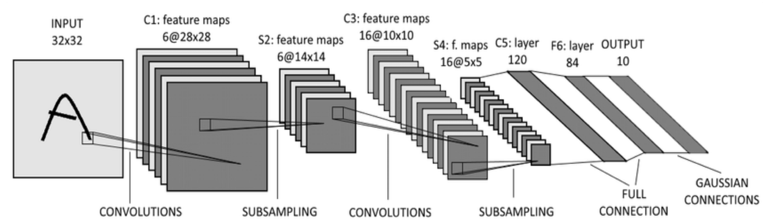
DeepCaps: Going Deeper with Capsule Networks

Jathushan Rajasegaran, Vinoj Jayasundara, Sandaru Jayasekara, Hirunima Jayasekara, Suranga Seneviratne and Ranga Rodrigo
The University of Moratuwa, Sri Lanka

MOTIVATION

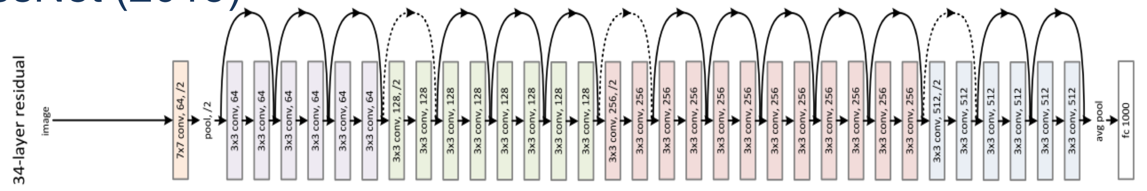
Convolutional Neural Networks

LeNet-5 (1998)

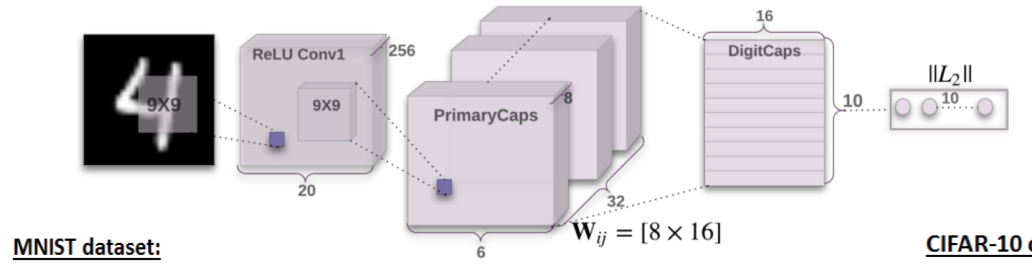


Going deeper in the network helps to understand rich features

ResNet (2015)



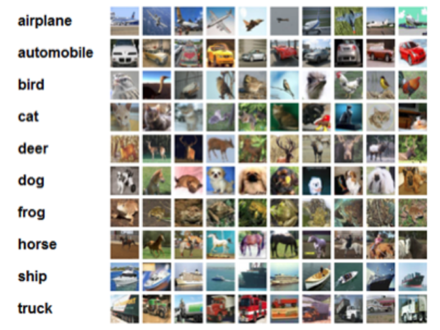
Capsule Neural Networks (2017)



MNIST dataset:

Low level features like edges and blobs to high level features

CIFAR-10 dataset:

