machine learning & knowledge extraction

MDPI

Review

Capsule Network with Its Limitation, Modification, and Applications—A Survey

Mahmood Ul Haq [1], Muhammad Athar Javed Sethi [1] and Atiq Ur Rehman [2,*]

[1] Department of Computer System Engineering, University of Engineering and Technology, Peshawar 25120, Pakistan; mahmoodulhaq.cse@uetpeshawar.edu.pk (M.U.H.); atharsethi@uetpeshawar.edu.pk (M.A.J.S.)
[2] Artificial Intelligence and Intelligent Systems Research Group, School of Innovation, Design and Engineering, Mälardalen University, 722 20 Västerås, Sweden
* Correspondence: atiq.ur.rehman@mdu.se

Abstract: Numerous advancements in various fields, including pattern recognition and image classification, have been made thanks to modern computer vision and machine learning methods. The capsule network is one of the advanced machine learning algorithms that encodes features based on their hierarchical relationships. Basically, a capsule network is a type of neural network that performs inverse graphics to represent the object in different parts and view the existing relationship between these parts, unlike CNNs, which lose most of the evidence related to spatial location and requires lots of training data. So, we present a comparative review of various capsule network architectures used in various applications. The paper's main contribution is that it summarizes and explains the significant current published capsule network architectures with their advantages, limitations, modifications, and applications.

Keywords: CNN; capsule network; machine learning

Citation: Haq, M.U.; Sethi, M.A.J.; Rehman, A.U. Capsule Network with Its Limitation, Modification, and Applications—A Survey. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 891–921. https://doi.org/10.3390/make5030047

Academic Editor: Andreas Holzinger

Received: 3 July 2023
Revised: 26 July 2023
Accepted: 31 July 2023
Published: 2 August 2023

1. Introduction

Computer vision is the field of artificial intelligence used in real-time applications such as ECG classification [1], face recognition [2], tumor identification [3], object segmentation [4], vehicle recognition [5], character recognition [6], and security. Symbolic artificial intelligence is unable to resolve these complex real-time problems [7], resulting in the introduction of deep learning models such as convolutional neural networks (CNNs) [8] and recurrent neural networks (RNNs) [9]. CNNs are the most efficient at solving and performing image classification [10] and image recognition [11], from basic items to complicated objects. CNNs predominate in computer vision-related issues. Variants of CNN have demonstrated successful results when used for classification across a variety of domains [12]. However, CNNs fail due to their pooling process [13]. Lots of valuable information, such as object pose and location, is discarded in the polling process [14]. Another drawback of CNNs is their lack of rotational invariance, which requires a lot of training data [15]. Alternate techniques were used to overcome the limitations of CNNs, such as reinforcement learning [16] and end-to-end connected layers [17]. However, the proposed approaches did not show any improvement, which led to the introduction of capsule networks (CapsNet) [18], which increased the model accuracy by 45% over CNN [19]. This paper highlighted the limitations of CNNs and reviewed the promising performance of the capsule network in the literature. The main contributions of this paper are:

- To present state-of-the-art capsule models to motivate researchers.
- To explore possible future research areas.
- To present a comparative study of current state-of-the-art CapsNet architectures.
- To present a comparative study of current state-of-the-art CapsNet architecture routing algorithms.

- To explore the factors affecting the performance of capsule neural network architectures with their modifications and applications.

Since capsule networks are a new and hot research topic, this paper attempts to explain the idea behind them. Secondly, we presented a comparative study of the CapsNet architectures employed in various applications, overcoming the drawbacks of the CNN with their strengths and drawbacks with possible future directions.

The paper is organized as follows: In Section 1, the objectives of the paper and the background of the field under consideration are provided. In Section 2, an overview of CNNs and their limitations are discussed. Sections 3–8 present a brief overview of CapsNet algorithms with their performance, modifications, applications, advancements, and limitations. A review of implementations, structure, and performance evaluation methods is presented in Section 9. Section 10 presents the survey and comparative analysis of different routing algorithms, while Section 11 gives future research directions, and Section 12 concludes the paper. The table in the end presents the nomenclatures used in each section.

2. Convolutional Neural Network (CNN)

Let us discuss the characteristics of an image to enable CNNs to identify features. Suppose a grayscale, 2 × 2 pixel image. Computers display grayscale images as a two-dimensional array with each pixel represented by 8 bits (from 0 to 255). The color intensity is defined by the range 0 to 255, where 0 is black and 255 is white. The grayscale intensity range between black and white is between 0 and 255. Figure 1a shows how a computer interprets a 2 × 2 grayscale image, whereas Figure 1b shows how a computer represents features.

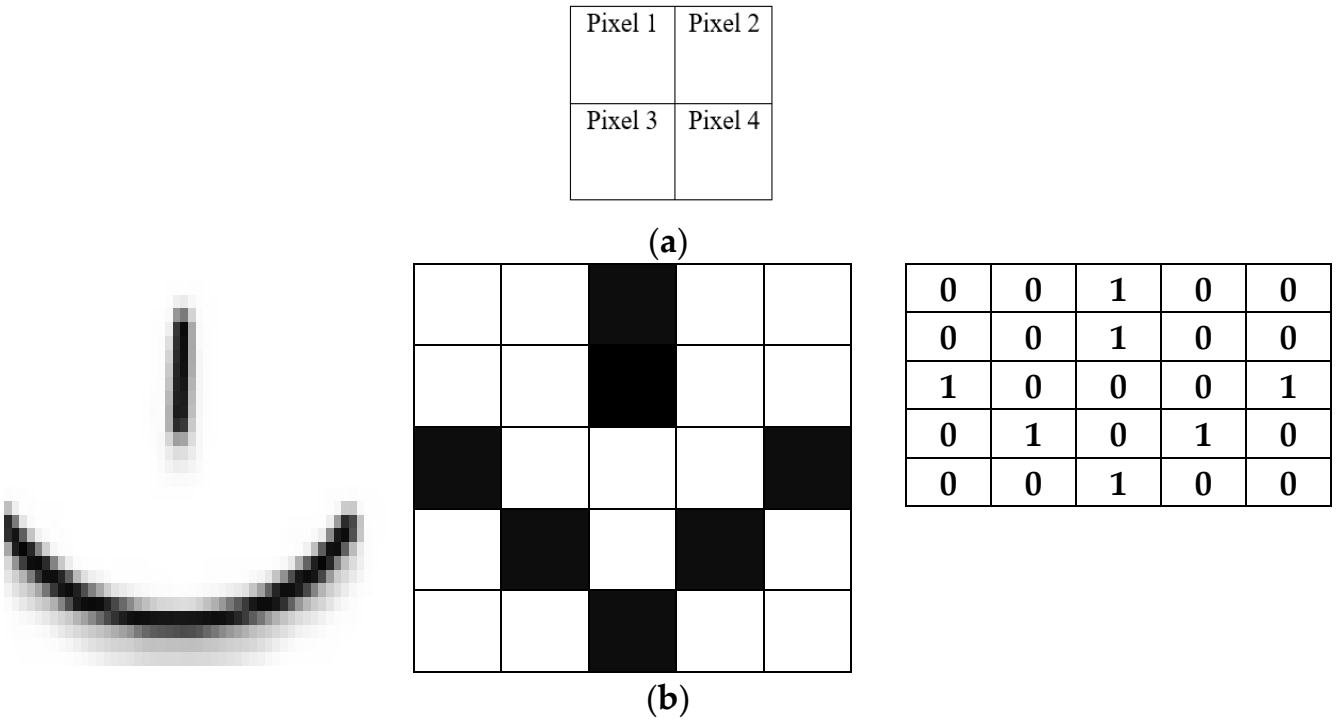

Figure 1. Image representation on a computer. (**a**) Interpretation of a 2 × 2 grayscale image, (**b**) computer representation of features.

A deep learning (DL) algorithm for image categorization is called a CNN. In essence, CNNs are made to scan picture data, prioritize different elements, and tell one class from another. A fully linked layer with an activation function, a pooling layer, and a convolution layer make up CNNs [20]. The input image is scanned to extract low-level features such

as edges in the convolutional layer. To make the model more non-linear and to cut down on computational complexity, the RELU [21] function is employed. A pooling layer, often referred to as down-sampling, is used to reduce memory requirements and recognize the same item in multiple images. Several types of pooling, including max pooling [22], min pooling, average pooling, and sum pooling, are utilized depending on the requirements. CNN's basic organizational structure [23] is shown in Figure 2.

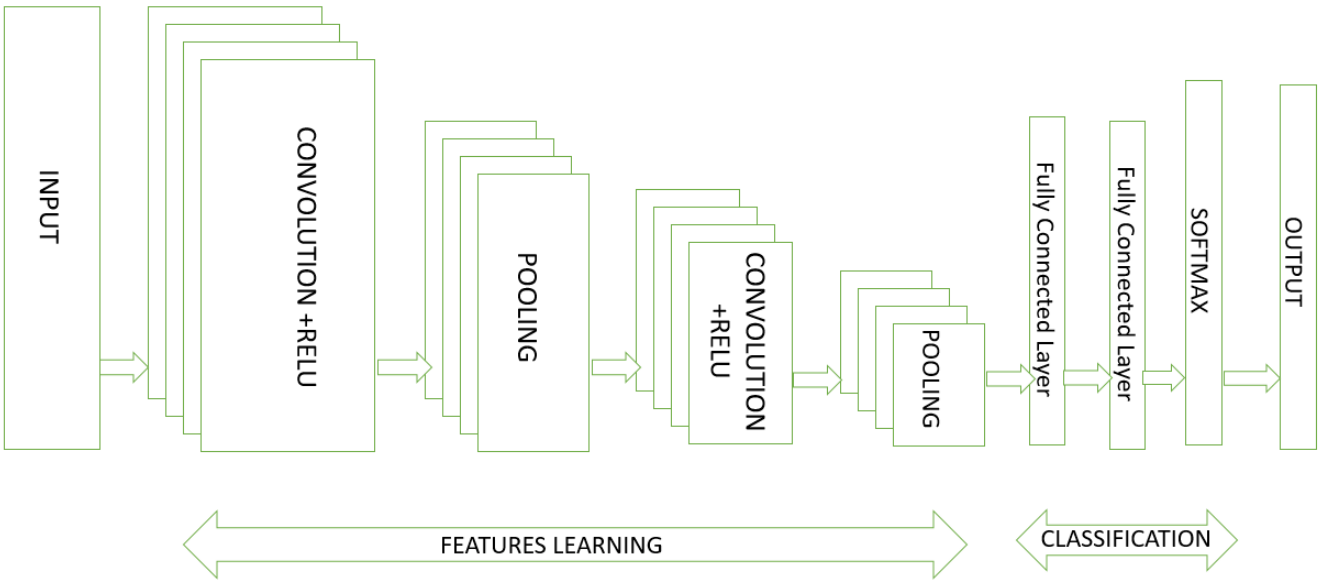

Figure 2. Basic structure of CNN.

The major problem with CNNs is that the pooling operation loses lots of features in an image. Therefore, CNNs are invariant, lack equivalence, and require lots of data and computational time to train [24]. Moreover, CNNs result in the wrong classification in cases of pixel perturbation [25].

3. Capsule Network (CapsNet)

The basic idea of CapsNet is to encrypt the relationship between various entities (scales, location, pose, and orientation). For example, a non-face image containing a nose, mouth, and eyes will be classified as a face by CNNs, despite the fact that a human clearly recognizes it as not a face. However, the capsule network will learn the relationship between features such as the nose and eyes located without a face and will successfully recognize them as not a face image.

Basically, a capsule network is a type of neural network that performs inverse graphics. For example, in object detection, the object is divided into subparts. To represent that object, a hierarchical relationship is developed between all subparts. The implementation of CapsNet is divided into three main parts. These parts are the input layer, hidden layer, and output layer.

Initially, capsule networks were presented in 2017 by Sabour and Hinton. This network involves two convolutional layers. The first convolutional layer includes 256 channels, made up of 9×9 filters with a RELU function with a stride of 1. The second layer was a convolutional capsule layer containing $6 \times 6 \times 32$ capsules with a stride of 2. Each primary capsule has eight convolutional units operating with a kernel of 9×9. The squashing function has been used as an activation function. The fully connected layer is the last layer of Capsnet with 16D capsules of size ten, known as DigitCaps. These capsules receive input from all capsules and perform classification based on ten classes. Figure 3 presents the structure of the capsule network.

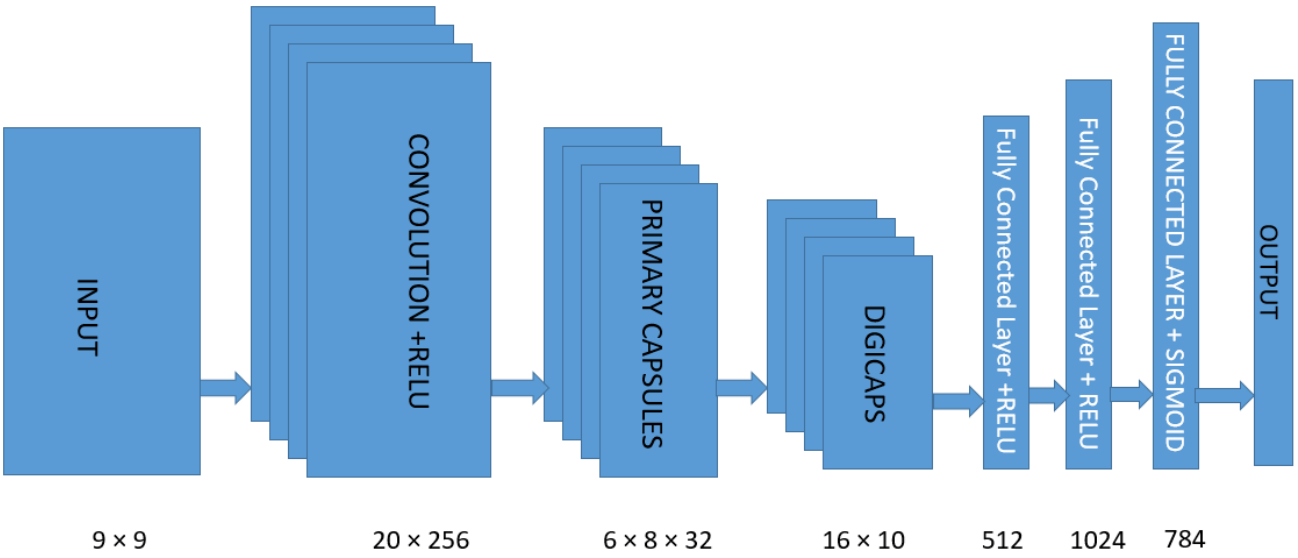

Figure 3. Structure of capsule network.

In terms of security, capsule networks have been shown to be more robust against certain types of attacks compared to traditional CNNs [26]. One of the most common attacks on neural networks is the adversarial attack, where an attacker perturbs the input data in a way that is imperceptible to humans but causes the model to misclassify the input. Capsule networks have been shown to be more robust against such attacks, as they are less affected by small changes in the input data.

