Robust fuzzy clustering models with applications in medical imaging

László Szilágyi

September 6, 2022

MICCAI

- Medical Image Computation and Computer Assisted Interventions
  - The yearly medical imaging conference where the academia meets industry
  - This is where industry really goes fishing
- MICCAI 2011 Toronto
  - “Meet the leaders” session organized for young researchers
  - My question was: “Is there a future for medical image processing or everything is already discovered, solved, etc.?”
  - The era of level sets, shape models, appearance models seemed to be over
  - The answer was: medical image processing is expected to have a great future. And indeed, it has.

What happened since MICCAI 2011?

- Data
  - Earlier everybody had to work with own data sets.
  - Now there are hundreds of challenges announced every year, most of which releasing data sets.
- Methodology
  - The evolution of computers and GPUs opened the horizon for CNN networks and deep learning. Much more complex methods can be implemented than earlier.
- MICCAI
  - Earlier: 250 accepted papers, 500+ participants from industry
  - Nowadays: 1000+ accepted papers, 2000+ participants from industry
  - MICCAI 2022: 30+ challenges

Need for ~~speed~~ automated image processing

- The number of medical imaging devices involved in clinical practice is rising
- The daily produced medical image data is constantly growing
- The number of human experts who can process the image data … (???)
- It would be expensive to train the necessary number of experts.
- Would it be possible to find enough candidates? Probably no.
- There is a need for automated methods and procedures
  - To perform the bulk of the image processing tasks
  - To find the “suspected to be positive” cases
  - To show the human expert the positive records, human expert has the final word
  - Sometimes the computer is more accurate than a single human expert (e.g. mammography)
- Most important thing is ACCURACY. Minimize FALSE NEGATIVEs.

History: FCM, PCM şi PFCM

- Fuzzy c-means (Bezdek, 1981), Possibilistic c-means (Krishnapuram & Keller, 1993), Possibilistic-fuzzy c-means (Pal et al, 2005)
- Objective function

$$J_{\mathrm{FCM}} = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^{m} ||\mathbf{x}_k - \mathbf{v}_i||_{\mathbf{A}}^2 \qquad J_{\mathrm{PCM}} = \sum_{i=1}^{c} \sum_{k=1}^{n} \left[t_{ik}^{p} ||\mathbf{x}_k - \mathbf{v}_i||_{\mathbf{A}}^2 + (1 - t_{ik})^p \eta_i \right] \qquad J_{\mathrm{PFCM}} = \sum_{i=1}^{c} \sum_{k=1}^{n} [a u_{ik}^{m} + b t_{ik}^{p}] d_{ik}^2 + \sum_{i=1}^{c} \eta_i \sum_{k=1}^{n} (1 - t_{ik})^p$$

- Constraints

$$\sum_{i=1}^{c} u_{ik} = 1 \qquad \forall k = 1 \ldots n \qquad \begin{cases} 0 \le t_{ik} \le 1 & \forall i = 1 \ldots c, \forall k = 1 \ldots n \\ 0 < \sum_{i=1}^{c} t_{ik} < c & \forall k = 1 \ldots n \end{cases}$$

- Partition update formula

$$u_{ik}^{\star} = \frac{d_{ik}^{-2/(m-1)}}{\sum_{j=1}^{c} d_{jk}^{-2/(m-1)}} \qquad \begin{matrix} \forall i = 1 \ldots c \\ \forall k = 1 \ldots n \end{matrix} \qquad t_{ik}^{\star} = \left[1 + \left(\frac{d_{ik}^2}{\eta_i} \right)^{1/(p-1)} \right]^{-1} \qquad \begin{matrix} \forall i = 1 \ldots c \\ \forall k = 1 \ldots n \end{matrix} \qquad t_{ik}^{\star} = \left[1 + \left(\frac{b d_{ik}^2}{\eta_i} \right)^{1/(p-1)} \right]^{-1} \qquad \begin{matrix} \forall i = 1 \ldots c \\ \forall k = 1 \ldots n \end{matrix}$$

- Cluster prototype update formula

$$\mathbf{v}_i^{\star} = \frac{\sum_{k=1}^{n} u_{ik}^{m} \mathbf{x}_k}{\sum_{k=1}^{n} u_{ik}^{m}} \qquad \forall i = 1 \ldots c \qquad \mathbf{v}_i^{\star} = \frac{\sum_{k=1}^{n} t_{ik}^{p} \mathbf{x}_k}{\sum_{k=1}^{n} t_{ik}^{p}} \qquad \forall i = 1 \ldots c \qquad \mathbf{v}_i^{\star} = \frac{\sum_{k=1}^{n} [a u_{ik}^{m} + b t_{ik}^{p}] \mathbf{x}_k}{\sum_{k=1}^{n} [a u_{ik}^{m} + b t_{ik}^{p}]} \qquad \forall i = 1 \ldots c$$

Some properties

- All these algorithms are very popular
- All of them have some disadvantages:
  - FCM is sensitive to noise (outliers)
  - PCM can produce coincident clusters (Keller 2009: “this is a property, not a disadvantage”)
  - PFCM can attenuate these effects, but cannot eliminate them
- It would be useful to have some algorithm that works like gravity
  - An outlier should not have any effect on the clusters
    - Classical approach: F(c+1)M (Dave 1992)
    - Fuzzy-possibilistic product partition (Szilágyi, MDAI 2011)

Fuzzy-possibilistic product partition

- Intuition: we need a cluster prototype update formula like this

$$\mathbf{v}_i^\star = \frac{\sum_{k=1}^{n} \mu_{ik}^m \tau_{ik}^p \mathbf{x}_k}{\sum_{k=1}^{n} \mu_{ik}^m \tau_{ik}^p} \qquad \forall\, i = 1 \ldots c$$

- Probabilistic term and possibilistic term, not necessary to be the same as in FCM and PCM
- The operation between them is weighted averaging but multiplication

FPPPCM (FP3CM)

- Fuzzy-Possibilistic Product Partition C-Means (Szilágyi L, MDAI 2011)
- Obiective function

$$J_{\mathrm{FP3CM}} = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^{m} \left[t_{ik}^{p} ||\mathbf{x}_k - \mathbf{v}_i||_{\mathbf{A}}^{2} + (1 - t_{ik})^{p} \eta_i \right]$$

- Constraints

$$\sum_{i=1}^{c} u_{ik} = 1 \qquad \forall\, k = 1 \ldots n \qquad \begin{cases} 0 \leq t_{ik} \leq 1 & \forall\, i = 1 \ldots c, \forall\, k = 1 \ldots n \\ 0 < \sum\limits_{i=1}^{c} t_{ik} < c & \forall\, k = 1 \ldots n \end{cases}$$

- Partition update formulas

$$t_{ik}^{\star} = \left[1 + \left(\frac{d_{ik}^2}{\eta_i} \right)^{1/(p-1)} \right]^{-1} \qquad \begin{array}{l} \forall\, i = 1 \ldots c \\ \forall\, k = 1 \ldots n \end{array}$$

$$u_{ik}^{\star} = \frac{\left[t_{ik}^{p} d_{ik}^{2} + \eta_i (1 - t_{ik})^{p} \right]^{-1/(m-1)}}{\sum\limits_{j=1}^{c} \left[t_{jk}^{p} d_{jk}^{2} + \eta_j (1 - t_{jk})^{p} \right]^{-1/(m-1)}} \qquad \begin{array}{l} \forall\, i = 1 \ldots c \\ \forall\, k = 1 \ldots n \end{array}$$

- Cluster prototypes update formula

$$\mathbf{v}_i^{\star} = \frac{\sum\limits_{k=1}^{n} u_{ik}^{m} t_{ik}^{p} \mathbf{x}_k}{\sum\limits_{k=1}^{n} u_{ik}^{m} t_{ik}^{p}} \qquad \forall\, i = 1 \ldots c$$

The FPPPCM (FP3CM) algorithm

Algorithm 1: The alternating optimization algorithm of FP3CM clustering algorithm

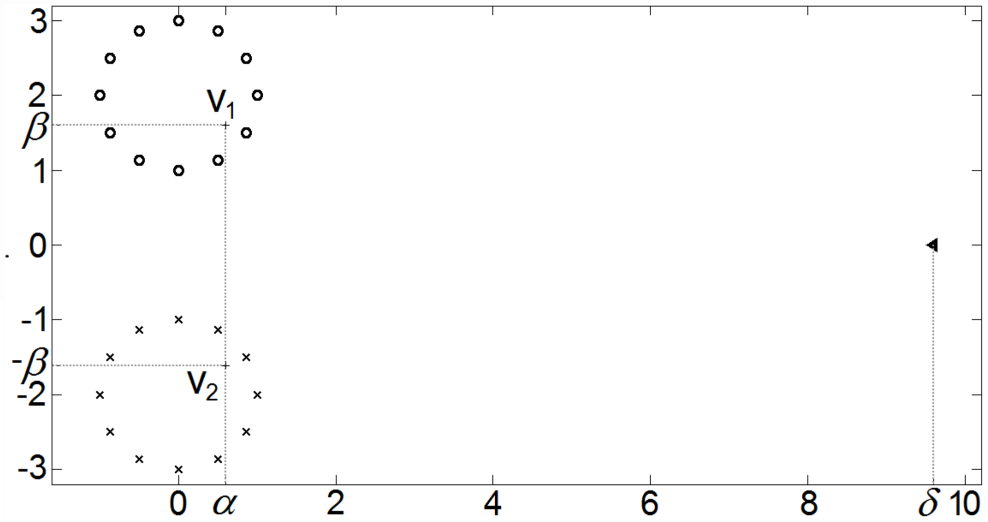
Data: Input data $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$
Result: Final cluster prototypes $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c$
Result: Partition matrices $\mathbf{U} = \{u_{ik}\}$ and $\mathbf{T} = \{t_{ik}\}$, with $i = 1 \ldots c$, $k = 1 \ldots n$
Fix the number of clusters c, $2 \leq c \leq n$;
Set fuzzy exponent m and possibilistic exponent p, both greater than 1;
Set possibilistic penalty terms η_i $(i = 1 \ldots c)$;
Initialize cluster prototypes $\mathbf{v}_i$ $(i = 1 \ldots c)$;
repeat
    Update possibilistic membership values using Eq. (44);
    Update probabilistic membership values using Eq. (48);
    Update cluster prototypes using Eq. (51);
until *cluster prototypes* $\mathbf{v}_i$ $(i = 1 \ldots c)$ *converge*;
Defuzzyfy of the obtained product partition as indicated in Eq. (53).

Advantages

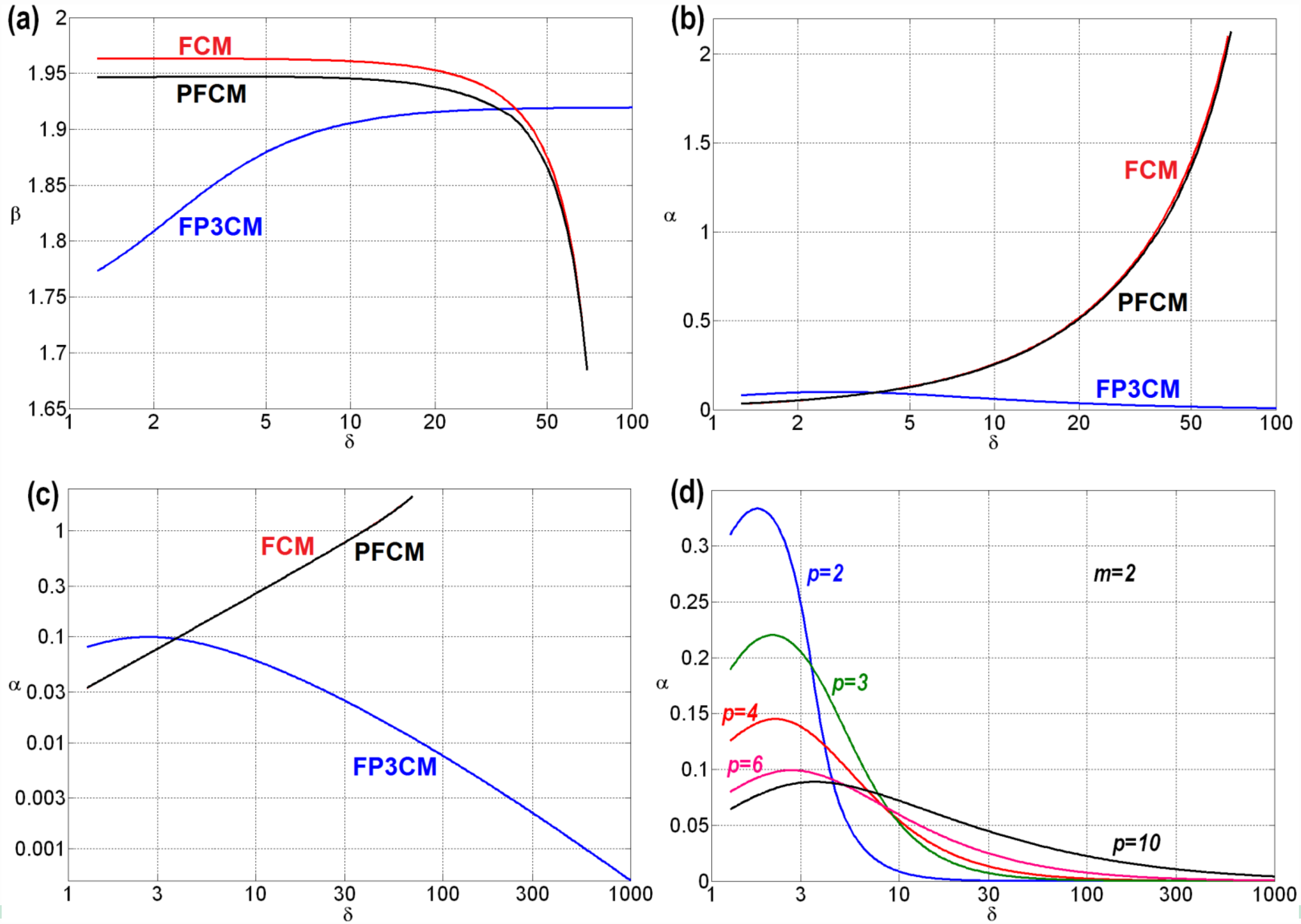
- It uses less parameters than PFCM (c+2 instead of c+4)
- Not sensible to outliers
- Creates fine partitions, comparable to PFCM if there are no outliers

- Disadvantage: initialization of cluster prototypes need more attention

One example

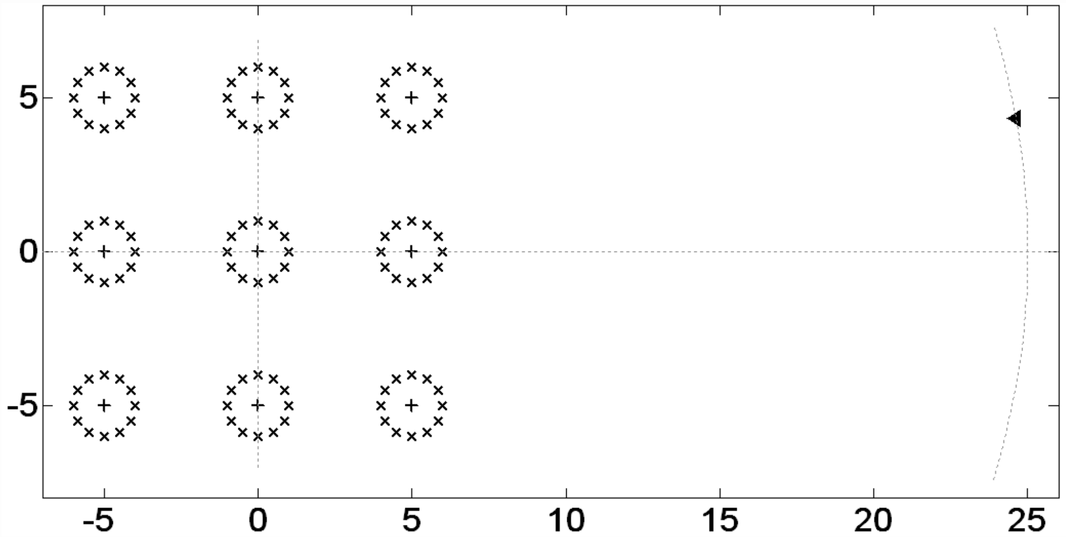


3
2
β
1
0
-1
-β
-2
-3
V1
V2
0
α
2
4
6
8
δ
10



(a)
FCM
PFCM
FP3CM
β
δ
(b)
FCM
PFCM
FP3CM
α
δ
(c)
FCM
PFCM
FP3CM
α
δ
(d)
p=2
p=3
p=4
p=6
p=10
m=2
α
δ

