INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE
ISSN: 2057-5688

Classification of chest X-ray images using capsule Network

Mr. Athukuri Prasad
Research Scholar, JJTU
mailsofprasad@gmail.com
Dept. of ECE

Dr. Alok Agarwal
Internal Guide
alokagarwal26aaa@gmail.com
Dept. of ECE

Dr.Ch. Rami Reddy
Co - Guide
crreddy229@gmail.com
Dept. of EEE

Abstract. One of the most important issues facing public health all around the world is TB, which is an infection brought on by a bacteria known as Mycobacterium tuberculosis. This research aims to facilitate and automate the prediction of tuberculosis by the TBCapsNet architecture using Capsule Networks and demonstrate that it is more accurate than traditional CNN architectures. The development of tuberculosis screening systems with high accuracy contributes to the improvement of medical solutions in general, but particularly in rural areas with limited medical resources.

Keywords :CapsNet, CNNs, dynamic routing, Tuberculosis.

1. Introduction

Tuberculosis is a common and serious respiratory infection that can be challenging to diagnose accurately from chest radiographs, especially in low income countries. The TBCapsNet model was developed to assist radiologists in the diagnosis of TB by automatically analysing chest radiographs and providing a prediction of whether or not the patient has TB. Capsule Networks or CapsNets and Convolutional Neural Networks (CNNs) are both deep learning models used for image processing tasks[1]. CapsNets, are a type of neural network architecture that are designed to overcome some of the limitations of traditional CNNs. In particular, CapsNets aim to address the issue of "invariance" in CNNs, which refers to their inability to handle variations in pose, orientation, or deformation of objects within an image. CapsNet was introduced by Sabour, Frosst, and Hinton in 2017 shown in figure 1[2]. CapsNet is designed to overcome the limitations of traditional neural networks in object recognition tasks, such as the inability to handle variations in the position, orientation, and size of objects.In CapsNet, the basic building blocks are called "capsules," which are groups of neurons that are responsible for detecting specific features of an object, such as its orientation, size, and color[3]. Each capsule outputs a vector that represents the probability of the presence of a specific feature.Capsules are organized into layers, and each layer consists of several capsules. The output of each capsule in a given layer is passed as input to all the capsules in the next layer. This allows CapsNet to capture the hierarchical relationships between features of an object.[4]

CapsNet also includes a mechanism called "dynamic routing," which allows the network to adjust the weights between capsules based on the agreement between their predictions[2]. This enables CapsNet to handle variations in object position, orientation, and size, and to provide more accurate predictions of object attributes.CapsNet is still an active area of research, and it has shown promising results in tasks such as object recognition, pose

ISSN: 2057-5688

estimation, and generative modelling.There are some significant differences between the two. CNNs use convolutional layers to extract features from images, followed by fully connected layers for classification. In contrast, capsule networks use a novel architecture that consists of capsule layers, which group together neurons to represent more complex features. One of the key advantages of capsule networks is their ability to capture spatial relationships between features, which is important for tasks like object recognition. Capsule networks also have the ability to handle viewpoint changes and deformation of objects, which can be challenging for traditional CNNs[5].

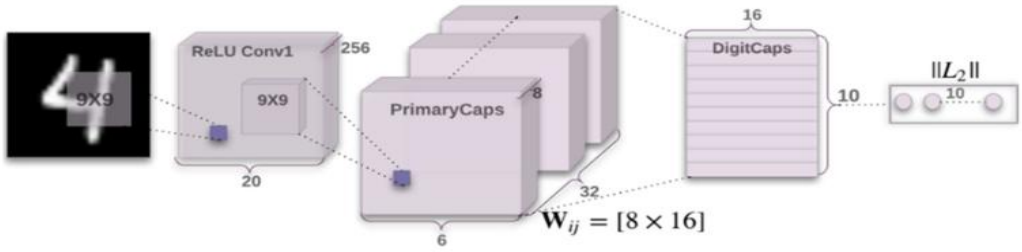

Figure 1 Original Capsule network architecture.

