Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 4984490, 15 pages
https://doi.org/10.1155/2022/4984490

Research Article

Uncertainty Estimation Using Variational Mixture of Gaussians Capsule Network for Health Image Classification

Patrick Kwabena Mensah,[1] **Mighty Abra Ayidzoe**,[1] **Alex Akwasi Opoku**,[2] **Kwabena Adu**,[1] **Benjamin Asubam Weyori**,[1] **Isaac Kofi Nti**,[1,3] **and Peter Nimbe**[1]

[1]*Department of Computer Science and Informatics, University of Energy and Natural Resources, P. O. Box 214, Sunyani, Ghana*
[2]*Department of Mathematics and Statistics, University of Energy and Natural Resources, P. O. Box 214, Sunyani, Ghana*
[3]*School of Information Technology, University of Cincinnati, OH, USA*

Correspondence should be addressed to Mighty Abra Ayidzoe; mighty.ayidzoe@uenr.edu.gh

Received 22 April 2022; Revised 13 July 2022; Accepted 20 July 2022; Published 30 September 2022

Academic Editor: Mohit Mittal

Capsule Networks have shown great promise in image recognition due to their ability to recognize the pose, texture, and deformation of objects and object parts. However, the majority of the existing capsule networks are deterministic with limited ability to express uncertainty. Many of them tend to be overconfident on out-of-distribution data, making them less trustworthy and hence reducing their suitability for practical adoption in safety-critical areas such as health and self-driving cars. In this work, we propose a capsule network based on a variational mixture of Gaussians to train distributions of network weights as opposed to a single set of weights and enable the model to express its predictive uncertainty on out-of-distribution data. Training distributions of weights have the added advantage of avoiding overfitting on smaller datasets which are common in health and other fields. Although Bayesian neural networks are known to exhibit slow training and convergence, experimental results show that the proposed model can retrieve only relevant features, converge faster, is less computationally complex, can effectively express its predictive uncertainties, and achieve performance values that are comparable to the state-of-the-art models. This is an indication that CapsNets can exhibit the transparency, credibility, reliability, and interpretability required for practical adoption.

1. Introduction

Recently, there has been an upsurge in the adoption of Deep Learning (DL) to perform complex tasks such as Visual Question Answering [1], and plant disease detection [2], among others, due to their excellent performance in terms of speed and accuracy compared to humans. Capsule Networks [3, 4], for example, have demonstrated the ability to recognize the pose, texture, and deformation of an object and its parts. They have thus been proposed for use in sensitive areas such as health [5, 6] and agriculture [7, 8], among others. Irrespective of the sensitivity of the application area, capsule networks (just like many other deep learning models) do not incorporate uncertainties in their predictions. The inability to model uncertainties leads to model over/under confidence [9]. We propose a Bayesian Capsule Network (BCN) motivated by [10, 11] and on the background that the Bayesian framework provides the capability for modeling uncertainties in neural network predictions [12]. Bayesian Neural Networks (BNNs) estimate uncertainties by defining a distribution over the network weight parameters whose posterior weight distribution $p(.\mid x)$ permits the BNN to capture the prediction uncertainties.

BNNs are known to have a longer convergence time during training [11] since training occurs on larger distribution parameters compared to single points in deterministic models. However, the choice of appropriate normalization and weight initialization schemes can allow the network to converge faster. Since Bayesian models replace the fixed weights with probability distributions, they are capable of training on smaller datasets without overfitting.

This work, therefore, proposes a Variational Mixture of Gaussian-based capsule network (CapsNet) that will contribute to solving problems such as those caused by the lack of huge datasets in critical areas (e.g., in health). Additionally, we aim at reducing model complexity, reducing convergence time, and improving accuracy on difficult datasets that are small and imbalanced. These are difficult targets for a Bayesian model known for its complexity, and inability to converge faster to achieve. We also aim to leverage the ability of the BNN to model uncertainties and introduce some form of reliability in the predictions of the model on input images. The motive is to enable such models to gain the confidence of the practitioner for practical adoption in safety-critical areas such as autonomous cars and medicine. The lack of sufficient training data is a major limiting factor to the adoption of deep learning in areas such as health due to concerns related to overfitting. This work, therefore, uses Bayesian NNs to elegantly avoid this problem by acting on the distributions weights as opposed to deterministic models which train on a single set of weights. For instance, the parameter θ of a distribution on the weights $p(w \mid \theta)$ is learned by Variational Inference leading to the minimization of Kullback–Leibler (KL) divergence. This method provides a principled framework for the usage of model components leading to better monitoring of model complexity and avoiding its associated problems such as overfitting. In addition, regularization is natural to BNNs such that the regularization parameters get consistent treatment in the Bayesian setting thus eliminating the need for techniques such as cross-validation [13]. Perhaps, one of the main benefits of our method to the health and other critical sectors is the model's ability to avoid overconfident predictions in regions of sparse data.

Experimental results show that our proposed Variational Mixture of Gaussians Routing (VMGs-Routing) achieves a significant reduction in model complexity while achieving competitive results compared to the state-of-the-art models. Our routing algorithm improves upon similar existing routing algorithms by training and learning faster to achieve convergence within a few epochs (approximately 100 epochs). This method further reduces the infinite likelihood and zero variance problem inherent in Maximum Likelihood solutions caused by Gaussian clusters that try to take sole possession of data points (also known as polarization in Capsules).

The contributions of this paper can be summarized as follows:

(1) We propose a routing method from a variational mixture of Gaussians that clearly relies on the maximization of the evidence lower bound (ELBO) to activate a capsule.

(2) We provide empirical results that are comparative to state-of-the-art previous works on Bayesian and deterministic capsules to demonstrate that our approach does not result in the loss of any of the inherent strengths of capsules such as viewpoint-invariance, robustness.

(3) We show that our proposed Bayesian CapsNet is not overconfident and is reliable from the high uncertainty it expresses on out-of-distribution data.

(4) The proposed model is less computationally complex and performs comparatively well with deep Bayesian CapsNet models from the literature in terms of accuracy, uncertainty estimation, and prediction. Comparatively, our model achieves better speedup during training and testing without performance degradation.

(5) We provide extensive visualizations of layer activation maps, and predictive uncertainty plots, among others in an attempt to increase the interpretability of our model which is presumed (as a Bayesian model) to be a complex probabilistic 'black box' model.

The rest of the paper is organized in the following way: Section 2 presents the related works in the literature followed by Section 3 which discusses the Bayesian methods adopted for this work. Section 4 presents the experiments and experimental results after which the paper is concluded in Section 5.