Another example



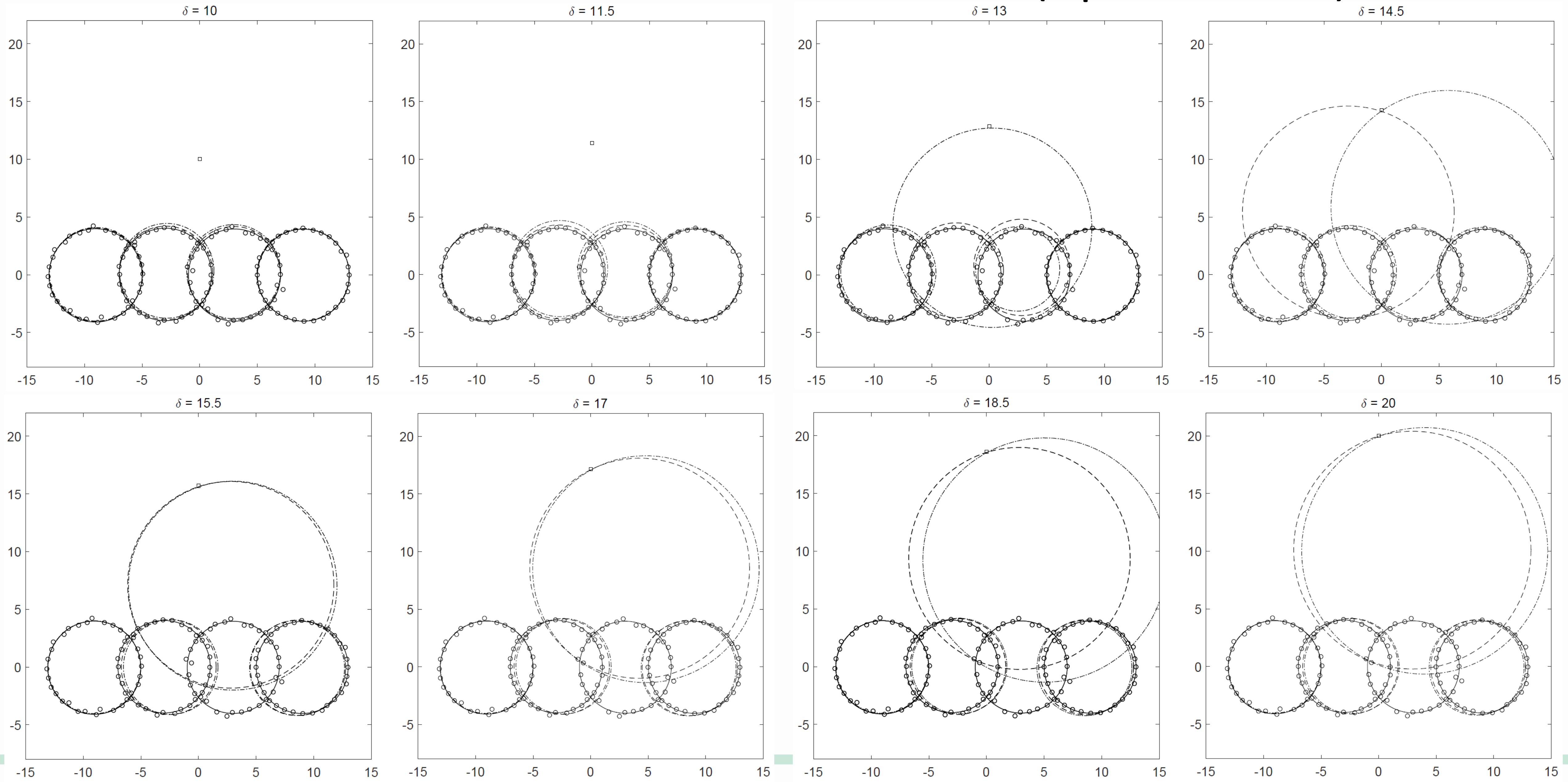
Algorithm	Circumstances m	p	$\sqrt{\eta_i}$	a	b	Limit distance
FCM	2					361
FPCM	2	5				361
FPCM	2	2				367
FPCM	2	1.2				401
PFCM	2	2	1.0	2	3	410
PFCM	2	2	1.5	2	3	479
PFCM	2	2	2.0	2	3	563
PFCM	2	2	2.5	2	3	649
PFCM	2	5	1.0	1	5	394
PFCM	2	5	1.5	1	5	421
PFCM	2	5	2.0	1	5	428
PFCM	2	5	2.5	1	5	370
PFCM	2	3	1.0	1	5	437
PFCM	2	3	1.5	1	5	521
PFCM	2	3	2.0	1	5	593
PFCM	2	3	2.5	1	5	546
PFCM	2	2	1.0	1	5	459
PFCM	2	2	1.5	1	5	602
PFCM	2	2	2.0	1	5	789
PFCM	2	2	2.5	1	5	1001
PFCM	2	2	3.0	1	5	1220
PFCM	2	2	4.0	1	5	1354
PFCM	2	2	5.0	1	5	1089
FP3CM	wide range					$+\infty$

Clustering IRIS data

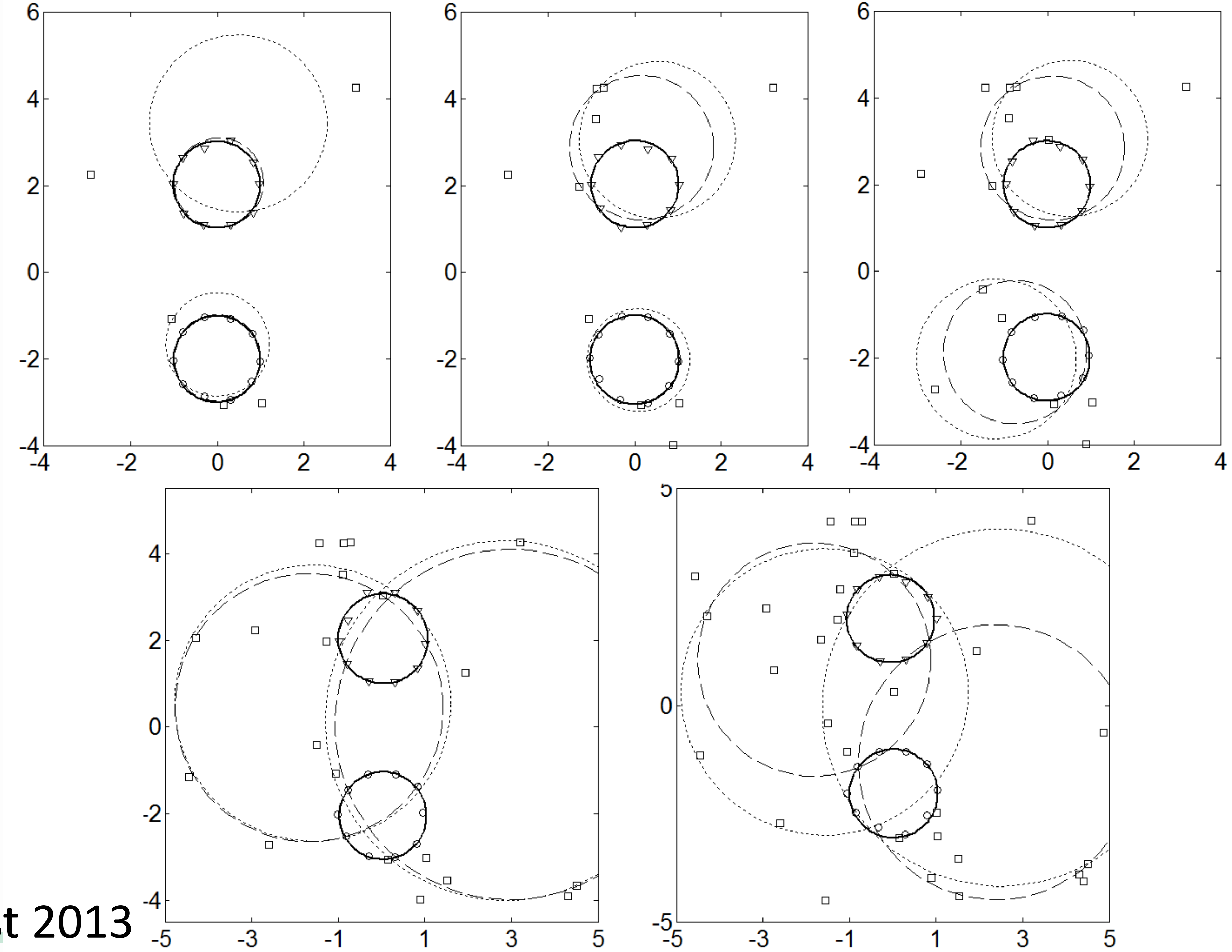
- IRIS DATA: 150 vectors, 4 dimensions, 3 classes, one additional outlier (δ,δ,δ,δ)

Circum-stances	IRIS type	FCM			PFCM			FP3CM			Correct decisions
		$\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	
no outlier added	Setosa	50	0	0	50	0	0	50	0	0	FCM $\rightarrow$ 136
	Versicolor	0	47	3	0	47	3	0	48	2	PFCM $\rightarrow$ 136
	Virginica	0	11	39	0	11	39	0	7	43	FP3CM $\rightarrow$ 141
outlier added at 20	Setosa	50	0	0	50	0	0	50	0	0	FCM $\rightarrow$ 134
	Versicolor	0	50	0	0	50	0	0	47	3	PFCM $\rightarrow$ 135
	Virginica	0	16	34	0	15	35	0	7	43	FP3CM $\rightarrow$ 140
outlier added at 30	Setosa	50	0	0	50	0	0	50	0	0	FCM $\rightarrow$ 128
	Versicolor	1	49	0	1	49	0	0	47	3	PFCM $\rightarrow$ 131
	Virginica	0	21	29	0	18	32	0	7	43	FP3CM $\rightarrow$ 140
outlier added at 50 or 10^6	Setosa	50	0	0	50	0	0	50	0	0	FCM crashes
	Versicolor	3	47	0	3	47	0	0	47	3	PFCM crashes
	Virginica	0	50	0	0	50	0	0	7	43	FP3CM $\rightarrow$ 140

FPPP based detection of circles (spheroids)

