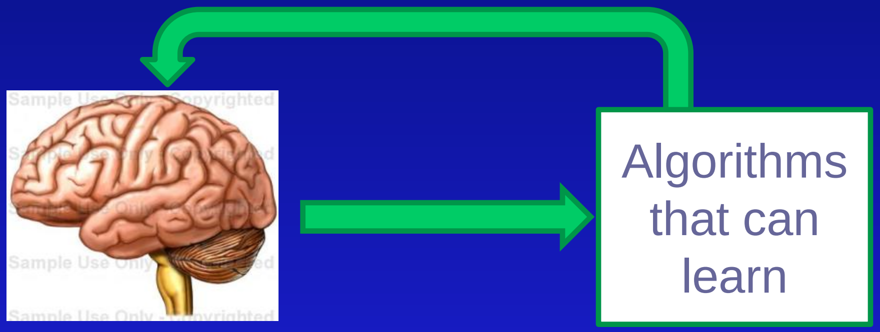
Mesh Learning for Classifying Cognitive Processes

Mete Özay*, Orhan Fırat*
İlke Öztekin**, Uygar Öztekin***,
Fatoş T. Yarman Vural*

* Middle East Technical University, Ankara
** Koc University, Istanbul
*** Google

Motivation: Can we model the brain activities measured by fMRI as a machine learning system?



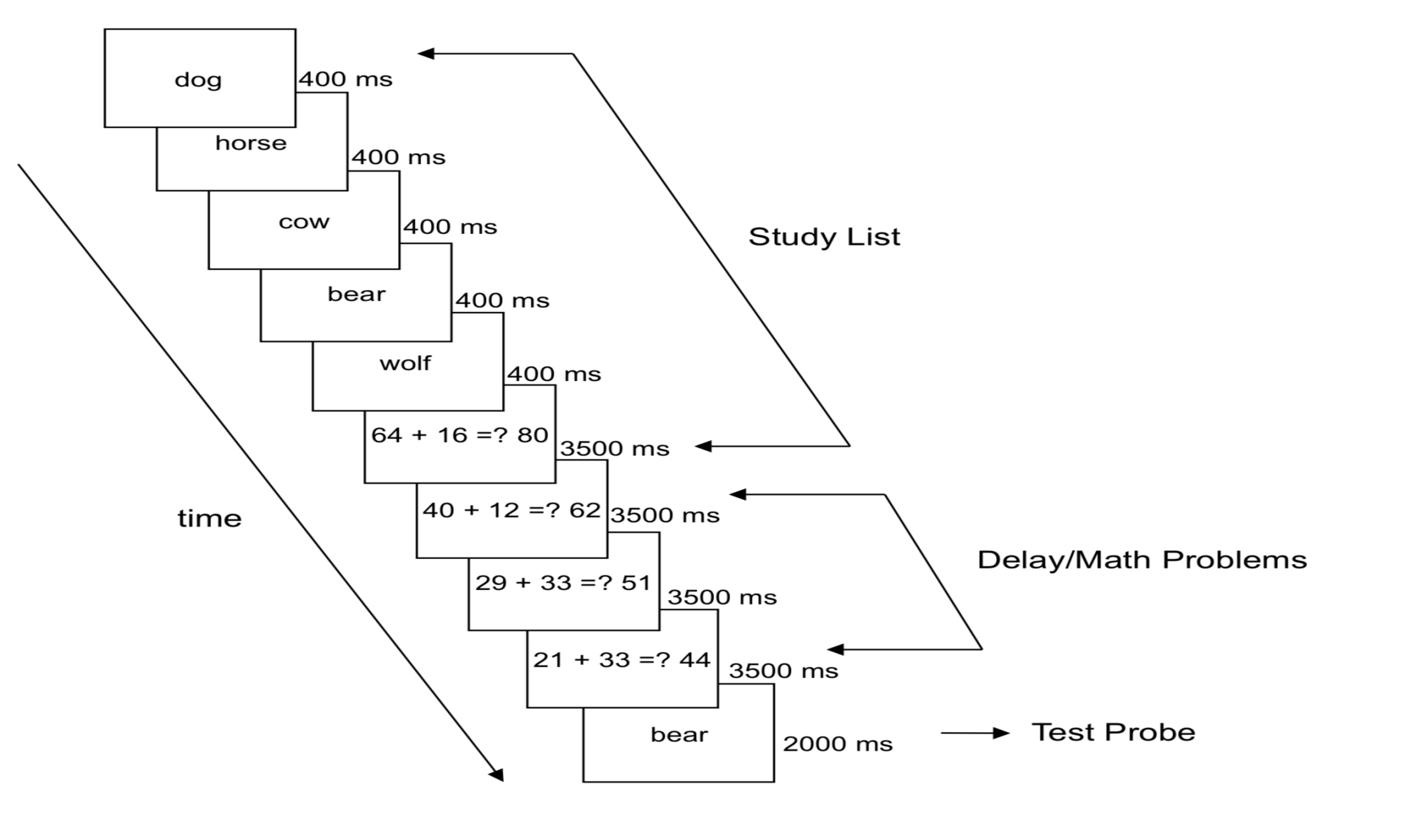
Human

Machine

Focus: Design a classifier to model the distributed patterns of activity in memory

Machine		Human
1.Training	⟷	1. Encoding: subject studies objects from a category
2.Test	⟷	2. Retrieval: subject is asked to recognize a test object

fMRI Data Acquisition



(Öztekin & McElree, 2007; Öztekin et al., 2009; Öztekin & Badre, 2011)