CapsNet capacity to gather data on the interactions between several features in an input image accounts for its greater resilience. CapsNet employ "capsules", which are collectives of neurons that cooperate to represent particular aspects of a picture. These capsules can then be utilized to produce an output image that is more faithful to the original and less vulnerable to adversarial attacks.

In terms of security, tasks such as object recognition [27], vote assaults [28], intrusion detection [29], and adversarial attacks [30] have the potential to be very helpful when using capsule networks. Capsule networks are more effective at spotting odd or suspicious objects or activities in a picture when they take into account the spatial relationships between objects in an image [31].

Consider a security camera watching over a parking lot, for instance. A conventional CNN might be able to recognize the presence of a car on the scene, but it might not be able to tell if the automobile is parked in a permitted or prohibited place. On the other hand, a capsule network might be able to identify the spatial relationships between the vehicle and other items in the area and decide whether the vehicle is parked in a suspicious or odd spot.

Capsule networks are not completely impervious to security assaults, though. Recent studies have demonstrated that capsule networks are susceptible to specific attacks, such as the "black-box" attack, where the attacker has little understanding of the model and its parameters. This kind of approach still allows the attacker to produce adversarial examples by exploiting the transferability across various models, which can result in misclassifications.

Overall, despite showing some promise in terms of security resilience, capsule networks are not completely impervious to attacks. It is crucial to carefully assess the security of capsule networks and to use the proper defenses to mitigate any vulnerabilities, as with any machine learning model.

3.1. Mathematical Model of Capsule Network

The mathematical model of a capsule network can be described as follows:

Input: The input to a capsule network is an image or a set of images.

Convolutional layer: The output of the convolutional layer is given by:

$$z_{ij,k} = \sum_{l=1}^{L} W_{i,j,k,l} X_{i,j,l} + b_{i,j,k} \quad (1)$$

where $z_{ij,k}$ is the activation of the k^{th} feature map at position (i,j) in the output, $W_{i,j,k,l}$ is the weight associated with the i^{th} input channel at position (i,j) in the filter of the k^{th} feature map, $X_{i,j,l}$ is the activation of the i^{th} input channel at position (i,j) in the input, and $b_{i,j,k}$ is the bias term associated with the k^{th} feature map at position (i,j).

Primary capsule layer: The output of the primary capsule layer is given by:

$$V_{i,j,k} = \text{Squash}\left(\sum_{l=1}^{L'} W_{i,j,k,l} u_{i,j,l}\right) \quad (2)$$

where $V_{i,j,k}$ is the output vector of the k^{th} primary capsule at position (i,j) in the output, $u_{i,j,l}$ is the input vector associated with the i^{th} detected feature or part at position (i,j) in the input, $W_{i,j,k,l}$ is the weight matrix that connects the i^{th} input vector to the k^{th} primary capsule, and Squash is a non-linear activation function that ensures that the output vector has a length between 0 and 1.

Routing by agreement: The output of the higher-level capsule layer is given by:

$$S_j = \sum_i c_{ij} \hat{u}_{j|i} \quad (3)$$

where S_j is the output of the j^{th} higher-level capsule, $\hat{u}_{j|i}$ is the predicted output vector of the j^{th} higher-level capsule based on the input from the i^{th} primary capsule, and $c_{i,j}$ is the coupling coefficient that represents the probability that the i^{th} primary capsule should be routed to the j^{th} higher-level capsule.

Capsule output: The final output of the capsule network is given by:

$$y_k = squash(s_k) \quad (4)$$

where y_k is the output vector of the k^{th} capsule, and s_k is the input vector associated with the k^{th} capsule.

These equations explain how information spreads across a capsule network. Backpropagation is used to optimize the network's weights and biases during training in order to reduce the loss function, which calculates the discrepancy between expected and actual outputs. To fit particular tasks and datasets, the capsule network design can be altered in a variety of ways. For instance, new routing algorithms can be utilized to better capture the interactions between capsules, or further layers can be added to the network to increase speed.

3.2. Training and Inference Optimization Methods of CapsNet

Capsule networks are a type of neural network architecture that was introduced in 2017 as an improvement over CNNs. They use a different type of neuron called a capsule to represent various properties of an object, such as orientation, color, and texture, and use these capsules to form hierarchies that help to improve the network's ability to recognize objects in images. Training and inference optimization methods are crucial for improving the performance of capsule networks. Some of the most popular methods for training and inference optimization in capsule networks are:

Dynamic routing: Dynamic routing [18], a technique for sending data between capsules, was first described in the capsule networks article. Each pair of capsules in the same layer must have a scalar weight calculated for them in order to know how strongly connected they are to one another. The output of the capsule is then updated using the weight before being sent on to the next layer. Iterations of this method are carried out until the network converges.

Margin loss: A loss function used to train capsule networks is margin loss [32]. It aids in ensuring that the capsules are correctly identified and that the network is resistant to slight input perturbations. The margin loss enables the network to learn to distinguish between diverse objects and their various attributes by computing it depending on the distance between the capsule's output and the desired output.

Reconstruction loss: Another loss function that is used to train capsule networks is reconstruction loss [32]. It encourages the network to produce correct reconstructions of the input images and helps to guarantee that the network is learning to recognize objects and their features. Based on the difference between the input image and its reconstruction, the reconstruction loss is calculated.

Dynamic routing with EM routing: An improvement to the first dynamic routing algorithm is dynamic routing with EM routing [18]. It entails adding the expectation-maximization (EM) algorithm to the routing process, which enhances the network's stability and ability to handle complicated inputs.

Capsule reconstruction: A method used to increase the interpretability and robustness of capsule networks is capsule reconstruction [32]. It entails employing a decoder network to train the network to reconstruct the input image from its learned capsule representations. This can give the network a greater in-depth understanding of how to make predictions while also assisting it in learning more discriminative and invariant characteristics. Utilizing strategies such as reconstruction loss and adversarial training, capsule reconstruction can be improved.

Adversarial training: Capsule networks can be trained via adversarial training [33] to become resistant to adversarial attacks. It involves creating hostile instances or inputs intended to lead the network to incorrectly classify the object. The adversarial samples are then used to train the network, which enhances its capacity to identify and categorize objects in the midst of noise and other sorts of disturbances.

Capsule dropout: Capsule dropout [33] is a regularization technique that can be used to prevent overfitting in capsule networks. It involves randomly dropping out capsules during training, forcing the network to learn more robust and generalizable representations. Capsule dropout can be optimized using techniques such as the dropout algorithm.

Overall, these training and inference optimization methods are crucial for improving the performance of capsule networks and ensuring that they are able to recognize objects and their various properties accurately and robustly.

3.3. NASCaps

A framework for neural architecture search (NAS) called NASCaps [34] was created primarily to enhance the precision and hardware effectiveness of CapsNets. Instead of manually developing and fine-tuning the architecture, NAS is a method for automatically determining the best neural network architecture for a given task [35].

The goal of NASCaps is to improve the performance of CapsNets by automatically searching for the optimal architecture that maximizes accuracy while minimizing hardware requirements, such as memory usage and computation time [36]. NASCaps uses a combination of reinforcement learning and evolutionary algorithms to search for the optimal architecture.

In summary, CapsNets and NASCaps are related concepts, with CapsNets being a specific type of neural network architecture and NASCaps being a framework for optimizing the performance of CapsNets. While both concepts are focused on improving the accuracy and efficiency of neural networks, they are distinct in terms of their specific goals and techniques.

4. Factors Affecting CapsNet's Performance

Datasets play an essential role in the performance of an algorithm [37]. Initially, CapsNet gave promising results on the MNIST dataset. But as compared to complex datasets with varying backgrounds, sizes, colors, noise, and multiple objects in a single

sample, this dataset is quite simple. However, CapsNet performs better than CNN and gives promising results on more complex datasets [38]. On SVHN [39] and CIFAR10 [40] datasets with high intra-class variation and background noise, as compared to the advanced algorithms such as VGG NET [41] and CNN [42], CapsNet's performance was not efficient but still gave a better result than CNN [43]. Moreover, in some situations, increasing or decreasing the number of iterations will not affect accuracy. For a deep study, readers should refer to [44].

The size of the training dataset can have an impact on how well capsule networks (CapsNets) function. When developing deep learning models, such as CapsNets, the quantity of the training dataset is extremely important. When discussing the performance of CapsNets dependent on dataset size, keep the following considerations in mind:

4.1. Small Datasets

Overfitting is more likely to occur while training CapsNets on tiny datasets. Insufficient training samples may cause the model to memorize the training data rather than learn useful representations. The performance on the training set may therefore be outstanding, but the generalization to new data (validation and test sets) may not be.

4.2. Large Dataset

Large datasets tend to yield better results for CapsNets. More robust and generalizable characteristics can be learned by the model with a significant amount of different training data. The likelihood of overfitting is decreased, and the model can perform better on untested data.

4.3. Transfer Learning

Transfer learning is a useful technique when working with tiny datasets. A CapsNet can perform better if it is pre-trained on a large dataset with similar features before being fine-tuned on the target dataset since the model can use the information learned from the bigger dataset.

4.4. Data Augmentation

Data augmentation strategies can be used to lessen the effects of limited data. The effective dataset size can be extended by using modifications such as rotation, flipping, or cropping to produce more training examples, which enables the model to learn more robust features.

4.5. CapsNet Architecture

The architecture of the network can affect how well CapsNet perform. To prevent overfitting, it is crucial to create a simpler architecture with fewer parameters for tiny datasets.

4.6. Capsule Routing

Another factor that might affect the speed in CapsNet is the number of routing iterations. To avoid the model becoming extremely sensitive to changes in the data, it may be advantageous to employ fewer routing iterations for small datasets.

4.7. Class Balancing

Class imbalances can be impacted by dataset size. If there are few samples available for particular classes, the model might find it difficult to learn those classes efficiently. Class imbalances can be addressed using strategies such as oversampling or class re-weighting.

4.8. Model Complexity

For smaller datasets, shallower CapsNet designs with fewer capsules and lower complexity are typically preferable. Larger datasets might be needed for Deep CapsNets in order to prevent overfitting and achieve adequate generalization.

In summary, the dataset size significantly impacts the performance of CapsNets. Larger datasets tend to lead to better generalization, while small datasets require careful handling through techniques such as transfer learning, data augmentation, and model simplification to achieve satisfactory results. When working with limited data, it is crucial to consider these strategies to optimize the performance of CapsNets on the target task.

Moreover, in terms of robustness to affine transformation, CNNs with a global average pooling layer perform better than CapsNet [45]. Furthermore, the dynamic routing and transformation processes can harm the robustness of this algorithm. The squashing function and conditional reconstruction are beneficial for learning semantic representation. But they are applied beyond CapsNet and are considered auxiliary components.

5. Modification in CapsNet

In a review of CapsNet's routing-by-agreement method, ref. [46] found that it does not always ensure that a higher-level capsule is linked to numerous lower-level capsules to construct a parse tree. An alternative method allows a lower-level capsule to choose a single parent instead of the original routing-by-agreement technique, which requires lower-level capsules to send their outputs to all higher-level capsules. This improvement increases the network's depth and resistance against white-box adversarial attacks [47].

Some researchers have looked into using a generative adversarial network (GAN) with a high-performing capsule-based discriminator to detect whether an image is real or artificially created (fake) [48]. According to [49], CapsNets have the potential to be superior to CNNs as GAN discriminators by maintaining important information without the requirement for pooling.

According to [50], the dynamic routing algorithm in CapsNet may be described as an optimization problem that involves minimizing an objective function. This stabilizes the training process by preventing the activation probabilities from falling severely out of balance during iterations. By doing regularization on the weight matrix with a margin loss, ref. [18] addressed the issue. According to [50], a more general method entails rescaling the weight matrix to guarantee that the inner product between input and the weighted sum (sj) of all individual primary capsule predictions for capsule j is less than 1 for each iteration.

Equal-weight initialization routing in the original CapsNet has a tendency to impede convergence and lower accuracy. An improved option is to train the initial routing weights via backpropagation by modeling them as trainable parameters [51]. The performance of multi-label classification problems can also be enhanced by taking into account the fact that primary capsule predictions are not independent.

Focusing on the length of a capsule rather than the individual capsule outputs has proven to be a more successful strategy for entity detection [52]. The presence of an entity can be denoted by the length of the capsule, and the pose attributes such as position, size, orientation, deformation, velocity, albedo, hue, and texture are represented by the orientation of the capsule.

Although a promising strategy for neural networks, capsules have certain implementation and performance issues. When determining the assignment probabilities between capsules in adjacent layers, SoftMax is frequently utilized; however, it has the drawback of convergent to uniform probability during routing iterations. The MaxMin function, which enables scale-invariant normalization and allows lower-level capsules to assume independent values, was suggested by [53] as a solution to this problem. This function enhances performance.

Densely connected convolutional layers can improve the learning of discriminative feature maps for CapsNets; however, doing so may cause the vanishing gradient problem when network depth is increased. Feature concatenations or ResNet-style skip connections can be used to add dense connections between layers to alleviate this issue [54,55].

The current CapsNet routing technique has a flaw in that the training process is not properly included in it. It is necessary to manually calculate the ideal number of routing iterations, which may not ensure convergence [56]. With further settings, such as higher

Conv and FC layers, CapsNets have shown potential on datasets other than MNIST or smallNORB [57].

However, due to the fact that the averaging of votes in a vector space does not result in equivariant mean estimates on the manifold of poses, dynamic routing-based CapsNets do not ensure equivariance or invariance. For this reason, ref. [58] developed group equivariant capsule layers, in which pose vectors are limited as components of a group, enabling guaranteed equivariance and invariance under specific circumstances in a general routing-by-agreement algorithm skip.

Numerous trials and academic studies have proven CapsNet's resistance to affine transformations. It can recognize objects despite changes in position, rotation, and scale thanks to the capsules' capacity to encode spatial relationships and the dynamic routing mechanism's consensus-building capabilities. In tasks such as object recognition and posture estimation, where the input photos can have different orientations and positions, CapsNet has demonstrated encouraging results. The original CapsNet design has undergone a number of changes and enhancements to increase its resistance to affine transformations. Table 1 presents several significant CapsNet changes and how they affect the system's robustness to affine transformation.

Table 1. Various CapsNet modifications to enhance affine transformation.

CapsNet Modification	Description	Impact on Robustness to Affine Transformations
Dynamic routing with RBA	Routing-by-agreement (RBA) variation of dynamic routing.	Enhances capsule consensus and adaptability to variations in position, rotation, and scale.
Aff-CapsNets	Affine CapsNets.	Significantly increases affine resilience with fewer parameters.
Transformation-aware capsules	Capsules explicitly designed to handle affine transformations.	Learns to detect and apply appropriate transformations to input features, improving invariance to affine transformations.
Capsule-capsule transformation (CCT)	Adaptive transformation between capsules.	Allows the network to handle varying degrees of affine transformations effectively.
Margin loss regularization	Adds margin loss terms during training.	Incentivizes larger margins between capsules, increasing resistance to affine transformations.
Capsule routing with EM routing	Utilizes an EM-like algorithm for capsule routing.	Improves capsule agreement process and feature learning for better robustness to affine transformations.
Self-routing	A supervised, non-iterative routing method.	Each capsule's secondary routing network routes it in a separate manner. As a result, the agreement between capsules is no longer necessary, but upper-level capsule activations and postures are still obtained in a manner similar to mixture of experts (MoE).
Adversarial capsule networks	Combines CapsNet with adversarial training techniques.	Helps the network learn robust features by training against adversarial affine transformations.
Capsule dropouts	Applies dropouts to capsules during training.	Enhances generalization and robustness by reducing capsule co-adaptations.
Capsule reconstruction	Augments CapsNet with a reconstruction loss term.	Encourages the network to preserve spatial information, improving robustness to affine transformations.
Capsule attention mechanism	Incorporates attention mechanisms into capsules.	Improves focus on informative features, aiding robustness against affine transformations.

