AUTOMATED MEDICAL DIAGNOSTICS USING DENSE CAPSULE NETWORKS

***Robert Langenderfer, *Ezzatollah Salari, **Jared Oluoch**

**Dept. of Electrical Engineering and Computer Science,*
*** Dept. of Engineering Technology,*
University of Toledo,
Toledo, Ohio, USA

ABSTRACT

Deep learning neural network techniques have been widely applied to medical radiological imaging diagnostics and have been proven to exceed the accuracy rates of highly trained radiologists. Numerous deep learning architectures have been applied to this task. Specifically, Convolutional Neural Networks (CNNs) have been widely applied and have demonstrated excellent performance characteristics. More recently, Capsule Networks (Capsnet) have managed to improve on the already stellar performance of CNNs by overcoming some of their weaknesses, such as pose and image transformation sensitivity. However, modifications to the Capsule Network can improve performance characteristics even further. The originally proposed Capsule Network utilizes convolutional techniques typical of CNNs in the initial layers of the network. By replacing the convolutional layers with dense layers, the Dense Capsule Network (DCNET) preserves more of the spatial image information which improves the accuracy of the network. In this paper the DCNET is applied to the problem of diagnosing pneumonia in ChestX-ray14, which is a publicly available chest X-ray dataset, consisting of over 100,000 radiological chest images capturing 14 distinct diseases. Diagnostic performance of DCNET is compared to that of CNNs, Capsnet, as well as that of expert radiologists.

Keywords: *Automated Medical Diagnostics, Capsule Neural Network, DCNET, Pneumonia, X-Ray, CNN, Radiology*

INTRODUCTION

Machine learning has been shown to exceed the diagnostic accuracy of trained medical personnel over many medical domains. Data from medical diagnostic tools such as X-ray, MRI, CT, and EKG have trained deep learning algorithms, and have been shown to effectively detect and diagnose a wide range of diseases[1][2][3][4]. In this paper we will focus on the application of machine learning to the classification of pneumonia in pediatric X-ray images.

Viruses, bacteria, chemicals, or fungi can all trigger the human biologic inflammatory response to an infection which is diagnosed as pneumonia. Diffusion capacity drops and in turn the blood oxygenation levels decrease due to fluids or cells displacing the gas capacity of lung alveoli. This increase in alveolar fluids can be detected by chest X-rays due to the increased lung sectional density. In 2019, more than 2.5 million people died from pneumonia and is the primary cause of mortality in young juveniles globally. Pneumonia is more common in developing countries suffering from a deficiency of medical facilities and increased environmental pollution[5]. Developed countries don't escape the high occurrence and costs of this disease, since pneumonia is the most common reason for disease related hospitalization in the US.

Pneumonia diagnosis requires a trained radiologist to visually interpret an image from an upper chest X-ray. Availability of trained radiologists is an issue, especially developing countries or even in remote areas of the US. The World Health Organization (WHO) has estimated that there is a global shortage of 4.3 million health professionals[6]. Also, one study revealed that 72 percent of pneumonia patients were misdiagnosed with pneumonia following a hospital re-admission[7]. In this paper we will explore machine learning techniques applied to chest X-ray images to automate pneumonia diagnosis, which could serve as a proxy for trained personnel in remote areas, or as a tool to improve diagnostic accuracy.

Current research has concentrated heavily on convolutional neural networks (CNNs) and deep learning neural networks (DNNs) to perform diagnostic classifications[8]. Improved networks such as Capsnet, exceed the impressive performance of older CNN networks. Techniques to increase the accuracy of Capsnet will be explored herein.

CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks have multiple layers which are composed of two basic types: convolutional and pooling. Convolutional layers implement 'kernels': small square matrices that are convoluted over an image used to enhance edges, perform blurring, sharpening, embossing, etc. Using backpropagation, the network learns the optimal kernel parameters to detect the features in the image data[9]. The pooling layer reduces each kernel output single value which decreases the data content layer to layer. Unfortunately, this data reduction discards important information contained in the image. It has been shown that pooling causes features in an image to lose important spatial relationships.

