Dynamic Routing Between Capsules

Beyond Convolutional Neural Networks

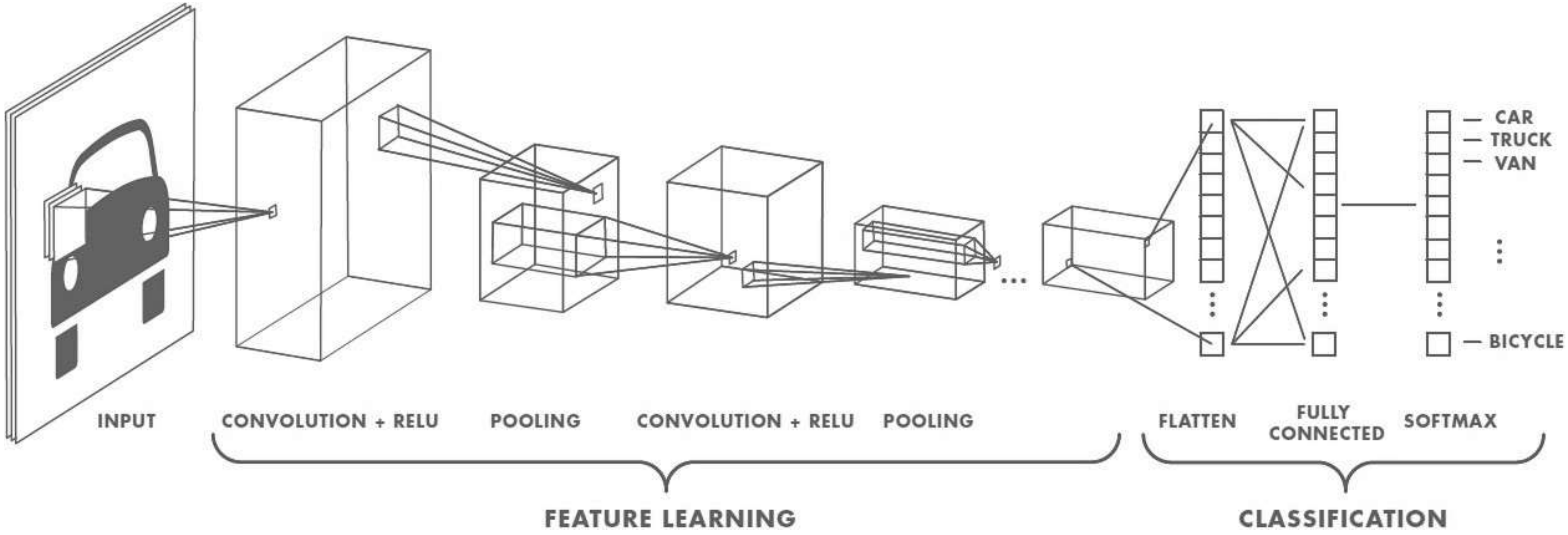
Kyu-Hwan Jung, Ph.D
Co-founder and CTO, VUNO Inc.

Background

Modern CNNs

Stack of Layers with Convolution, Subsampling and Nonlinearity Operations

- **Convolution Layer**
  - Filtering of unimportant information and extraction of salient local feature.
- **Subsampling Layer**
  - Introduction of local transition invariance, reduction of computation and enlargement of receptive field.
- **Nonlinearity**
  - Increase of capacity to model more complex input-output relationship.

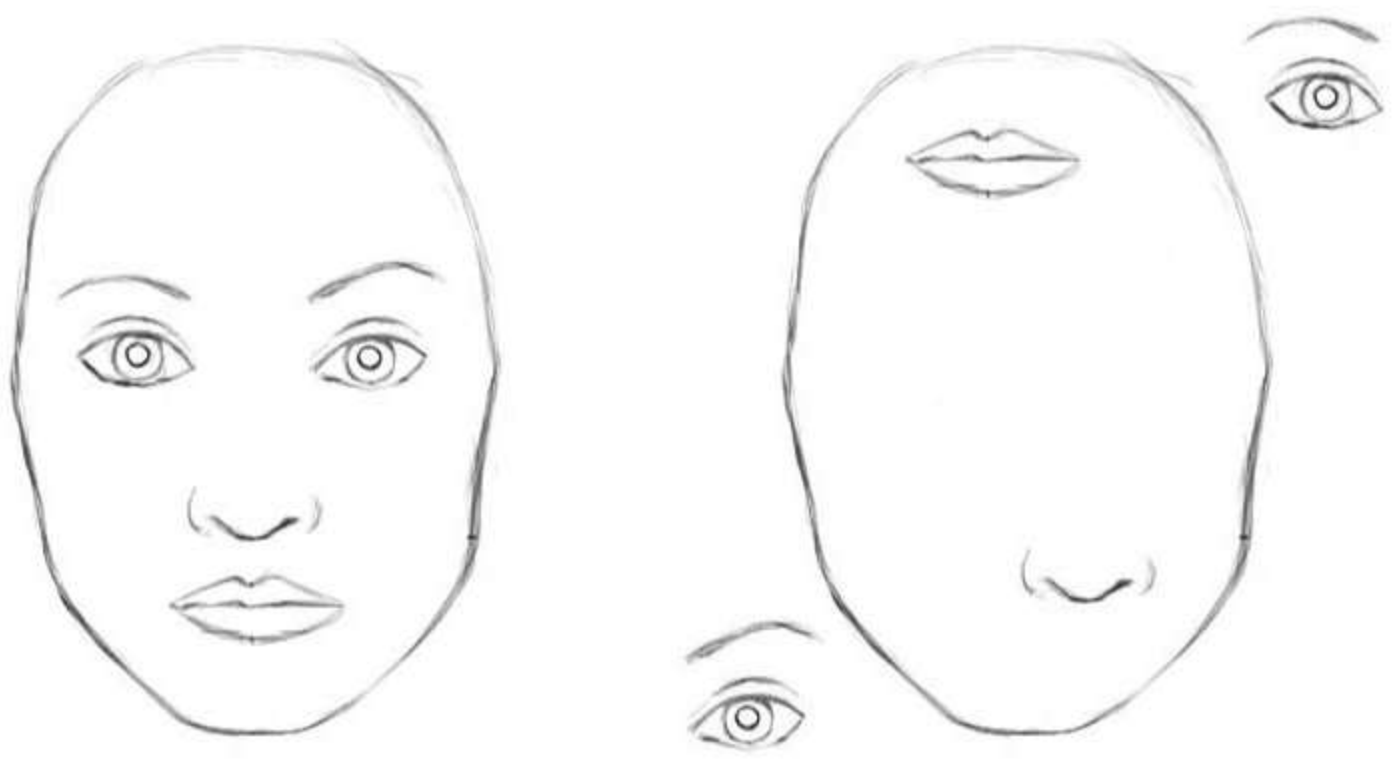


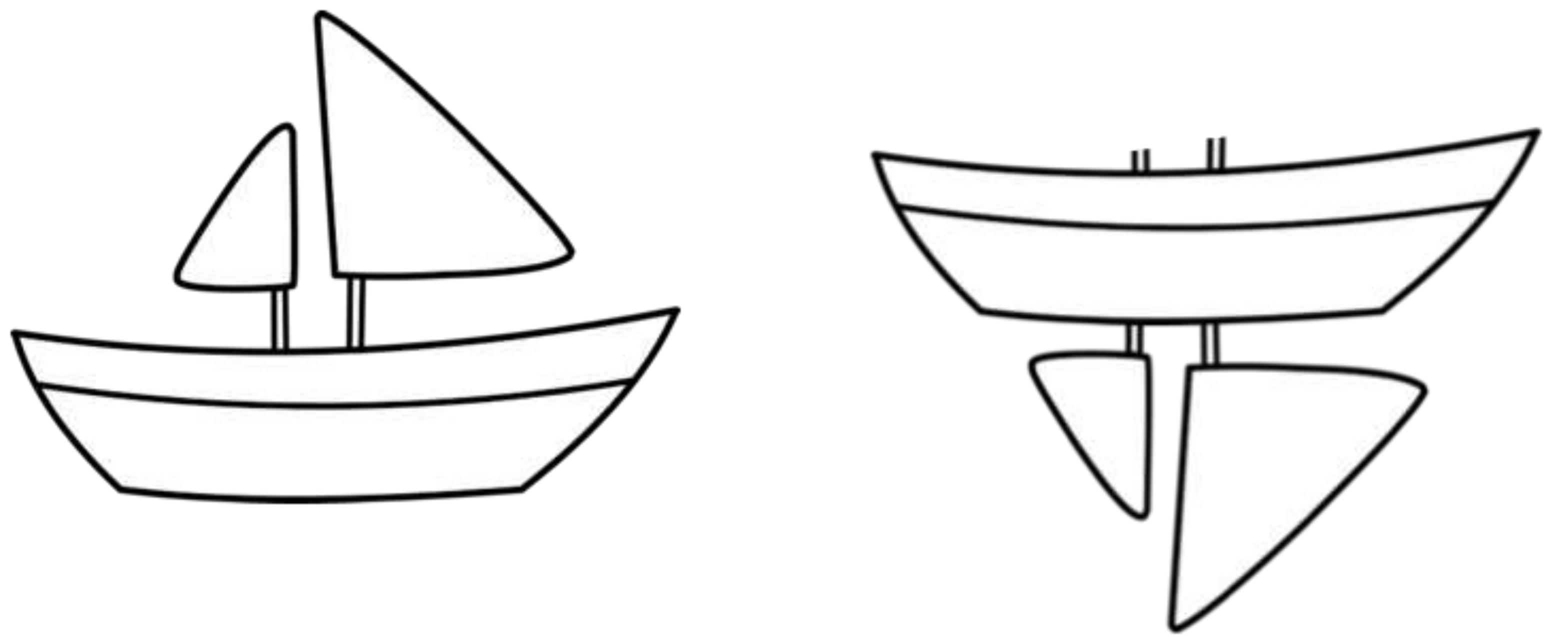
What Geoff. Hinton Says

Mistake and Disaster

“The pooling operation used in convolutional neural networks is a big mistake and the fact that it works so well is a disaster.”

