March 22, 2022

Exploring Entropy pruning coupled with Capsule Neural Network(CapsNet) for leaf disease classification.

Jennifer Jepkoech[1]*, David Muchangi Mugo[2] Edna Chebet Too[3]

[1,2]Department of Computing and Information Technology,University of Embu, PO BOX 6 – 60100, Embu, Kenya.

[3]Department of Computer Science and ICT,Chuka University, PO BOX 109-60400, Chuka, Kenya.

corresponding author: jepkoech.jennifer@embuni.ac.ke, PO BOX 6 EMBU.
echebet@chuka.ac.ke

Abstract

In an attempt to detect plant leaf diseases faster, accurately, and more efficiently, researchers have adopted deep learning methods using models like CNN and CapsNet with some success. However, there is a need for improvement as the current models are computationally expensive in terms of time and model complexity. Our work improves the Capsule Neural Network(CapsNet) model by adding convolutional layers to collect enough data and prune using the Entropy-based method. Entropy-based pruning reduces the number of characters in the model by getting rid of useless features and retaining only the useful features for detection. This considerably reduces the parameter size and reduces model complexities in terms of time and computation. We used F1, F5, and varying folds to assess accuracy in pruned mode against normal models. We tested our idea using 9080 images of tomatoes from PlantVillage and on three models, namely; ResNet-50, VGG-16, and CapsNet. CapsNet was the best among the pruned models with 98.9% followed by ResNet-50 with 93% and VGG-16 at 89.99%. We observed that pruning might be a Superior and less computationally expensive method than VGG-16 and ResNet-50. This implies that the accuracy of such models can be improved through the introduction of pruning.

Keywords: CapsNet; CNN; ResNet; Convolution; Pruning; Complexity; Computation.

1 Introduction

Deep neural networks have advanced rapidly in the field of computer vision over the last few years, from basic image classification tasks like the ImageNet recognition challenge [1],[2] , and [3] to more advanced applications like object detection [4], semantic segmentation [5], image captioning [6], and many others. In an attempt to solve misclassification problems, many researchers have used deep learning models for disease detection. This has achieved some success. However, researchers are still modifying the developed models to make them more accurate by reducing complexities. CapsNets by Sabour et al. [7] have shown great improvement in detection and classification due to their ability to encode features in vector form and preserve their pose. The work done by Sabour et al. [7] used two convolutional layers and a routing algorithm and achieved some success. However, only two convolutional layers are insufficient to collect as many features as possible for better classification. In an attempt to increase the features for better detection, Toranam et al. [8] increased the number of convolutional layers but then used Max-pooling to reduce the number of features. Research done by [8] improved the normal CapsNet architecture and used it to examine 231 covid-19 images. The improved model was tested on multi-class (COVID-19, and No-findings and Pneumonia) and binary class (COVID-19, and No-findings). The study added convolutional layers to the CapsNet model to enable maximum feature collection. The model in [8] achieved some promising results. However, an increase in convolutional layers leads to increased features and translates to model parameters. This may lead to complexities in the model. To reduce the parameters in the model, the work done by [8]

used Max pooling to reduce the parameters in the[1]ir model.

Although the results were promising, pooling is not recommended as it leads to significant data loss, which may affect the model's accuracy. Pruning can be adopted for CNNs and CapsNets to resize deep learning models which are used for disease detection and/or classification. In this work, we present an entropy-based approach for pruning various filters that do not contribute significantly to the model's accuracy to accelerate and compress CapsNet models in both training and testing. To recover discrimination capacity, we fine-tuned the trimmed. We used a learning schedule to seek a trade-off between training speed and model accuracy because the naıve iterative pruning technique is time-consuming. We test our pruning framework for image classification using CapsNet, ResNet-50 ResNet-50 [9], and VGG-16 [10]. We used 9080 tomato images from plantVillage dataset spread across ten tomato classes. The contribution of this work is to avoid Max pooling which leads to loss of data, increase the number of convolution layers for maximum feature collection, and use entropy-based pruning to reduce the model complexities in terms of computation and time.