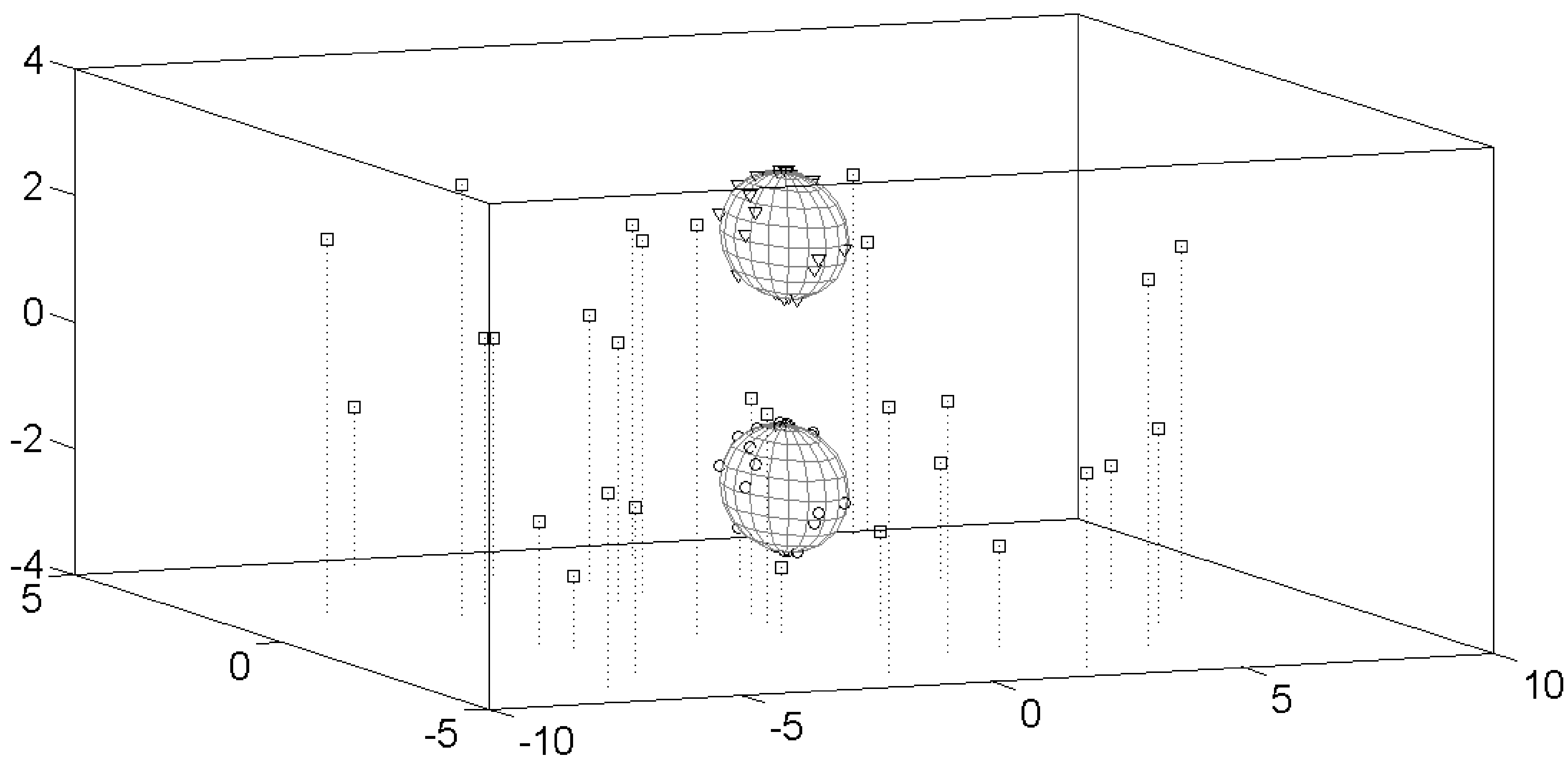


Two circles and more outliers



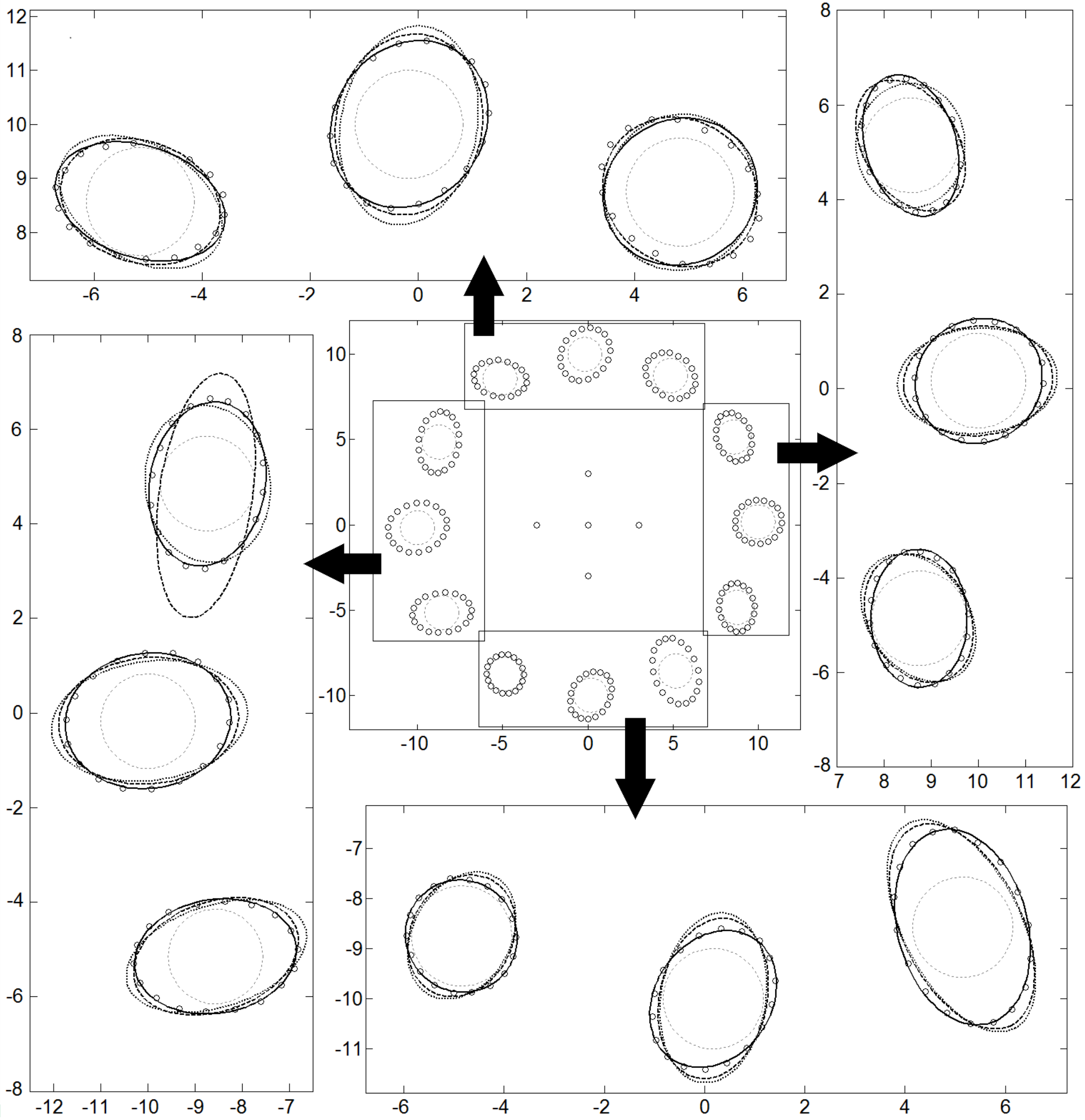
Szilágyi, Int J Intell Syst 2013

3D example



Szilágyi, Int J Intell Syst 2013

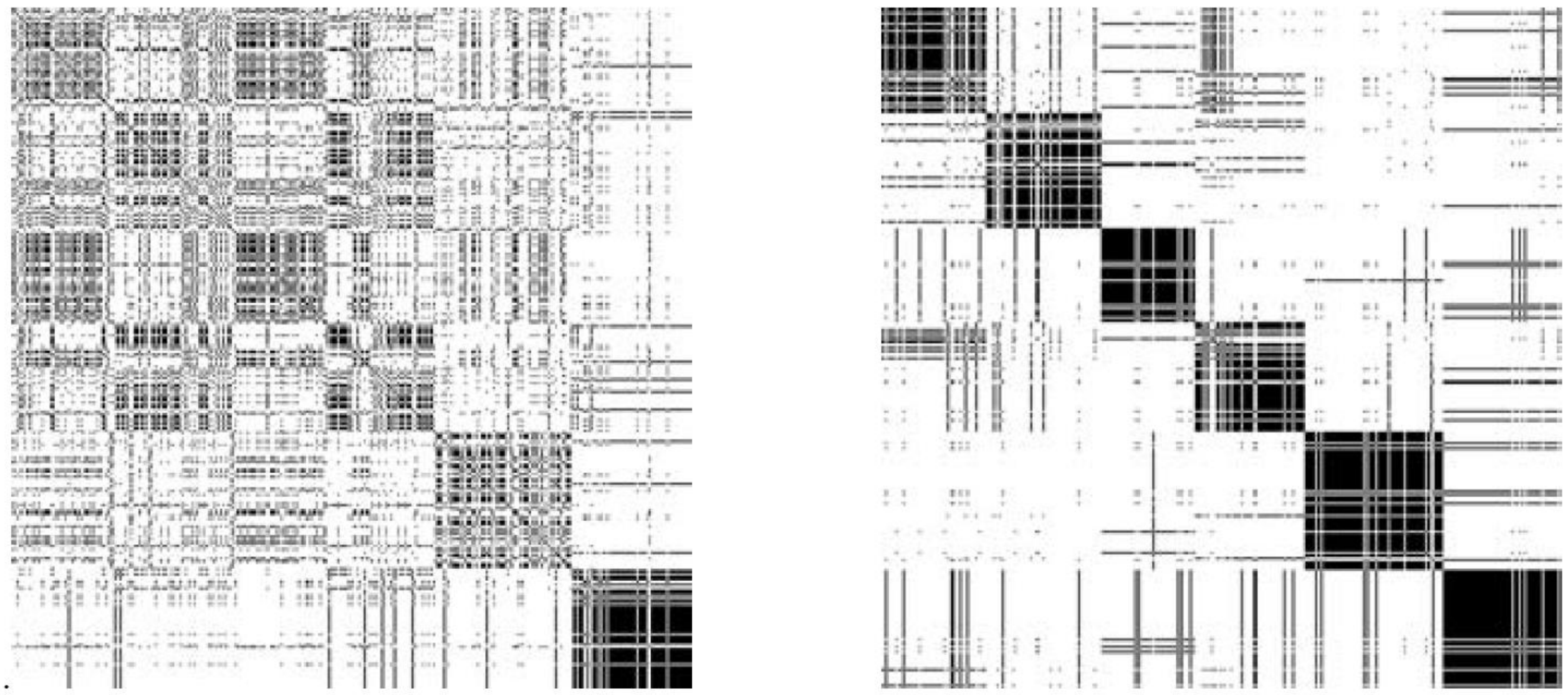
Detection of ellipses in the presence of outliers



Szilágyi et al, MDAI 2014

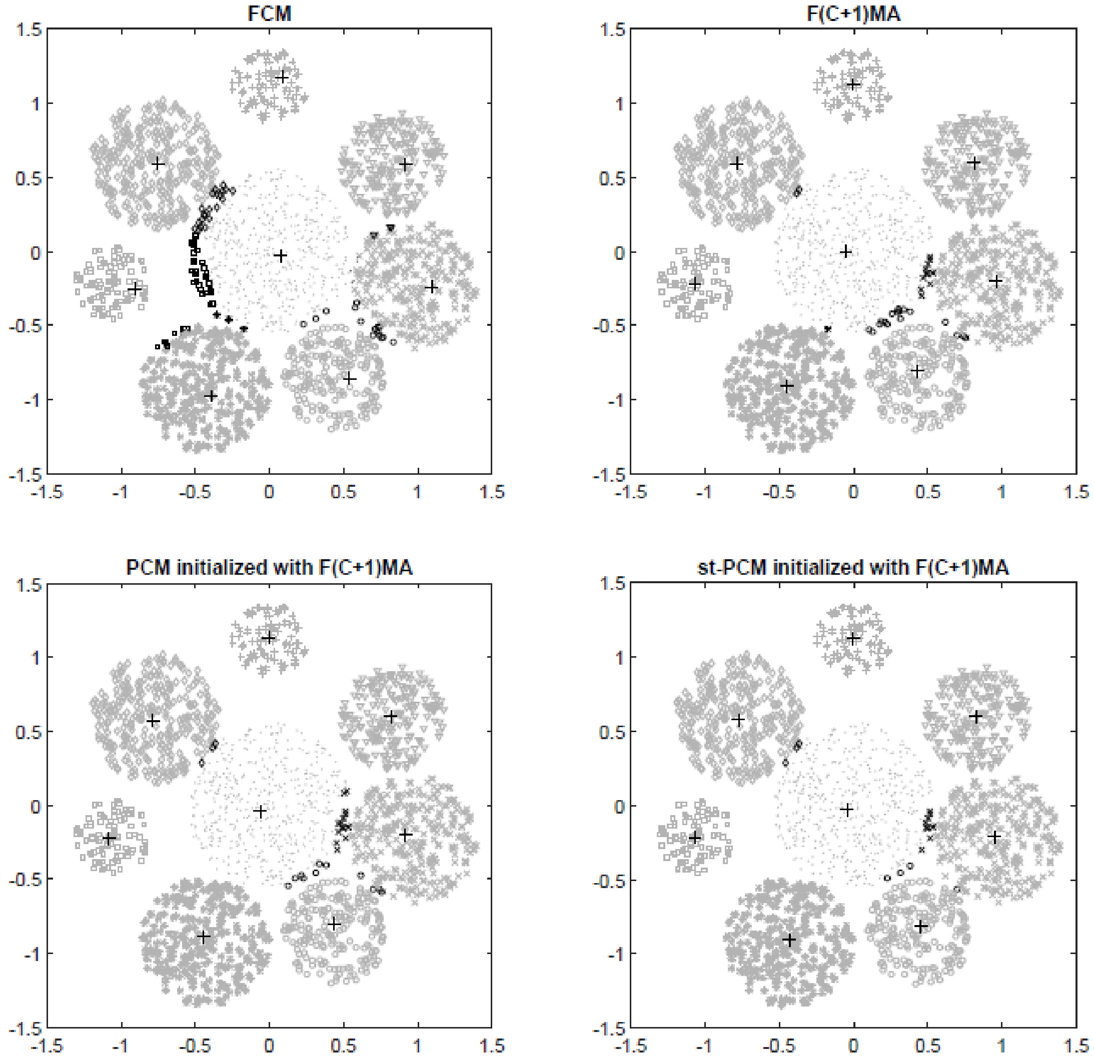
Real-life application

- Gosztolya & Szilágyi, Acta Polytech Hung 2015
- Blind speaker clustering, 6 speakers
- Confusion matrix: classical approach vs. FPPP based approach

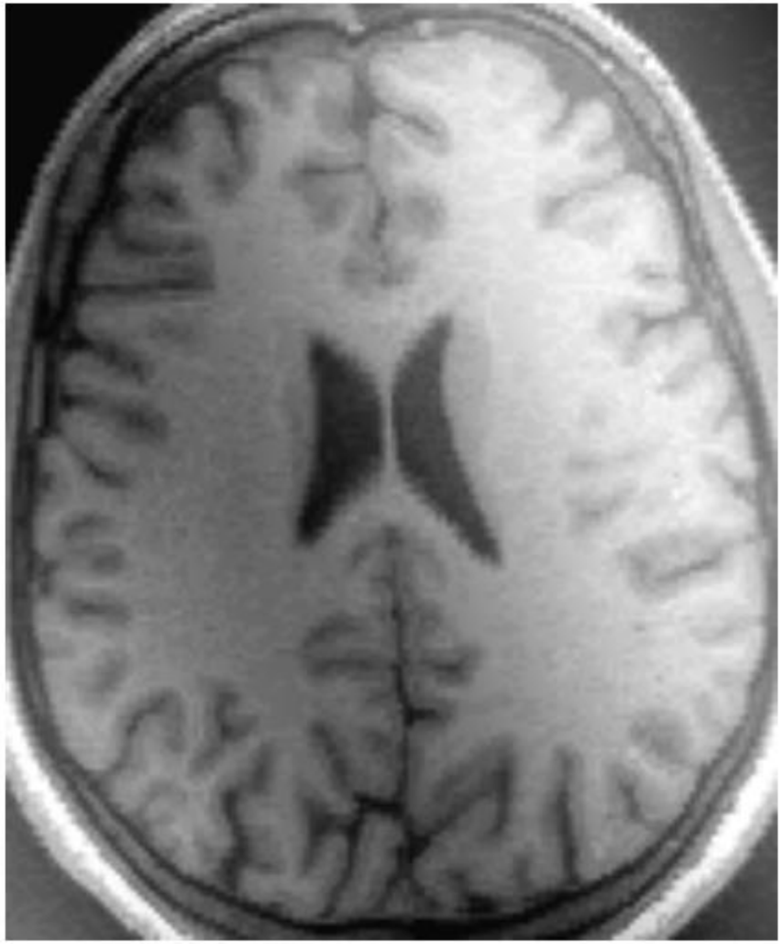


Self-tuning possibilistic c-means

- Szilágyi et al, Int J Fuzz Uncert Knowl Based Syst, 2019
- Combines
  - Possibilistic c-means
  - Cluster size regulatory variable
- Initialized by (c+1)-means



Intensity non-uniformity compensation and segmentation of MRI data



- MRI: the same tissue can be represented by different intensity values
- INU: noise of low frequency but high amplitude

- Additive noise model: $x_k = y_k - b_k$

- x_k: real intensity of pixel k
- y_k: observed intensity of pixel k
- b_k: estimated noise at pixel k

Noise compensation with c-means clustering

- Objective function:

$$J_{\mathrm{FCM-b}} = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^{m}(y_k - b_k - v_i)^2$$

$$J_{\mathrm{FCM-qb}} = \sum_{i=1}^{c}\sum_{l\in\Omega^{(t)}} H_l^{(t)} u_{il}^{m}(l - v_i)^2$$

- Partition

$$u_{ik} = \frac{(y_k - b_k - v_i)^{-2/(m-1)}}{\sum_{j=1}^{c}(y_k - b_k - v_j)^{-2/(m-1)}} \qquad i = 1\ldots c \quad k = 1\ldots n$$

$$u_{il} = \frac{(l - v_i)^{-2/(m-1)}}{\sum_{j=1}^{c}(l - v_j)^{-2/(m-1)}} \qquad i = 1\ldots c \quad l \in \Omega^{(t)}$$

- Cluster prototypes

$$v_i = \frac{\sum_{k=1}^{n} u_{ik}^{m}(y_k - b_k)}{\sum_{k=1}^{n} u_{ik}^{m}} \qquad i = 1\ldots c$$

$$v_i = \frac{\sum_{l\in\Omega^{(t)}} H_l^{(t)} u_{il}^{m} l}{\sum_{l\in\Omega^{(t)}} H_l^{(t)} u_{il}^{m}} \qquad i = 1\ldots c$$

- Estimated noise:

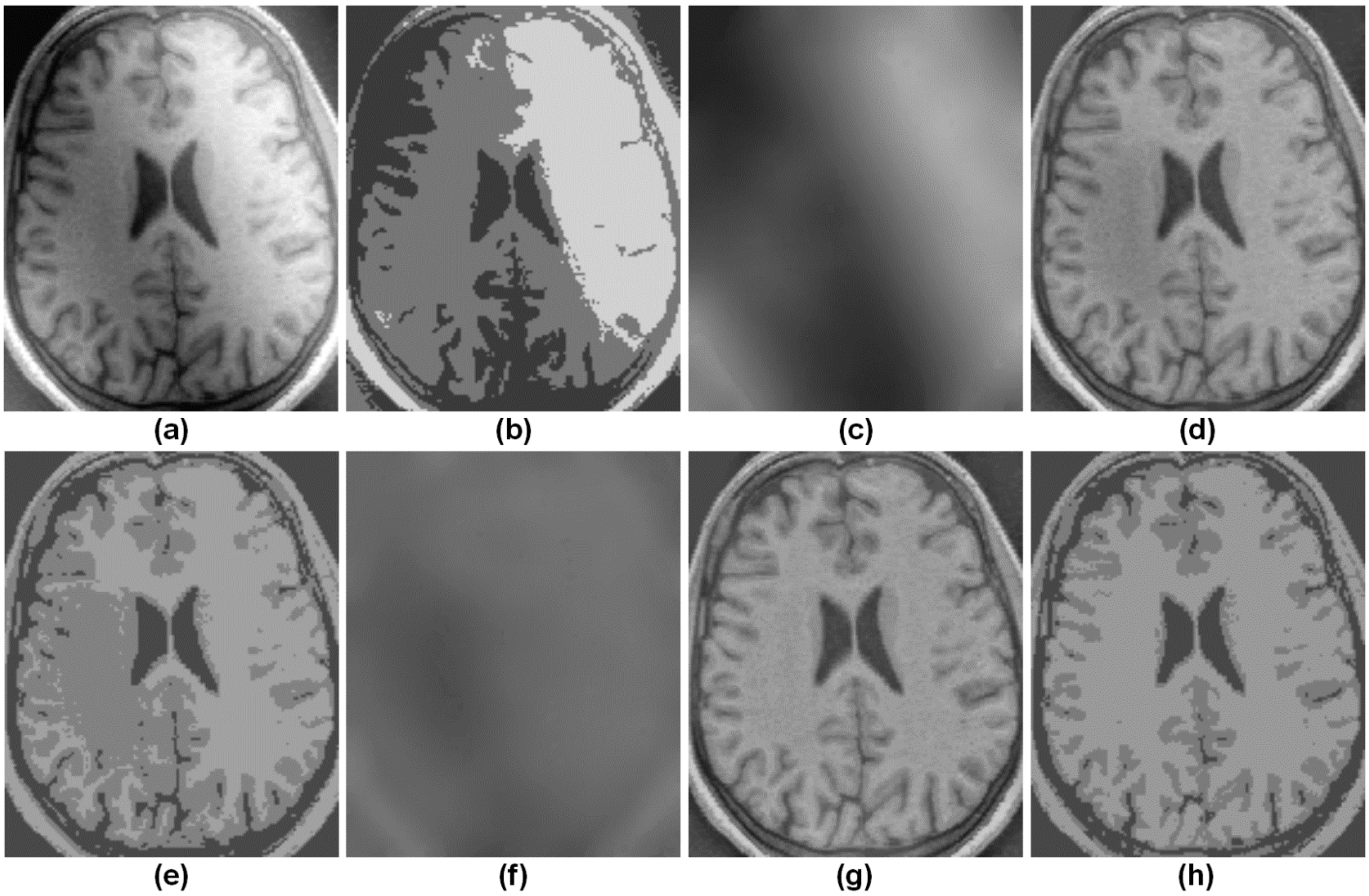
$$b_k = y_k - \frac{\sum_{i=1}^{c} u_{ik}^{m} v_i}{\sum_{i=1}^{c} u_{ik}^{m}} \qquad k = 1\ldots n$$

$$b_k = y_k - q_{l_k} \qquad l_k = y_k - b_k^{(t-1)}$$

$$q_l = \frac{\sum_{i=1}^{c} u_{il}^{m} v_i}{\sum_{i=1}^{c} u_{il}^{m}} \qquad k = 1\ldots n \quad l \in \Omega^{(t)}$$