“Internal data representation of a convolutional neural network does not take into account important spatial hierarchies between simple and complex objects.”

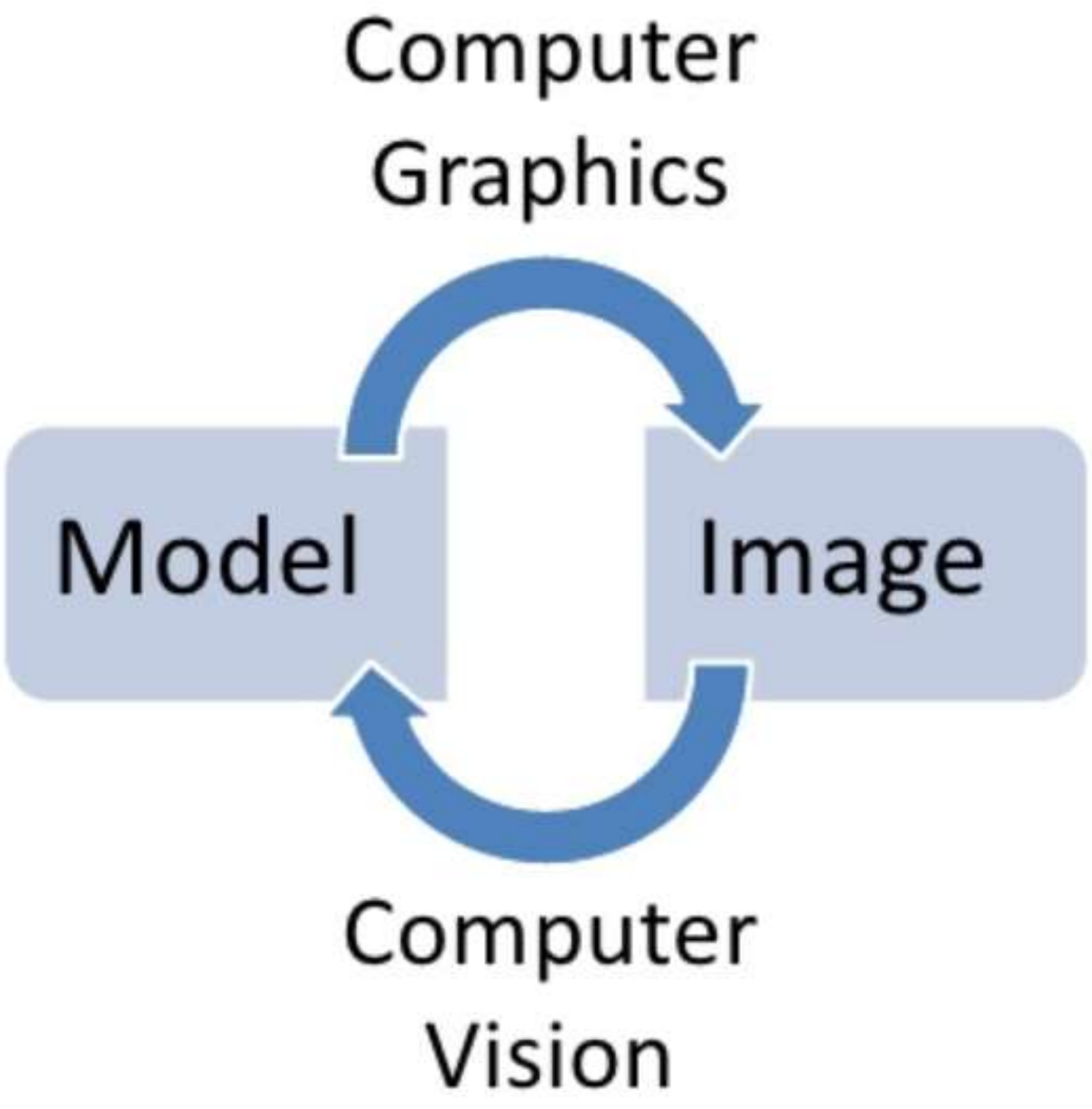


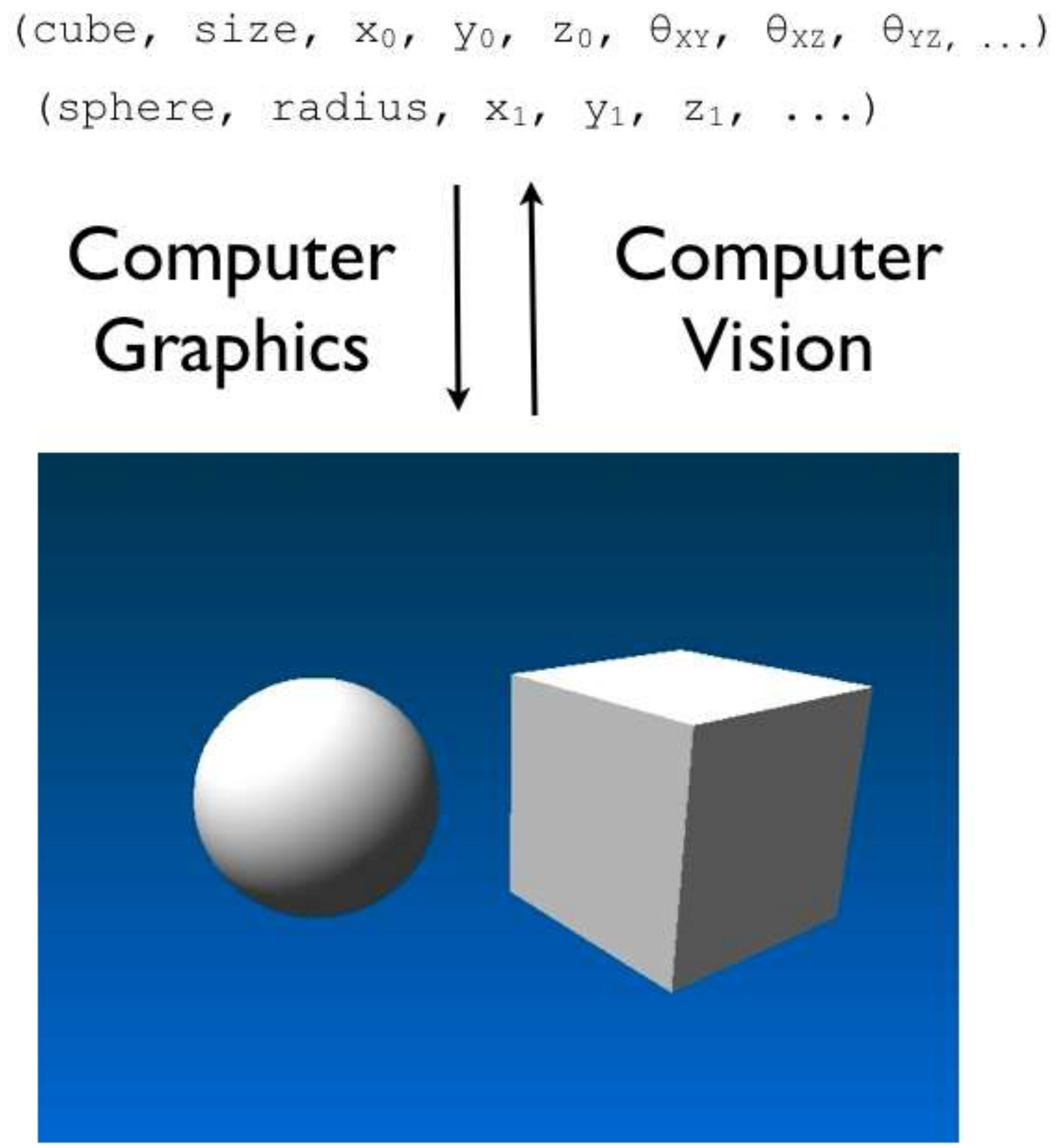


Inverse Graphical Approach

Computer Graphics vs Computer Vision

- What our brain does when we see the world is ‘inverse graphics’
  - Extraction of hierarchical representation of the visual information
  - The representation is ‘equivariance’, not ‘invariance’ to changes in viewpoint




(cube, size, x_0, y_0, z_0, θ_{XY}, θ_{XZ}, θ_{YZ}, ...)
(sphere, radius, x_1, y_1, z_1, ...)
Computer Graphics
Computer Vision

Invariance vs Equivariance

Better Representation via Better Objective

- Sub-sampling tries to make the neural activities invariant for small changes in viewpoint.
  - This is a silly goal, motivated by the fact that the final label needs to be viewpoint-invariant.
- Its better to aim for equivariance: Changes in viewpoint lead to corresponding changes in neural activities.
  - In the perceptual system, its the weights that code viewpoint-invariant knowledge, not the neural activities.