2 Related work

Pruning Pruning, when compared to older approaches based on manually constructed visual characteristics, deep neural networks have attained state-of-the-art performance in various disciplines. Deep neural networks have a significant computational and storage overhead, which is a barrier for a mobile device with limited processing resources [10]. The VGG-16 model [11], for example, contains 138.34 million parameters, requires more than 500MB of storage space and requires 15.5 billion float point operations (FLOPs) to classify a single picture. A model of this complexity can easily exceed the computational capacity of the tiniest devices, such as cellphones. As a result, network compression has piqued the interest of both academics and industry. Many researchers have discovered that the deep model suffers greatly from over-parameterization. Denil et al. [12] demonstrated, for example, that a network can be efficiently reconstructed using only a subset of its original parameters. It should be noted that this redundancy appears to be necessary during the model training stage, as the highly non-convex optimization problem is difficult to solve with the current technique [13], [14]. As a result, the majority of compression strategies aim to reduce a pre-trained model to a small-scale model. In most deep learning models, the parameters in each layer generate a dense and huge matrix which causes processing and storage issues. If this dense matrix with low-rank small-scale matrices can be approximated, the matrix-vector multiplication may be completed rapidly using an approach such as the fast Fourier transform (FFT). As a result, both computational complexity and memory footprint may be drastically reduced. Inspired by this notion, many approaches have been presented for the approximation of weights. Sindhwani et al. [15] model the original parameter matrix using a linear combination of various structural matrices. Some mathematical operations can be used to turn these structured matrices into very low-rank matrices. Matrix factorization is used by Denton et al. [13] to investigate the linear structure of neural networks. They build the approximation using singular value decomposition.

Product quantization [16] is a popular approach that decomposes the space into a Cartesian product of low-dimensional subspaces and quantizes each subspace separately. Gong et al. [17] evaluated product quantization approaches and discovered that even with a simple k-means-based method, their approach could obtain excellent results. Chen et al. [18] proposed Hashed-Nets after introducing the hashing approach into model compression. All connections shared the same parameter value mapped to the same hash bucket. The Deep Compression technique [19] employs a similar concept.

Network pruning is a well-known issue in model compression that has been extensively researched for many years. Pruning was previously used in the early 1900s to limit the number of connections and prevent over-fitting [20],[21]. Han et al. [19] have suggested a pruning strategy for removing redundancy from deep models. Small-weight connections that fall below a certain threshold would be removed, resulting in a minimalist design. However, their solution did not lower the size of the activation tensor, which dominates the memory footprint when batch size is big. As a result, some academics concentrate their efforts on filter pruning to minimize the number of channels in the activation tensor.

Ferentinos [22] evaluated five Convolutional Neural Network (CNN) architectures for detecting plant illnesses using a dataset of 87,848 leaf pictures of 25 plant species, separated into 58 plant-disease pairings, attaining an overall success rate of 99.53 percent with the VGG design. Images from both the laboratory and the field were used. When photos from the laboratory were utilized for training and real-life images for testing, the results were significantly reduced. Kothari et al. [23] used two well-

known CNN models (AlexNet and GoogLeNet) on the PlantVillage [24] dataset for disease diagnosis, with varied train-test ratios, on colored, grayscale, and segmented pictures, attaining a maximum accuracy of 99.35 percent. Their goal was to create an easy-to-use mobile-friendly application that could be delivered to farmers. Similarly, Zhang et al. [25] evaluated AlexNet, GooLeNet, and ResNet with different hyperparameters (training method, batch size, iterations, and so on) to get the maximum accuracy of 97.28 percent in diagnosing Tomato leaf diseases. The identification and classification of nine tomato plant diseases using CNN as a learning algorithm has been successfully implemented with 99.18 percent accuracy [26]. Lu et al. [27]used photos of healthy and sick leaves taken naturally to obtain an overall accuracy of 95.48 percent with a CNN-based model for identifying 10 illnesses in rice crop. Fuentes et al. [28] created a deep learning-based detection algorithm for tomato disease and pest identification. The photographs were acquired in real time using digital cameras that caught features such as backdrop, sunshine, size, and so on. Since their inception, capsule networks have been used to classify brain tumors, forecast traffic flow, generate 3D images, classify cervical images, test for lung cancer, and recognize emotions [29]. However, there is a scarcity of literature on their use to the diagnosis of plant diseases in agriculture. Xiao et al. [30] used capsule networks to classify peanut leaf diseases. They have also been used to identify UAV pictures [31] and to classify hyperspectral images [32].Figure 1 shows tomato images drawn from PlantVillage database affected by different diseases.

