International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 12, No.1 (Jul-2022)

https://dx.doi.org/10.12785/ijcds/120104

Overview of CapsNet Performance Evaluation Methods for Image Classification using a Dual Input Capsule Network as a Case Study

Patrick Kwabena Mensah[1] **and Mighty Abra Ayidzoe**[1,2]

[1]*Department of Computer Science and Informatics, University of Energy and Natural Resources, Sunyani, Ghana*
[2]*School of Information and Software Engineering,University of Electronic Science and Technology of China, Chengdu, P.R. China*

Received 26 Mar. 2021, Revised 10 Feb. 2022, Accepted 15 Jun. 2022, Published 1 Jul. 2022

Abstract: Performance evaluation is a critical part of deep learning (DL) that requires careful conduct to enhance confidence and reliability. Several metrics exist to evaluate DL models, however, choosing one for a given model is not trivial, since it is not a one-fit-all solution. Practically, accuracy is the most popularly used evaluation metric for capsule networks (CapsNets). This is problematic for sensitive applications (e.g. health), since accuracy is overly optimistic in the presence of class imbalance, and does not permit the exact reporting of a model's risk of bias and potential usefulness. This paper, therefore, aims at demonstrating the usefulness of other metrics for performance evaluation as well as interpretability through the implementation of a custom capsule model. The metrics are effective in measuring the real performance of the models in terms of accuracy (93.03% for proposed model), number of parameters ($\approx$ 4 million fewer for proposed model), ability to scale and fail-safe, and the effectiveness of the routing process when evaluated on the datasets. Evaluating a CapsNet model with all these metrics has the potential to enhance the practitioner's confidence and also improve model understandability and reliability.

Keywords: Capsule networks, Deep learning, Performance evaluation, COVID-19, Explainable artificial intelligence

1. Introduction

Capsule Networks (CapsNet) [1] are shown to have equivalence compared to the invariance in Convolutional Neural Networks (CNNs). They can additionally encode the texture, deformation, hue, albedo, etc. of an object making it suitable for a wide range of applications where the availability of large datasets is a problem. To evaluate their performance, most researchers adopt the confusion matrix-based measures (especially accuracy) [2] in the literature. However, accuracy is highly inappropriate as the only metric for evaluating a classifier; especially under class imbalance [3].

Since performance evaluation is a very important task in deep learning (DL), it must be rigorously carried out to ensure reliability and confidence. Reliability and confidence are important desirable properties required to earn the trust of industry players to adopt DL (e.g. CapsNet) models for critical applications such as health and agriculture. Since CapsNets are implemented as classification models, they are generally subject to most of the performance evaluation metrics used to evaluate DL supervised models. However, many of these metrics are not adopted in evaluating CapsNets models in the literature. The consequence is that critical decisions are taken based on confusion matrix-based measures which can easily be biased.

The dynamic routing (DR) algorithm [1] performs coupling that groups features into clusters. The performance of the routing algorithm must be evaluated in terms of cluster separability. This study implements two CapsNet models trained on COVID-19 [4], [5] and Plant Village datasets [6] to demonstrate and reinforce the need to re-evaluate CapsNet image classification models whose decisions are based on only confusion matrix-related performance measures. The intention, however, is not to present all performance evaluation metrics for classification algorithms as being suitable for CapsNets, but to provide some form of a guide to researchers to choose existing metrics that will enhance reliability, explainability, and confidence in the models. The methods suggested in this work, are experimentally demonstrated to be feasible in evaluating the performance of CapsNets as well as providing some form of understanding to components in the "black box". The adoption of all

E-mail address: patrick.mensah@uenr.edu.gh, mighty.ayidzoe@uenr.edu.gh

these methods should provide the rigor needed to achieve explainability, reliability, and the confidence required for practical adoption.

The objectives of this paper are to (1) evaluate the performance of CapsNet on the datasets, (2) examine the appropriateness of performance measures used in existing CapsNet image classification models in the literature and also establish the need for the use of more suitable performance metrics, (2) recommend evaluation methods suitable for CapsNets to ensure reliability, explainability, and understandability leading to improved confidence in model results, (3) experimentally demonstrate that the recommended methods are appropriate and accurate in reporting a model's risk of bias and potential usefulness.

The rest of the paper is organized as follows: Section 2 presents related works in the literature. In section 3, experimental settings and the methods adopted to demonstrate the feasibility of the metrics are presented. Section 4 presents a discussion of the experimental results and demonstrates the suitability of the methods suggested in this study. The study is concluded in Section 5 and the scope for future studies is presented.

2. Related Work

The volume, velocity and variety of data (e.g. health records) presents a challenge for humans to perform efficient cost-effective analysis. This calls for the adoption of machine learning algorithms, which on the other hand, face challenges in terms of feature extraction since they are heavily dependent on manual feature engineering. Unlike machine learning algorithms, DL algorithms are capable of automatic extraction of features [7], but are also faced with the unavailability of large amounts of annotated data and class-imbalance [8]. The search for possible solutions to these problems has heightened the interest of researchers to apply DL algorithms such as CapsNets to health care images [9], [10]. Studies such as the detection of protein biomarkers in saliva using multi-lane CapsNets [11], identification of COVID-19 cases using CapsNets [12], drug discovery [13], and other interesting related research can be found in the literature. Novel diseases such as COVID-19 are difficult to diagnose by practitioners due to its overlap with other lung diseases [12]. The ability of CapsNets to encode spatial information and also perform well on smaller and complex datasets makes it a natural candidate in solving this problem. As a consequence, capsule networks have shown good classification performance on smaller chest X-ray images [12] of COVID-19 and normal patients. The ability to train on smaller datasets makes the model useful in helping to combat the sudden and rapid rise in COVID-19 cases. The rapid separation of infected and non-infected images using CapsNet [14] is an important step to optimize resource allocation in healthcare and for the early prevention of infectious diseases. Operator unreliability and dependency in using ultrasound to diagnose rotator cuff lesions can be addressed through the use of CapsNet [15] models to improve the extraction of discriminative features. With the aid of VGG16, a CapsNet model can differentiate discriminative patches from whole slide pathological images [16] required to, for example, determine the duration for which a biological organism is expected to die. For early detection of retinal diseases, CapsNet segmentation can be employed to extract thin and overlapping vessels [17]. This operation is challenging to perform manually as it is prone to errors, and also time-consuming. Capsule networks have been used to classify medical images [18] and to determine human emotions [19] as they can affect people's cognition, behavior, and decision-making. Other health conditions such as schizophrenia, cancer, and malaria can be identified by CapsNets [20], [21] in a bid to improve the quality of life. The aforementioned CapsNet applications are mostly evaluated using accuracy (see Table I), which does not present a fair assessment of model performance in the presence of class imbalance. It is clear that a doctor won't implement a model's recommendation that is solely based on high accuracy. It is not sufficient to earn the trust of practitioners as they may not understand mechanisms leading to this (high) accuracy. A clear understanding of the model's ability to (1) fail safely (determined by ablation studies), (2) determine regions of interests (saliency maps) in the input images, (3) form separable clusters (measures the effectiveness of CapsNet algorithm), (4) properly reconstruct the input images, and (5) effectively extract the correct features (from feature map visualization) are paramount to the process of understanding a model's performance and its suitability for a given health application.

Another sensitive area that requires attention in the quest to improve human life is security. Just like health, CapsNets have found use in this field due to the easiness with which they automatically extract features, work with smaller imbalanced datasets, and their resilience to perturbation. CapsNet applications in this field also require rigorous evaluation to avoid using results of biased metrics as the basis for making critical decisions. A list of the evaluation methods adopted for these models is shown in Table I.

Advances in media technology have made available state-of-the-art tools to forge images and video in real-time [22]. Making the situation critical is the availability of social media networks where the raw and forged images can be obtained and posted without verification. Since social media contains thousands of images, it is near impossible for a human expert to identify forged images and videos. Also, artificial intelligence methods developed to combat this problem soon become obsolete as the attackers modify the mode of attacks. It is critical to find solutions to this problem since forged images and videos can be used to by-pass facial recognition-based authentication systems as well as being used to disseminate fake information. Consequently, Nguyen et al., [23] proposed the use of a capsule network to detect spoofs from printed images, videos, and replay attacks. The model was evaluated using the half total error rate **(HTER)** and accuracy, however, the

effect of the combination of recognition and anti-spoofing on performance was not analyzed as suggested in [24]. In an earlier work [25], the same authors proposed a CapsNet model to detect different attacks using images and fake videos that can be used to carry out malicious actions including privacy breaches, and security violations.