2. Related Work

Some works in the literature have relied on variational inference to propose capsules to solve varied problems. Smith et al. [14] proposed a probabilistic capsule (CapsNet) to encode the capsule assumptions and separate the generative and inference parts from each other. They showed that their model can generalize well on out-of-distribution data, but did not express the uncertainty of their model. Ribeiro et al. [11] proposed a Bayesian CapsNet routing algorithm based on a mixture of transforming Gaussians to address the variance collapse problem and to model the uncertainty of the pose parameters. However, experimental results of the uncertainty of the pose parameters were not provided. In this implementation, a parent capsule j is activated if there is an agreement between the votes of adjacent capsules. The agreement is measured by the entropy of the multivariate Gaussian distribution. A conditional variational CapsNet [15] was proposed to detect classes that are not known during training as a contribution to the open set recognition problem. To this end, they adopted the variational autoencoder approach enabling similar features to assume the shape of a Gaussian, such that each unique feature assumed a different Gaussian. A flow-based model with a long flow structure is capable of finding the approximate posterior probability compared to utilizing a simple family of distributions to approximate the intractable posterior. However, as the data increase in dimensionality, this solution gives rise to huge computational complexity and variance. To address this shortcoming, Hua et al. [16] utilized a dynamic routing flow with variational inference to achieve a shorter flow structure and a significant improvement in precision and accuracy. To introduce routing uncertainties in CapsNet, Ribeiro et al. [17] proposed a global view of the local iterative routing between capsules of adjacent layers, enabling them to capture the uncertainty in the assignment of parts to objects. Compared to the two previous works mentioned earlier, this partial Bayesian CapsNet produced results on out-of-distribution predictive entropies that were consistent with uncertainties of model

predictions. To avoid the singularity problem caused by maximum likelihood estimation (MLE), a variational routing CapsNet [18] has been proposed to utilize the variational distribution and integrate the prior distribution for automatic determination of the class of data and avoid overfitting. A Bayesian capsule encoder [19] was proposed to regulate the standard deviation and mean in latent space. The authors argue that it is a better approach for the retrieval of relevant features and image reconstruction from latent space. To demonstrate that deep variational CapsNets can achieve better performance on image synthesis and analysis, Huang et al. [20] proposed a variational model in which the divergence between a capsule and a given prior distribution defines the presence of different entities in an object.

Traditionally, uncertainty is modeled with probability theory and is increasingly becoming more relevant due to the adoption of deep learning (DL) models in practical and safety-critical applications such as medicine and self-driving cars. This type of modeling uses a single probability distribution to capture the required knowledge and struggles to express the two types of uncertainties in a DL model [21]. Aleatoric uncertainty arises from the element of randomness due to the variability of the outcome of events, while epistemic uncertainty measures the modeler(s) inability to design the best model for the task at hand. In the literature, Bayesian networks with latent variables have been proposed [22] to measure both the predictive aleatoric and epistemic uncertainties. This approach played a significant role in the interpretability of the model, which, like other neural network models is perceived to be a "black box." With the inherent advantages of CapsNets over other neural networks, our work proposes a variational mixture of Gaussians routing-based capsules to effectively capture the predictive uncertainty on the in and out-of-distribution data to improve reliability, interpretability, and model confidence for safety-critical applications.

3. Proposed Methods

In this section, we outline a brief introduction to the concepts of Variational Inference and Gaussian mixture models on which our routing algorithm is based.

3.1. Bayesian Mixture of Gaussians. Suppose X assumes a Gaussian distribution; a linear combination of these Gaussians forms the basis for the formulation of a mixture of probabilistic (Gaussian) models known as a mixture of Gaussians [10]. This convex combination creates the opportunity to adjust the means, covariances, and coefficients as a basis for approximating any continuous density function to arbitrary accuracy. Considering a superposition of K-Gaussian densities taking the form of the joint probability $p(x,z) = p(x \mid z)p(z)$, z can be marginalized out to give $p(x) = \sum_{z=1}^{K} p(x,z) = \sum_{z=1}^{K} p(z)p(x \mid z)$. Realizing that the mixing coefficient $\pi_z = p(z) = 1/K$ (K is a one-hot-vector) is the probability of choosing one cluster out of K clusters, the marginal probability can be rewritten in the form of a Gaussian Mixture Model (GMM), shown in equation (1):

$$p(x) = \sum_{k=1}^{K} \pi_z \mathcal{N}(x_i \mid \mu_k, \Sigma_k). \tag{1}$$

The Gaussian density (also called component) in the above expression has its own mean μ_k and covariance Σ_k.

Since routing in capsules operates on the concept of clustering, they can naturally be modeled via a mixture of transforming Gaussians [11].

3.2. Variational Bayes. Bayesian algorithms perform inference on unknown random variables by finding a posterior probability density [23] in situations where the posterior is intractable to compute. Approximate inference (using Variational Inference (VI)) provides a reasonable approximation to the problem compared to Markov Chain Monte Carlo (MCMC) methods that provide an exact solution but with slow convergence time.

Using the Bayes theorem, the posterior probability density can be computed as follows:

$$\begin{aligned} p(\vartheta \mid x) &= \frac{p(\vartheta, x)}{p(x)} \\ &= \frac{p(x \mid \vartheta)p(\vartheta)}{p(x)} \\ &= \frac{p(x \mid \vartheta)p(\vartheta)}{\int_{\vartheta} p(x,\vartheta)d\vartheta}, \end{aligned} \tag{2}$$

where $\int_{\theta} p(x,\vartheta)\mathrm{d}\vartheta$ is the marginal probability (also called the evidence). This term is intractable, requiring the use of approximate solutions such as VI. VI does this by searching a family of distributions Q for the distribution q that is closest to the posterior $p(. \mid x)$. The distance between the variational ("*nice*") distribution q and the true posterior $p(. \mid x)$; is measured by the Kullback–Leibler (KL) divergence.

$$\begin{aligned} KL[q \| p(. \mid x)] &= \int q(\vartheta) \log \frac{q(\vartheta)}{p(\vartheta \mid x)} d\vartheta \\ &= \mathbb{E}_q \log \frac{q(\vartheta)}{p(\vartheta \mid x)} \\ &= \log p(x) - \int q(\vartheta) \log \frac{p(x,\vartheta)}{q(\vartheta)} \mathrm{d}\vartheta. \end{aligned} \tag{3}$$

Therefore, minimization of the KL over q now becomes maximization of the Evidence Lower Bound (ELBO)

$$\begin{aligned} \mathrm{ELBO} &= \mathscr{L}[q](x) \\ &= \int q(\vartheta) \log \frac{p(x,\vartheta)}{q(\vartheta)} \mathrm{d}\vartheta, \end{aligned} \tag{4}$$

to avoid the intractability issues of the true posterior $p(\vartheta \mid x)$. To maximize the ELBO, the vector of hidden random variables $\theta = (\theta_1, \theta_2, \ldots, \theta_n)$ (distributed according to the

variational distribution q) are assumed to be made up of independent random variables allowing their joint distribution to be obtained from the product of their marginal distributions.

$$q(\theta) = q(\theta_1, \theta_2, \ldots \theta_n) = \prod_{i=1}^{n} q_i(\theta_i). \tag{5}$$

This mean-field (MF) approximation makes it possible to obtain a free-form optimization of the ELBO $\mathscr{L}[q]$ with respect to all the distributions $q_i(\theta_i)$ by optimizing each of the factors in turn. When the $\mathscr{L}[q]$ is fully described by the MF distribution, every data point described by a variational distribution will have its own free parameters. The task is to then find the free parameters that will maximize $\mathscr{L}[q]$.

