Evolutionary Computation and Evolutionary Deep Learning for Image Analysis, Signal Processing and Pattern Recognition

Mengjie Zhang[1] and Stefano Cagnoni[2]

1 Evolutionary Computation Research Group, Victoria University of Wellington, Wellington, New Zealand

2 IBIS Lab, University of Parma, Parma, Italy

Mengjie.zhang@ecs.vuw.ac.nz, cagnoni@ce.unipr.it

http://gecco-2018.sigevo.org/

GECCO '18 Companion, July 15--19, 2018, Kyoto, Japan

 978-1-4503-5764-7/18/07...$15.00

DOI https://doi.org/10.1145/3205651.3207859

Instructors

- **Mengjie Zhang** is a Professor of Computer Science at the School of Engineering and Computer Science, Victoria University of Wellington (VUW), New Zealand. His research is mainly focused on evolutionary computation, particularly genetic programming, particle swarm optimization and evolutionary deep learning in image analysis, multi-objective optimization, classification with unbalanced data, feature selection and reduction, and job shop scheduling. He has published over 500 academic papers in refereed international journals and conferences. He has been serving as an associated editor or editorial board member for five international journals (including IEEE Transactions on Evolutionary Computation and the Evolutionary Computation Journal) and as a reviewer of over fifteen international journals. He has been serving as a steering committee member and a program committee member for over eighty international conferences.

- Stefano Cagnoni is an Associate Professor at the University of Parma. Recent research grants include co-management of a project funded by Italian Railway Network Society (RFI) aimed at developing an automatic inspection system for train pantographs, and a "Marie Curie Initial Training Network" grant, for a four-year research training project in Medical Imaging using Bio-Inspired and Soft Computing. Editor-in-chief of the "Journal of Artificial Evolution and Applications" from 2007 to 2010. Since 1999, he has been chair of EvoIASP, an event dedicated to evolutionary computation for image analysis and signal processing, now a track of the EvoApplications conference. Since 2005, he has co-chaired MedGEC, workshop on medical applications of evolutionary computation at GECCO. Co-editor of special issues of journals dedicated to Evolutionary Computation for Image Analysis and Signal Processing. Member of the Editorial Board of the journals "Evolutionary Computation" and "Genetic Programming and Evolvable Machines".

Outline

- Computer vision and image analysis
- ECV methods
- ECV applications

- Evolutionary deep learning

- Major events
- References
- Acknowledgement

Computer Vision

- The "art" of making computers see (and understand what they see)
- Computer vision vs image processing
- Sub-topics:
  - Image acquisition
  - Image enhancement
  - Image segmentation
  - 3D-information recovery/feature extraction
  - Image understanding

Computer and Human Vision

HUMAN	COMPUTER
Perception	Image acquisition
Selective information extraction	Feature enhancement (signal/image processing)
Grouping by 'similarity'	Segmentation
Extraction of spatial relationships	3D-information Recovery
Object recognition and semantic interpretation	Image Understanding

Computer and Human Vision

HUMAN	COMPUTER
Perception	Image acquisition
Selective information extraction	Feature enhancement (signal/image processing)
Grouping by 'similarity'	Segmentation
Extraction of spatial relationships	3D-information Recovery
Object recognition and semantic interpretation	Image Understanding

LOW-LEVEL VISION

Computer and Human Vision

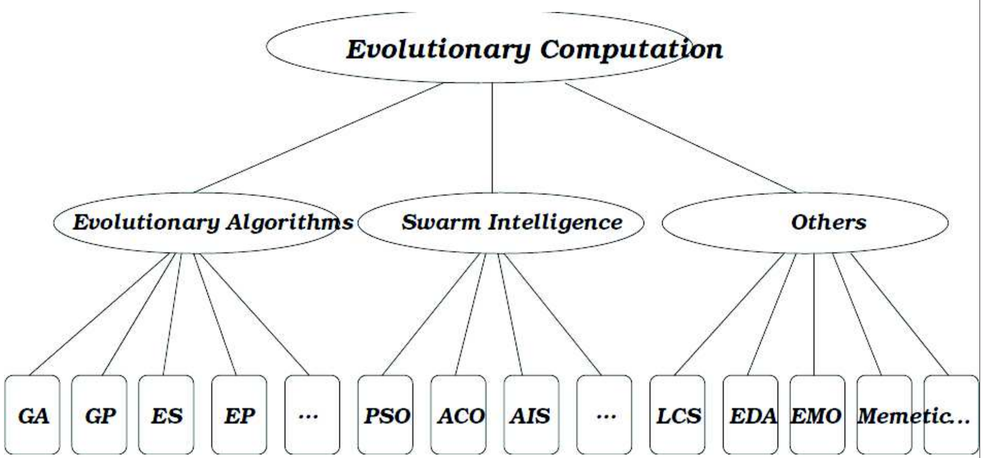
HUMAN	COMPUTER
Perception	Image acquisition
Selective information extraction	Feature enhancement (signal/image processing)
Grouping by 'similarity'	Segmentation
Extraction of spatial relationships	3D-information Recovery
Object recognition and semantic interpretation	Image Understanding

HIGH-LEVEL VISION

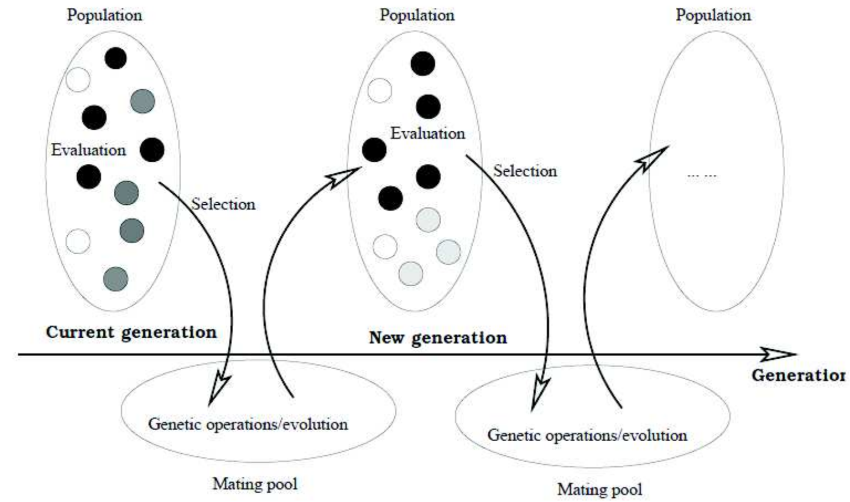
Computational Intelligence (CI)

- Symbolic intelligence vs CI
- Neural Networks
- Evolutionary Computation
  - Evolutionary Algorithms
  - Swarm Intelligence
  - Others
- Fuzzy Systems
- Other

Evolutionary Computation



Evolutionary Computation Process



Application Taxonomy

- EC techniques
  - GA, GP, ES, EP, PSO, DE, LCS, EMO, EDA, etc.
- Solution types
  - Optimisation of parameters of specific solutions (using GA, ES, PSO…)
    - Related with a well-defined task or for a whole system
  - Generation of solutions from scratch (GP, …)
    - Performance optimization based on specific objective functions
    - It is difficult to choose a model with reasonable assumptions
- Role of EC techniques
  - Interactive qualitative comparisons between solutions
  - Generation of emergent collective solutions
    - Achievement of higher-level and complex tasks from collective use of trivial, local, hard-wired behaviours: generation of full EC-based solutions, NOT parameter optimization tasks

Applications (Stefano Cagnoni)

- Optimization of filter/detector AND algorithm parameters for event detection and image segmentation
- Design of implicitly parallel binary image operators and classifiers
- Qualitative optimization of image processing algorithms
- Object detection, segmentation, tracking

Applications

- EC techniques: GP, PSO, LCS, EMO
- Image Analysis
  - Object tracking
  - Edge detection
  - Segmentation
  - Motion detection
  - Object/digit recognition
- English stress detection(signal processing)
- Pattern Recognition: feature selection and biomarker detection

GP for ECV Applications

Genetic Programming -- Origin

- Genetic algorithms (GAs) with tree-like representation
- Automatic programming: one of the major challenges of computer science --- use a computer to do what needs to be done without telling/knowing the specific steps.
- GP = Automatic programming + GAs
- GP genetically breeding a population of computer programs using principles of Darwinian natural selection and biologically inspired operations

GP: Representations

- Tree based GP: John Koza
  - Lisp programs
  - Koza:92 vs 1980s: Cramer
  - Most commonly used
- Linear GP: Wolfgang Banzhaf
  - C/C++/Java programs
  - Graph: like NNs but not fully connected and more flexible
- Grammar based GP/Grammatical Evolution: Peter Whigham, Bob McKay, Michael O'Neill
- Cartesian GP: Julian Miller

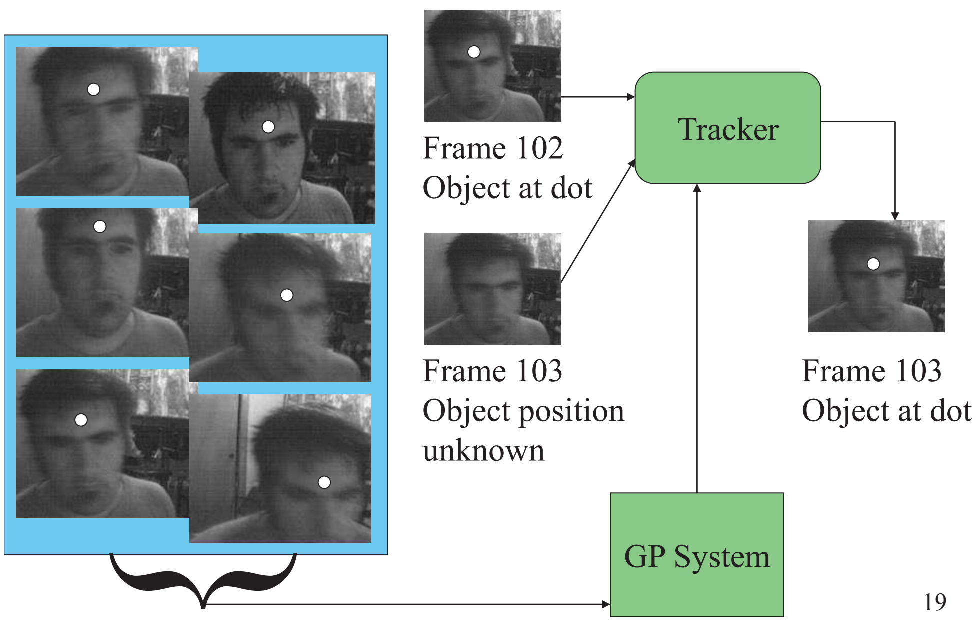
GP for Vision Tasks

- Object detection
- Object classification
- Object tracking
- Motion detection
- Edge detection
- Segmentation
- Many domains: medical, military, agriculture, biology, transportation, …

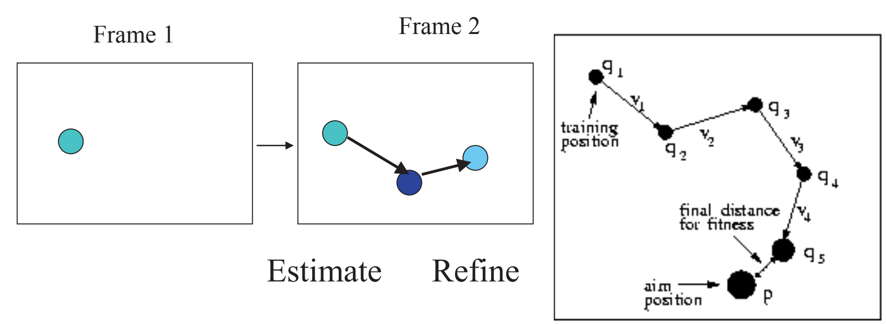
GP for Object Tracking

- Use GP to track an object in low-quality webcam footage, at a real-time speed.
- Test the GP method on two object tracking problems of varying difficulty.

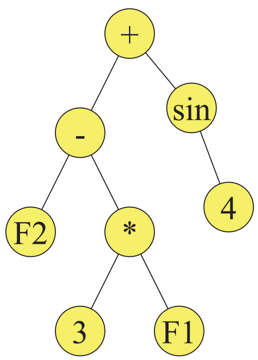
Object Tracking Task



Tracker Programs



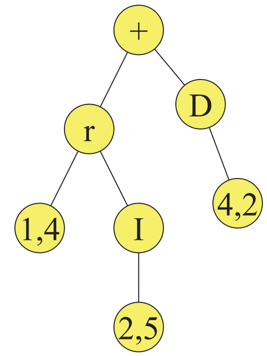
Standard Evolved Programs



(F2 – 3 * F1) + sin(4)

- Evolved programs (or genetic programs)
  - Tree-like expression structure
  - Internal nodes are functions
  - Leaf nodes (terminals) are constants or input (feature) values.
  - Evaluating program produces a single value.