Food is a basic requirement for the healthy growth of every human being. It is obtained from plants that are plagued with pests and diseases. The consequences of not identifying and controlling plant diseases on time are malnutrition, poverty, insecurity, among others. However, manual plant disease recognition is complex, costly, time-consuming, and prone to errors. Researchers have, therefore, proposed the use of CapsNets to automatically recognize plant diseases to overcome these limitations.

Kurup et al. [26] proposed a capsule model to identify plant diseases as well as plant species. For the plant disease diagnoses, a total of 54,306 images constituted the dataset obtained from 14 different plants. The dataset is made up of 38 classes; 26 sick and 12 healthy images. An inspection of the dataset [6] shows that it is highly imbalanced. The class with the minimum number of samples ("healthy Potato") has 121 images while the class with the largest number of samples ("huanglongbing Orange") has 4,405 images. A training-validation split of 80%-20% was applied on an augmented dataset to train and validate the model. Confusion matrix-based performance measures (validation accuracy, precision, recall, F1 score, and Area Under the Curve (AUC)) were adopted to evaluate the performance of the model. The Area Under the Precision-Recall Curve (AU PRC) compared to accuracy is more appropriate for measuring the performance of the classifier [27] under class imbalance, but it was not indicated by Kurup et al. [26] whether the AUCs they obtained were for PR curves or the Receiver Operating Characteristic Curves (ROC).

Considering the nutritional and economic importance of potatoes, a CapsNet model was implemented to classify the diseases of the plant [28]. The model was trained and validated on the potato dataset in the Plant Village dataset. This dataset is small and imbalanced but was artificially balanced by ensuring that each of the three classes; "late blight", "early blight", and "healthy" each had 1000 images. Validation accuracy was the only metric used to measure the performance of the model even though an AUC of the ROC curve could have been appropriate as an additional performance measure. Again, inference was not carried out to determine whether the model can generalize well on unseen images.

A capsule network was proposed [29] for the recognition of peanut leaf diseases. The dataset was a custom dataset constructed to obtain five classes (diseases). The dataset was imbalanced with the largest class ("Brow Spot") containing 4,028 images and the smallest class ("Net Blotch") with 1,578 images. Data augmentation was carried out to increase the training set to 11,132 images and 2,600 images for validation constituting 8:2 respectively. Only validation accuracy was used as the performance measure in the presence of class imbalance.

Gabor capsules [32], [33] have been proposed for plant disease detection. The tomato dataset, a subset of the Plant Village dataset, and the Citrus dataset [34] were used to train the models. These datasets are imbalanced as well as being small. Notwithstanding, the models were evaluated with validation accuracy, precision, sensitivity, and specificity. With the imbalanced nature of the datasets, the AUC of the PR curve is better suited to measure the performance of the models. However, other important performance indicators such as the number of parameters and reconstruction were reported.

CapsNet models were proposed for the detection of banana leaf diseases [30] and also for the recognition of the diseases in the entire Plant Village dataset. The banana leaf disease model was trained with custom data made up of three classes each with 1,000 images. Validation and test accuracies were used to evaluate the performance of the model. In the other model, irrespective of the fact that the Plant Village dataset is highly imbalanced, the model's performance was evaluated with only validation accuracy. Existing literature shows that CapsNet's performance evaluation metrics that significantly contribute to model explainability are rare compared to simple validation accuracy. Unfortunately, a model is not useful if it attains high accuracy with no means to show that it is reliable, understandable, and trustworthy. This paper, therefore, recommends and demonstrates the adoption of evaluation measures that will enhance the reliability and explainability needed to gain the trust and confidence of industry players.

3. Methodology

The experiments in this study were carried out on a Keras (TensorFlow backend) on 64-bit Windows PC that has NVIDIA GeForce GTX 1060 GPU with CUDA 10.1 and an 8GB RAM. Each model was trained for 100 epochs with a learning rate of 0.001 and a learning rate decay to 0.9. The number of routing iterations was varied from 1 to 6 for each of the datasets and the margin loss function [1] adopted. The margin loss is set to $L_k = T_k max(0, m^+ - \|v_k\|)^2 + \lambda(1-T_k)max(0, \|v_k\| - m^-)^2$ where $T_k = \begin{cases} 1, & \textit{if class K is active} \\ 0, & \text{otherwise,} \end{cases}$ λ=0.5 , m^+=0.9, m^-=0.1 The patience or early stopping hyperparameter is set to 10 for training and the best model saved. This study builds upon the code at [35].

A. Datasets and data preprocessing

The models were trained with the tomato dataset of the Plant Village dataset consisting of 18,159, 256×256×3 images, and the COVID-19 dataset. The tomato dataset is grouped into nine classes of infected leaves plus one class of healthy leaves. The dataset is imbalanced as the

TABLE I. SAMPLE APPLICATIONS OF CAPSNETS AND THEIR PERFORMANCE EVALUATION METHODS. THE CONFUSION MATRIX-BASED METHODS ARE PREDOMINANT WHILE RECONSTRUCTION AND FEATURE VISUALIZATION ARE RARELY USED.

	Application	Input Data	Base Algorithm	Evaluation Method	Ref
Health	COVID-19 diagnosis	3D Chest X-ray images	CapsNet [1]	Accuracy, Specificity, Sensitivity, AUC	[12]
	COVID-19 diagnosis	COVID-CT-MD	UNET, CapsNet	Accuracy, Specificity, Sensitivity, AUC	[14]
	Classification of rotator cuff lesions	MRI shoulder images	CapsNet	Accuracy, Precision, Recall, F1-Score	[15]
	GBM and LUSC prediction	Whole slide pathological images	VGG16, CapsNet	Accuracy	[16]
	Medical image classififcation	PatchCamelyon (PCam) dataset	CapsNet	Accuracy, AUC	[18]
	EEG-based emotion recognition	DEAP and DREAMER datasets	CapsNet	Accuracy	[19]
	Schizophrenia identification	MRI and fMRI scans	CapsNet	Accuracy, Specificity, Sensitivity	[20]
	Malaria parasite detection	Microscopic red blood cell images	ConvNet, CapsNet	Confusion Matrix, Accuracy, Precision, Recall, F1-Score	[21]
Security	Detection of forged images/ videos	Deepfake dataset	VGG-19, CapsNet	Half total error rate, Accuracy	[23]
	Image & Video Attack detection	Deepfake, Face2Face, FaceForensics++, FaceSwap	CapsNet	Activation maps, Equal error rate, Half total error rate, Accuracy	[25]
	Banana leaf disease detection	Custom dataset	CapsNet	Accuracy	[30]
	Plant disease detection	Plant Village	CapsNet	Accuracy, Sensitivity, Specificity.	[31]

maximum per-class image size of 5,357 and 372 being the number of images in the smallest class. The COVID-19_Radiography_Dataset [5], [4] comprises of 16,933 (i.e. 80%) Chest X-ray images in the training set and 4,232 (i.e. 20%) images in the test set. Each set is made up of four classes, namely; COVID, Lung_Opacity, Viral Pneumonia, and Normal. The dataset is highly imbalanced with 269 images in the test set of Viral Pneumonia and 2,038 images in that of the Normal class. CIFAR10 [36] is a primary dataset made up of 32×32 images comprising of 50,000 and 10,000 training and test samples respectively. This dataset is complex due to the presence of varied backgrounds and background objects in the images. No pre-processing aside from resizing of the images to 28×28 is applied to both datasets during training. During inference, the Augmentor [37] augmentation library is used to introduce variability in the images belonging to the test sets of each dataset.

B. CapsNet Architecture

Two architectures are implemented in this work. The first model uses the architecture of the original CapsNet (referred to as dynamic routing - DR) while the second model is made up of a custom architecture (referred to as dual-input – DI). Details of DR's architecture can be found in [1] with Figure 1 depicting the architecture of the DI model. The input images are resized to 28×28×3 and supplied to the DI model whose architecture comprises 2 Conv layers used as feature extractors each having 125, 7×7 kernels with ReLU activation and a stride of 1. Each of these layers produces 125, 22×22 feature maps concatenated to produce 125, 44×22 feature maps. These are used as input to Conv3 which consists of 96, 7×7 kernels, also with ReLU activation at a stride of 1 to produce 96, 38×16 feature maps. The primary capsule (PC) layer is composed of 5×5 kernels at stride 2. It has 16, 8-dimensional component capsules, each of size 6×17 making up a total of 16*17*6 = 1632 capsules in the PC layer. The Recognition Caps form the secondary capsule with ten capsules (classes). Each PC capsule will couple (form a cluster) with a secondary capsule based on the agreement a_{ij} (similarity) between them. The decoder network has three fully connected (FC) layers with 512, 1024, and 28*28*3 = 2352 nodes respectively in the first, second, and third layers. The last layer of the decoder is used to reconstruct the input images.