Image samples



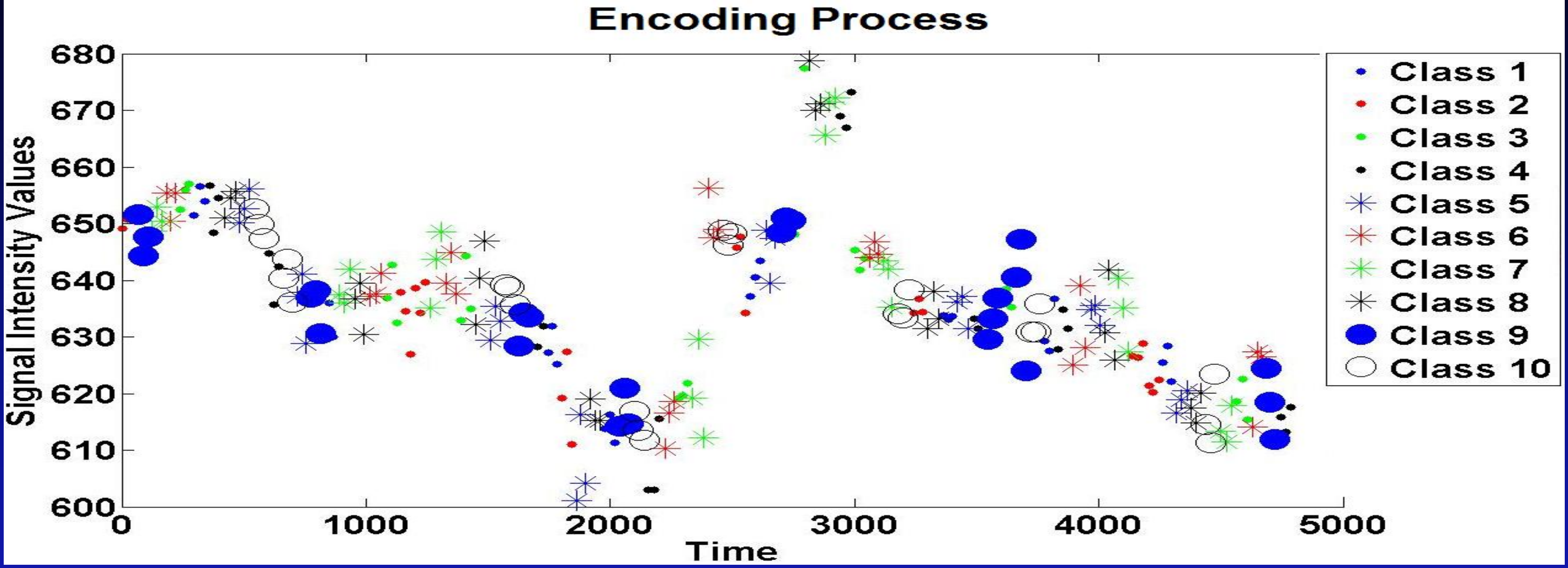


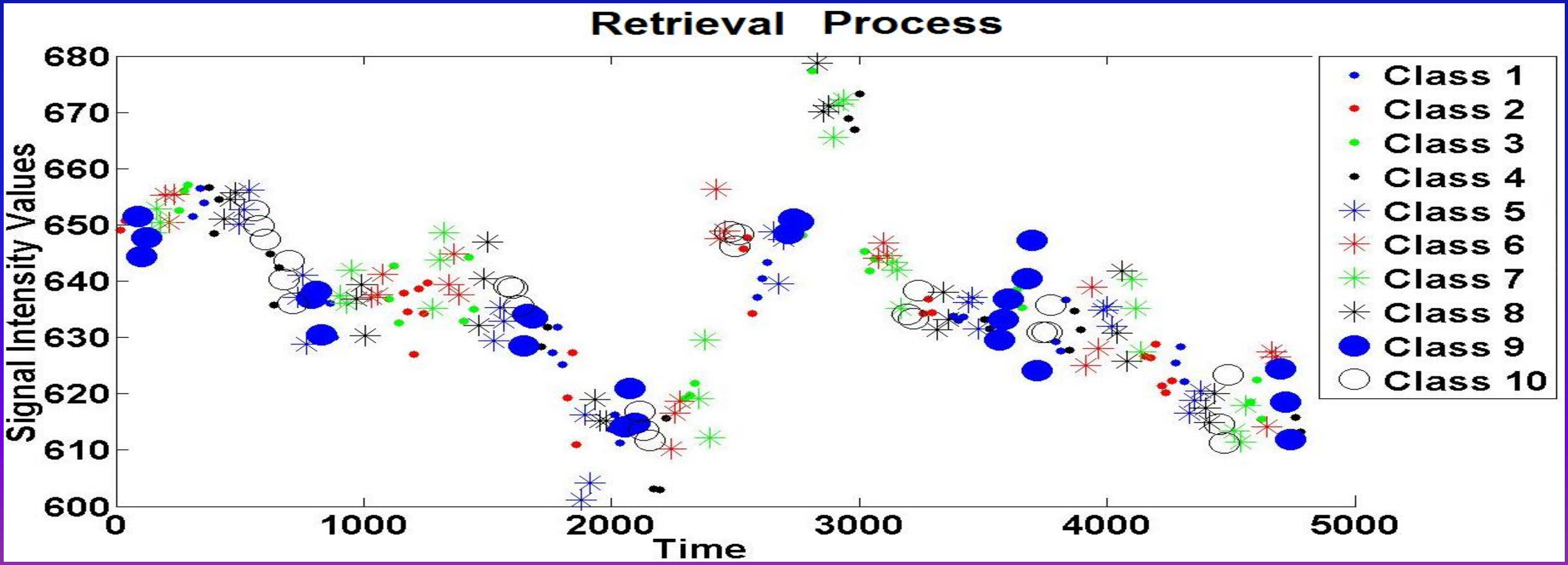


fMRI dataset

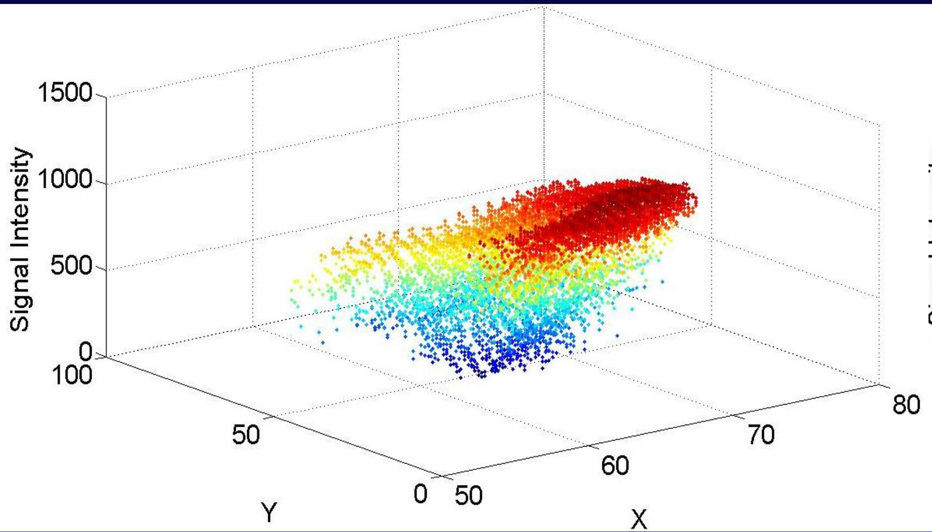
- 10 semantic categories:
  - *animals, colors, furniture, body parts, fruits, herbs, clothes, chemical elements, vegetables and tools.*
- Dataset:
  - 24 samples /category
  - 240 training + 240 test samples from the encoding and retrieval phase and
- Number of voxels:
  - Memory: 8142
  - Whole Brain: 82 600

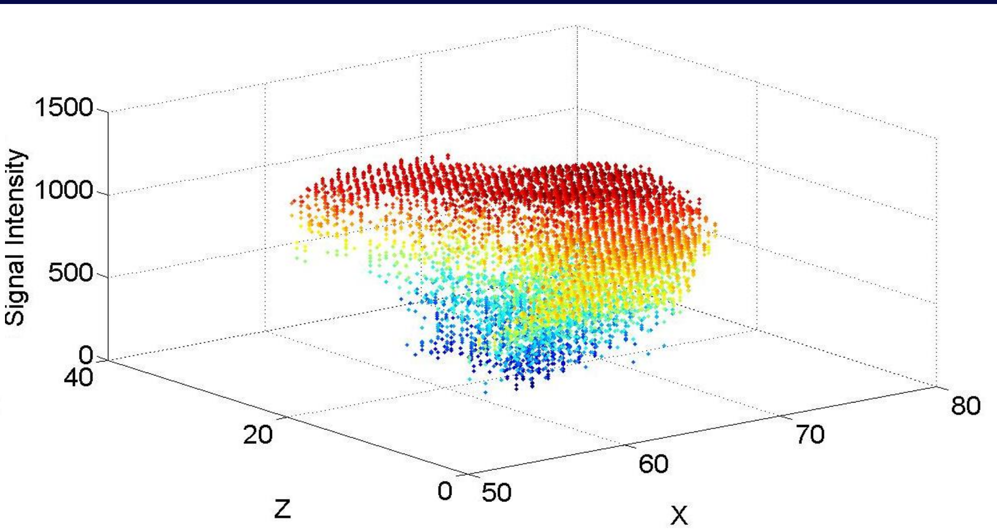
fMRI intensity values of a voxel

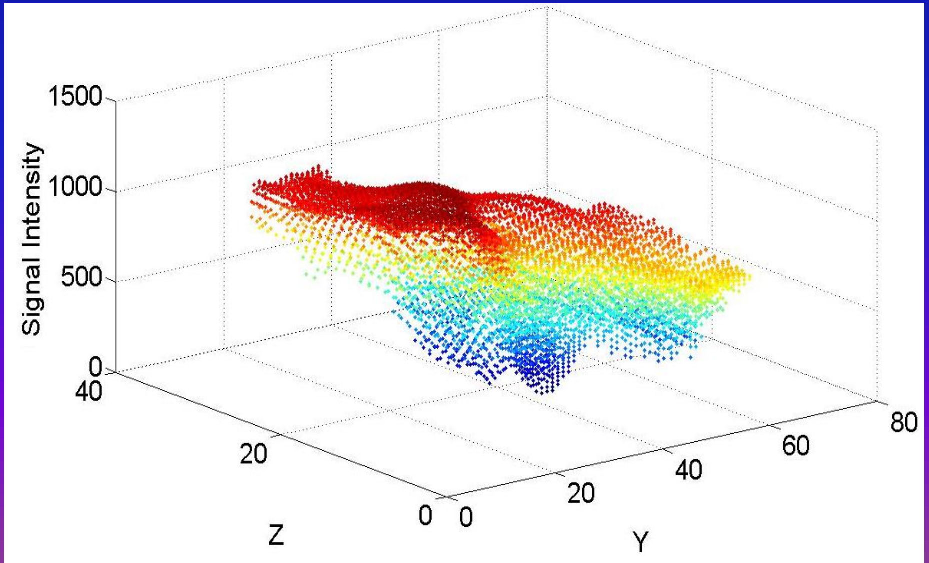




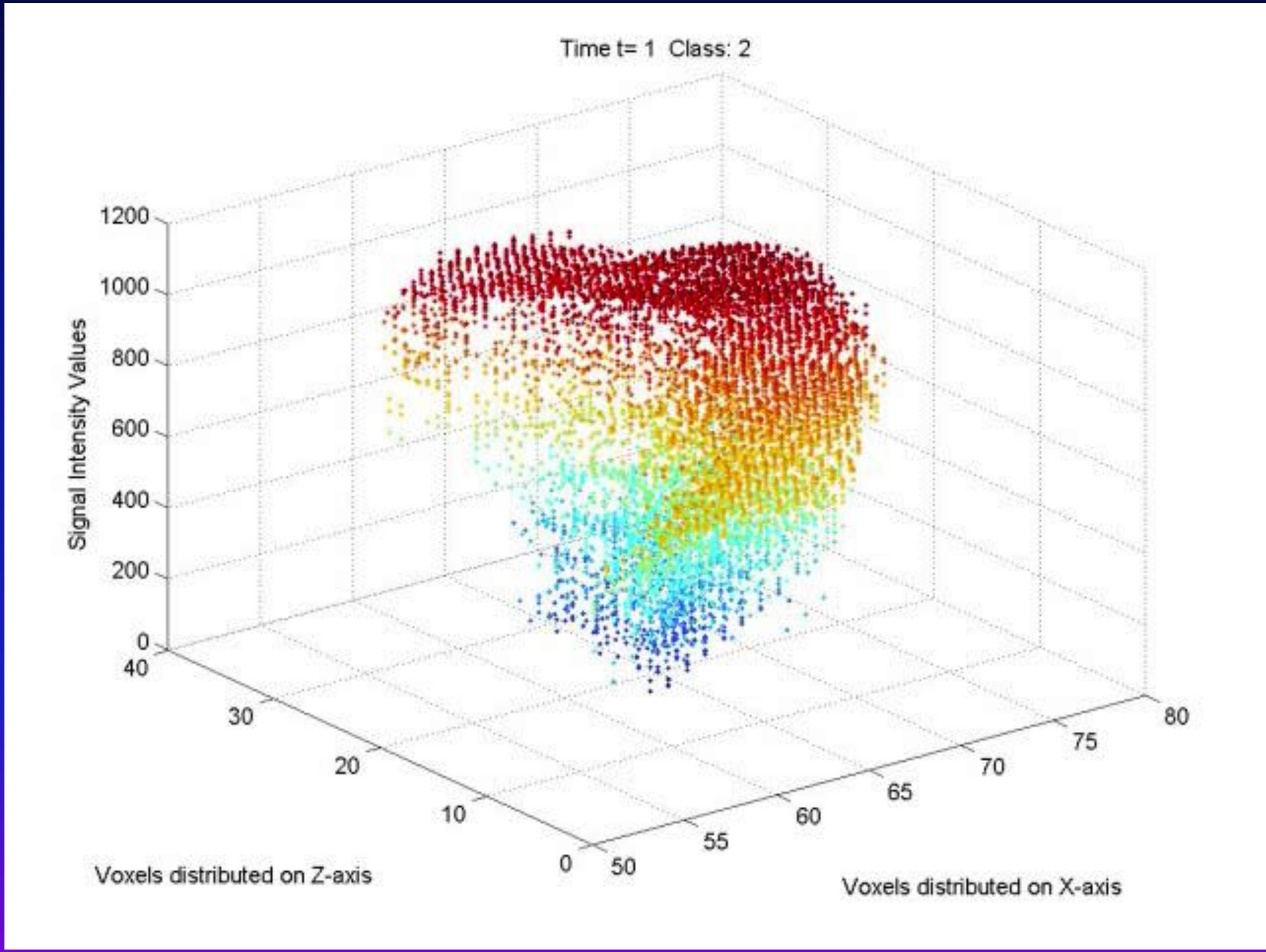
Spatial Distribution of Voxel Intensities for a time instant *t*