Another advantage of capsule networks is their ability to generate vectors that encode different attributes of objects, such as size, orientation, and position. This can be useful for tasks like image captioning, where the network needs to generate natural language descriptions of images. One potential disadvantage of capsule networks is that they require more computational resources and training time compared to CNNs[6]. Additionally, capsule networks are a relatively new architecture, so there are fewer pre-trained models and resources available compared to CNNs[7].

2. Architecture details of Caps Net

The core idea behind CapsNet is the use of "capsules" instead of traditional neurons. Capsules are groups of neurons that represent the instantiation parameters of an object, such as its pose, deformation, and texture. Each capsule outputs a vector that encodes these parameters, which is then routed to higher-level capsules to form a hierarchy of capsules representing increasingly complex features. The CapsNet architecture consists of two main parts, the encoder and the decoder. The encoder is responsible for extracting features from the input image and encoding them into capsules. The decoder takes the output of the encoder and reconstructs the input image. The loss between the input image and the reconstructed image is used to train the network.

Input
|
Convolutional Capsules
|
Primary Capsules
|
Secondary Capsule Layer

ISSN: 2057-5688

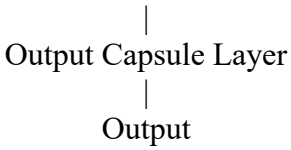

Figure 2 Architecture of caps Network

The architecture of a CapsNet typically consists of several layers of capsules, organized into two main types: "convolutional capsules" and "capsule layers". The figure 2 shows an example architecture of a CapsNet with two convolutional capsule layers and two fully connected capsule layers.Input layer takes in the input image, which is typically a 3-dimensional tensor representing the RGB channels of the image.Convolutional capsule layers perform convolutional operations on the input, similar to the way that CNNs work[8]. However, instead of producing scalar outputs like in a traditional CNN, each convolutional capsule outputs a vector that represents a particular entity or feature in the input.Primary capsule layer takes the output of the convolutional capsule layers and groups them into capsules based on their similarity. Each primary capsule represents a particular type of entity or feature in the input, and its output vector encodes information about the presence and pose of that entity.Secondary capsule layer takes the output of the primary capsule layer and computes the "coupling coefficients" between pairs of capsules. The coupling coefficients indicate how likely it is that two capsules are related to each other, based on their similarity and other factors. The digit capsules layer is where the Capsule Network identifies the object in the image. Each capsule in the digit capsules layer represents a possible object in the image. The capsules in the digit capsules layer are connected to the capsules in the primary capsules layer. The connection between the capsules in the primary capsules layer and the digit capsules layer is determined by a weight matrix, which is learned during training.

Output capsule layer uses the coupling coefficients to compute the final output vectors for each capsule, which represent the probability that a particular entity or feature is present in the input. The capsule with the highest output probability is taken as the final output of the network. The architecture of a CapsNet allows it to capture more fine-grained information about the entities and features present in an input, and to handle variations in pose, orientation, and other factors that can be challenging for traditional CNNs.Each capsule in the Capsule Network is represented by a vector. The length of the vector represents the probability of the object, and the orientation of the vector represents the pose of the object. The Capsule Network uses a dynamic routing algorithm to update the weight matrix during training, which allows the network to learn the relationship between the features and the objects in the image.

One of the key features of CapsNet is the dynamic routing algorithm used to combine the outputs of capsules at different levels of the hierarchy[9]. In traditional CNNs, the outputs of neurons are simply summed or averaged, whereas in CapsNet, the outputs of capsules are combined using a routing mechanism that takes into account the agreement between the predicted and actual output of higher-level capsules. This routing mechanism

ISSN: 2057-5688

allows CapsNet to capture the relationships between different parts of an object and to handle variations in object appearance. A key component of a capsule network is the squashing function, which is applied to each vector output by a capsule. The squashing function is designed to "squash" the length of the vector output to a value between 0 and 1, while preserving its orientation and direction. The purpose of this function is to ensure that the output of each capsule represents a probability distribution over the possible properties of the entity it represents.The specific form of the squashing function used in capsule networks is the "vector norm" function, which is defined as

$$\|v\| = \sqrt{\sum v_i^2} \qquad (1)$$

where v is a vector of length n, and v_i is the i th element of the vector. The output of the squashing function is

$$s = \frac{\|v\|^2}{(1+\|v\|^2)} \frac{v}{\|v\|} \qquad (2)$$

Where s is the output vector of length n, and the denominator in the equation ensures that the output vector has length between 0 and 1. The squashing function ensures that the output of each capsule represents a probability distribution over the possible properties of the entity it represents, and allows the capsule network to learn relationships between entities in an image, such as the relative positions and orientations of objects and parts.