4. Results and Discussion

A. Confusion Matrix

The multi-class confusion matrices depicted in Figure 2 summarizes the performance of the DI model on the plant disease dataset. It is observed that the model correctly identified the true classes indicated by the high TP values. Few images were misclassified. From this matrix, other performance measures such as Precision, Sensitivity, Specificity, and per-class accuracy can be derived. True positive (TP), true negative (TN), false positive (FP), and false-negative (FN) are also obtainable from the confusion matrix and are crucial for making decisions, even though the interpretation can be confusing and situation-dependent in some cases. For example, a classifier should obtain as many false positives (FP) as possible for the existence of a disease, since it is not fatal (in most cases) to classify a healthy person as sick. On the other hand, many FN predictions under these circumstances indicate a poor outcome by the model since it is disastrous to classify a sick person as healthy. Such situation-dependent interpretations are not common in CapsNet implementations that adopt these performance metrics in the literature. The average accuracy for a model is given in (1). The other metrics are obtained using (2) – (5).

$$accuracy = \frac{\sum\ of\ correct\ predictions}{total\ number\ of samples\ in\ test\ set} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Sensitivity/Recall = \frac{TP}{TP + FN} \tag{3}$$

$$Specificity = \frac{TN}{TN + FN} \tag{4}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

Aside from accuracy, these are powerful per-class met-

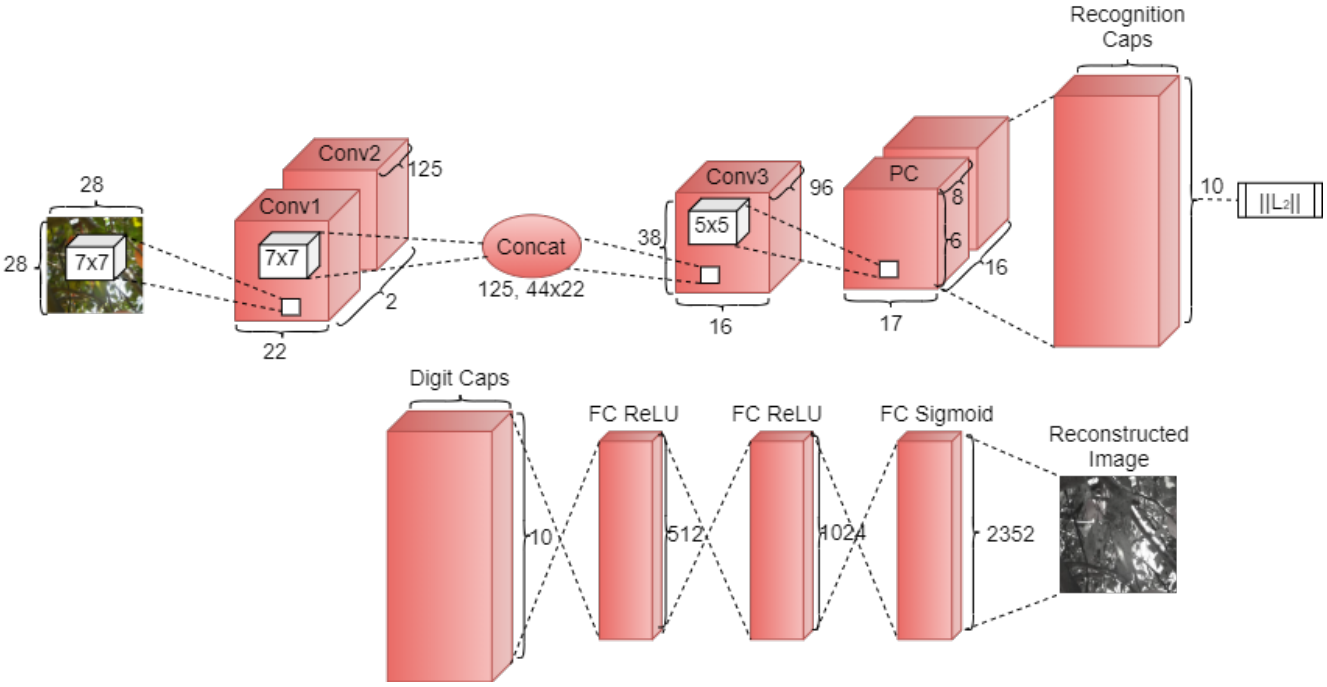


Figure 1. The architecture of the dual-input (DI) model. The architecture for the DR model is shown in Appendix A, Fig A3

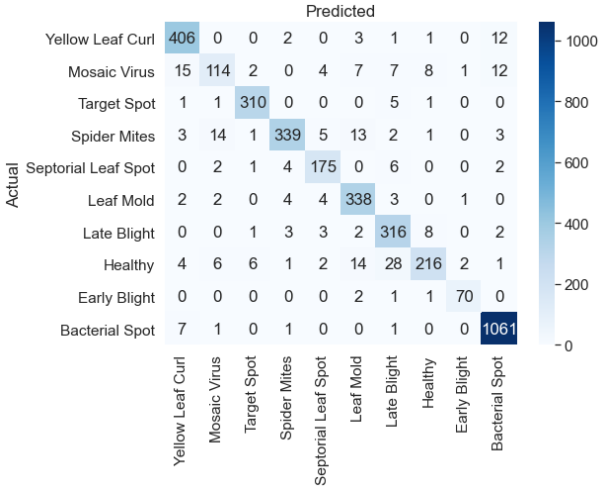


Figure 2. Confusion matrix for the DI model trained with the plant disease dataset. A host of other performance evaluation methods can be derived from the confusion matrix.

rics that are rarely used to evaluate capsule network models in practice. This paper, however, focuses on the metrics that can be used to further enhance the reliability and trust of the models.

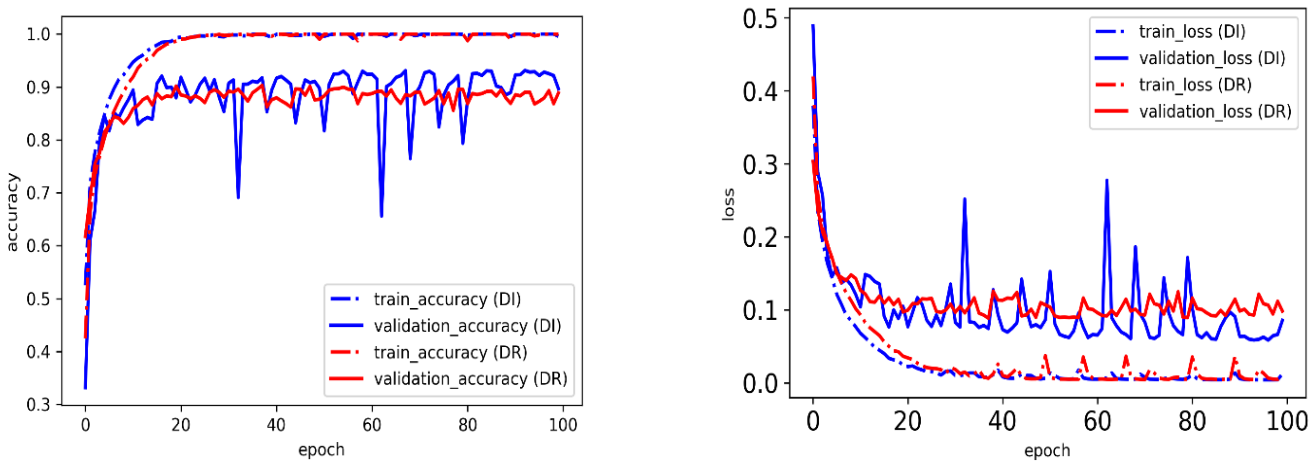
B. *Training and Validation Accuracy*

Some datasets are mostly small and highly imbalanced. Accuracy being the most predominant evaluation metric [38] for classification algorithms (despite its drawbacks) is very poor at differentiating one class from another under class imbalance [39]. It is, therefore, not sufficient to measure performance [40] since "accuracy paradox" [41] will cause the accuracy of the larger classes to overshadow those of the smaller classes. The result is an overall accuracy that is biased towards the accuracy of the larger class. Besides, it does not take into consideration asymmetric misclassification costs while at the same time ignoring the probability estimates of the classification that shows the confidence with which the predictions are made.

