Capsule Network : An enhancement to the Convolutional Neural Network

Aishwarya T[1], Dr. Ravi Kumar V[2]
[1]M.Tech Student, Vidyavardhaka College Of Engineering
[2]Professor and Head, Vidyavardhaka College Of Engineering
(E-mail: aishwaryats95@gmail.com, ravikumarv@vvce.ac.in)

***Abstract*—**A Capsule Neural Network (CapsNet) is a machine learning system that is a kind of Artificial Neural Network (ANN) that can be used to better model hierarchical relationships. It is an attempt to more closely mimic biological neural organization. The objective and idea is to add structures called capsules to a Convolutional Neural Network (CNN) and to reuse output from several of those capsules to form more steasy and stable representations for higher order capsules. The output is a vector consisting of the probability of an observation, and a pose for that observation. This vector is similar to what is done for example when doing classification with localization in CNNs.

***Keywords*—** *Capsule Networks, Convolutional Neural Network, Artificial Neural Network, Capsules.*

I. INTRODUCTION

An Artificial Neural Network (ANN) is an information processing archetype that is influenced by the way biological nervous systems, like the brain, process information. The novel structure of the information processing system is the key element of this model or the paradigm. It is comprised of a large number of highly interconnected processing elements called neurons, working simultaneously and in unison to solve specific problems. ANNs, like humans, learn by example. An ANN is configured for a specific application through a learning process such as pattern recognition or data classification. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons [1].

A Convolutional neural network, CNN or ConvNet is a category of deep, feed-forward Artificial Neural Networks that has efficaciously been applied to analyzing visual imagery. It is based on the simple fact that a vision system needs to use the same learning and knowledge at all regions and locations in the image. This is attained by tying the weights of feature detectors so that features learnt at one location are available at other locations. Convolutional capsules extend the sharing of knowledge across locations to include knowledge about the part-whole relationships that characterize a familiar shape. The intention of capsules is to make good use of the underlying linearity, both for dealing with viewpoint variations and for improving segmentation decisions. Convolutional Neural Networks (CNNs) use translated copy or replicas of learned feature detectors. This allows them to translate knowledge about good weight values acquired at one position in an image to other positions. This has proven to be extremely useful in image interpretation.

A capsule is a collection of neurons whose activity vector represents the instantiation parameters of a particular type of entity such as an object or an object part. Active capsules at one level make predictions, via transformation matrices, for the instantiation parameters of higher-level capsules. When multiple predictions agree, a higher level capsule becomes active. A discriminatively trained, multi-layer capsule system achieves cutting edge and state-of-the-art performance and is considerably better than a Convolutional network at recognizing extremely overlapping digits. To accomplish these results, an iterative routing-by-agreement mechanism is employed. A lower-level capsule prefers to send its output to higher level capsules whose activity vectors have a big scalar product with the prediction coming from the lower-level capsule [2].

II. CONVOLUTIONAL NEURAL NETWORK

A Convolutional Neural Network (CNN, or ConvNet) is a deep learning algorithmwhich can take in an input image, asiign importance to various objects in the image and be able to differentiate one from the other. CNNs use a alteration of multilayer perceptrons designed to require minimal preprocessing. They are also called as Shift Invariant or Space Invariant Artificial Neural Networks (SIANN), based on their shared-weights architecture and translation invariance characteristics. A CNN consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers. The architecture of a ConvNet is analogous to that of the connectivity pattern of neurons in the human brain and was inspired by the organization of the visual cortex.