Several variants of CapsNet have been proposed since its introduction, including recursive CapsNets, which use a recursive routing algorithm to allow capsules to communicate with each other across different levels of the hierarchy, and Capsule Networks with Convolutional Layers (CapsuleNet-CL), which combine CapsNet with traditional convolutional layers to improve performance on image classification tasks.CapsNet has shown promising results on several benchmark datasets, including MNIST, CIFAR-10, and SVHN. However, there are still some challenges associated with CapsNet, such as the need for large amounts of training data and the difficulty in optimizing the dynamic routing algorithm.

3. Other Applications of CapsNet

Capsule GAN [10] is a variation of the Generative Adversarial Network (GAN) framework[11] that incorporates Capsule Networks (CapsNets) in the generator architecture. The use of Capsule Networks in the generator architecture allows the model to capture hierarchical relationships and more nuanced information about the generated images. This can potentially result in more detailed and coherent image synthesis.

ChxCapsNet[12] is a deep learning model that uses capsule networks and transfer learning to evaluate pneumonia in pediatric chest radiographs. The ChxCapsNet model uses a deep capsule network, which is a type of neural network architecture that is designed to model hierarchical relationships between features in an image. Capsule networks are

ISSN: 2057-5688

particularly good at detecting complex spatial relationships between objects in an image, which makes them well-suited for image classification tasks. In addition to the capsule network, the ChxCapsNet model also incorporates transfer learning, which involves using pre-trained models to improve the performance of the network. Specifically, the model uses a pre-trained convolutional neural network (CNN) as a feature extractor to extract features from the input chest radiographs, which are then fed into the capsule network for classification[13]. The ChxCapsNet model was trained and tested on a large dataset of paediatric chest radiographs with and without pneumonia, achieving a high accuracy in diagnosing pneumonia in these images. The model has the potential to be a valuable tool for radiologists in assisting with the diagnosis of pneumonia in paediatric patients, which could improve patient outcomes and reduce healthcare costs.

4. Proposed TBcapsNet

Chest X-ray images are gathered for the TB data set by Shenzhen No.3 Hospital in Shenzhen, Guangdong providence,China[14]. The dataset contains images in JPEG format. There are 326 normal x-raysand 336 abnormal x-rays showing various manifestations of tuberculosis. Prepare the collected data for training by performing various preprocessing steps such as resizing, normalization, and augmentation. Data augmentation techniques like rotation, flipping, and adding noise can help increase the model's ability to generalize.TBcapsnet was designed for Tb detection using CapsNet architecture as shown in figure 3. Initialize the TBcapsnet model using random weights or pre-trained weights from a similar task, such as ImageNet. Pre-trained weights can help speed up the training process and improve performance, especially if the initial dataset is limited.Train TBcapsnet on the prepared dataset.

During training, the model learns to extract relevant features from the TB images and classify them as either TB-positive or TB-negative. This involves feeding the images through the network, calculating the loss, and updating the model's weights using an optimization algorithm like stochastic gradient descent (SGD) or Adam.CapsNets utilize a specific loss function called the "margin loss" or "spread loss" to train the network. The margin loss is designed to encourage a separation between the activations of different classes within the network.The margin loss in CapsNets is typically defined in equation 3.

$$L = T_c \max(0, m^+ - \|V_c\|)^2 + \lambda * (1 - T_c) \max\left(0, \left|\left|v_c\right|\right| - m^-\right)^2 \quad (3)$$

Where L represents the margin loss, T_c is equal to 1 if the class c is the target class and 0 otherwise. $m^+$is a constant parameter, typically set to 0.9, which controls the lower bound on the margin. $||v_c||$ represents the length (norm) of the output vector V_c for class c. m^- is a constant parameter, typically set to 0.1, which controls the upper bound on the margin. λ is a regularization parameter that can be adjusted to control the influence of the second termThe margin loss encourages the length of the output vector corresponding to the target class to be larger than m^+, while keeping the length of non-target class vectors below m^-. The margin loss penalizes any deviations from this desired margin.