The difference in performance between the two models in terms of accuracy can be observed in Figure 3 and Table II. However, it does not indicate the extent to which a majority class influenced the total accuracy. It is therefore imperative to perform additional performance measurements to confirm which model is superior (DI vs DR for plant disease). In this light, this paper proposes that accuracy be calculated under different circumstances such as during training and validation (Figure 3), ablation studies (Table III), prediction or inference (Figure 6), and during scaling (Figure 4) as the number of routing iterations are varied to increase or reduce model capacity. Classification loss can be analyzed similarly since it is the price paid for inaccurate classification.

C. *Measuring Model's Ability to Scale*

The performance of the dynamic routing algorithm is largely dependent on the number of routing iterations [1] as the algorithm scales up or down. CapsNet models should, therefore, be evaluated by varying the number of routing iterations to evaluate the ability of the network to scale up or down without overfitting. Even though this method is



(a) (b)

Figure 3. Training and validation results for the two models on the plant disease dataset. (a) training and validation accuracies, and (b) training and validation losses.

TABLE II. MODEL ACCURACIES. THE FIRST TWO WERE TRAINED ON THE PLANT DISEASE DATASET.

Model	Accuracy (%)
Dual Input (DI)	93.03
Original CapsNet (DR)	90.55
DI_COVID-19	89.17
DI_CIFAR 10	76.58
DR_CIFAR 10	67.21

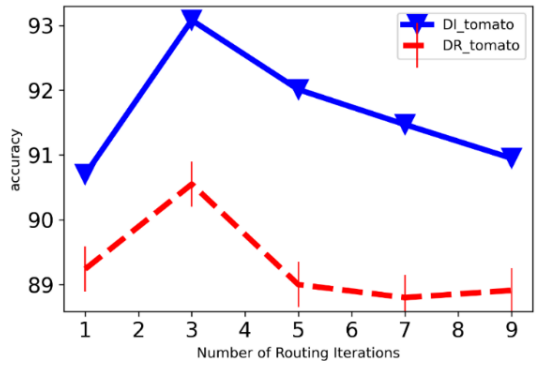


Figure 4. Accuracy against the number of routing iterations.

very relevant, it has been rarely adopted to evaluate existing capsule models. As shown in Figure 4, the network capacity increases from 1 to 9 as the models are evaluated on the dataset. The accuracy for both DR and DI peaks at 3 routing iterations and begins to drop. The implication is that, as the network capacity increases beyond 3 routing iterations, the model begins to overfit the training set. Further experiments conducted concerning this phenomenon show some consistency with the original CapsNet [1] which established that the optimal number of iterations on MNIST is 3. However, this number may vary for a given implementation [42], hence the need to search for the optimal number for which the model does not overfit the dataset. From Figure 4, it is observed that the method is capable of distinguishing (confirming) the superior model (DI) from the inferior one (DR). It goes further to show details of the exact iteration number at which the performance is optimal.

D. Measuring Model's ability to Fail-safe

This paper uses an ablation study [42] to measure the performance of a CapsNet model when certain components are missing or malfunctioning. It is also suggested as a means by which a model's ability to degrade gracefully is measured. Graceful degradation is a property required by CapsNets applied in critical applications to avoid failed or degraded components from grinding the network to a halt. Again, ablation serves as a basic step for explainability since it can identify the contributions and importance of each component in the network to the entire system performance. It also provides an opportunity to test the robustness of the model to architectural changes; the results of which may enhance confidence in the model. Furthermore, analyses of ablation results can uncover network components that can stand-in for damaged parts and contribute to the recovery of the network.

Table III shows the results of the ablation study carried on the DI model. It is seen that the Conv3 layer has a major impact on the performance of the model; probable because it samples features from the two lower-level layers enabling it to form a higher-level representation of the input images (see DI's Conv3 feature maps in Figure 8). Conv1 also showed a slight impact on network performance. The presence of these two Conv layers (see Table III, row 1) has a positive impact on performance. From rows 1, 4, and 5 in Table III, the failure of Conv2 may not be catastrophic to the performance of the model. All the ablation results

TABLE III. RESULTS OF ABLATION STUDY OF DI MODEL

No.	Conv1	Conv2	Conv3	Accuracy(%)
1	yes	no	yes	92.23
2	yes	yes	no	90.18
3	no	yes	yes	91.06
4	no	no	yes	90.45
5	yes	no	no	90.02
6	no	yes	no	89.87
7	yes	yes	yes	93.03

were obtained using 3 routing iterations.

E. Evaluating model's performance on smaller and imbalanced datasets

The area under the curve (AUC) is preferred to accuracy as it is invariant to apriori probability distributions of the classes and also independent of the decision threshold [2].

The degrees of consistency and discriminancy have been used [2] to show that AUC is a superior metric to accuracy. A classifier with a large AUC is preferred to that with a smaller AUC. AUC can be computed for both the Receiver Operating Characteristic Curve (ROC) and the Precision-Recall Curve (PR). It is recommended that the ROC curve be used to evaluate balanced datasets as it tends to be overly optimistic in cases where there is a large skew in the dataset class distribution [43]. The PR curve is suitable for imbalanced datasets [44]. The ROC curve should not be used when it is not feasible to generate sufficient data for the model [45]. From Figure 5, it is observed that the ROC and PR curves for the DI models have large areas under the curves compared to the DR model. The objective of a class in ROC and PR spaces is to be in the upper left and upper right corners respectively. The classes in the DI model achieve this goal better than those in the DR and DI COVID-19 models as depicted in Figure 5. This translates into AUCs of 0.96 and 0.99 for DI's ROC and PR curves respectively, while those of DR are 0.91, 0.95 respectively for the ROC and PR curves. Inspection of the ROC curves confirms the assertion that they are not suitable for imbalanced datasets. For example, a comparison of the ROC curve area values for individual classes in Figures 5 (a) and (c) creates the impression that the performance of the two models is almost the same. However, a comparison of the corresponding values of the PR curves clearly shows that some classes underperform in the DR model causing it to lag behind the DI model in terms of performance. Again, the ROC curve of the DI COVID-19 model (Figure 5 (e)) and its accuracy in Table II suggest that the model's overall performance is good. However, a critical look at the PR curve (Figure 5 (f)) shows that the model performed woefully on the imbalanced classes.

F. F-Scores

F-Score is defined as the harmonic mean of precision and recall [45]. It is computed via (7). When α is 1 (α can be 0.5, 1, or 2), the metric called F1-Score is obtained (see (8)).

$$F_\alpha - Score = \frac{(1+\alpha^2)PR * RC}{\alpha^2 * PR + RC} \tag{6}$$

$$F_1 - Score = \frac{2(PR * RC)}{PR + RC} \tag{7}$$

By default, F_α applies additional weights and values to recall than precision (i.e. biased towards recall (RC) when α<1) or vice versa (i.e. biased towards precision (PR) when α>1) [45]. Consequently, this paper proposes the use of F1-Score to evaluate the performance of CapsNets since it is balanced when α=1. From Figure 5, the iso-F1-curves are plotted on the PR curves containing points in the precision/recall space with equal F1-Scores. It is observed that the minimum and maximum F1-Scores for both models are respectively 0.2 and 0.8. However, most of the classes in the DR and DI COVID-19 models have F1-scores below 0.8 indicating inferior performance. In other words, most of the classes in the DI's PR curve (except one class) achieves the goal of falling around the upper-right corner of the PR curve where the F1-Score is 0.8 and above indicating superior performance.

G. Error rate

The error rate is a function of the confusion matrix that finds the ratio of the sum of wrong predictions to the total number of samples. It is computed via (8) or (9). During inference, the (test) error rate quantifies the proportion of instances in which the prediction is wrong.

$$Error\ rate = \frac{FP + FN}{TP + FP + TN + FN} \tag{8}$$

$$Error\ rate = 1 - \frac{TP + TN}{TP + FP + TN + FN} \tag{9}$$

This metric is subject to the flaws of accuracy and can be highly optimistic under class imbalance [46]. The choice of accuracy or error rate is dependent on the personal preference of the researcher.

H. Measuring the generalization ability of the model and its ability to reconstruct input images

A CapsNet model must be able to indicate the level of certainty to which an unseen image belongs to a class during inference. This quantifies the confidence of the model in its output and is very crucial for taking critical decisions in health. In other words, the need to rank the images based on likelihood serves as a means by which the model demonstrates trust in its predictions. The probabilities eliminate ambiguity by providing additional details about which class it strongly predicts as the target class for the image. In this light, CapsNet models that generalize on unseen data must also provide this capability to eliminate ambiguity.