In this study, it is assumed that data points, which are the realization of the random variables $X_1, \ldots, X_N$, are taken from the m-dimensional Euclidean space R^D. Thus, the dataset $X = (X_1, \ldots, X_N)$ is a vector with R^D-valued random coordinates that are to be classified into K clusters with random centroids $H_1, \ldots, H_K$ that are multinormally distributed, i.e., $H_k \sim N(\mu_k, \Delta_k^{-1})$, where $k = 1,.., K, \mu_{\mathbf{k}}$ is the $1 \times D$ mean-vector and $\Delta_{\mathbf{k}}^{-1}$ the $D \times D$ covariance matrix. In what follows, f_k will be written for the density of $N(\mu_k, \Delta_k^{-1})$. Whenever the random variable X_n is in the k^{th} cluster, it then assumes the distribution of the centroid of that cluster. Thus, each data point X_n is distributed according to $N(\mu_k, \Delta_k^{-1})$, for some $= 1,.., K$. In the sequel we denote by C_n, the cluster label of the random variable X_n, for $n = 1, \ldots, N$. To each data point X_n, corresponds a latent variable Z_n, that is a 1-of-K binary vector with π_k being the probability $Z_{nk} = 1$, for some $k = 1,.., K$. Therefore, $\pi = (\pi_1, \ldots, \pi_K)$, called the vector of mixing coefficients, is a probability vector and $N = (N_1, \ldots, Y_K) = Z_1 + Z_2 + \ldots + Z_N$ is a random vector with K nonnegative coordinates that sum up to N. In fact, Y is multinomially distributed with parameters N and π. Observe that for any $n = 1, \ldots, \mathrm{N}$, the probability that $Z_n = z_n$ is given by the following equation:

$$p(z_n) = \prod_{k=1}^{K} \pi_k^{z_{nk}}. \tag{6}$$

Putting $\theta = (Z, \pi, \mu, \Lambda)$, with $Z = (\mathrm{Z}_1, \ldots, \mathrm{Z}_\mathrm{N})$, $\mu = (\mu_1, \mu_2, \ldots, \mu_\mathrm{K})$ and $\Lambda = (\Lambda_1, \Lambda_2, \ldots, \Lambda_\mathrm{K})$, the joint distribution of X and θ can be written as follows:

$$\begin{aligned} p(X, \theta) &= p(X \mid Z, \mu, \Lambda) p(Z \mid \pi, \mu, \Lambda) p(\pi \mid \mu, \Lambda) p(\mu \mid \Lambda) p(\Lambda) \\ &= p(X \mid Z, \mu, \Lambda) p(Z \mid \pi) p(\pi) p(\mu \mid \Lambda) p(\Lambda). \end{aligned} \tag{7}$$

The second equality of equation (7) uses that $p(Z \mid \pi, \mu, \Lambda) = p(Z \mid \pi)$ and $p(\pi \mid \mu, \pi) = p(\pi)$. We assume further that conditioning on θ, the components of X are independent. Similarly, given π and Λ, the components of Z and μ are respectively independent. Furthermore, the components of Λ are also independent. In addition to the above prescription, we use the plate notation (directed graph) [10, 24] to derive our priors and put the problem in a Bayesian setting. Thus, using the conjugate priors of , Λ and π, and the above-given result in

$$\begin{aligned} \pi &\sim \mathrm{SymDir}(K, \propto_0), \\ \Lambda_{k=1,\ldots,K} &\sim Wi(W_0, \nu_0), \\ \mu_{k=1,\ldots,K} &\sim N\big((\mu_0), (\beta_0 \Lambda_k)^{-1}\big), \\ Y = (Y_1, \ldots, Y_K) &\sim \mathrm{Mult}(N, \pi), \\ X_{i=1,\ldots,N} &\sim N\big(\mu_{C_i}, \Lambda_{C_i}^{-1}\big), \end{aligned} \tag{8}$$

Therefore,

$$p(x \mid z, \mu, \Lambda) = \prod_{n=1}^{N} \prod_{k=1}^{K} \mathbf{f_k}(x_n)^{z_{nk}}, \tag{9}$$

$$p(z \mid \pi) = \prod_{n=1}^{N} \left(\prod_{k=1}^{K} (\pi_k)^{z_{nk}} \right), \tag{10}$$

$$p(\pi) = \frac{\Gamma(K \propto_0)}{\Gamma(\propto_0)^K} \prod_{k=1}^{K} \pi_k^{\alpha_0 - 1}, \tag{11}$$

$$p(\mu \mid \Lambda) = \prod_{k=1}^{K} f_k^0(\mu_k), \tag{12}$$

$$p(\Lambda) = \prod_{k=1}^{K} Wi(\Lambda_k), \tag{13}$$

where

$$\begin{aligned} f_k(x_n) &= \frac{1}{2\pi^{D/2}} \frac{1}{|\Lambda_k^{-1}|^{1/2}} \exp\left\{ -\frac{1}{2} (x_n - \mu_k)^T \Lambda_k (x_n - \mu_k) \right\}, \\ f_k^0(\mu_k) &= \frac{1}{2\pi^{D/2}} \frac{1}{|(\beta_0 \Lambda_k)^{-1}|^{1/2}} \exp\left\{ -\frac{1}{2} (\mu_k - \mu_0)^T \beta_0 \Lambda_k (\mu_k - \mu_0) \right\}, \\ Wi(\Lambda_k) &= B(W_0, \nu_0) |\Lambda_k|^{(\nu_0 - D - 1)/2} \exp\left(-\frac{1}{2} Tr\left(W_k^{-1} \Lambda_k \right) \right), \\ B(W_0, \nu_0) &= |W_0|^{-\nu_0/2} \left\{ 2^{\nu_0 D/2} \pi^{D(D-1)/4} \prod_{\mathrm{i}=1}^{D} \Gamma\left(\frac{\nu_0 + 1 - i}{2} \right) \right\}^{-1}. \end{aligned} \tag{14}$$