6. Applications of CapsNet

By introducing "capsules" to better capture hierarchical relationships between features in data, CapsNets are a kind of neural network architecture designed to address some of the drawbacks of conventional convolutional neural networks (CNNs). Here is a thorough note on some of the most important applications for which CapsNets have demonstrated promise:

6.1. Image Classification and Recognition

Image identification and classification tasks are one of the main uses of capsule networks. CapsNets' superior ability to accurately describe part-whole connections and record spatial hierarchies enables them to recognize things with many components or different orientations. This makes them particularly effective for jobs where objects have distinct sections, such as classifying photographs of items taken from various angles or identifying handwritten numerals with varied writing styles.

6.2. Object Pose Estimation

In computer vision applications, CapsNets can be used for estimating object poses. CapsNets can estimate the 3D posture of things in photos by learning capsules that represent different object pieces and their associated spatial relationships. Due to the fact that it enables robots to comprehend the spatial orientation of objects in the environment, this capacity is extremely significant in robotics and augmented reality.

6.3. Image Generation

For image synthesis applications, generative models based on capsule networks can be used. CapsNets can produce coherent and realistic visuals with meaningful structures by modeling part-whole relationships. They can transform images while maintaining the underlying structures, which makes them helpful in applications such as image-to-image translation.

6.4. Medical Image Interpretation

Capsule networks have demonstrated promising outcomes in the interpretation of medical images. CapsNets' capacity to capture hierarchical relationships is helpful in segmenting organs, detecting anomalies, or even diagnosing diseases based on medical scans. Medical imaging frequently contains complicated structures. CapsNets can improve the precision of automated systems for analyzing medical images, which may help doctors make diagnoses.

6.5. Natural Language Processing (NLP)

Although CapsNets were primarily developed for computer vision tasks, researchers have also looked into their potential use in NLP. They can capture hierarchical links between words and phrases in sentences by modifying CapsNets to analyze sequential data. This could lead to better machine translation, text production, sentiment analysis, and other text analysis activities.

6.6. Video Analysis

CapsNets can be used for tasks such as activity tracking and recognition in video analysis. Since they can model temporal hierarchies, they are able to recognize complicated events and track objects through time in video sequences by capturing long-range dependencies.

6.7. Few-Shot Learning

With a limited amount of labeled data, a model must recognize and generalize to new classes in a few-shot learning scenario. In such cases, CapsNets' capacity to record part-whole interactions can be useful. Even if there are only a few instances given during training, they might be able to generalize more effectively to new classes.

Capsule networks are one of the newest additions to the field of machine learning. The capsule network is still in its infant, research, and development phases; as a result, there are no commercial applications that are based on it yet. Still, they can be used to solve real-life problems such as machine translation [59], autonomous cars [60], handwritten and text recognition [61], mood and emotion detection [62], intent detection [63], abnormal driving [64] on a complex road, predicting traffic speed [65], adversarial attacks [66],

self-driving cars [67], facial recognition systems [68], classifying brain tumors [69], and so on.

7. Why CapsNet Is Superior to CNN in Most Cases

Capsule networks (CapsNets) and convolutional neural networks (CNNs) are both deep learning architectures used for image recognition and classification tasks. CapsNets were introduced as an improvement over CNNs, particularly in terms of their ability to recognize and classify images with variations in orientation, scale, and deformation. One of the advantages that CapsNets have over CNNs is their ability to capture hierarchical relationships between features, which can be particularly useful in tasks such as object recognition. However, this does not necessarily mean that CapsNets are more robust than CNNs in all scenarios.

In fact, some research studies have shown that CNNs can outperform CapsNets in certain situations, such as when dealing with small datasets, when faced with adversarial attacks [66], or when having fewer parameters. Adversarial attacks are a type of cyberattack where small, carefully crafted perturbations are added to an input data point to fool the model into making the wrong prediction. Furthermore, CNNs have been used extensively in many real-world applications, such as self-driving cars [67] and facial recognition systems [68], and have proven to be highly effective and robust.

However, whether CapsNets are superior to CNNs in most cases is still an area of active research and debate. Here are some reasons why CapsNets might be superior to CNNs in some cases:

7.1. Better Understanding of the Spatial Relationship between Features

CapsNets can preserve spatial relationships between features better than CNNs. In CNNs, each filter considers only a small region of the input image, and the relationship between different features is lost. CapsNets, on the other hand, maintain the relative spatial relationships between features in the input image, which is important for recognizing complex objects.

7.2. Handling Variations

CapsNets use capsules (groups of neurons) to represent parts of an image and their properties, such as orientation and position. These capsules can capture variations in the image that CNNs might miss, especially in cases where the object of interest can appear in different positions, orientations, or scales.

7.3. Better Generalization

CapsNets have shown better generalization ability than CNNs, especially when the training data are limited or noisy. CapsNets can learn to recognize features in images that are not present in the training set, which can lead to better performance on unseen images.

Additionally, CapsNets are still a relatively new architecture, and there is limited research and practical experience with them compared to CNNs. Therefore, whether CapsNets are superior to CNNs in most cases is still an open question and depends on the specific task and dataset.

7.4. Hierarchical Representation

Both CNNs and CapsNets are capable of capturing hierarchical representations of feature representations, but CapsNets provide a more explicit mechanism for retaining spatial hierarchies, which can be useful for problems involving different points of view.

7.5. Translation Invariance

While CNNs are by themselves translation invariant, CapsNets require dynamic routing in addition to pooling to achieve invariance and manage spatial changes.

1. Viewpoint variations: CapsNets are made to be more resilient to changes in viewpoint and pose, which can be a problem for regular CNNs, particularly when dealing with 3D objects or objects in 3D scenarios.

7.6. Performance

While there is ongoing research into CapsNets' performance and they have shown promise, CNNs continue to be the most popular architecture for a variety of computer vision tasks because of their longevity, depth of research, and usability.

CNNs have received a lot of attention and are frequently used for a variety of image-related tasks, whereas CapsNets provide an alternate strategy that may be more effective for issues requiring improved handling of spatial hierarchy and viewpoint variations. CapsNets are still undergoing investigation; therefore, how well they perform in comparison to CNNs will depend on the task at hand and the dataset.

8. Limitation/Challenges of Capsule Network

Capsule networks are a relatively new type of neural network architecture that was proposed as an improvement over traditional convolutional neural networks (CNNs). Although CapsNets show promising results in certain applications, they also have some limitations/challenges. Here are a few:

8.1. Limited Understanding

CapsNets are a relatively new concept, and researchers are still working to understand how they work and how to optimize their performance. There are still many open questions about how CapsNets represent and process information and how they can be improved.

8.2. Computational Cost

CapsNets are more computationally expensive than CNNs due to the added complexity of the routing-by-agreement algorithm used to compute the activations of the capsules. This can make training CapsNets slower and more resource intensive.

8.3. Limited Real-World Applications

CapsNets have shown promising results on certain image recognition tasks, but their performance on other real-world problems is yet to be explored. There are still many open research questions about how CapsNets could be used in more complex and varied domains.

8.4. Model Complexity

CapsNets have a more complex architecture compared to traditional convolutional neural networks (CNNs). This increased complexity makes them more difficult to train and optimize.

8.5. Limited Interpretability

While the idea of capsules is to capture different properties of an object, the interpretation of these properties is difficult. Understanding how different capsules contribute to the final output of the network is not straightforward, which can make CapsNets less interpretable than CNNs.

8.6. Sensitive to Small Variations

CapsNets can be sensitive to small variations in the input, which can cause them to produce different output predictions for visually similar objects. This can be problematic for tasks that require high levels of precision and consistency.

8.7. Limited Availability of Pre-Trained Models

Pre-trained models for CapsNets are still not as widely available as those for CNNs, which can make it difficult for researchers and practitioners to use CapsNets in their work.

9. Literature Survey

Hinton et al., in 2017, presented a new method called capsule network (CapsNet) to overcome the limitations of CNNs. The CapsNet does not just extract and learn information about the image features; it also learns the relationship between these features, which results in a more accurate model. The proposed method was tested on the MNIST digit dataset, where the CapsNet model outclassed the state-of-the-art CNN models.

The researchers in [69] presented an investigative analysis of capsule networks for classifying brain tumors. According to researchers, CapsNet is truly becoming a trend in the field of image classification and gives promising results as compared to the other mentioned algorithms. The authors proposed that accuracy can be increased by changing the number of feature maps. Figure 4 shows the proposed architecture for brain tumor classification.

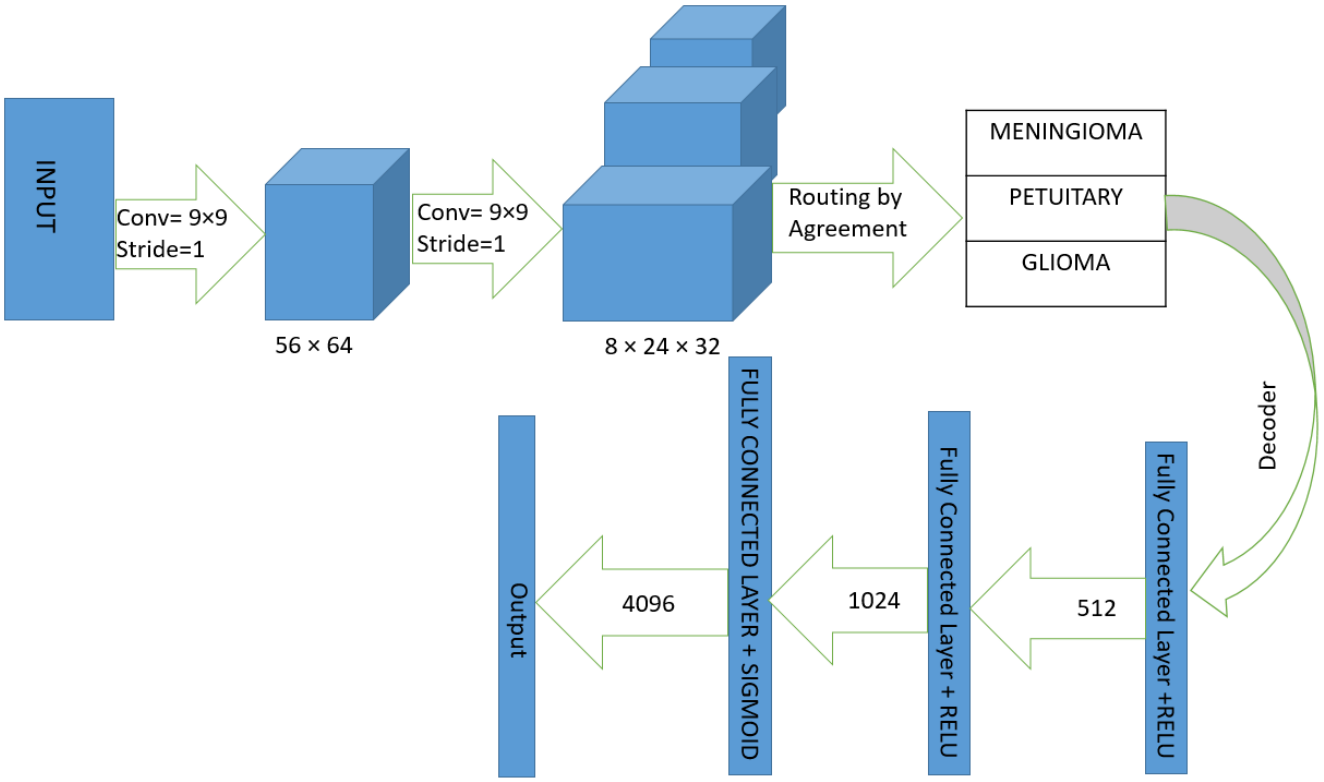

Figure 4. Capsule neural network for brain tumor detection.

The authors in [70] presented a comparative analysis of CapsNet with Fisherfaces, LeNet, and ResNet. These algorithms were tested on four datasets, including faces, traffic signs, and other objects. The authors claim that the CapsNet algorithm gives promising results on a small dataset. While achieving a lower error rate, CapsNet requires more sample images per class. The CapsNet architecture used in this paper is presented in Figure 5.

In [71], the authors presented a new algorithm, CapsGAN, by demonstrating the weakness of CNN-based GAN architecture in generating 3D images. The authors claim that CapsGAN accomplishes better results than CNN-GAN in generating 3D images with high geometric transformation. However, the proposed algorithm was tested on a simple dataset (MNIST), and additional experiments will be needed using complex datasets. The proposed architecture is presented in Figure 6.

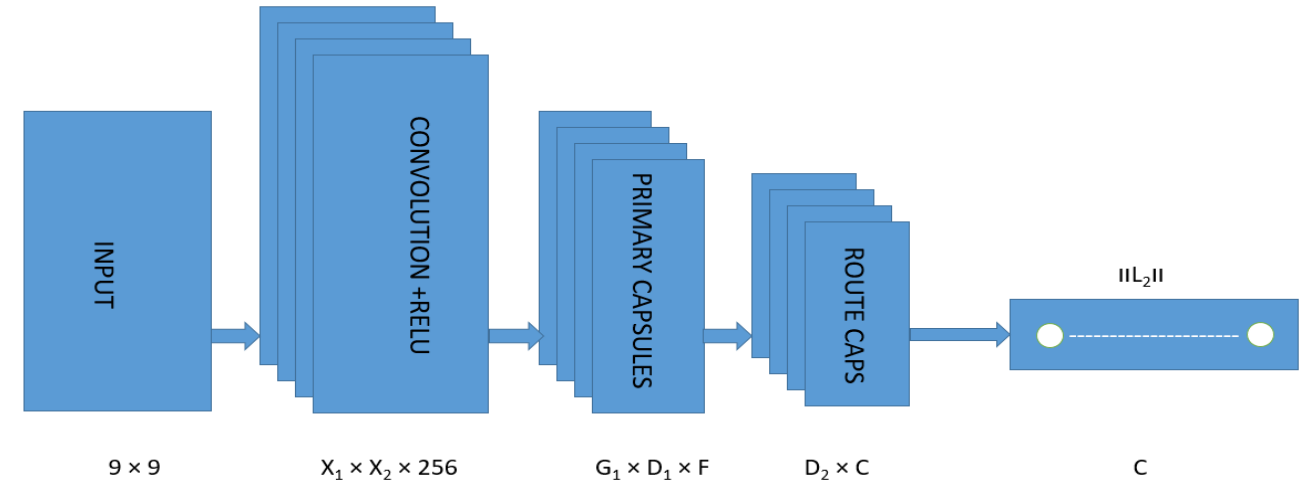

Figure 5. CapsNet architecture.