Szilágyi et al, CMPB 2012

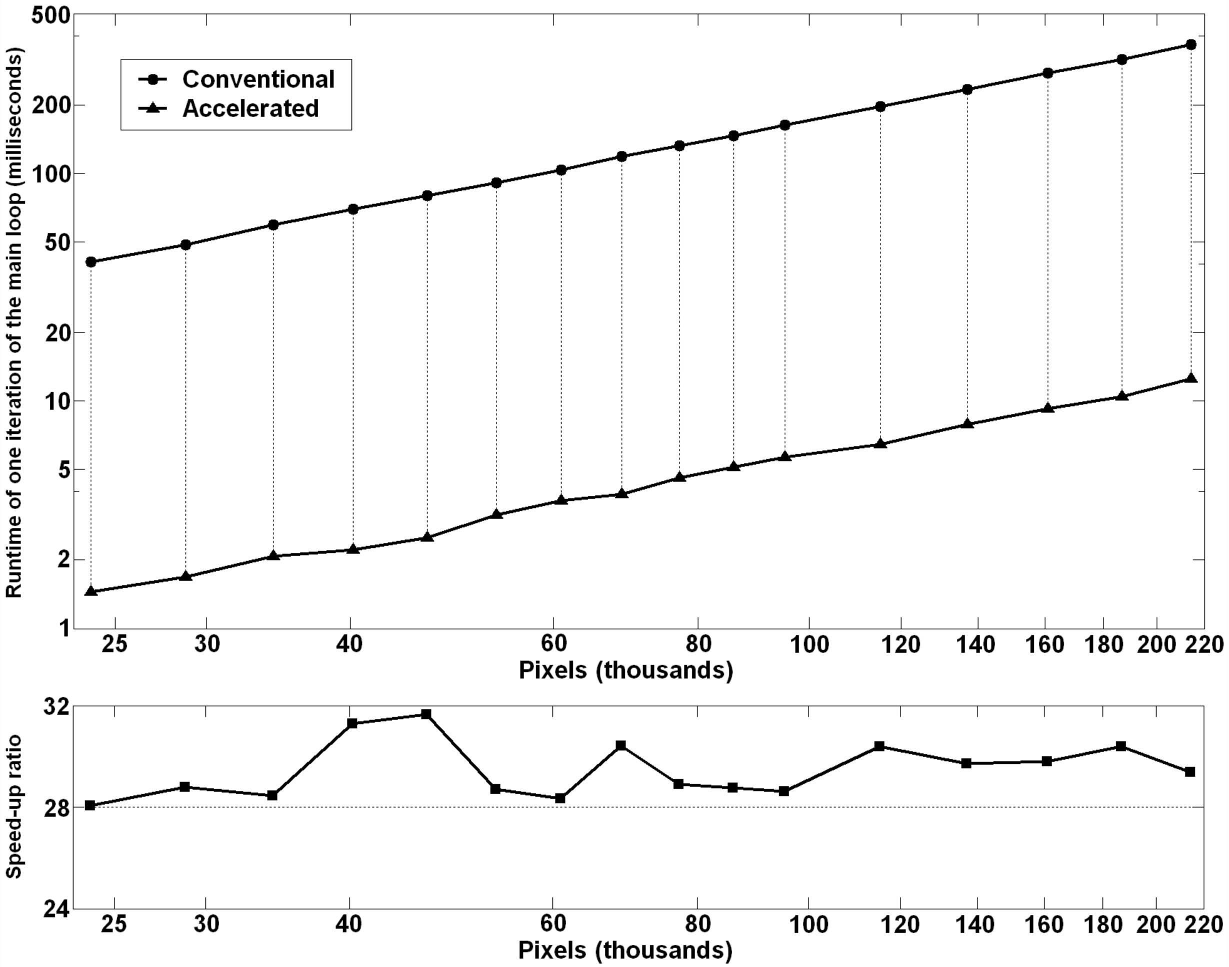
Results

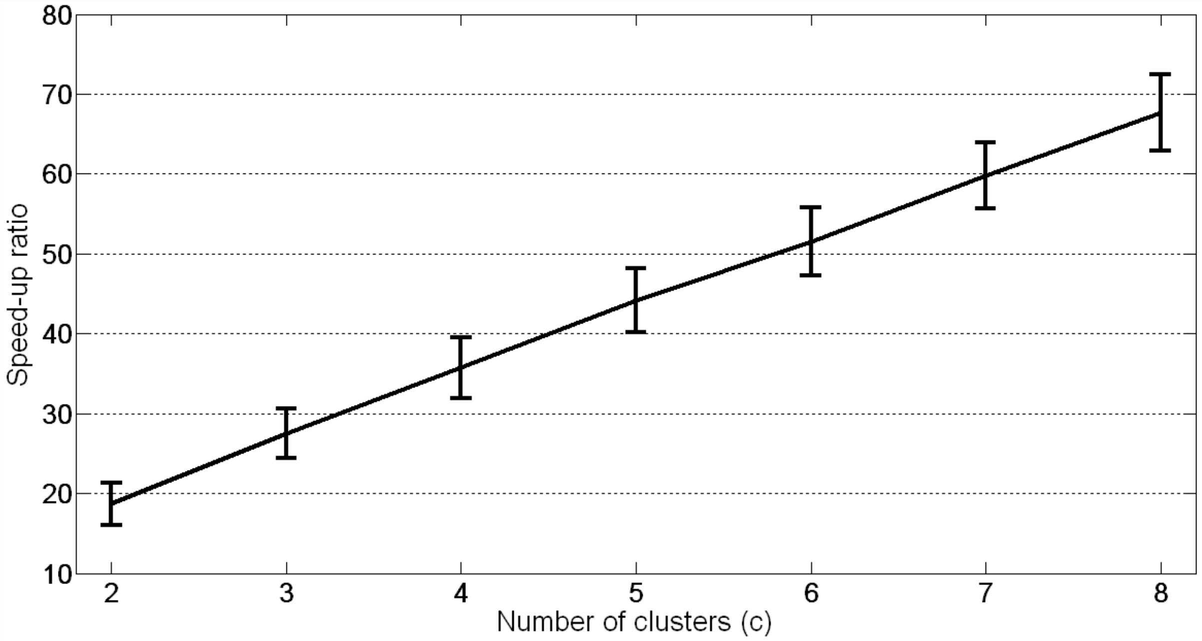


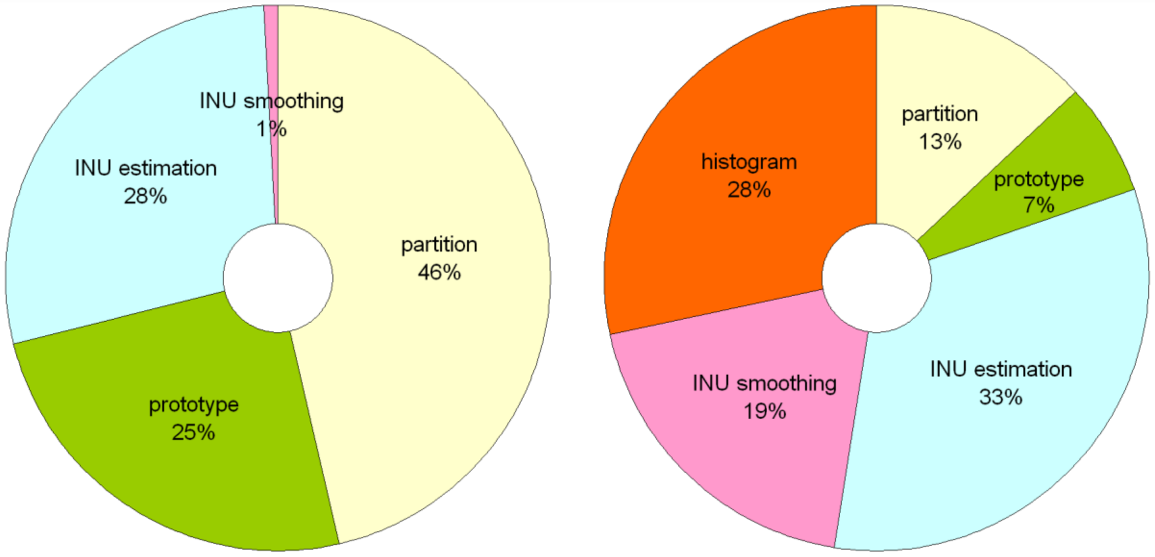
Algorithm complexity

Algorithmic step	Conventional (FCM-b)	Accelerated (FCM-qb)
Partition updating	$\mathcal{O}(nc^2)$	$\mathcal{O}(\omega c^2)$
Cluster prototype updating	$\mathcal{O}(nc)$	$\mathcal{O}(\omega c)$
Bias estimation	$\mathcal{O}(nc)$	$\mathcal{O}(n + \omega c)$
Bias smoothing	$\mathcal{O}(n)$	$\mathcal{O}(n)$
Histogram updating	–	$\mathcal{O}(n)$

Runtime





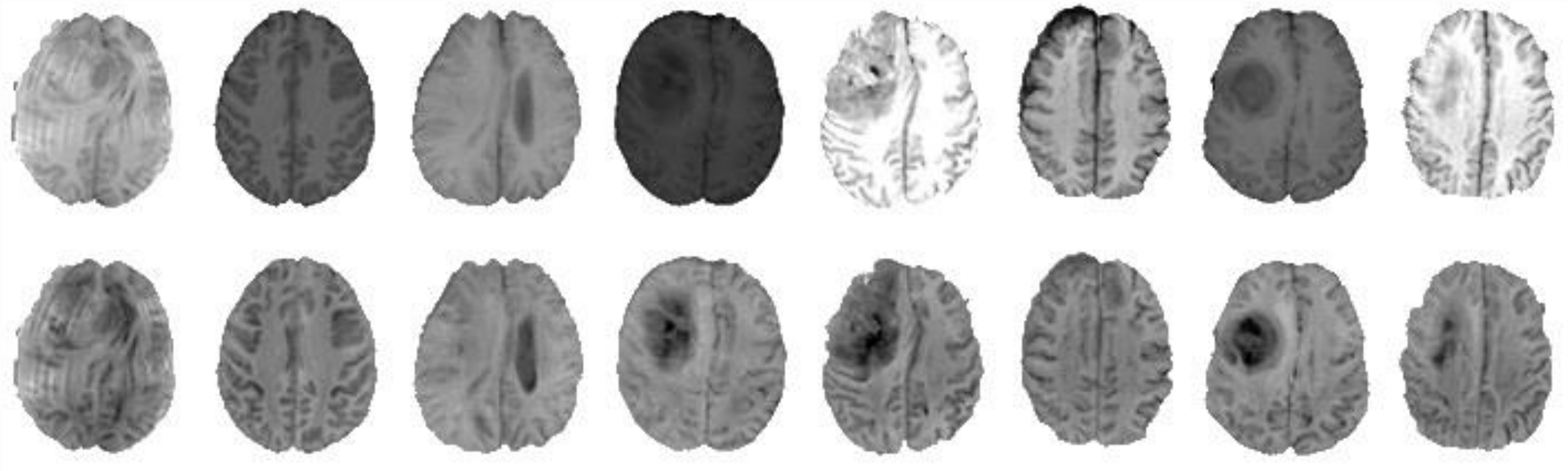


Conventional FCM-based INU compensation

Accelerated FCM-based INU compensation

Histogram normalization in MRI

- There is no absolute scale in MRI data
- Intensity values must be interpreted together with their context



Existing methods

- Nyúl et al (2000) – cited by 782, piecewise linear transform
- Leung et al (2010) – cited by 133, segmentation + tissue based alignment
- Weisenfeld et al (2004) – cited by 77, Kullback-Leibler divergence
- Shinohara et al (2011) – cited by 45, PCA
- Jäger et al (2006) – cited by 32, hidden Markov random fields
- Linear transform

Brain tissue and brain tumor segmentation

- Widely used method: Nyúl et al (2000)
  - Piecewise linear transform established via matching predefined milestones of the histograms
  - Most papers use it without saying any details of the chosen milestones
  - Some papers (Soltaninejad 2018, Pinto 2018) say they use 10-12 milestones
  - Some papers (Tustison 2015) say that a linear transform led to better segmentation
- Question: which method is more accurate?

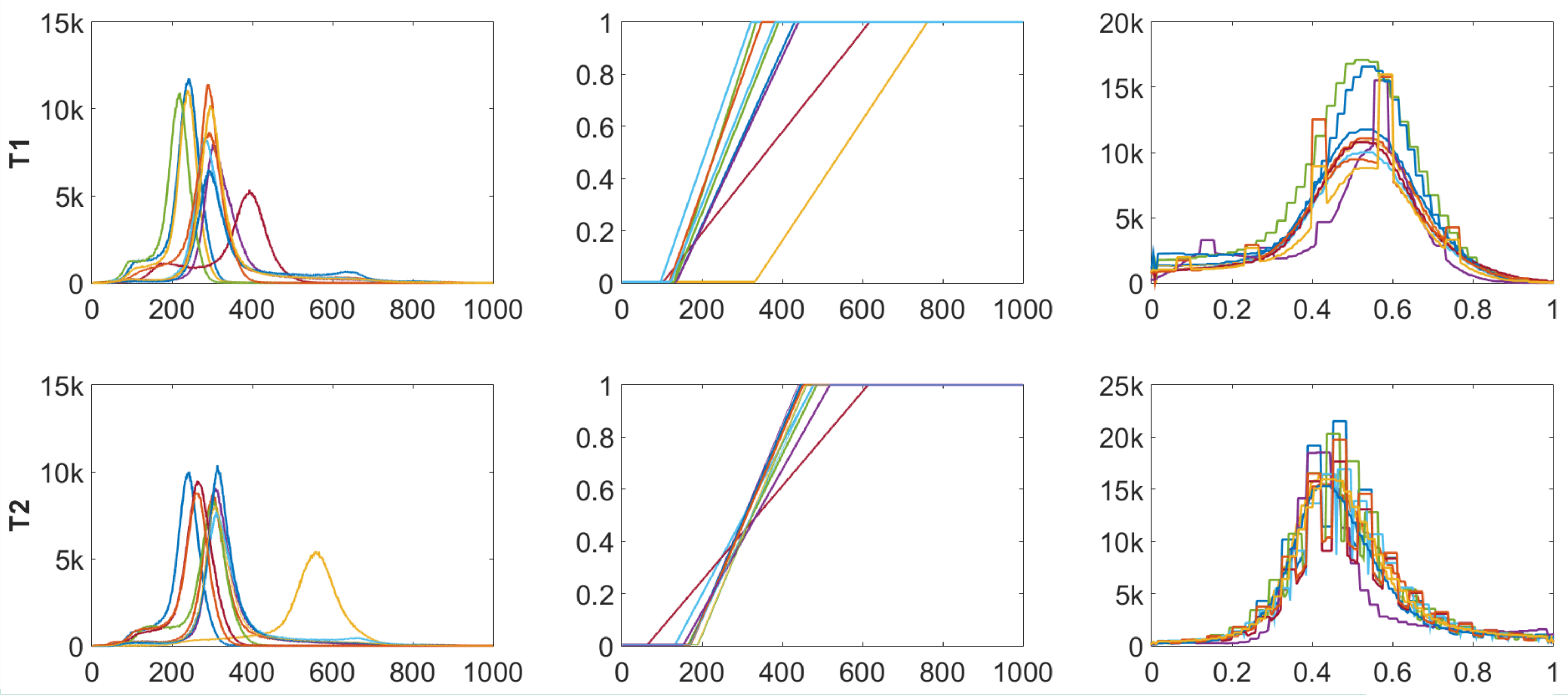
Input Data

- Medical Image Computation and Computer Aided Interventions (MICCAI)
- Brain Tumor Segmentation Challenge (BraTS) since 2012
- BraTS train dataset 2019
  - 76 low-grade (LG) and 259 high-grade (HG) volumes
- Multispectral (T1, T2, T1C, FLAIR)
- 155 x 240 x 240 image voxels
- Ground truth (GT): negative, enhancing core, tumor core, edema
- Skull removed
- This study uses 50 selected LG volumes