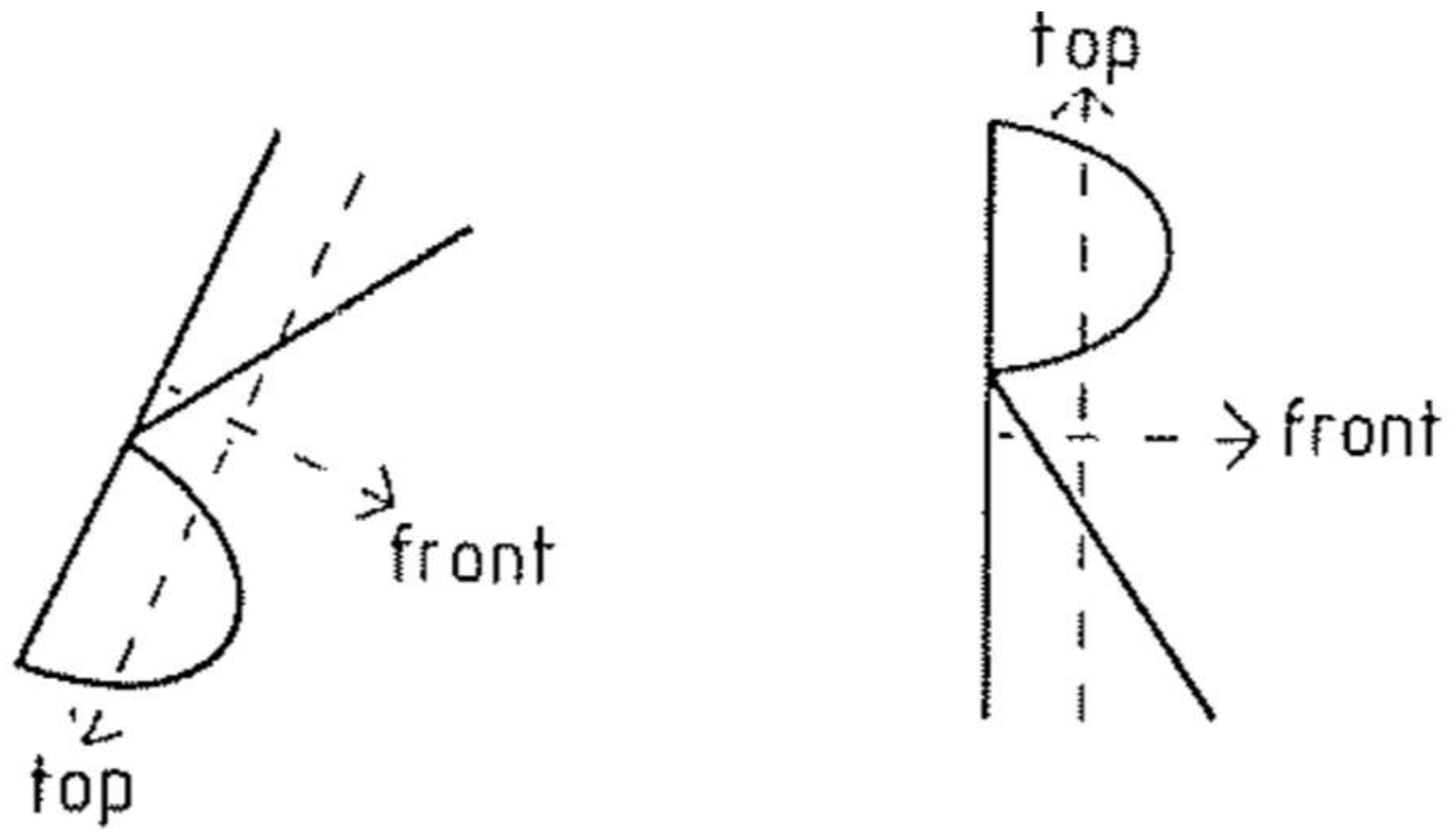
Your brain can easily recognize this is the same object, even though all photos are taken from different angles. CNNs do not have this capability.

Capsules

Arguments on Pooling

Pros and Cons

- Transitional invariance
  - The output is unchanged by small shift of object or parts
- Reduction of input
  - Allowing more types of feature in the next layer



- Bad fit to psychology of shape perception
  - We have intrinsic coordinate frames to objects and use it while recognizing things.
- Solution to wrong problem
  - We need equivariance not invariance. Disentangling rather than discarding.
- Fails to use underlying linear structure
  - We need to make used of the natural linear manifold which will help us to extrapolate.
- Poor wary to do dynamic routing
  - We need to route each part of the input to the neurons that know how to deal with it. Routing via most active neuron is primitive.

Two types of equivariance

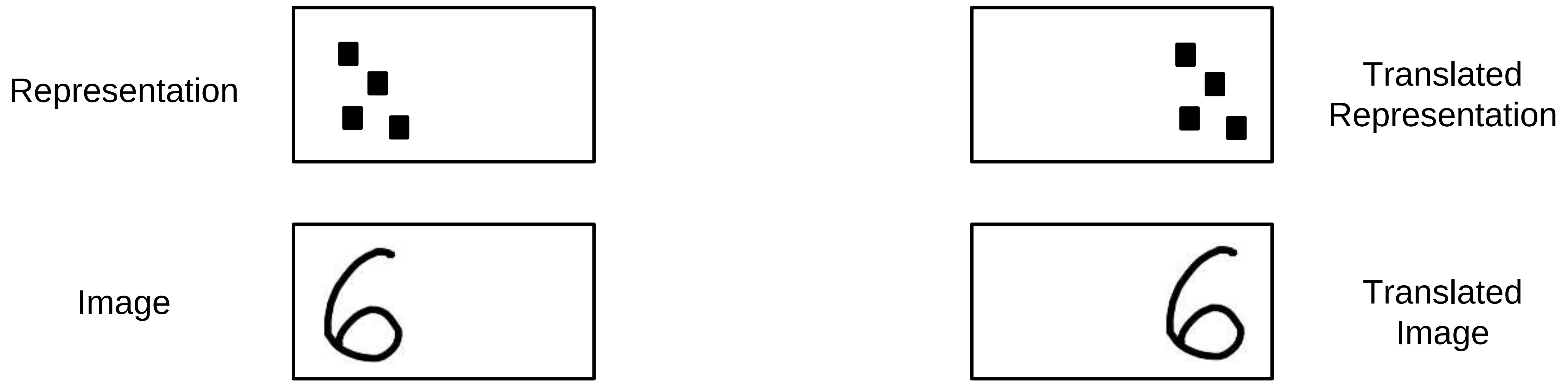
Place-coding vs Rate-coding

- If a low-level part moves to a very different position it will be represented by a different capsule
  - ‘Place-coded’ equivariance

- If a part only moves a small distance it will be represented by the same capsule but the pose of outputs of the capsule will change
  - ‘Rate-coded’ equivariance

- Higher-level capsules have bigger domains so low-level ‘place-coded’ equivariance gets converted into high-level ‘rate-coded’ equivariance.

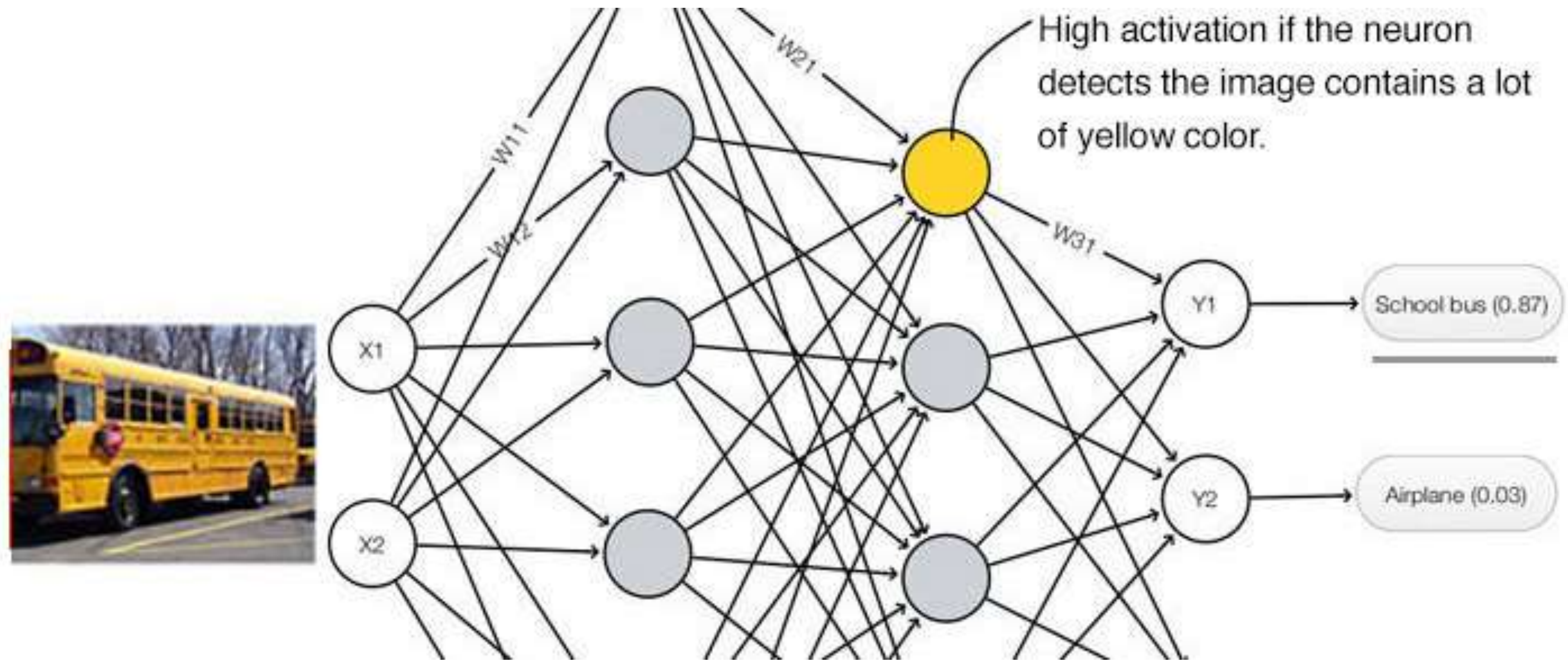
- Without sub-sampling, CNNs can give ‘place-coded’ equivariance for discrete translations.