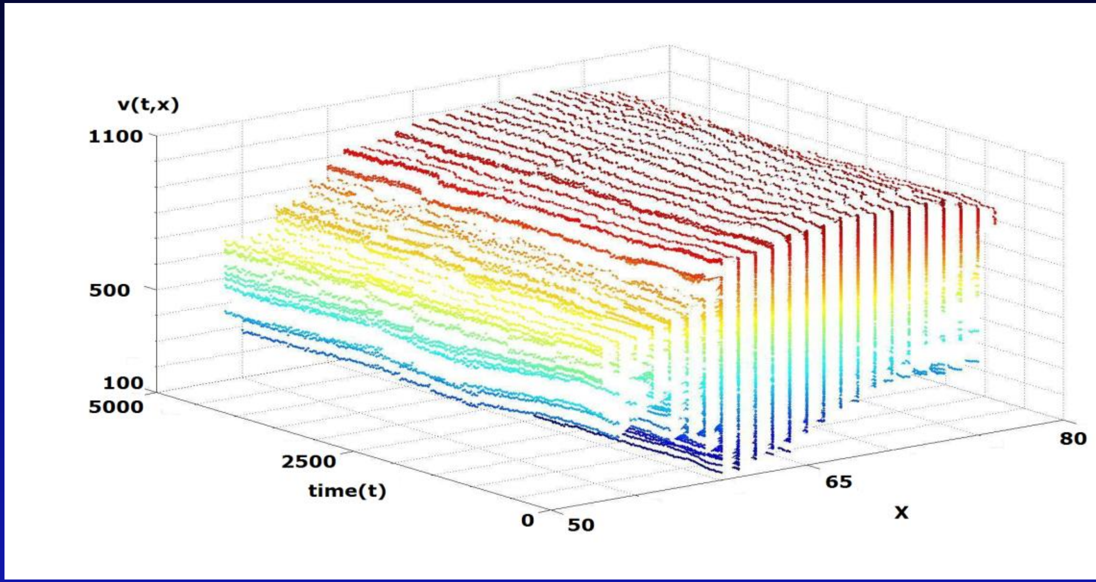


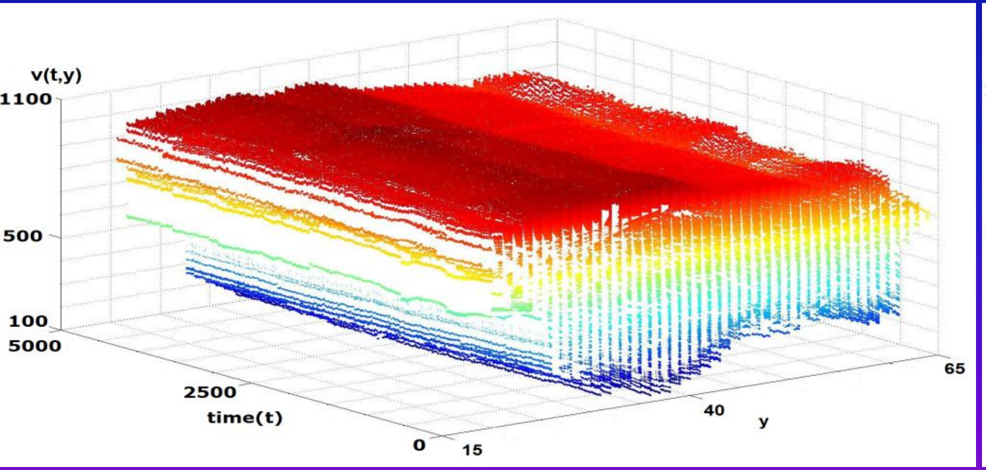
fMRI intensity values as a function of time

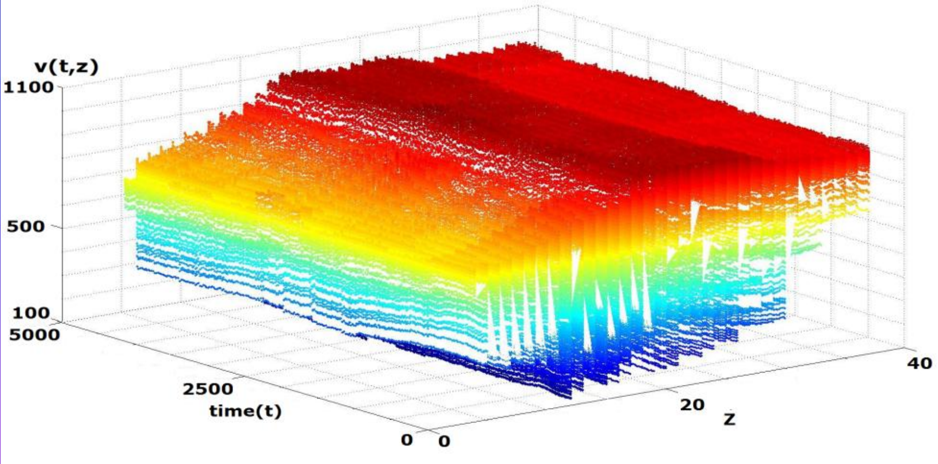


State of the art performance ~25-30%
(e.g. Öztekin & Badre, 2011)

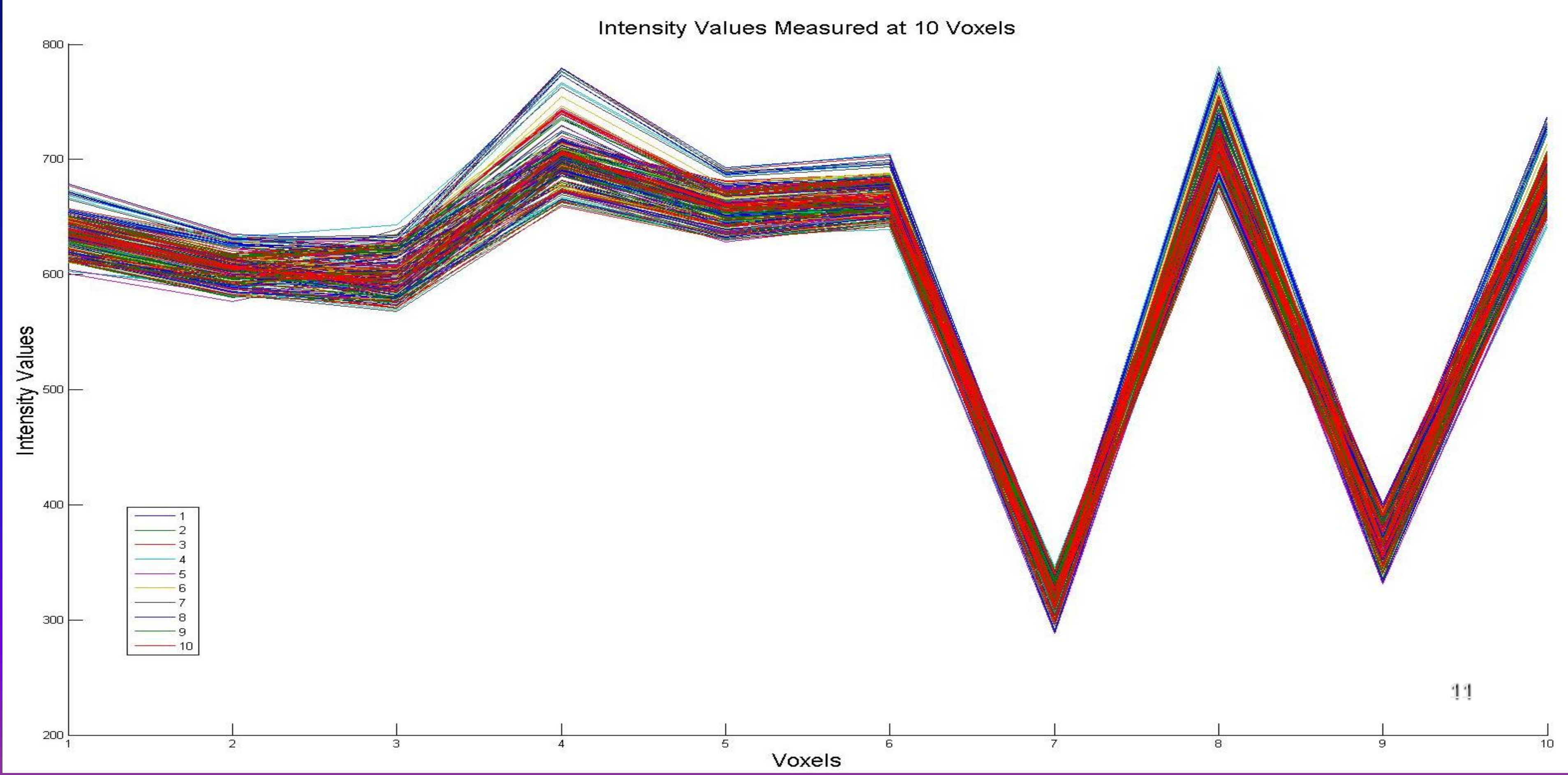
Spatio-temporal Distribution of Voxel Intensities