- 6-month infant brain Segmentation Challenge (iSeg-2017, iSeg-2019)
- iSeg-2017 train dataset
  - 10 volumes
- Multispectral (T1, T2)
- 256 x (144 x 192) image voxels
- Ground truth: cerebro-spinal fluid (CSF), grey matter (GM), white matter (WM)
- Skull removed
- This study uses all 10 volumes of iSeg-2017 train dataset

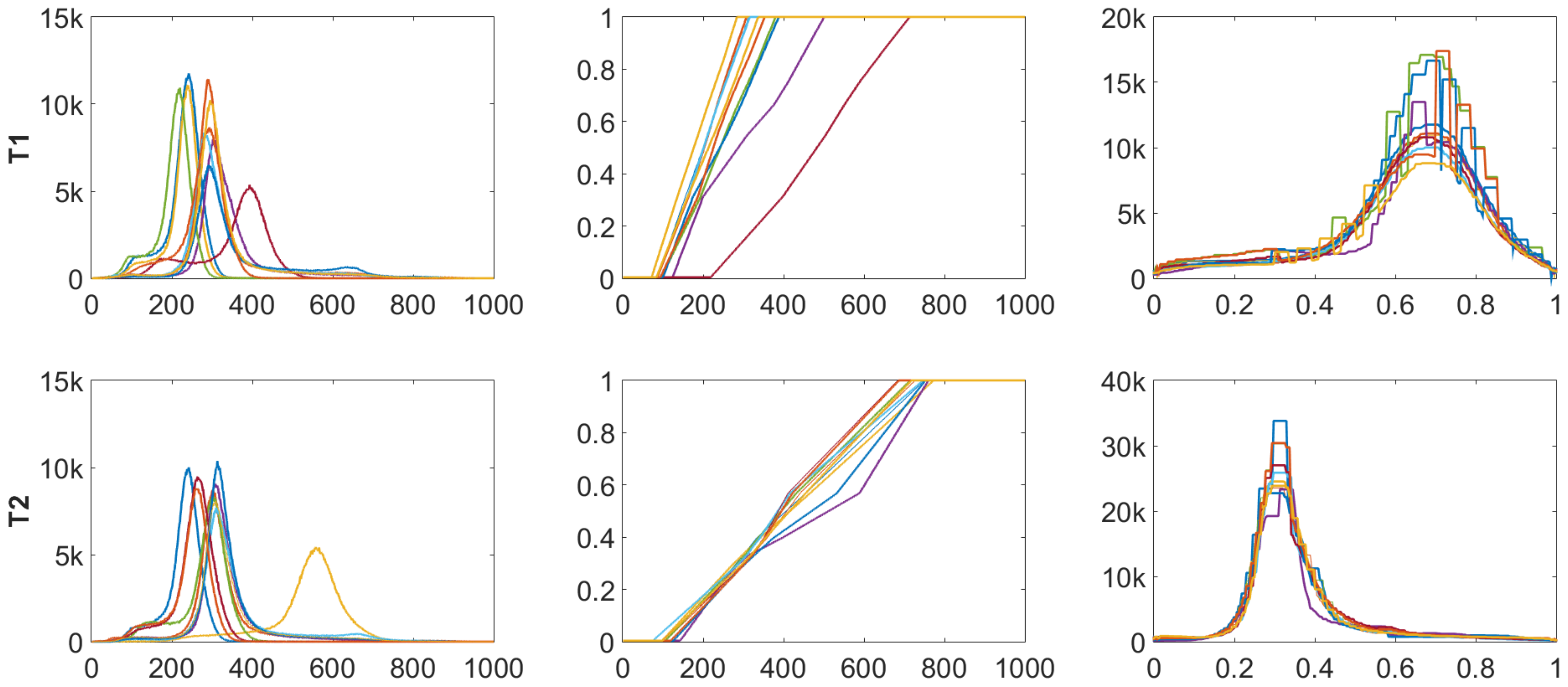
Algorithm A1 – Linear transform

- Input histograms Linear transforms Output histograms



Algorithm A2 – Method of Nyúl et al (2000)

- Input histograms Piecewise linear transforms Output histograms

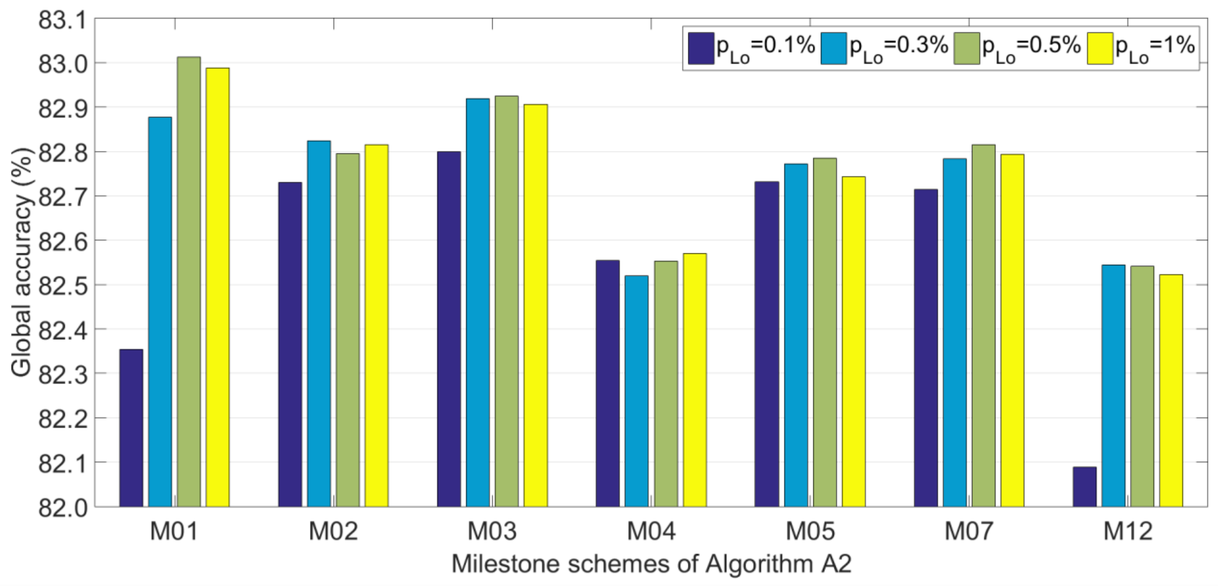


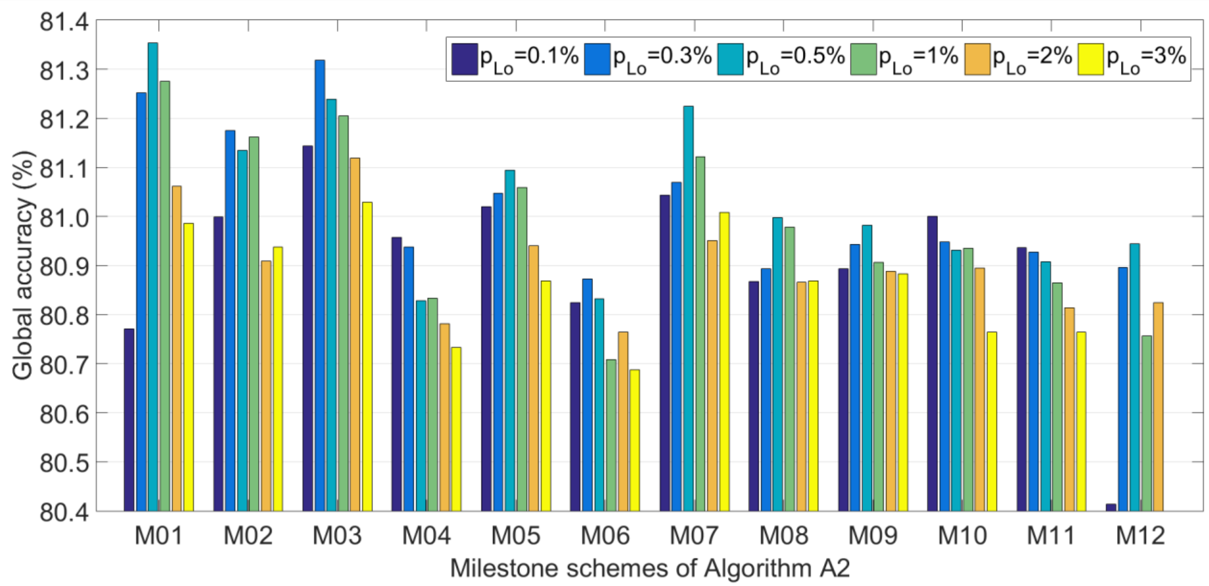
Parameters

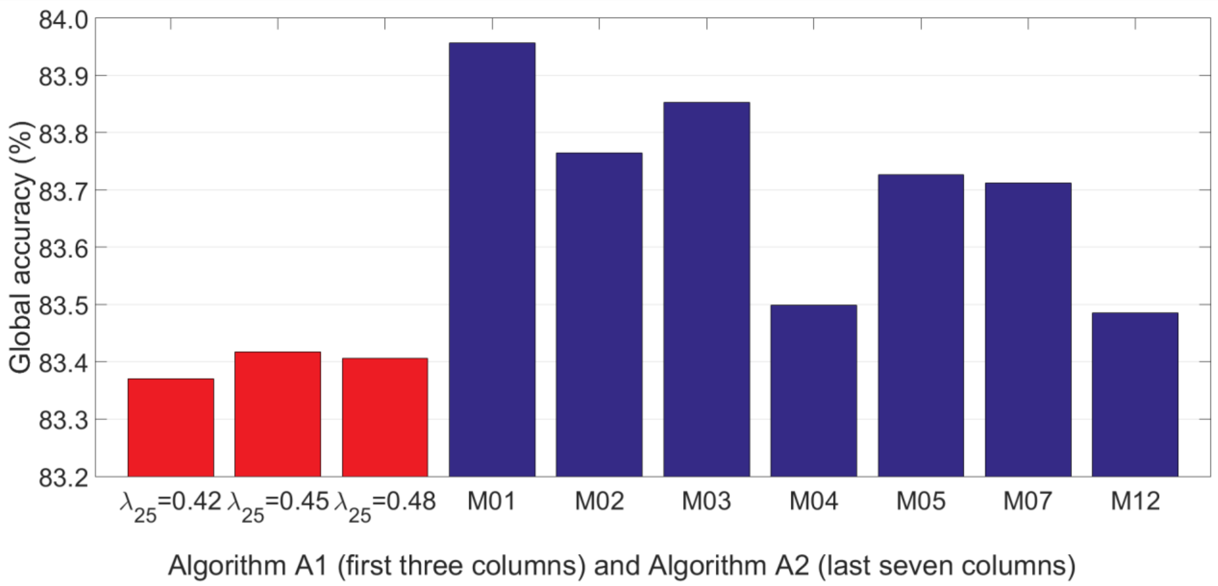
- Linear transform (Algorithm A1)
  - Single parameter $\lambda_{25} \in [0.3, 0.5)$, percentile $p_{25} \rightarrow \lambda_{25}$ and $p_{75} \rightarrow (1 - \lambda_{25})$
- Nyúl et al (2000) (Algorithm A2)
  - Parameter $p_{Lo} < 0.03$, $p_{Hi} = 1 - p_{Lo}$ define the tails of the input histogram to be cut
  - Set of milestones defined as percentiles to be matched in all histograms

Scheme	Landmark points
M01	$p_{\mathrm{Lo}}, p_{50}, p_{\mathrm{Hi}}$
M02	$p_{\mathrm{Lo}}, p_{25}, p_{75}, p_{\mathrm{Hi}}$
M03	$p_{\mathrm{Lo}}, p_{25}, p_{50}, p_{75}, p_{\mathrm{Hi}}$
M04	$p_{\mathrm{Lo}}, p_{10}, p_{50}, p_{90}, p_{\mathrm{Hi}}$
M05	$p_{\mathrm{Lo}}, p_{20}, p_{40}, p_{60}, p_{80}, p_{\mathrm{Hi}}$
M06	$p_{\mathrm{Lo}}, p_{10}, p_{25}, p_{75}, p_{90}, p_{\mathrm{Hi}}$
M07	$p_{\mathrm{Lo}}, p_{20}, p_{35}, p_{50}, p_{65}, p_{80}, p_{\mathrm{Hi}}$
M08	$p_{\mathrm{Lo}}, p_{10}, p_{25}, p_{50}, p_{75}, p_{90}, p_{\mathrm{Hi}}$
M09	$p_{\mathrm{Lo}}, p_{10}, p_{25}, p_{40}, p_{60}, p_{75}, p_{90}, p_{\mathrm{Hi}}$
M10	$p_{\mathrm{Lo}}, p_{10}, p_{25}, p_{40}, p_{50}, p_{60}, p_{75}, p_{90}, p_{\mathrm{Hi}}$
M11	$p_{\mathrm{Lo}}, p_{10}, p_{20}, p_{30}, p_{40}, p_{60}, p_{70}, p_{80}, p_{90}, p_{\mathrm{Hi}}$
M12	$p_{\mathrm{Lo}}, p_{10}, p_{20}, p_{30}, p_{40}, p_{50}, p_{60}, p_{70}, p_{80}, p_{90}, p_{\mathrm{Hi}}$

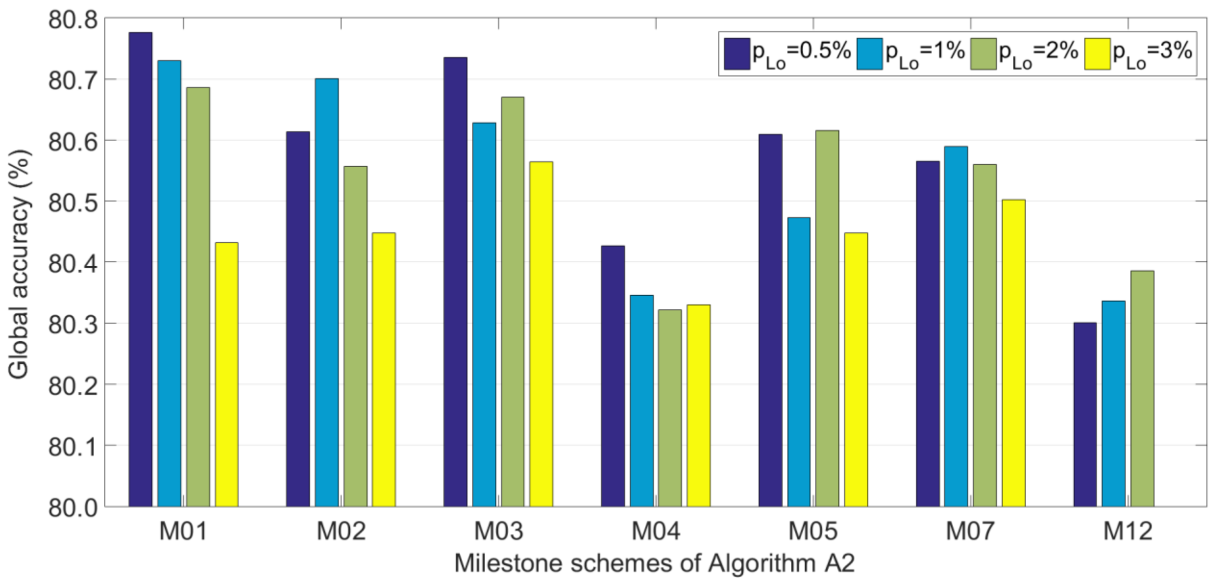
Algorithm A2
Random forest
iSeg-2017 data

