E9: 309 ADL 30-12-2020

http://leap.ee.iisc.ac.in/sriram/teaching/ADL2020/

Housekeeping

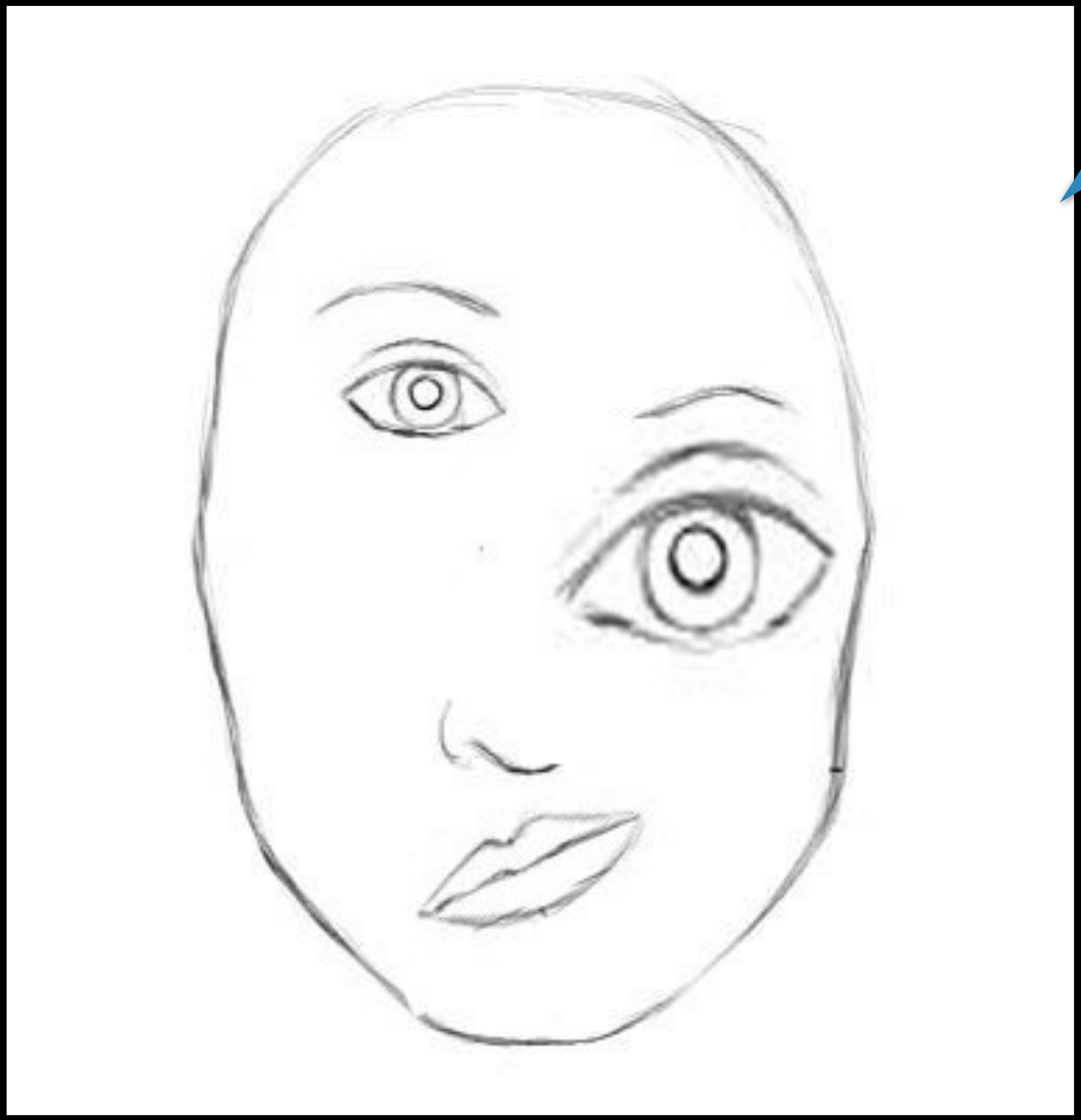
- Midterm project III
  - Abstract submission deadline (Jan 10th)
    - ✓ Evaluation after final exam (1st week of Feb)

- Final Exam (as per IISc schedule)
    - ✓ Jan 23rd afternoon!

Problem with current deep learning networks

- Question
  - Can we learn the presence of object parts and their implicit spatial information (position)
- Invariance versus equivariance
  - Invariance - resistance to translations and shifts
  - Equivariance - relationship of parts with other parts have to preserved.