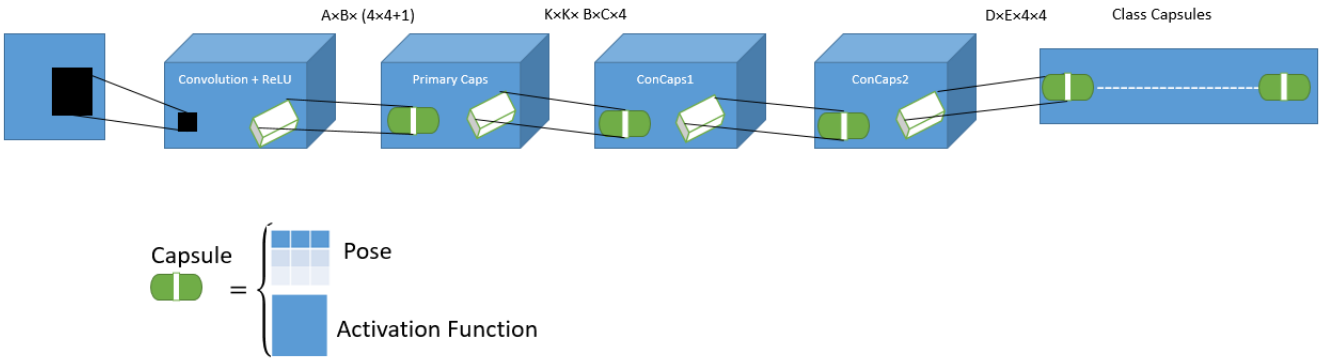

Figure 6. Proposed CapsGAN architecture for generating 3D images.

The authors in [72] proposed CapsNet to classify object height using ultrasonic data taken from automotive ultrasonic sensors. In preprocessing, the signal quality was improved, and the classification performance was improved to allow easy data interpretation by a human. A dataset of 21,600 measurements was collected for training and testing the proposed approach. The proposed approach achieved 99.6% accuracy using complex CapsNet as compared to complex CNNs (98.9%). Moreover, the authors suggested that lots of research is needed to make ultrasonic technology appropriate for self-driving vehicles. Figure 7 shows the proposed CapsNet used for ultrasonic data classification in self-driving vehicles.

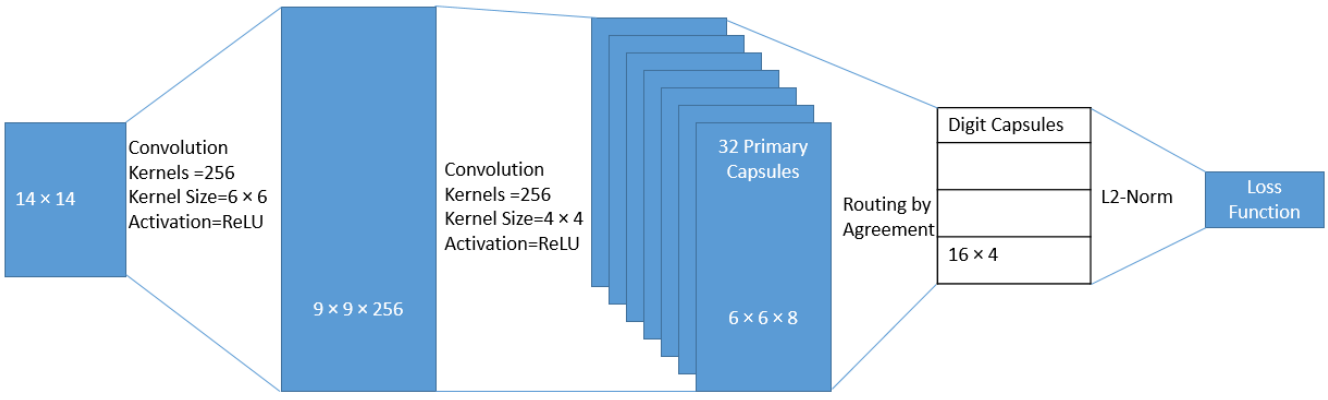

Figure 7. CapsNet for ultrasonic data classification in self-driving vehicles.

After producing better results on the MNIST dataset, the authors in [73] extended their work by using CapsNet for face recognition. A three-layer capsule network with two convolutional and one fully connected layer was used in their work. They used the LFW dataset and achieved 93.7% accuracy, beating the performance of traditional CNNs. Figure 8 presents the proposed three-layer CapsNet architecture.

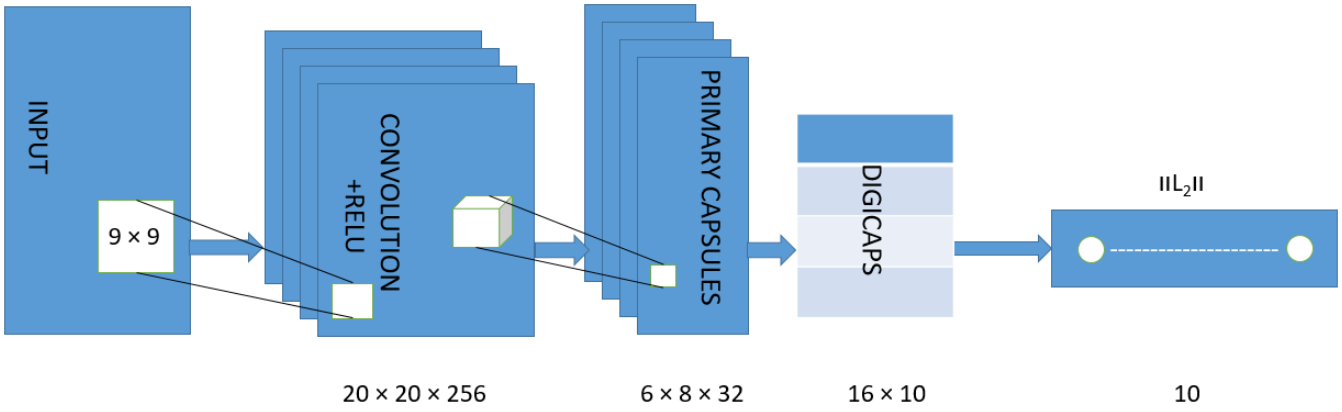

Figure 8. Three-layer CapsNet architecture.

Teto et al. [74] presented a comparative study of the C-CapsNet and CNN operations to identify animals in the wilderness. The authors discuss the ability of the capsule to rebuild any size and resolution image and the efficient learning efforts of the capsules from their convolutional layer, which achieved 96.48% accuracy.

The authors in [75] presented a CapsNet architecture with a smaller number of parameters. Moreover, dynamic routing was replaced with a non-iterative, parallelizable routing algorithm to handle a reduced number of capsules. The proposed methodology was tested by MNIST, MultiMNIST, and smallNORB. On these three datasets, the capsule network achieved higher accuracy with a lower number of parameters.

Pan et al. [76] presented a new version of the capsule network named the prediction-tuning (PT) capsule network. He also introduces fully connected PT capsules and locally connected PT capsules. The proposed model is different from the existing CapsNet architecture and can be used for more complex vision tasks. The proposed model provides better performance than CNN-based architectures on complex tasks. The results present an improvement in performance and considerable parameter reduction as compared to others. Figure 9 shows the proposed architecture of PT-CapsNet.

The researchers in [77] performed the classification of hierarchical multi-label text with a simple CapsNet approach. To determine its superior performance, they compared the proposed algorithm with SVM, LSTM, ANN, and CNN. For experimental purposes, the BGC and WOS datasets were used. The proposed algorithm correctly classified multi-label text as compared to other baseline algorithms. The presented structure of CapsNet is shown in Figure 10.

The authors in [78] proposed a CapsNet-based framework to diagnose COVID-19 disease in chest CT scan images. They referred to the algorithm as COVID-FACT. The classification of chest images was performed in two steps, each containing several convolutional and capsule layers. Based on their experiment, the authors claim that COVID-FACT achieved high accuracy with far fewer trainable parameters. The proposed approach achieved 90.82% accuracy with 94.55 sensitivity and a specificity of 86.04. Figure 11 presents the proposed COVID-FACT architecture.

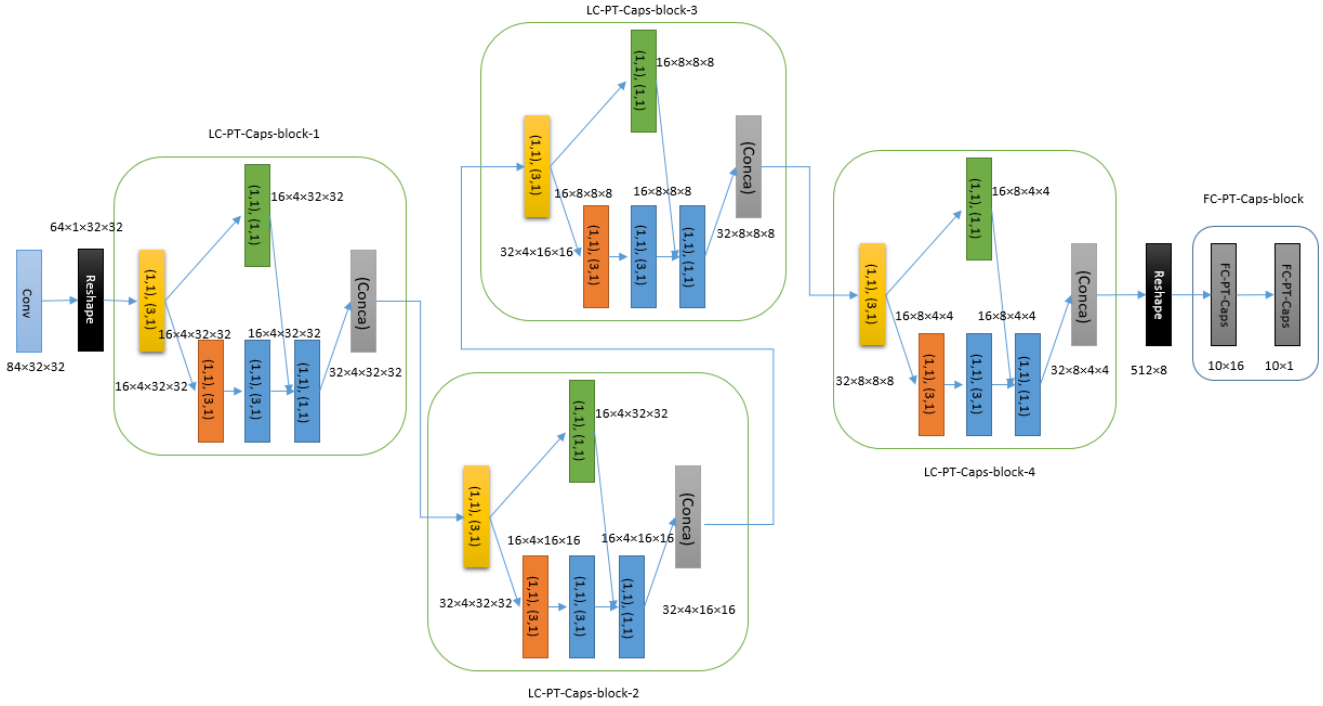

Figure 9. PT-CapsNet for image classification.

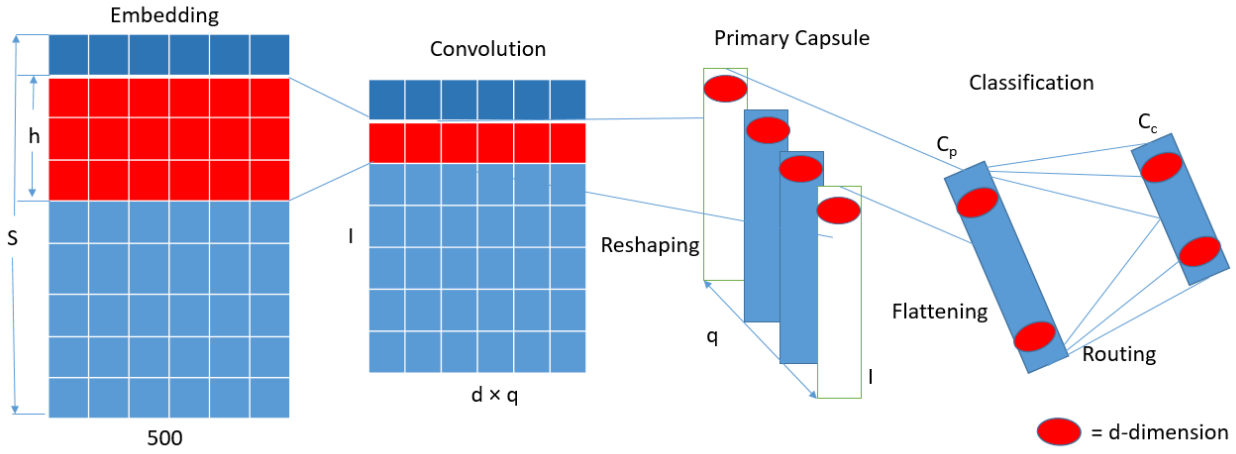

Figure 10. Proposed CapsNet architecture.

A three-in-one deep neural model architecture named CapsNet-ConvNet has been presented in [79]. The researcher used the CapsNet network with dynamic routing to predict the textual bullying content and a CNN to predict the graphical bullying content. The authors tested the proposed approach on a dataset containing 10,000 posts and comments taken from YouTube, Twitter, and Instagram and achieved an accuracy of 0.9705.

The authors in [80] presented two new deep learning models, CNN, CapsNet, and VGG, CapsNet, to detect COVID-19 disease using radiography images. The proposed model was tested on 2905 images with 219 COVID-19 patients, 1345 pneumonia patients, and 1341 normal patients. Based on experimental results, the authors claim that the VGG-CapsNet model outperforms the CNN-CapsNet model by achieving an accuracy of 97% for classifying COVID-19 and non-COVID-19 patients and 92% for classifying COVID-19, normal, and pneumonia samples. The proposed architecture is presented in Figure 12.

