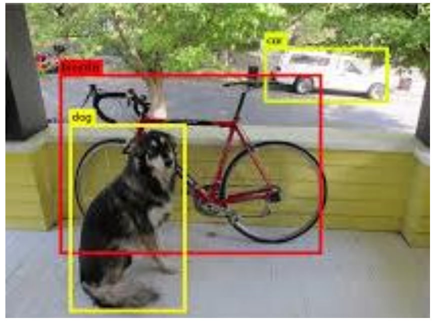
Object detection

Description

- Get train/test data sample from open sources or generate it yourself. Try on human faces, human shapes, cars, etc. Any type of object that you can capture with your laptop camera is preferable.
- Train the detector and estimate the accuracy against the test set
- Run detector in live mode, detect objects in video stream captured by a laptop camera

Assessment criteria

○ Build own detector:	40%
○ Use pre-trained detector:	20%
○ Precision-recall chart:	20%
○ Detecting objects in live mode:	20%
* Benchmark results with some alternative detectors	30%



Texture Segmentation

Description

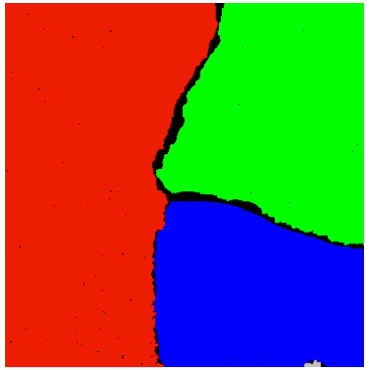
- Get train/test data sample from open sources or generate it yourself. Try on human faces, human shapes, cars, etc. Any type of object that you can capture with your laptop camera is preferable.

- Train the detector and estimate the accuracy against the test set

- Run detector in live mode, detect objects in video stream captured by a laptop camera

Assessment criteria

○ Build own detector:	40%
○ Use pre-trained detector:	20%
○ ROC chart for 2 classes problem:	20%
○ Processing images in live mode:	20%
* Benchmark results with some alternative detectors	30%

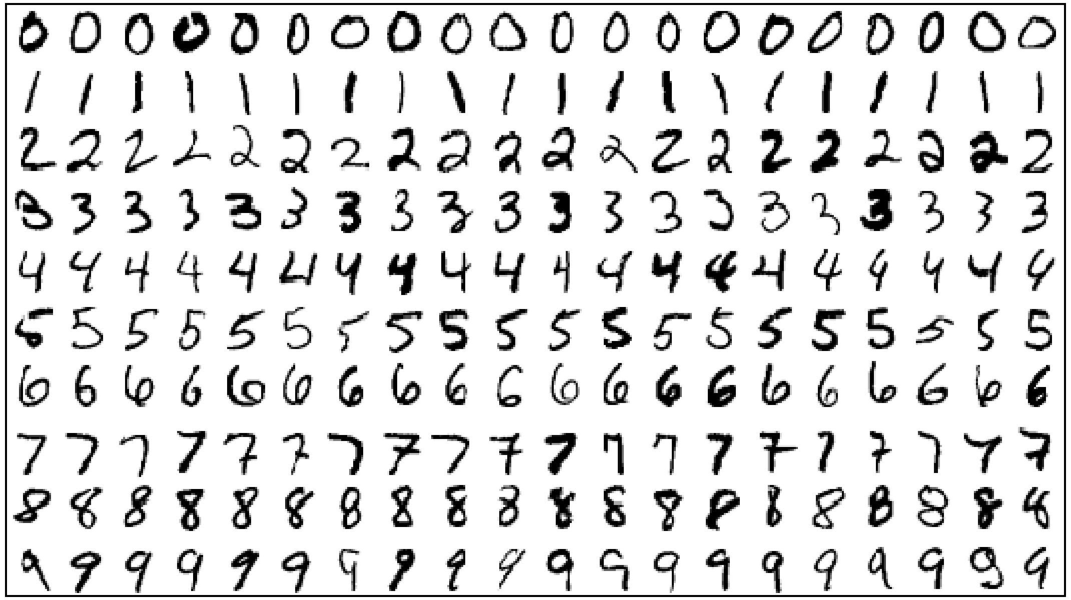


Stereo Vision

Assessment criteria

○ BM approach:	25%
○ SGBM:	25%
○ SIFT approach:	25%
○ Template Matching approach:	25%