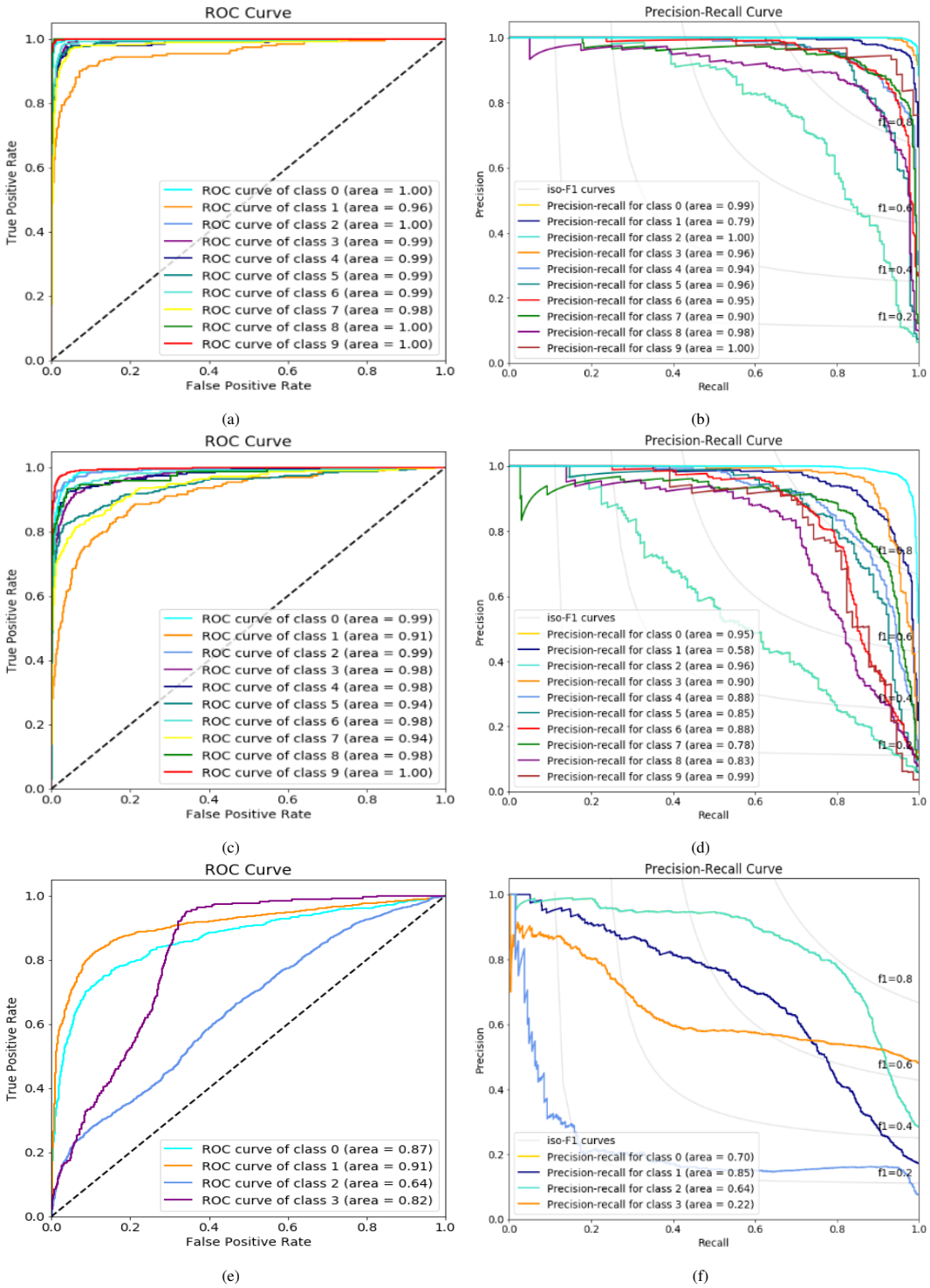
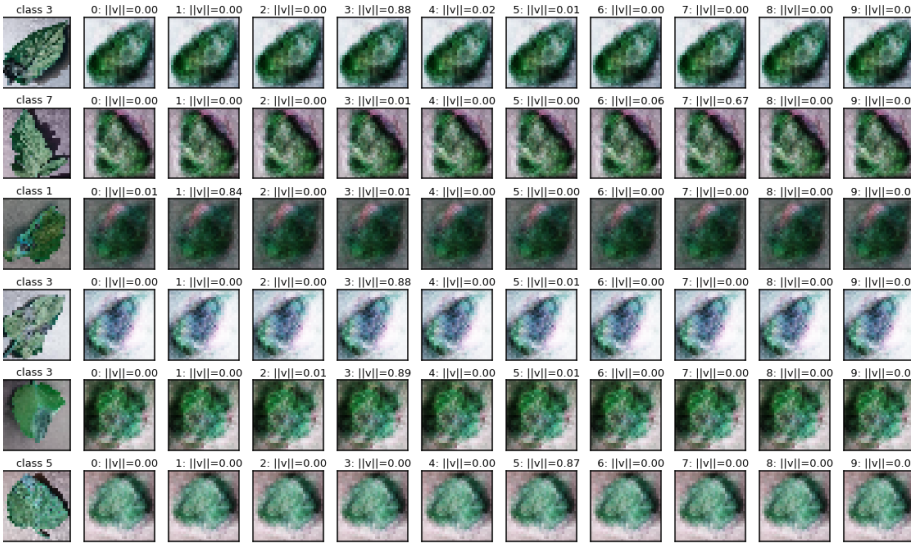
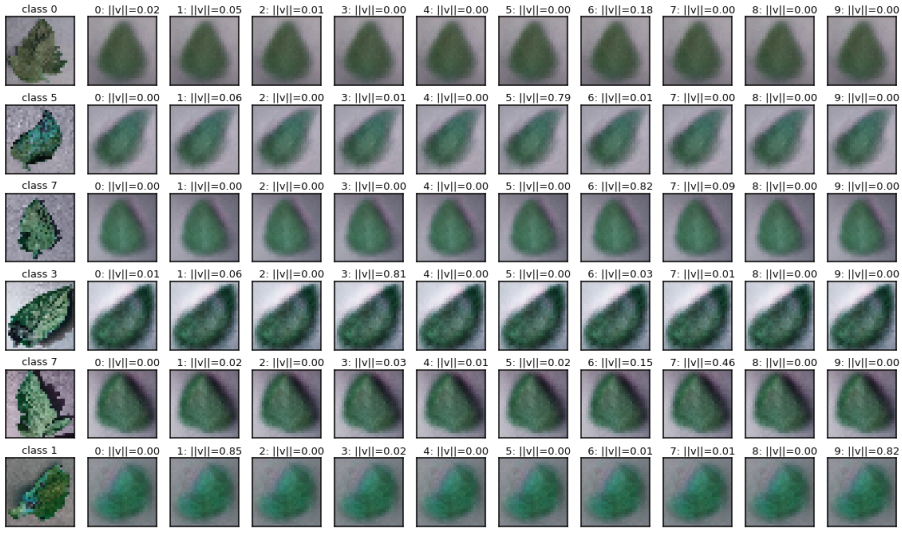


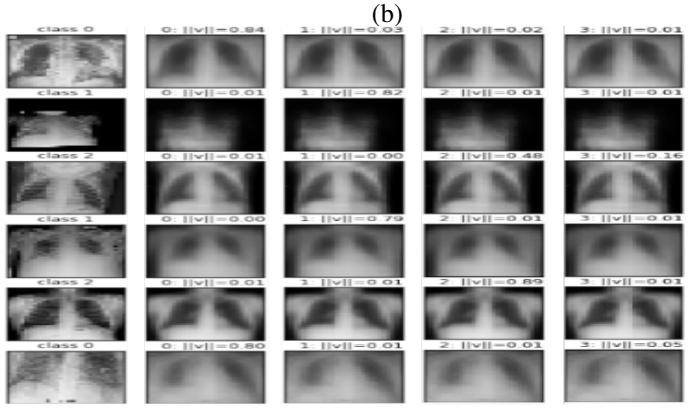
Figure 5. Accuracy Comparison of the multi-class PR and ROC curves for the two models. (a) ROC curve for DI, (b) PR curve for DI, (c) ROC curve for DR, (d) PR curve for DR, (e) ROC curve for DI COVID-19, (f) PR curve for DI COVID-19. Also shown on the PR curves are the iso-F1-curves. Plots (a) to (d) are obtained from the plant disease dataset. The ROC and PR curves for DI and DR (trained with CIFAR 10) are shown in Appendix A Figure A1.