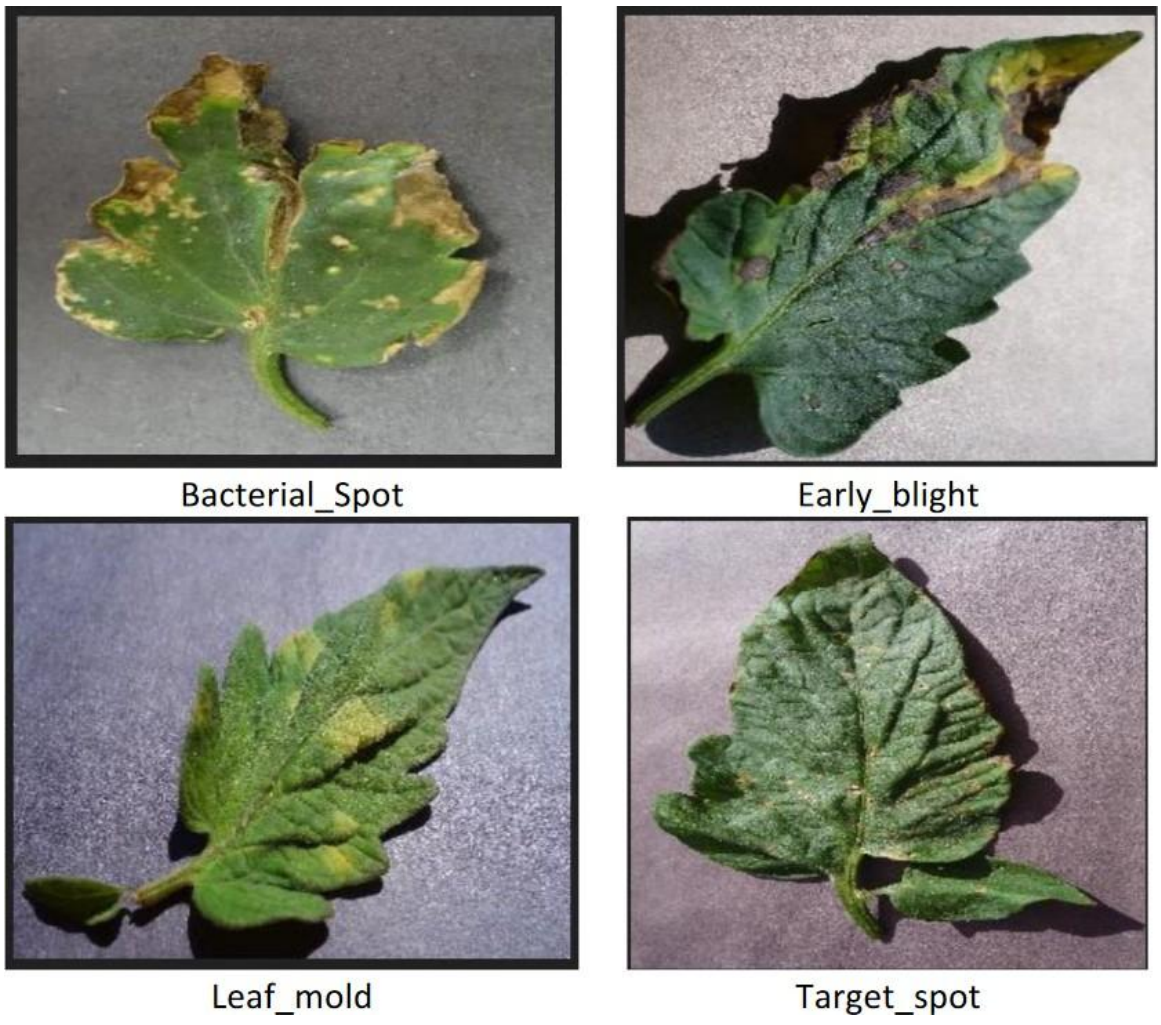

Figure 1: Tomato images from Plant-Village

In an attempt to avoid misclassifications, artificial intelligence through machine learning has come up with automated methods and algorithms for accurately interrogating x-ray images. In the medical field deep learning has been successfully used for breast cancer diagnosis [33], [34], lung segmentation

[35], [36], pneumonia [37], [38], [39], [40], [41], [42], and brain injuries among others. Research done by Hemdan et al. [43] developed a model for diagnosing covid-19 and obtained a percentage accuracy of 74.29%. The work considered a recall, precision, and f1-scores in their accuracy determination. Even though the accuracy was not up to a maximum of 100%, it was better than eye assessment by specialists. Narin et al. [2] developed the ResNet50, InceptionV3, and InceptionResNetv2 deep learning models to diagnose COVID-19 using chest X-ray images. The binary classification procedure was used in the study, and the results were verified using five-fold cross-validation. The models' performance was tested using five distinct criteria, and the ResNet50 model attained an average of 98% accuracy. To diagnose COVID-19 instances, Afshar et al. [44]