GP Tracker Program

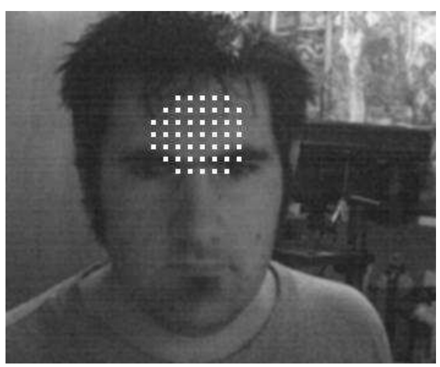


((1,4) rot. by I(2,5)) + D(4,2)

- Non-standard
- Nodes deal with two-valued vectors
  - Not single values
  - Program output is a two-valued vector
- Input features are functions
  - Not terminals
  - Not just elements in a fixed length feature-vector
- Still uses a tree

Training

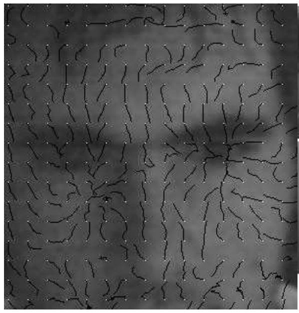
- Specify target object position
- Evaluate tracker program at a set of training points around target producing refined estimates.
- Fitness of program = avg. distance from target



Data Sets

- We used two short pieces of webcam footage of a person moving around at a fixed distance from the camera.
  - 358 x 288, 15 fps, 256 shade, greyscale
  - Very low quality.
  - Fast movement looks very blurry.
  - Include some tricky movements like moving close to the border, looking up, moving quickly and obscuring face.
- Two tasks
  - Left eye
  - Centre of forehead

Experiment 1:Tracking the left eye

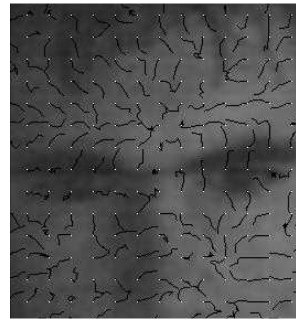


Trails of tracker convergence

Tracking, every 20^{th} frame

- Tracks well, even when the face was quite blurry due to fast movement.

Experiment 2:Tracking the head



Trails of tracker convergence

Tracking, every 20^{th} frame

- Tracks well, even when the face was quite blurry due to fast movement and when the head looks up.

Example Videos

Summary

- GP programs very successfully tracked the objects in the video-sequences.
- No domain knowledge was necessary
  - programs automatically constructed
  - Just 15 images with object positions located
- Non-standard GP program structure was critical.
  - Vector outputs
  - Feature functions
- Evolution identifies a small number of point features to compute while tracking
  - Efficient.
- Tracks about nine times faster than real-time
  - This is with non-compiled evolved programs
  - Compiled would be faster.

GP for Image Segmentation

- *A figure-ground segmentation method* is developed using GP to evolve segmentors from the local image information.
- Based on this proposed method, a wide range of features have been investigated as *terminal sets*.

Construction of GP-based Method using Local Information

Generate a classifier

↓

Use the classifier to sweep the test image, assign a label to sub-images

↓

Use a voting scheme to finalize each pixel's label, generate the output.

Function Set

Function Set

Function Name	Definition	Type
Add(a_1, a_2)	$a_1 + a_2$	Arithmetic
Sub(a_1, a_2)	$a_1 - a_2$	Arithmetic
Mul(a_1, a_2)	$a_1 * a_2$	Arithmetic
Div(a_1, a_2)	$\begin{cases} a_1/a_2 & \text{if } a_2 != 0 \\ 0 & \text{if } a_2 == 0 \end{cases}$	Arithmetic
IF(a_1, a_2, a_3)	$\begin{cases} a_2 & \text{if } a_1 \text{ is true.} \\ a_3 & \text{if } a_1 \text{ is false.} \end{cases}$	Relation
$<= (a_1, a_2)$	$\begin{cases} true & \text{if } a_1 <= a_2 \\ false & \text{if otherwise} \end{cases}$	Relation
$>= (a_1, a_2)$	$\begin{cases} true & \text{if } a_1 >= a_2 \\ false & \text{if otherwise} \end{cases}$	Relation
$== (a_1, a_2)$	$\begin{cases} true & \text{if } a_1 == a_2 \\ false & \text{if otherwise} \end{cases}$	Relation
Between(a_1, a_2, a_3)	$\begin{cases} true & \text{if } a_2 <= a_1 <= a_3 \\ false & \text{if otherwise} \end{cases}$	Relation

Terminal Sets

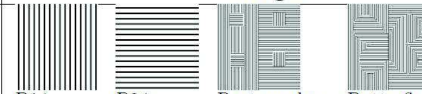
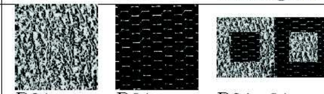
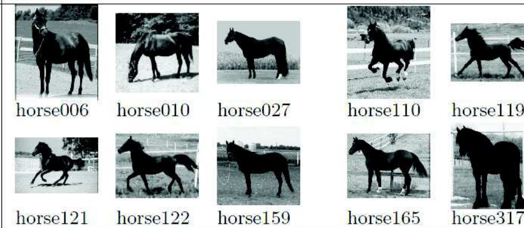
Terminal Set

	Features	Category
Terminal Set 1	Raw Pixel Values	Brightness
Terminal Set 2	Histogram Statistics	Texture
Terminal Set 3	GLCM Statistics (Grey-Level Co-occurrence Matrix)	
Terminal Set 4	LBP (Local Binary Patterns)	
Terminal Set 5	Fourier Power Spectrum	
Terminal Set 6	Gabor Features	
Terminal Set 7	Moments + Gradient Statistics	Shape

Fitness Function and Parameters

$$f = \frac{Number\ of\ correctly\ classified\ samples}{Number\ of\ total\ training\ samples}$$

Population Size	**500**	**Generation Number**	**51**
Crossover Rate	**0.9**	**Mutation Rate**	**0.1**
Max tree depth for initialization	**6**	**Max tree depth for evolution process**	**17**

Database	Images	Descriptions
Bitmap	 P14 P24 Rectangular Butterfly	Size: 256*256 Synthetic, binary images
Brodatz	 D24 D34 D24vs34	Size: 320*160 Grayscale images
Weizmann	 horse006 horse010 horse027 horse110 horse119 horse121 horse122 horse159 horse165 horse317	Average Size:248*211 Real images Varing horse positions One object
PASCAL (Name prefix: 2007_00)	 0033 0256 0738 1288 1761 2099 2266 2376	Average Size:500*350 Real images Varing object locations/sizes Multiple objects

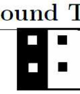
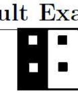
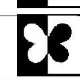
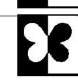
Result Evaluation Measures

Evaluation Measures

- Segmentation Accuracy — *Simple; commonly-used.* *The higher the better.* **Problem: insufficient (e.g. a small object in a test image is segmented totally as background, the accuracy can still be high)**
- F1 Measure — *Combine precision and recall together. Best at 1, worst at 0.*
- Negative Rate Metric (NRM) — *Consider mismatches between a prediction and ground truth. Best at 0, worst at 1.*

Results

Results on Bitmap Images

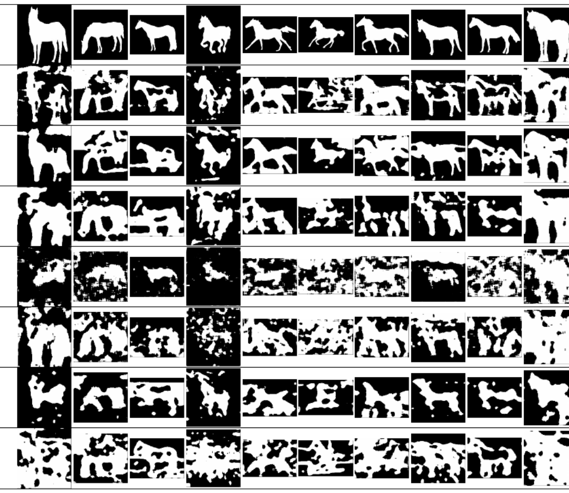
	Ground Truth	Result Examples	Accuracy(%)	F_1	NMR
Intensity			98.81 ± 0.22	0.99	0.20
Intensity			95.96 ± 1.27	0.96	0.12

Results on Texture Images

Feature	Result Examples	Accuracy(%)	F_1	NMR
Ground Truth		-	-	-
Intensity		**94.26$\pm$ 2.75**	**0.94**	0.15
Histogram Statistics		93.98 $\pm$ 2.30	**0.94**	**0.07**
GLCM Statistics		92.67 $\pm$ 1.45	0.92	0.15
LBP		66.82 $\pm$ 10.06	0.53	0.35
Fourier Power Spectrum		91.16 $\pm$ 0.94	0.90	0.13
Gabor		90.91 $\pm$ 0.72	0.90	0.15
Moments + Gradient		92.02 $\pm$ 2.11	0.92	0.39

Results

Results on Weizmann Images



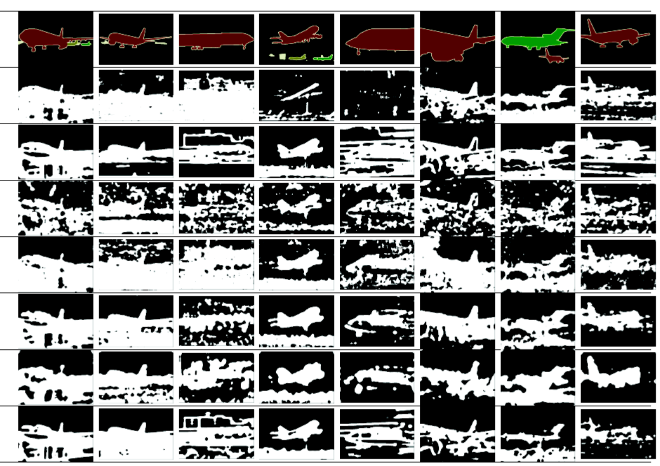
Results

Statistical Results on Weizmann Images

Feature	Accuracy (%)	F_1	NRM
Intensity	74.41 $\pm$ 8.37	0.62	0.47
Histogram Statistics	77.37 $\pm$ 9.09	**0.84**	0.47
GLCM Statistics	76.74 $\pm$ 3.92	0.68	0.47
LBP	66.19 $\pm$ 10.95	0.52	0.48
Fourier	68.38 $\pm$ 7.38	0.61	0.50
Gabor	**78.29$\pm$5.40**	0.66	**0.42**
Moments + Gradient statistics	65.04 $\pm$ 10.39	0.58	0.50

Results

Results on PASCAL Images



Results

Statistical Results on PASCAL Images

Feature	Accuracy (%)	F_1	NRM
Intensity	71.39 ± 10.63	0.49	0.50
Histogram Statistics	74.56 ± 6.89	0.61	0.50
GLCM Statistics	67.39 ± 9.60	0.49	0.52
LBP	63.75 ± 14.07	0.54	0.50
Fourier	75.10 ± 7.90	0.61	**0.46**
Gabor	**75.60±8.10**	**0.62**	**0.46**
Moments + Gradient statistics	74.53 ± 7.83	0.59	0.48

Summary

- When segmenting complex images, *higher-level information* (e.g. spectral or statistical information) is necessary.
- The GP-based method using local image information can achieve accurate segmentation *across a wide range of images*.
- Results on images from Weizmann and PASCAL datasets are obviously worse than those on binary or texture images. Need better features
- This local information based method often produces inaccurate boundaries. Need global information

GP for Motion Detection: without noise

Motion Detection: with noise

Test Detector 1 in raining day

Test Detector 2

Detector 2 can perform well on videos with additive noise of variance **50**.

Test Detector 2 in raining day

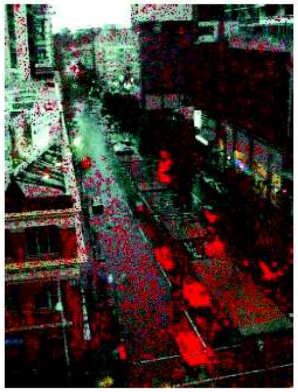
Moving camera

Comparison: Background Modelling -1

All pixels in motion are marked in red, including pedestrians and minor false positives.

Note: the camera position was fixed in this experiment.

Comparison: Background Modelling -2



Left: model trained on no-noise data performing on raining day. (No camera movement)

Right: model trained on noise data performing on raining day. (No camera movement)

Comparison: Background Modelling-3

Applying background model on unseen data.

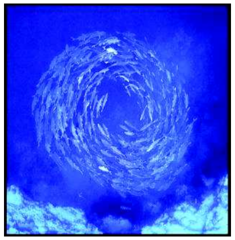
Left: unseen raining condition.

Right: changed camera angel (not even a moving camera).