As a result of this data loss, CNNs are sensitive to simple 2D affine transformations such as scale, translation, rotation, reflection, and skew. There for a small shift or size variation in an image could result in a misclassification. These networks are trained on augmented data in an attempt to prevail over this issue. Unfortunately, augmentation can dramatically increase the effective training dataset size and therefore causes a corresponding increase in the training time.

To overcome these drawbacks of CNNs, we introduce the Capsule Neural Network. The Capsule Neural Network is an innovative approach to deep learning neural networks recently introduced by Geoffrey Hinton, who is considered one of the "Godfathers of Deep Learning."

CAPSULE NEURAL NETWORKS

There are two basic parts to the Capsule neural networks (Capsnet): the encoder and the decoder.

A. *The Encoder*

The first two layers of the encoder are composed of two convolutional layers which are identical to those in a CNN. Where Capsule networks differ significant from CNNs is that layers use vectors instead of scalar values. These vectors are weighted according to the following formula:

$$\hat{u}_{ji} = W_{ij} u_i$$

Vector u_i is the output of capsule *i* which is multiplied by the weight matrix W_{ij} to create the vector W_{ij}. This vector is the output for the next higher *j* level capsule. Capsule networks implement a connection method known as 'routing by agreement.' Capsules in each layer try to predict the output of the next higher layer. The better the accuracy of the prediction, or correlation, the more the capsules are connected through the 'coupling coefficient' c_{ij}. The output vector $\hat{u}_{ji}$ is multiplied by this coupling coefficient. This formula is as follows:

$$s_j = \sum_i c_{ij}\, \hat{u}_{ji}$$

where s_j is the input of capsule j to the next higher layer, and c_{ij} is obtained using the following softmax formula:

$$softmax(b_{ij}) = c_{ij} = \frac{e^{b_{ij}}}{\sum_k e^{b_{ik}}}$$

here variable b_{ij} is a scalar reflecting the strength of the connection between capsules *i* and *j*. Parameter b_{ij} starts out at zero, meaning there is no effective connection between the layers. The softmax function enhances the largest values and suppresses values which are significantly below the maximum value, while scaling the vector such that the outputs add up to 1.

Classic neural networks implement non-linear activation function such as ReLu or the sigmoid function, but in order to operate on vectors instead, the Capsule network implements a squashing function:

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

where v_j is the final output of capsule j. This formula normalizes the vector components to the range of zero to one[10]. This function produces a curve that resembles the upper half of the sigmoid function when viewed from 2 dimensions. The probabilities b_{ij} from the *softmax*(b_{ij}) function must be updated using the dot product of v_j and $\hat{u}_{ji}$.

$$b_{ij} = b_{ij} + v_j \cdot \hat{u}_{ji}$$

This equation is the key to routing by agreement because the greater the alignment of the two vectors the more the coupling coefficient is increased since it is based on b_{ij}.

The encoder portion of the network uses the following margin loss function

$$L_k = T_k \max(0, m^+ - ||v_k||)^2 + \lambda(1 - T_k)\max(0, ||v_k|| - m^-)^2$$

The hyperparameters λ, m^+, and m^- are assigned the values 0.5, 0.1, and 0.9 respectively. Now when a category of class k is present, then $T_k = 1$. The total loss is $\sum_k L_k$ which is the sum of the loss function for to each of the capsules *k*.

The number of capsules in the last layer of the encoder match the number of categories in the dataset. In this case, our dataset has two output categories: normal and pneumonia. Therefore, there are two last layer capsules in the network. These two capsules then connect to the encoder section of the Capsule network.

Table 1. Pseudo-Code for Dynamic Routing

Dynamic routing function

```
Routing( $\hat{u}_{ji}$, $r$, $l$)
    foreach capsule i in layer $l$
        foreach capsule j in layer $(l + 1)$
            $b_{ij} \leftarrow 0$
foreach r
    foreach capsule i in layer l
        $c_i \leftarrow softmax(b_{ij})$
```

```
foreach capsule j in layer (l + 1)
    $s_j = \sum_i c_{ij}\, \hat{u}_{ji}$
     foreach capsule j in layer (l + 1)
       $v_j \leftarrow squash(s_j)$
 foreach capsule i in layer l
    foreach capsule j in layer (l + 1)
      $b_{ij} \leftarrow b_{ij} + v_j \cdot \hat{u}_{ji}$
return $v_j$
```

B. *The Decoder*

The decoder is composed of fully connected network layers whose final output matches the size of the input layer. The encoder recreates the input images and trains the network using backpropagation the use of a loss function that is simply the N-dimensional Euclidian distance between the decoded image and the input image. This image generation technique is referred to as inverse graphics.