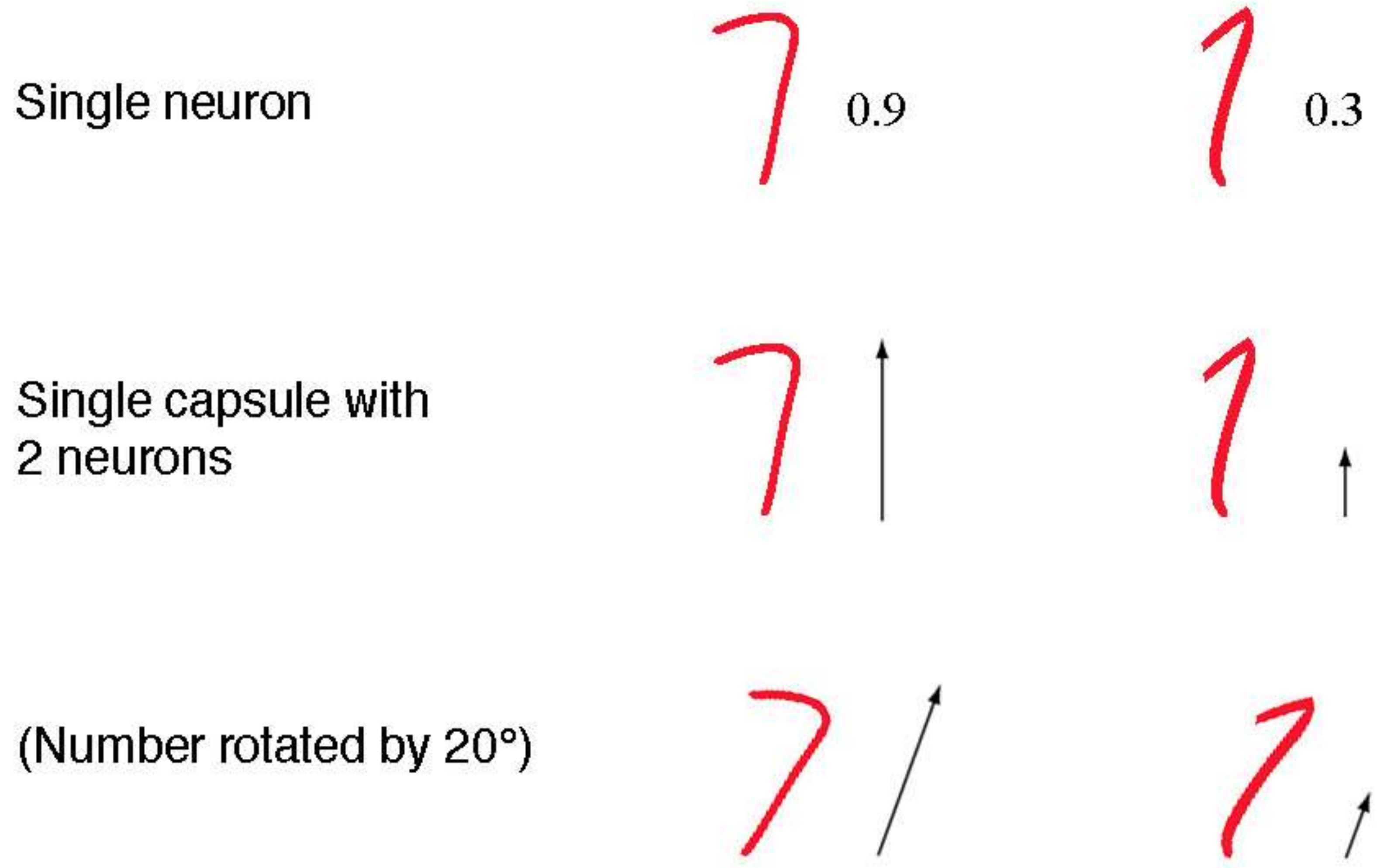


Neuron vs Capsule

- **Neuron**
  - The activation of a specific neuron = the likelihood of specific feature in the input image.
  - We need specialized mechanism(e.g. generative models) to make each neuron represent parameters of feature.

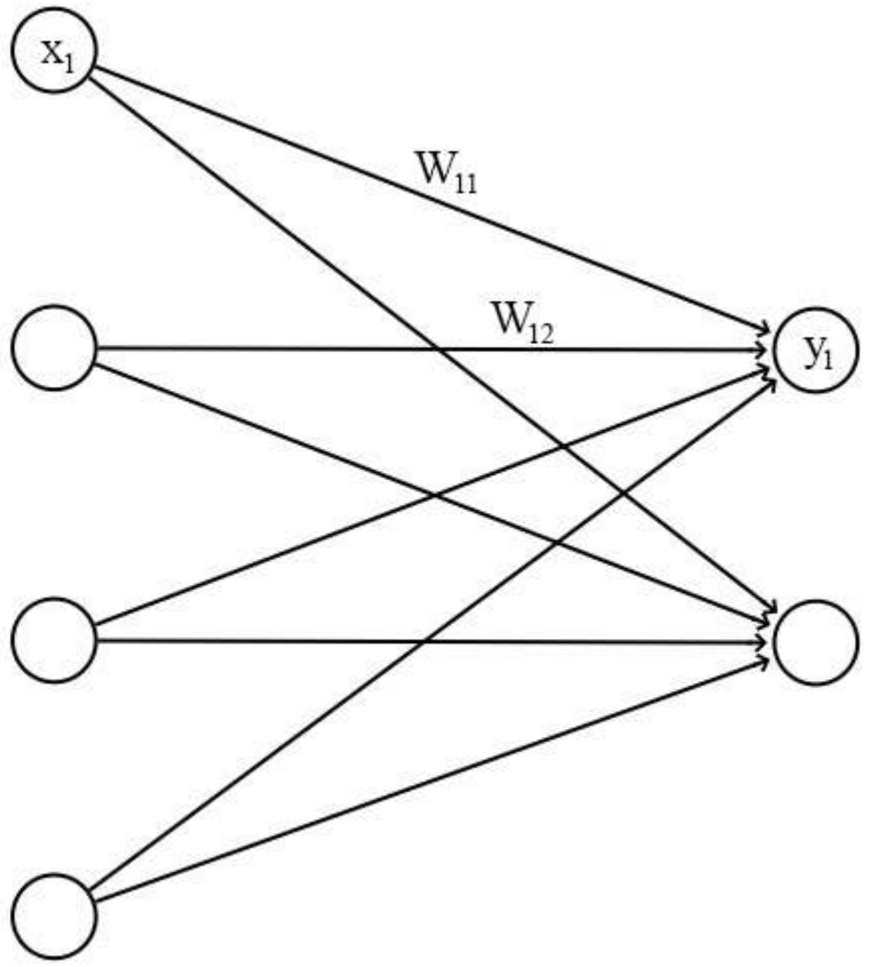
- **Capsule**
  - A group of neurons which not only capture the likelihood of specific features but also the parameters related to the features.
  - The magnitude of activity vector represents the probability of detecting specific features and its orientation represents its parameters or properties (position, orientation, scale, deformation, color, hue, texture …)


High activation if the neuron detects the image contains a lot of yellow color.
W11
W12
W21
W31
X1
X2
Y1
Y2
School bus (0.87)
Airplane (0.03)



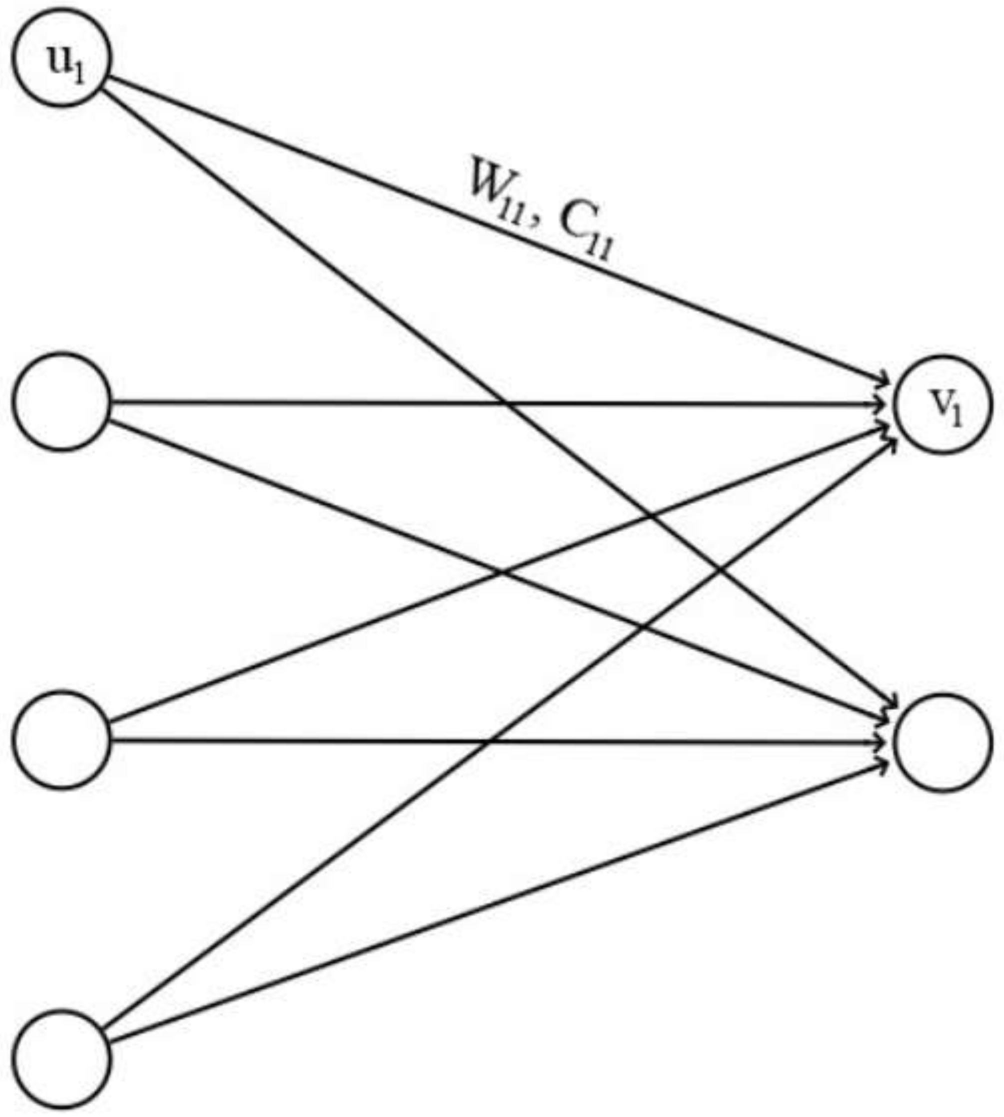
Neuron vs Capsule

- **Neuron**
  - The inputs of lower layers and the outputs of upper layers are **scalars**.
  - After aggregating inputs multiplied by weights, nonlinearity function such as ReLU is applied.