PSO for Edge Detection

Particle Swarm Optimisation

- PSO as a global optimisation method was proposed by Kennedy and Eberhart in 1995
- It is a simulation of a simplified social model like bird flocking and fish schooling



Key Concepts in PSO

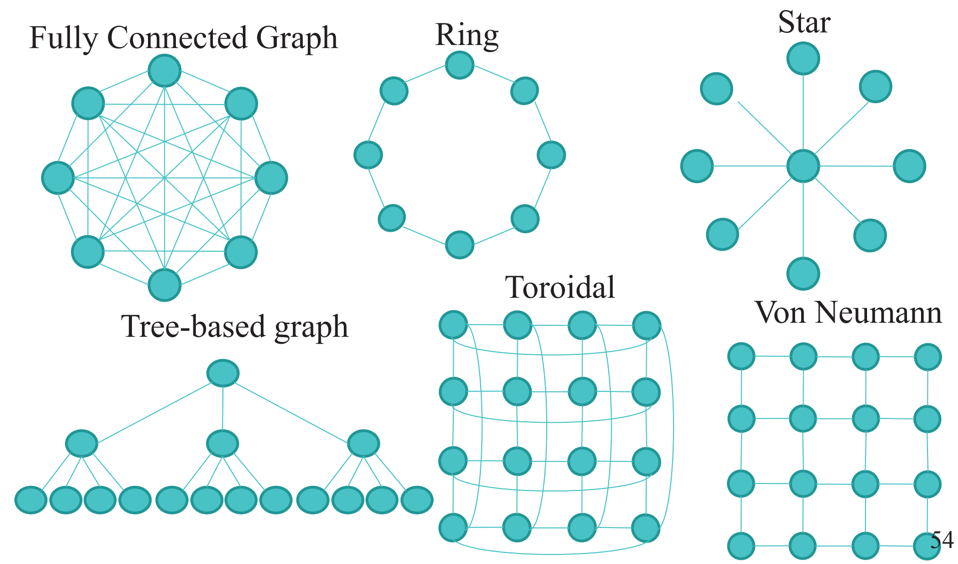
- **Particle:** there is a population containing *m* potential solutions (called particles)
- **Velocity and Position Equations:** the particles move through *n*-dimensional search space according to position and velocity update equations

$$\vec{X}_i(t+1) = \vec{X}_i(t) + \vec{V}_i(t+1)$$

$$\vec{V}_i(t+1) = w\vec{V}_i(t) + C_1 Rand_1(\vec{X}_{pbest_i} - \vec{X}_i(t)) + C_2 Rand_2(\vec{X}_{leader} - \vec{X_i}(t))$$

- **Topology:** defines how particles are connected to each other as an information sharing or exchanging mechanism

Some of Well-Known Static Topologies



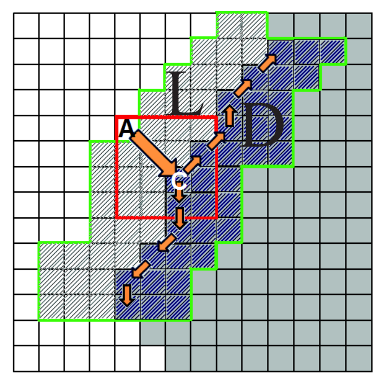
Convolution of Red Rectangle on an Image

Original Image

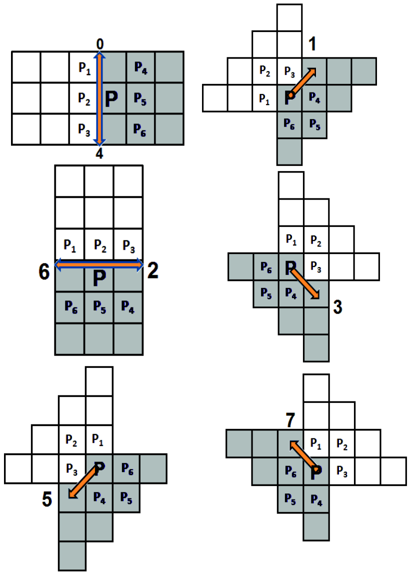
Edge Map

A New Fitness Function

- Maximise distances among an intensity of pixels belonging to two regions separated by a continuous edge and minimise distances within the regions.



Probability Score of Curve C



$$EdgeMag_m(P) = \frac{InterDis_m(P)}{1 + IntraDis_m(P)}$$

$$InterDis_m(P) = min\left(1, |avg_{m_d(P)} - avg_{m_l(P)}|/w_1\right)$$

$$IntraDis_m(P) = \frac{1}{\binom{n}{2}}\left(\sum_{\substack{\forall P_i, P_j \in D \\ i > j}} min\left(1, |I_{P_i} - I_{P_j}|/w_2\right) + \sum_{\substack{\forall P_i, P_j \in L \\ i > j}} min\left(1, |I_{P_i} - I_{P_j}|/w_2\right)\right)$$

$$PScore_m(P) = \frac{1}{1 + e^{-\frac{1}{TH}(EdgeMag_m(P) - TH)}} \times \frac{1}{1 + e^{-2(NMS_d(P) - 4)}}$$

$$NMS_m(P) = |\{P_i | i \in 1..6, EdgeMag_m(P_i) < EdgeMag_m(P)\}|$$

$$U_C = \frac{1}{255 * (max)} \sum_{i=1}^{max} |I_{P_{i+1}} - I_{P_i}|$$

$$PScore(C) = \frac{\sum_{\forall P_i \in C} PScore_{m_i}(P_i)/(max + 1)}{1 + U_C}$$

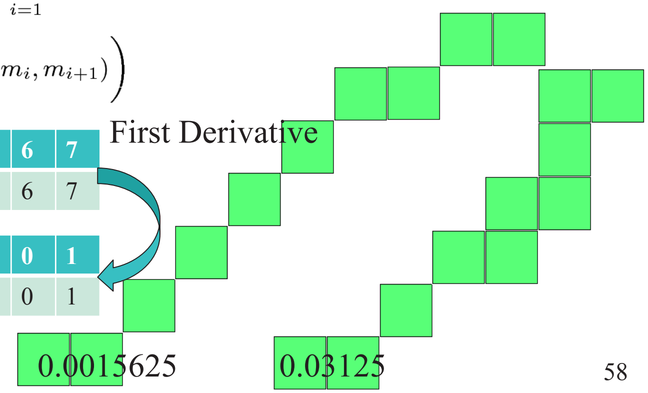
Curvature Cost of Continuous Edges

$$CC(m_i, m_{i+1}) = \begin{cases} |m_i - m_{i+1}|/w_3 & , \quad |m_i - m_{i+1}| \le 4 \\ (8 - |m_i - m_{i+1}|)/w_3 & , \quad otherwise \end{cases}$$

$$CCost(C) = \frac{1}{max - 2}\left(\sum_{i=1}^{max/2-1} CC(m_i, m_{i+1}) + \sum_{i=max/2+1}^{max-1} CC(m_i, m_{i+1})\right)$$

7	6	7	6	6	6	6	6	7
7	5	5	7	5	7	6	6	7

0	1	1	1	0	0	0	0	1
0	2	0	2	2	2	1	0	1



A New Fitness Function With Two Constraints

$$Fitness(C) = PScore(C) - CCost(C)$$

- Subject to two constraints:
  - The curve C never crosses itself.
  - The probability score of the curve C must be larger than the predefined threshold HP.

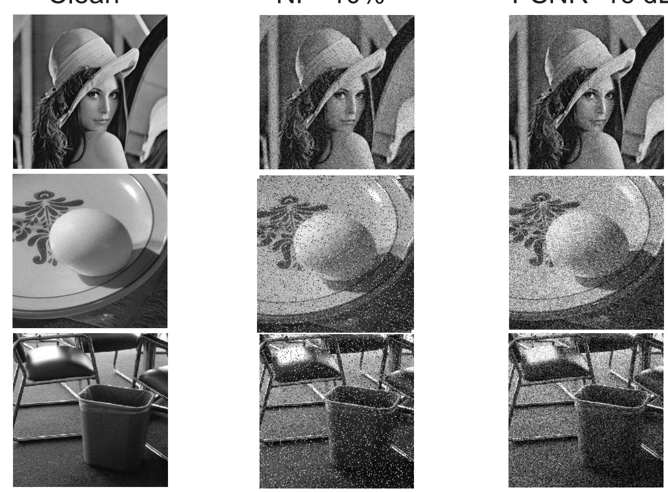
$$Cross(C) = 0 \quad and \quad PScore(C) > HP$$

- A simple preservation method is used to handle these constraints in PSO.

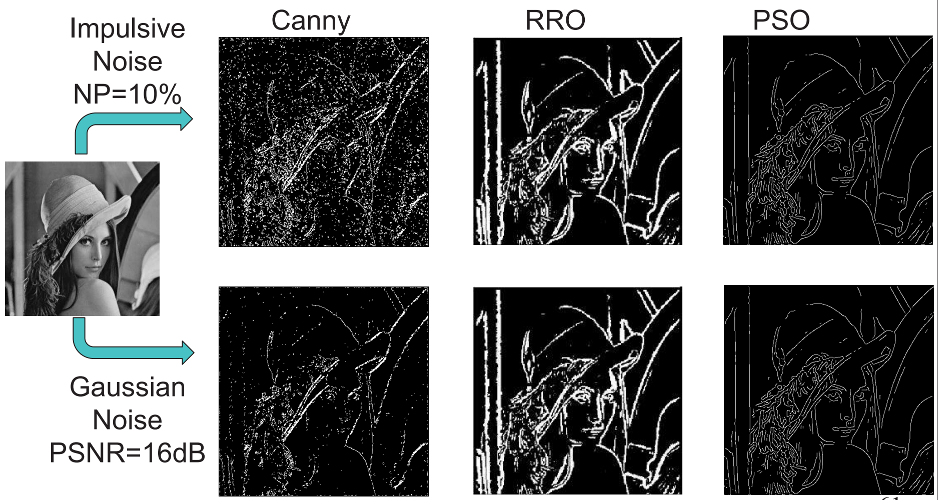
Image Set

(From South Florida University Database)

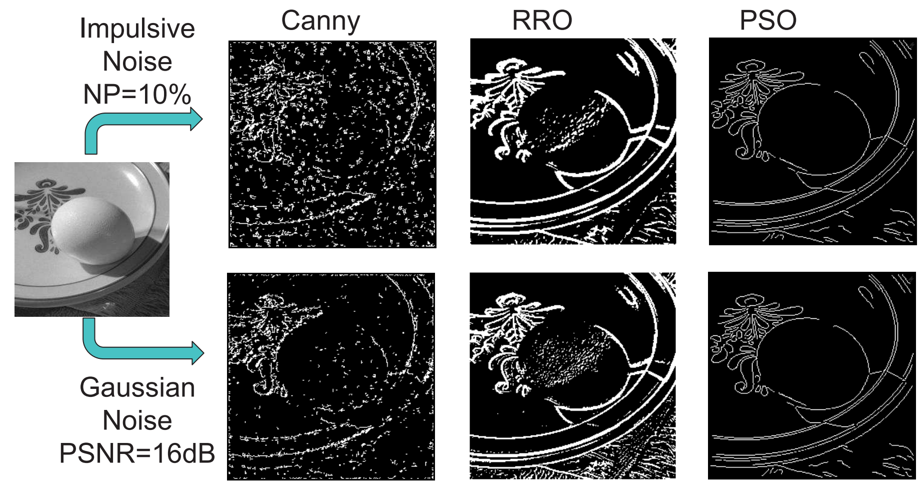
Clean	Impulsive noise NP=10%	Gaussian Noise PSNR=16 dB



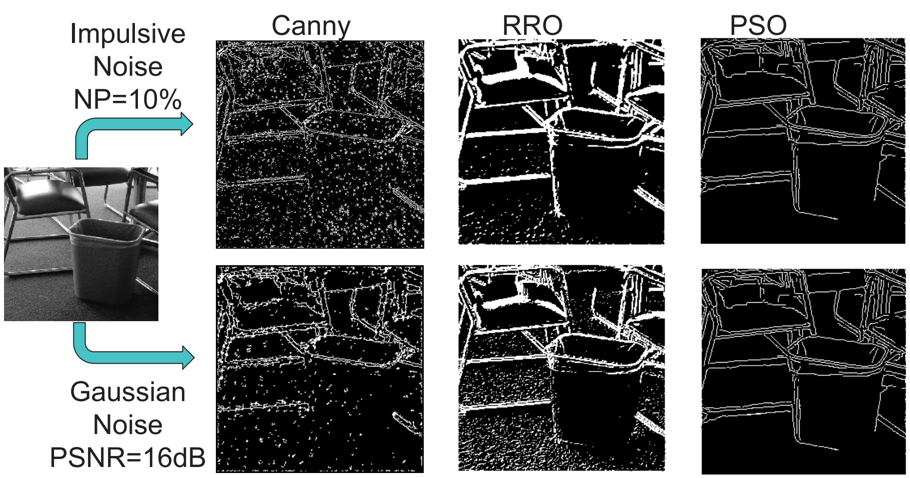
Example 1: PSO vs Canny and RRO



Example 2: PSO vs Canny and RRO



Example 3: PSO vs Canny and RRO

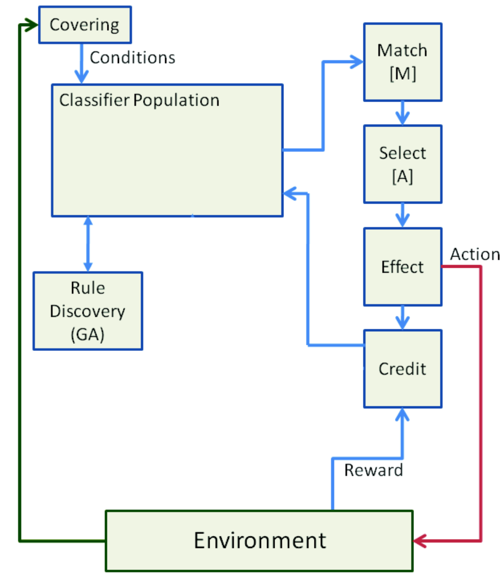


LCS for Hand-written Digit Recognition

Learning Classifier Systems

- Machine learning for Robotics:
  - Needs to be **reinforcement**-based and **online**
  - Preferably also **adaptive** and **transparent**
- Learning from visual input is hard:
  - High-dimensionality vs. sparseness of data
- Why Learning Classifier Systems
  - Robust reinforcement learning
  - Limited applications for visual input

Learning Classifier Systems



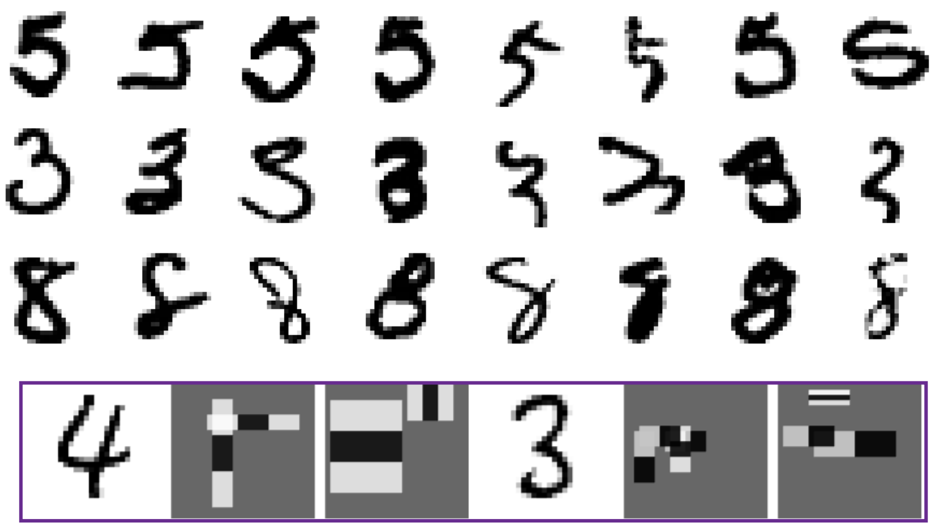
MNIST Digits Dataset

- Well known handwritten digits dataset
- **60 000** training examples, **10** classes
- Examples from **250** subjects
- **28x28** pixel grey-scale (**0..255**) images
- **10 000** evaluation examples (test set, different subjects)

MNIST results

- Performance:
  - Training set: **92%**
  - Test set: **91%**
  - Increase to **96%** (after improvement)
- Supervised and off-line methods reach 99%
- Encouraging initial results for reinforcement learning

Why not 100% performance?