used capsule networks. accuracy scores, Specificity, and sensitivity were used to assess the performance of the suggested model, and 98.3 percent accuracy was obtained. Mobiny et al. [45] used X-ray images to create capsule networks that could diagnose COVID-19 patients. The suggested technique performed better when the developed model was compared to Inceptionv3, ResNet50, and DenseNet121. In the recent past, covid-19 diagnosis has been done using radiology images. Apostolopoulos et al.[46], devel- oped a method to diagnose Covid-19 using deep learning methods. The work used 500 no-findings X-rays, 700 bacterial pneumonia, and 224 covid-19 images, and the researchers used specificity values, sensitivity, and accuracy to gauge the model's performance. In general, the model achieved 93.48% COVID-19 vs. No-findings vs. pneumonia, 98.78% for covid-19 vs. No-findings. Wang et al. [47] developed a deep learning algorithm for detecting COVID-19 illness. The suggested method's performance was tested using sensitivity and accuracy measures, and the results were compared to deep learning models VGG19 and ResNet50. At the end of the trial, the average accuracy was 93.3 percent. Sethy et al.[48] used deep learning to extract features from COVID-19 X-ray images and SVM to classify these characteristics. Kappa values and f1-scores were used to assess the effectiveness of this hybrid model, which was built by mixing ResNet50 and SVM models. The study observed good performance as compared to other models. Zheng et al.[49] developed a new deep learning model and tested it on 499 CT images. They observed an accuracy of 88.55%, measured using AUC, f1-scores precision, and recall. CT scans were applied in research done by Ying et al. [50] for identification of COVID-19 disease and to differentiate it from pneumonia. The developed deep learning model was compared to several models in the literature and found to be better. The availability of deep learning models has enabled easy access to large datasets. CNN-based models are the most often used deep learning models in medicine, as they are in many other domains. CNN de- signs, however, have inherent limits. Max-pooling is one of these constraints. Max-pooling is a method of resizing the model, and sometimes in the process, the most useful features are lost. Furthermore, current CNN models are unable to maintain the objects' pose. To address CNN's weaknesses, Sabour et al. [51] presented a novel neural network dubbed Capsule Networks in 2017. The researchers observed that the developed model overcame the challenges associated with Convolutional Neural Network (CNN). Normal CNN's use Scalar input activation functions like Sigmoid, Tangent, and ReLU. Capsule networks have been intelligently developed to preserve the pose and attributes of items in an image and simulate their hierarchical connections [51]. CNN's are not abler to preserve pose, and they also lose data through pooling [7]. Capsules can enclose pose and other features in vector form, allowing vector routings to represent the image parameters [51].

3 Materials and Methods

3.1 Capsule Neural Networks (CapsNets)

If the value of the lower capsule is j, then its corresponding output is $\hat{u}_j|i$ the prediction of the higher level capsule i, is calculated as shown in (1) :

$$\hat{u}_{j|i} = w_{ij} u_{j|i} \tag{1}$$

w_{ij} represents the weighting matrix, and $u_{j|i}$ is the vector that i uses to predict the j th capsule in the model. Each capsule is expected to predict the higher capsule. If the prediction agrees to the output of the higher-level capsules, there will be an increase in the coupling coefficients[7]. Equation (2) shows the SoftMax function used for the calculation of coupling coefficients[7].

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \tag{2}$$

The coupling coefficient c_{ij} is encompassed by variables in equation (2), the log probabilities b_{ij} which is set to zero at the beginning of the routing algorithm. To determine if the lower-level capsule i can be coupled with the higher-level capsule j, the log probabilities are used. Equation (3) shows how the input vector to higher capsules is calculated .

$$c_{ij} = \hat{u}_{j|i} \cdot v_j + b_j \text{ where; } b = 0 \tag{3}$$

Here,the output of$j$ is v_j. The length of the output vector represents the probability of existence of a vector. Therefore, long vectors are increased to a bear one value while short vectors are reduced to an almost zero value using equation (4);

$$v_j = \frac{(||s_j||^2)s_j}{1 + ||s_j|| ||s_j||} \tag{4}$$

where v_j the vector output of capsule j and s_j is its total input while b_{ij} is updated during routing v_j and $u_{j|i}$. The inner product will be large if any two vectors agree[7].