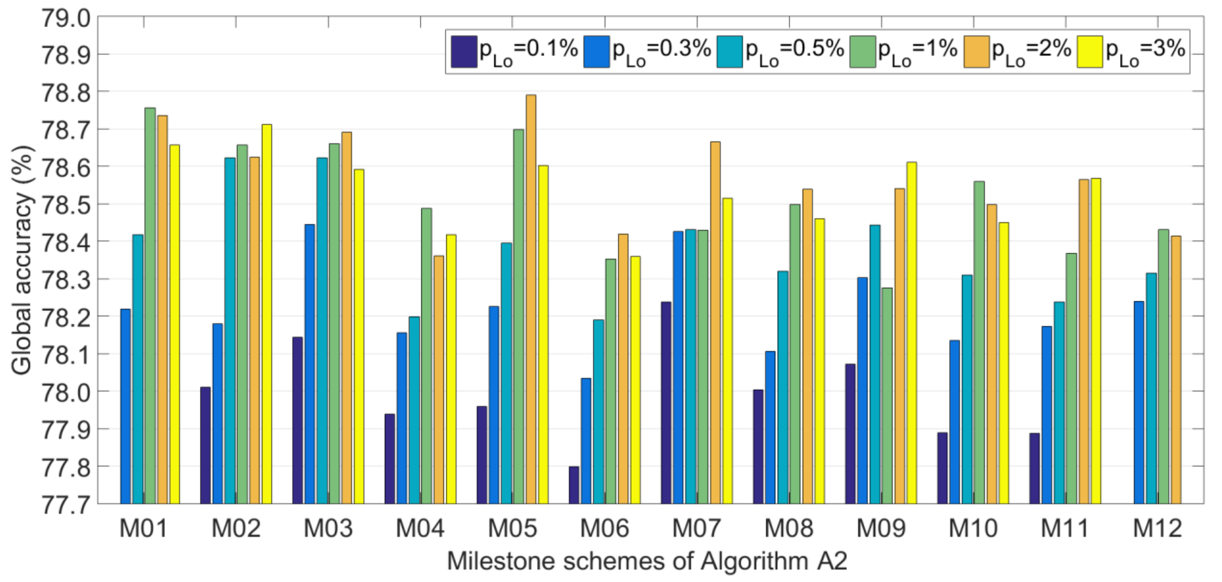


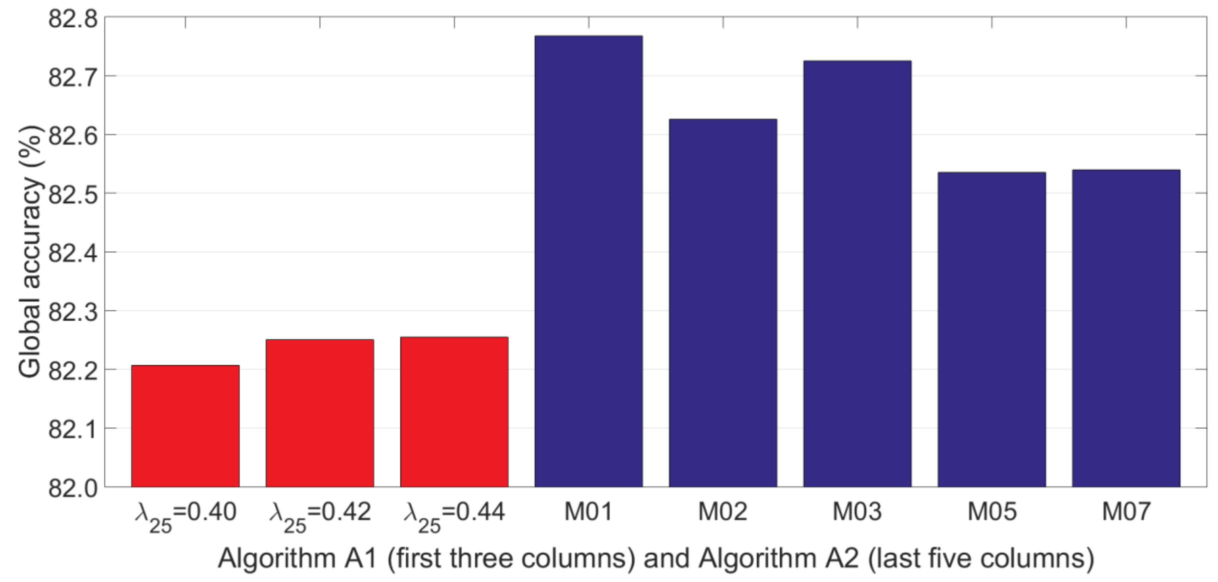




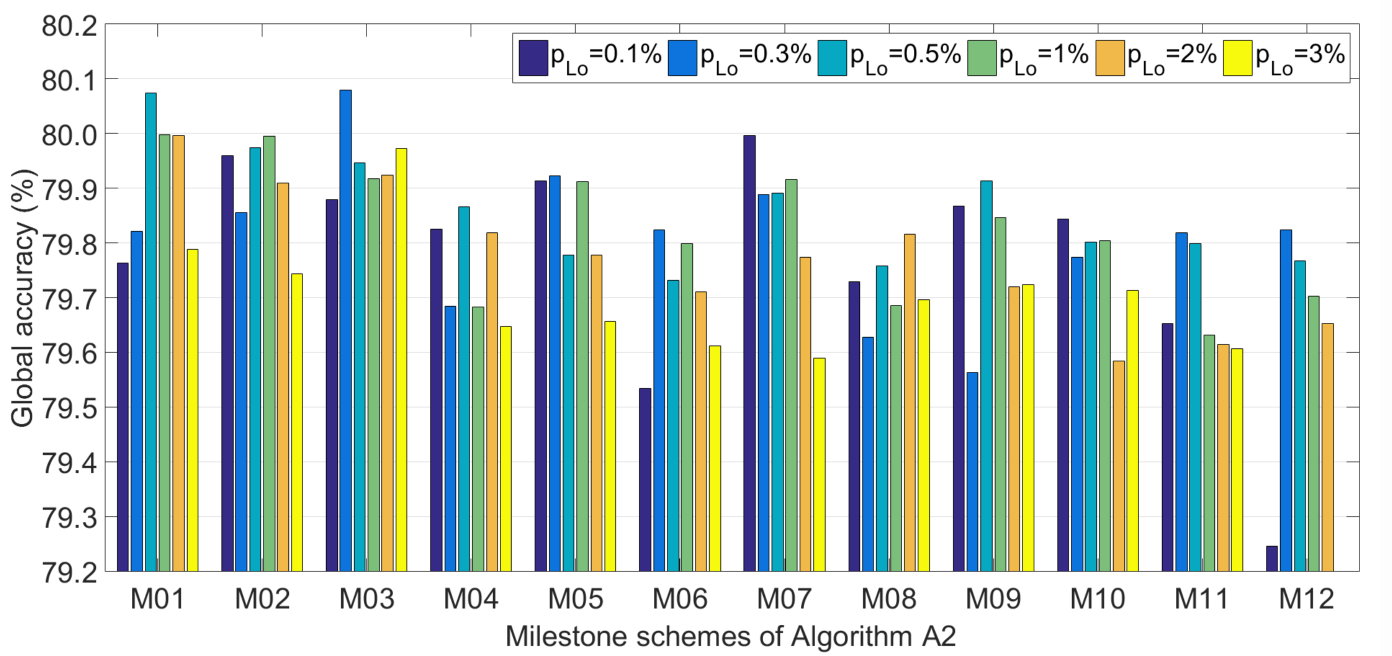
Algorithm A2
KNN
iSeg-2017 data



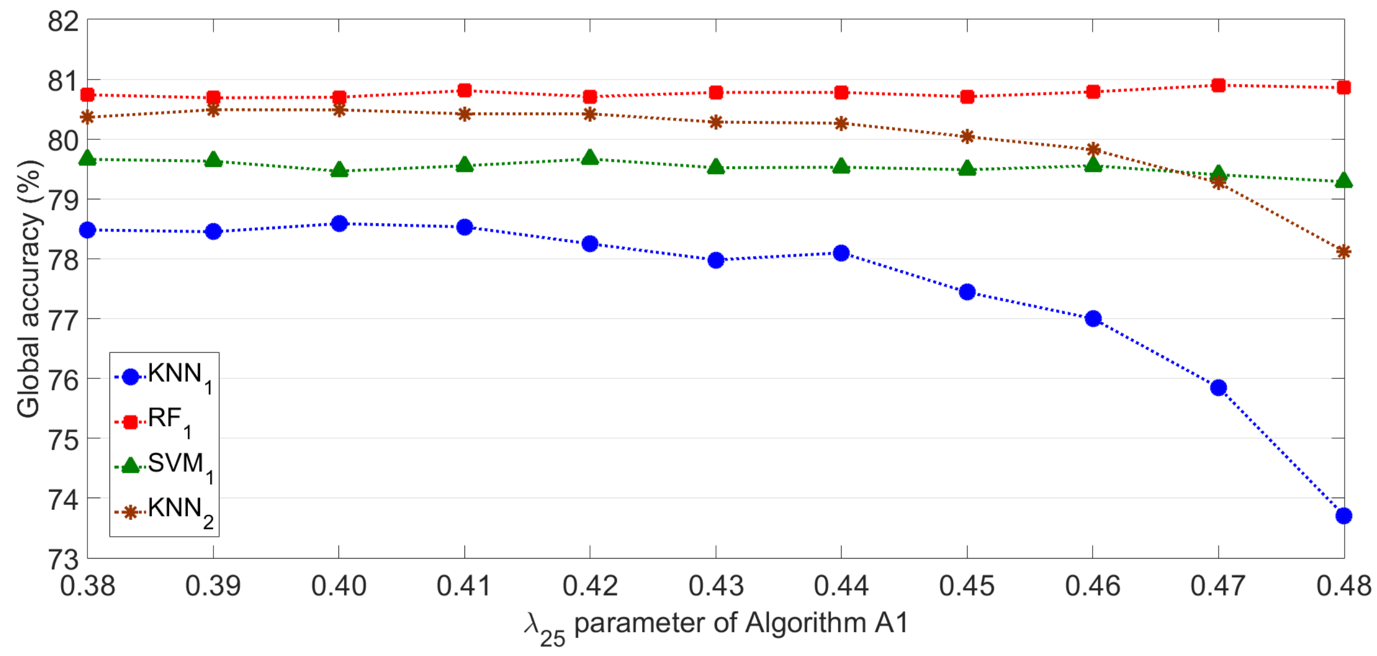




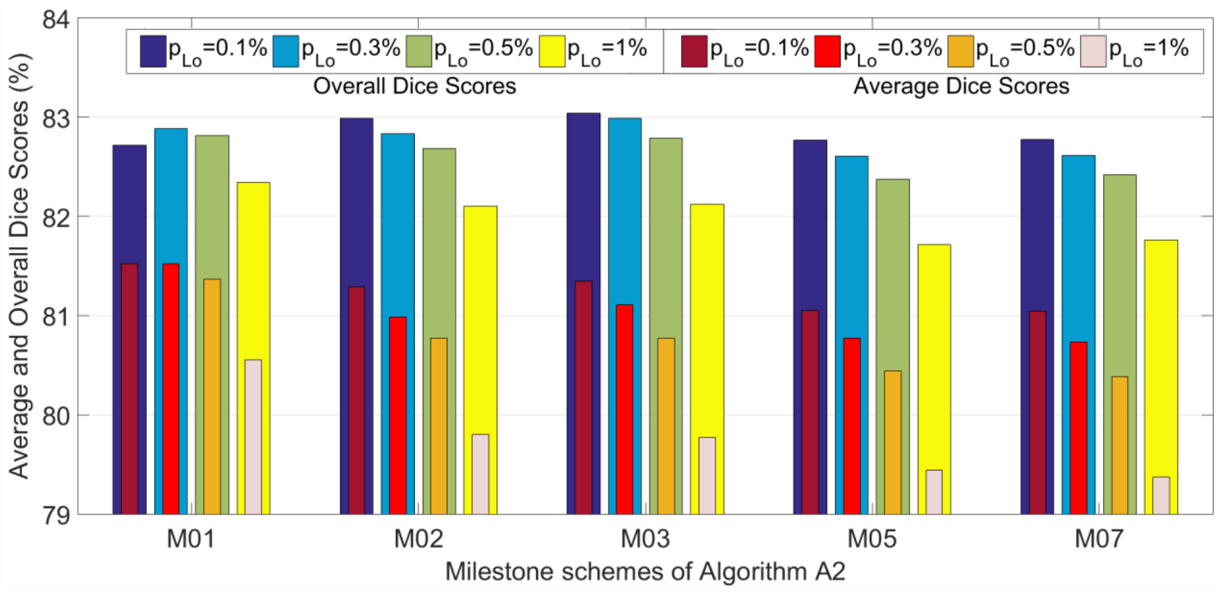
Algorithm A2
SVM
iSeg-2017 data

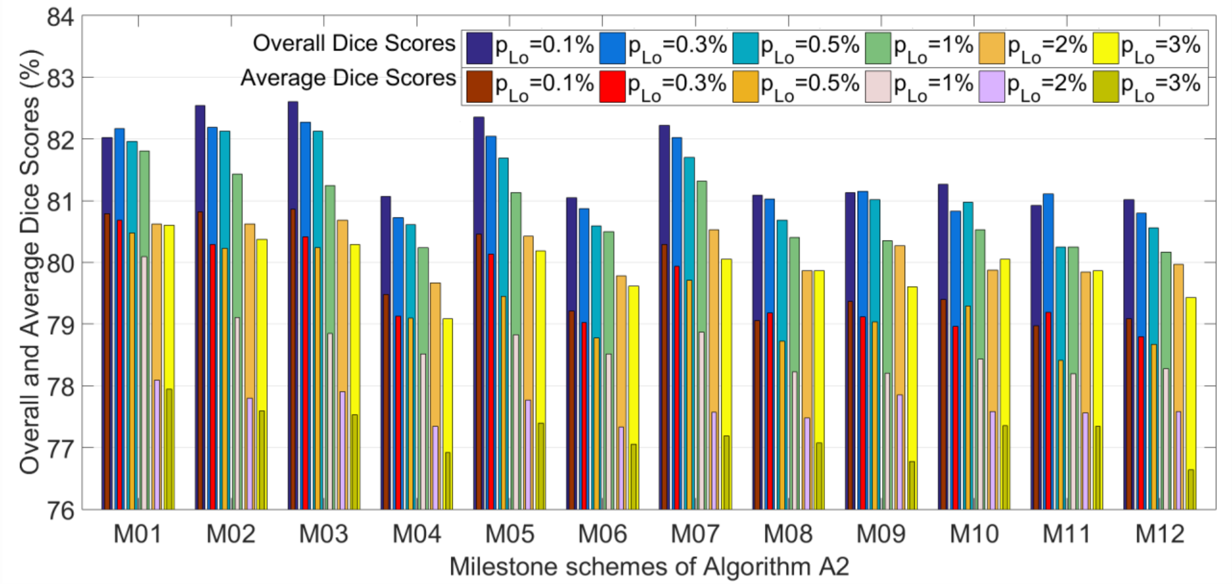


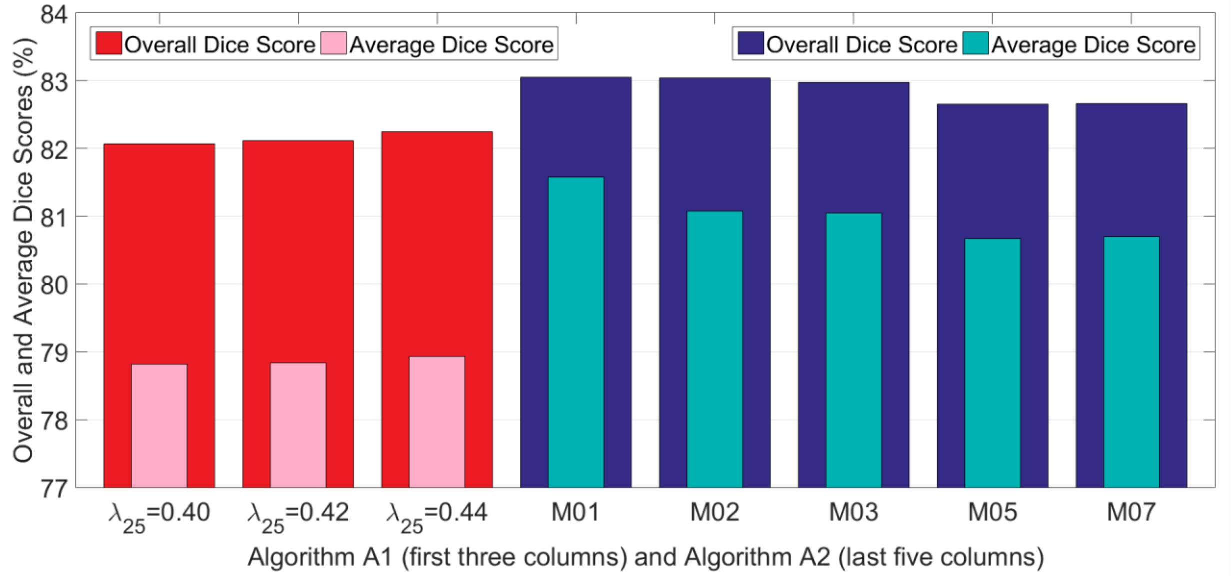
Algorithm A1
RF, KNN, SVM
iSeg-2017 data