Demo

- Feature Pattern Classifier System (FPCS)
  - Handwritten Digit Classification with LCS

- Toktam Ebadi, Ignas Kukenys, Will N. Browne, Mengjie Zhang: Human-Interpretable Feature Pattern Classification System Using Learning Classifier Systems. Evolutionary Computation 22(4): 629-650 (2014)

GP for English Stress Detection – Signal Processing

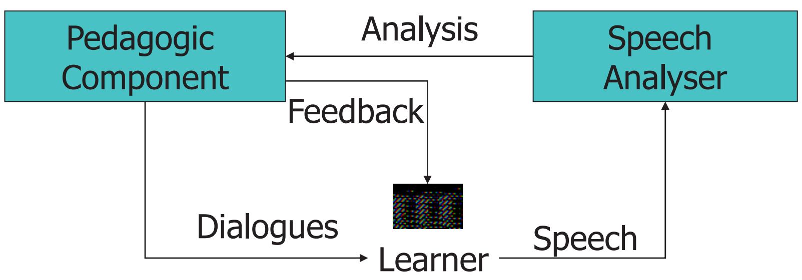
- English becomes more and more important as a communication tool in the world.
- Provide P2P training to ESL students is very expensive. Therefore, software is desirable.
- Correct *rhythmic* stress in ESL students' speech is a key point to make the speech sound like native. Therefore, to accurately detect rhythmic stress in spoken English becomes an important functionality in this kind of software.

Known Stress Classifiers

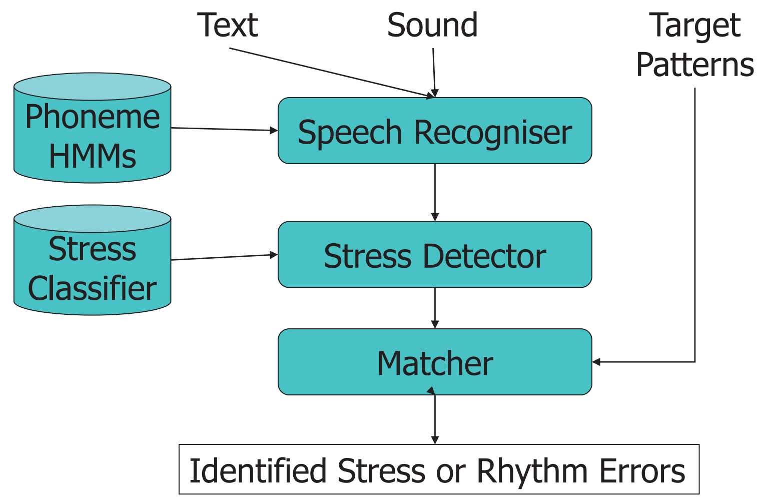
- Bayesian classifier
- Support vector machine classifier
- Decision tree classifier
- Neural networks classifier

The best accuracy is around 85%. It is not high enough for a commercial use.

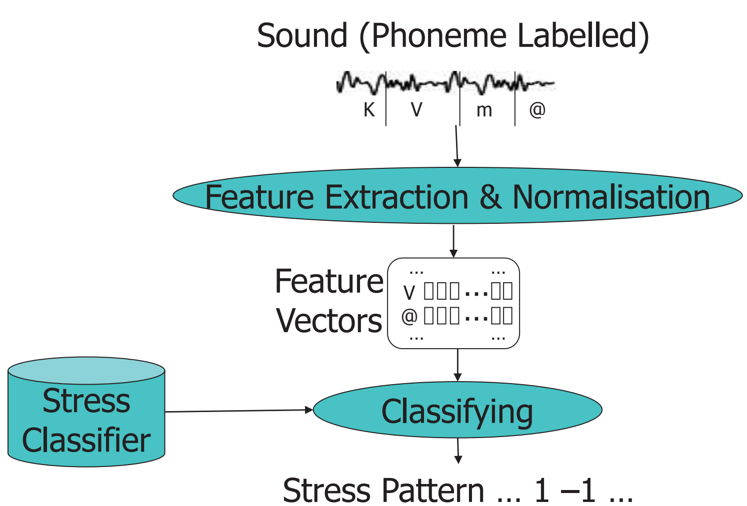
Overview of the whole project



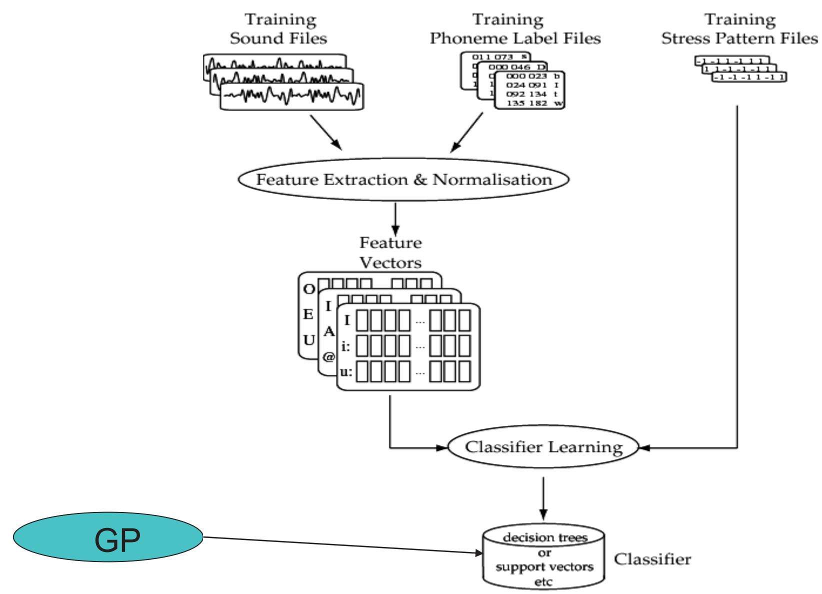
The Speech Analyser



The Stress Detector



Classifier Learning Procedure



GP adapted to stress detection

- Feature extraction & normalisation
- GP configuration
  - Terminal sets and the function set
  - Fitness function
  - Genetic parameters
  - Termination criteria

Feature extraction & normalisation

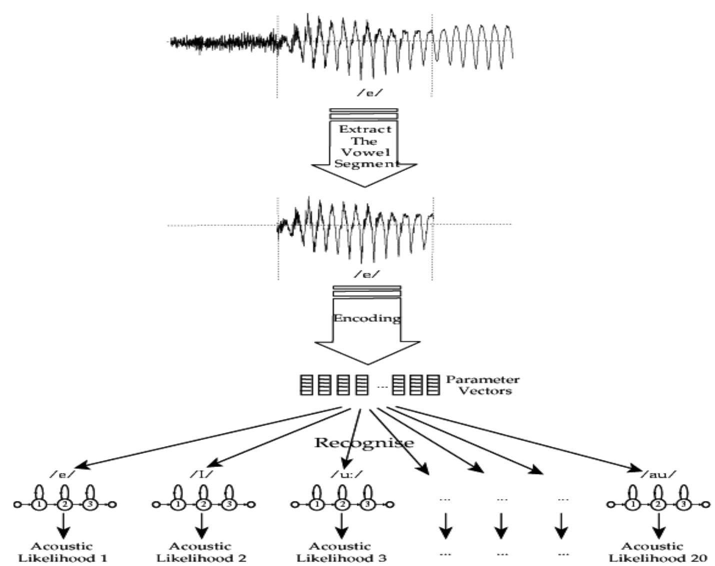
Rhythmic stress is related to:

- Prosodic features – such as duration, amplitude, pitch, and etc.
- Vowel quality – full vowel and reduced vowel. It is defined by the configuration of the tongue, jaw, and lips.

Feature extraction & normalisation – Prosodic (cont.)

- Calculation of prosodic features is well known
  - Duration is how long a syllable lasts
  - Amplitude relates to the loudness of the syllable
  - Pitch is the perceptual correlate of the fundamental frequency of the sound signal
- Need several levels of normalisations to reduce variations of differences between speakers, recording situations or utterance context, etc.
- There are 17 prosodic features used in our study

Feature extraction & normalisation – Vowel Quality (cont.)



Feature extraction & normalisation – Vowel Quality (cont.)

- Find the score of the expected vowel type S_e, the score of the best matching full vowel S_f and the score of the best matching reduced vowel S_r from the above 20 scores.
- Compare S_f and S_r to S_e respectively and measure the difference between the likelihoods and the ratio of the likelihoods.

$$R_d = \begin{cases} -log(S_r - S_e) & \text{if } S_e < S_r \\ 0 & \text{if } S_e = S_r \\ log(S_e - S_r) & \text{if } S_e > S_r \end{cases} \qquad R_r = \log(S_e/S_r) = \log S_e - \log S_r$$

$$F_d = \begin{cases} -\log(S_f - S_e) & \text{if } S_e < S_f \\ 0 & \text{if } S_e = S_f \\ \log(S_e - S_f) & \text{if } S_e > S_f \end{cases} \qquad F_r = \log(S_e/S_f) = \log S_e - \log S_f$$

- Also include a Boolean feature to deal with cases where these 4 features can't be calculated if the vowel segment is so short.

GP configuration

- A linear-structured GP
- Terminal sets
  - I : 17 prosodic features
  - II : 5 vowel quality features
  - III : combination of sets I and II
- The Function Set

  {abs, sqrt, cos, sin, +, -, *, /, iflt, ifpr, ifnr}

GP configuration (cont.)

- Fitness function – error rate.
- Genetic parameters
  - Population size: 1024
  - Tournament size: 4
  - Initial program size: 80
  - Max program size: 256

	I	II	III
Crossover rate	71%	57%	47%
Mutation rate	97%	87%	83%

- The learning process stops when:
  - Max number of generations without improvement reaches 200
  - Fitness of the best program is zero on the training data set

Experiment Design

- Data set: 703 vowels in 60 utterances of ten distinct sentences produced by 6 female speakers – 340 stressed and 363 unstressed
- Scaled feature values in the range [-1,1] are also used.
- Three experiments are conducted on the three terminal sets respectively.
- 10 times 10-fold cross validation for training and testing
- Comparing with
  - DT -- C4.5
  - SVM -- LibSVM (with Radial Basis Function kernel and C = 1)
  - GP: Discipulus

Detection Accuracy (%)

Terminal Set I (prosodic features)			
	GP	DT	SVM
Unscaled	91.9	80.4	79.7
Scaled	91.6	80.6	83.2
Terminal Set II (vowel quality features)			
	GP	DT	SVM
Unscaled	85.4	79.7	79.1
Scaled	84.6	78.9	80.5
Terminal Set III (combination)			
	GP	DT	SVM
Unscaled	92.0	79.9	81.3
Scaled	92.6	80.1	82.0

Summary

- Amongst prosodic features, duration has a bigger impact than amplitude and pitch.
- In vowel quality features, features reflecting reduced vowel quality are far more useful than those reflecting full vowel quality.

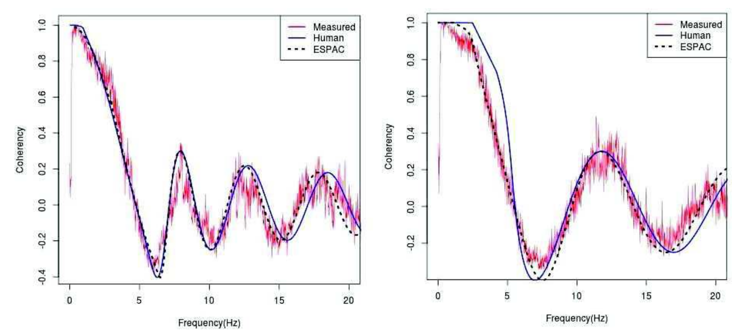
- GP can be used to construct an automatic rhythmic stress detector.
- GP outperforms DT and SVM on this data set
- GP is more robust at handling irrelevant features and has stronger feature selection ability than DT and SVM on our data set

EC for Pattern Recognition

- GP for mathematical modelling
- PSO and EMO for Feature Selection

Mathematical Modelling

Assessing Christchurch Earthquake Liquefaction Potential



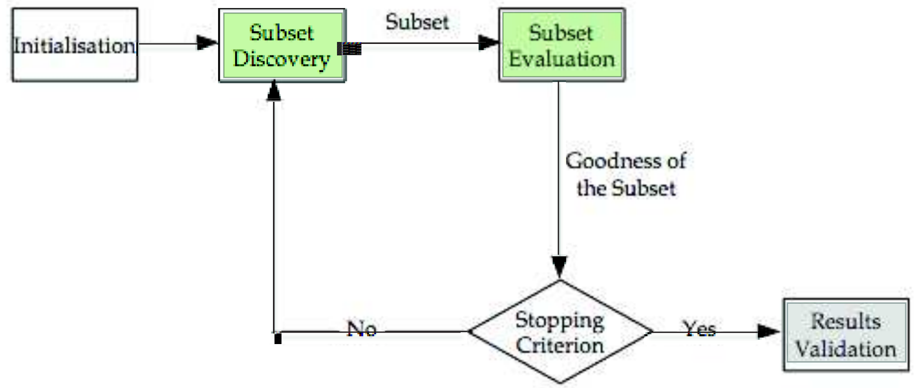
Human competitive results!