The agreement a_{ij} updating between log probabilities b_{ij} and the coupling coefficients c_{ij} is evaluated as:

$$a_{ij} = v_j \cdot \hat{u}_{j|i}. \tag{5}$$

Equations (2) and (5) are used to describe the whole routing procedure for computing high-level vectors. Vectors are able to enclose pose and the relationship between the entities of an image [7]a technique that allows capsules to learn various features within an image easily. l_k is the loss function used to provide equity of values between zero and one. Equation (6) is used to calculate the loss function.

$$l_k = T_k \max(0, m^+ - ||v_k||)^2 + \lambda(1 - T_k) \max(0, ||v_k|| - m^-)^2 \tag{6}$$

if the digit of k is available, T_k will be equal to one. Otherwise it will be equal to zero.

3.2 Our Approach

3.2.1 Data

We test our pruning framework for image classification using CapsNet, ResNet-50 [9], and VGG-16 [10]. We used 9080 tomato images from plantVillage database. Our work resized the images to 28 x 28 to create equity of size in all the images.

3.2.2 Pruning

In our work, we use an entropy-based pruning to evaluate the relevance of each filter.We convert tensors into vectors using global average pooling . We choose global average pooling [52] because it acts as a regularizer because it summarizes spacial information using average values. Our work has used $(L_i, W_i,$ Conv)to represent convolution in layer i. $L_i \in R^{c*h*w}$ represents an input tensor while $W_i \in R^{d*c*k*k}$ is a set filter weights. The main objective is to get rid of useless filters W_i. In this work each filter represents one channel of its activation vector. If a channel of the activation tensor contains less information, its related filter is less significant, and hence will be eliminated during filter pruning. We evaluated our filters using the entropy method whereby when entropy values are calculated, they are arranged in ascending order. A higher entropy number indicates that the corresponding filter has more information and is thus not pruned. A low entropy value shows that the feature is useless and is pruned.We used equation (7) to calculate the entropy value:

$$H_j = \sum_{i=1}^{n} p_i \log p_i \tag{7}$$

Where the probability bin i is denoted by p_i,the entropy channel of j is denoted by H_j. a small value of H_j means that the channel is not significant and hence will be removed from the model.We convert tensors into vectors using global average pooling. We have used blue to illustrate the useful feature maps and red to show the useless models removed during pruning. After pruning, we did fine tuning to stabilize the model.

3.2.3 The proposed framework

We present a novel CapsNet developed using four convolutional layers to classify 28 $\times$ 28 images. The reason for increasing the convolution layers is to better classify through many feature maps. After each convolution, this work introduces pruning to remove useless feature maps. We also use fine-tuning to stabilize the model for better performance. Figure 2 shows the pictorial representation of the proposed architecture on pruning.

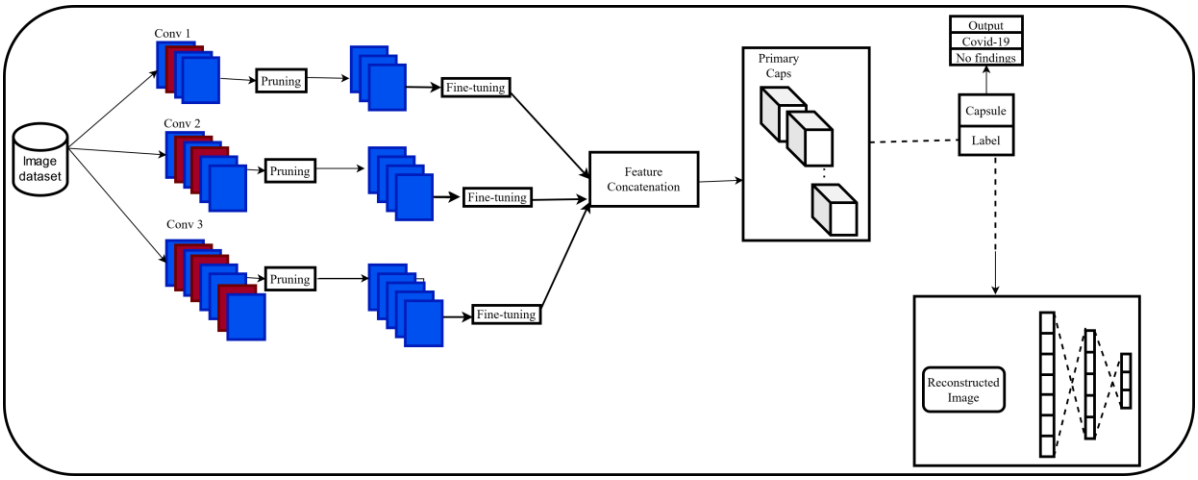

Figure 2: Proposed architecture on pruning.