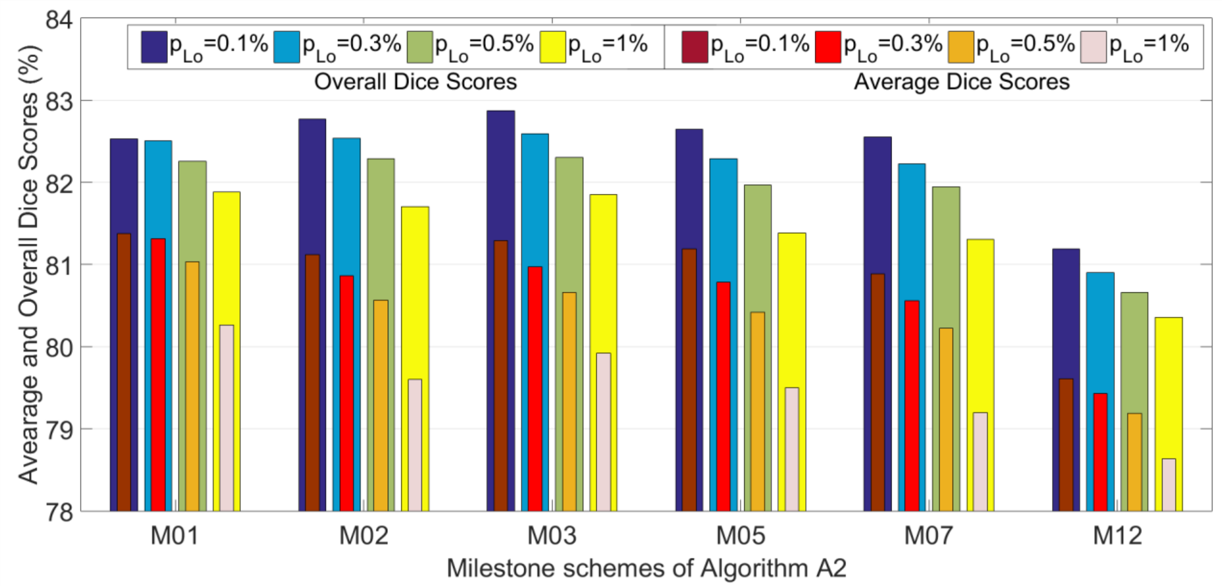
Algorithm A2
Random forest
BraTS 2019 data