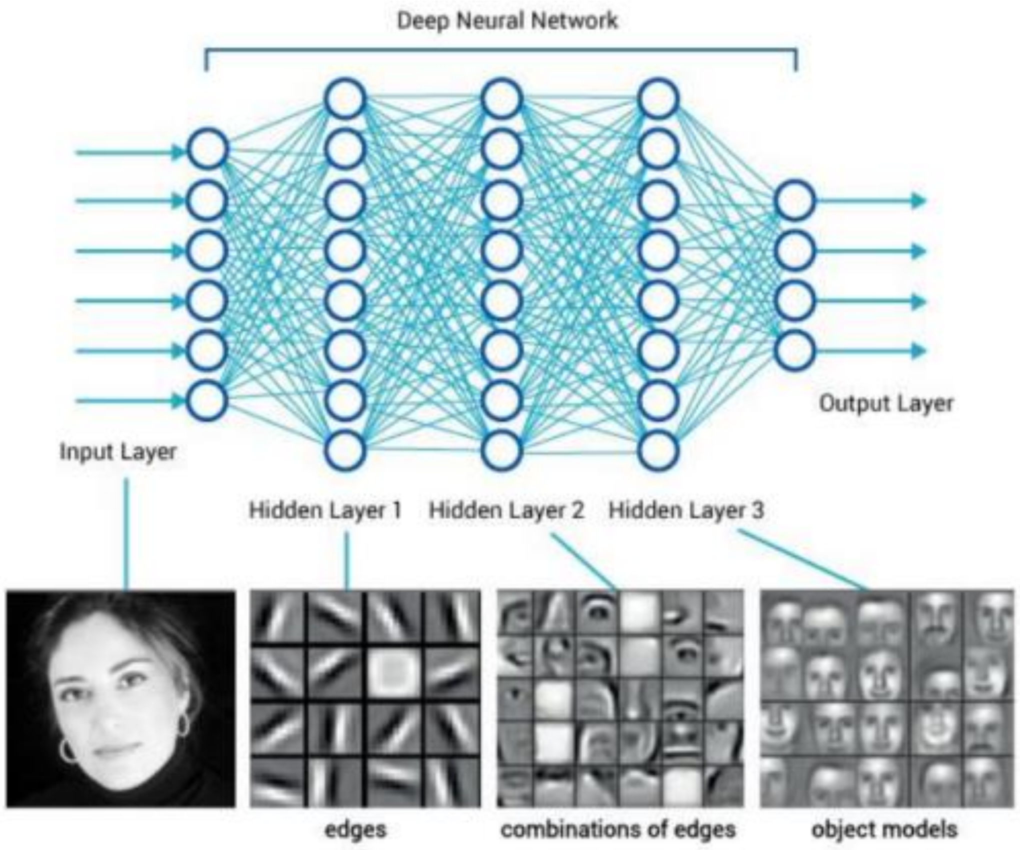
OCR



Assessment criteria

Build confusion matrices:	60%
Build the accuracy trend:	20%
Illustrate typical errors:	20%

Feature Visualization



Assessment criteria

○ Visualize convolution filters:	50%
○ Visualize layers:	50%

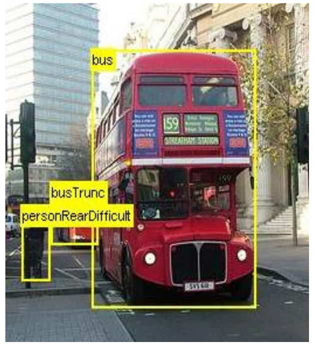
* Overview existing frameworks for feature visualization	30%

Tech Talks

- Object detection
- Style Transfer
- GAN
- 3D object reconstruction
- Image to text
- Segmentation
- Colorization
- Denoising
- DeepFake
- Pose Estimation
- NLP

Tech Talks

Object detection



Style transfer

GAN

3D object reconstruction

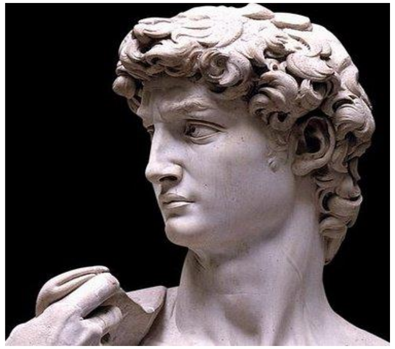
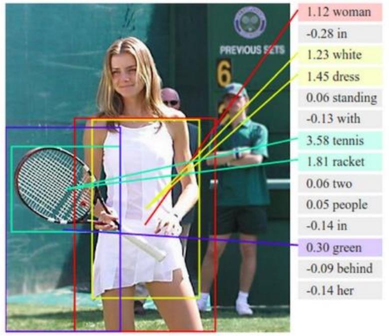
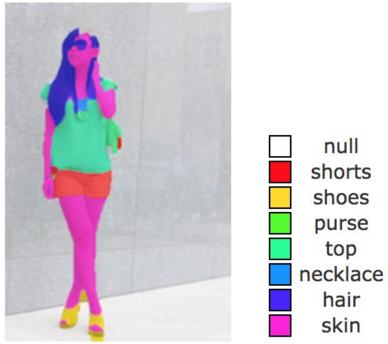


Image to text



Segmentation

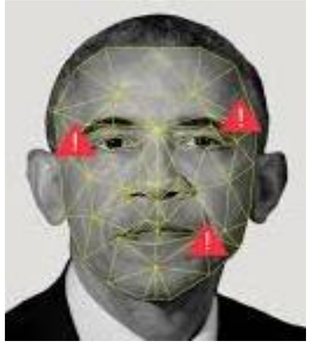


Tech Talks

Colorization

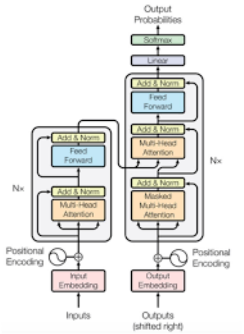
Denoising

DeepFake



Pose estimation

NLP



More...