Our approach used 16 kernels of size 5 $\times$ 5 with a stride of 1 in the first layer. We then applied pruning to remove the useless feature maps. We maintained the same pattern for the second and third layers containing 32 5 $\times$ 5 and 64 9 $\times$ 9, layers respectively. Our primary Caps layer contains 9 $\times$ 9 kernels with a stride of 1 applied to 32 capsules. The Label Caps layer has 16-dimensional (16D) capsules for two classes while the ReLU activation function is used for all layers. We evaluated our model performance using 9 fold cross-validation where we divided our data into ratios of 10%:90%, 20%:80%, 30%:70%, 40%:60%, 50%:50%, 60%:40%, 70%:30%, 80%:20%, 90%:10% for testing and training respectively. Further, we used True Positive (TP), True Negative(TN), False Negative(FN), and False Positive(FP) to compare the performance.

$$Sen = \frac{TP}{(TP + FN)} * 100 \tag{8}$$

$$Spe = \frac{TN}{(TN + FP)} * 100 \tag{9}$$

$$Pre = \frac{TP}{(TP + FP)} \tag{10}$$

$$F1 = \frac{2TP}{(2TP + FP + FN)} * 100 \tag{11}$$

4 Results and Discussion

Regarding Top-1 accuracies, it was observed that VGG-16 and ResNet-50 displayed higher percentages when measured without pruning. This may seem better but according to [7] CNNs lose relevant data due to pooling. Hence the two models would still not be the best bet for accurate classification. Top-1 accuracy for a pruned CapsNet on the other hand was higher than that of the original model.

This shows that a pruned CapsNet would be good for detection and classification because it encodes spatial information[7] about a feature. Additionally, we eliminated nonsignificant feature maps, hence reducing computational and time complexities. There was an observed increment between Top-1 and Top-5 accuracy. The pruned CapsNet model was observed to be having the highest accuracy

Table 1: Performance changes of models before and after Pruning.

		TOP -1	TOP-5
VGG	Original	72.41%	91.32%
	Pruned	70.0%	89.99%
ResNet	Original	73.10%	94.45%
	Pruned	72.43%	93.33
CapsNet	Original	77.1%	95.7%
	Pruned	78.52%	98.9%

as compared to both VGG-16 and ResNet-50. This indicates that pruning could go a long way in improving accuracy in image processing. Table: 1 shows performance changes of models before and after pruning.

Table: 2 shows classification per fold results without pruning while table: 3 shows classification per fold results with pruning. Our observation was that after pruning, the Models' accuracy increased considerably. On comparing the folds, when the train was at 80% and testing at 20%, the % F1 was high in both pruned and unpruned models which confirms the 80%:20% rule acknowledged by [53]in their work. In general the results observed show that pruning can be considered as one of the best methods that reduce model complexity and increase efficiency without the loss of meaningful data

Table 2: Classification per fold results without pruning.

Train: Test	Sen (%)	Spe (%)	Pre (%)	F1 (%)	ACC (%)
10:90	96.69	99.54	99.51	98.08	98.11
20:80	99.44	97.59	97.64	98.56	98.54
30:70	99.49	94.73	94.99	97.19	97.11
40:60	98.49	95.63	95.77	97.11	97.06
50:50	96.39	97.34	97.31	96.85	96.86
60:40	98.34	98.34	98.34	98.34	98.34
70:30	99.00	97.09	97.13	98.05	98.04
80:20	97.54	98.49	98.47	98.00	98.01
90:10	93.73	98.49	98.40	96.01	96.11

Table 3: Classification per fold results with pruning

Test: Train (%)	Sen (%)	Spe (%)	Pre (%)	F1 (%)	Acc (%)
10:90	97.19	99.95	99.92	98.49	98.52
20:80	99.94	98.09	98.14	99.06	99.04
30:70	99.54	94.78	95.04	97.24	97.16
40:60	98.54	95.68	95.82	97.16	97.15
50:50	99.44	97.39	97.36	97.82	97.82
60:40	98.39	98.39	98.39	98.39	98.39
70:30	99.05	97.04	97.08	98.00	97.99
80:20	97.59	98.64	98.62	99.09	99.10
90:10	93.78	98.54	98.45	99.84	99.94

5 Conclusion

We explored entropy pruning in this work and discovered that it improved the accuracy levels from 95.7% to 98.9% and reduced complexities. Our pruned model displayed a higher performance compared to its original version. Our method can be adopted for any deep learning application because it doesn't rely on a particular library. In future we intend to to explore the effect of dynamic pruning in CapsNets.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[5] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[6] J. Johnson, A. Karpathy, and L. Fei-Fei, "Densecap: Fully convolutional localization networks for dense captioning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4565–4574, 2016.