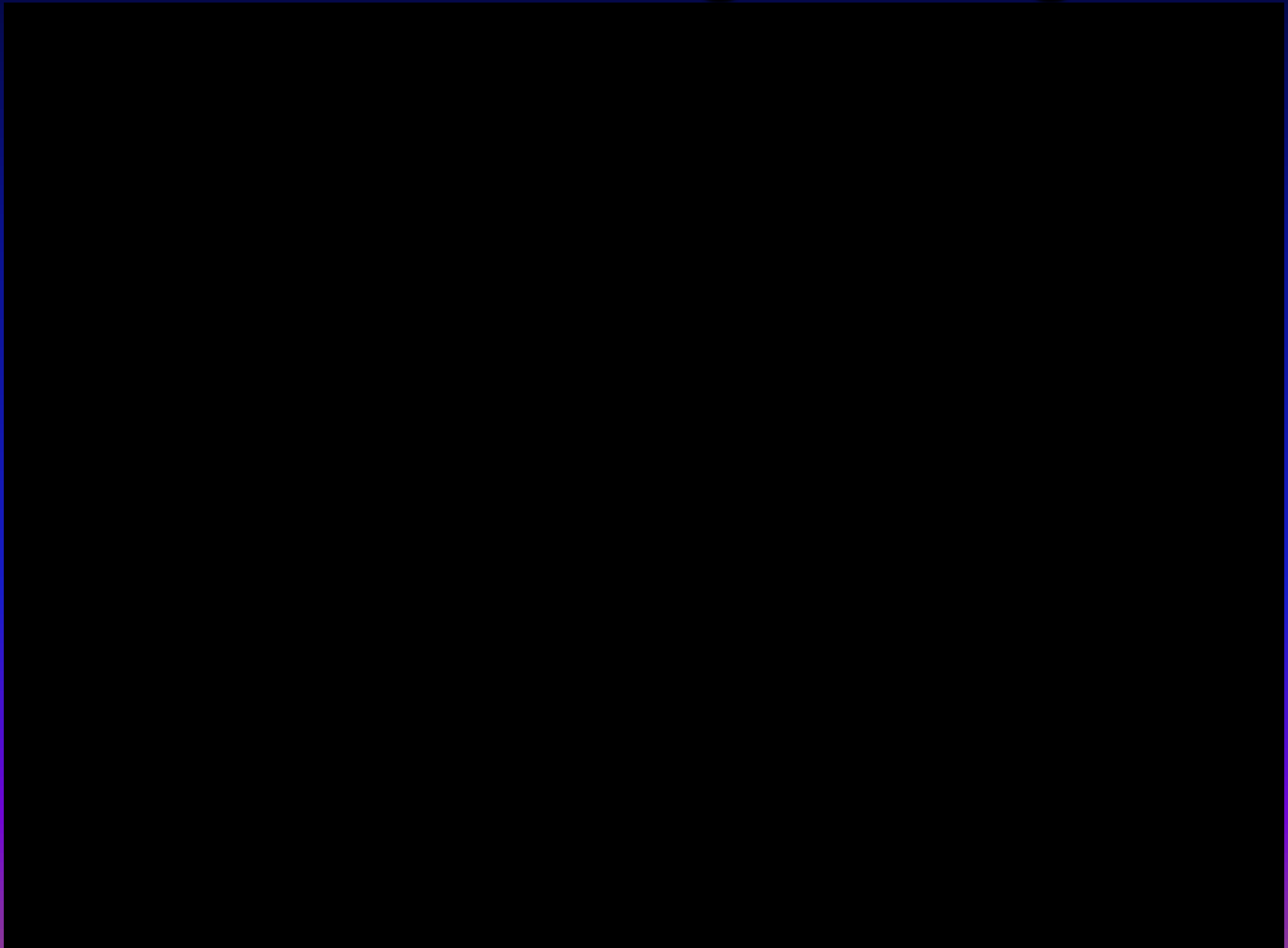




Voxel intensity values for 10 class at 10 neighboring voxels



Difference of intensity values between two neighboring voxels



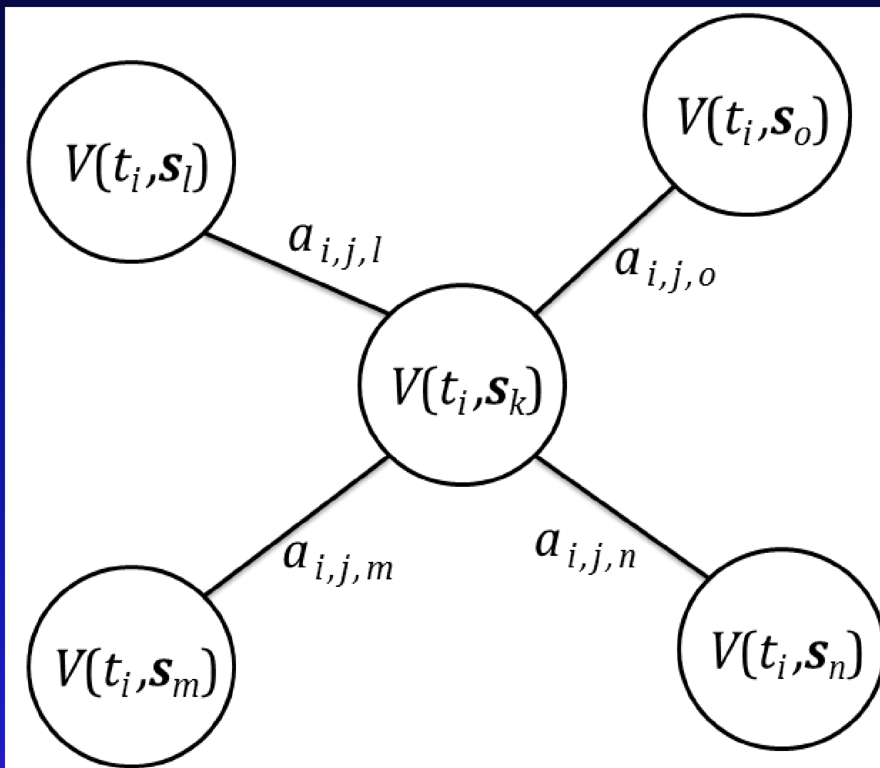
Neurons are massively interconnected

- Relationships among the voxels are more discriminative then the individual voxel intensity values to represent a certain category

↓

Need to model the relationships among the voxels

A Local Mesh Model:



A voxel is represented in a neighborhood system

$$M\left(\eta_p\left[v\left(t_i,\bar{s}_j\right)\right]\right)=\left(v\left(t_i,\bar{s}_j\right)\in\eta_p, a_{i,j,k}\in A\right)$$

$$v\left(t_i,\bar{s}_j\right)=\sum_{\bar{s}_k\in\eta_p} a_{i,j,k}\,v\left(t_i,\bar{s}_k\right)+\varepsilon_{i,j}$$

$$\eta_1\left[v\left(t_i,\bar{s}_j\right)\right]=\left\{v\left(t_i,\bar{s}_k\right):\left\|\bar{s}_j-\bar{s}_k\right\|\le\left\|\bar{s}_j-\bar{s}_l\right\|,\ \forall\ v\left(t_i,\bar{s}_l\right)\in D\right\}$$

$$\eta_p\left[v\left(t_i,\bar{s}_j\right)\right]=\left\{v\left(t_i,\bar{s}_k\right)\cup\eta_{p-1}v\left(t_i,\bar{s}_j\right):\left\|\bar{s}_j-\bar{s}_k\right\|\le\left\|\bar{s}_j-\bar{s}_l\right\|,\forall v\left(t_i,\bar{s}_l\right)\in\eta_{p-1}\left[v\left(t_i,\bar{s}_j\right)\right]^c\right\}$$

Local Relational Features: LRF

$$\overline{a}_{i,j} = \left[a_{i,j,1} \; a_{i,j,2} \ldots a_{i,j,p} \right]$$

Algorithm 1 : Extract Linear Relation Features (LRF); *lrf*

Input: Dataset : $\mathcal{D} = \{ v(t_i, \overline{s}_j) \}$,

Order of LRF : p

Begin

$F = [\;]$;

1. *for* j=1 to M
2. *for* i=1 to N
3. Compute p-neighborhood $\eta_n\left[v(t_i, \overline{s}_j) \right]$ of $v(t_i, \overline{s}_j)$;
4. Compute $\overline{a}_{i,j}$ optimizing (ε^2ij
5. *endfor* (i)
6. Construct A_j using $\overline{a}_{i,j}$;
7. *endfor* (j)
8. Construct F using A_j ;

End

Output: Feature matrix F

$$A_j = \left[\overline{a}_{1,j} \; \overline{a}_{2,j} \; \ldots \; \overline{a}_{N,j} \right]^T$$

Mesh Learning with Spatial Neighborhood

Algorithm **:** Classification with Linear Relation Features (LRF)**;** *classify.lrf*

Input: Training and Test Datasets : $\mathcal{D}^{tr} = \{ v^{tr}(t_i, \bar{s}_j) \}$, $\mathcal{D}^{te} = \{ v^{te}(t_i, \bar{s}_j) \}$,

Training Labels : $L_{tr} = \{l_i\}_{i=1}^{N}$

Order of LRF : p

Begin

$F^{tr} = [\ ]$, $F^{te} = [\ ]$;

1. $F_{tr} \leftarrow lrf\ (\mathcal{D}^{tr}, p)$;
2. $F_{te} \leftarrow lrf\ (\mathcal{D}^{te}, p)$;
3. Perform classification on F_{tr} and F_{te} using a classification algorithm with the algorithm parameters θ;

$$\hat{L}_{te} = \{\hat{l}_i\}_{i=1}^{N} \leftarrow classify(F_{tr}, L_{tr}, F_{te}, \theta);$$