(a)



(b)



(c)

Figure 6. Prediction and reconstruction of the (a) DI model on plant disease, (b) DR model on plant disease, and (c) DI model for COVID-19. The DI model predicted the target classes with high probabilities compared to the DR model. DI also has clearer reconstructed images relative to DR. The first columns are the original images while the rest are reconstructed images each with the probability of prediction.

Figure 6 (a) and (b) show samples of predicted images from all the models. It can be observed that the DI model for plant disease and COVID-19 imposes huge confidence in predicting the correct classes via large probability values (mostly greater than 0.8). Comparatively, the DR model underperforms in this wise as well as, producing 2 wrong predictions. These results are obtained by performing a comparison with the ground truth (GT) images. They also show the models' robustness in their ability to generalize well on unseen data since the test images are preprocessed with the Augmentor image augmentation library to generate different artificial forms of the image for prediction. Reconstruction, on the other hand, allows visual verification of the performance of the model with the ability to boost the confidence of the modeler in the model's output. It is also used by the network as a regularizer in the avoidance of overfitting. The clarity of reconstructed images has a linear relationship with a model's performance. For instance, the reconstructed images in Figure 6(a) are relatively clearer and consistent with the superior performance of the DI model. For capsules, the quality of the reconstruction is a measure of how effectively the network layers use the instantiation parameters (of the ground truth).

I. *Measuring the model's complexity*

Smaller capsule network models just like other deep learning models are beneficial [47] and more efficient for implementation on FPGA and other embedded devices with limited memory. Additionally, they are suitable for distributed online training and introduces smaller overhead during online file transfers. This relatively smaller number of parameters makes the model less computationally complex, reduces the resources needed for inference, and ensures that the model is not exposed to overfitting. Overfitting can be monitored theoretically by ensuring that a k-layer deep learning model has kn_d parameters needed to fit a d-dimensional dataset with n samples [48] perfectly. Overfitting has a negative consequence on performance making it a necessary parameter to monitor in CapsNet models.

Research [49], [42] shows that CapsNet models with smaller parameters can also represent complex functions and outperform deeper models with several millions of parameters. The results in Table IV, confirm this assertion as it can be observed that the high-performing DI model has approximately 4 Million fewer parameters than the DR model on the same dataset.

J. *Evaluating the performance of coupling*

To understand the level to which a CapsNet model can distinguish between the class types, the instantiation parameters can be visualized with the t-distributed stochastic neighbor embedding (TSNE) [50]. The features can be modeled as clusters arising from the coupling between the primary capsules and class capsules depending on whether the agreement a_{ij} between them is high. Qualitatively, the separability of the feature space into distinct clusters; each corresponding to one class, can be used to measure the performance of the coupling. Figure 7 shows a visualization of the feature space beginning with the raw test set shown in Figure 7(a). It can be observed that the raw test set, before routing, has no visible clusters. Figure 7(b) depicts the clusters of features formed by the DI model after routing. The effectiveness of DI's routing algorithm (on the plant disease dataset) can be seen in how separable the clusters are, relative to those formed by the DR and COVID-19 models (Figure 7(c)).

K. *Measuring the model's feature extraction capabilities*

Understanding the decisions made by a CapsNet model requires further investigations to uncover network layers that get more activated by specific regions of the input image. This tends to show the effectiveness of the layers in extracting edge, texture, and shape features. The method is useful in identifying layers with redundant features that give the network some form of robustness when some parts degrade, taking into cognizance the importance of failure avoidance as a major contributor to performance [45]. On the other hand, this method can help determine whether redundant layers have to be removed to reduce the number of parameters and hence reduce model complexity, size, and over-fitting. The necessity to determine the parts of the network that are lacking in feature extraction is paramount since such situations may introduce excessive oscillations and prolong convergence time during training [42]. Figure 8 identifies the Conv3 layer of the DI model (plant disease) as an efficient extractor compared to the rest of the layers. This layer is a higher level one, enabling it to sample features from lower-level layers (Conv1 and Conv2) to represent complete parts of the input image. As a consequence, the DI's PC layer receives enough important features required for the classification. Worth mentioning also is the usefulness of this method to explainability and understandability necessary to achieve the confidence required for practical adoption of CapsNets in critical applications.

5. Conclusion and Future Work

This paper examines existing CapsNet recognition models in health, security, and plant disease recognition and proposes a set of methods to improve performance evaluation to enhance model reliability and confidence. The paper demonstrates the feasibility of the proposed methods by implementing three CapsNet models that are validated experimentally on the plant disease, COVID-19 and CIFAR 10 datasets. The metrics consistently agree on one (the DI) model as the superior model without any contradiction. The use of appropriate performance metrics has the potential to increase the practical adoption of capsule networks in solving critical problems such as early plant disease detection and early cancer diagnoses. In the future, the explainability of capsule networks will be explored to further enhance their acceptance for practical adoption.

References

[1] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *arXiv preprint arXiv:1710.09829*, 2017.

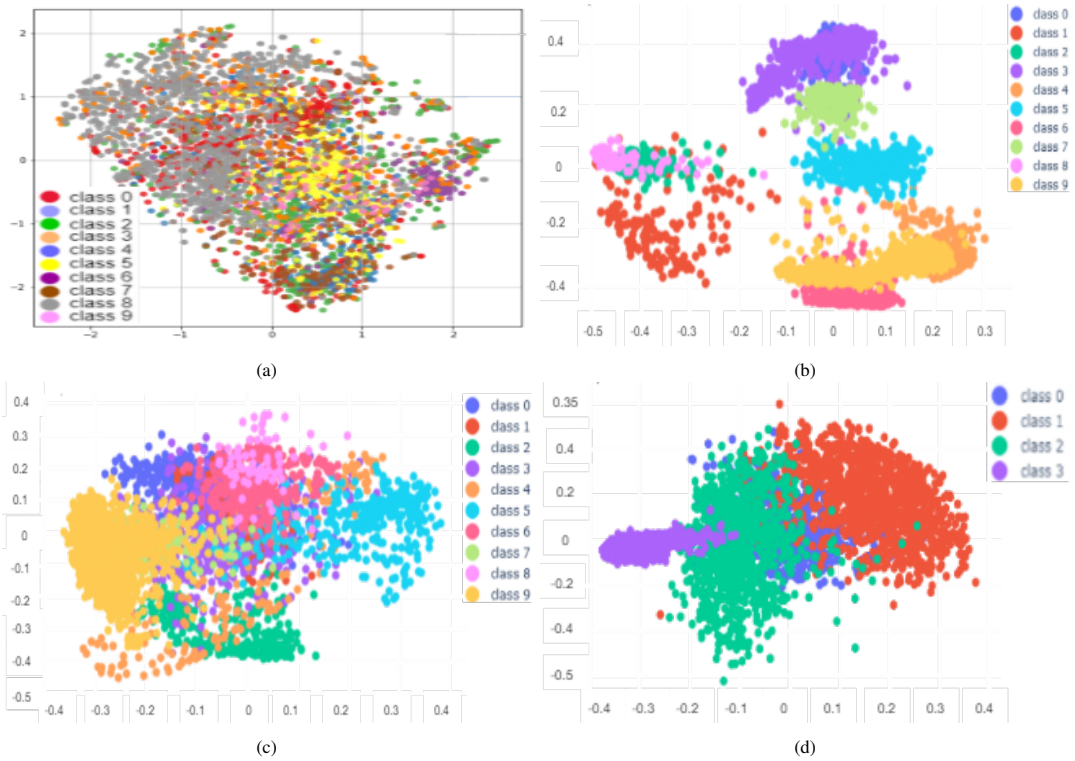


Figure 7. Visualization of the (a) raw plant disease test set, and the cluster of features formed by the (b) DI model on plant disease, (c) DR model on plant disease, (d) DI model on COVID-19 datasets. The cluster of features for DI and DR (trained with CIFAR 10) are shown in Appendix A Figure A2.