From the joint distribution in (7), we identify the posterior and variational ('*nice*') distributions as $p(Z, \mu, \Lambda, \pi \mid \mathrm{X})$ and $q(Z, \mu, \Lambda, \pi)$ i.e., the $p(\theta \mid X)$ and $q(\theta)$ respectively, providing the ingredients for the computation of $KL[q(Z, \mu, \Lambda, \pi) \| p(Z, \mu, \Lambda, \pi \mid X)]$. Accordingly, the variational distribution (VD) is factorized based on the MF approximation method to obtain $q(\mathrm{Z}, \mu, \Lambda, \pi) = q(\mathrm{Z}) q(\mu, \Lambda, \pi)$. Meanwhile, from the MF approximation, it can be shown that the best distribution q_j for maximizing the ELBO is $q_j^*(. \mid x)$, satisfying $\ln q_j^*(z \mid x) = \ln p(z_j, x) + \text{constant}$. We consequently model the joint distribution in (7) according to the aforementioned best variational distribution. Initial calculations involve the determination of $q^*(z \mid x)$ followed by $q^*(\pi, \mu, \Lambda)$. In other words,

$$\log \mathbf{q}^{*}(z \mid x) = \mathbb{E}_{q(\pi,\mu,\Lambda)}[\log(p(x \mid z,\mu,\Lambda)p(z \mid \pi)p(\pi)p(\mu,\Lambda))] \&9; + \text{const.} \tag{15}$$

Pushing the variables not dependent on z (i.e. $p(\pi)p(\mu,\Lambda)$) into the constant, we obtain the following equation:

$$\log \mathbf{q}^{*}(z \mid x) = \mathbb{E}_{q(\pi,\mu,\Lambda)}[\log(p(x \mid z,\mu,\Lambda)p(z \mid \pi))] + \text{const} \tag{16}$$

Substituting (9) and (10) into the expression for $\log q^{*}(z \mid x)$, produces

$$\log \mathbf{q}^{*}(\mathbf{z} \mid \mathbf{x}) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\log(\rho_{nk}(x_n)) + \text{const}, \tag{17}$$

where

$$\log \rho_{nk}(x_n) = \mathbb{E}_{q(\pi)}[\log(\pi_k)] - \frac{1}{2}\mathbb{E}_{q(\pi)}\left[\log\left(\left|\Lambda_k^{-1}\right|\right)\right] - \frac{D}{2}\log(2\pi) - \frac{1}{2}\mathbb{E}_{q(\mu_k,\Lambda_k)}\left[(x_n-\mu_k)^{T}\Lambda_k(x_n-\mu_k)\right]. \tag{18}$$

Exponentiating $\log q^{*}(z \mid x)$ and normalizing it to let ρ_{nk} sum to 1 over all the values of k produces

$$q^{*}(z \mid x) = \prod_{n=1}^{N}\prod_{k=1}^{K} r_{nk}^{z_{nk}}(x_n), \tag{19}$$

where

$$r_{nk}(x_n) = \frac{\rho_{nk}(x_n)}{\sum_{j=1}^{K}\rho_{nj}(x_n)}. \tag{20}$$

The best $q^{*}(z \mid x)$, therefore, is a product of categorical distributions for each latent variable having r_{nk} for $k = 1, 2, \ldots, K$ as parameters.

On the other hand, the best variational distribution $q^{*}(\pi,\mu,\Lambda)$ can be divided into two components $q^{*}(\pi)$ and $q^{*}(\mu,\Lambda)$. It follows from the product rule, the deductions leading to equations (15), (9) and (10) that $q^{*}(\pi)$ satisfies

$$\begin{aligned}\log q^{*}(\pi) &= \log p(\pi) + \mathbb{E}_{q(z)}[\log p(z \mid \pi)] + \text{const}\\ &= (\alpha_0 - 1)\sum_{k=1}^{K}\ln \pi_k + \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\ln \pi_k + \text{const}\\ &= (\alpha_0 - 1)\sum_{k=1}^{K}\ln \pi_k + \sum_{k=1}^{K} y_k \ln \pi_k + \text{const}.\end{aligned} \tag{21}$$

Taking exponentials of both sides of the above expression and taking care of the normalizing term result in

$$\begin{aligned}q^{*}(\pi) &= \text{Dir}(K,\alpha_1,\ldots,\alpha_K)\\ &= C(\alpha_1,\ldots,\alpha_K)\prod_{k=1}^{K}\pi_k^{\alpha_k - 1},\end{aligned} \tag{22}$$

where

$$\begin{aligned}\alpha_k &= \alpha_0 + y_k \text{ and}\\ y_k &= \sum_{n=1}^{N} z_{nk}.\end{aligned} \tag{23}$$

Upon some computations, the variational distribution $q^{*}(\mu,\Lambda)$ for the joint distribution $q(\mu,\Lambda)$ takes the form

$$\begin{aligned}\log(q^{*}(\mu,\Lambda)) &= \log(p(\mu,\Lambda)) + \mathbb{E}_{q(z)}[\log(p(x \mid z,\mu,\Lambda))] + \text{const}\\ &= \sum_{k=1}^{K}\left[\log\left(f_k^{0}(\mu_k)\right) + \log(Wi(\Lambda_k))\right] + \text{const},\end{aligned} \tag{24}$$

where f_k^{0} and Wi are respectively the Gaussian and Wishart densities (see equations (12) and (13)) with parameters m_k, β_k, W_k, and ν_k. These parameters are given as follows:

$$\begin{aligned}\beta_k &= \beta_0 + y_k,\\ m_k &= \frac{1}{\beta_k}(\beta_0\mu_0 + y_k\overline{x}_k),\\ W_k^{-1} &= W_0^{-1} + y_k S_k + \frac{\beta_0 y_k}{\beta_0 + y_k}(\overline{x}_k - \mu_0)(\overline{x}_k - \mu_0)^{T},\\ \nu_k &= \nu_0 + y_k,\\ y_k &= \sum_{n=1}^{N} z_{nk},\\ \overline{x}_k &= \frac{1}{y_k}\sum_{n=1}^{N} z_{nk}x_n,\\ S_k &= \frac{1}{y_k}\sum_{n=1}^{N} z_{nk}(x_n - \overline{x}_k)(x_n - \overline{x}_k)^{T}.\end{aligned} \tag{25}$$

To evaluate r_{nk}, the quantities in ρ_{nk} are expressed as follows:

$$\begin{aligned}\mathbb{E}_{q^{*}(\mu_k,\Lambda_k)}\left[(x_n-\mu_k)^{T}\Lambda_k(x_n-\mu_k)\right] &= D\beta_k^{-1} + \nu_k(x_n-m_k)^{T}W_k(x_n-m_k),\\ \ln\tilde{\Lambda}_k \equiv \mathbb{E}\left[\ln\left|\tilde{\Lambda}_k\right|\right] &= \sum_{i=1}^{D}\psi\left(\frac{\nu_k+1-i}{2}\right) + D\ln 2 + \ln|W_k|,\\ \ln\tilde{\pi}_k \equiv \mathbb{E}[\ln|\pi_k|] &= \psi(\alpha_k) - \psi\left(\sum_{i=1}^{K}\alpha_i\right),\end{aligned} \tag{26}$$

where ψ is the log derivative of the multinomial gamma function.