End

Output: $\hat{L}_{te} = \{\hat{l}_i\}_{i=1}^{N}$

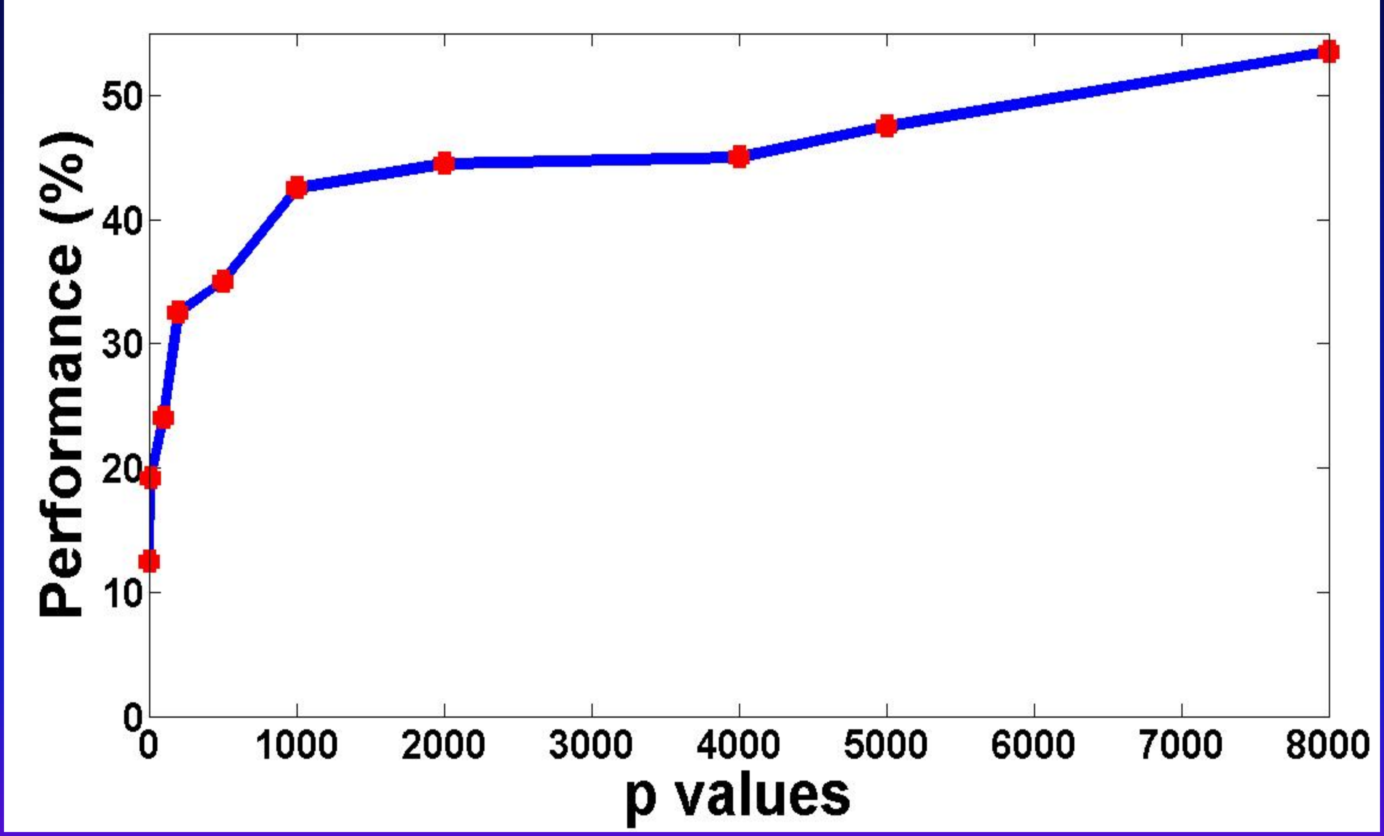
Performance of Mesh Learner with Spatial Neighborhood

10 class classification performances using 8142 voxels.

%	LRF Order Values (p)			Without LRF	KPCA	PCA	ICA
	8	9	10				
SVM	44	41	45	40	11	40	10
k-nn	56	56	57	48	26	44	11

ICA: Independent component analysis
PCA: Principal Component Analysis
KPCA: Kernel Principal Component Analysis

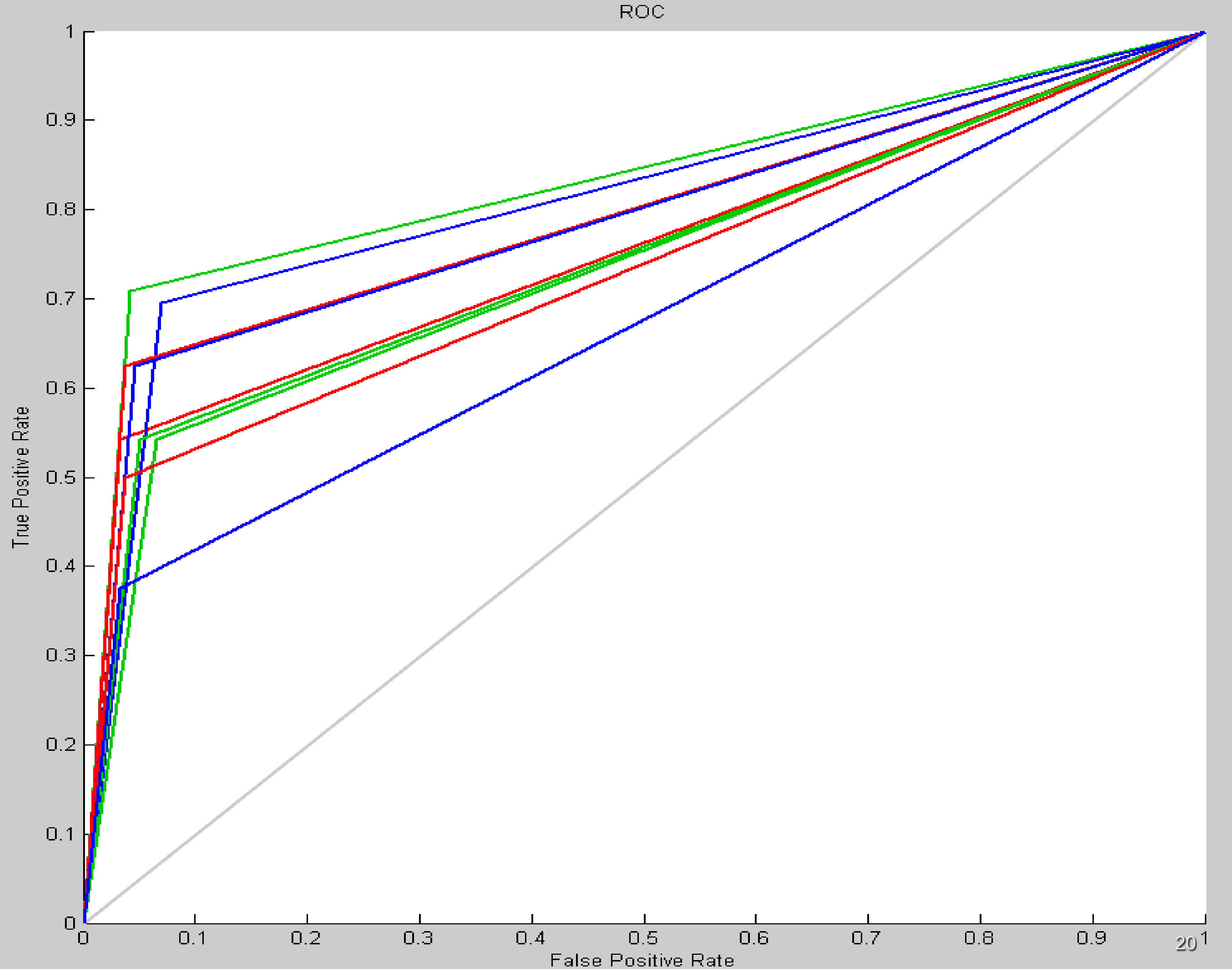
Experiments on Mesh Learner with Spatial Neighborhood



Single voxel performance for 10 classes.

Confusion Matrix

Output Class \ Target Class	1	2	3	4	5	6	7	8	9	10	
1	**13** 5.4%	**2** 0.8%	**1** 0.4%	**3** 1.3%	**2** 0.8%	**0** 0.0%	**0** 0.0%	**1** 0.4%	**4** 1.7%	**1** 0.4%	48.1% 51.9%
2	**2** 0.8%	**17** 7.1%	**0** 0.0%	**0** 0.0%	**1** 0.4%	**0** 0.0%	**3** 1.3%	**2** 0.8%	**0** 0.0%	**1** 0.4%	65.4% 34.6%
3	**0** 0.0%	**0** 0.0%	**13** 5.4%	**1** 0.4%	**0** 0.0%	**4** 1.7%	**0** 0.0%	**0** 0.0%	**2** 0.8%	**0** 0.0%	65.0% 35.0%
4	**4** 1.7%	**0** 0.0%	**1** 0.4%	**16** 6.7%	**1** 0.4%	**0** 0.0%	**0** 0.0%	**1** 0.4%	**4** 1.7%	**4** 1.7%	51.6% 48.4%
5	**3** 1.3%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**13** 5.4%	**0** 0.0%	**0** 0.0%	**4** 1.7%	**0** 0.0%	**4** 1.7%	54.2% 45.8%
6	**0** 0.0%	**0** 0.0%	**4** 1.7%	**0** 0.0%	**1** 0.4%	**15** 6.3%	**3** 1.3%	**0** 0.0%	**0** 0.0%	**0** 0.0%	65.2% 34.8%
7	**0** 0.0%	**1** 0.4%	**0** 0.0%	**1** 0.4%	**0** 0.0%	**3** 1.3%	**15** 6.3%	**2** 0.8%	**1** 0.4%	**2** 0.8%	60.0% 40.0%
8	**1** 0.4%	**4** 1.7%	**3** 1.3%	**0** 0.0%	**2** 0.8%	**1** 0.4%	**2** 0.8%	**13** 5.4%	**0** 0.0%	**1** 0.4%	48.1% 51.9%
9	**1** 0.4%	**0** 0.0%	**2** 0.8%	**1** 0.4%	**2** 0.8%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**12** 5.0%	**2** 0.8%	60.0% 40.0%
10	**0** 0.0%	**0** 0.0%	**0** 0.0%	**1** 0.4%	**2** 0.8%	**1** 0.4%	**1** 0.4%	**1** 0.4%	**1** 0.4%	**9** 3.8%	56.3% 43.8%
	54.2% 45.8%	70.8% 29.2%	54.2% 45.8%	69.6% 30.4%	54.2% 45.8%	62.5% 37.5%	62.5% 37.5%	54.2% 45.8%	50.0% 50.0%	37.5% 62.5%	**56.9%** **43.1%**


ROC
True Positive Rate
False Positive Rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Discussion on Mesh Learning with Spatial Neighborhoods

- Spatial neighborhood with L2-norm implies anatomical surroundings of a voxel; which may not be the case in cognitive process.

  ⇨ Employ functional connectivity.

- Selecting the optimal value of p is not validated and introduced as a user parameter.