ISSN: 2057-5688

Experiment with different hyperparameters, such as learning rate, batch size, and regularization techniques, to find the best configuration for your model. This process typically involves running multiple training iterations with different hyperparameter settings and evaluating the model's performance on a validation set. Training and loss curves of TBcapsnet during training is shown figure 4. Once training is complete, evaluate the trained model on a separate test set that was not used during training or hyperparameter tuning. Measure metrics such as accuracy, precision, recall, and F1 score to assess the model's performance in TB detection.If the model's performance is not satisfactory, then perform further iterations of training, fine-tuning, or dataset augmentation to improve the model's accuracy and generalization[15].

```
model.summary()
```

```
Model: "capsule_network"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
Convolution_Layer (Conv2D)   multiple                  20992
_________________________________________________________________
PrimaryCapsule (Conv2D)      multiple                  5308672
_________________________________________________________________
dense (Dense)                multiple                  82432
_________________________________________________________________
dense_1 (Dense)              multiple                  525312
_________________________________________________________________
dense_2 (Dense)              multiple                  803600
=================================================================
Total params: 8,215,568
Trainable params: 8,215,568
Non-trainable params: 0
_________________________________________________________________
```

Figure 3: TBcapsNet model summery

ISSN: 2057-5688

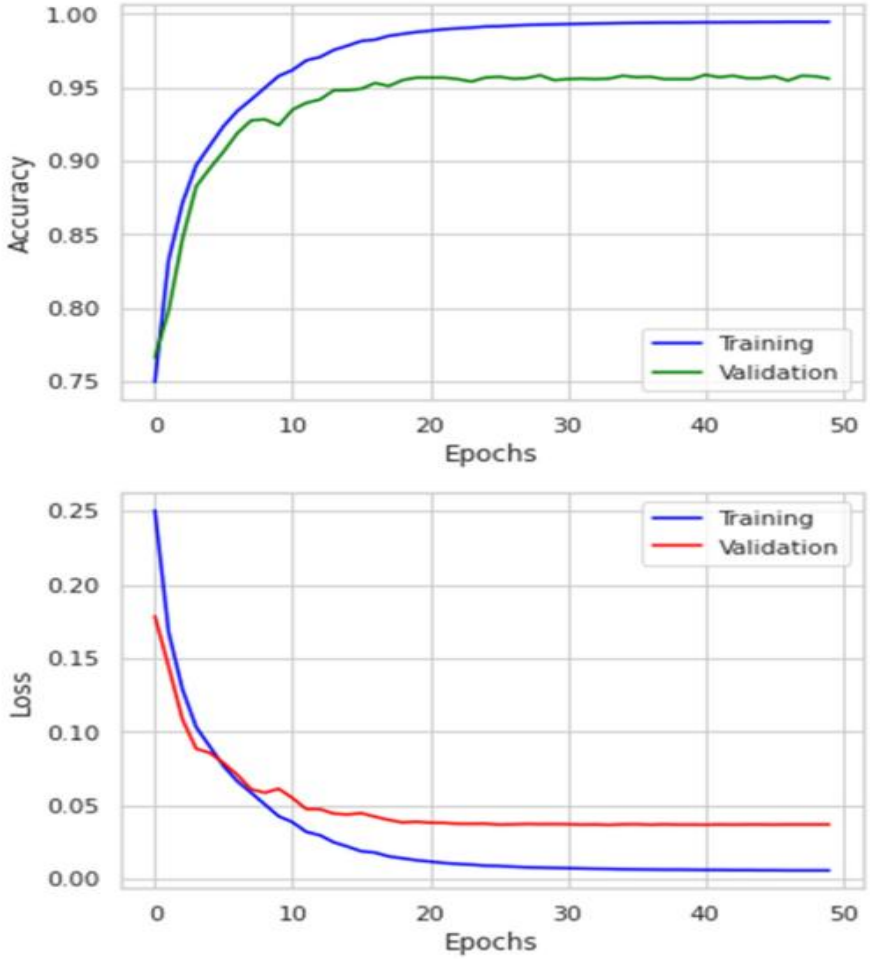

Figure 4: Training and loss curves of TBCapsNet

5. Conclusion

Capsule Networks are a promising approach to overcoming some of the limitations of traditional convolutional neural networks. By introducing a new type of neuron that models object pose and deformation, CapsNets are able to handle variations in the input data more effectively. While CapsNets are still a relatively new technology and have some limitations, they are an important area of research for the future of computer vision and deep learning.TBcapanet accuracy depending on the implementation details and available resources.