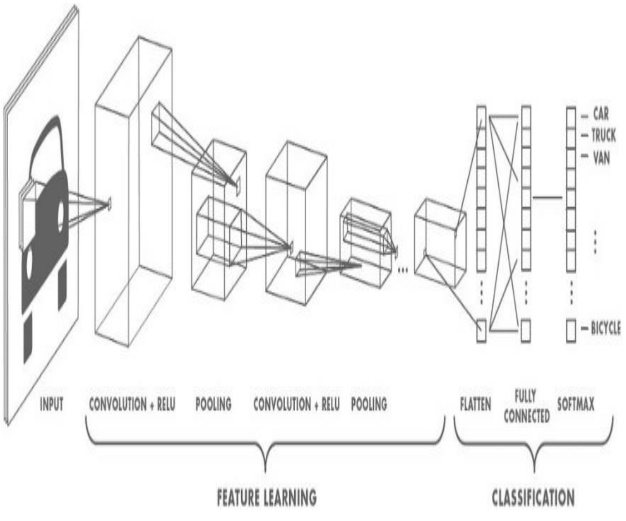

Figure 1: Layers of Convolutional Neural Networks

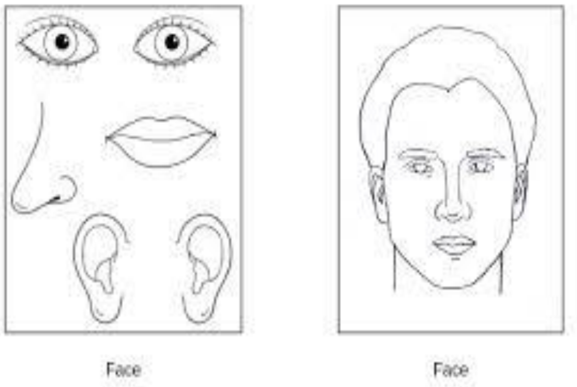

Figure 2: The CNN's prediction of a face

Figure 2 shows how the CNN's classify both a perfect face and another face with misplaced features of a face as a perfect face. This was not acceptable and is treated as a false positive image.

III. DRAWBACKS OF CONVOLUTIONAL NEURAL NETWORK

In recent years, a Convolutional Neural Network (CNN) has shown a predominant performance in the image recognition field. Nonetheless, a CNN requires a substantial amount of data for training as well as iterative computations. Furthermore, it also requires a high level of hardware performance. In recent years, the structure of CNN has been expanded; training data with only single CPU could take a long time due to a large amount of computation [3].

CNNs perform exceptionally great when they are classifying images which are very close to the data set. If the images have rotation, tilt or any other different orientation then CNNs have poor performance.

CNNs don't handle ambiguity very well. They cannot perform well on crowded scenes. They were trained on huge numbers of images or they reused parts of Neural networks.

Pooling helps in creating the positional invariance. Otherwise CNNs would fit only for images or data which are very close to the training set. This invariance also leads to triggering false positive for images for example, images which have the components of a ship but not in the correct order.

Invariance makes a CNN tolerant to small changes in the viewpoint. Equivariance makes a CNN understand the rotation or proportion change and adapt itself accordingly so that the spatial positioning inside an image is not lost.

CNNs work by accumulating sets of features at each layer. It starts of by finding edges, then shapes, then actual objects. However, the spatial relationship information of all these features is lost. In addition to being easily fooled by images with features in the wrong place a CNN is also easily confused when viewing an image in a different orientation. One way to combat this is with excessive training of all possible angles, but this takes a lot of time and seems counter intuitive.

IV. CAPSULE NETWORK

Though CNNs have shown supreme competence in image processing, there are still few problems to be looked after. To overcome the problems of CNNs, Sabour and Hinton put forward capsule network in recent years [4]. A capsule is a group of neurons whose outputs represent different properties of the same entity. Each layer in a capsule network contains many capsules. A basic idea is encoding a part-whole relationship between various entities which are objects or object parts and achieving translation equivariance [5]. A version of capsules is described in which each capsule has a logistic unit to represent the presence of an entity and a 4x4 matrix which could learn to represent the relationship between that entity and the viewer (the pose). Capsules reduce the number of test errors by 45% compared to the state-of-the-art.

Capsules also show far more resistance to white box adversarial attacks than our baseline convolutional neural network. Much like a regular neural network, a CapsNet is organized in multiple layers. The capsules in the lowest layer are called primary capsules: each of them receives a small region of the image as input, called its receptive field and it tries to detect the presence and pose of a particular pattern, for example a rectangle. Capsules in higher layers, called routing capsules, detect larger and more complex objects, such as boats.

Convolutional neural nets are based on the simple fact that a vision system needs to use the same knowledge at all locations in the image. This is achieved by tying the weights of feature detectors so that features learned at one location are available at other locations.

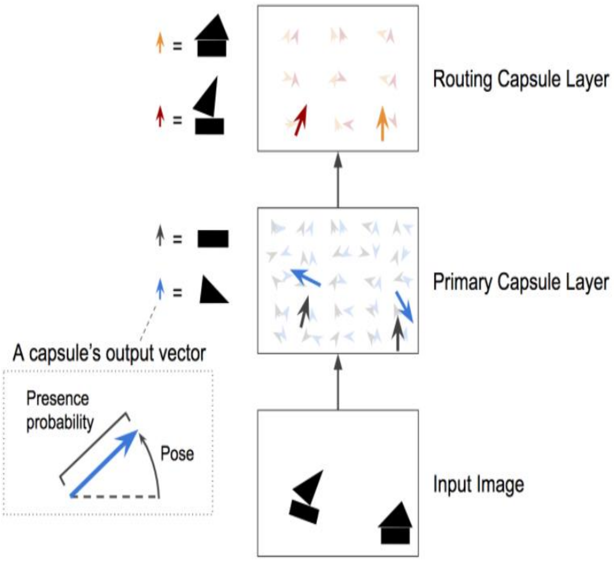

Figure 3: Layers of a Capsule network.