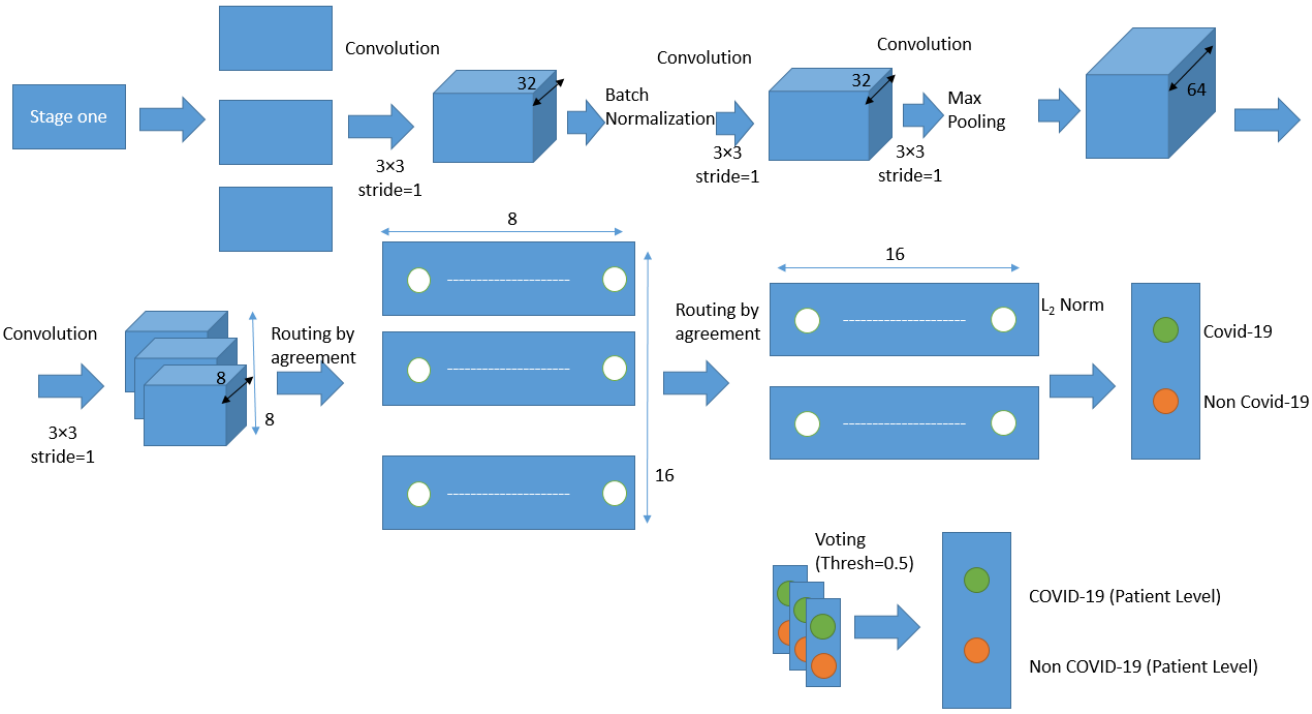

Figure 11. COVID-FACT architecture.

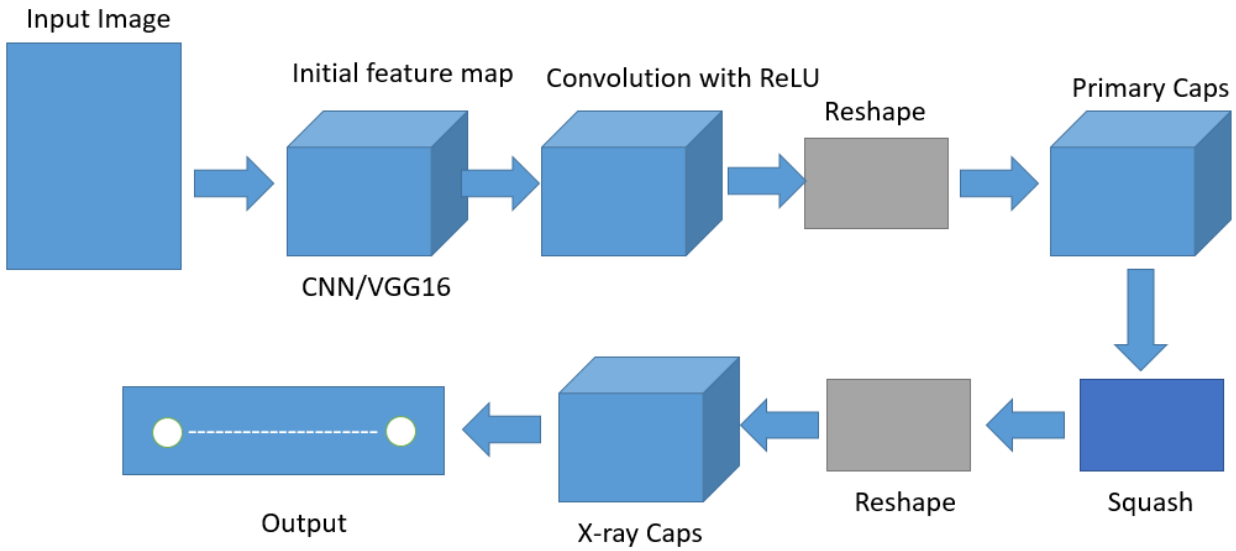

Figure 12. VGG-CapsNet architecture.

The researchers in [81] proposed a capsule network called MIXCAPS. The proposed model is based on a mixture of experts, automatically splits the dataset through a gating network based on convolution, and does not require fine annotation. The proposed approach is independent of pre-defined hand-shaped features, with an accuracy of 92.88%, a sensitivity of 93.2%, and a specificity of 92.3%. The MIXCAPs model is shown in Figure 13.

Based on its promising performance in other applications, the authors presented the CapsNet model for drowsiness detection using spectrogram images [82]. The proposed model was compared with a CNN, which was outperformed by the proposed model, which obtained an accuracy of 86.44% against an accuracy of 75.86% by the CNN. Figure 14 presents the proposed CapsNet approach.

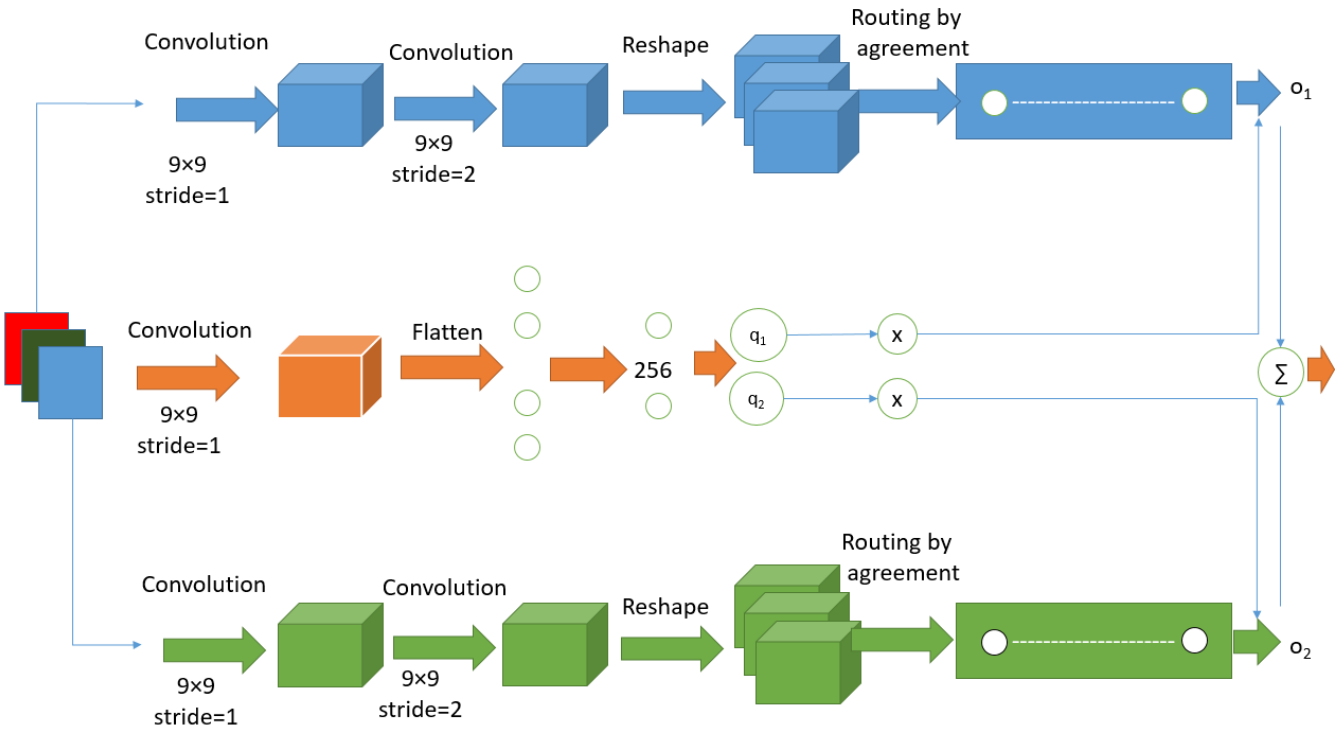

Figure 13. MixCaps architecture.

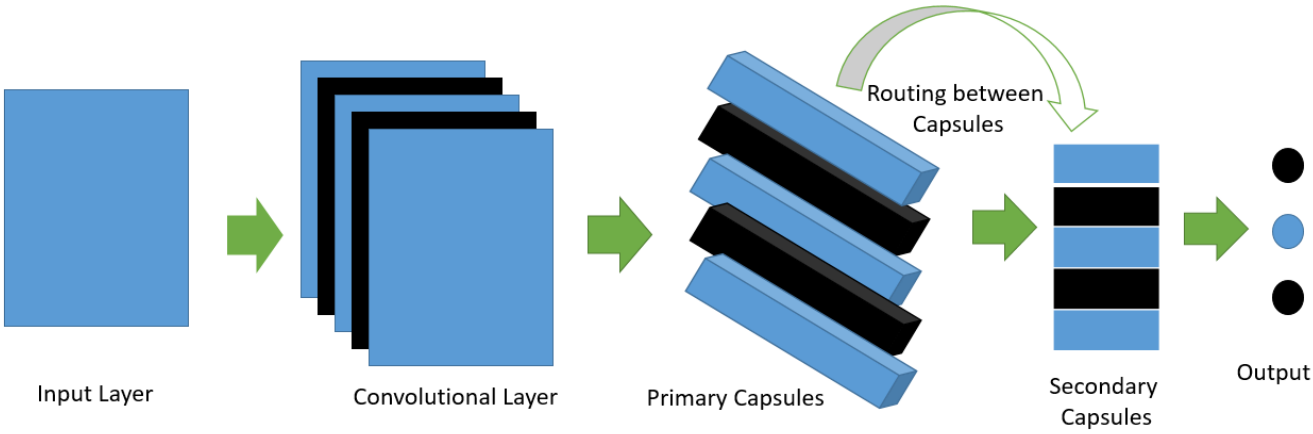

Figure 14. CapsNet model.

The authors in [83] proposed a disease-related compound identification model-based five-layer capsule network to correctly identify the disease-related compound. The performance of CapsNet was compared with that of SVM, gcForest, RF, and forgeNet. Based on their results, the authors claim that CapsNet gets better ROC curves as compared to other techniques and makes an improvement of 1.7–12.9% in terms of AUC.

The authors in [84] presented a framework of capsule networks for Urdu handwritten digit recognition. The authors collected their own dataset of handwritten digits from 900 people, with 6086 training images and 1301 images for testing. The proposed framework achieves promising results (98.5% accuracy) as compared to deep auto-encoder (97.3% accuracy) and CNN (96% accuracy).

The authors in [85] proposed a variant of capsule networks (multi-level capsule networks) for five different types of military object recognition. The proposed approach outclasses CNNs based on its performance and achieved an accuracy of 96.54%. Proposed approaches are presented in Figure 15. Table 2 presents the performance summary of

proposed capsule networks with their architecture, dataset used, comparison with state-of-the-art algorithms, accuracy, and limitations.

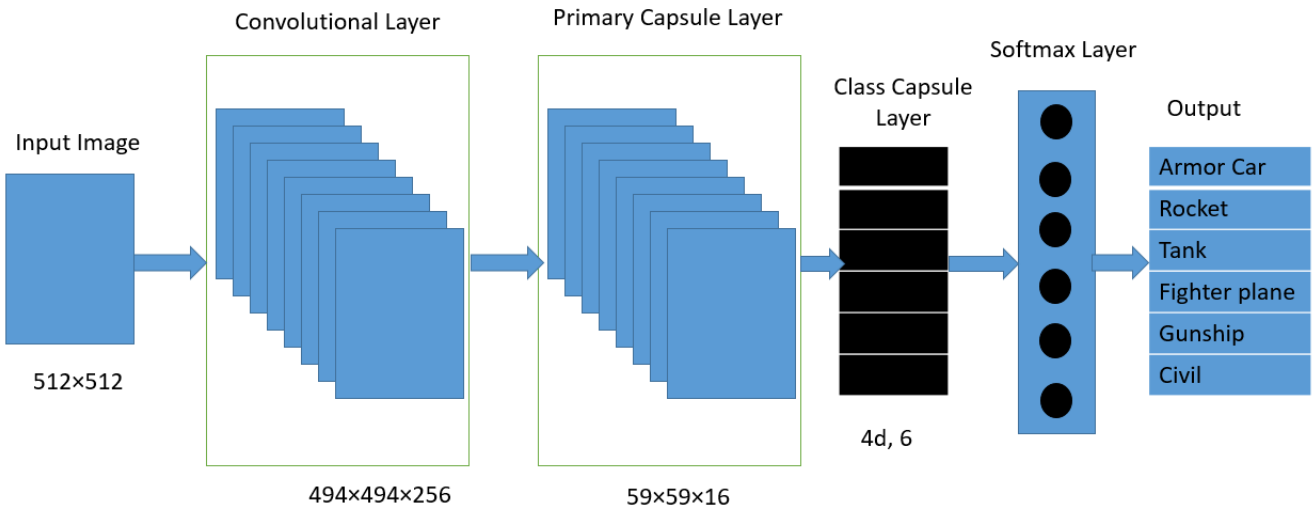

Figure 15. Multi-level capsule network.

Table 2. Performance summary of numerous capsule networks.

Ref No	Application/Problem Definition	Architecture and Parameters	Dataset	Compared with	Accuracy (%)	Recommendation/ Limitation
[69]	Brain tumor type classification.	Primary layer: 64 × 9 × 9 convolution filters and stride of 1. Primary capsule layer: 256 × 9 × 9 convolutions with strides of 2. Decoder: Fully connected layers with 512 × 1024 × 4096 neurons.	Dataset used in [86].	CNN presented by [87].	86.56.	Accuracy can be increased by varying the number of feature maps.
[70]	Comparative analysis of CapsNet with Fisherfaces, LeNet, and ResNet.	256 feature maps, using a 9 × 9 kernel and valid padding.	Yale Face database B, MIT CBCL Face dataset, Belgium TS Traffic Sign dataset and CIFAR-100.	Fisherfaces, LeNet, and ResNet.	95.3% on Yale dataset, 99.87% on MIT CBCl, 92% on Belgium TS Traffic Sign dataset, and 18% accuracy on CIFAR-100.	With more training iterations, CapsNet may have better results.
[71]	3D image generation to find improved discriminators for generative adversarial network (GAN).	Dynamic routing: Convolution 1 layer: 32 × 32 input, with 9 × 9 kernel of stride 1 Primary capsule layer: 32 channels each 8 × 8 × 8, 8D. Output capsule layer: 16 × 1.	MNIST and Small NORB.	DCGAN [88].		The MNIST used in this paper is a simplistic image, and additional experiments are needed using complex datasets. CapsGAN has the ability to capture geometric transformations.
[65]	To classify ultrasonic data for self-driving cars.	Dynamic routing by agreement. Convolution: 256 kernels of 6 × 6 size, Activation function: ReLU. Primary Capsule: 32 channels of 6 × 6 × 8. Digit Caps: 16 × 4.	A dataset of 21,600 measurements.	Complex CNN.	99.6% using complex CapsNet as compared to CNN (98.9).	A lot of research is needed to make ultrasonic technology appropriate for self-driving vehicles.
[73]	Face recognition.	A 3-layer capsule network having two convolutional and one fully connected layer. Primary capsules: 32 channels of convolutional 8-dimensional capsules. Final layer: 16 dimensional capsules per class.	LFW dataset.	Deep CNN.	93.7.	Due to their unique equivariance properties, CapsNets can perform better than CNN on unseen transformed data. Experimenting with larger training set size and fewer epochs could avoid overfitting and improve accuracy.