Is this a face ?

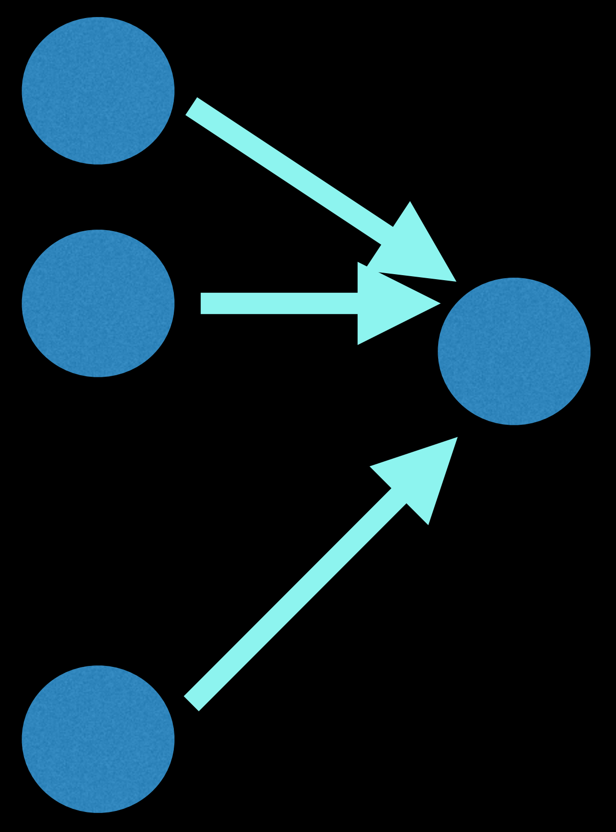


Capsule networks

- Traditional networks
  - Weigh the inputs and generate a scalar output
- Key idea in capsule networks (neurons to capsules)
  - ✓ Move inidividual neuron outputs from scalar to vector
  - ✓ Encode the probability of presence of an attribute along the magnitude of the vector output
  - ✓ Encode the pose (translation + rotation) in the angle of the vector output

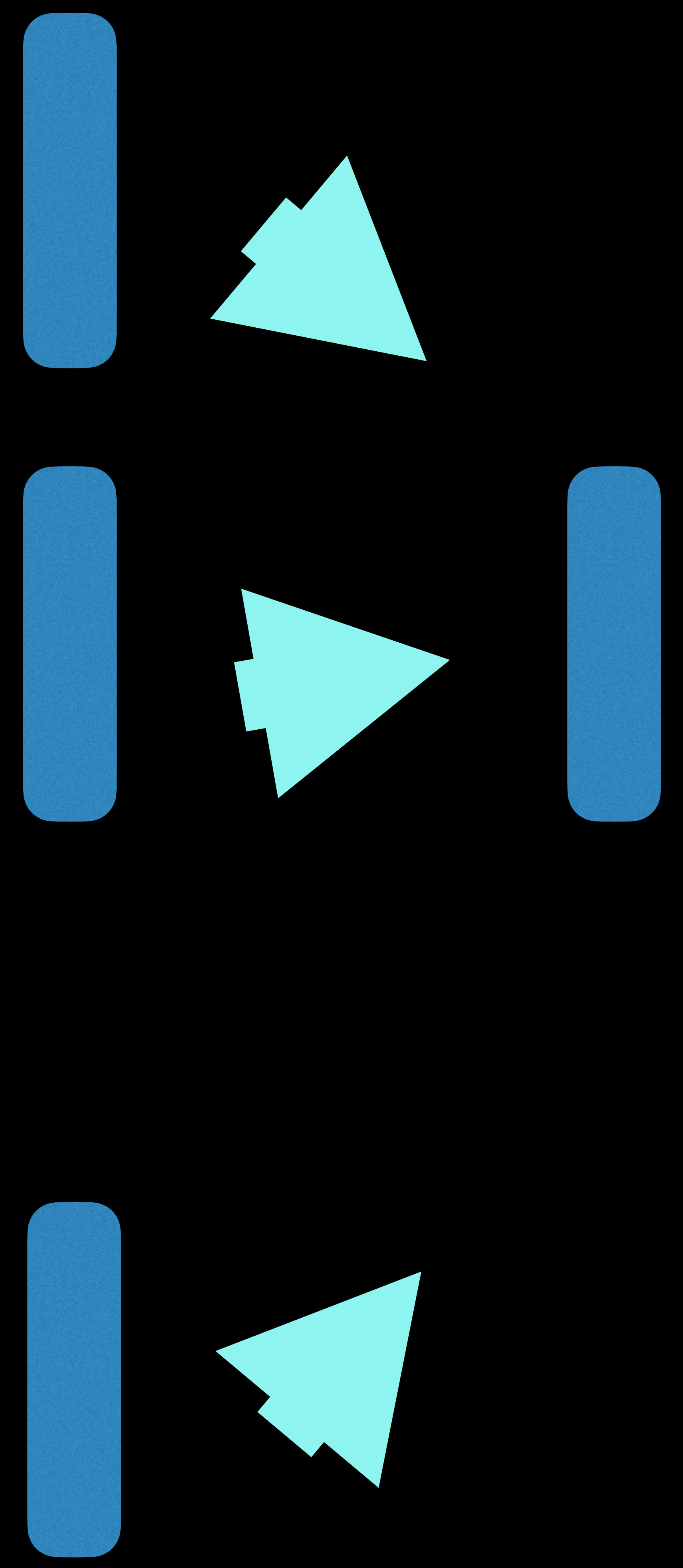
From a layer of neurons to layer of capsules