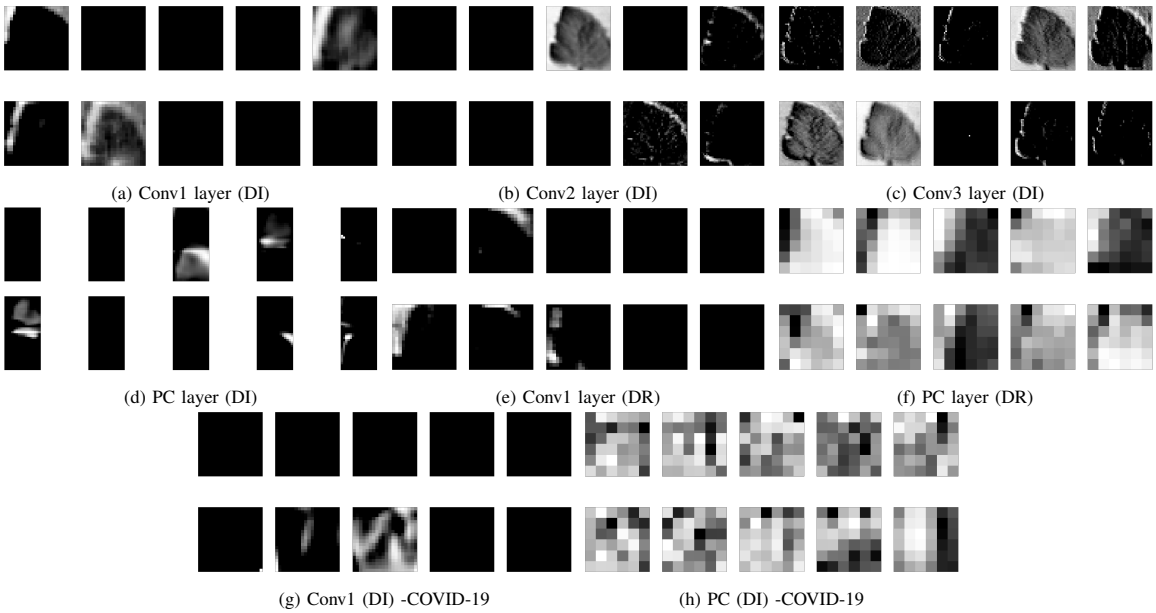


Figure 8. Feature maps for the models. The first three rows are from the plant disease dataset while the last row is from the COVID-19 dataset.

TABLE IV. COMPARISON OF MODEL PARAMETERS

Model	Trainable	Non-trainable	Total
Original CapsNet (DR)	9,864,240	0	9,864,240
Dual Input CapsNet (DI)	6.040,844	458 6,041,302	
COVIDCaps	4,366,312	400	4,366,712
DI_CIFAR 10	5,476,311	422	5,475,889
DR_CIFAR 10	9,348,321	0	9,348,321
Difference (Plant disease)			3,822,938

[2] C. X. Ling, J. Huang, and H. Zhang, "Auc: a better measure than accuracy in comparing learning algorithms," in *Conference of the canadian society for computational studies of intelligence*. Springer, 2003, pp. 329–341.

[3] N. Japkowicz and M. Shah, "Performance evaluation in machine learning," in *Machine Learning in Radiation Oncology*. Springer, 2015, pp. 41–56.

[4] T. Rahman, A. Khandakar, Y. Qiblawey, A. Tahir, S. Kiranyaz, S. B. A. Kashem, M. T. Islam, S. Al Maadeed, S. M. Zughaier, M. S. Khan *et al.*, "Exploring the effect of image enhancement techniques on covid-19 detection using chest x-ray images," *Computers in biology and medicine*, vol. 132, p. 104319, 2021.

[5] M. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M. Kadir, Z. Mahbub, K. Islam, M. Khan, A. Iqbal, N. Al-Emadi *et al.*, "Can ai help in screening viral and covid-19 pneumonia? arxiv 2020," *arXiv preprint arXiv:2003.13145*.

[6] D. Hughes, M. Salathé *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.

[7] D. Ravi, C. Wong, F. Deligianni, M. Berthelot, J. Andreu-Perez, B. Lo, and G.-Z. Yang, "Deep learning for health informatics," *IEEE journal of biomedical and health informatics*, vol. 21, no. 1, pp. 4–21, 2016.

[8] A. Jimenez-Sanchez, S. Albarqouni, and D. Mateus, "Capsule networks against medical imaging data challenges," in *Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*. Springer, 2018, pp. 150–160.

[9] D. Shen, G. Wu, and H.-I. Suk, "Deep learning in medical image analysis," *Annual review of biomedical engineering*, vol. 19, pp. 221–248, 2017.

[10] A. Esteva, K. Chou, S. Yeung, N. Naik, A. Madani, A. Mottaghi, Y. Liu, E. Topol, J. Dean, and R. Socher, "Deep learning-enabled medical computer vision," *NPJ digital medicine*, vol. 4, no. 1, pp. 1–9, 2021.

[11] W. Du, Y. Sun, G. Li, H. Cao, R. Pang, and Y. Li, "Capsnet-ssp: multilane capsule network for predicting human saliva-secretory proteins," *BMC bioinformatics*, vol. 21, no. 1, pp. 1–17, 2020.

[12] 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.

[13] Y. Wang, L. Huang, S. Jiang, Y. Wang, J. Zou, H. Fu, and S. Yang, "Capsule networks showed excellent performance in the classification of herg blockers/nonblockers," *Frontiers in pharmacology*, vol. 10, p. 1631, 2020.

[14] S. Heidarian, P. Afshar, N. Enshaei, F. Naderkhani, M. J. Rafiee, F. B. Fard, K. Samimi, S. F. Atashzar, A. Oikonomou, K. N. Plataniotis *et al.*, "Covid-fact: A fully-automated capsule network-based framework for identification of covid-19 cases from chest ct scans," *Frontiers in Artificial Intelligence*, vol. 4, 2021.

[15] A. Sezer and H. B. Sezer, "Capsule network-based classification of rotator cuff pathologies from mri," *Computers & Electrical Engineering*, vol. 80, p. 106480, 2019.

[16] B. Tang, A. Li, B. Li, and M. Wang, "Capsurv: capsule network for survival analysis with whole slide pathological images," *IEEE Access*, vol. 7, pp. 26 022–26 030, 2019.

[17] C. Kromm and K. Rohr, "Inception capsule network for retinal blood vessel segmentation and centerline extraction," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2020, pp. 1223–1226.

[18] Z. Zhang, S. Ye, P. Liao, Y. Liu, G. Su, and Y. Sun, "Enhanced capsule network for medical image classification," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2020, pp. 1544–1547.

[19] Y. Liu, Y. Ding, C. Li, J. Cheng, R. Song, F. Wan, and X. Chen, "Multi-channel eeg-based emotion recognition via a multi-level features guided capsule network," *Computers in Biology and Medicine*, vol. 123, p. 103927, 2020.

[20] T. Wang, A. Bezerianos, A. Cichocki, and J. Li, "Multikernel capsule network for schizophrenia identification," *IEEE Transactions on Cybernetics*, 2020.

[21] A. S. Sayyed, D. Saha, A. R. Hossain, and C. Shahnaz, "Effectiveness of convolutional and capsule network in malaria parasite detection," in *2019 IEEE International Conference on Signal Processing, Information, Communication & Systems (SPICSCON)*. IEEE, 2019, pp. 68–73.

[22] A. Mehra, "Deepfake detection using capsule networks with long short-term memory networks," Master's thesis, University of Twente, 2020.

[23] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-forensics: Using capsule networks to detect forged images and videos," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 2307–2311.

[24] S. Z. Li, *Encyclopedia of Biometrics: I-Z*. Springer Science &

Business Media, 2009, vol. 2.

[25] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Use of a capsule network to detect fake images and videos," *arXiv preprint arXiv:1910.12467*, 2019.

[26] R. V. Kurup, M. Anupama, R. Vinayakumar, V. Sowmya, and K. Soman, "Capsule network for plant disease and plant species classification," in *International conference on computational vision and bio inspired computing*. Springer, 2019, pp. 413–421.

[27] K. Hajian-Tilaki, "Receiver operating characteristic (roc) curve analysis for medical diagnostic test evaluation," *Caspian journal of internal medicine*, vol. 4, no. 2, p. 627, 2013.

[28] S. Verma, A. Chug, and A. P. Singh, "Exploring capsule networks for disease classification in plants," *Journal of Statistics and Management Systems*, vol. 23, no. 2, pp. 307–315, 2020.

[29] M. Dong, S. Mu, T. Su, and W. Sun, "Image recognition of peanut leaf diseases based on capsule networks," in *International CCF Conference on Artificial Intelligence*. Springer, 2019, pp. 43–52.