[7] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *Advances in neural information processing systems*, vol. 30, 2017.

[8] S. Toraman, T. B. Alakus, and I. Turkoglu, "Convolutional capsnet: A novel artificial neural network approach to detect covid-19 disease from x-ray images using capsule networks," *Chaos, Solitons & Fractals*, vol. 140, p. 110122, 2020.

[9] K. S. Rajput, S. Wibowo, C. Hao, and M. Majmudar, "On arrhythmia detection by deep learning and multidimensional representation," *arXiv preprint arXiv:1904.00138*, 2019.

[10] A. Narin, C. Kaya, and Z. Pamuk, "Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks," *Pattern Analysis and Applications*, vol. 24, no. 3, pp. 1207–1220, 2021.

[11] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[12] M. Denil, B. Shakibi, L. Dinh, M. Ranzato, and N. De Freitas, "Predicting parameters in deep learning," *Advances in neural information processing systems*, vol. 26, 2013.

[13] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus, "Exploiting linear structure within convolutional networks for efficient evaluation," *Advances in neural information processing systems*, vol. 27, 2014.

[14] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[15] V. Sindhwani, T. Sainath, and S. Kumar, "Structured transforms for small-footprint deep learning," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[16] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 117–128, 2010.

[17] Y. Gong, L. Liu, M. Yang, and L. Bourdev, "Compressing deep convolutional networks using vector quantization," *arXiv preprint arXiv:1412.6115*, 2014.

[18] W. Chen, J. Wilson, S. Tyree, K. Weinberger, and Y. Chen, "Compressing neural networks with the hashing trick," in *International conference on machine learning*, pp. 2285–2294, PMLR, 2015.

[19] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," *Advances in neural information processing systems*, vol. 28, 2015.

[20] A. Levin, T. Leen, and J. Moody, "Fast pruning using principal components," *Advances in neural information processing systems*, vol. 6, 1993.

[21] B. Hassibi and D. Stork, "Second order derivatives for network pruning: Optimal brain surgeon," *Advances in neural information processing systems*, vol. 5, 1992.

[22] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and electronics in agriculture*, vol. 145, pp. 311–318, 2018.

[23] J. D. Kothari, "Plant disease identification using artificial intelligence: Machine learning approach," *Jubin Dipakkumar Kothari (2018). Plant Disease Identification using Artificial Intelligence: Machine Learning Approach. International Journal of Innovative Research in Computer and Communication Engineering*, vol. 7, no. 11, pp. 11082–11085, 2018.

[24] D. Hughes, M. Salath´e, *et al.*, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv preprint arXiv:1511.08060*, 2015.

[25] K. Zhang, Q. Wu, A. Liu, and X. Meng, "Can deep learning identify tomato leaf disease?," *Advances in multimedia*, vol. 2018, 2018.

[26] M. Brahimi, K. Boukhalfa, and A. Moussaoui, "Deep learning for tomato diseases: classification and symptoms visualization," *Applied Artificial Intelligence*, vol. 31, no. 4, pp. 299–315, 2017.

[27] G. Jaswal and R. C. Poonia, "Selection of optimized features for fusion of palm print and finger knuckle-based person authentication," *Expert Systems*, vol. 38, no. 1, p. e12523, 2021.

[28] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors*, vol. 17, no. 9, p. 2022, 2017.

[29] M. E. Paoletti, S. Moreno-Álvarez, and J. M. Haut, "Multiple attention-guided capsule networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[30] L. Xiao, H. Ouyang, C. Fan, T. Umer, R. C. Poonia, and S. Wan, "Gesture image segmentation with otsu's method based on noise adaptive angle threshold," *Multimedia Tools and Applications*, vol. 79, no. 47, pp. 35619–35640, 2020.

[31] Y. Li, M. Qian, P. Liu, Q. Cai, X. Li, J. Guo, H. Yan, F. Yu, K. Yuan, J. Yu, *et al.*, "The recognition of rice images by uav based on capsule network," *Cluster Computing*, vol. 22, no. 4, pp. 9515–9524, 2019.

[32] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. Plaza, J. Li, and F. Pla, "Capsule networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 4, pp. 2145–2160, 2018.

[33] L. Shen, L. R. Margolies, J. H. Rothstein, E. Fluder, R. McBride, and W. Sieh, "Deep learning to improve breast cancer detection on screening mammography," *Scientific reports*, vol. 9, no. 1, pp. 1–12, 2019.