After the substitutions, ρ_{nk} becomes,

$$
\begin{aligned}
ln\rho_{nk} &= [\psi(\alpha_k) - \psi(\widehat{\alpha}_k)] + \frac{1}{2}\left[\sum_{i=1}^{D}\psi\left(\frac{\nu_k + 1 - i}{2}\right) + Dln2 + ln|W_k|\right] \\
&\quad - \frac{D}{2}\ln(2\pi) - \frac{1}{2}\left[D\beta_k^{-1} + \nu_k(x_n - m_k)^T W_k(x_n - m_k)\right], \\
ln\rho_{nk} &= \psi(\alpha_k) - \psi(\widehat{\alpha}_k) + \frac{1}{2}\sum_{1=1}^{D}\psi\left(\frac{\nu_k + 1 - i}{2}\right) + \frac{1}{2}Dln2 \\
&\quad + \frac{1}{2}ln|W_k| - \frac{D}{2}\ln(2\pi) - \frac{1}{2}D\beta_k^{-1} - \frac{1}{2}\nu_k(x_n - m_k)^T W_k(x_n - m_k),
\end{aligned}
\tag{27}
$$

where $\sum_{i=1}^{K}\alpha_i = \widehat{\alpha}_i$. There is a circular dependency between these variational parameters requiring n iterative updates that ensure the algorithm converges to an approximate posterior.

Using equation (7), the ELBO for a VGM model is obtained as follows:

$$
\mathscr{L} = \mathbb{E}[\text{In } p(X, Z, \pi, \mu, \Lambda)] - \mathbb{E}[\text{In } q(Z, \pi, \mu, \Lambda)]. \tag{28}
$$

Applying the product rule, we obtain the following equation:

$$
\begin{aligned}
\mathscr{L} = \{&\mathbb{E}[\ln p(X \mid Z, \mu, \Lambda)] + \mathbb{E}[\ln p(Z \mid \pi)] \\
&+ \mathbb{E}[\ln p(\pi)] + \mathbb{E}[\ln p(\mu, \Lambda)]\} \\
&-\{\mathbb{E}[\ln q(Z)] - \mathbb{E}[\ln q(\pi)] - \mathbb{E}[\ln q(\mu, \Lambda)]\},
\end{aligned}
\tag{29}
$$

and substituting the following expressions,

$$
\begin{aligned}
\mathbb{E}[\ln p(X \mid Z, \mu, \Lambda)] &= \frac{1}{2}\sum_{k=1}^{K} Y_k\left\{\ln\widetilde{\Lambda}_k - D\beta_k^{-1} - \nu_k Tr(S_k W_k) - \nu_k(\overline{x_k} - m_k)^T W_k(\overline{x_k} - m_k) - D\ln(2\pi)\right\}, \\
\mathbb{E}[\ln p(Z \mid \pi)] &= \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} ln\widetilde{\pi}_k, \\
\mathbb{E}[\ln p(\pi)] &= \ln C(\alpha_0) + (\alpha_0 - 1)\sum_{k=1}^{K}\ln\widetilde{\pi}_k, \\
\mathbb{E}[\ln p(\mu, \Lambda)] &= \frac{1}{2}\sum_{k=1}^{K}\left\{Dln\left(\frac{\beta_0}{2\pi}\right) + ln\widetilde{\Lambda}_k - \frac{D\beta_0}{\beta_k} - \beta_0\nu_k(m_k - m_0)^T W_k(m_k - m_0)\right\} \\
&\quad + K\ln B(W_0, \nu_0) + \frac{(\nu_0 - D - 1)}{2}\sum_{k=1}^{K} ln\widetilde{\Lambda}_k - \frac{1}{2}\sum_{k=1}^{K}\nu_k Tr\left(W_0^{-1}W_k\right), \\
\mathbb{E}[\ln q(Z)] &= \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\ln r_{nk}, \\
\mathbb{E}[\ln q(\pi)] &= \sum_{k=1}^{K}(\alpha_k - 1)\ln\widetilde{\pi}_k + \ln C(\alpha),
\end{aligned}
\tag{30}
$$

and

$$
\mathbb{E}[\ln q(\mu, \Lambda)] = \sum_{k=1}^{K}\left\{\frac{1}{2}\ln\widetilde{\Lambda}_k + \frac{D}{2}\ln\left(\frac{\beta_k}{2\pi}\right) - \frac{D}{2} - \mathscr{H}[q(\Lambda_k)]\right\}, \tag{31}
$$

where $\mathscr{H}[q(\Lambda_k)]$ is the entropy of the Wishart distribution. $\mathscr{L}$ then becomes the objective function to maximize and is given by the following equation:

$$\begin{aligned}\mathscr{L} = &\left[\frac{1}{2}\sum_{k=1}^{K} Y_k\left\{ln\tilde{\Lambda}_k - D\beta_k^{-1} - v_k Tr\left(S_k W_k\right) - v_k\left(\overline{x}_k - m_k\right)^T W_k\left(\overline{x}_k - m_k\right) - Dln\left(2\pi\right)\right\}\right]\\ &+\left[\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk} ln\check{\pi}_k\right] + \left[lnC\left(\alpha_0\right) + \left(\alpha_0 - 1\right)\sum_{k=1}^{K} ln\check{\pi}_k\right]\\ &+\left[\frac{1}{2}\sum_{k=1}^{K}\left\{Dln\left(\frac{\beta_0}{2\pi}\right) + ln\tilde{\Lambda}_k - \frac{D\beta_0}{\beta_k} - \beta_0 v_k\left(\left(m_k - m_0\right)^T W_k\left(m_k - m_0\right)\right)\right\}\right.\\ &\left.+ KlnB\left(W_0, v_0\right) + \left(\frac{v_0 - D - 1}{2}\right)\sum_{k=1}^{K} ln\tilde{\Lambda}_k - \frac{1}{2}\sum_{k=1}^{K} v_k Tr\left(W_0^{-1} W_k\right)\right]\\ &-\left[\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk} lnr_{nk}\right] - \left[\sum_{k=1}^{K}\left(\alpha_k - 1\right) ln\check{\pi}_k + lnC\left(\alpha\right)\right]\\ &-\left[\sum_{k=1}^{K}\left\{\frac{1}{2} ln\tilde{\Lambda}_k + \frac{D}{2} ln\left(\frac{\beta_k}{2\pi}\right) - \frac{D}{2} - \mathscr{H}\left[q\left(\Lambda_k\right)\right]\right\}\right],\end{aligned} \tag{32}$$