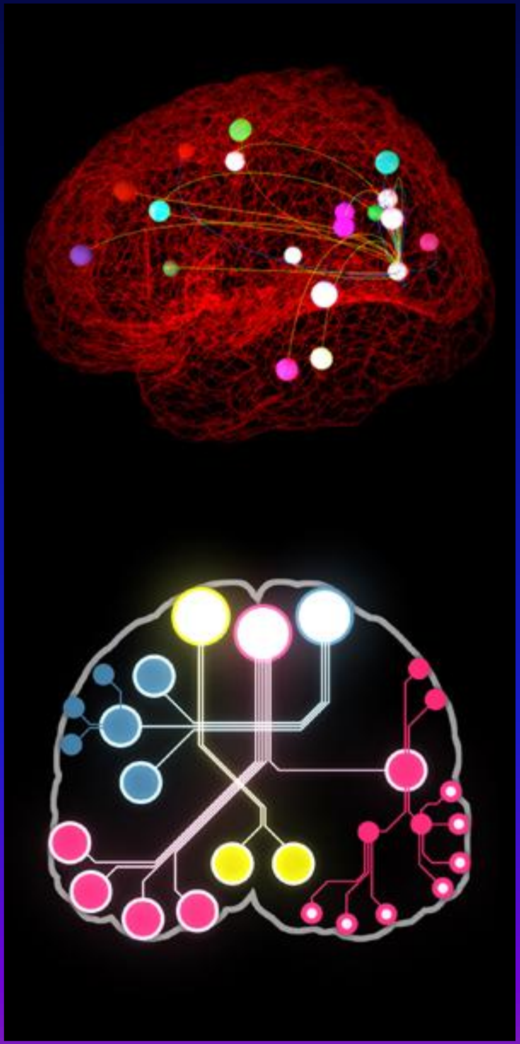
  ⇨ Find voxels which are highly correlated to other voxel. p changes for each voxel.

Mesh Learning with Functional Neighborhood

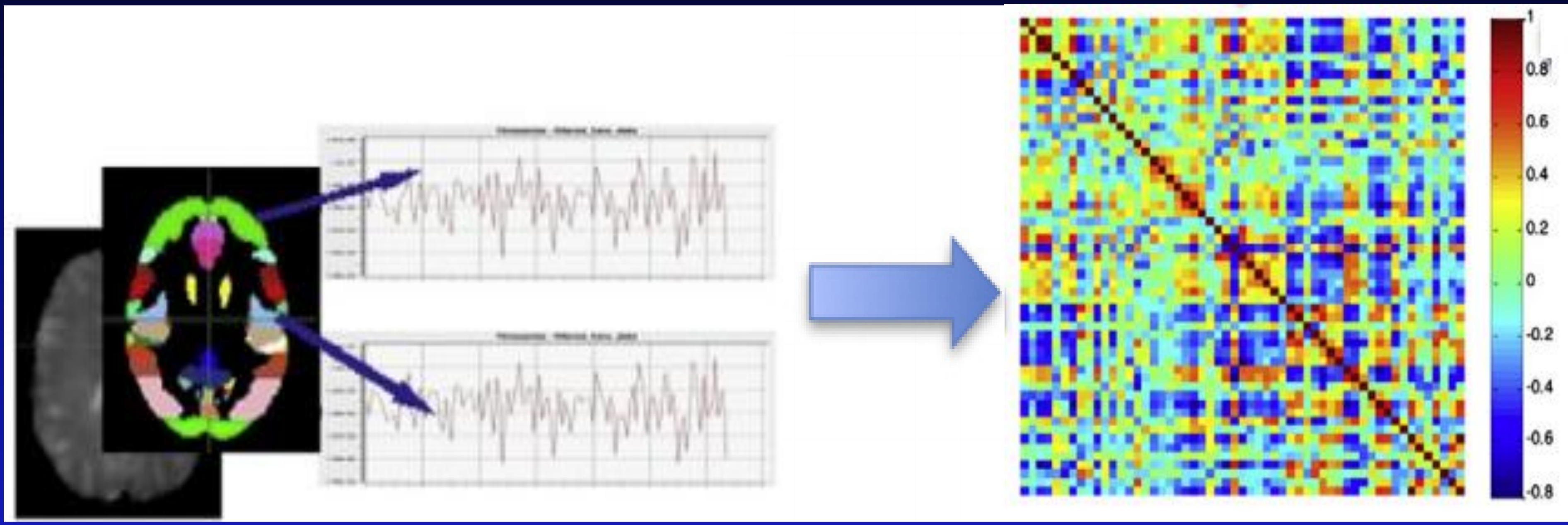
1. Need to define functional connectivity among the voxels
2. Define Functional Neighborhood
3. Apply functional neighborhood to k-nn to and select *k-functionally-closest* neighbors which implies coupled-activation in cognitive process

Functional Connectivity

- Statistical association or dependency among the time series of voxels



Modeling Methods of Brain Using Connectivity



1. Find correlation between time series of two voxels, using a correlation metric .
2. Construct correlation matrix by using correlation measure of each pair of voxels.

Cross Correlation Metric

- Cross-correlation of any two individual time-series (i,j), at lag h, $\rho_{ij}(h)$, is defined as

$$\rho_{ij}(h) = \frac{cov_{ij}(t+h)}{\sqrt{var_i(t).var_j(t+h)}}$$

Scalability of Functional Connections

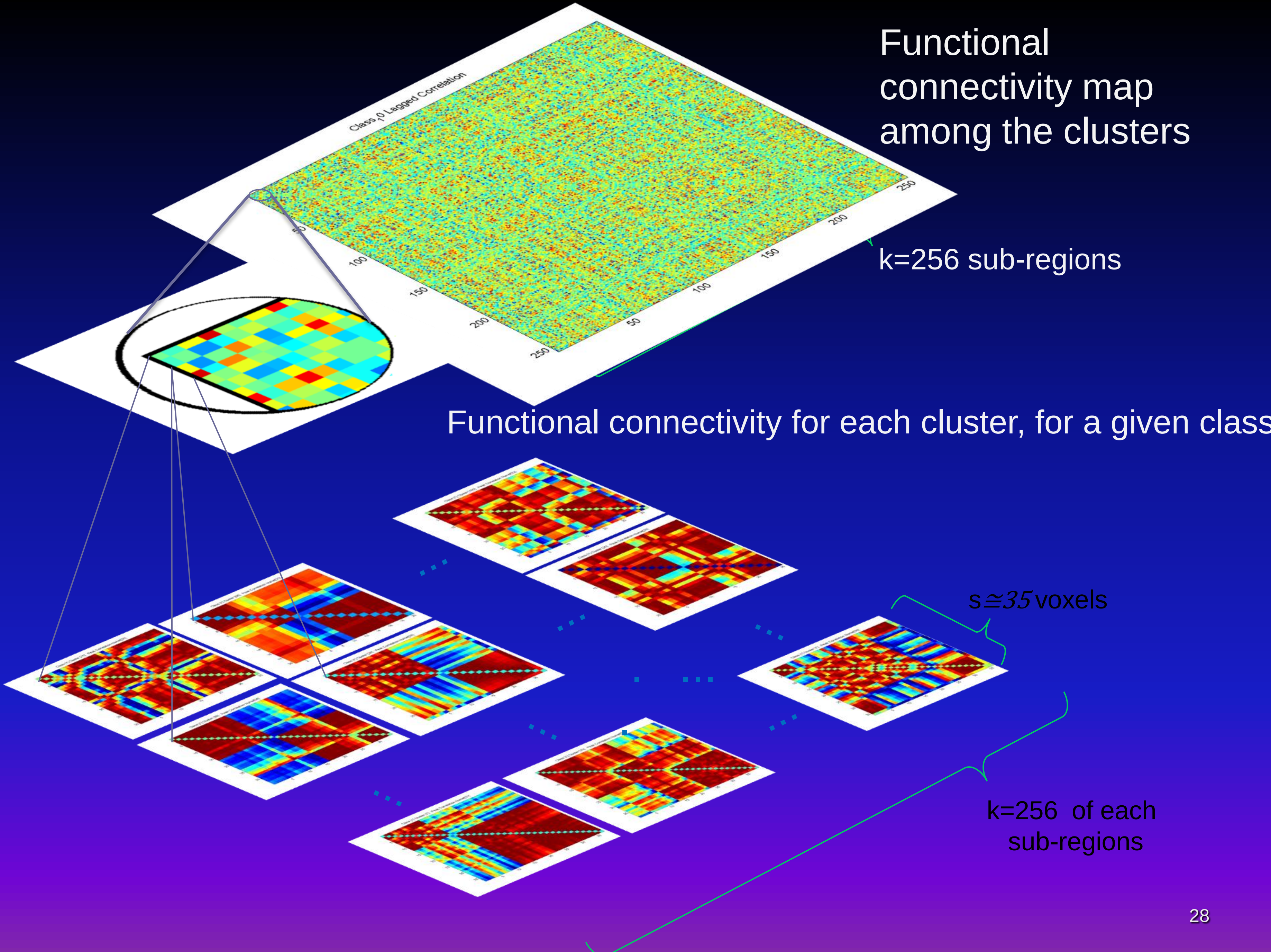
- Connectivity matrices are expensive in voxel level, when no approximations are made

- Considering functional relations of a voxel with all other voxels;