[34] G. Hamed, M. A. E.-R. Marey, S. E.-S. Amin, and M. F. Tolba, "Deep learning in breast cancer detection and classification," in *The International Conference on Artificial Intelligence and Computer Vision*, pp. 322–333, Springer, 2020.

[35] B. A. Skourt, A. El Hassani, and A. Majda, "Lung ct image segmentation using deep neural networks," *Procedia Computer Science*, vol. 127, pp. 109–113, 2018.

[36] Y. Gordienko, P. Gang, J. Hui, W. Zeng, Y. Kochura, O. Alienin, O. Rokovyi, and S. Stirenko, "Deep learning with lung segmentation and bone shadow exclusion techniques for chest x-ray analysis of lung cancer," in *International conference on computer science, engineering and education applications*, pp. 638–647, Springer, 2018.

[37] A. K. Jaiswal, P. Tiwari, S. Kumar, D. Gupta, A. Khanna, and J. J. Rodrigues, "Identifying pneumonia in chest x-rays: a deep learning approach," *Measurement*, vol. 145, pp. 511–518, 2019.

[38] S. S. Yadav and S. M. Jadhav, "Deep convolutional neural network based medical image classification for disease diagnosis," *Journal of Big Data*, vol. 6, no. 1, pp. 1–18, 2019.

[39] I. M. Baltruschat, H. Nickisch, M. Grass, T. Knopp, and A. Saalbach, "Comparison of deep learning approaches for multi-label chest x-ray classification," *Scientific reports*, vol. 9, no. 1, pp. 1–10, 2019.

[40] R. H. Abiyev and M. K. S. Ma'aitaH, "Deep convolutional neural networks for chest diseases detection," *Journal of healthcare engineering*, vol. 2018, 2018.

[41] O. Stephen, M. Sain, U. J. Maduh, and D.-U. Jeong, "An efficient deep learning approach to pneumonia classification in healthcare," *Journal of healthcare engineering*, vol. 2019, 2019.

[42] V. Chouhan, S. K. Singh, A. Khamparia, D. Gupta, P. Tiwari, C. Moreira, R. Damaˇseviˇcius, and V. H. C. De Albuquerque, "A novel transfer learning based approach for pneumonia detection in chest x-ray images," *Applied Sciences*, vol. 10, no. 2, p. 559, 2020.

[43] F. Demir and B. Ta¸scı, "An effective and robust approach based on r-cnn+ lstm model and ncar feature selection for ophthalmological disease detection from fundus images," *Journal of Personalized Medicine*, vol. 11, no. 12, p. 1276, 2021.

[44] P. Afshar, S. Heidarian, F. Naderkhani, A. Oikonomou, K. N. Plataniotis, and A. Mohammadi, "Covid-caps: A capsule network-based framework for identification of covid-19 cases from x-ray images," *Pattern Recognition Letters*, vol. 138, pp. 638–643, 2020.

[45] A. Mobiny, P. A. Cicalese, S. Zare, P. Yuan, M. Abavisani, C. C. Wu, J. Ahuja, P. M. de Groot, and H. Van Nguyen, "Radiologist-level covid-19 detection using ct scans with detail-oriented capsule networks," *arXiv preprint arXiv:2004.07407*, 2020.

[46] I. D. Apostolopoulos and T. A. Mpesiana, "Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks," *Physical and engineering sciences in medicine*, vol. 43, no. 2, pp. 635–640, 2020.

[47] L. Wang, Z. Q. Lin, and A. Wong, "Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images," *Scientific Reports*, vol. 10, no. 1, pp. 1–12, 2020.

[48] P. K. Sethy and S. K. Behera, "Detection of coronavirus disease (covid-19) based on deep features," 2020.

[49] C. Zheng, X. Deng, Q. Fu, Q. Zhou, J. Feng, H. Ma, W. Liu, and X. Wang, "Deep learning-based detection for covid-19 from chest ct using weak label," *MedRxiv*, 2020.

[50] Y. Song, S. Zheng, L. Li, X. Zhang, X. Zhang, Z. Huang, J. Chen, R. Wang, H. Zhao, Y. Chong, *et al.*, "Deep learning enables accurate diagnosis of novel coronavirus (covid-19) with ct images," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 18, no. 6, pp. 2775–2780, 2021.

[51] L. A. Dombetzki, "An overview over capsule networks," *Network Architectures and Services*, 2018.

[52] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[53] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2097–2106, 2017.