where

$$\begin{aligned}\ln\check{\pi}_k &= \mathbb{E}\left[\ln\pi_k\right]\\ B\left(W, v\right) &= \left|W\right|^{-v/2}\left\{2^{vD/2}\pi^{D(D-1)/4}\prod_{i=1}^{D}\Gamma\left(\frac{v + 1 - i}{2}\right)\right\}^{-1}\\ C\left(\alpha\right) &= \frac{\Gamma\left(\widehat{\alpha}\right)}{\Gamma\left(\alpha_1\right)\cdots\Gamma\left(\alpha_k\right)} \text{ and}\\ \mathscr{H}\left[\Lambda\right] &= -\ln B\left(W, v\right) - \frac{\left(v - D - 1\right)}{2}\mathbb{E}\left[\ln\left|\Lambda\right|\right] + \frac{vD}{2}.\end{aligned} \tag{33}$$

In this paper, we implement the maximization of equation (32) through the iterative updates of the GMM parameters mentioned earlier.

3.3. Variational Mixture of Gaussians (VMGs) Routing-Based Capsule Network. Motivated by [10, 11], and [4] based on the discussions in Sections 3.1 and 3.2, we let $\mathscr{L}_n$ and $\mathscr{L}_k$, respectively, represent capsules at the lower and higher-level layers. Let $X_{k|n} \in \mathbb{R}^{4x4}$ matrix represent the show of similarity between the features of a lower-level capsule n to a higher-level capsule k, with $x_{k|n} \in \mathbb{R}^D$ as its vectorized version (i.e. $x_{k|n}$ is a flattened vector of the matrix $X_{k|n}$ with $D = 16$). A higher-level capsule's pose matrix $M_k \in \mathbb{R}^{4x4}$ is flattened to obtain capsule k's pose vector $\mu_k \in \mathbb{R}^D$. For ease of computations, we use the precision matrix Λ_k instead of the covariance matrix Σ, and use $\lambda_k \in \mathbb{R}^D$ to represents the diagonal entries of Λ_k. As mentioned earlier, r_{nk} represents the vector form of the routing responsibilities while π_k is the mixing coefficient used for a single one-hot-vector representation $(1/k)$ necessary for indicating the choice of a cluster(capsule). On a larger scale, z is a latent variable that serves as a collection of one-hot-vectors with similar features signifying the preference of each lower-level capsule feature to a corresponding higher-level capsule Gaussian cluster of features. Finally, we compute the activation probability a_n to represent the likelihood that cluster k is activated by computing the ELBO (equation (32)) and paying a fixed cost of β_a as indicated in [4]. Based on the above-given discussions, we derive Algorithm 1 as the routing procedure between capsules.

3.4. Uncertainty Estimation. Aleatoric and epistemic uncertainties are common with neural network models. Randomness is a property that characterizes aleatoric uncertainty [21]. For this type of uncertainty, there is sufficient variability in the outcome of events as a result of a random phenomenon. Epistemic uncertainty, on the other hand, expresses the uncertainty resulting from the designer's lack of knowledge of the best design choices leading to the development of the best model. Both uncertainties together form the total uncertainty of the model. Several other methods exist for finding the total uncertainty of a model, but there is no consensus on which method is the best [25].

In this work, we experimentally determine the aleatoric and epistemic uncertainties of our model on some of the datasets. Since a deterministic model has no epistemic uncertainty [25], we determine its aleatoric uncertainty on the *in* and *out*-of-distribution data. For our Bayesian model, we determine both uncertainties.

4. Experiments

The experiments in this work were carried out using *PyTorch 1.7* GPU version on a 64 bit NVIDIA GeForce GTX 1060 Windows machine. Each model was trained for 100 epochs using a learning rate of 0.001, 3 routing iterations, and patience of 10,000. During training, the best model is saved to be used for inference. The code used in our

(1) **function** VMG ROUTING $(a_n, x_{k|n})$
(2) Initialize weights $\forall n$, k: $r_{nk} \leftarrow 1/\text{size}[\mathscr{L}_k]$
(3) Initialize priors $\forall k$: $\alpha_0, m_0, \beta_0, S_0, \nu_0$
(4) for i iterations do
(5) $r_{nk} \leftarrow r_{nk} \odot a_n$
(6) UPDATE BEST $q(\pi, \mu, \Lambda)$
(7) UPDATE BEST $q(z \mid x)$
(8) $a_k \leftarrow \mathscr{L}[q] + \beta_a - \beta_\mu$
(9) return a_k, m_k
(1) **function** UPDATE BEST $q(\pi, \mu, \Lambda)$
(2) $y_k \leftarrow \sum_n r_{nk}$
(3) $\alpha_k \leftarrow \alpha_0 + y_k$
(4) $\beta_k \leftarrow \beta_0 + y_k$
(5) $\nu_k \leftarrow \nu_0 + y_k$
(6) $\overline{x}_k \leftarrow 1/y_k \sum_{n=1}^{N} r_{nk} x_{k|n}$
(7) $S_k \leftarrow 1/y_k \sum_{n=1}^{N} r_{nk} (x_{k|n} - \overline{x}_k)(x_{k|n} - \overline{x}_k)^T$
(8) $m_k \leftarrow 1/\beta_k (\beta_0 m_0 + y_k \overline{x}_k)$
(9) $W_k^{-1} \leftarrow W_0^{-1} + y_k S_k + (\beta_0 y_k / \beta_0 + y_k)(\overline{x}_k - \mu_0)(\overline{x}_k - \mu_0)^T$
(10) $\ln|W_k| \leftarrow -2 * \text{spur}[\ln \text{Cholesky}(W_k^{-1})]$
(1) **function** UPDATE BEST $q(z)$
(2) $ln\tilde{\pi}_k \leftarrow \psi(\alpha_k) - \psi(\sum_{n=1}^{N} \alpha_n)$
(3) $\ln\tilde{\Lambda}_k \leftarrow \sum_{i=1}^{D} \psi(\nu_k + 1 - i/2) + D \ln 2 + \ln|W_k|$
(4) $\mathbb{E}[(x_n - \mu_k)^T \Lambda_k (x_n - \mu_k)] \leftarrow D\beta_k^{-1} + \nu_k (x_{k|n} - m_k)^T W_k (x_{k|n} - m_k)$
(5) $lnu_j \leftarrow \ln \tilde{\Lambda}_k - \mathbb{E}[(\mathrm{x_n} - \mu_\mathrm{k})^T \Lambda_\mathrm{k} (\mathrm{x_n} - \mu_\mathrm{k})]/2$
(6) $r_{nk} \leftarrow \text{softmax}(\mathbb{E}[\ln\tilde{\pi}_k] + \ln u_j)$

Algorithm 1: Variational mixture of Gaussians routing.