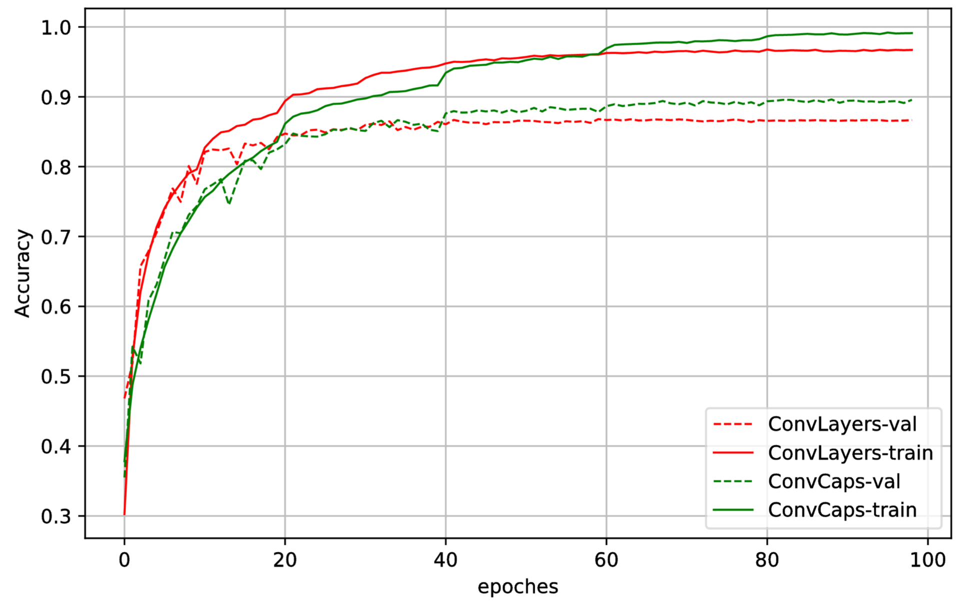


Achieve state-of-the art results on MNIST (99.75%)

CIFAR10, it achieves only 89.4% with 7-ensemble.

Hence, we need to go deeper in capsule networks.

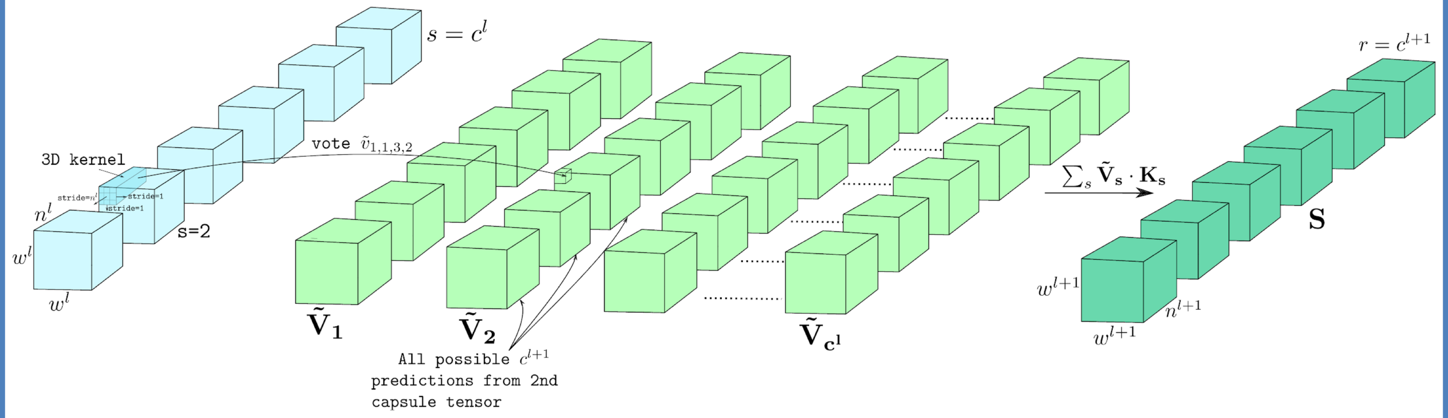
- When going deeper, dynamic routing is a computationally expensive procedure.



- Stacking capsule layers is not efficient, also stacking convolutional layers causes degradation.
- Hence, to overcome these issues, we introduce the novel CapsCell architecture.

3D CONVOLUTION BASED ROUTING

- We use 3D convolution kernels to transform low-level capsules to higher-level capsules.
- Keeping strides equal to the number of atoms in each capsule allows to separately transform capsules to higher level, with sharing the weights.
- Multiple such kernels generate next set of capsules and a squash function squashes capsule tensors to produce the final capsules.