The primary capsule layer is implemented using a few regular convolutional layers. Convolutional capsules extend the sharing of knowledge across locations to include knowledge about the part-whole relationships that characterize a familiar shape. Viewpoint changes have complicated effects on pixel intensities but simple, linear effects on the pose matrix that represents the relationship between an object or object-part and the viewer. The aim of capsules is to make good use of this underlying linearity, both for dealing with viewpoint variations and for improving segmentation decisions.

Capsules use high-dimensional coincidence filtering: a familiar object can be detected by looking for agreement between votes for its pose matrix. These votes come from parts that have already been detected. A part produces a vote by multiplying its own pose matrix by a learned transformation matrix that represents the viewpoint invariant relationship between the part and the whole. As the viewpoint changes, the pose matrices of the parts and the whole will change in a coordinated way so that any agreement between votes from different parts will persist. a fast iterative process called "routing by-agreement" that updates the probability with which a part is assigned to a whole based on the proximity of the vote coming from that part to the votes coming from other parts that are assigned to that whole. This is a powerful segmentation principle that allows knowledge of familiar shapes to derive segmentation, rather than just using low-level cues such as proximity or agreement in color or velocity.

An important difference between capsules and standard neural nets is that the activation of a capsule is based on a comparison between multiple incoming pose predictions whereas in a standard neural net it is based on a comparison between a single incoming activity vector and a learned weight vector.

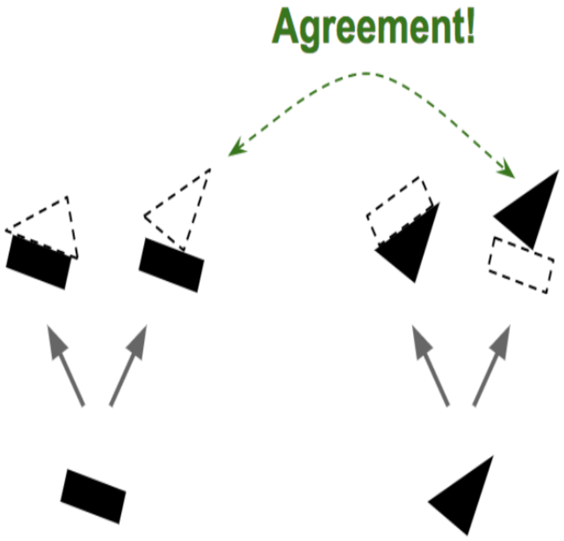

Figure 4: Routing By Agreement – Predicting the presence and pose.

Figure 4 represents the first step in Routing By Agreement, Predicting the pose and the presence of objects based on the presence and pose of the objects parts, then look for agreement between the predictions.

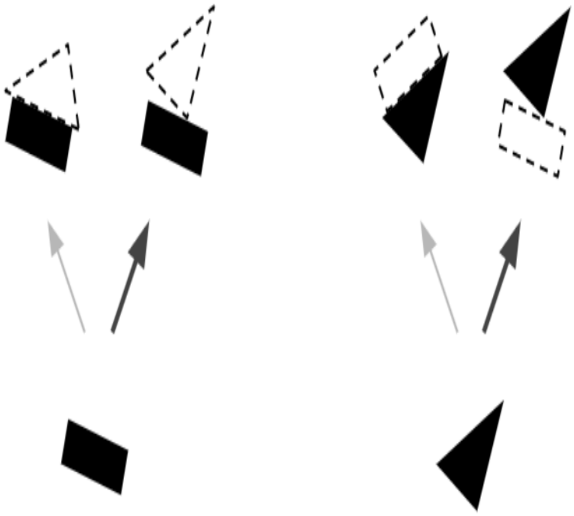

Figure 5: Updating the routing weights.

Figure 5 indicates the second step in the Routing-By-Agreement, updating the weights after the presence and pose are obtained.

The activities of the neurons within an active capsule represent the various properties of a particular entity that is present in the image. These properties can include many different types of instantiation parameter such as pose (position, size, orientation), deformation, velocity, albedo, hue, texture, etc. One very special property is the existence of the instantiated entity in the image. The fact that the output of a capsule is a vector makes it possible to use a powerful dynamic routing mechanism to ensure that the output of the capsule gets sent to an appropriate parent in the layer above. Initially, the output is routed to all possible parents but is scaled down by coupling coefficients that sum to 1. For each possible parent, the capsule computes a “prediction vector” by multiplying its own output by a weight matrix. If this prediction vector has a large scalar product with the output of a possible parent, there is top-down feedback which increases the coupling coefficient for that parent and decreasing it for other parents. This increases the contribution that the capsule makes to that parent thus further increasing the scalar product of the capsule’s prediction with the parent’s output. This type of “ routing-by-agreement” should be far more effective than the very primitive form of routing implemented by max-pooling, which allows neurons in one layer to ignore all but the most active feature detector in a local pool in the layer below.