Computer Vision –
Satellite and Medical Image Analysis

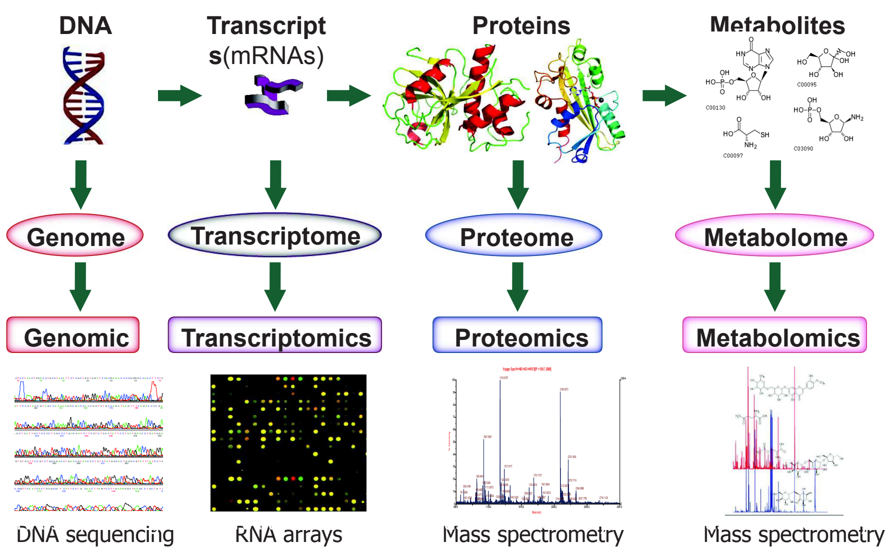
Satellite image –
Land, water, snow, cloud

Object detection --
Human retina image

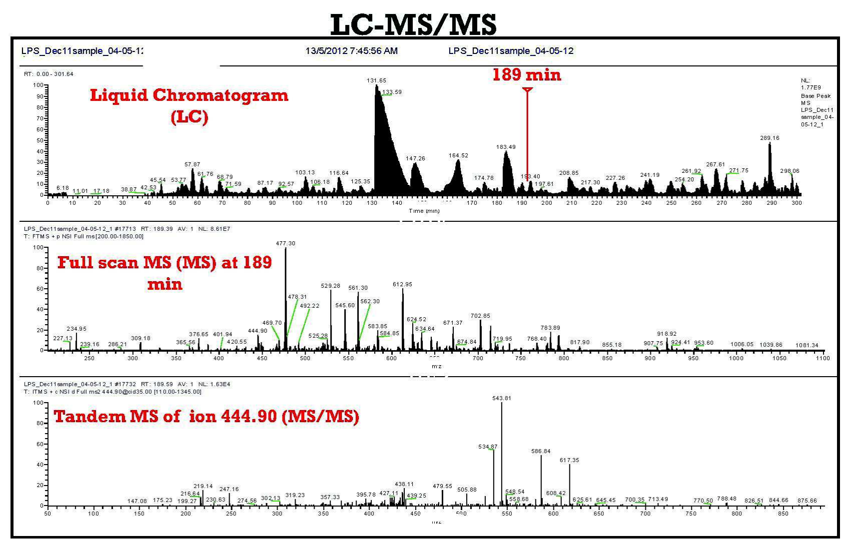
Feature Selection and Biomarker Detection



Biology data – LC-MS/MS



Cancer Diagnosis



Why Feature Selection ?

❖ "Curse of the dimensionality"
  - Large number of features: 100s, 1000s, even millions

❖ Not all features are useful (relevant)

❖ Redundant or irrelevant features may reduce the performance (e.g. **classification accuracy**)

❖ Costly: time, memory, and money

❖ Feature selection
  - to select a small subset of **relevant** features from the original large set of features in order to maintain or even improve the performance

Challenges in Feature Selection

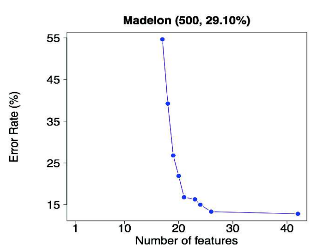
❖ Large search space: 2^n possible feature subsets
  - 1990: n < 20
  - 1998: n <= 50
  - 2007: n ≈ 100s
  - Now: 1000s, 1 000 000s

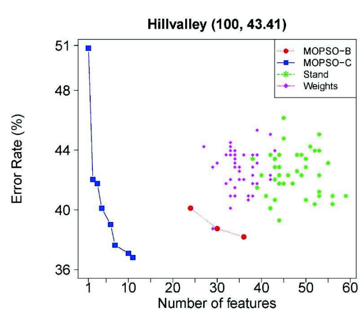
❖ Feature interaction
  - Relevant features may become redundant
  - Weakly relevant or irrelevant features may become highly useful

❖ Slow processing time, or even not possible

Multi-objective PSO for Feature Selection

❖ Introduce and develop the first multi-objective PSO approach to feature selection
  - Simultaneously minimise the number of features and the error rate





Bing Xue, Mengjie Zhang, Will Browne. "Particle swarm optimization for feature selection in classification: A multi-objective approach, IEEE Transactions on Cybernetics, vol. 43, no. 6, pp. 1656-1671, 2013. **[ARC/ERA Tier A, IF 3.783]**

PSO and Information Theory for Feature Selection

❖ Information theory in evolutionary feature selection
  - Fast algorithm — mutual information
  - New measures, evaluate multiple features
  - Evolutionary multi-objective filter feature selection

F-MI	0.05	0.05	0.05	0.06	0.07	0.09	0.15	0.18
F-E	2.88	97.7	8.64	27.95	9.85	256.57	2.96	236.42
F-RS	2.07	2485.61	8.21	55.3	14.81	1372.93	0.69	928.25
F-PRS	2.86	2766.29	8.28	38.36	9.95	1827.06	0.68	911.3
W-SVM	24.41	5143.18	53.28	270.64	118.37	2441.21	5.4	10937.87
W-5NN	6.12	9311.59	18.89	264.51	72.72	4095.07	1.68	1936.67
W-DT	5.19	189.43	10.53	43.15	47.87	244.55	3.82	529.7
W-NB	13.46	304.08	15.89	150.37	19.42	377.24	4.13	706.23

Bing Xue, Liam Cervante, Lin Shang, Will Browne, Mengjie Zhang. "A Multi-Objective Particle Swarm Optimisation for Filter Based Feature Selection in Classification Problems". Connection Science. Vol. 24, No. 2-3, pp. 91-116, 2012.

Bing Xue, Liam Cervante, Lin Shang, Will N. Browne, Mengjie Zhang. "Evolutionary Algorithms and Information Theory for Filter Based Feature Selection in Classification". International Journal on Artificial Intelligence Tools. Vol. 22, Issue 04, August 2013. pp. 1350024 -- 1 - 31. DOI: 10.1142/S0218213013500243.

EC and Statistical feature clustering

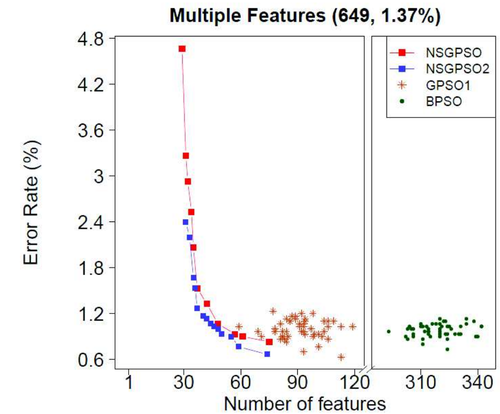
❖ Introduce statistical feature clustering to feature selection and develop the first approach

- reduce the size of the search space
- #features: from 600 to ~12
- implicitly consider feature interaction
- Example:
  - our method achieved accuracy 100%: {10, 7, 3}
  - Single feature ranking: 7, 10, 12, 1, 9, 11, 6, 2, 13, 5, 4, 3

Hoai Bach Nguyen, Bing Xue, Ivy Liu, Peter Andreae, Mengjie Zhang. "Gaussian Transformation based Representation in Particle Swarm Optimisation for Feature Selection". Proceedings of the 18th European Conference on the Applications of Evolutionary Computation (EuroApplications 2015). Lecture Notes in Computer Science. Vol. 9028. Copenhagen, Denmark. 8-10 April 2015. pp. 541-553 (Nominated as Best Paper)

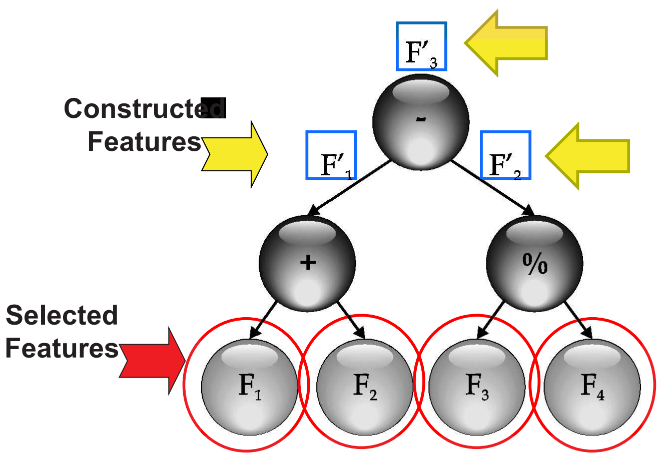
EC and Statistical feature clustering - Multi-Objective

- Feature selection:
  - minimise the number of features
  - minimise the error rate
- In MO, we aim to find the *Pareto front* of non-dominated solutions
- Two new MO methods: NSGPSO, NSGPSO2



Bing Xue, Micthell C. Lane, Ivy Liu, Mengjie Zhang, "Particle Swarm Optimisation for Feature Selection Based on Statistical Clustering", Evolutionary Computation (Journal, MIT Press), *Passed first round review with positive comments* **[ARC/ERA Tier A]**

GP for Feature Construction



Soha Ahmed, Mengjie Zhang, Lifeng Peng and Bing Xue."Multiple Feature Construction for Effective Biomarker Identification and Classification using Genetic Programming". Proceedings of 2014 Genetic and Evolutionary Computation Conference (GECCO 2014). ACM Press. Vancouver, BC, Canada. 12-16 July 2014.pp.249--256.

Biomarker Detection

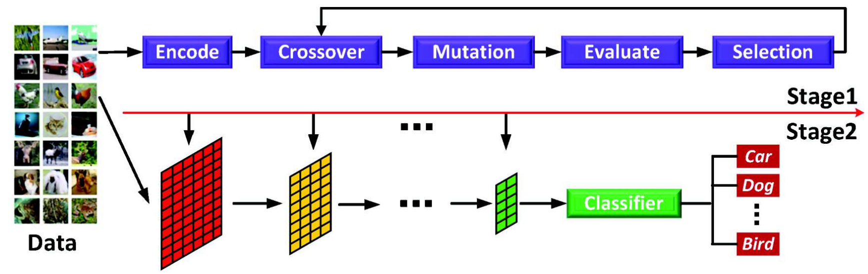
Apple minus m/z (5 biomarkers)	New Method (5)	Method B (2)
463.0	Yes	No
447.09	Yes	Yes
273.03	Yes	Yes
435.13	Yes	No
227.07	Yes	No

Soha Ahmed, Mengjie Zhang, Lifeng Peng and Bing Xue."Multiple Feature Construction for Effective Biomarker Identification and Classification using Genetic Programming". Proceedings of 2014 Genetic and Evolutionary Computation Conference (GECCO 2014). ACM Press. Vancouver, BC, Canada. 12-16 July 2014.pp.249--256.