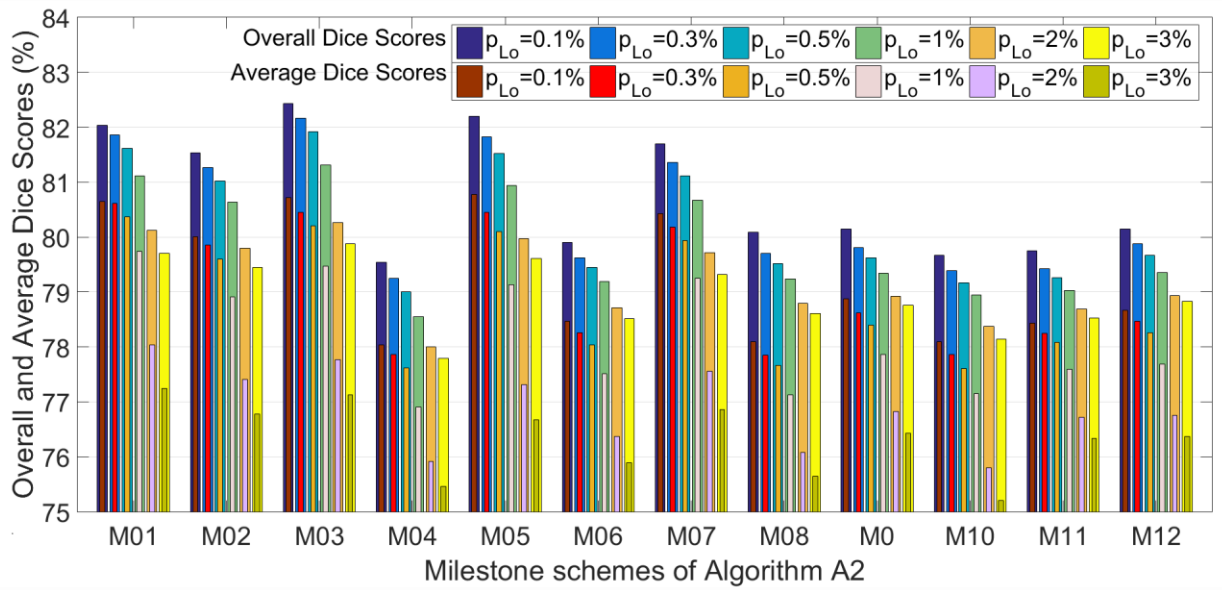


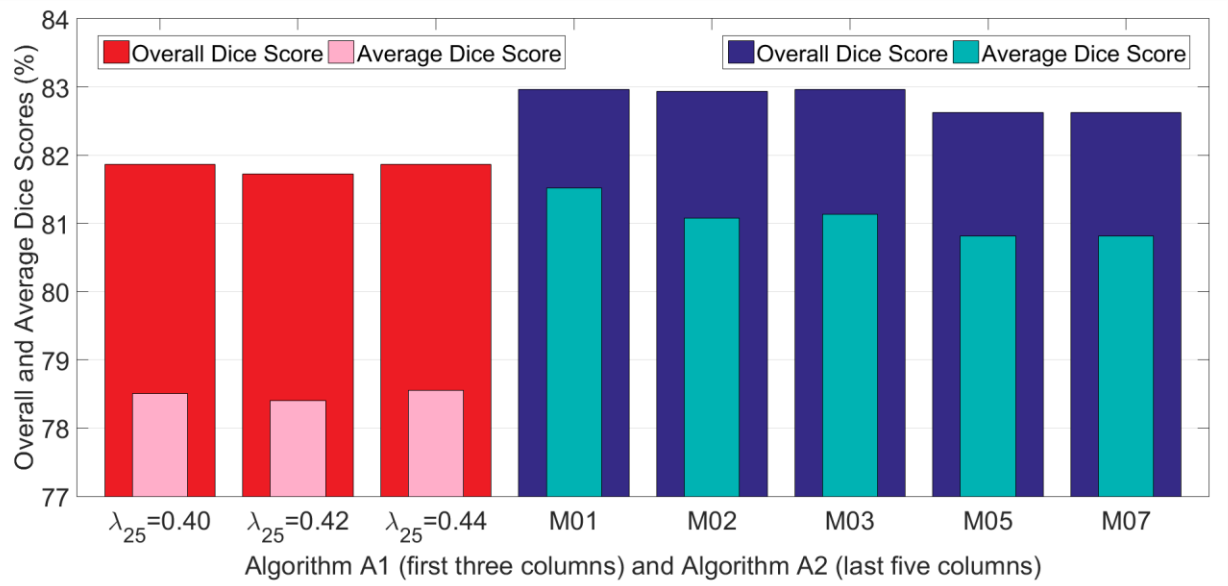




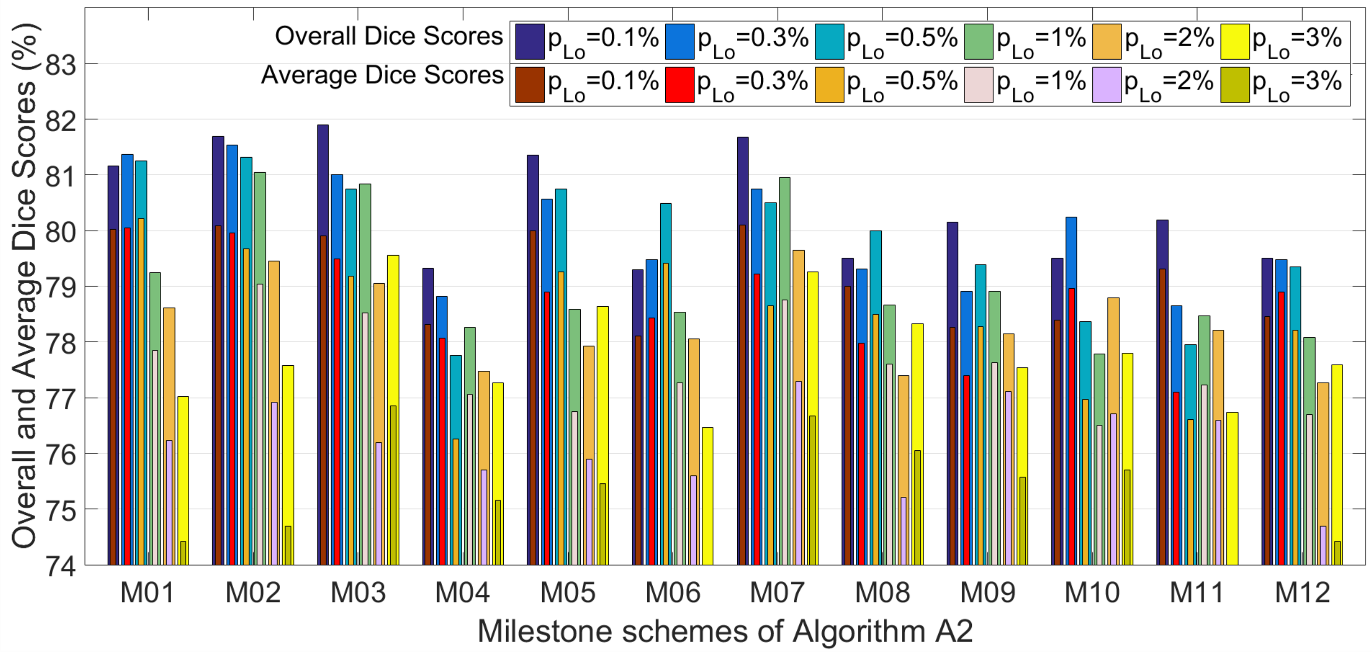
Algorithm A2
KNN
BraTS 2019 data



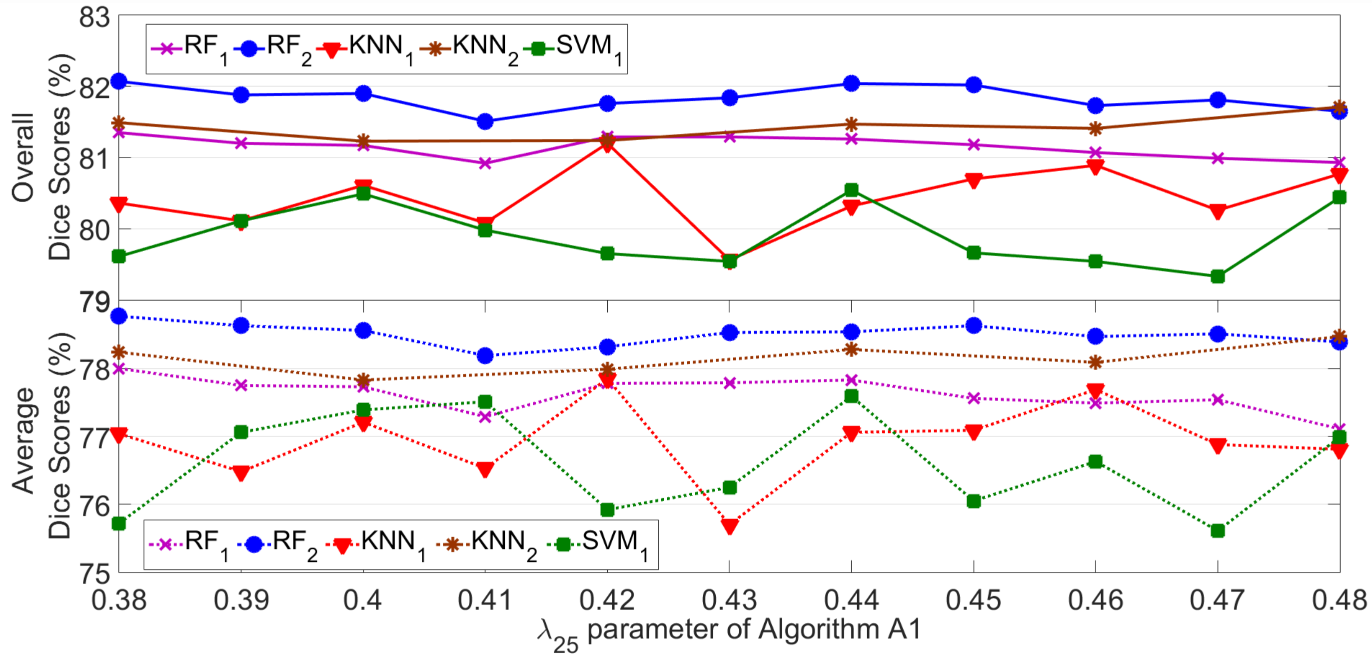




Algorithm A2
SVM
BraTS 2019 data

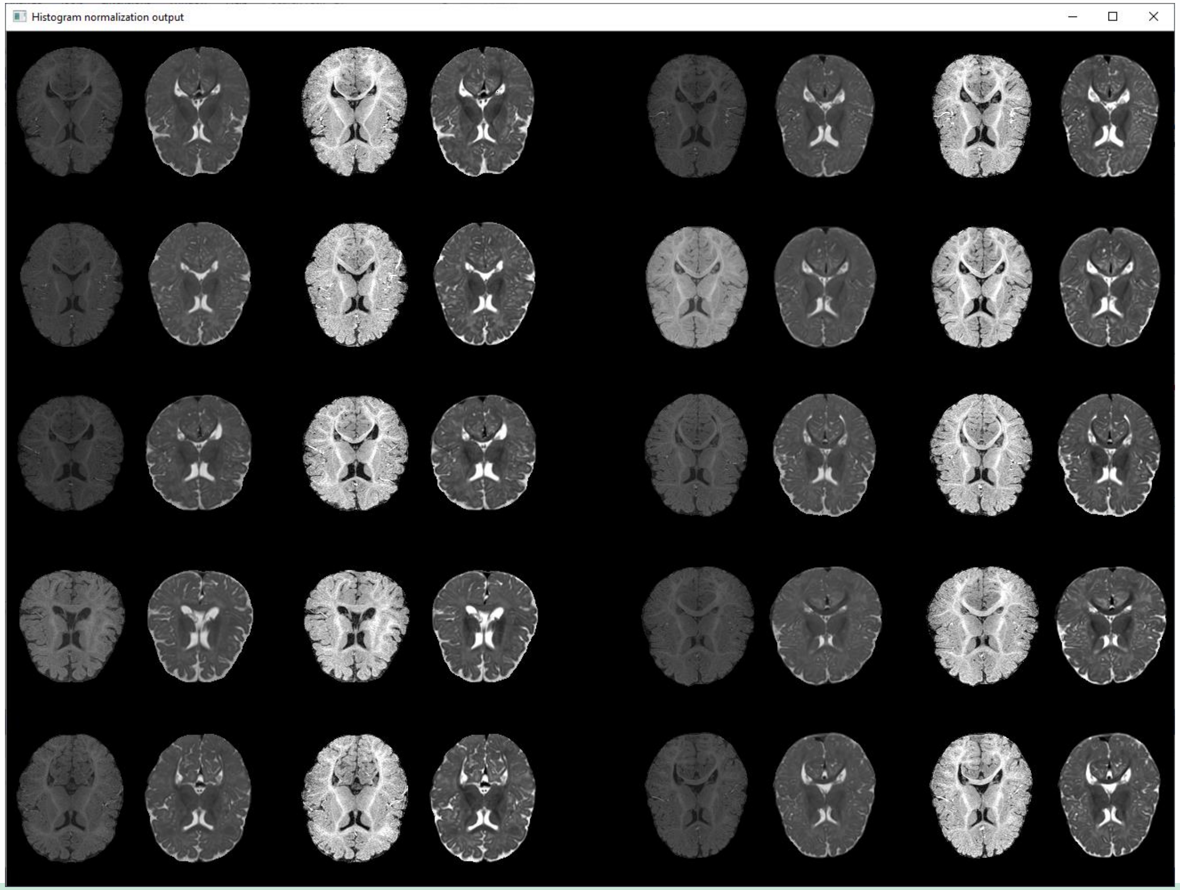


Algorithm A1
RF, KNN, SVM
BraTS 2019 data

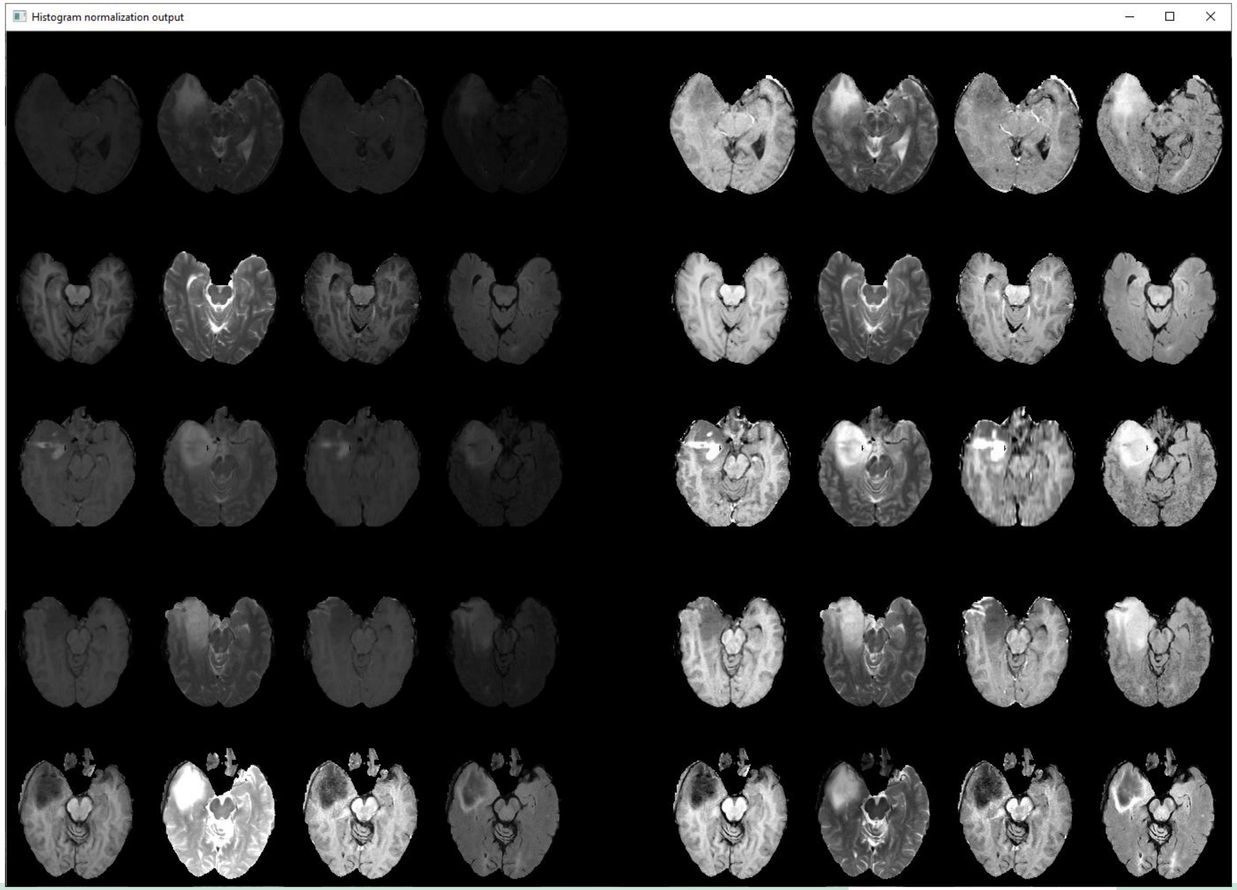


Before and after histogram normalization

Before		After		Before		After	
T1	T2	T1	T2	T1	T2	T1	T2



Before				After			
T1	T2	T1c	FLAIR	T1	T2	T1c	FLAIR



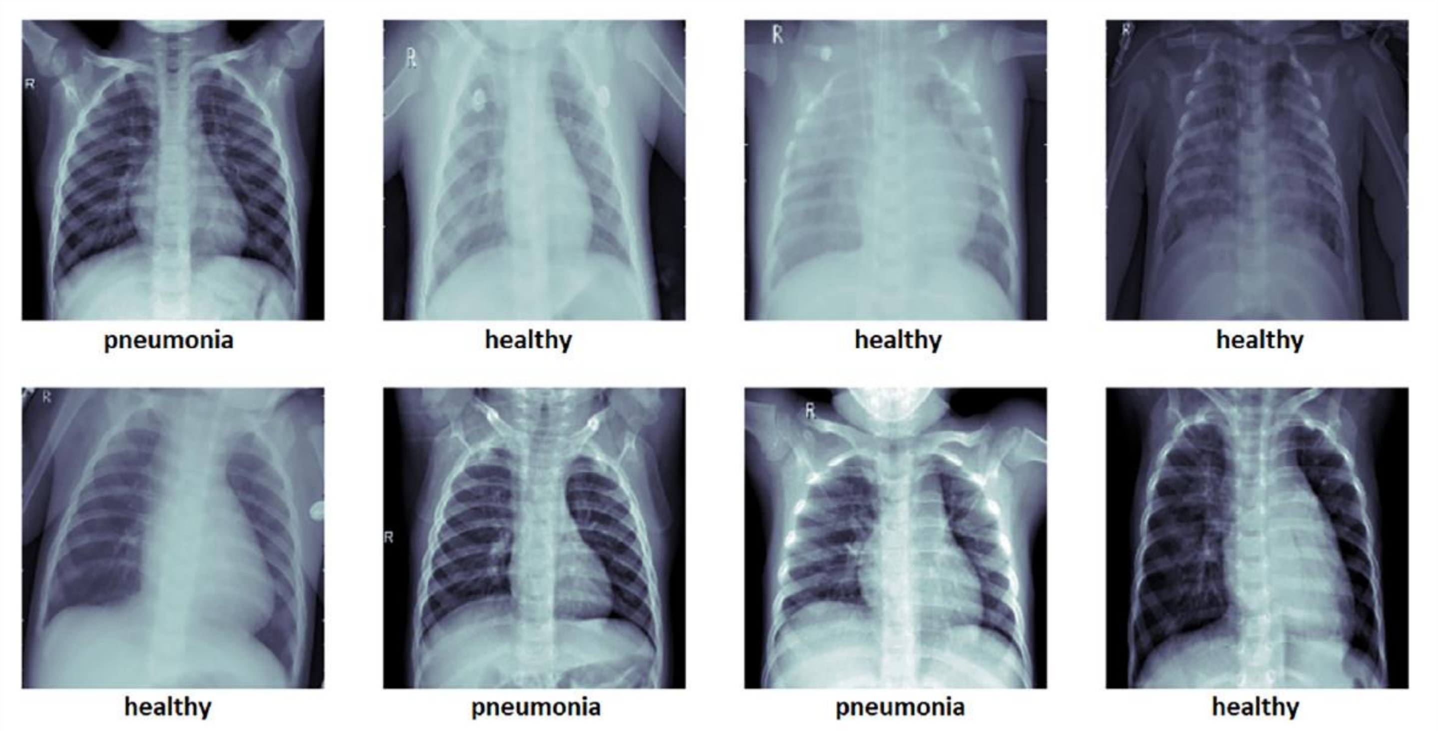
Conclusions

- Algorithm A2 – can perform better than the linear transform
- Right parameter setting is required
  - Not too many milestones
  - First milestone better at p_{20} than p_{10}, last one better at p_{80} than p_{90}
- Considerable part of the brain segmentation research community may use the method of Nyúl et al the wrong way
- They may achieve Dice scores up to 1% higher via using the right parameter setting

Scheme	Landmark points
M01	$p_{\mathrm{Lo}}, p_{50}, p_{\mathrm{Hi}}$
M02	$p_{\mathrm{Lo}}, p_{25}, p_{75}, p_{\mathrm{Hi}}$
M03	$p_{\mathrm{Lo}}, p_{25}, p_{50}, p_{75}, p_{\mathrm{Hi}}$
M04	$p_{\mathrm{Lo}}, p_{10}, p_{50}, p_{90}, p_{\mathrm{Hi}}$
M05	$p_{\mathrm{Lo}}, p_{20}, p_{40}, p_{60}, p_{80}, p_{\mathrm{Hi}}$
M06	$p_{\mathrm{Lo}}, p_{10}, p_{25}, p_{75}, p_{90}, p_{\mathrm{Hi}}$
M07	$p_{\mathrm{Lo}}, p_{20}, p_{35}, p_{50}, p_{65}, p_{80}, p_{\mathrm{Hi}}$
M08	$p_{\mathrm{Lo}}, p_{10}, p_{25}, p_{50}, p_{75}, p_{90}, p_{\mathrm{Hi}}$
M09	$p_{\mathrm{Lo}}, p_{10}, p_{25}, p_{40}, p_{60}, p_{75}, p_{90}, p_{\mathrm{Hi}}$
M10	$p_{\mathrm{Lo}}, p_{10}, p_{25}, p_{40}, p_{50}, p_{60}, p_{75}, p_{90}, p_{\mathrm{Hi}}$
M11	$p_{\mathrm{Lo}}, p_{10}, p_{20}, p_{30}, p_{40}, p_{60}, p_{70}, p_{80}, p_{90}, p_{\mathrm{Hi}}$
M12	$p_{\mathrm{Lo}}, p_{10}, p_{20}, p_{30}, p_{40}, p_{50}, p_{60}, p_{70}, p_{80}, p_{90}, p_{\mathrm{Hi}}$

Pneumonia detection using CNN

- Szepesi & Szilágyi, Biocybern Biomed Eng 2022
- Signs of pneumonia not really visible
- Chest x-ray scans of infants (1-5 years)



Modified CNN model: using dropout in the convolutional part of the network

Table 2 – Comparison of network architectures involved in this study, and their obtained benchmark values during testing. Accuracy, recall, precision, F_1 score, and AUC are presented as average value ± standard deviation.

Network	Proposed model		Transfer learning		
model	without dropout	with dropout	InceptionV3	ResNet50	VGG-19
Parameters	10.6 M	10.6 M	26.2 M	24.8 M	145.2 M
Accuracy (%)	95.67 ± 1.50	97.21 ± 1.13	90.94 ± 1.72	89.06 ± 1.64	61.19 ± 1.13
Recall (%)	95.54 ± 1.95	97.34 ± 1.56	89.10 ± 1.55	86.23 ± 2.31	61.88 ± 1.61
Precision (%)	95.50 ± 1.22	97.40 ± 1.21	91.89 ± 1.04	91.43 ± 1.59	62.33 ± 0.88
F_1 score (%)	95.52 ± 1.46	97.37 ± 1.32	90.47 ± 1.24	88.75 ± 1.88	62.10 ± 1.14
AUC	0.970 ± 0.005	0.982 ± 0.006	0.936 ± 0.010	0.921 ± 0.008	0.682 ± 0.016
Training time (50 epochs)	2304 s	2433 s	6247 s	5750 s	6577 s
Single inference time	122 ms	122 ms	307 ms	298 ms	313 ms

Table 3 – Test performance of the proposed model at various dropout rates. All indicators are presented as average value ± standard deviation.

Dropout rate	No dropout	10%	20%	30%	40%	50%
Accuracy (%)	95.67 ± 1.50	95.88 ± 0.95	96.05 ± 1.44	96.54 ± 1.39	97.21 ± 1.13	95.52 ± 1.12
Recall (%)	95.54 ± 1.95	95.70 ± 1.21	95.55 ± 1.89	96.14 ± 1.16	97.34 ± 1.56	96.12 ± 0.96
Precision (%)	95.50 ± 1.22	95.48 ± 1.78	95.60 ± 1.52	96.43 ± 1.40	97.40 ± 1.21	95.64 ± 1.19
F_1 score (%)	95.52 ± 1.46	95.59 ± 1.44	95.57 ± 1.68	96.28 ± 1.27	97.37 ± 1.32	95.88 ± 1.06
AUC	0.970 ± 0.005	0.972 ± 0.003	0.974 ± 0.004	0.976 ± 0.004	0.982 ± 0.006	0.971 ± 0.005

Comparison with recent solutions

Table 5 – Comparison with state-of-the-art methods from the literature.

Paper	Year	Method	Data	F_1 score	Runtime
Brunese et al. [4]	2020	VGG-16	6523 CXR	97%	2.5 s
Panwar et al. [5]	2020	VGG-19 + GradCAM	2482 CT + 6382 CXR	95.61%	2 s
Mahmud et al. [13]	2020	customized CNN (CovXNet)	6161 CXR	97.4%	N/A
Ouchicha et al. [14]	2020	customized CNN (CVDNet)	2905 CXR	96.7%	N/A
Wang et al. [15]	2020	3D-ResNet	4697 CXR	93.3%	N/A
Choudhury et al.[31]	2020	DenseNet201	3487 CXR	97.94%	N/A
Ren et al. [32]	2020	CNN + Bayesian Network	35,389 CXR	87%	N/A
Arias et al. [33]	2020	CNN	79,500 CXR	91.5%	N/A
Sakib et al. [34]	2020	customized CNN (DL-CRC)	5367 CXR	94%	N/A
Ozturk et al. [35]	2020	YOLO via DarkNet	1000 CXR	87–98%	< 1 sec
Alhudjaif et al. [10]	2021	DenseNet-201	1218 CXR	94.96%	"within seconds"
Nikolaou et al. [36]	2021	EfficientNet models	15,153 CXR	95%	N/A
Das et al. [37]	2021	CNN + transfer learning	1004 CXR	95%	"few seconds"
Munusamy et al. [38]	2021	FractalCovNet	473 CT + 11,934 CXR	92–98%	N/A
Joshi et al. [39]	2021	DarkNet-53	6884 CXR	97.11%	0.137 s
Singh and Tripathi [40]	2022	Quaternion CNN	5856 CXR	93.75%	N/A
Dash and Mohapatra [41]	2022	CNN + transfer learning	1272 CXR	97.12%	N/A
Gour and Jain [42]	2022	VGG-19, Xception	4645 CT + 3040 CXR	97.5%	0.029–3.66 s
Proposed method	2022	CNN + modified dropout	5856 CXR	97.4%	0.122 s

Transparent neural networks

- EU regulations: all machine-made decisions affecting human lives must be accompanied by explanation
- Explainable artificial intelligence
  - E.g. decision trees, random forest, KNN are explainable
  - Conventional neural networks are not explainable
- Explainable solutions are needed for complex decisions
- Collaborator team led by O. Csiszár has a potential solution under development (Csiszár et al, Knowl. Based Syst 2020, 2021)
- Future work: provide explainable solutions for medical diagnosis problems
- Main challenge: diagnosis needs complex decisions; current transparent model needs to be extended and possibly assisted by clustering methods that decompose complex problems into several, less complex ones

THANKS FOR
YOUR ATTENTION

AND

ANY
QUESTIONS ?