[30] B. F. Oladejo and O. O. Ademola, "Automated classification of banana leaf diseases using an optimized capsule network model," in *CS & IT Conference Proceedings*, vol. 10, no. 9. CS & IT Conference Proceedings, 2020.

[31] G. ALTAN, "Performance evaluation of capsule networks for classification of plant leaf diseases," *International Journal of Applied Mathematics Electronics and Computers*, vol. 8, no. 3, pp. 57–63, 2020.

[32] P. M. Kwabena, B. A. Weyori, and A. A. Mighty, "Gabor capsule network for plant disease detection," *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 11, no. 10, 2020.

[33] M. K. Patrick, B. A. Weyori, and A. A. Mighty, "Max-pooled fast learning gabor capsule network," in *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*. IEEE, 2020, pp. 1–8.

[34] H. T. Rauf, B. A. Saleem, M. I. U. Lali, M. A. Khan, M. Sharif, and S. A. C. Bukhari, "A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning," *Data in brief*, vol. 26, p. 104340, 2019.

[35] "Capsnet-keras." [Online]. Available: https : //github.com/XifengGuo /CapsNet-Keras

[36] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[37] M. Bloic, "Augmentor-image augmentation library in python for machine learning," *Retrieved June*, vol. 9, p. 2020, 2017.

[38] N. Japkowicz and M. Shah, *Evaluating learning algorithms: a classification perspective*. Cambridge University Press, 2011.

[39] F. Provost, T. Fawcett, and R. Kohavi, "The case against accuracy estimation for comparing induction algorithms 1998," in *Proceedings of the 15th international conference on machine learning ICML-98 Morgan Kaufmann. San Mateo, CA*.

[40] Y. Zhao and Y. Cen, *Data mining applications with R*. Academic Press, 2013.

[41] F. J. Valverde-Albacete and C. Peláez-Moreno, "100% classification accuracy considered harmful: The normalized information transfer factor explains the accuracy paradox," *PloS one*, vol. 9, no. 1, p. e84217, 2014.

[42] P. M. Kwabena, B. A. Weyori, and A. A. Mighty, "Exploring the performance of lbp-capsule networks with k-means routing on complex images," *Journal of King Saud University-Computer and Information Sciences*, 2020.

[43] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 233–240.

[44] P. Singla and P. Domingos, "Discriminative training of markov logic networks," in *AAAI*, vol. 5, 2005, pp. 868–873.

[45] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation," in *Australasian joint conference on artificial intelligence*. Springer, 2006, pp. 1015–1021.

[46] H. Daumé, *A course in machine learning*. Hal Daumé III, 2017.

[47] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[48] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," *Communications of the ACM*, vol. 64, no. 3, pp. 107–115, 2021.

[49] C. W. Wu, "Prodsumnet: reducing model parameters in deep neural networks via product-of-sums matrix decompositions," *arXiv preprint arXiv:1809.02209*, 2018.

[50] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

Appendix A

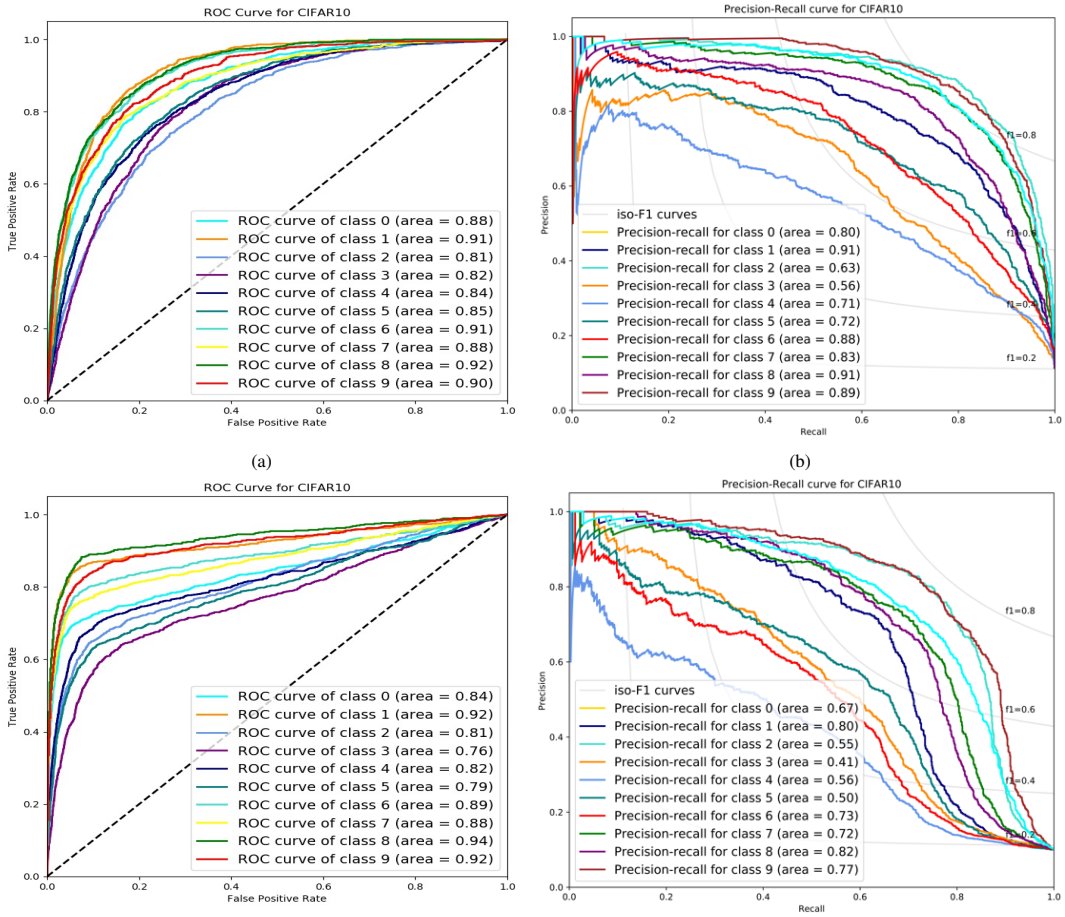


Figure A1. Comparison of the multiclass ROC and PR curves for the two models on CIFAR 10. (a) ROC curve for DI, (b) PR curve for DI, (c) ROC curve for DR, (d) PR curve for DR. Also shown on the PR curves are the iso-F1-curves.

Figure A2. Visualization of the cluster of features at the class capsule layer for (a) DI on CIFAR 10 and (b) DR on CIFAR 10. The DI model outperformed the DR model as classes 5 and 8 do not form compact clusters for the DR model. Class 5 for DI is also not compact but it does not form two separate clusters as is the case for the DR model.

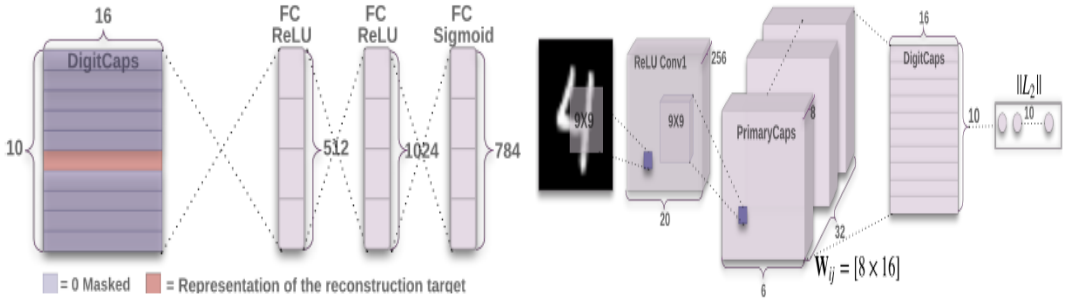


Figure A3. The DR architecture from [1].

Patrick Kwabena Mensah Patrick Kwabena Mensah is a PhD student in Computer Science and a Senior lecturer at the Department of Computer Science and Informatics, University of Energy and Natural Resources, Sunyani, Ghana. He received his BSc and MSc in Computer Science from UDS and AUST respectively in 2009 and 2011. His research interest includes Deep Learning for computer vision, Artificial Intelligence of Things and Residue Number Systems.

Mighty Abra Ayidzoe Mighty Abra Ayidzoe is a PhD Software Engineering Student at the University of Electronic Science and Technology of China (UESTC). She is currently a lecturer at the Department of Computer Science and Informatics, University of Energy and Natural Resources, Sunyani, Ghana. She has a BSc in Computing and an MEng in Software Engineering from UESTC, China. She specializes in computer vision algorithms.