- Conventional neurons



$$v_j^{l+1} = \sum_i w_i^l u_i^l$$

- Capsule network



Prediction vector

$$\hat{\mathbf{u}}_{j|i}^{l+1} = \mathbf{W}_{ij}\mathbf{u}_i^l$$

Coupling

$$\mathbf{s}_j^{l+1} = \sum_i c_{ij}\hat{\mathbf{u}}_{j|i}^{l+1}$$

Instantiation parameter

$$\mathbf{v}_j^{l+1} = Squash(\mathbf{s}_j^{l+1})$$

Capsule network

✳ Squashing non-linearity

$$Squash(\mathbf{s}_j^{l+1}) = \frac{||\mathbf{s}_j||^2}{1+||\mathbf{s}_j||^2}\frac{\mathbf{s}_j}{||\mathbf{s}_j||}$$

➡ Makes the vector magnitude range from (0,1)

✓ Interpretation of the length of the vector as a probability of the presence of a particular object part

➡ Preserves the direction of the original vector.

Capsule network

- Coupling

$$c_{ij} = \frac{e^{b_{ij}}}{\sum_k e^{b_{ik}}}$$

- The coefficients

$$b_{ij} \leftarrow b_{ij} + \mathbf{u}_{j|i}^T \mathbf{v}_j$$

  - ✓ The dot product measures the agreement of the layer output at j with the prediction made only based on i
  - ✓ Depend on the location and type of the capsule