$$z_j = \sum_i W_{ij} x_i$$
$$y_j = ReLU(z_j)$$

- **Capsule**
  - The inputs of the lower layers and the outputs of upper layers are **vectors**.
  - Instead of using nonlinearity function such as ReLU, we used **squashing** function.



$$\hat{u}_{j|i} = W_{ij} u_i$$
$$s_j = \sum_i c_{ij} \hat{u}_{j|i}$$

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|}$$

Dynamic Routing Algorithm

Participants in the Algorithm

- **Prediction vector**
  - With the previous capsule output u_i and transformation matrix W_{ij} , we computer prediction vector as

$$\hat{u}_{j|i} = W_{ij} u_i$$

- **The capsule output of next layer**
  - Then the capsule output of the next layer v_j is computed as

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|} \text{ , where } s_j = \sum_i c_{ij} \hat{u}_{j|i}$$

- **Coupling coefficient**
  - Here, c_{ij} are called coupling coefficient which is trained with dynamic routing algorithm.
  - We impose restriction that $\sum_i c_{ij}$ = 1 which is achieved by softmax function of relevancy or similarity score b_{ij} which is initialized with zeros and progressively updated as follows(which reminds me of Hebbıan learning rule):

$$similarity = \hat{u}_{j|i} \cdot v_j$$

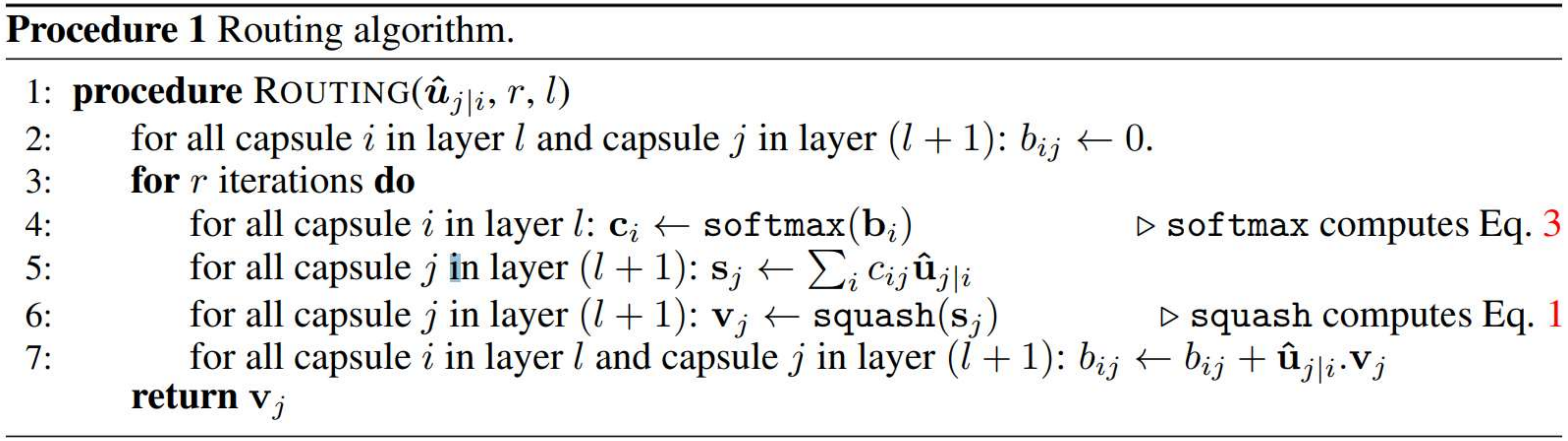
$$b_{ij} \leftarrow b_{ij} + similarity$$

$$c_{ij} = \frac{\exp b_{ij}}{\sum_k \exp b_{ik}}$$

Dynamic Routing Algorithm

Routing Algorithm and Loss Function

- **Routing Algorithm**
  - The pseudo-code of routing algorithm is as follows:



Procedure 1 Routing algorithm.

```
procedure ROUTING($\hat{\mathbf{u}}_{j|i}$, $r$, $l$)
    for all capsule $i$ in layer $l$ and capsule $j$ in layer $(l+1)$: $b_{ij} \leftarrow 0$.
    for $r$ iterations do
        for all capsule $i$ in layer $l$: $\mathbf{c}_i \leftarrow \texttt{softmax}(\mathbf{b}_i)$        ▷ softmax computes Eq. 3
        for all capsule $j$ in layer $(l+1)$: $\mathbf{s}_j \leftarrow \sum_i c_{ij}\hat{\mathbf{u}}_{j|i}$
        for all capsule $j$ in layer $(l+1)$: $\mathbf{v}_j \leftarrow \texttt{squash}(\mathbf{s}_j)$        ▷ squash computes Eq. 1
        for all capsule $i$ in layer $l$ and capsule $j$ in layer $(l+1)$: $b_{ij} \leftarrow b_{ij} + \hat{\mathbf{u}}_{j|i}.\mathbf{v}_j$
    return $\mathbf{v}_j$
```

- **Loss Function**
  - For each digit capsule k , the loss function is margin loss as

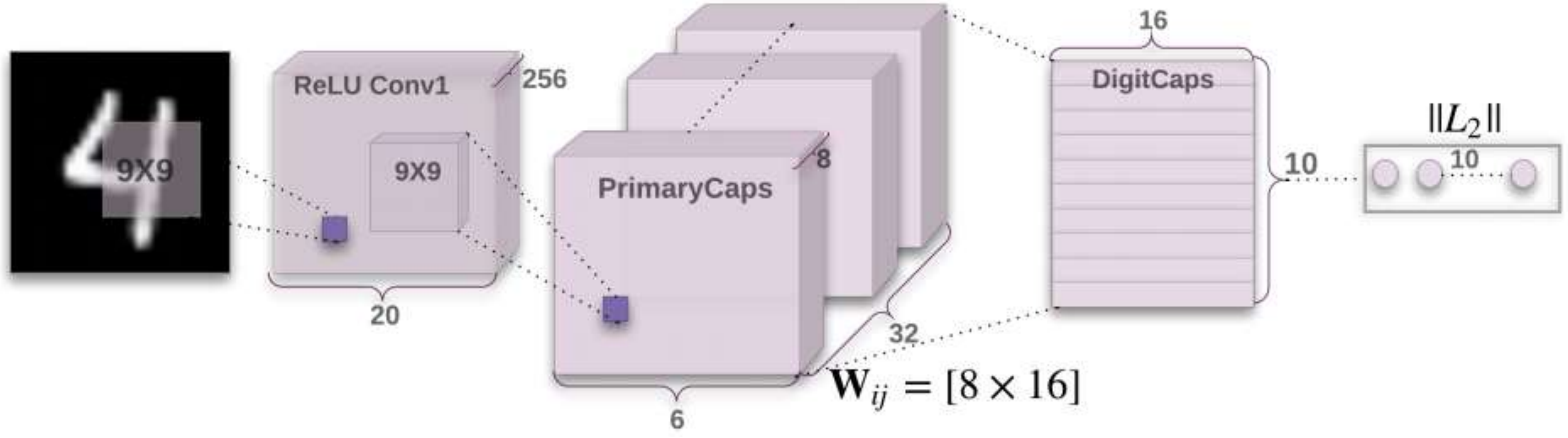
$$L_k = T_k \ \max(0, m^+ - ||\mathbf{v}_k||)^2 + \lambda\,(1 - T_k)\ \max(0, ||\mathbf{v}_k|| - m^-)^2$$

where $T_k = 1$ when digit k is present and $m^+ = 0.9$ $m^- = 0.1$. The default $\lambda = 0.5.$