References

[1] M. Kim *et al.*, "Deep learning in medical imaging," *Neurospine*, vol. 16, no. 4, pp. 657–668, 2019, doi: 10.14245/ns.1938396.198.

[2] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *Adv. Neural Inf. Process. Syst.*, pp. 3857–3867, 2017.

[3] K. Armanious, Y. Mecky, S. Gatidis, and B. Yang, "Adversarial Inpainting of Medical Image Modalities," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2019-May, pp. 3267–3271, 2019, doi: 10.1109/ICASSP.2019.8682677.

[4] Y. Li, H. Su, and J. Zhu, "AdvCapsNet: To defense adversarial attacks based on Capsule networks," *J. Vis. Commun. Image Represent.*, vol. 75, no. January, p. 103037, 2021, doi: 10.1016/j.jvcir.2021.103037.

[5] A. M. Hafiz, S. A. Parah, and R. U. A. Bhat, "Attention mechanisms and deep learning for machine vision: A survey of the state of the art," pp. 0–24, 2021, [Online]. Available: http://arxiv.org/abs/2106.07550.

[6] V. Acharya *et al.*, "AI-Assisted Tuberculosis Detection and Classification from Chest X-Rays Using a Deep Learning Normalization-Free Network Model," *Comput. Intell. Neurosci.*, vol. 2022, 2022, doi: 10.1155/2022/2399428.

[7] M. Puttagunta and S. Ravi, "Medical image analysis based on deep learning approach," *Multimed. Tools Appl.*, vol. 80, no. 16, pp. 24365–24398, 2021, doi: 10.1007/s11042-021-10707-4.

[8] R. Hooda, A. Mittal, and S. Sofat, "Automated TB classification using ensemble of deep architectures," *Multimed. Tools Appl.*, vol. 78, no. 22, pp. 31515–31532, 2019, doi: 10.1007/s11042-019-07984-5.

[9] L. Jiao and J. Zhao, "A Survey on the New Generation of Deep Learning in Image Processing," *IEEE Access*, vol. 7, pp. 172231–172263, 2019, doi: 10.1109/ACCESS.2019.2956508.

[10] K. Marusaki and H. Watanabe, "Capsule GAN Using Capsule Network for Generator Architecture," 2020, [Online]. Available: http://arxiv.org/abs/2003.08047.

[11] M. Puttagunt, R. Subban, and C. Kennedy, Nelson Babu, "A Novel COVID-19 Detection Model Based on DCGAN and Deep Transfer Learning," *Procedia Comput. Sci.*, vol. 204, pp. 65–72, 2022, doi: 10.1016/j.procs.2022.08.008.

[12] J. D. Bodapati and V. N. Rohith, "ChxCapsNet: Deep capsule network with transfer learning for evaluating pneumonia in paediatric chest radiographs," *Measurement*, vol. 188, p. 110491, 2022, doi: https://doi.org/10.1016/j.measurement.2021.110491.

[13] S. Rajaraman and S. K. Antani, "Modality-Specific Deep Learning Model Ensembles Toward Improving TB Detection in Chest Radiographs," *IEEE Access*, vol. 8, pp. 27318–27326, 2020, doi: 10.1109/ACCESS.2020.2971257.

[14] A. Koeslag and G. de Jager, "Computer Aided Diagnosis of Miliary Tuberculosis," *Proc. Pattern Recognit. Assoc. South Africa*, no. September, 2001.

[15] S. Rajaraman and S. K. Antani, "Modality-Specific Deep Learning Model Ensembles Toward Improving TB Detection in Chest Radiographs," *IEEE Access*, vol. 8, pp. 27318–27326, 2020, doi: 10.1109/ACCESS.2020.2971257.

INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE
ISSN: 2057-5688