Table 2. *Cont.*

Ref No	Application/Problem Definition	Architecture and Parameters	Dataset	Compared with	Accuracy (%)	Recommendation/ Limitation
[74]	To identify the animals in the wilderness.	C-CapsNet.	Serengeti dataset.	CNN.	96.48.	-
[75]	Capsule network with self-attention routing.	Non-iterative, parallelizable routing algorithm instead of dynamic routing.	MNIST, smallNORB, and MultiMNIST.	CNN.	-	Achieved higher accuracy with a considerably lower number of parameters.
[76]	PT-CapsNet for semantic segmentation, classification, and detection tasks.	Prediction-tuning capsule network (PT-CapsNet) with connected PT capsules and locally connected PT capsules.	CIFAR-10, CIFAR-100, Fashion-MNIST, DenseNet-100, ResNet-110.	DenseNet, ResNet, and CNN.	-	PT-CapsNet can perform better than CNN-based models with challenging datasets, higher image sizes, with a smaller number of network parameters.
[77]	Hierarchical multi-label text classification.	The features encoded in capsules and routing algorithm are combined.	BGC and WOS datasets.	SVM, LSTM, ANN, and CNN architectures.	-	Proposed algorithm performs efficiently. Future work should involve cascading capsule layers.
[78]	COVID-19 patient detection through chest CT scan images.	U-net-based segmentation model, capsule network.	COVID-CT-MD.	-	90.82.	Proposed algorithm has been tested using a simple dataset of 171 images of COVID-19 positive patients, 60 patients with pneumonia, and 76 normal patients.
[79]	Cyberbullying detection.	CapsNet with dynamic routing and CNN.	10,000 comments taken from YouTube, Twitter, and Instagram.	KNN, SVM, and NB.	97.05.	High-dimensional, skewed, cross-lingual, and heterogeneous data are the limitations of proposed approach. Future work can be directed toward detecting and recognizing wordplay, creative spellings, and slang words.
[80]	To detect COVID-19 disease by means of radiography images.	VGG-CapsNet.	2905 images having 219, 1345, and 1341 images of COVID-19 patients available at [89].	CNN-CapsNet.	VGG-CapsNet has 97% for classifying COVID-19, non-COVID samples, and pneumonia with 92% accuracy.	Proposed approach can be used for clinical practices.
[81]	Lung nodule malignancy prediction.	MIXCAPS.	LIDC dataset [90] and IDRI dataset [91].	-	92.88%, with sensitivity of 93.2% and specificity of 92.3%.	The proposed approach is independent of pre-defined hand-shaped features and does not require fine annotation.
[82]	Drowsiness detection.	Generic CapsNet.	EEG signals.	CNN.	86.44.	To improve drowsiness detection, a proper dataset will be required in the future.
[83]	To identify pneumonia-related compounds.	Five-layer capsule network.	88 positive samples and 264 negative samples.	SVM, gcForest, RF, and forgeNet.	An improvement of 1.7–12.9% in terms of AU.	-
[84]	Urdu digit recognition.	Primary capsule layer: 8 convolutional units with two strides, 32 channels, and 9 × 9 kernel. Batch size = 100. Epochs = 50.	Own collected dataset of handwritten characters and digits of 900 people with 6086 training images and 1301 images for testing.	Autoencoder [92] and CNN.	97.3.	The assumption that one pixel of an image has at most one instance type is a challenging task. So that it can be accurately represented by capsule.
[85]	Military vehicle recognition.	Multi-level CapsNet with class capsule layer.	3500 images collected from the Internet.	CNN.	96.54%.	-

Table 2. *Cont.*

Ref No	Application/Problem Definition	Architecture and Parameters	Dataset	Compared with	Accuracy (%)	Recommendation/ Limitation
[93]	Emotion recognition.	Features: Feature matrix Convolution 1 layer: 16 × 16 × 256 channels. Primary capsule layer: 7 × 7 × 256D vector. Class capsule layer: 2 × 49 × 32D. Dynamic routing.	DEAP dataset [94].	Compared with five versions of capsule networks, SVM, Bayes classifier and Gaussian naive Bayes.	Arousal = 0.6828 valence = 0.6673 dominance = 0.6725.	To verify its comprehensiveness, the proposed method will be tested on more datasets of emotion recognition.
[95]	Emotion of tweets prediction.	Embedding layer, Bi-GRU layer with capsule network. Flattening with SoftMax layer. Dynamic routing by agreement.	WASSA 2018.	GRU with hierarchical attention, GRU with CNN. GRU with CapsNet.	F1 score = 0.692.	
[96]	Brain tumor classification.	Same as traditional CapsNet. Instead of 256 feature maps in convolutional layer, they used 64.	Dataset used in [97].	CNN.		
[98]	Lung cancer screening.	Encoder: 32 × 32 image to 24 × 24 with 256 channels. Primary capsule: 8 × 8 NoduleCapsule: 64 × 256 × 16 each. Decoder: 8 × 8 × 16.	226 unique CT scan images.	CNN.	Three times faster with the same accuracy.	In the future, unsupervised learning will be explored with CNN.
[99]	Biometric recognition system (face and iris).	Fuzzified image filter is used as preprocessing step to remove background noise. Before the capsule network, Gabor wavelet transform is used to detect and extract the vascular pattern of eye retinal images. Max pooling enabled with dynamic routing.	Face 95 [100] and CASIA-Iris-Thousand [101].	CNN.	99% accuracy with an error rate of 0.3% to 0.5%.	To implement the proposed method in public sector for biometric recognition.
[102]	Low-resolution image recognition.	DirectCapsNet with targeted reconstruction loss and HR-anchor loss.	CMU multi-PIE dataset [103], SVHN dataset, and UCCS dataset.	Robust partially coupled nets, LMSoftmax, L2Softmax, and Centerloss for VLR.	95.81%.	In the future, VLR FR can be performed in the presence of aging, adversarial attacks, and spectral variations.
[104]	Audio classification.	Agreement-based dynamic routing, flattening capsule network, Bi-GRU layer, SoftMax function, and embedding layer.	WASSA implicit emotion shared task [105].	GRU + CNN. GRU + hierarchical attention.	50%.	-
[106]	Image classification of gastrointestinal endoscopy.	Combination of CapsNet and midlevel CNN features (L-DenseNetCaps).	HyperKvasir dataset [107] and Kvasir v2 dataset [108].	VGG16, DenseNet121, DenseNetCaps, and L-VGG16Caps.	94.83.	This model can also be applied to the diagnosis of skin cancer, etc.
[109]	Hate speech detection.	HCovBi-Caps (convolutional, BiGRU, and capsule network).	DS1 dataset and DS2.	DNN, BiGRU, GRU, CNN, LSTM, etc.	Training accuracy = 0.93, and validation accuracy = 0.90.	The proposed model detects date propagation in speech only.
[110]	Object detection.	NASGC-CapANet.	MS COCO Val 2017 dataset.	Faster R-CNN.	43.8% box mAP.	When compared to the present attention mechanism, performance is improved by incorporating the capsule attention module into the highest level of FPN.

10. Routing Algorithms

Capsule networks, a type of neural network design, aim to get around some of the drawbacks of conventional convolutional neural networks (CNNs) in applications such as object recognition and natural language processing. Routing algorithms are a key component of capsule networks. Routing methods are employed in capsule networks to compute the coupling coefficients, which control how much data is transported between

capsules in different layers. Table 3 presents some common routing algorithms used in capsule networks.

Table 3. Routing algorithms.

Ref	Routing Algorithms	Description	Advantages	Applications	Limitation
[18]	Dynamic routing	A routing algorithm where the weighting of each capsule's output depends on how well its prediction matches the output of the layer above. The network can learn to send information from lower-level capsules to higher-level capsules thanks to this weighting.	Better performance in complex tasks than static routing, ability to handle input transformations, translation equivariance, and viewpoint invariance.	Object recognition, speech recognition, natural language processing.	Computational complexity, sensitivity to initialization, and potential for overfitting.
[111]	EM routing	A routing technique that iteratively updates the coupling coefficients between capsules using the expectation-maximization (EM) algorithm. The coupling coefficients are changed at each iteration by maximizing a lower constraint on the data's logarithmic likelihood.	Better performance than dynamic routing, more stable convergence, and improved generalization.	Object recognition, speech recognition, natural language processing.	Computationally expensive, high memory requirements, and potential for getting stuck in local optima.
[112]	Sparse routing	A routing algorithm that selects a small subset of capsules to route information to the next layer based on a sparsity constraint. The selected capsules are those with the highest activations, and the remaining capsules are discarded.	Reduced computational complexity, improved scalability, and increased robustness to adversarial attacks.	Object recognition, speech recognition, natural language processing.	Information loss due to discarding capsules, reduced expressiveness, and potential for overfitting.
[113]	Deterministic routing	A routing algorithm where each capsule in the lower layer is deterministically assigned to a specific capsule in the layer above based on the maximum activation.	Simple and computationally efficient, easy to implement.	Object recognition, speech recognition, natural language processing.	Lack of robustness to input transformations and viewpoint changes.
[114]	Routing by agreement or competition (RAC)	A routing algorithm where each capsule sends its output to multiple capsules in the layer above, and the coupling coefficients are computed based on either the agreement or the competition between the outputs.	Increased flexibility and performance compared to other routing algorithms.	Object recognition, speech recognition, natural language processing.	Requires additional parameters to compute the agreement or competition, computationally expensive.

Table 3. *Cont.*

Ref	Routing Algorithms	Description	Advantages	Applications	Limitation
[115]	K-means routing	A routing algorithm where each capsule in the lower layer is assigned to the closest cluster center in the layer above, which is learned using K-means clustering.	Simple and computationally efficient, captures higher-level concepts in a more structured way.	Speech recognition, natural language processing, and object recognition.	Requires setting the number of cluster centers in advance, sensitive to the initialization of the cluster centers.
[116]	Routing with temporal dynamics	A routing algorithm that uses recurrent neural networks to capture the temporal dynamics of the capsule activations over time.	Ability to handle sequential data, improved performance in video recognition tasks.	Video recognition, action recognition, speech recognition.	Higher computational complexity, potential for overfitting, requires careful initialization.
[117]	Localized routing	A routing algorithm that uses a localized attention mechanism to route information from lower-level capsules to nearby higher-level capsules.	Better performance than global routing, more robust to input transformations and viewpoint changes.	Object recognition, speech recognition, natural language processing.	Higher computational complexity, requires additional parameters to compute the attention.
[118]	Attention-based routing	A routing algorithm that uses an attention mechanism to weigh the contributions of each capsule in the lower layer to each capsule in the layer above.	Improved performance compared to other routing algorithms, can capture complex relationships between capsules.	Object recognition, speech recognition, natural language processing.	Higher computational complexity, requires additional parameters to compute the attention.
[119]	Graph-based routing	A routing algorithm that models the relationships between capsules in the lower layer and the layer above as a graph and performs message passing between the nodes in the graph to compute the coupling coefficients.	Better performance in capturing hierarchical relationships between objects, more robust to input transformations and viewpoint changes.	Object recognition, natural language processing.	Higher computational complexity, requires additional parameters to model the graph structure.
[120]	Dynamic routing with routing signals	A routing algorithm that uses a dynamic routing approach combined with routing signals to adjust the coupling coefficients during the inference stage.	Improved performance compared to other routing algorithms, can capture more complex relationships between capsules.	Object recognition, speech recognition, natural language processing.	Higher computational complexity, requires additional parameters to compute the routing signals.

Table 4 lists the various routing algorithms used in capsule networks, discusses their benefits and limits, and ranks them according to how well they perform in various areas.

Table 4. Comparison of different routing algorithms.

Routing Algorithm	Advantages	Limitations	Best Aspect	Worst Aspect
Dynamic routing	Allows dynamic learning of activations	Computationally expensive with long sequences	General performance	Computational complexity
EM routing	Improved convergence and stability	Can still be computationally expensive	Convergence stability	Potential Local Optima
MaxMin routing	Reduces uniform probabilities	Parameter tuning for optimal balance	Probabilities distribution	Parameter Tuning
Routing by greement	Simplifies routing process	Suboptimal results without correct iteration choice	Simplicity	Suboptimal iteration choice
Orthogonal Routing	Improved stability and generalization	Additional regularization for orthogonality	Stability and invariance	Computational overhead
Group Equivariant capsules	Equivariance and invariance guarantees	Complex implementation and group choice	Equivariance and invariance	Complex implementation
Sparse routing	Efficient utilization of computation	Complex selection process for active capsules	Computation efficiency	Selection complexity
Deterministic routing	Deterministic routing for stable output	Restricted to deterministic assignment of capsules	Deterministic outputs	Limited capsule interaction
K-mean routing	Efficient use of clustering	May not handle complex data distributions	Efficient clustering	Limited data distributions
Routing with temporal dynamics	Considers temporal information	Increased complexity in modeling temporal relationships	Temporal information	Increased model complexity
Graph-based routing	Captures spatial relationships	Computational overhead in graph-based operations	Spatial relationships	Computational overhead
Attention-based routing	Focuses on informative capsules	May require careful tuning and be sensitive to hyperparameters	Attention mechanisms	Hyperparameter sensitivity

11. Future Research Directions

Although Geoff Hinton and his team's CapsNet architecture has demonstrated potential in a number of tasks, further effort and advancements are still needed. Following are a few suggested directions for further study on capsule networks:

11.1. Better Routing Mechanisms

CapsNets' existing routing-by-agreement mechanism has some drawbacks. Alternative routing techniques may be investigated in the future to increase CapsNets' training effectiveness and convergence.

11.2. Handling Big Data

The performance of CapsNets on big data could be investigated and improved. To properly manage massive datasets, methods such as distributed training and parallel processing may be researched.

11.3. Task Adaptation

CapsNets have mainly been investigated for image-related tasks. Future studies can look into how well they work and whether they can be applied to different fields such as video analysis and natural language processing.

11.4. CapsNets in Transfer Learning

It may be investigated to better understand and utilize CapsNets in transfer learning scenarios. Pre-training on big datasets and fine-tuning on smaller target datasets are possible research areas.

11.5. Interpretability and Explainability

An area of study is the interpretability of capsule networks. Future research can concentrate on approaches to improving the interpretability of CapsNets predictions, allowing users to comprehend how the model makes judgments.

11.6. CapsNets for Few-Shot Learning

Examine the suitability of CapsNets for jobs requiring the model to generalize from a small number of samples per class.

11.7. Dynamic Routing Improvements

Look for ways to make the dynamic routing algorithm more effective at handling longer sequences of capsules while retaining pertinent spatial data.

11.8. CapsNets in Reinforcement Learning

Examine CapsNets' potential in tasks requiring reinforcement learning, in which the model interacts with its surroundings and gains knowledge through trial and error.