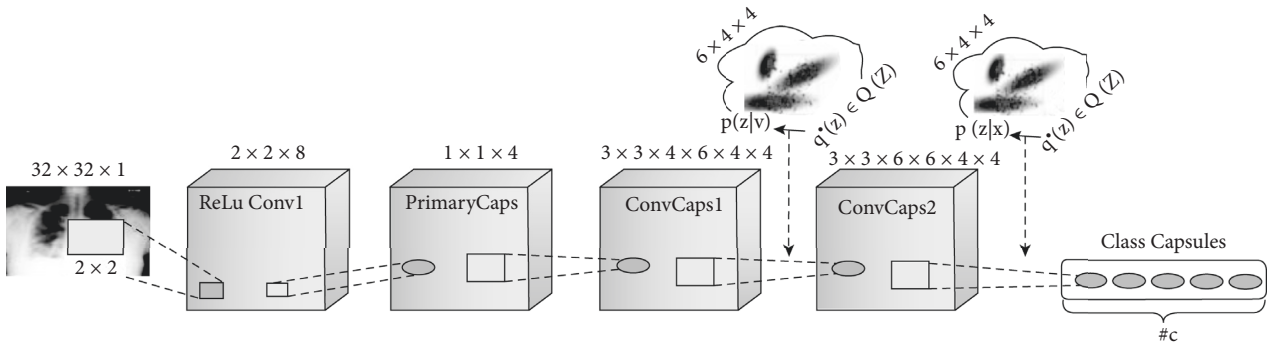

Figure 1: Architecture of the proposed VMG CapsNet model.

implementation is a modification of the code in [11], which can be found in [26].

4.1. Loss Function. We adopted the spread loss in [4] as well as the negative likelihood loss as used in [11].

4.2. Model Architecture. Our model begins with a 2×2-filter convolutional layer to perform convolutions on a $32 \times 32 \times 1$ input image with a stride of 2. This layer precedes three capsule layers and the ensuing VMG routing layers before the final class capsule layer which produces one capsule for each capsule class. Each capsule layer converts its respective filters into a 4×4 p_i capsule pose matrix and activation. The final layer broadcasts its weight matrices to produce a capsule p_4 per class for each category in the dataset. Taking the filter f and the capsule types p_i produced by each capsule layer into consideration, the network for the model can be represented as $[f, p_1, p_2, p_3, p_4]$. The complete architecture is shown in Figure 1.

4.3. Datasets and Data Preprocessing. Three popular computer vision datasets and one health-related dataset were adopted to experimentally evaluate the methods proposed in this paper. MNIST [27] is a handwritten dataset consisting of 70,000 28×28 grayscale images commonly partitioned into 60,000 training and 10,000 test sets. Comparatively, this dataset is less complex but effective and very popular for testing the performance of computer vision algorithms. Fashion-MNIST [28] is another dataset obtained from 70,000 greyscale fashion products. The original partition into training and test sets is similar to MNIST. This dataset is

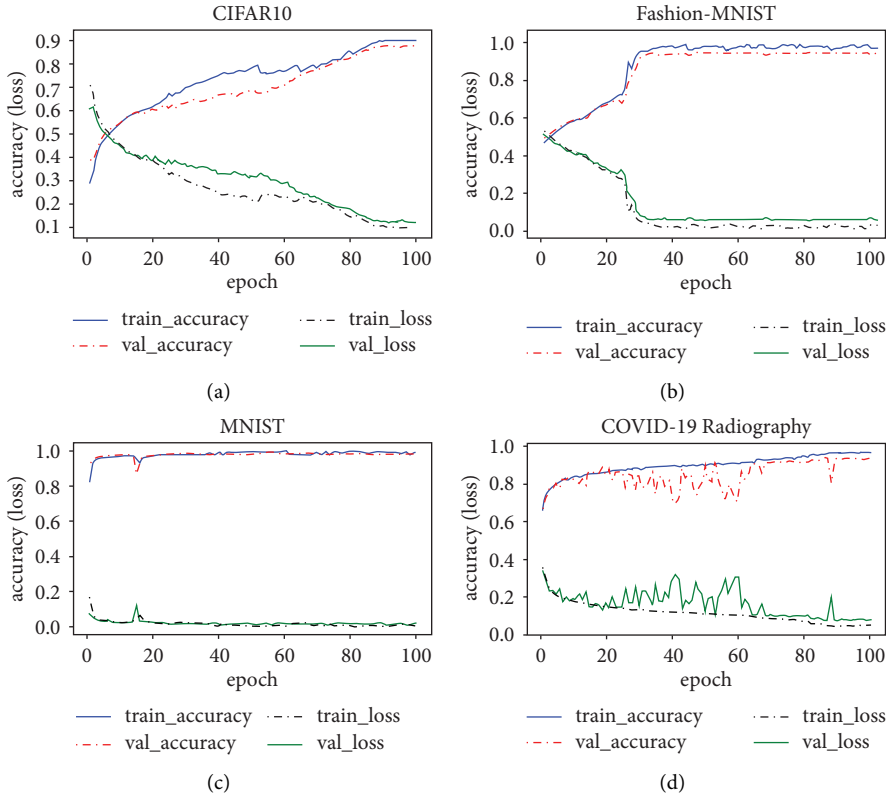

Figure 2: Training/validation accuracy/loss curves for the proposed model. The model learns and converges on (b) Fashion-MNIST and (c) MNIST as early as epoch 30. Learning takes time for (a) CIFAR-10 and (d) COVID-19 Radiography images but achieves convergence at epoch 90. We notice that the model converges faster for the images that are less complex compared to CIFAR-10 and COVID-19.

relatively complex to MNIST. The third and most complex dataset among the three is CIFAR-10 [29]. This dataset is very challenging to most computer vision algorithms due to the presence of background as well as background objects. Each of the aforementioned datasets is made up of ten classes and was partitioned into 55000 training, 5000 validation, and 10,000 test sets.