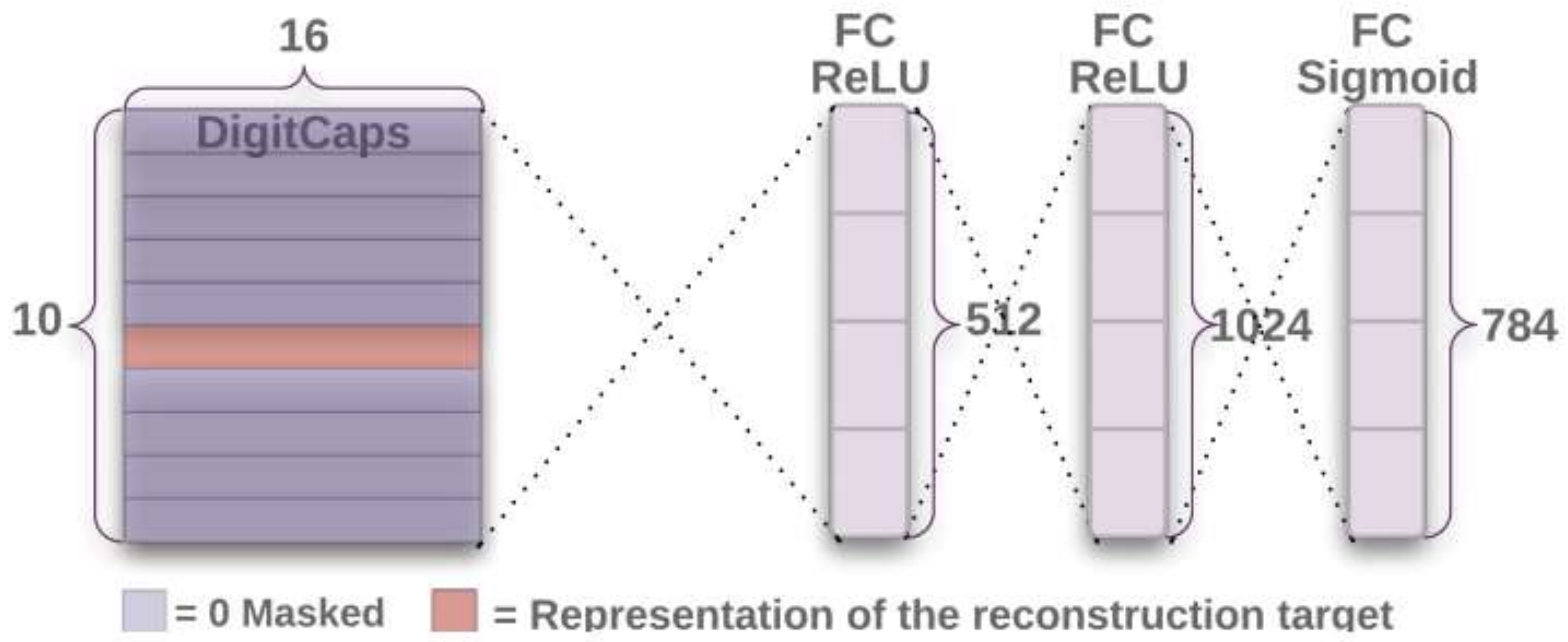
The Architecture of Capsule Network

CapsNet for MNIST

- **Model Architecture**



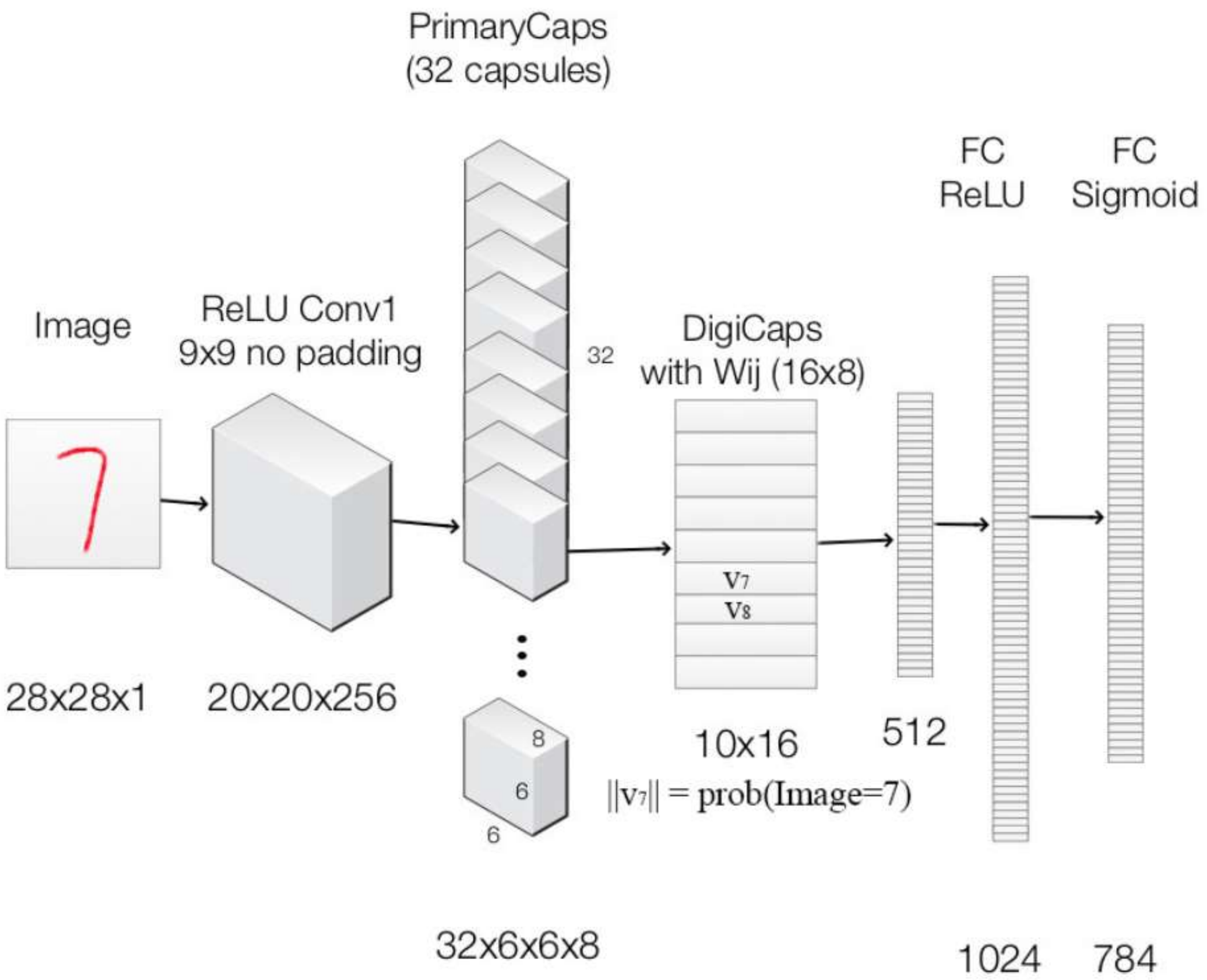
- **Reconstruction Regularization**
  - To encourage the digit capsule to encode instantiation parameters of the input digits, we add reconstruction loss (with weight 0.0005) to margin loss using a decoder with 3 fully connected layers.

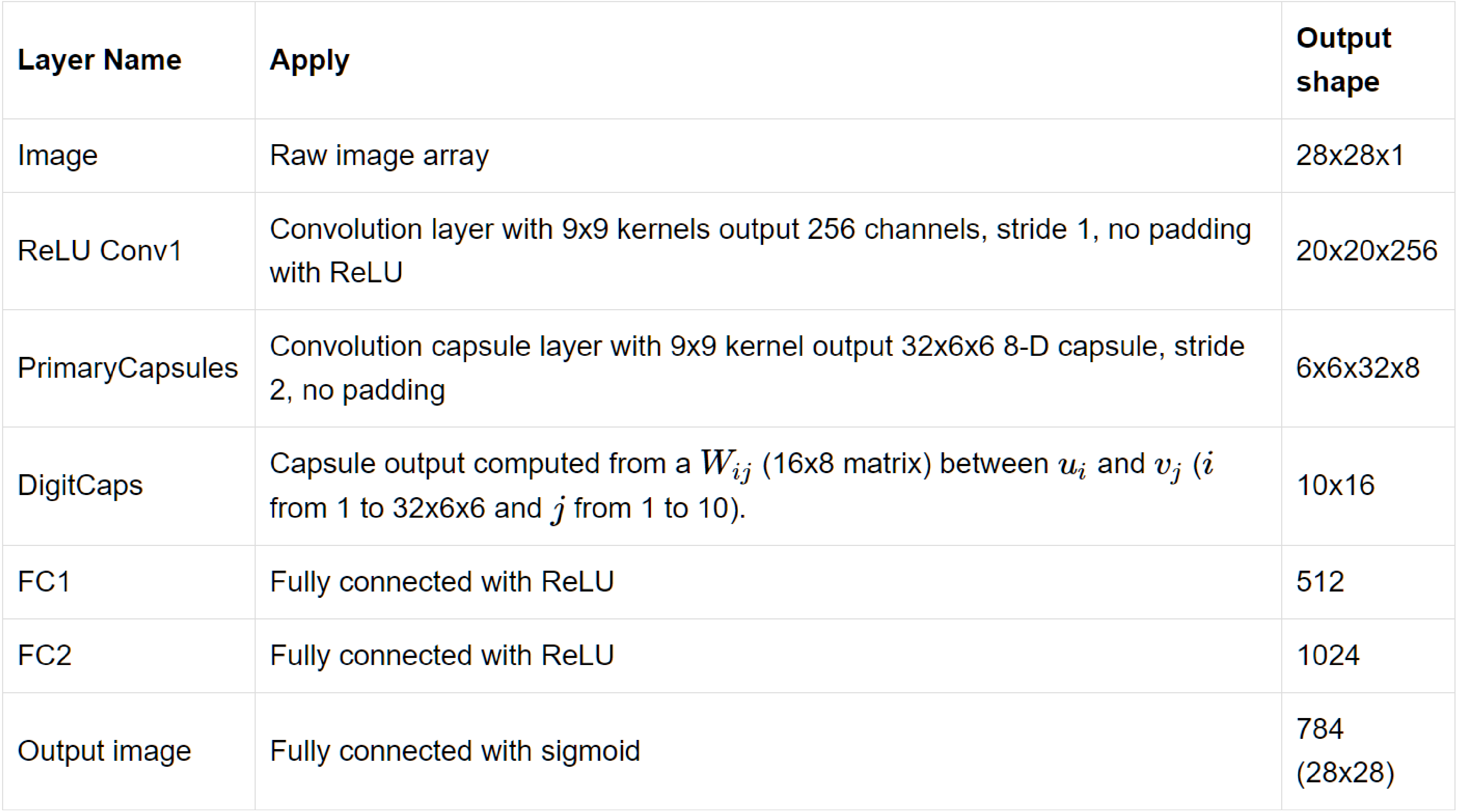


The Architecture of Capsule Network

CapsNet for MNIST

- **Classification + Reconstruction in One Diagram**





Layer Name	**Apply**	**Output shape**
Image	Raw image array	28x28x1
ReLU Conv1	Convolution layer with 9x9 kernels output 256 channels, stride 1, no padding with ReLU	20x20x256
PrimaryCapsules	Convolution capsule layer with 9x9 kernel output 32x6x6 8-D capsule, stride 2, no padding	6x6x32x8
DigitCaps	Capsule output computed from a W_{ij} (16x8 matrix) between u_i and v_j (i from 1 to 32x6x6 and j from 1 to 10).	10x16
FC1	Fully connected with ReLU	512
FC2	Fully connected with ReLU	1024
Output image	Fully connected with sigmoid	784 (28x28)