- 8142 voxels makes 33M functional relations

Design and Use of Functional Connectivity

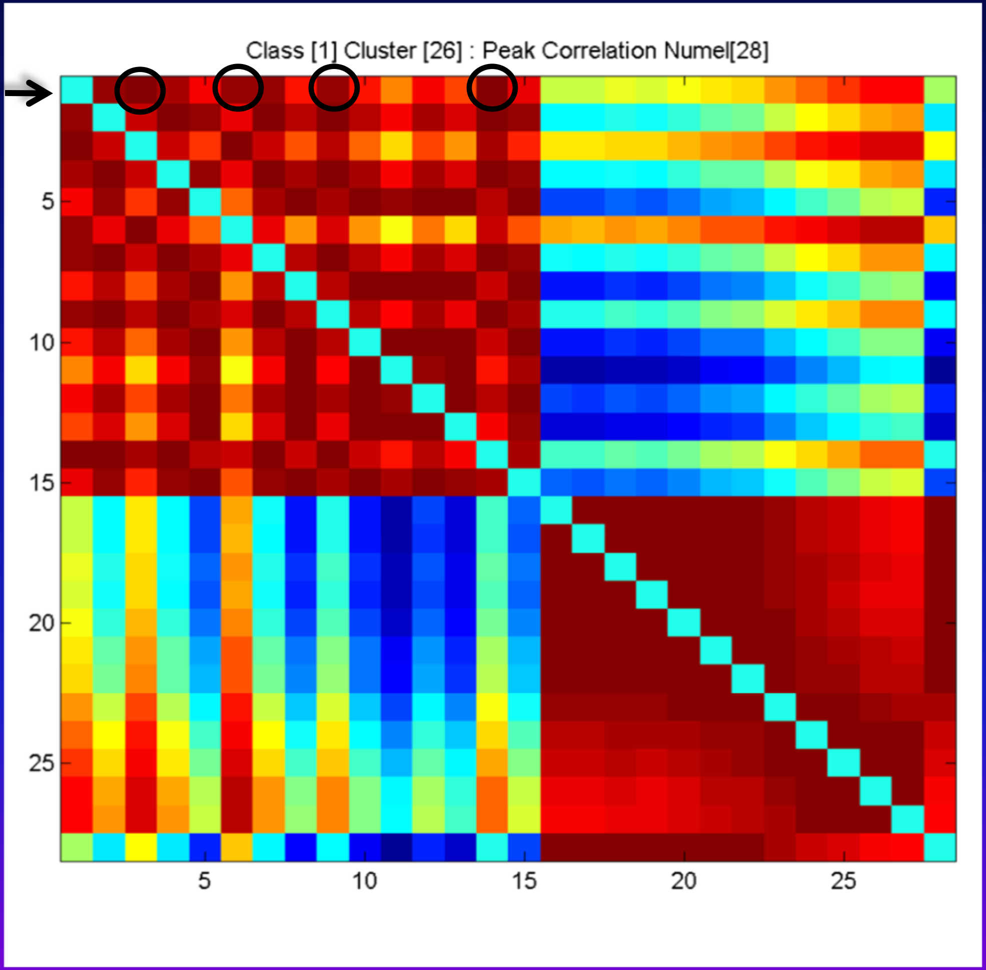
- Cluster voxels by their locations
- Measure correlation metric within clusters to generate connectivity matrices
- Use connectivity matrices to find functionally-nearest neighbors


Functional connectivity map among the clusters
Class_0 Lagged Correlation
50
100
150
200
250
k=256 sub-regions
Functional connectivity for each cluster, for a given class
s≅35 voxels
k=256 of each sub-regions

Design of New Neighborhood System

- Rather then selecting *p-spatially* closest points by L2-norm; select *p-functionally* closest points
- Select *p-functionally* closest points analyzing rows of within-cluster connectivity matrix
- Construct neighborhood-set with *p-functionally* closest voxels and calculate LRF accordingly

Given a voxel Select *p-Functionally* Closest Voxel(s)



Selecting $p=4$ functionally closest voxels v_j

For voxel v_i in cluster c_k

where $i=1$ and $k=26$;

Resulting neighbor indexes by considering highly correlated voxels in the cluster:

$j=\{3,6,9,14\}$

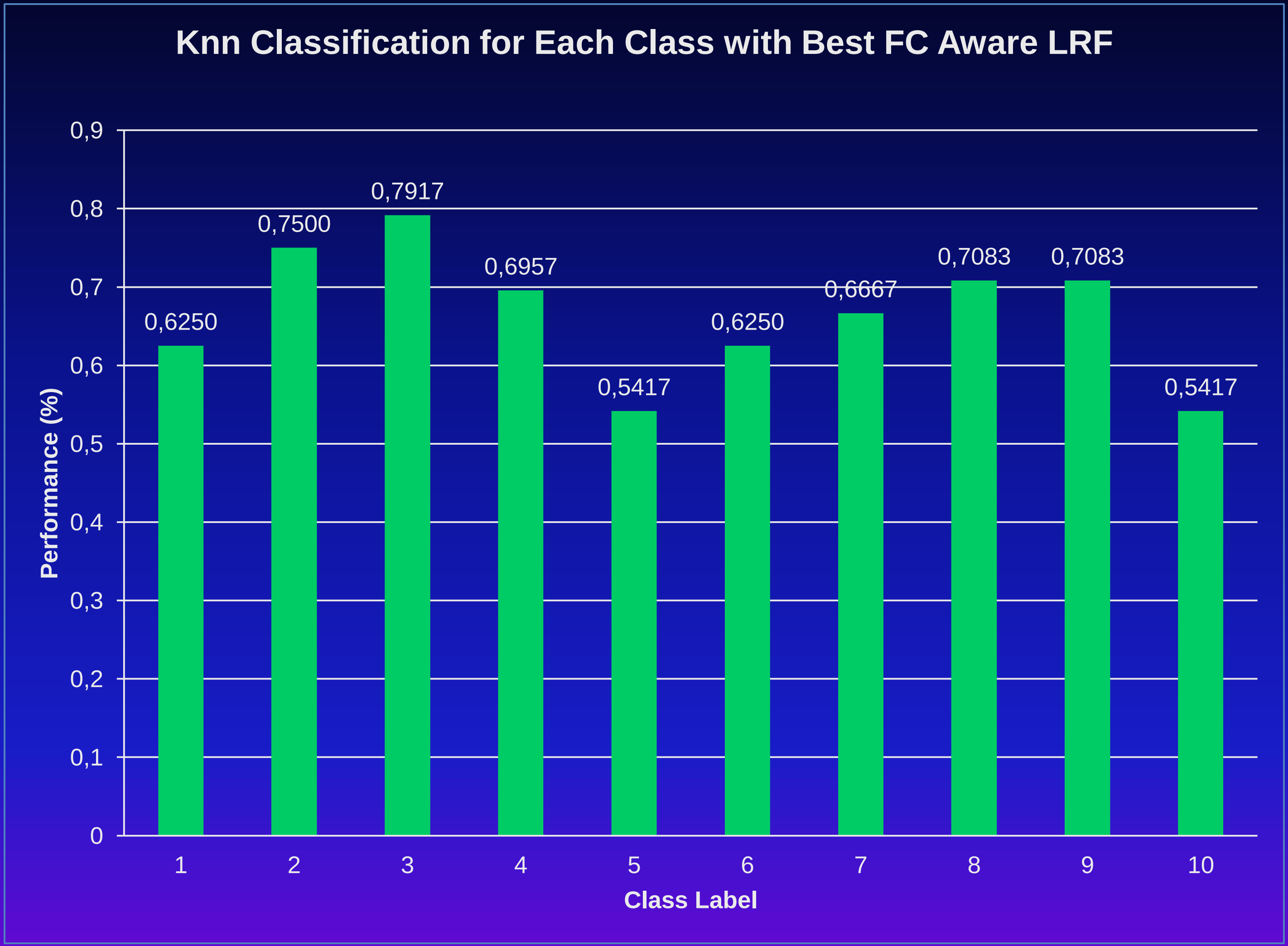
Classification Performances (%) with Spatial and Functional connectivity

%	LRF Order Values with Functional Connectivity (p)									Without FC (raw LRF)		
	2	3	4	5	6	7	8	9	10	8	9	10
k-nn	61,9384	62,3732	62,3551	**64,4565**	62,7899	64,0399	60,2717	61,9384	61,1051	56	56	**57**

Performance of Mesh Learning with Functional Connectivities, p=10

Confusion Matrix

Output Class \ Target Class	1	2	3	4	5	6	7	8	9	10	
1	14 5.9%	2 0.8%	0 0.0%	1 0.4%	0 0.0%	1 0.4%	0 0.0%	2 0.8%	2 0.8%	0 0.0%	63.6% 36.4%
2	3 1.3%	16 6.7%	0 0.0%	0 0.0%	1 0.4%	1 0.4%	3 1.3%	1 0.4%	0 0.0%	0 0.0%	64.0% 36.0%
3	0 0.0%	1 0.4%	16 6.7%	0 0.0%	1 0.4%	5 2.1%	0 0.0%	0 0.0%	2 0.8%	0 0.0%	64.0% 36.0%
4	2 0.8%	0 0.0%	1 0.4%	15 6.3%	1 0.4%	0 0.0%	0 0.0%	1 0.4%	3 1.3%	2 0.8%	60.0% 40.0%
5	2 0.8%	2 0.8%	0 0.0%	0 0.0%	13 5.4%	0 0.0%	0 0.0%	1 0.4%	0 0.0%	1 0.4%	68.4% 31.6%
6	0 0.0%	0 0.0%	2 0.8%	1 0.4%	1 0.4%	14 5.9%	3 1.3%	1 0.4%	0 0.0%	1 0.4%	60.9% 39.1%
7	0 0.0%	1 0.4%	0 0.0%	1 0.4%	0 0.0%	1 0.4%	15 6.3%	2 0.8%	1 0.4%	2 0.8%	65.2% 34.8%
8	1 0.4%	0 0.0%	3 1.3%	0 0.0%	2 0.8%	1 0.4%	1 0.4%	15 6.3%	0 0.0%	0 0.0%	65.2% 34.8%
9	2 0.8%	0 0.0%	1 0.4%	5 2.1%	3 1.3%	0 0.0%	1 0.4%	0 0.0%	15 6.3%	5 2.1%	46.9% 53.1%
10	0 0.0%	2 0.8%	1 0.4%	0 0.0%	2 0.8%	1 0.4%	1 0.4%	1 0.4%	1 0.4%	13 5.4%	59.1% 40.9%
	58.3% 41.7%	66.7% 33.3%	66.7% 33.3%	65.2% 34.8%	54.2% 45.8%	58.3% 41.7%	62.5% 37.5%	62.5% 37.5%	62.5% 37.5%	54.2% 45.8%	61.1% 38.9%



Her sınıf için elde edilen en yüksek performans değeri

Conclusion

- Mesh Learning model
  - allows us to identify and differentiate classes of information represented in the brain during memory encoding and retrieval processes
  - Functional connectivity represents the mesh better than the spatial connectivity

Implications

- We ultimately aim to read minds
  - Better understand intention
  - Better interpret feedback
  - ...
- Although we are not there yet, we are as close as we can get!