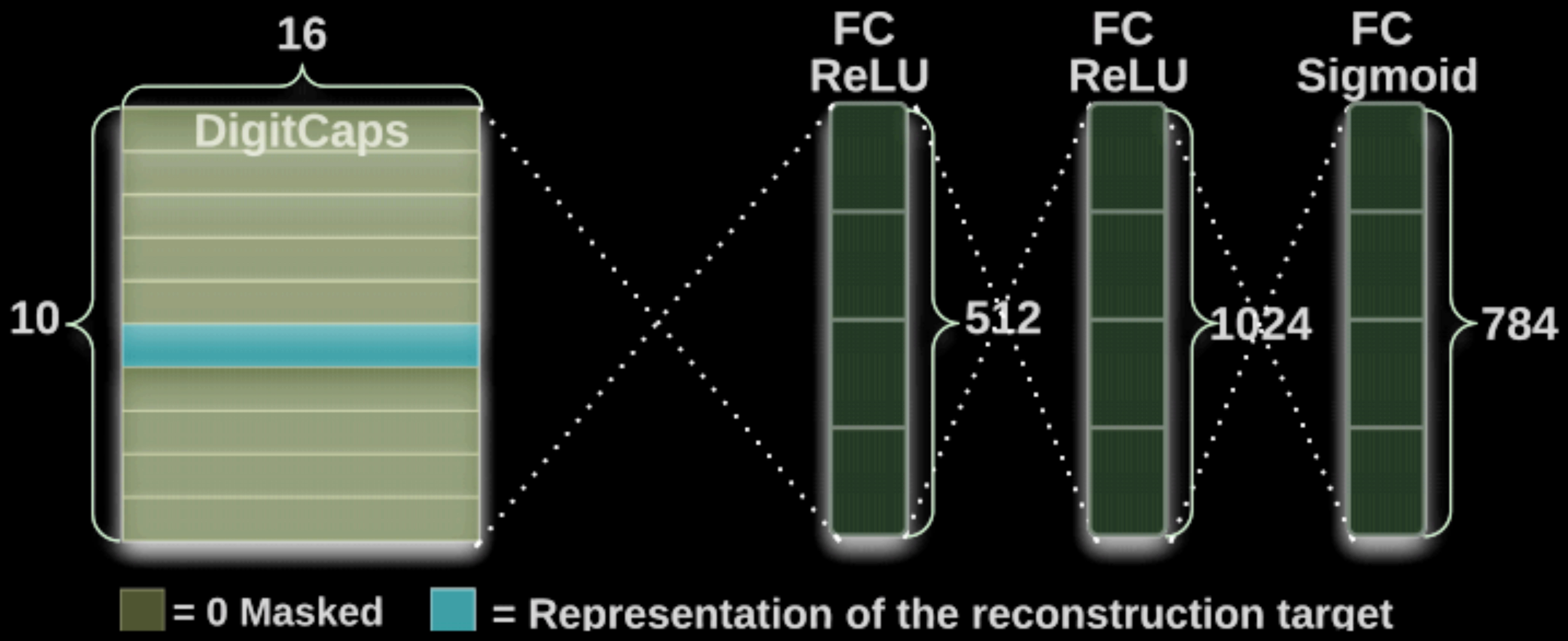
Neurons versus Capsules

Capsule vs. Traditional Neuron			
Input from low-level capsule/neuron		vector($\mathbf{u}_i$)	scalar(x_i)
Operation	Affine Transform	$\hat{\mathbf{u}}_{j\|i} = \mathbf{W}_{ij}\mathbf{u}_i$	–
	Weighting	$\mathbf{s}_j = \sum_i c_{ij}\hat{\mathbf{u}}_{j\|i}$	$a_j = \sum_i w_i x_i + b$
	Sum		
	Nonlinear Activation	$\mathbf{v}_j = \frac{\|\mathbf{s}_j\|^2}{1+\|\mathbf{s}_j\|^2}\frac{\mathbf{s}_j}{\|\mathbf{s}_j\|}$	$h_j = f(a_j)$
Output		vector($\mathbf{v}_j$)	scalar(h_j)