DENSE CAPSULE NEURAL NETWORKS

The Capsule neural network is a significant deviation from the popular CNN models, but the Dense capsule neural network deviates even further. The initial convolutional layers in the Capsule network are replaced with Dense networks instead. Dense networks add forward feeding dense connections between layers. The addition of layer-bypassing dense connections reduces the number of parameters and helps with the diminishing gradient issue. The diminishing gradient occurs when backpropagation error is 'diluted' by the large number of layers in deep neural network architectures, causing weight adjustments to stop or slow dramatically[11]. These bypass layers are illustrated in Fig. 1.

Given that the topology bypasses one or more layers, this allows high- and low-level feature granularities to be passed into the capsule layers which enhances the ability of the network to recognize features multiple levels of hierarchy. This pairs nicely with the ability of Capsule networks to feed more information through the network layers

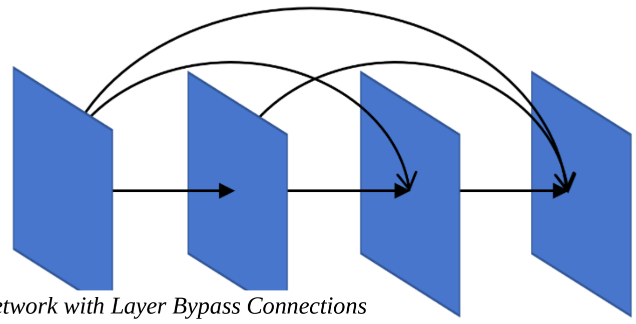

Figure 1: Dense Network with Layer Bypass Connections

Assume that the dense network is composed of *L* layers, each implementing a non-linear transformation H_L. This non-linear function can be a composite function of operations such as ReLU, or sigmoid. Traditional feed-forward networks connect the output of the L^{th} layer as input to the $(L+1)^{th}$ layer. Instead, in a dense connection scheme, each layer is connected to all subsequent layers. Therefore, each L^{th} layer receives the feature-maps of all preceding layers $x_0,...,x_{L-1}$, as inputs. The formula:

$$\mathbf{x}_L = H_L([\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_{L-1}])$$

where $[\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_{L-1}]$, refers to the concatenation of the feature maps from layers 0 through $L-1$, and $\mathbf{x}_L$ is the output of later L. To simplify implementation, these multiple inputs are concatenated into a single tensor[12]. As mentioned above, this dense layer is substituted for the CNN layers in the original Capsule network architecture. For this implementation a 3-layer dense network architecture was deployed.

DATASET

The described networks were trained and tested with the open source pneumonia MNIST dataset, which is a component of the MedMNIST data[13]. This grayscale dataset is based on a 5,856 pediatric chest X-Ray images which have a range of sizes: (384x127) – (2,916x2,713). These 256 level grayscale source images have been cropped and resized to 28 by 28 pixels. The dataset consists of 5,332 training images and 524 test images. These images were classified by radiologists into normal or pneumonia categories. The training set contains 1,835 normal and 3,497 pneumonia examples, while the test set consists of 135 normal and 389 pneumonia samples.