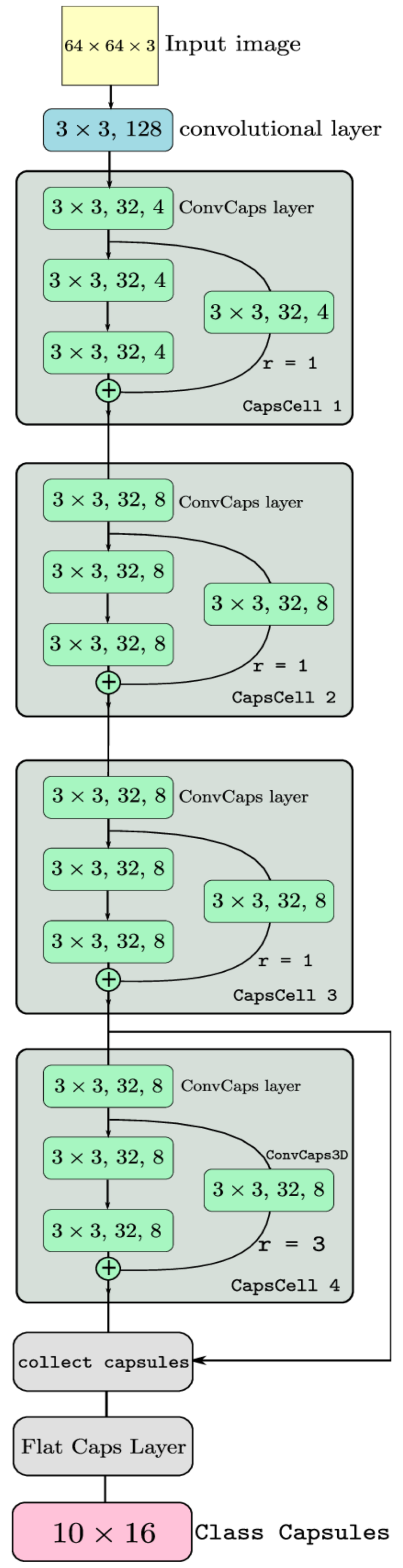


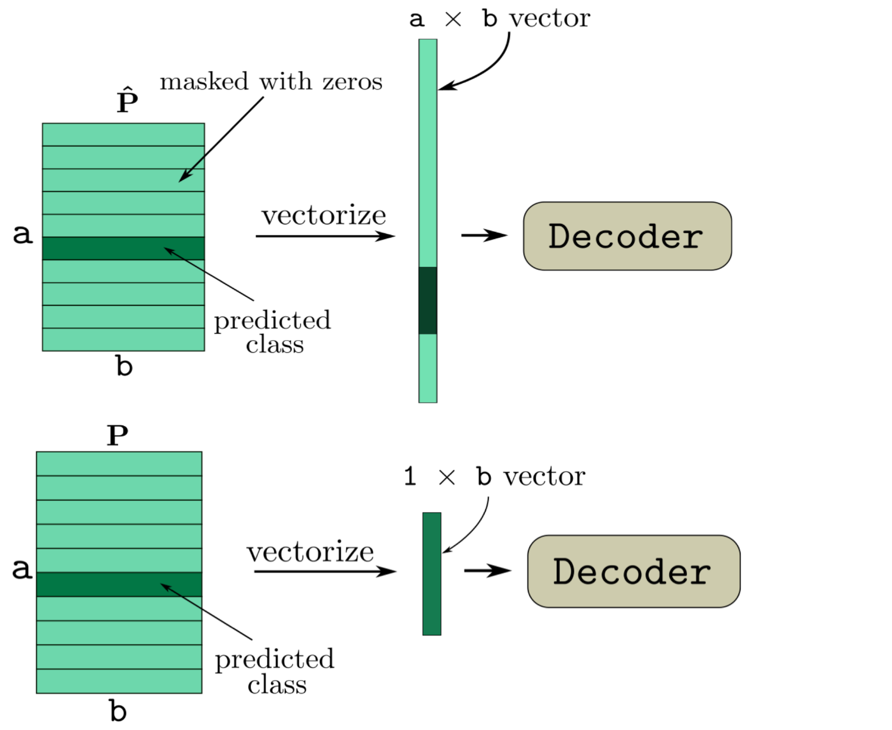
Algorithm 1 Dynamic Routing using 3D convolution

procedure ROUTING
Require: $\Phi^l \in \mathbb{R}^{(w^l, w^l, c^l, n^l)}$, r and c^{l+1}, n^{l+1}
  $\tilde{\Phi}^l \leftarrow \mathtt{Reshape}(\Phi_l) \in \mathbb{R}^{(w^l, w^l, c^l \times n^l, 1)}$
  $\mathbf{V} \leftarrow \mathtt{Conv3D}(\tilde{\Phi}^l) \in \mathbb{R}^{(w^{l+1}, w^{l+1}, c^l, c^{l+1} \times n^{l+1})}$
  $\tilde{\mathbf{V}} \leftarrow \mathtt{Reshape}(\mathbf{V}) \in \mathbb{R}^{(w^{l+1}, w^{l+1}, n^{l+1}, c^{l+1}, c^l)}$
  $\mathbf{B} \leftarrow \mathbf{0} \in \mathbb{R}^{(w^{l+1}, w^{l+1}, c^{l+1}, c^l)}$
  Let $p \in w^{l+1}, q \in w^{l+1}, r \in c^{l+1}$ and $s \in c^l$
  for r iterations **do**
    for all p, q, r, $k_{pqrs} \leftarrow \mathtt{softmax\_3D}(b_{pqrs})$
    for all s, $S_{pqr} \leftarrow \sum_s k_{pqrs} \cdot \tilde{V}_{pqrs}$
    for all s, $\hat{S}_{pqr} \leftarrow \mathtt{squash\_3D}(S_{pqr})$
    for all s, $b_{pqrs} \leftarrow b_{pqrs} + \hat{S}_{pqr} \cdot \tilde{V}_{pqrs}$
  return $\Phi^{l+1} = \hat{\mathbf{S}}$