11.9. CapsNets with Attention Mechanisms

Examine the use of attention mechanisms in combination with CapsNets to concentrate on pertinent capsules or portions of the input, potentially enhancing performance and efficacy.

11.10. Capsule Networks for 3D Data

Learn how to use the CapsNets extension to work with 3D data, such as point clouds or volumetric data, for applications such as 3D object detection and reconstruction.

11.11. CapsNets in Generative Models

Examine CapsNets' potential for image synthesis and data production tasks in generative models.

Future research on capsule networks should concentrate on overcoming existing constraints, expanding their applicability to diverse domains, and investigating fresh ways to enhance their performance, interpretability, and scalability for real-time problems such as object segmentation [121], biometric identification [122,123], and so on. In the upcoming years, there will be many opportunities for ground-breaking research and fascinating developments because the subject of capsule networks is so new.

12. Conclusions

To eliminate the challenges faced by the traditional CNN algorithms, capsule networks were introduced, which have performed creditably well so far. However, to realize its full potential, the mentioned approach requires further understanding. This paper, therefore, presents a study comparing the efficiency of different algorithms in the literature that are impactful in the field. We have reviewed the implementation of existing capsule network architectures and shed light on their implementation results, limitations, and modifications. We have also reviewed several routing algorithms published in the literature. This survey will be helpful for the computer vision community to influence the failures and successes of capsule networks to design a robust machine vision algorithm through further research.

Author Contributions: Conceptualization, M.U.H. and M.A.J.S.; methodology, M.U.H. and M.A.J.S.; validation, M.U.H. and A.U.R.; data curation, M.U.H., M.A.J.S. and A.U.R.; writing—original draft preparation, M.U.H.; writing—review and editing, M.A.J.S. and A.U.R.; visualization, M.U.H.; supervision, M.A.J.S. and A.U.R.; funding acquisition, A.U.R. All authors have read and agreed to the published version of the manuscript.

Funding: This study did not receive any funding in any form.

Data Availability Statement: The article reviews existing studies only, which are available online on different platforms e-g google scholar etc.

Conflicts of Interest: The authors declare that there are no conflict of interest regarding the publication of this paper.

Nomenclatures

Notation	Description
CapsNet	Capsule network
CNN	Convolutional neural network
MNIST dataset	Modified National Institute of Standards and Technology dataset.
CIFAR	Canadian Institute for Advanced Research
RNN	Recurrent neural network
SVHN dataset	Street View House Number dataset
SVM	Support vector machine
ANN	Artificial neural network
KNN	K-nearest neighbor
CapsGAN	Capsule GAN
PT-CapsNet	Prediction-tuning capsule network
FC Layers	Fully connected layers
LSTM	Long short-term memory
GAN	Generative adversarial network
WOS	Web of Science
ResNet	Residual neural network
FR	Face recognition
SVM	Support vector machine
NB	Naïve Bayesian
GRU	Gated recurrent unit
DirectCapsNet	Dual-directed capsule network model
NN	Neural network
VLR	Very low resolution
ReLU	Rectified linear unit
BGC	Blurb genre collection
LFW	Labeled faces in the wild
NAS	Neural architecture search
NASCaps	NAS capsule

References

1. Fahad, K.; Yu, X.; Yuan, Z.; Rehman, A.U. ECG classification using 1-D convolutional deep residual neural network. *PLoS ONE* **2023**, *18*, e0284791.
2. Haq, M.U.; Shahzad, A.; Mahmood, Z.; Shah, A.A.; Muhammad, N.; Akram, T. Boosting the face recognition performance of ensemble based LDA for pose, non-uniform illuminations, and low-resolution images. *KSII Trans. Internet Inf. Syst.* **2019**, *13*, 3144–3164.
3. Rehman, A.U.; Aasia, K.; Arslan, S. Hybrid feature selection and tumor identification in brain MRI using swarm intelligence. In Proceedings of the 2013 11th International Conference on Frontiers of Information Technology, Islamabad, Pakistan, 16–18 December 2013; pp. 49–54.
4. Patel, C.I.; Garg, S.; Zaveri, T.; Banerjee, A. Top-down and bottom-up cues based moving object detection for varied background video sequences. *Adv. Multimed.* **2014**, *2014*, 879070. [CrossRef]
5. Zhang, L.; Leng, X.; Feng, S.; Ma, X.; Ji, K.; Kuang, G.; Liu, L. Azimuth-Aware Discriminative Representation Learning for Semi-Supervised Few-Shot SAR Vehicle Recognition. *Remote Sens.* **2023**, *15*, 331. [CrossRef]

6. Aamna, B.; Arif, A.; Khalid, W.; Khan, B.; Ali, A.; Khalid, S.; Rehman, A.U. Recognition and classification of handwritten urdu numerals using deep learning techniques. *Appl. Sci.* **2023**, *13*, 1624.
7. Patrick, M.K.; Adekoya, A.F.; Mighty, A.A.; Edward, B.Y. Capsule networks—A survey. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 1295–1310.
8. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. In Proceedings of the 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August 2017; pp. 1–6.
9. Zhang, D.; Wang, D. Relation classification via recurrent neural network. *arXiv* **2015**, arXiv:1508.01006.
10. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 777–780. [CrossRef]
11. Patel, C.; Bhatt, D.; Sharma, U.; Patel, R.; Pandya, S.; Modi, K.; Cholli, N.; Patel, A.; Bhatt, U.; Khan, M.A.; et al. DBGC: Dimension-based generic convolution block for object recognition. *Sensors* **2022**, *22*, 1780. [CrossRef]
12. Xi, E.; Bing, S.; Jin, Y. Capsule network performance on complex data. *arXiv* **2017**, arXiv:1712.03480.
13. Wang, Y.; Ning, D.; Feng, S. A novel capsule network based on wide convolution and multi-scale convolution for fault diagnosis. *Appl. Sci.* **2020**, *10*, 3659. [CrossRef]
14. Liu, S.; Wang, Z.; An, Y.; Zhao, J.; Zhao, Y.; Zhang, Y.-D. EEG emotion recognition based on the attention mechanism and pre-trained convolution capsule network. *Knowl.-Based Syst.* **2023**, *265*, 110372. [CrossRef]
15. Sreelakshmi, K.; Akarsh, S.; Vinayakumar, R.; Soman, K.P. Capsule neural networks and visualization for segregation of plastic and non-plastic wastes. In Proceedings of the 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 15–16 March 2019; pp. 631–636.
16. Li, Y.; Fu, K.; Sun, H.; Sun, X. An aircraft detection framework based on reinforcement learning and convolutional neural networks in remote sensing images. *Remote Sens.* **2018**, *10*, 243. [CrossRef]
17. Xu, T.B.; Cheng, G.L.; Yang, J.; Liu, C.L. Fast aircraft detection using end-to-end fully convolutional network. In Proceedings of the 2016 IEEE International Conference on Digital Signal Processing (DSP), Beijing, China, 16–18 October 2016; pp. 139–143.
18. Sabour, S.; Frosst, N.; Hinton, G.E. Dynamic routing between capsules Advances in Neural Information Processing Systems. In Proceedings of the Annual Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
19. Available online: https://en.wikipedia.org/wiki/Capsule_neural_network (accessed on 2 July 2023).
20. Wu, J. Introduction to convolutional neural networks. *Natl. Key Lab Nov. Softw. Technol. Nanjing Univ. China* **2017**, *5*, 495.
21. Kuo, C.C.J. Understanding convolutional neural networks with a mathematical model. *J. Vis. Commun. Image Represent.* **2016**, *41*, 406–413. [CrossRef]
22. Bhatt, D.; Patel, C.; Talsania, H.; Patel, J.; Vaghela, R.; Pandya, S.; Modi, K.; Ghayvat, H. CNN variants for computer vision: History, architecture, application, challenges and future scope. *Electronics* **2021**, *10*, 2470. [CrossRef]
23. Saha, S. A comprehensive guide to convolutional neural networks—The ELI5 way. *Towards Data Sci.* **2018**, *15*, 15.
24. Shahroudnejad, A.; Afshar, P.; Plataniotis, K.N.; Mohammadi, A. Improved explainability of capsule networks: Relevance path by agreement. In Proceedings of the 2018 IEEE Global Conference on Signal and Information Processing (GLOBALSIP), Anaheim, CA, USA, 26–29 November 2018; pp. 549–553.
25. Su, J.; Vargas, D.V.; Sakurai, K. One pixel attack for fooling deep neural networks. *IEEE Trans. Evol. Comput.* **2019**, *23*, 828–841. [CrossRef]
26. Gu, J. Interpretable graph capsule networks for object recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, virtual, 2–9 February 2021; Volume 35, pp. 1469–1477.
27. Gu, J.; Wu, B.; Tresp, V. Effective and efficient vote attack on capsule networks. *arXiv* **2021**, arXiv:2102.10055.
28. Hu, X.D.; Li, Z.H. Intrusion Detection Method Based on Capsule Network for Industrial Internet. *Acta Electonica Sin.* **2022**, *50*, 1457.
29. Devi, K.; Muthusenthil, B. Intrusion detection framework for securing privacy attack in cloud computing environment using DCCGAN-RFOA. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e4561. [CrossRef]
30. Marchisio, A.; Nanfa, G.; Khalid, F.; Hanif, M.A.; Martina, M.; Shafique, M. SeVuc: A study on the Security Vulnerabilities of Capsule Networks against adversarial attacks. *Microprocess. Microsyst.* **2023**, *96*, 104738. [CrossRef]
31. Wang, X.; Wang, Y.; Guo, S.; Kong, L.; Cui, G. Capsule Network with Multiscale Feature Fusion for Hidden Human Activity Classification. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 2504712. [CrossRef]
32. Tokish, J.M.; Makovicka, J.L. The superior capsular reconstruction: Lessons learned and future directions. *J. Am. Acad. Orthop. Surg.* **2020**, *28*, 528–537. [CrossRef]
33. Xiang, C.; Zhang, L.; Tang, Y.; Zou, W.; Xu, C. MS-CapsNet: A novel multi-scale capsule network. *IEEE Signal Process. Lett.* **2018**, *25*, 1850–1854. [CrossRef]
34. Kang, J.S.; Kang, J.; Kim, J.J.; Jeon, K.W.; Chung, H.J.; Park, B.H. Neural Architecture Search Survey: A Computer Vision Perspective. *Sensors* **2023**, *23*, 1713. [CrossRef]
35. Marchisio, A.; Massa, A.; Mrazek, V.; Bussolino, B.; Martina, M.; Shafique, M. NASCaps: A framework for neural architecture search to optimize the accuracy and hardware efficiency of convolutional capsule networks. In Proceedings of the 39th International Conference on Computer-Aided Design, Virtual, 2–5 November 2020; pp. 1–9.

36. Marchisio, A.; Mrazek, V.; Massa, A.; Bussolino, B.; Martina, M.; Shafique, M. RoHNAS: A Neural Architecture Search Framework with Conjoint Optimization for Adversarial Robustness and Hardware Efficiency of Convolutional and Capsule Networks. *IEEE Access* **2022**, *10*, 109043–109055. [CrossRef]
37. Haq, M.U.; Sethi, M.A.J.; Ullah, R.; Shazhad, A.; Hasan, L.; Karami, G.M. COMSATS Face: A Dataset of Face Images with Pose Variations, Its Design, and Aspects. *Math. Probl. Eng.* **2022**, *2022*, 4589057. [CrossRef]
38. Gordienko, N.; Kochura, Y.; Taran, V.; Peng, G.; Gordienko, Y.; Stirenko, S. Capsule deep neural network for recognition of historical Graffiti handwriting. *arXiv* **2018**, arXiv:1809.06693.
39. Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A.Y. Reading Digits in Natural Images with Unsupervised Feature Learning. 2011. Available online: https://www.researchgate.net/publication/266031774_Reading_Digits_in_Natural_Images_with_Unsupervised_Feature_Learning (accessed on 2 July 2023).
40. Krizhevsky, A.; Hinton, G. Learning Multiple Layers of Features from Tiny Images. 2009. Available online: https://www.bibsonomy.org/bibtex/cc2d42f2b7ef6a4e76e47d1a50c8cd86 (accessed on 2 July 2023).
41. Zhang, X. The AlexNet, LeNet-5 and VGG NET applied to CIFAR-10. In Proceedings of the 2021 2nd International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE), Zhuhai, China, 24–26 September 2021; pp. 414–419.
42. Doon, R.; Rawat, T.K.; Gautam, S. Cifar-10 classification using deep convolutional neural network. In Proceedings of the 2018 IEEE Punecon, Pune, India, 30 November–2 December 2018; pp. 1–5.
43. Jiang, X.; Wang, Y.; Liu, W.; Li, S.; Liu, J. Capsnet, cnn, fcn: Comparative performance evaluation for image classification. *Int. J. Mach. Learn. Comput.* **2019**, *9*, 840–848. [CrossRef]
44. Nair, P.; Doshi, R.; Keselj, S. Pushing the limits of capsule networks. *arXiv* **2021**, arXiv:2103.08074.
45. Gu, J.; Tresp, V.; Hu, H. Capsule network is not more robust than convolutional network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 14309–14317.
46. Amer, M.; Maul, T. Path capsule networks. *Neural Process. Lett.* **2020**, *52*, 545–559. [CrossRef]
47. Peer, D.; Stabinger, S.; Rodriguez-Sanchez, A. Training deep capsule networks. *arXiv* **2018**, arXiv:1812.09707.
48. Jaiswal, A.; AbdAlmageed, W.; Wu, Y.; Natarajan, P. Capsulegan: Generative adversarial capsule network. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018.
49. Saqur, R.; Vivona, S. Capsgan: Using dynamic routing for generative adversarial networks. In *Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC)*; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; Volume 21, pp. 511–525.
50. Pérez, E.; Ventura, S. Melanoma recognition by fusing convolutional blocks and dynamic routing between capsules. *Cancers* **2021**, *13*, 4974. [CrossRef]
51. Ramasinghe, S.; Athuraliya, C.D.; Khan, S.H. A context-aware capsule network for multi-label classification. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018.
52. Zhang, L.; Edraki, M.; Qi, G.J. Cappronet: Deep feature learning via orthogonal projections onto capsule subspaces. In Proceedings of the Annual Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; Volume 31.
53. Zhao, Z.; Kleinhans, A.; Sandhu, G.; Patel, I.; Unnikrishnan, K.P. Capsule networks with max-min normalization. *arXiv* **2019**, arXiv:1903.09662.
54. Phaye, S.; Samarth, R.; Sikka, A.; Dhall, A.; Bathula, D. Dense and diverse capsule networks: Making the capsules learn better. *arXiv* **2018**, arXiv:1805.04001.
55. Larsson, G.; Maire, M.; Shakhnarovich, G. Fractalnet: Ultra-deep neural networks without residuals. *arXiv* **2016**, arXiv:1605.07648.
56. Chen, Z.; Crandall, D. Generalized capsule networks with trainable routing procedure. *arXiv* **2018**, arXiv:1808.08692.
57. Nguyen, H.P.; Ribeiro, B. Advanced capsule networks via context awareness. In *Artificial Neural Networks and Machine Learning—ICANN 2019: Theoretical Neural Computation, Proceedings of the 28th International Conference on Artificial Neural Networks, Munich, Germany, 17–19 September 2019*; Springer International Publishing: Berlin/Heidelberg, Germany, 2019; Volume 28, Part I; pp. 166–177.
58. Lenssen, J.E.; Fey, M.; Libuschewski, P. Group equivariant capsule networks. In Proceedings of the Annual Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; Volume 31.
59. Wang, D.; Liu, Q. An Optimization View on Dynamic Routing between Capsules. 2018. Available online: https://openreview.net/forum?id=HJjtFYJDf (accessed on 2 July 2023).
60. Kumar, A.D. Novel deep learning model for traffic sign detection using capsule networks. *arXiv* **2018**, arXiv:1805.04424.
61. Mandal, B.; Dubey, S.; Ghosh, S.; Sarkhel, R.; Das, N. Handwritten indic character recognition using capsule networks. In Proceedings of the 2018 IEEE Applied Signal Processing Conference (ASPCON), Kolkata, India, 7–9 December 2018; pp. 304–308.
62. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
63. Xia, C.; Zhang, C.; Yan, X.; Chang, Y.; Yu, P.S. Zero-shot user intent detection via capsule neural networks. *arXiv* **2018**, arXiv:1809.00385.
64. Kim, M.; Chi, S. Detection of centerline crossing in abnormal driving using CapsNet. *J. Supercomput.* **2019**, *75*, 189–196. [CrossRef]
65. Ma, X.; Dai, Z.; He, Z.; Ma, J.; Wang, Y.; Wang, Y. Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors* **2017**, *17*, 818. [CrossRef]