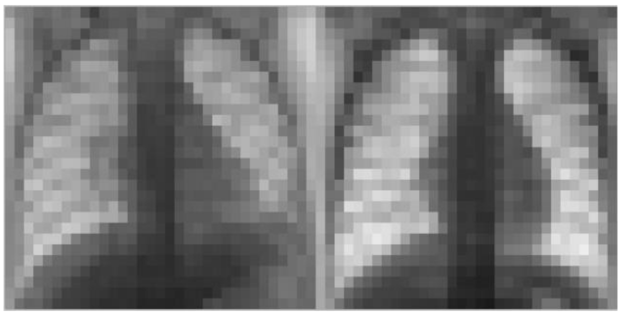

Figure 2: 28 x 28 X-Ray Images
Pneumonia (left) Normal (right)

IMPLEMENTATION

Three open-source libraries: Tensorflow, Keras, and Sklearn, were used with the Python language to implement the Dense Neural Network. GPU acceleration was enabled on Kaggle where all training and tested iterations were performed.

There was no data augmentation performed on this dataset. After training, the output of the decoder section recreates examples of the input images. Generally, these recreations lack the clarity and sharpness of the original images, but this is an excellent way to visually review the progress and results of the network training.

RESULTS

The performance of the network was evaluated using the sensitivity, accuracy, and area under the curve (AUC) metrics. Sensitivity and accuracy are calculated using the following formulas:

$$sensitivity = \frac{TP}{TP + FN}\ 100\%$$

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} 100\%$$

Where TP is the number of true positives, TN the number of true negatives, and FN is the number of false negatives, and FP the false positives. These values and their percentages are listed in the confusion matrix of Figure 2. The AUC, accuracy, sensitivity results are listed in Table 2.

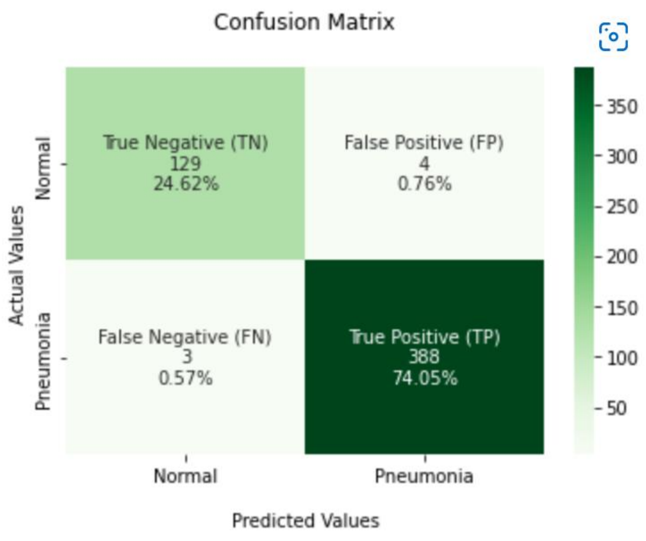

Figure 2: Confusion Matrix for Dense Capsule Network Results

Table 2: Performance Metrics

Metric		
AUC	**Accuracy**	**Sensitivity**
0.997	98.7%	99.2%

AUC is calculated by determining the receiver operating curve (ROC) which is the area under the curved plot of the true positive rate (TPR) against the false positive rate (FPR) over the range of threshold settings. The ROC curve is depicted in Figure 3.

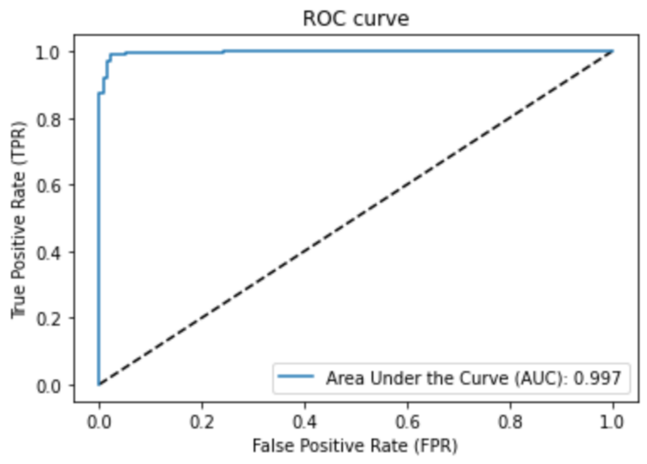

Figure 3: Receiver Operating Curve

Comparative results are listed in Table 3. The AUC and accuracy numbers are compared to several of the popular CNN models, such as ResNet and Google AutoML Vision, as shown in Table 3. The proposed Dense Capsule network method outperformed the accuracy of Google AutoML Vision, by 4.1%, and the original Capsule Network by 0.4%.