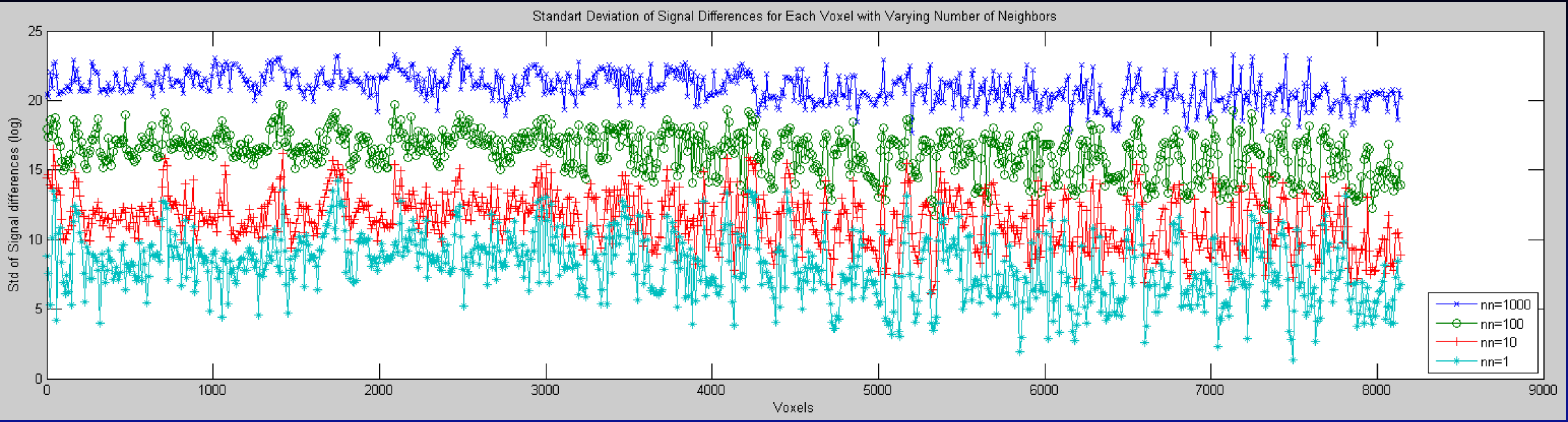
Thanks to Google

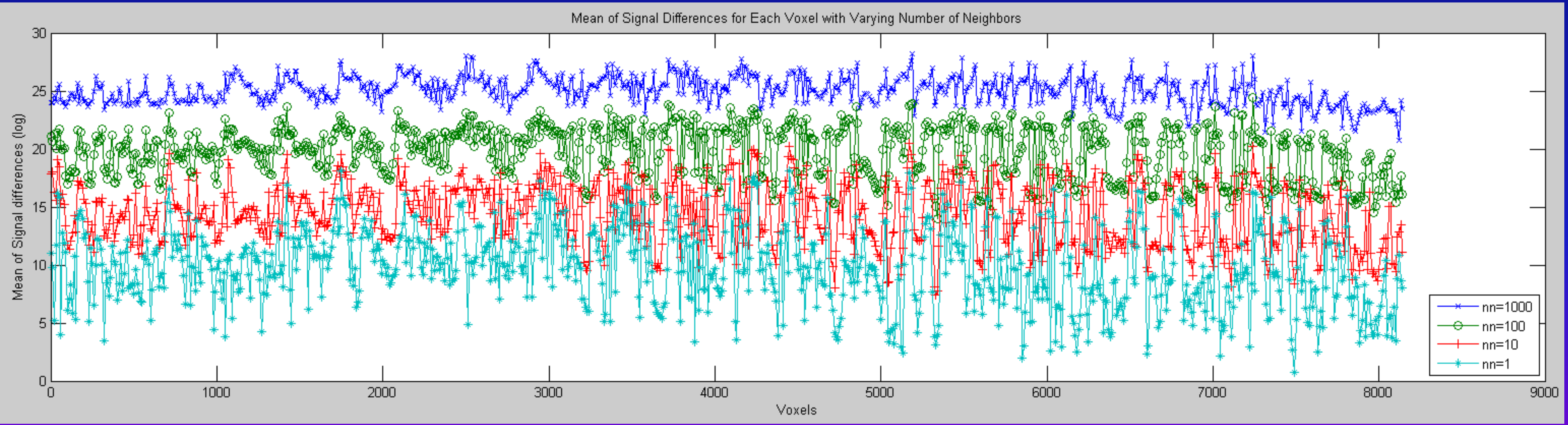
Project Website:

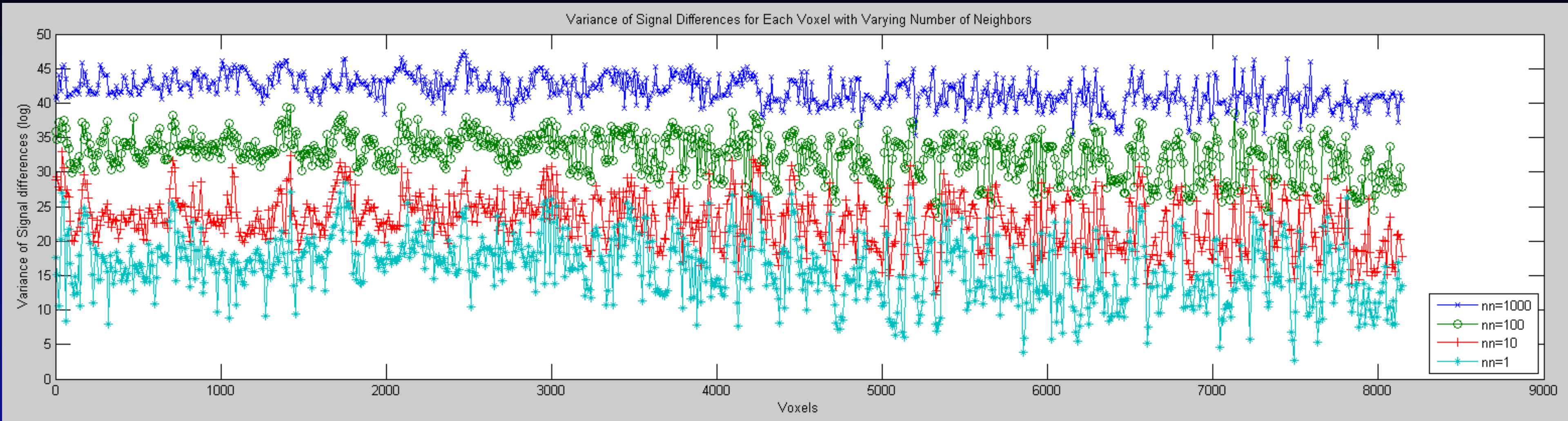
neuro.ceng.metu.edu.tr

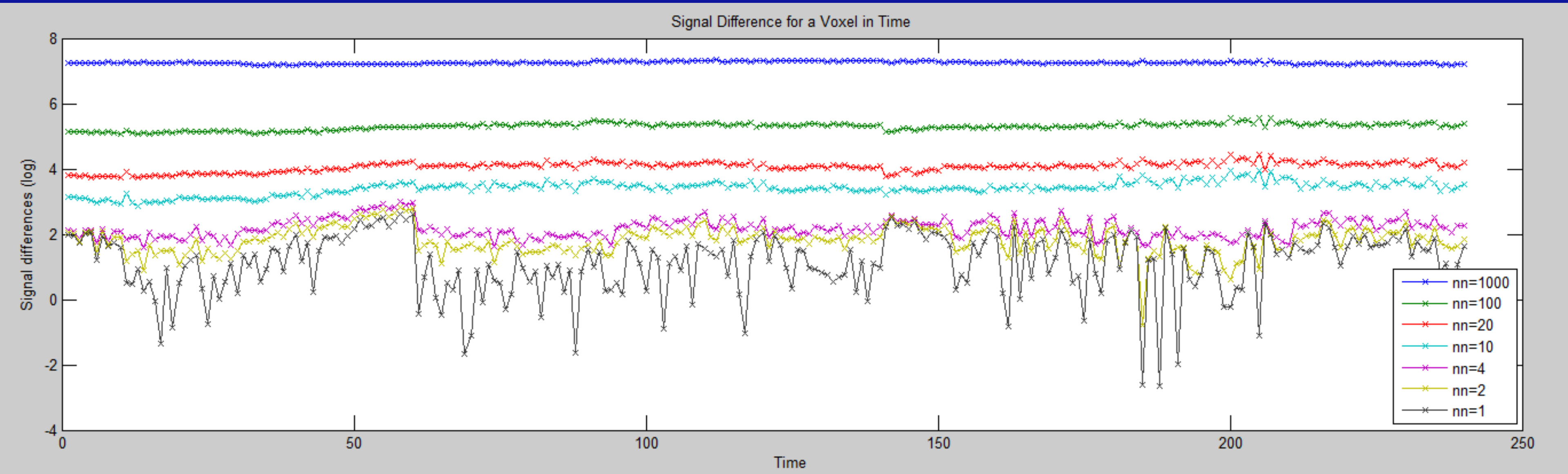
Open Issues

- Estimating the true number of clusters
- Hierarchical neighbor selection
- Network measures will be incorporated
- Combination and use of between cluster metrics


Standart Deviation of Signal Differences for Each Voxel with Varying Number of Neighbors
Std of Signal differences (log)
Voxels
0
5
10
15
20
25
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
nn=1000
nn=100
nn=10
nn=1


Mean of Signal Differences for Each Voxel with Varying Number of Neighbors
Mean of Signal differences (log)
Voxels
0
5
10
15
20
25
30
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
nn=1000
nn=100
nn=10
nn=1


Variance of Signal Differences for Each Voxel with Varying Number of Neighbors
Variance of Signal differences (log)
Voxels
nn=1000
nn=100
nn=10
nn=1


Signal Difference for a Voxel in Time
Signal differences (log)
Time
nn=1000
nn=100
nn=20
nn=10
nn=4
nn=2
nn=1