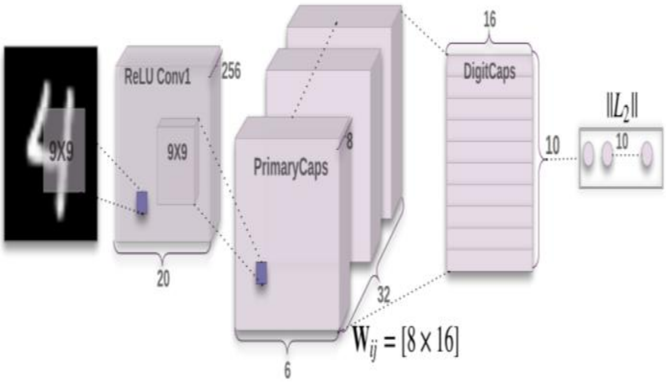

Figure 6: A Capsule Network architecture

A simple CapsNet architecture is shown in Figure 5.4. The architecture is shallow with only two convolutional layers and one fully connected layer. Conv1 has 256, 9 × 9 convolution kernels with a stride of 1 and ReLU activation. This layer converts pixel intensities to the activities of local feature detectors that are then used as inputs to the primary capsules. The primary capsules are the lowest level of multi-dimensional entities and, from an inverse graphics perspective, activating the primary capsules corresponds to inverting the rendering process.

V. ADVANTAGES OF CAPSULE NETWORK

Capsules avoid these exponential inefficiencies by converting pixel intensities into vectors of instantiation parameters of recognized fragments and then applying transformation matrices to the fragments to predict the instantiation parameters of larger fragments. Transformation matrices that learn to encode the intrinsic spatial relationship between a part and a whole constitute viewpoint invariant knowledge that automatically generalizes to novel viewpoints. Hinton et al. proposed transforming auto encoders to generate the instantiation parameters of the Primary Capsule layer and their system required transformation matrices to be supplied externally. We propose a complete system that also answers "how larger and more complex visual entities can be recognized by using agreements of the poses predicted by active, lower-level capsules".

Capsules make a very strong representational assumption: At each location in the image, there is at most one instance of the type of entity that a capsule represents. This assumption, which was motivated by the perceptual phenomenon called "crowding", eliminates the binding problem and allows a capsule to use a distributed representation, its activity vector to encode the instantiation parameters of the entity of that type at a given location.

Capsules use neural activities that vary as viewpoint varies rather than trying to eliminate viewpoint variation from the activities. This gives them an advantage over "normalization" methods like spatial transformer networks. They can deal with multiple different affine transformations of different objects or object parts at the same time.

Capsules are also very good for dealing with segmentation, which is another of the toughest problems in vision, because the vector of instantiation parameters allows them to use routing-by-agreement, as we have demonstrated in this paper. The importance of dynamic routing procedure is also backed by biologically plausible models of invariant pattern recognition in the visual cortex. Capsule Networks have the ability to perform well even on crowded scenes and is great in handling ambiguities well.

VI. CONCLUSION

A CapsNet is composed of capsules rather than neurons. A capsule is a small group of neurons that learns to detect a particular object within a given region of the image, and it outputs a vector. The key ideas are extremely promising and it seems likely that they just need a few tweaks to reach their full potential. Research on capsules is now at a similar stage to research on recurrent neural networks for speech recognition at the beginning of this century. The fact that a simple capsules system already gives unparalleled performance at segmenting overlapping digits is an early indication that capsules are a direction worth exploring.

REFERENCES

[1] Sonali. B. Maind and Priyanka Wankar, Research Paper on Basic of Artificial Neural Network, International Journal on Recent and Innovation Trends in Computing and Communication, Volume: 2 Issue: 1, 2014.

[2] Geoffrey Hinton, Sara Sabour, Nicholas Frosst, “MATRIX CAPSULES WITH EM ROUTING”, Published as a conference paper at ICLR 2018.

[3] Sejin Choi and Kwangyeob Lee, A CUDA-based Implementation of Convolutional Neural Network, 2015.

[4] Bo Tang, Ao Li, Bin Li, Minghui wang, “Capsurv : A capsule network for survival analysis with wholw slide pathological images”, IEEE journal, 2017.

[5] Canqun Xiang, Lu Zhang, Yi Tang, Wenbin Zhou, Chen Xu, “MS CapsNet : A novel multiscale capsulenetwork”, IEEE signal processing letters, July 2018.

[6] Geoffrey Hinton, Sara Sabour, Nicholas Frosst, Dynamic Routing Between Capsules, 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 2017.

[7] A. Lebedev, V. Khryashchev, A. Priorov, O. Stepanova, Face Verification Based on Convolutional Neural Network and Deep Learning, 978-1-5386-3299-4, IEEE, 2017.

[8] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, 2017.