Convolutional capsule networks

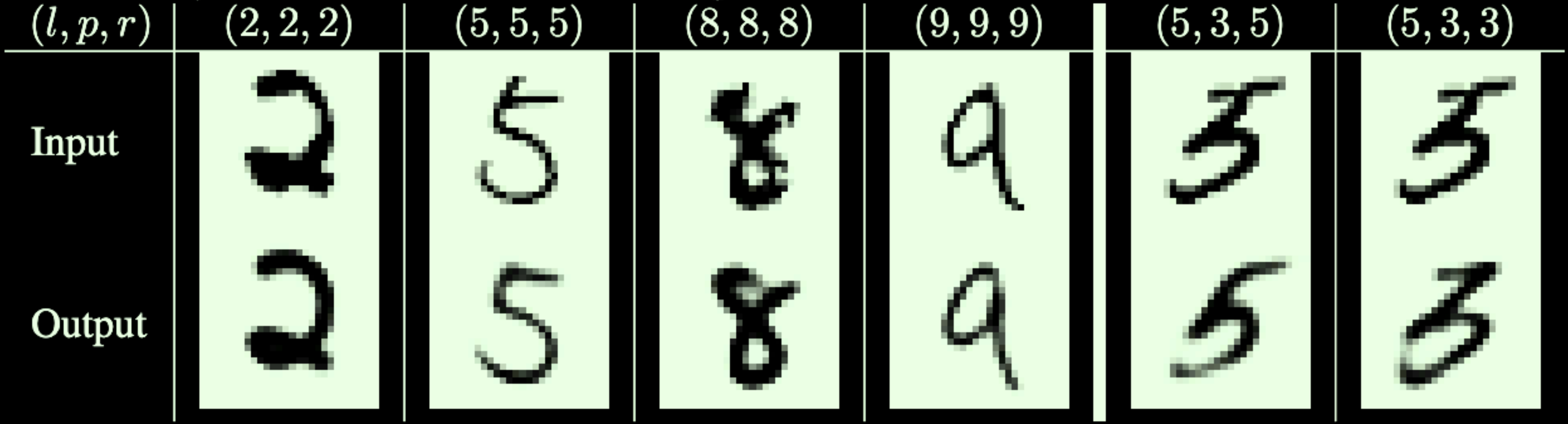
Reconstruction as a regularization loss

* Use the final capsules to reconstruct the digit images



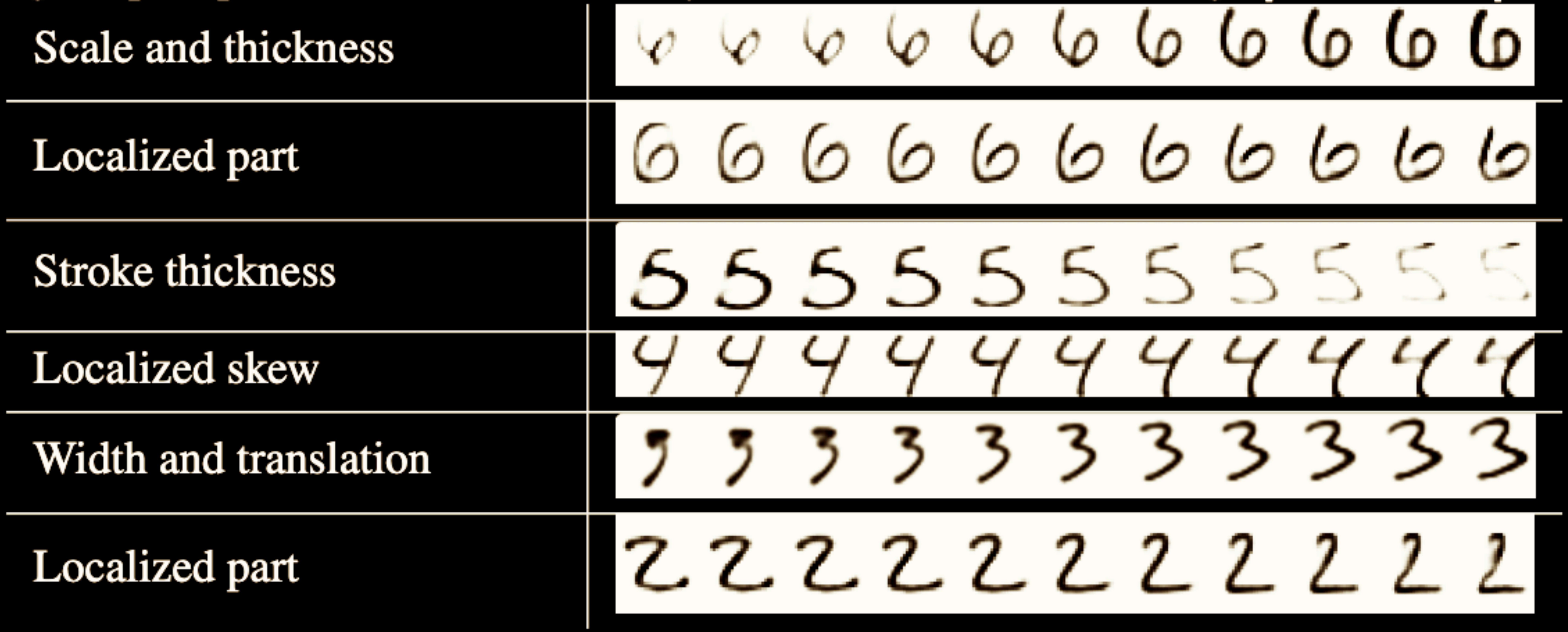
Capsule networks

Figure 3: Sample MNIST test reconstructions of a CapsNet with 3 routing iterations. (l, p, r) represents the label, the prediction and the reconstruction target respectively. The two rightmost columns show two reconstructions of a failure example and it explains how the model confuses a 5 and a 3 in this image. The other columns are from correct classifications and shows that model preserves many of the details while smoothing the noise.



Understanding the capsule output

Figure 4: Dimension perturbations. Each row shows the reconstruction when one of the 16 dimensions in the DigitCaps representation is tweaked by intervals of 0.05 in the range $[-0.25, 0.25]$.