Evolutionary Deep Learning for Image Classification

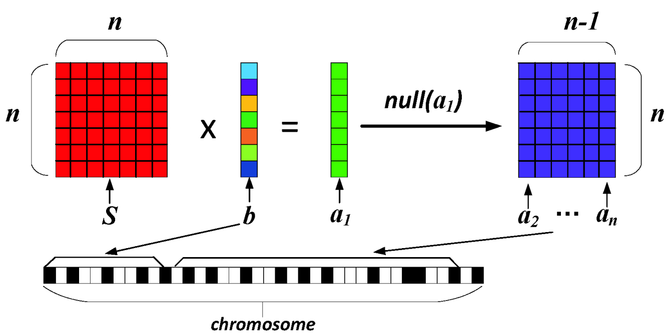
- GAs and PSO for evolving CNNs
- GAs for evolving Auto-encoders and CNNs
- GP for deep learning

Evolving Unsupervised Deep Neural Networks for Learning Meaningful Representations



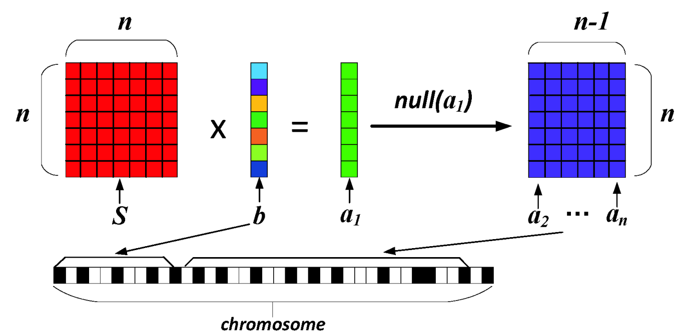
- The algorithm is composed of two stages
- The first stage one is for exploration, and the second stage is for exploitation
- The two stages collectively guarantee the best performance

Representation *of Evolving Unsupervised Deep Neural Networks for Learning Meaningful Representations*



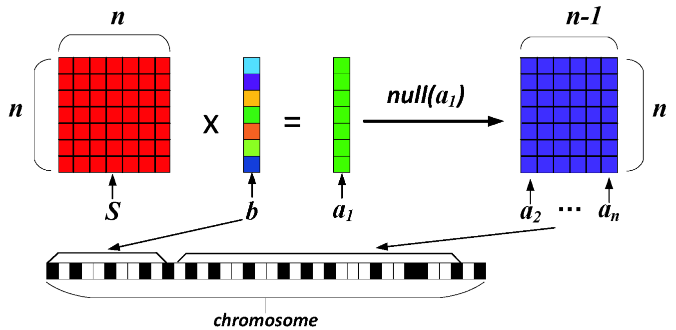
- The upper bound of the weight for a n-dimensional input is set to be $n \times n$ based on Yang's principle
- Randomly initialize a matrix with the size of $n \times n$
- Randomly initialize a n-dimensional vector b

Representation *of Evolving Unsupervised Deep Neural Networks for Learning Meaningful Representations*



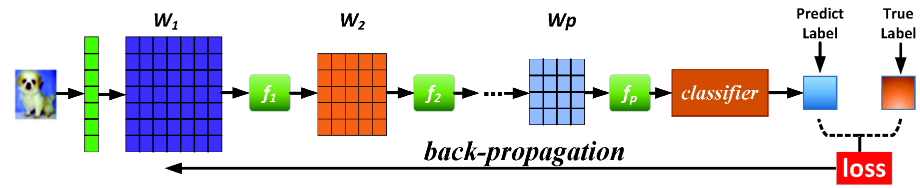
- Compute $a_1 = S \times b$
- Find the null space of a_1
- Encode b with real number and the using of $\{a_2, \cdots, a_n\}$ with bit string

Representation *of Evolving Unsupervised Deep Neural Networks for Learning Meaningful Representations*



- Another two bits in the chromosome are used to encode the type of predefined three activation functions
- Each chromosome is with the same length

Fitness Evaluation *of Evolving Unsupervised Deep Neural Networks for Learning Meaningful Representations*



- Using the learned matrix to initialize the network
- BP is used to train the network
- A SVM is added to the top of the network to estimate the classification accuracy on validate data

Genetic Operator *of Evolving Unsupervised Deep Neural Networks for Learning Meaningful Representations*

- One-point crossover operator is used twice for the parameters encoding b and $\{a_2, \cdots, a_n\}$
- Polynomial mutation operator with distribution index of 20 is used for mutation
- Top 20% elitisms are kept into the next generation

Datasets and Experimental Design *of Evolving Unsupervised Deep Neural Networks for Learning Meaningful Representations*

- MNIST, MNIST-basic, MNIST-rot, MNIST-back-rand, MNIST-back-image, MNIST-rot-back-image, Rectangles, Rectangles-image, Convex and Cifar10-bw are used as the datasets
- The proposed algorithm is implemented based on auto-encoder and RBM (EUDNN-AE, EUDNN-RBM)
- Denosing auto-encoder (DAE), contractive auto-encoder (CAE), sparse auto-encoder (SAE) and deep belief network (DBN) with up to three layers are used the compared algorithms

Datasets and Experimental Design *of Evolving Unsupervised Deep Neural Networks for Learning Meaningful Representations*

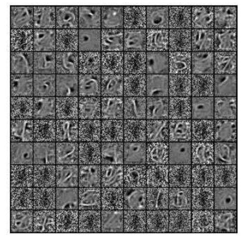
- Learning rate is selected from $\{0.0001, 0.001, 0.01, 0.1\}$, batch sizes vary in $\{10, 100, 200\}$
- Number of neurons for compared algorithms vary from 200 to $3{,}000$
- Independently run 30, and Mann-Whitney-Wilcoxon rank-sum test with a 5% significant level for statistical conclusion

Experimental Results *of Evolving Unsupervised Deep Neural Networks for Learning Meaningful Representations*

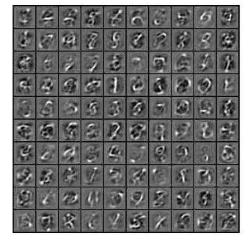
- Classification Accuracy

Benchmark	EUDNN		DAE	CAE	SAE	DBN
	AE	RBM				
MNIST	0.9878(0.00751)	**0.9885(0.00255)**	0.9820(0.00506)(+)	0.9843(0.00699)(+)	0.9832(0.00891)(+)	0.9771(0.00959)(+)
MNIST-basic	0.9674(0.00616)	0.9633(0.00473)	0.9580(0.00352)(+)	0.9635(0.00831)(+)	**0.9776(0.00585)**(-)	0.9658(0.00550)(+)
MNIST-rot	**0.7952(0.00917)**	0.7549(0.00286)	0.7274(0.00757)(+)	0.7706(0.00754)(+)	0.7852(0.00380)(+)	0.7639(0.00568)(+)
MNIST-back-rand	0.8843(0.00076)	0.8386(0.00054)	0.7725(0.00531)(+)	0.5741(0.00779)(+)	**0.8851(0.00934)**(=)	0.8221(0.00130)(+)
MNIST-back-image	0.4325(0.00569)	**0.4830(0.00469)**	0.4022(0.00012)(+)	0.4010(0.00337(+)	0.4638(0.00162)(+)	0.4587(0.00794)(+)
MNIST-rot-back-image	**0.8925(0.00906)**	0.8879(0.00815)	0.8691(0.00127)(+)	0.6574(0.00913)(+)	0.8733(0.00632)(+)	0.8830(0.00098)(=)
Rectangles	0.9627(0.00311)	**0.9681(0.00829)**	0.9232(0.00166)(+)	0.6275(0.00602)(+)	0.9408(0.00263)(+)	0.9622(0.00154)(=)
Rectangles-image	0.7521(0.00689)	0.7716(0.00048)	0.7598(0.00451)(+)	**0.7810(0.00784)**(=)	0.7725(0.00002)(-)	0.7628(0.00913)(+)
Convex	**0.8113(0.00052)**	0.8085(0.00826)	0.7930(0.00538(+)	0.8016(0.00996)(+)	0.8053(0.00878)(+)	0.7895(0.00443)(+)
Cifar10-bw	**0.4798(0.00107)**	0.4331(0.00962)	0.4309(0.00005)(+)	0.4860(0.00775)(+)	0.4423(0.00817)(+)	0.4598(0.00869)(+)
	+/-/=		10/0/0	9/0/1	7/2/1	8/0/2

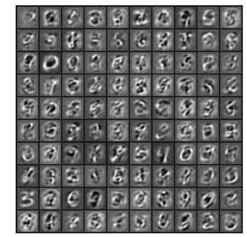
- Visualization



First layer



Second layer

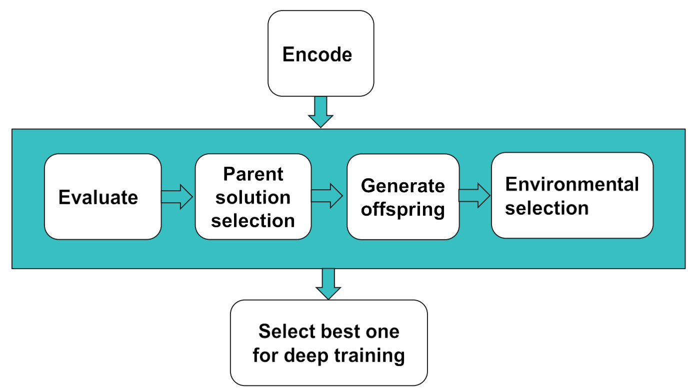


Third layer

Summary *of Evolving Unsupervised Deep Neural Networks for Learning Meaningful Representations*

- The proposed algorithm is capable of improving the classification accuracy against the compared algorithms
- Meaningful representations are learned in the intermediary layers

Evolving Deep Convolutional Neural Networks for Image Classification



Representation *of Evolving Deep Convolutional Neural Networks for Image Classification*

- One chromosome is divided into two parts
- The first part is for convolutional and pooling layers
- The second part for fully connected layers

- Encoded information of the convolution/pooling/fully connected layer

Unit Type	Encoded Information
convolutional layer	the filter width, the filter height, the number of feature maps, the stride width, the stride height, the convolutional type, the standard deviation and the mean value of filter elements
pooling layer	the kernel width, the kernel height, the stride width, the stride height, and the pooling type (i.e., the average or the maximal)
fully connected layer	the number of neurons, the standard deviation of connection weights, and the mean value of connection weights

Representation *of Evolving Deep Convolutional Neural Networks for Image Classification*

- The mean value and standard derivation are also encoded into the chromosome for initializing the weight matrix

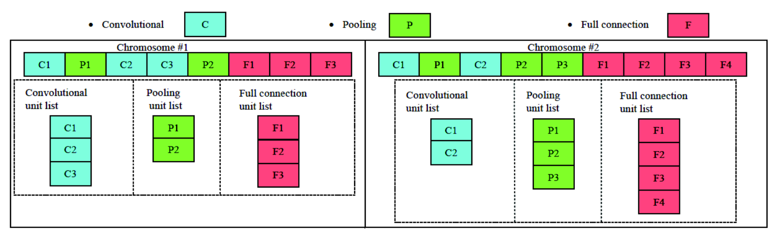
- Each chromosome may be with different lengths

Fitness Evaluation *of Evolving Deep Convolutional Neural Networks for Image Classification*

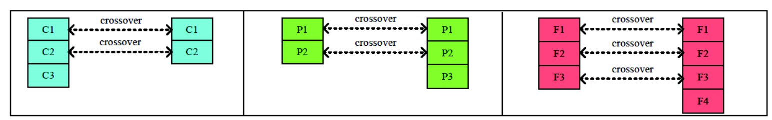
- Each individual is trained by BP with only several epochs for the classification accuracy
- The mean and standard derivation of the classification accuracy in the last epoch are calculated
- Using the mean value as the fitness of the individual, if with the same mean value, the standard derivation is used
- Large mean value or small standard derivation denotes the individual with good quality

Genetic Operator *of Evolving Deep Convolutional Neural Networks for Image Classification*

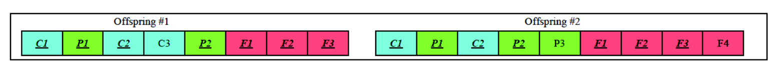
- The crossover is composed of three steps: unit collection, unit align and crossover and unit restore



(a) Unit Collection



(b) Unit Align and Crossover



(c) Unit Restore

Genetic Operator *of Evolving Deep Convolutional Neural Networks for Image Classification*

- An unit with the type of convolution/pooling fully connected may be added/removed/modified during the mutation
- The probability is 1/3 for choosing the unit type, and also 1/3 for the adding, removing and modifying operation
- During modifying, the polynomial mutation operator is used

Datasets and Experimental Design *of of Evolving Deep Convolutional Neural Networks for Image Classification*

- Fashion, Rectangle, Rectangle Images (RI), Convex Sets (CS), MNIST Basic (MB), MNIST with Background Images (MBI), MNIST with Random Background (MRB), MNIST with Rotated Digits (MRD), MNIST with RD plus Background Images (MRDBI) datasets are used
- The proposed algorithm (EvoCNN) is compared with a series of state-of-the-arts including 2C1P2F+Dropout, 2C1P, 3C2F, 3C1P2F+Dropout, GRU+SVM+Dropout, GoogleNet, AlexNet, SqueezeNet-200, MLP 256-128-64, and VGG16 on the Fashion dataset, and CAE-2, TIRBM, PGBM+DN-1, ScatNet-2, RandNet-2, PCANet-2 (softmax), LDANet-2, SVM+RBF, SVM+Poly, NNet, SAA-3 and DBN-3 on other datasets.