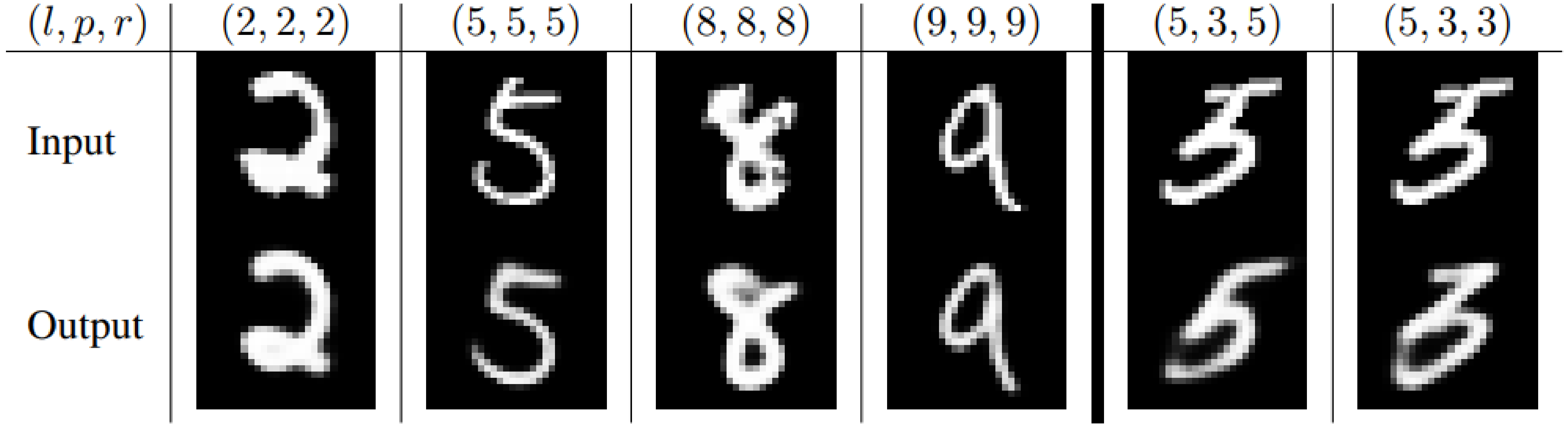
Experiments

Hand-written Digit Classification

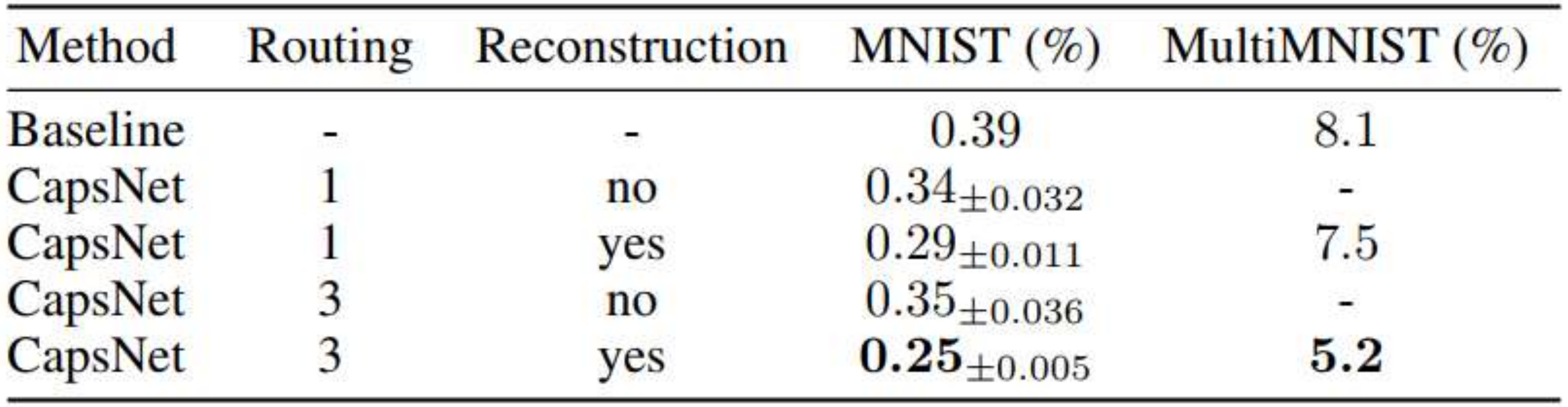
MNIST Dataset

- **Prediction and Reconstruction Example**

  (l, p, r) = label, prediction, reconstruction target



- **Classification Accuracy**

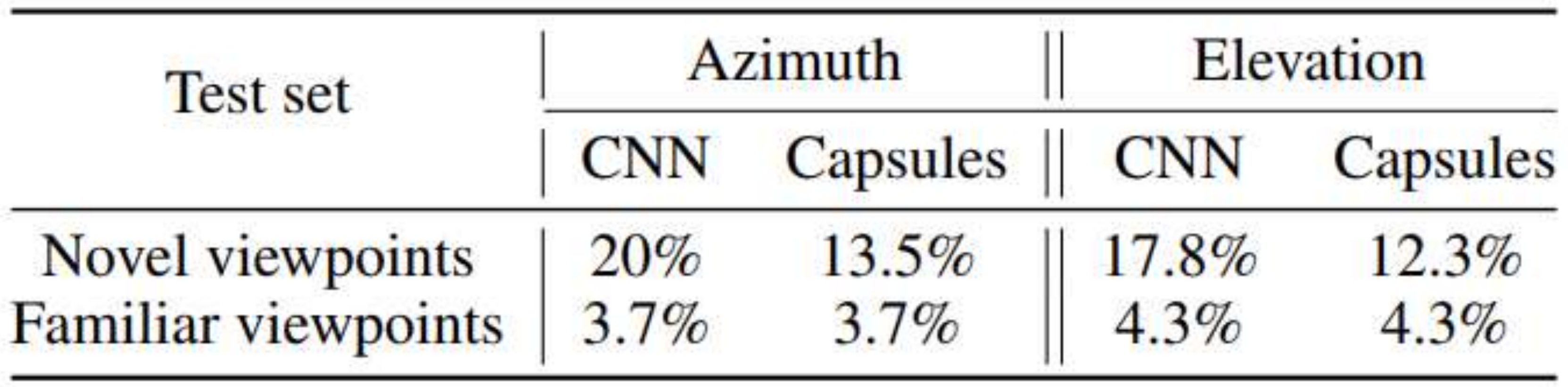


Method	Routing	Reconstruction	MNIST (%)	MultiMNIST (%)
Baseline	-	-	0.39	8.1
CapsNet	1	no	$0.34_{\pm 0.032}$	-
CapsNet	1	yes	$0.29_{\pm 0.011}$	7.5
CapsNet	3	no	$0.35_{\pm 0.036}$	-
CapsNet	3	yes	$\mathbf{0.25}_{\pm 0.005}$	**5.2**