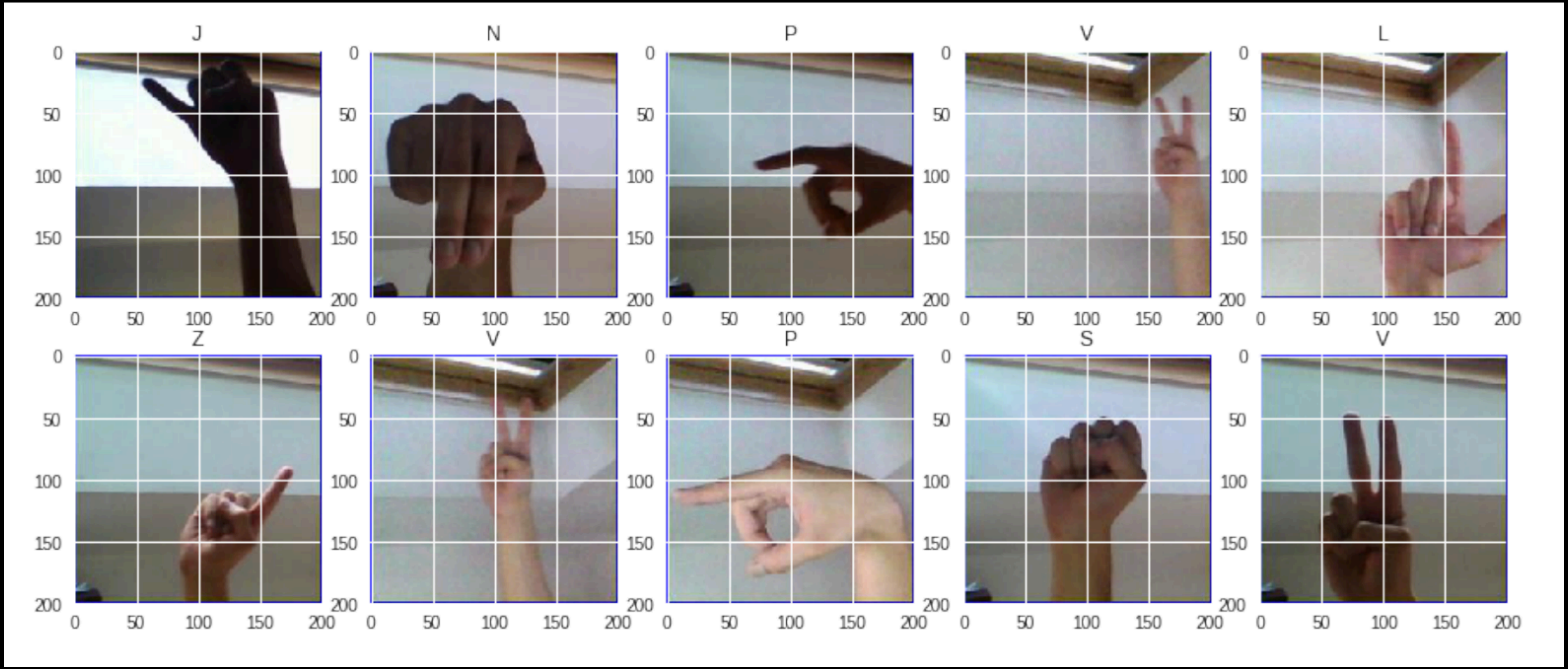
Capsule network performance

Table 1: CapsNet classification test accuracy. The MNIST average and standard deviation results are reported from 3 trials.

Method	Routing	Reconstruction	MNIST (%)	MultiMNIST (%)
Baseline	-	-	0.39	8.1
CapsNet	1	no	$0.34_{\pm 0.032}$	-
CapsNet	1	yes	$0.29_{\pm 0.011}$	7.5
CapsNet	3	no	$0.35_{\pm 0.036}$	-
CapsNet	3	yes	$\mathbf{0.25}_{\pm 0.005}$	**5.2**