Datasets and Experimental Design *of Evolving Deep Convolutional Neural Networks for Image Classification*

- EvoCNN is initialized with the population size of 100
- Independently run 30 times
- The classification accuracy of the compared algorithms are from their seminal papers

Experimental Results *of Evolving Deep Convolutional Neural Networks for Image Classification*

- Classification Accuracy on Fashion dataset

classifier	error(%)	# parameters	# epochs
2C1P2F+Drouout	8.40(+)	3.27M	300
2C1P	7.50(+)	100K	30
3C2F	9.30(+)	—	—
3C1P2F+Dropout	7.40(+)	7.14M	150
GRU+SVM+Dropout	10.30(+)	—	100
GoogleNet [43]	6.30(+)	101M	—
AlexNet [3]	10.10(+)	60M	—
SqueezeNet-200 [53]	10.00(+)	500K	200
MLP 256-128-64	10.00(+)	41K	25
VGG16 [54]	6.50(+)	26M	200
EvoCNN (best)	5.47	6.68M	100
EvoCNN (mean)	7.28	6.52M	100

Experimental Results *of Evolving Deep Convolutional Neural Networks for Image Classification*

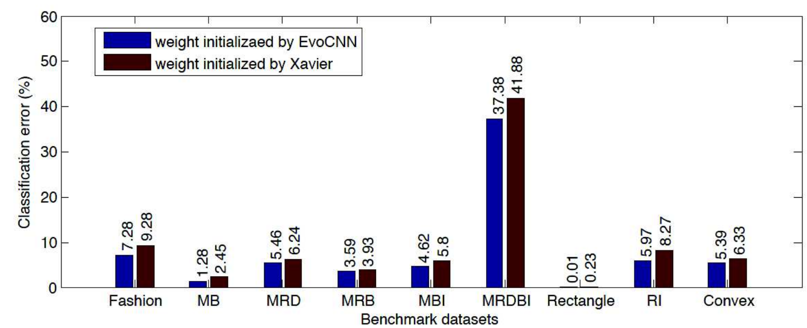
❖ Classification Accuracy

classifier	MB	MRD	MRB	MBI	MRDBI	Rectangle	RI	Convex
CAE-2 [55]	2.48(+)	9.66(+)	10.90(+)	15.50(+)	45.23(+)	1.21(+)	21.54(+)	—
TIRBM [56]	—	4.20(-)	—	—	35.50(+)	—	—	—
PGBM+DN-1 [57]	—	—	6.08(+)	12.25(+)	36.76(+)	—	—	—
ScatNet-2 [58]	1.27(+)	7.48(+)	12.30(+)	18.40(+)	50.48(+)	0.01(=)	8.02(+)	6.50(+)
RandNet-2 [59]	1.25(+)	8.47(+)	13.47(+)	11.65(+)	43.69(+)	0.09(+)	17.00(+)	5.45(+)
PCANet-2 (softmax) [59]	1.40(+)	8.52(+)	6.85(+)	11.55(+)	35.86(+)	0.49(+)	13.39(+)	4.19(-)
LDANet-2 [59]	1.05(-)	7.52(+)	6.81(+)	12.42(+)	38.54(+)	0.14(+)	16.20(+)	7.22(+)
SVM+RBF [52]	3.03(+)	11.11(+)	14.58(+)	22.61(+)	55.18(+)	2.15(+)	24.04(+)	19.13(+)
SVM+Poly [52]	3.69(+)	15.42(+)	16.62(+)	24.01(+)	56.41(+)	2.15(+)	24.05(+)	19.82(+)
NNet [52]	4.69(+)	18.11(+)	20.04(+)	27.41(+)	62.16(+)	7.16(+)	33.20(+)	32.25(+)
SAA-3 [52]	3.46(+)	10.30(+)	11.28(+)	23.00(+)	51.93(+)	2.41(+)	24.05(+)	18.41(+)
DBN-3 [52]	3.11(+)	10.30(+)	6.73(+)	16.31(+)	47.39(+)	2.61(+)	22.50(+)	18.63(+)
EvoCNN (best)	1.18	5.22	2.80	4.53	35.03	0.01	5.03	4.82
EvoCNN (mean)	1.28	5.46	3.59	4.62	37.38	0.01	5.97	5.39

Experimental Results *of Evolving Deep Convolutional Neural Networks for Image Classification*

❖ Performance regarding weight initialization

❖ The compared algorithm is the widely used Xavier method



Summary *of Evolving Deep Convolutional Neural Networks for Image Classification*

❖ The proposed algorithm can find a set of convolutional neural networks with significantly differences on the model complexity but similar performance

❖ The best model found by the proposed algorithm can achieve the comparative performance to state-of-the-art, but quite less number of parameters

❖ The proposed algorithm can find the best architecture of convolutional neural network, which gives rise to a better classification accuracy

❖ The proposed algorithm can also find the appreciable parameters for initializing the weight matrix

An Experimental Study on Hyper-parameter Optimization for Stacked Auto-Encoders

Using PSO to optimize the hyper-parameters in auto-encoder models

Using grid search to optimize the hyper-parameters of auto-encoder models

compare the classification accuracy on benchmark

compare the computational complexity

❖ This is an experimental study on hyper-parameter optimization for stacked auto-encoders

❖ The compared optimization methods are particle swarm optimization and grid search

Representation *of An Experimental Study on Hyper-parameter Optimization for Stacked Auto-Encoders*

- All auto-encoder models are general auto-encoder (AE), sparse auto-encoder (SAE), denosing auto-encoder (DAE) and contractive auto-encoder (CAE), their encoded parameters are:

Model	Hyper-parameter
AE	number of neurons n weight balance factor λ
SAE	number of neurons n weight balance factor λ predefined sparsity ρ sparsity balance factor β
DAE	number of neurons n weight balance factor λ corruption level z
CAE	number of neurons n weight balance factor λ contractive term balance factor γ

Fitness Evaluation *of An Experimental Study on Hyper-parameter Optimization for Stacked Auto-Encoders*

- Each individual encoded by PSO is trained by BP with only several epochs in the pre-training and fine tuning for the classification accuracy
- The classification accuracy is used as the fitness
- The standard velocity update operation of PSO is used

Datasets and Experimental Design *of of Evolving Deep Convolutional Neural Networks for Image Classification*

- MNIST Basic (MB), MNIST with Background Images (MBI), MNIST with Random Background (MRB), MNIST with Rotated Digits (MRD), MNIST with RD plus Background Images (MRDBI), Rectangle, Rectangle Images (RI) and Convex Sets (CS) datasets are used as the benchmarks.
- The PSO for hyper-parameter optimization (PSO-HO) and grid search for hyper-parameter optimization are the compared algorithms

Datasets and Experimental Design *of of Evolving Deep Convolutional Neural Networks for Image Classification*

- The epochs of pre-training, fine tuning and final training are set to be 30, 20 and 100.
- The PSO for hyper-parameter optimization (PSO-HO) and grid search for hyper-parameter optimization are the compared algorithms
- Independently run 20 times, Mann-Whitney-Wilcoxon rank-sum test with a 5% significant level is employed to statistically conduct the results

Datasets and Experimental Design *of Evolving Deep Convolutional Neural Networks for Image Classification*

- The auto-encoder models are investigated with up to three layers
- The range of searched parameters are:

Layer number	Search range for the number of neurons	Interval	Number of trials
First layer	[500, 3, 000]	0.12	23
Second layer	[500, 4, 000]	0.12	26
Third layer	[1, 000, 6, 000]	0.12	23

Name of hyper-parameter	Search range	Interval	Number of trials
weight balance factor λ	$[1e-10, 1e-5]$	0.15	112
predefined sparsity ρ	$[1e-10, 7e-1]$	2.0	18
sparsity balance factor β	$[1e-10, 1e+1]$	1.0	20
corruption level z	$[1e-10, 7e-1]$	2.0	18
contractive term balance factor γ	$[1e-10, 1e+1]$	1.0	20

Experimental Results *of Evolving Deep Convolutional Neural Networks for Image Classification*

- Classification Accuracy on the first layer

Benchmark	AE		SAE		DAE		CAE	
	PSO-HO	Grid Search	PSO-HO	Grid Search	PSO-HO	Grid Search	PSO-HO	Grid Search
MB	0.958(8.3E-4)	0.958(1.0E-3) =	0.930(1.2E-3)	0.926(8.1E-3) +	0.958(7.2E-3)	0.952(1.3E-3) +	0.960(1.3E-2)	0.955(1.5E-2) +
MBI	0.752(8.4E-3)	0.749(8.7E-3) +	0.683(6.3E-3)	0.705(9.8E-3) -	0.739(1.3E-2)	0.752(5.9E-3) -	0.759(8.9E-3)	0.743(1.7E-2) +
MRB	0.780(1.2E-3)	0.779(1.0E-3) =	0.737(3.8E-3)	0.765(3.5E-2) +	0.793(9.3E-3)	0.771(7.3E-3) +	0.833(4.0E-3)	0.823(7.8E-3) +
MRD	0.869(1.5E-3)	0.867(1.6E-3) +	0.544(1.5E-2)	0.463(8.2E-2) +	0.863(3.5E-2)	0.844(1.3E-3) +	0.652(3.3E-3)	0.455(8.1E-3) +
MRDBI	0.449(3.4E-3)	0.450(4.2E-3) +	0.336(7.3E-3)	0.333(3.2E-2) +	0.434(2.1E-3)	0.391(1.2E-3) +	0.442(4.9E-3)	0.422(3.9E-2) +
Rectangle	0.922(3.7E-4)	0.922(2.8E-4) =	0.688(8.9E-3)	0.649(5.8E-2) +	0.915(3.5E-2)	0.932(2.1E-4) -	0.872(5.7E-3)	0.777(7.5E-2) +
RI	0.761(2.8E-3)	0.761(3.5E-3) =	0.736(1.2E-2)	0.555(9.0E-2) +	0.754(1.0E-2)	0.755(5.8E-3) =	0.754(2.8E-3)	0.754(8.4E-3) =
CS	0.797(4.1E-3)	0.799(4.8E-3) -	0.671(2.6E-2)	0.557(3.9E-2) +	0.795(9.3E-3)	0.794(5.2E-3) +	0.752(8.5E-3)	0.735(3.6E-2) +
+/-/=		3/1/4		7/1/0		5/2/1		7/0/1

Experimental Results *of Evolving Deep Convolutional Neural Networks for Image Classification*

- Classification Accuracy on the second layer

Benchmark	AE		SAE		DAE		CAE	
	PSO-HO	Grid Search	PSO-HO	Grid Search	PSO-HO	Grid Search	PSO-HO	Grid Search
MB	0.959(1.5E-3)	0.959(1.4E-3) =	0.943(2.3E-3)	0.918(2.1E-2) +	0.955(5.9E-3)	0.959(1.5E-3) =	0.962(7.9E-3)	0.950(1.5E-2) +
MBI	0.775(1.8E-2)	0.773(1.1E-2) +	0.705(9.7E-3)	0.664(8.7E-2) +	0.762(1.4E-2)	0.742(1.4E-2) +	0.772(8.3E-3)	0.748(4.6E-2) +
MRB	0.769(6.8E-2)	0.768(6.6E-2) =	0.770(7.8E-3)	0.736(1.5E-1) +	0.794(1.4E-2)	0.794(7.3E-2) =	0.850(2.6E-3)	0.847(5.3E-3) +
MRD	0.867(4.1E-3)	0.866(3.3E-3) =	0.167(9.6E-3)	0.160(9.6E-3) +	0.866(1.88E-2)	0.855(1.2E-3) +	0.138(4.3E-3)	0.112(8.7E-3) +
MRDBI	0.450(1.0E-2)	0.449(9.5E-3) +	0.352(3.0E-3)	0.280(8.9E-2) +	0.420(1.4E-3)	0.400(2.9E-3) +	0.441(7.1E-3)	0.408(4.5E-2) +
Rectangle	0.938(2.3E-3)	0.940(2.6E-3) -	0.647(4.5E-2)	0.546(6.3E-2) +	0.936(3.2E-2)	0.932(1.6E-3) +	0.885(3.5E-3)	0.803(9.4E-2) +
RI	0.756(5.0E-3)	0.755(5.5E-3) +	0.523(5.0E-2)	0.500(3.9E-3) +	0.743(1.7E-2)	0.750(4.6E-3) -	0.741(6.8E-3)	0.738(5.6E-2) +
CS	0.800(9.9E-3)	0.802(8.5E-3) =	0.620(3.2E-2)	0.517(1.9E-2) +	0.805(8.7E-3)	0.799(6.9E-3) +	0.753(7.5E-3)	0.671(9.6E-2) +
+/-/=		3/1/4		8/0/0		5/1/2		8/0/0

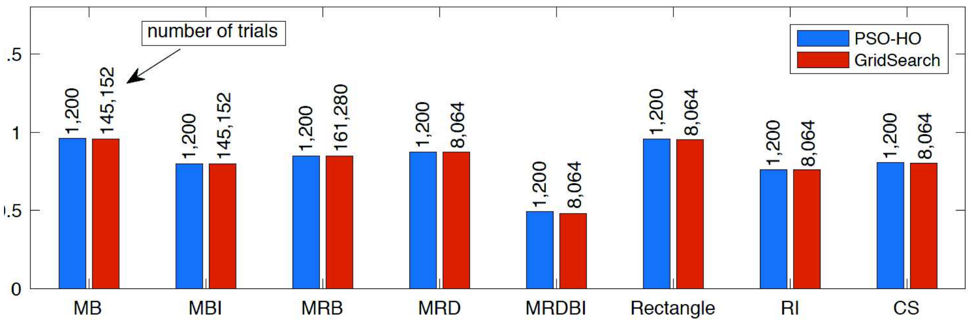
Experimental Results *of Evolving Deep Convolutional Neural Networks for Image Classification*