DEEPCAPS ARCHITECTURE

- Deep capsule network architecture is build with modular building blocks of CapsCells .
- A CapsCells has 4 ConvCaps Layers with a skip connection with element wise addition.
- At the early stages of the network we keep routing iterations to one, and we increase it at the end of the network.
- DeepCaps is followed by a class independent decoder network, to better regularize the training.

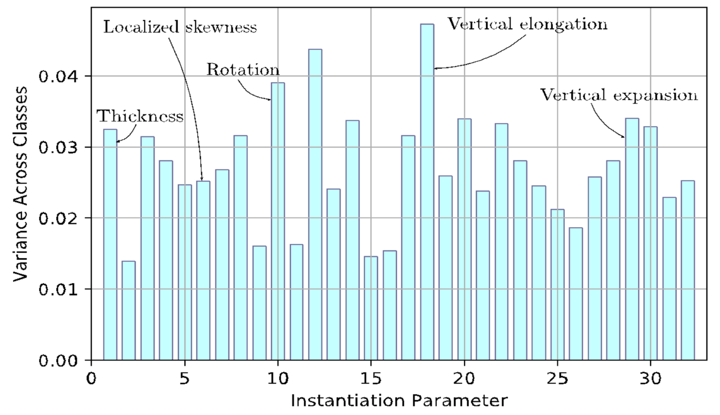


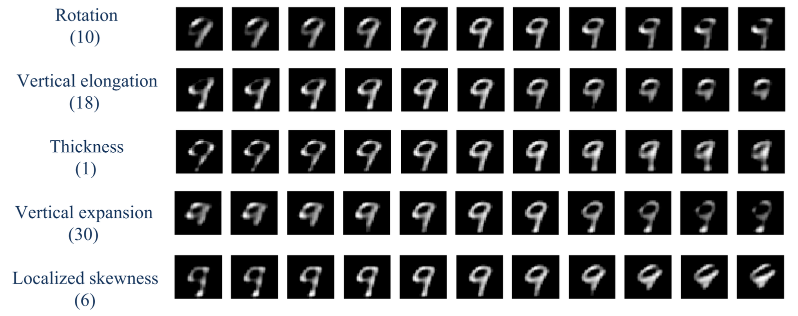


CLASS INDEPENDENT DECODER

- With the class independent decoder we can learn all the latent distributions in a constrained space. This allows us to jointly learn instantiation parameters that cause visual changes.
- Due to the joint learning, we can uniquely identify the instantiation parameter that causes a particular physical change.
- This allows to generate new data, with specific styles across all the classes.

- Instead of masking the vectorized instantiation parameter, we only pass the instantiation vector to the decoder.
- Not only able to identify the same variations across all the classes, high variance parameters cause global variations such as rotation, elongation, while the rest is localized variations.





PERFORMANCE OF DEEPCAPS

Model	CIFAR 10	SVHN	F-MNIST	MNIST
DenseNet	96.40%	98.41%	-	-
Wan et al.	-	-	-	99.79%
Zhong et al.	96.92%	-	96.35%	
Sabour et al.	89.40%	95.70%	-	**99.75%**
Nair et al.	67.53%	91.06%	89.80%	99.50%
HitNet	73.30%	94.50%	92.30%	99.68%
DeepCaps	91.01%	97.16%	94.46%	99.72%
DeepCaps (7-ensemble)	**92.74%**	**97.56%**	**94.73%**	-

CONCLUSION

- Our DeepCaps model surpass state-of-the art accuracy on CIFAR10, SVHN and F-MNIST and achieve state-of-the results on MNIST, with 52% reduction in inference time and 61% less parameters.
- Although our results surpass the state-of-the-art performance in the domain of capsule networks, we still behind, the STOA CNNs.