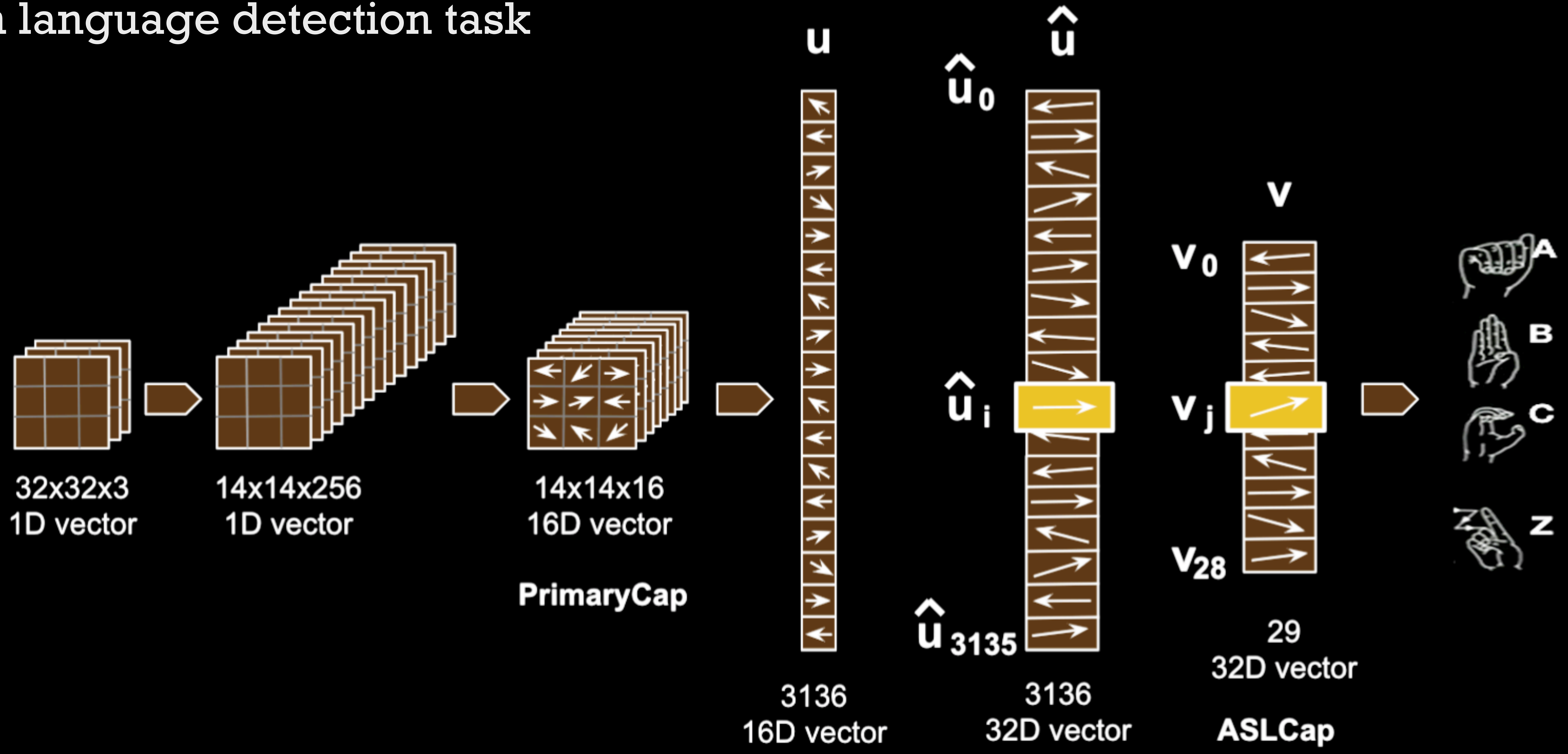
Automatic Sign Language Detection Task

- Sign language detection task

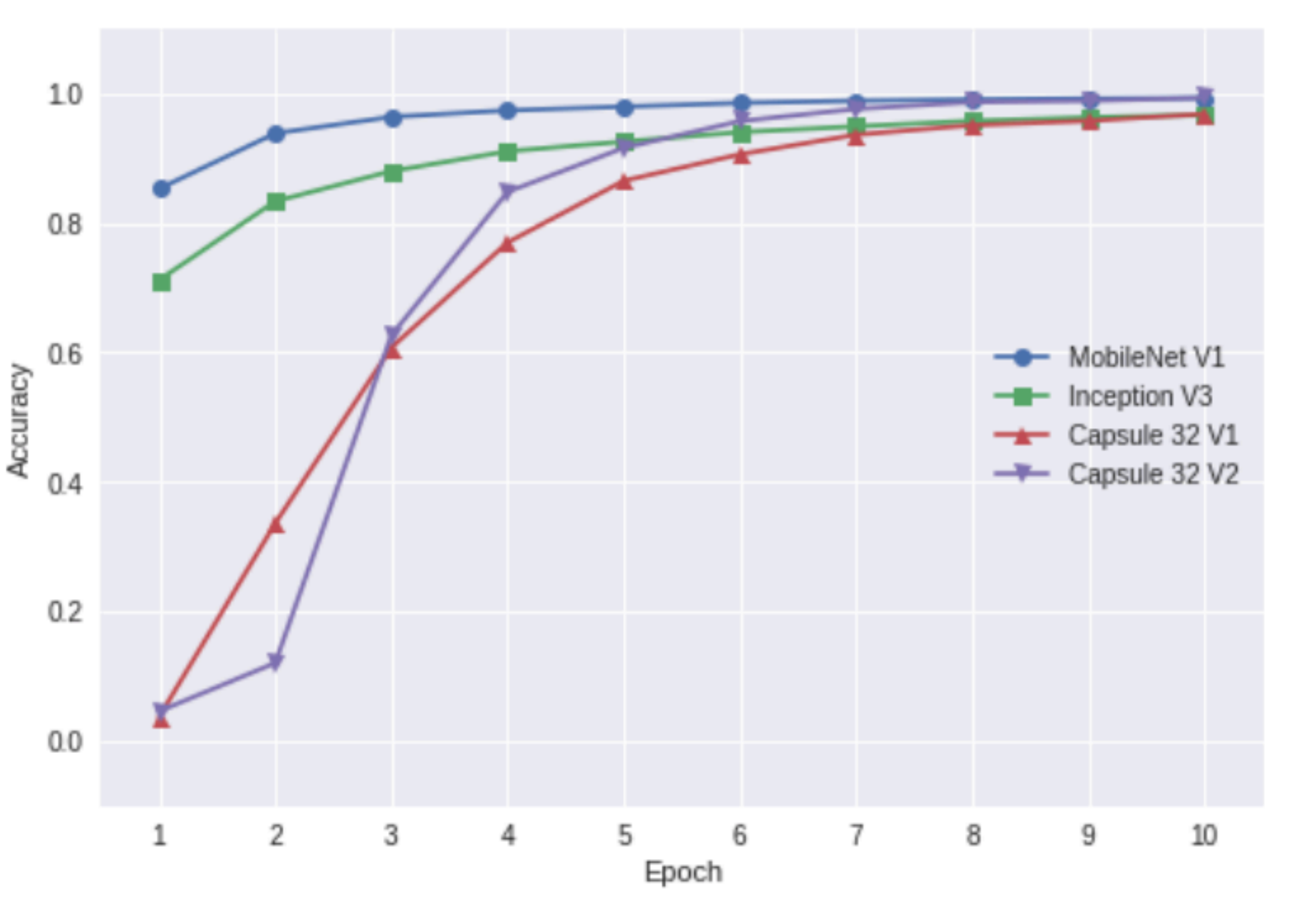


Automatic Sign Language Detection Task

* Sign language detection task



Comparing capsule networks with other architectures



Deep learning on graphs

Graph definition (Undirected)

- (V,E) denotes the vertices and edges in a graph
  - ✓ |V| - denotes the number of vertices
- $\mathbf{A} = [a_{ij}]$ denote the adjacency matrix of the graph
  - ✓ Similarity or affinity of the vertices.
  - ✓ Symmetric and typically sparse matrix
- $\mathbf{D} = diag[d_1, d_2, ..., d_N]$ denote the degree matrix

$$d_i = \sum_j a_{ij}$$

Defining graphs

- Input features

$$\mathbf{x}_i \in \mathcal{R}^D \quad i = 1...N$$

- Input feature space

$$\mathbf{X} \in \mathcal{R}^{N \times D}$$

- Hidden layer initialization

$$\mathbf{H}^0 = \mathbf{X}$$

3-steps in Graph convolutional networks

✳ I. Feature propagation

$$\bar{\mathbf{h}}_i^k = \frac{\mathbf{h}_i}{d_i + 1} + \sum_{j=1}^{N} \frac{a_{ij}}{\sqrt{(d_i+1)(d_j+1)}} \mathbf{h}_j^{k-1}$$

$$\mathbf{S} = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}},$$

$$\bar{\mathbf{H}}^{(k)} \leftarrow \mathbf{S}\mathbf{H}^{(k-1)}.$$

Graph convolutions

- Non-linearity and activations

$$\mathbf{H}^{(k)} \leftarrow \mathrm{ReLU}\left(\bar{\mathbf{H}}^{(k)}\boldsymbol{\Theta}^{(k)}\right)$$

$$\hat{\mathbf{Y}}_{\mathrm{GCN}} = \mathrm{softmax}\left(\mathbf{S}\mathbf{H}^{(K-1)}\boldsymbol{\Theta}^{(K)}\right)$$