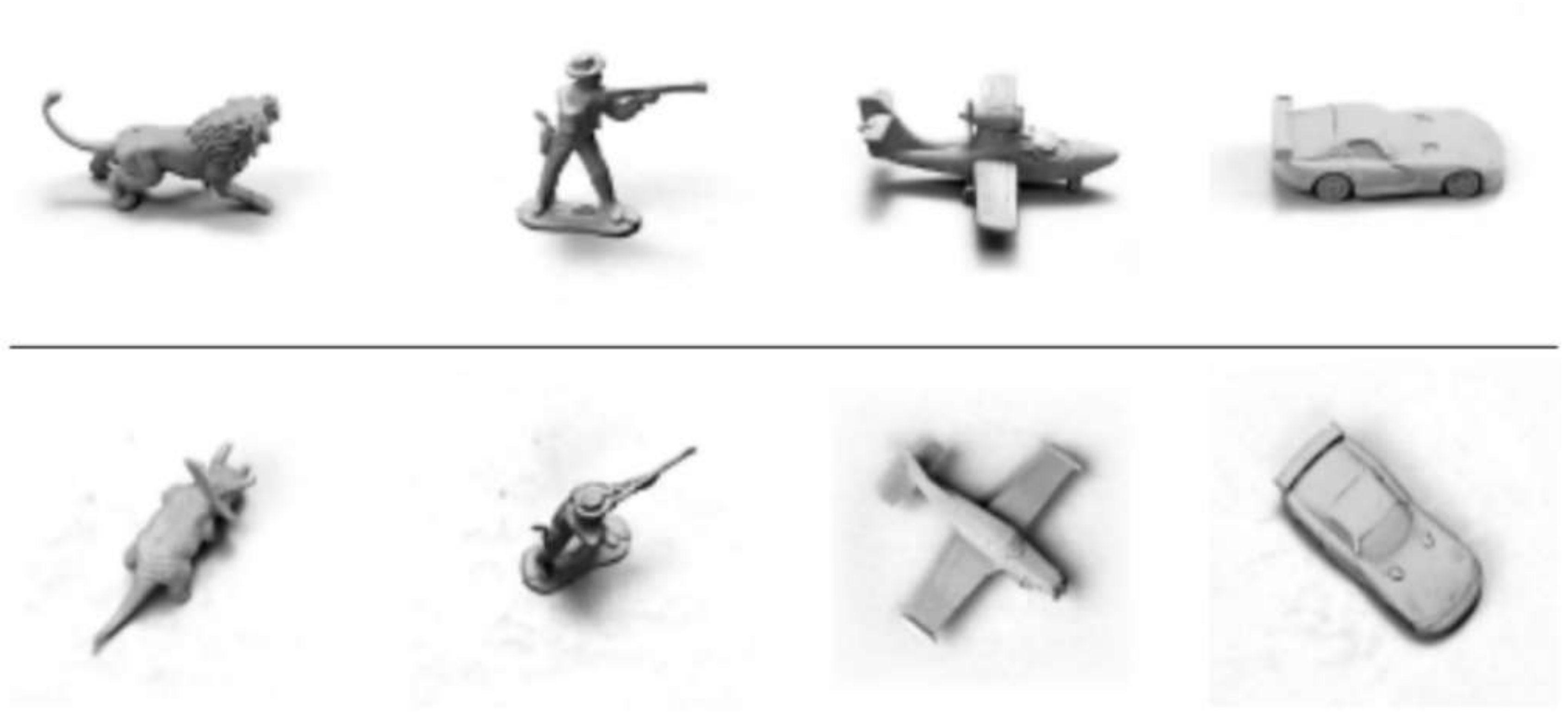
Object Recognition from Various Angle

smallNORB Dataset

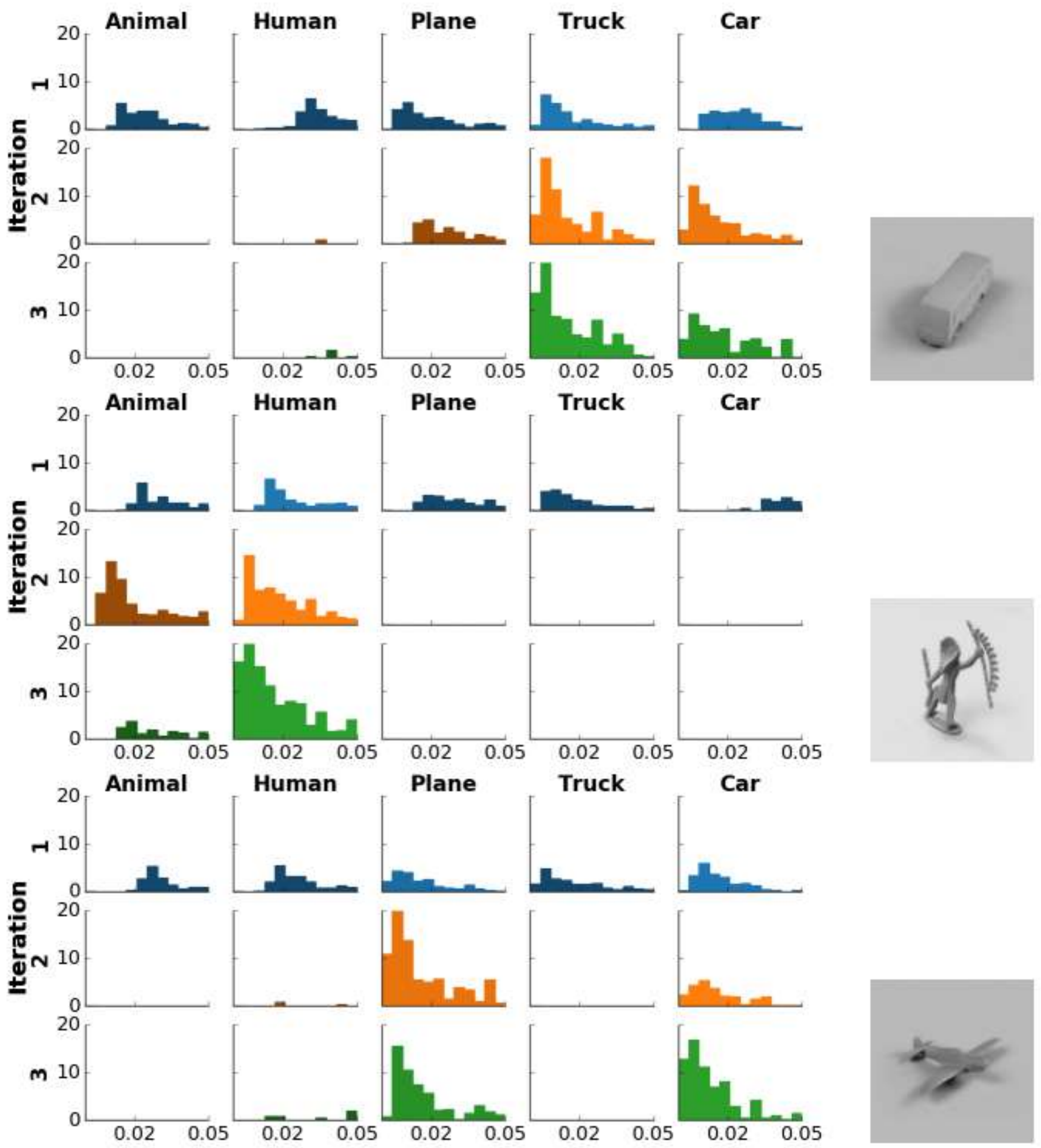
- Improvement of recognition performance on novel viewpoints



Test set	Azimuth		Elevation	
	CNN	Capsules	CNN	Capsules
Novel viewpoints	20%	13.5%	17.8%	12.3%
Familiar viewpoints	3.7%	3.7%	4.3%	4.3%



The capsule network is much better than other models at telling that images in top and bottom rows belong to the same classes, only the view angle is different. The latest papers decreased the error rate by a whopping 45%. Source.



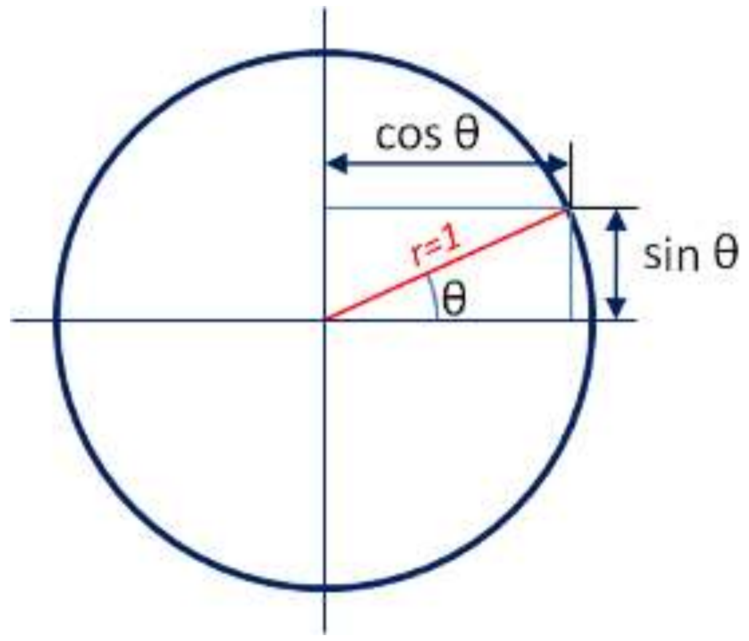
Conclusion

Discussion

Departure from ‘Conventional Neural Networks’

- We can train better models with less training data which is quite beneficial for medical image analysis where the annotations are scarce and expensive to get.
- We need to study more on how our brain works. Pure data-driven approach with overcomplex models might be wasting some resource(both data and computations)
- The CapsNet is the combination of credit assignment mechanism of CNN and instantization parameter learning of generative models.

- CapsNet is slower than CNN counterpart according to the routing update iterations.
- CpasNet has not yet proven its effectiveness in large-scale visual recognition problems such as ‘ImageNet’
- Each element in the activity vector does not always represent meaningful properties of input image
- Cosine similarity may not be the best measure of similarity between the predicted capsule output and actual capsule output.



References

- **Paper**
  - Original NIPS 2017 Paper : https://arxiv.org/abs/1710.09829
  - ICLR 2018 Paper : https://openreview.net/pdf?id=HJWLfGWRb

- **Video**
  - Geofrrey Hinton's talk - "What is wrong with convolutional neural nets?"
    - https://www.youtube.com/watch?v=rTawFwUvnLE&feature=youtu.be
  - Siraj Raval lecture - "Capsule Networks : An Improvement to Convolutional Networks"
    - https://www.youtube.com/watch?v=VKoLGnq15RM&feature=youtu.be&t=13m59s

- **Related News and Blog Posts**
  - https://medium.com/@pechyonkin/understanding-hintons-capsule-networks-part-i-intuition-b4b559d1159b
  - https://hackernoon.com/what-is-a-capsnet-or-capsule-network-2bfbe48769cc
  - https://www.nextobe.com/single-post/2017/11/02/CapsNetCapsule-Network
  - https://www.wired.com/story/googles-ai-wizard-unveils-a-new-twist-on-neural-networks/
  - https://jhui.github.io/2017/11/03/Dynamic-Routing-Between-Capsules/

- **Implementation**
  - https://github.com/nishnik/CapsNet-PyTorch
  - https://github.com/debarko/CapsNet-Tensorflow
  - https://github.com/XifengGuo/CapsNet-Keras
  - https://github.com/jaesik817/adv_attack_capsnet

vuno

Putting the world’s medical data to work

hello@vuno.co