Table 3: Comparison of Model Performance

Comparison of Performance		
Network	**AUC**	**Accuracy**
ResNet-18 (28)	0.944	85.4%
ResNet-18 (224)	0.956	86.4%
ResNet-50 (28)	0.948	85.4%
ResNet-50 (224)	0.962	88.4%
auto-sklearn	0.942	85.5%
AutoKeras	0.947	87.8%
Google AutoML Vision	0.991	94.6%
Capsule Network	0.996	98.3%
Dense Capsule Network	**0.997**	**98.7%**

CONCLUSION

Capsule networks have been demonstrated to be very accurate when applied to the classification of pneumonia in low-resolution x-ray images. The pneumonia MNIST database, which is composed of 5,856 grayscale pediatric radiology images, was utilized for training and testing of the network architecture. The performance of the original Capsule network can be improved by replacing the convolutional layers with Dense network layers and demonstrated a classification accuracy of 98.7%. These results exceed the accuracy of the average radiologist and show that Dense capsule networks could be used as a tool to verify human diagnosis, and potentially reducing healthcare costs.

REFERENCES

[1] G. Marques, D. Agarwal, and I. de la Torre Díez, "Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network," *Appl. Soft Comput.*, vol. 96, p. 106691, Nov. 2020, doi: 10.1016/J.ASOC.2020.106691.

[2] K. R. Kruthika, Rajeswari, and H. D. Maheshappa, "Alzheimer's Disease Neuroimaging Initiative. CBIR system using capsule networks and 3D CNN for Alzheimer's disease diagnosis," *Inform. Med. Unlocked.*, vol. 14, pp. 59–68, Jan. 2019, doi: 10.1016/j.imu.2018.12.001.

[3] K. J. Cios, K. Chen, and R. A. Langenderfer, "Use of Neural Networks in Detecting Cardiac Diseases from Echocardiographic Images," *IEEE Eng. Med. Biol. Mag.*, vol. 9, no. 3, pp. 58–60, 1990, doi: 10.1109/51.59215.

[4] T. Neela and S. Namburu, "ECG signal classification using capsule neural networks," *IET Networks*, vol. 10, no. 3, pp. 103–109, May 2021, doi: 10.1049/NTW2.12018.

[5] C. L. Fischer Walker, I. Rudan, L. Liu,, "Global burden of childhood pneumonia and diarrhoea," *Lancet*, vol. 381, no. 9875, pp. 1405–1416, 2013, doi: 10.1016/S0140-6736(13)60222-6.

[6] A. T. Society, "Top 20 Pneumonia Facts — 201 9," 2016. .

[7] H. F. H. System, "Pneumonia often misdiagnosed on patient readmissions, studies find -- ScienceDaily." https://www.sciencedaily.com/releases/2010/10/101022123749.htm (accessed Apr. 22, 2022).

[8] P. Rajpurkar *et al.*, "CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning," Nov. 2017, Accessed: May 20, 2022. [Online]. Available: http://arxiv.org/abs/1711.05225.

[9] "CVPR 2021 Open Access Repository." https://openaccess.thecvf.com/content/CVPR2021/html/Gu_Capsule_Network_Is_Not_More_Robust_Than_Convolutional_Network_CVPR_2021_paper.html (accessed Mar. 27, 2022).

[10] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic Routing Between Capsules," *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, pp. 3857–3867, Oct. 2017, doi:

10.48550/arxiv.1710.09829.

[11] S. Samarth, R. Phaye, A. Sikka, A. Dhall, and D. Bathula, “Dense and Diverse Capsule Networks: Making the Capsules Learn Better,” May 2018, doi: 10.48550/arxiv.1805.04001.

[12] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 2261–2269, Aug. 2016, doi: 10.48550/arxiv.1608.06993.

[13] J. Yang, “MedMNIST v2: A Large-Scale Lightweight Benchmark for 2D and 3D Biomedical Image Classification,” Oct. 2021, doi: 10.48550/arxiv.2110.14795.