- Classification Accuracy on the third layer

Benchmark	AE		SAE		DAE		CAE	
	PSO-HO	Grid Search	PSO-HO	Grid Search	PSO-HO	Grid Search	PSO-HO	Grid Search
MB	0.959(3.1E-3)	0.955(2.7E-2) +	0.850(4.1E-2)	0.221(1.9E-1) +	0.960(7.5E-3)	0.959(3.4E-3) +	0.103(5.4E-3)	0.100(5.0E-3) +
MBI	0.783(2.0E-2)	0.698(1.8E-2) +	0.619(1.5E-1)	0.619(1.0E-1) =	0.799(2.4E-2)	0.797(1.1E-2) +	0.720(3.0E-2)	0.126(8.8E-2) +
MRB	0.785(3.6E-2)	0.790(2.8E-3) -	0.752(2.8E-2)	0.545(2.5E-1) +	0.804(2.9E-2)	0.816(6.2E-3) -	0.821(9.2E-3)	0.722(1.9E-1) +
MRD	0.875(8.5E-3)	0.875(7.8E-3) =	0.150(2.2E-2)	0.153(2.8E-2) =	0.865(1.3E-2)	0.864(1.2E-3) +	0.182(2.7E-3)	0.130(2.2E-3) +
MRDBI	0.492(8.6E-3)	0.481(1.6E-2) +	0.312(1.8E-2)	0.212(3.4E-2) +	0.451(6.3E-2)	0.422(5.3E-3) +	0.101(6.2E-3)	0.100(3.1E-3) =
Rectangle	0.956(2.6E-3)	0.953(4.1E-4) =	0.516(1.7E-2)	0.500(3.1E-4) +	0.949(4.3E-2)	0.945(1.8E-3) +	0.500(3.1E-4)	0.500(3.0E-4) =
RI	0.761(6.9E-3)	0.751(9.7E-3) +	0.500(2.2E-3)	0.500(2.3E-3) =	0.759(1.5E-2)	0.758(5.2E-3) =	0.499(2.2E-3)	0.500(2.2E-3) =
CS	0.803(5.8E-3)	0.786(8.4E-3) +	0.670(2.1E-2)	0.507(1.2E-2) +	0.803(8.1E-3)	0.799(6.7E-3) +	0.510(1.1E-2)	0.400(2.2E-3) +
+/-/=		5/1/2		5/0/3		6/1/1		5/0/3

Experimental Results *of Evolving Deep Convolutional Neural Networks for Image Classification*

❖ The numbers of trials are calculated for measuring the computational complexity



Summary *of Evolving Deep Convolutional Neural Networks for Image Classification*

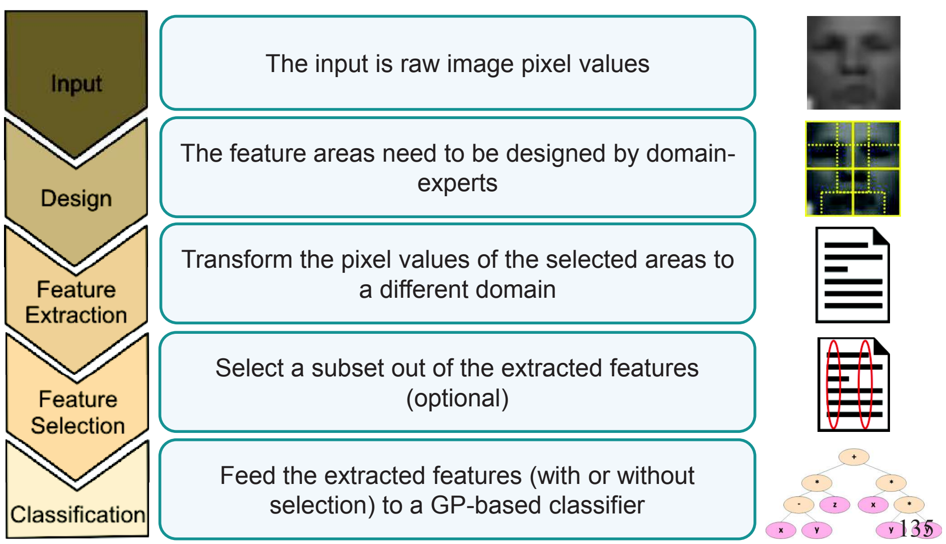
❖ PSO-HO can achieve the comparative classification accuracy but only take 10% to 1% computational complexity to that of grid search.

❖ J. Yang, A. F. Frangi, J. Y. Yang, D. Zhang, and Z. Jin, “Kpca plus lda: a complete kernel fisher discriminant framework for feature extraction and recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 2, pp. 230–244, 2005.

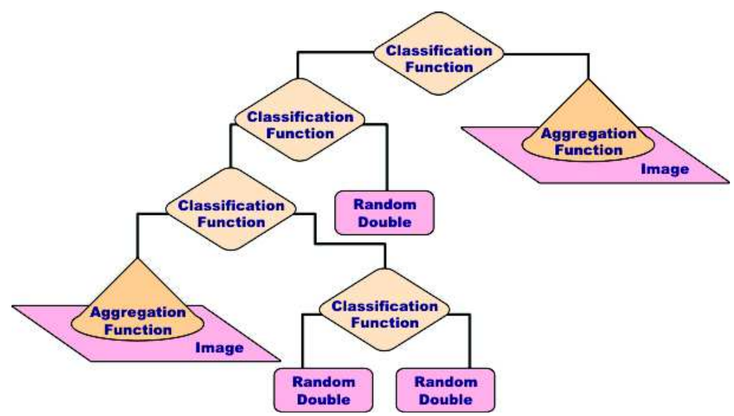
GP for Image Recognition/Classification

The traditional way

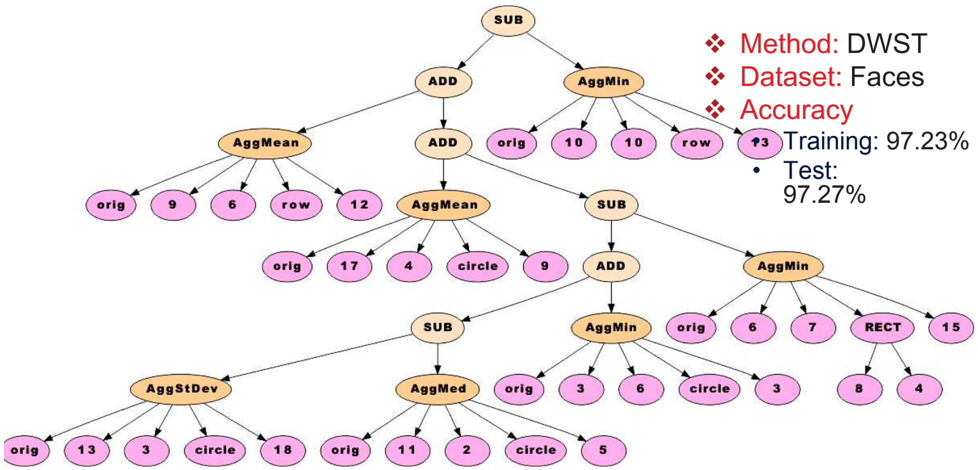
Domain-specific pre-extracted features approach



Two-Tier GP Method



Analysis



- Method: DWST
- Dataset: Faces
- Accuracy
  - Training: 97.23%
  - Test: 97.27%

Other Applications

- There have been many ECV applications for the past >15 years
- The work can be seen from;
  - Past EvoIASP workshop proceedings
  - CEC proceedings, special session on ECV
  - GECCO applications, EuroGP proceedings
  - IEEE TEC, IEEE TSMC (Part B) or TCYB, ECJ, PRL letters, …

Conclusions

- EC techniques play more and more important role in image analysis, signal processing and pattern recognition tasks
- Difficult and Challenging tasks even need more EC.
- Evolutionary deep learning will be the current and future directions for image classification for the next 5-10 years

Upcoming Conferences

- Special Session on Evolutionary Computer Vision, CEC 2019: IEEE Congress on Evolutionary Computation
  - Organisers: Mengjie Zhang, Victor Ciesielski, Mario Koeppen,
  - **Paper Submission deadline: 07 Jan 2019**
- Evostar 2019/EvoIASP 2019: Track on Evolutionary Computation in Image Analysis, Signal Processing and Pattern Recognition
  - Track Chairs: Stefano Cagnoni, Mengjie Zhang
  - Time/Venue: Amsterdam
  - **Paper Submission deadline: 10 Nov 2018**
- Special Session on Evolutionary Computation in Feature Selection and Construction, CEC 2019: IEEE Congress on Evolutionary Computation
  - Organisers: Bing Xue, Mengjie Zhang, Yaochu Jin
  - **Paper Submission deadline: 07 Jan 2019**

10-13 June 2019

The 31st Australasian Joint Conference on Artificial Intelligence

4-7 December 2018, Wellington, New Zealand

https://ecs.victoria.ac.nz/Events/AI2018/

Acknowledgement

- Marsden Fund council from the government funding (VUW1209, VUW1509, VUW1615), administrated by the Royal Society of New Zealand.
- Thanks my colleagues and research students particularly Dr Bing Xue, Dr Yanan Sun, A/Prof Will Browne, Dr Will Smart and Dr Ignas Kukenys, Dr Toktam Ebadi, Dr Mahdi Setayesh, Dr Wenlong Fu, Dr Andy Song, Harith Al-Sahaf, Dr Yuyu Liang, Liam Cervante, Mitch lane, etc.
- Thanks GECCO2018 organisers

More Recent Group Photo

35 people -- several people are missing!

References

- Yanan Sun, Gary G. Yen, Zhang Yi, "Evolving Unsupervised Deep Neural Networks for Learning Meaningful Representations". IEEE Transactions on Evolutionary Computation. DOI:10.1109/TEVC.2018.2808689.
- Toktam Ebadi, Ignas Kukenys, Will N. Browne, Mengjie Zhang: Human-Interpretable Feature Pattern Classification System Using Learning Classifier Systems. Evolutionary Computation 22(4): 629-650 (2014)
- Bing Xue, Mengjie Zhang, Will N. Browne: Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach. IEEE T. Cybernetics 43(6): 1656-1671 (2013)
- Soha Ahmed, Mengjie Zhang, Lifeng Peng and Bing Xue."Multiple Feature Construction for Effective Biomarker Identification and Classification using Genetic Programming". Proceedings of 2014 Genetic and Evolutionary Computation Conference (GECCO 2014). ACM Press. Vancouver, BC, Canada. 12-16 July 2014.pp.249--256.
- Bing Xue, Liam Cervante, Lin Shang, Will Browne and Mengjie Zhang. “Binary PSO and rough set theory for feature selection: a multi-objective filter based approach". International Journal of Computational Intelligence and Applications, Vol. 13, No. 2 (2014). pp. 1450009 -- 1- 34. DOI: 10.1142/S1469026814500096
- Hoai Bach Nguyen, Bing Xue, Ivy Liu, Peter Andreae, Mengjie Zhang. "Gaussian Transformation based Representation in Particle Swarm Optimisation for Feature Selection". Proceedings of the 18th European Conference on the Applications of Evolutionary Computation (EuroApplications 2015). Lecture Notes in Computer Science. Vol. 9028. Copenhagen, Denmark. 8-10 April 2015. pp. 541-553

References

- Yanan Sun, Bing Xue, Mengjie Zhang, Gary G. Yen, An Experimental Study on Hyper-parameter Optimization for Stacked Auto-Encoders, 2018 IEEE Congress on Evolutionary Computation (CEC). Accepted.
- Bing Xue, Mengjie Zhang, Will Browne, Xin Yao. "A Survey on Evolutionary Computation Approaches to Feature Selection", IEEE Transaction on Evolutionary Computation, vol. 20, no. 4, pp. 606-626, Aug. 2016. doi: 10.1109/TEVC.2015.2504420.
- Stefano Cagnoni, GECCO 2008 and GECCO 2014 Tutorial on ECV/IASP
- Will Smart, Mengjie Zhang. "Tracking Object Positions in Real-time Video using Genetic Programming". In *Proceeding of Image and Vision Computing International Conference,* 2004. pp. 113-118.
- Huayang Xie, Mengjie Zhang, Peter Andreae. "Genetic Programming for Automatic Stress Detection in Spoken English". Proceedings of EvoWorkshops 2006 (EvoIASP 2006), *Lecture Notes in Computer Science*, **Vol. 3907.** Springer. 2006. pp.460-471.
- Peter Andreae, Huayang Xie, Mengjie Zhang. "Genetic Programming for Detecting Rhythmic Stress in Spoken English". *International Journal of Knowledge-Based and Intelligent Engineering Systems (KES Journal). Special Issue on Genetic Programming.* **Vol. 12**, No. 1, 2008. pp. 15-28.
- Mahdi Setayesh, Mengjie Zhang, Mark Johnston:Investigating Particle Swarm Optimisation Topologies for Edge Detection in Noisy Images. Australasian Conference on Artificial Intelligence 2011: 609-618

References

- Yuyu Liang, Mengjie Zhang, Will N. Browne: A Supervised Figure-Ground Segmentation Method Using Genetic Programming. EvoApplications 2015: 491-503
- Mahdi Setayesh, Mengjie Zhang, Mark Johnston: A novel particle swarm optimisation approach to detecting continuous, thin and smooth edges in noisy images. Inf. Sci. 246: 28-51 (2013)
- Aaron Scoble, Will N. Browne, Bill Stephenson, Zane Bruce, Mengjie Zhang: Evolutionary spatial auto-correlation for assessing earthquake liquefaction potential using Parallel Linear Genetic Programming. IEEE Congress on Evolutionary Computation 2013: 2940-2947
- Andy Song, Mengjie Zhang: Genetic programming for detecting target motions. Connect. Sci. 24(2-3): 117-141 (2012)
- Harith Al-Sahaf, Andy Song, Kourosh Neshatian, Mengjie Zhang: Two-Tier genetic programming: towards raw pixel-based image classification. Expert Syst. Appl. 39(16): 12291-12301 (2012)
- Harith Al-Sahaf, Andy Song, Kourosh Neshatian, Mengjie Zhang: Extracting image features for classification by two-tier genetic programming. IEEE Congress on Evolutionary Computation 2012: 1-8
- Harith Al-Sahaf, Mengjie Zhang, Mark Johnston: Genetic Programming for Multiclass Texture Classification Using a Small Number of Instances. SEAL 2014: 335-346