66. Fezza, S.A.; Bakhti, Y.; Hamidouche, W.; Déforges, O. Perceptual evaluation of adversarial attacks for CNN-based image classification. In Proceedings of the 2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX), Berlin, Germany, 5–7 June 2019; pp. 1–6.
67. Badue, C.; Guidolini, R.; Carneiro, R.V.; Azevedo, P.; Cardoso, V.B.; Forechi, A.; De Souza, A.F. Self-driving cars: A survey. *Expert Syst. Appl.* **2021**, *165*, 113816. [CrossRef]
68. Weng, Z.; Meng, F.; Liu, S.; Zhang, Y.; Zheng, Z.; Gong, C. Cattle face recognition based on a Two-Branch convolutional neural network. *Comput. Electron. Agric.* **2022**, *196*, 106871. [CrossRef]
69. Afshar, P.; Mohammadi, A.; Plataniotis, K.N. Brain tumor type classification via capsule networks. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 3129–3133.
70. Mukhometzianov, R.; Carrillo, J. CapsNet comparative performance evaluation for image classification. *arXiv* **2018**, arXiv:1805.11195.
71. Saqur, R.; Vivona, S. Capsgan: Using dynamic routing for generative adversarial networks. *arXiv* **2018**, arXiv:1806.03968.
72. Guarda, L.; Tapia, J.E.; Droguett, E.L.; Ramos, M. A novel Capsule Neural Network based model for drowsiness detection using electroencephalography signals. *Expert Syst. Appl.* **2022**, *201*, 116977. [CrossRef]
73. Chui, A.; Patnaik, A.; Ramesh, K.; Wang, L. Capsule Networks and Face Recognition. 2019. Available online: https://lindawangg.github.io/projects/capsnet.pdf (accessed on 2 July 2023).
74. Teto, J.K.; Xie, Y. Automatically Identifying of animals in the wilderness: Comparative studies between CNN and C-Capsule Network. In Proceedings of the 2019 3rd International Conference on Compute and Data Analysis, Kahului, HI, USA, 6–8 March 2019; pp. 128–133.
75. Mazzia, V.; Salvetti, F.; Chiaberge, M. Efficient-capsnet: Capsule network with self-attention routing. *Sci. Rep.* **2021**, *11*, 14634. [CrossRef]
76. Pan, C.; Velipasalar, S. PT-CapsNet: A novel prediction-tuning capsule network suitable for deeper architectures. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 11996–12005.
77. Manoharan, J.S. Capsule Network Algorithm for Performance Optimization of Text Classification. *J. Soft Comput. Paradig.* **2021**, *3*, 1–9. [CrossRef]
78. Heidarian, S.; Afshar, P.; Enshaei, N.; Naderkhani, F.; Rafiee, M.J.; Fard, F.B.; Samimi, K.; Atashzar, S.F.; Oikonomou, A.; Plataniotis, K.N.; et al. Covid-fact: A fully-automated capsule network-based framework for identification of COVID-19 cases from chest ct scans. *Front. Artif. Intell.* **2021**, *4*, 8932. [CrossRef]
79. Kumar, A.; Sachdeva, N. Multimodal cyberbullying detection using capsule network with dynamic routing and deep convolutional neural network. *Multimed. Syst.* **2021**, *28*, 2043–2052. [CrossRef]
80. Tiwari, S.; Jain, A. Convolutional capsule network for COVID-19 detection using radiography images. *Int. J. Imaging Syst. Technol.* **2021**, *31*, 525–539. [CrossRef]
81. Afshar, P.; Naderkhani, F.; Oikonomou, A.; Rafiee, M.J.; Mohammadi, A.; Plataniotis, K.N. MIXCAPS: A capsule network-based mixture of experts for lung nodule malignancy prediction. *Pattern Recognit.* **2021**, *116*, 107942. [CrossRef]
82. Pöpperli, M.; Gulagundi, R.; Yogamani, S.; Milz, S. Capsule neural network-based height classification using low-cost automotive ultrasonic sensors. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 661–666.
83. Yang, B.; Bao, W.; Wang, J. Active disease-related compound identification based on capsule network. *Brief. Bioinform.* **2022**, *23*, bbab462. [CrossRef]
84. Iqbal, T.; Ali, H.; Saad, M.M.; Khan, S.; Tanougast, C. Capsule-Net for Urdu Digits Recognition. In Proceedings of the 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Metz, France, 18–21 September 2019; Volume 1, pp. 495–499.
85. Janakiramaiah, B.; Kalyani, G.; Karuna, A.; Prasad, L.V.; Krishna, M. Military object detection in defense using multi-level capsule networks. *Soft Comput.* **2021**, *27*, 1045–1059. [CrossRef]
86. Cheng, J.; Huang, W.; Cao, S.; Yang, R.; Yang, W.; Yun, Z.; Wang, Z.; Feng, Q. Enhanced performance of brain tumor classification via tumor region augmentation and partition. *PLoS ONE* **2015**, *10*, e0140381. [CrossRef]
87. Paul, J.S.; Plassard, A.J.; Landman, B.A.; Fabbri, D. Deep learning for brain tumor classification. In *Medical Imaging 2017: Biomedical Applications in Molecular, Structural, and Functional Imaging*; SPIE: Bellingham, WA, USA, 2017; Volume 10137, pp. 253–268.
88. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434.
89. Available online: https://www.kaggle.com/datasets/andrewmvd/convid19-x-rays (accessed on 2 July 2023).
90. Armato, S.G., III; McLennan, G.; Bidaut, L.; McNitt-Gray, M.F.; Meyer, C.R.; Reeves, A.P.; Zhao, B.; Aberle, D.R.; Henschke, C.I.; Hoffman, E.A.; et al. The lung image database consortium (LIDC) and image database resource initiative (IDRI): A completed reference database of lung nodules on CT scans. *Med. Phys.* **2011**, *38*, 915–931. [CrossRef]
91. Clark, K.; Vendt, B.; Smith, K.; Freymann, J.; Kirby, J.; Koppel, P.; Moore, S.; Phillips, S.; Maffitt, D.; Pringle, M.; et al. The Cancer Imaging Archive (TCIA): Maintaining and operating a public information repository. *J. Digit. Imaging* **2013**, *26*, 1045–1057. [CrossRef]
92. Ali, H.; Ullah, A.; Iqbal, T.; Khattak, S. Pioneer dataset and automatic recognition of Urdu handwritten characters using a deep autoencoder and convolutional neural network. *SN Appl. Sci.* **2020**, *2*, 152. [CrossRef]

93. Chao, H.; Dong, L.; Liu, Y.; Lu, B. Emotion recognition from multiband EEG signals using CapsNet. *Sensors* **2019**, *19*, 2212. [CrossRef]
94. Koelstra, S.; Muhl, C.; Soleymani, M.; Lee, J.S.; Yazdani, A.; Ebrahimi, T.; Pun, T.; Nijholt, A.; Patras, I. Deap: A database for emotion analysis; using physiological signals. *IEEE Trans. Affect. Comput.* **2011**, *3*, 18–31. [CrossRef]
95. Rathnayaka, P.; Abeysinghe, S.; Samarajeewa, C.; Manchanayake, I.; Walpola, M. Sentylic at IEST 2018: Gated recurrent neural network and capsule network-based approach for implicit emotion detection. *arXiv* **2018**, arXiv:1809.01452.
96. Afshary, P.; Mohammadiy, A.; Plataniotis, K. Brain tumor type classification via capsule networks. *arXiv* **2018**, arXiv:1802.10200.
97. Cheng, J.; Yang, W.; Huang, M.; Huang, W.; Jiang, J.; Zhou, Y.; Yang, R.; Zhao, J.; Feng, Y.; Feng, Q.; et al. Retrieval of brain tumors by adaptive spatial pooling and fisher vector representation. *PLoS ONE* **2016**, *11*, e0157112. [CrossRef]
98. Mobiny, A.; Nguyen, H.V. Fast capsnet for lung cancer screening. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Online, 26 September 2018; Springer: Cham, Switzerland, 2018; pp. 741–749.
99. Jacob, I.J. Capsule network based biometric recognition system. *J. Artif. Intell.* **2019**, *1*, 83–94.
100. Gunasekaran, K.; Raja, J.; Pitchai, R. Deep multimodal biometric recognition using contourlet derivative weighted rank fusion with human face, fingerprint and iris images. *Autom. Časopis Autom. Mjer. Elektron. Računarstvo Komun.* **2019**, *60*, 253–265. [CrossRef]
101. Zhao, T.; Liu, Y.; Huo, G.; Zhu, X. A deep learning iris recognition method based on capsule network architecture. *IEEE Access* **2019**, *7*, 49691–49701. [CrossRef]
102. Singh, M.; Nagpal, S.; Singh, R.; Vatsa, M. Dual directed capsule network for very low-resolution image recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 340–349.
103. Sim, T.; Baker, S.; Bsat, M. The CMU pose, illumination, and expression (PIE) database. In Proceedings of the Fifth IEEE International Conference on Automatic Face Gesture Recognition, Washington, DC, USA, 21 May 2002; pp. 53–58.
104. Jain, R. Improving performance and inference on audio classification tasks using capsule networks. *arXiv* **2019**, arXiv:1902.05069.
105. Klinger, R.; De Clercq, O.; Mohammad, S.M.; Balahur, A. IEST: WASSA-2018 implicit emotions shared task. *arXiv* **2018**, arXiv:1809.01083.
106. Wang, W.; Yang, X.; Li, X.; Tang, J. Convolutional-capsule network for gastrointestinal endoscopy image classification. *Int. J. Intell. Syst.* **2022**, *22*, 815. [CrossRef]
107. Pogorelov, K.; Randel, K.R.; Griwodz, C.; Eskeland, S.L.; de Lange, T.; Johansen, D.; Spampinato, C.; Dang-Nguyen, D.T.; Lux, M.; Schmidt, P.T.; et al. Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection. In Proceedings of the 8th ACM on Multimedia Systems Conference, Taipei, Taiwan, 20–23 June 2017; pp. 164–169.
108. Borgli, H.; Thambawita, V.; Smedsrud, P.H.; Hicks, S.; Jha, D.; Eskeland, S.L.; Randel, K.R.; Pogorelov, K.; Lux, M.; Nguyen, D.T.D.; et al. HyperKvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy. *Sci. Data* **2020**, *7*, 283. [CrossRef]
109. Khan, S.; Kamal, A.; Fazil, M.; Alshara, M.A.; Sejwal, V.K.; Alotaibi, R.M.; Baig, A.R.; Alqahtani, S. HCovBi-caps: Hate speech detection using convolutional and Bi-directional gated recurrent unit with Capsule network. *IEEE Access* **2022**, *10*, 7881–7894. [CrossRef]
110. Hinton, G.E.; Sabour, S.; Frosst, N. Matrix capsules with EM routing. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
111. Viriyasaranon, T.; Choi, J.H. Object detectors involving a NAS-gate convolutional module and capsule attention module. *Sci. Rep.* **2022**, *12*, 3916. [CrossRef]
112. Sabour, S.; Frosst, N.; Hinton, G. Matrix capsules with EM routing. In Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 15 February 2018; Volume 115.
113. Gomez, C.; Gilabert, F.; Gomez, M.E.; López, P.; Duato, J. Deterministic versus adaptive routing in fat-trees. In Proceedings of the 2007 IEEE International Parallel and Distributed Processing Symposium, Long Beach, CA, USA, 26–30 March 2007; pp. 1–8.
114. Dou, Z.Y.; Tu, Z.; Wang, X.; Wang, L.; Shi, S.; Zhang, T. Dynamic layer aggregation for neural machine translation with routing-by-agreement. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 17 July 2019; Volume 33, pp. 86–93.
115. Hinton, G. How to represent part-whole hierarchies in a neural network. *Neural Comput.* **2022**, *7*, 1–40. [CrossRef]
116. Wu, H.; Mao, J.; Sun, W.; Zheng, B.; Zhang, H.; Chen, Z.; Wang, W. Probabilistic robust route recovery with spatio-temporal dynamics. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1915–1924.
117. Osama, M.; Wang, X.-Z. Localized routing in capsule networks. *arXiv* **2020**, arXiv:2012.03012.
118. Choi, J.; Seo, H.; Im, S.; Kang, M. Attention routing between capsules. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27–28 October 2019.
119. Huang, L.; Wang, J.; Cai, D. Graph capsule network for object recognition. *IEEE Trans. Image Process.* **2021**, *30*, 1948–1961.
120. Dombetzki, L.A. An Overview over Capsule Networks. Network Architectures and Services. 2018. Available online: https://www.net.in.tum.de/fileadmin/TUM/NET/NET-2018-11-1/NET-2018-11-1_12.pdf (accessed on 2 July 2023).

121. TajikTajik, M.N.; Rehman, A.U.; Khan, W.; Khan, B. Texture feature selection using GA for classification of human brain MRI scans. In Proceedings of the Advances in Swarm Intelligence: 7th International Conference, ICSI 2016, Bali, Indonesia, 25–30 June 2016; Proceedings, Part II 7. Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 233–244.
122. Ullah, H.; Haq, M.U.; Khattak, S.; Khan, G.Z.; Mahmood, Z. A robust face recognition method for occluded and low-resolution images. In Proceedings of the 2019 International Conference on Applied and Engineering Mathematics (ICAEM), Taxila, Pakistan, 27–29 August 2019; pp. 86–91.
123. Munawar, F.; Khan, U.; Shahzad, A.; Haq, M.U.; Mahmood, Z.; Khattak, S.; Khan, G.Z. An empirical study of image resolution and pose on automatic face recognition. In Proceedings of the 2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Islamabad, Pakistan, 8–12 January 2019; pp. 558–563.