The fourth dataset is a COVID-19 Radiography dataset [30–32] collected from four countries by a team of doctors. It consists of three classes of infected chest X-ray images and one class of healthy X-rays. This dataset is highly imbalanced and for purposes of this work, was partitioned into 16,952 training, 2,000 validation, and 4,227 test images. Even though the performance of some machine vision algorithms largely depends on extensive preprocessing to obtain high informative image data, we did not employ any of these preprocessing algorithms irrespective of the fact that digital images contain Gaussian noise introduced by the limitations of the acquisition sensor/camera during image capturing. Fortunately, there are techniques to reduce its effect [33]. However, we evaluated the model on the raw images, enabling us to understand the actual extent to which the model can recognize real-life digital images (such as the COVID-19 images) without human interference.

4.4. Experimental Results. The results presented in this section are from the implementation of our model (Variational Mixture of Gaussians Routing model-VMG-Routing), the baseline Multilane LBP-Gabor Capsule (ML) network [32], and the VB-Routing [11] {64, 8, 16, 16, #c} architecture; where #c is the number of output classes. However, our GPU device could not run the higher architectures of the other VB-Routing models, consequently, for those models, we reported the results from the work in [11].

4.4.1. Model Learning and Convergence. The training and validation curves in Figure 2 show the proposed model's ability to learn and converge faster. For less complex images such as MNIST and Fashion-MNIST, the model converges as early as epoch 30. For relatively complex and imbalanced images such as CIFAR-10 and COVID-19 Radiography, the model attains an accuracy approximately equal to the final accuracy at epoch 90. Our VMG-Routing learns faster compared to the models in [11] which only show stability beginning from epoch 150. Fast learning and convergence are desirable attributes for image recognition systems applied in critical areas such as self-driving cars where every passing minute counts and is valuable.

Table 1 reports a comparison of the error rates of the VMG-Routing capsule and the other capsule network (CapsNet) models. Even with the moderate (shallow) size of the VMG-Routing model, it performs comparatively well with the deep and multilane models. The difference in

Table 1: Comparison of the error rates between the VMG-Routing model and some models from the literature. (∗) indicates the models that our device could not implement due to memory limitations. The values reported here were thus obtained from the literature. (–) indicates unavailable values. (#c) represents the number of classes in the dataset.

Algorithm	Error rate (%)			
	CIFAR-10	Fashion-MNIST	MNIST	COVID-19 radiography
VB-routing {64, 8, 16, 16, #*c*} [11]	13.10	5.46	0.99	8.01
VB-routing ∗ {64, 16, 32, 32, #*c*} [11]	11.2 ± .09	5.61	—	—
VB-routing ∗ {64, 16, 16, 16, #*c*} [11]	12.40	5.2 ± .07	—	—
EM-routing ∗ {64, 16, 16, 16, #*c*} [4]	—	6.14	—	—
EM-routing ∗ {64, 16, 32, 32, #*c*} [4]	11.2 ± .09	—	—	—
Multi-lane LBP-gabor capsule [32]	11.43	5.17	1.00	8.04
Dynamic routing [3]	35.57	22.45	0.25	8.09
VMG-routing {32, 4, 8, 8, #*c*} (**ours**)	12.19	5.38	1.00	7.01

Table 2: Comparison of the number of parameters generated by each model. The VMG-Routing model produced the least number of parameters with the ML producing the largest number of parameters. (∗) indicates the models that our device could not implement due to memory limitations. The values reported here were thus obtained from the literature. (–) indicates unavailable values. (#c) represents the number of classes in the dataset.

Algorithm	Number of parameters			
	CIFAR-10	Fashion-MNIST	MNIST	COVID-19 radiography
VB-routing {64, 8, 16, 16, #*c*}	145 K	145 K	145 K	120 K
VB-routing ∗ {64, 16, 32, 32, #*c*}	323 K	323 K	—	—
VB-routing ∗ {64, 16, 16, 16, #*c*}	172 K	172 K	—	—
EM-routing ∗ {64, 16, 16, 16, #*c*}	—	323 K	—	—
EM-routing ∗ {64, 16, 32, 32, #*c*}	323 K	—	—	—
Multi-lane LBP-gabor capsule	4.10 M	4.10 M	4.10 M	3.70 M
Dynamic routing	9.3 M	8.2 M	8.2 M	9.8 M
VMG-routing {32, 4, 8, 8, #*c*} (**ours**)	14 K	15.5 K	14 K	10.2 K

Table 3: Results of testing on 10,000 sample images of MNIST, CIFAR-10, and Fashion-MNIST dataset. 4,227 samples were used for testing the models on the COVID-19 Radiography dataset.

Algorithm	MNIST (%)	CIFAR-10 (%)	Fashion-MNIST (%)	COVID-19 Radiography	Average time
VB-routing {64, 8, 16, 16, #*c*}	98.53	85.06	92.97	90.92%	30 s, 14 ms
Multi-lane LBP-gabor capsule	98.89	86.97	94.00	91.09%	18 s, 25 ms
Dynamic routing	99.21	65.11	76.32	90.02%	20 s, 11 ms
VMG-routing {32, 4, 8, 8, #*c*} (**ours**)	98.96	86.82	93.71	92.15%	15 s, 23 ms

accuracy on CIFAR-10 between the proposed VMG-Routing CapsNet and the largest model is only 1.07% with our model having an added advantage of being less computationally complex.

4.4.2. Model Complexity. The VMG-Routing CapNet produced fewer parameters compared to its counterparts in the literature as can be seen in Table 2. This makes the VMG-Routing model less computationally complex and increases its potential for implementation on embedded and mobile devices that naturally have limited memory. In addition, model complexity poses a threat of overfitting [34] that ultimately leads to poor performance.

4.4.3. Inference. To test the models' generalizability on unseen images, we used the trained (saved) models to perform inference, respectively, on 10,000 and 4,227 sample images from MNIST, CIFAR-10, Fashion-MNIST, and the COVID-19 Radiography datasets. A comparison of the test accuracies is reported in Table 3. The average time for each model to perform inference on the sample images is also reported in Table 3. It can be observed that the VMG-Routing model produced results that compare favorably well with the results of other state-of-the-art models.

We further performed inference on individual in-distribution images for both models to determine the level of confidence/certainty each model places on its prediction probabilities. Figure 3 shows that the deterministic model is overconfident in its predictions (column 3) while the VMG-Routing CapsNet exercises some caution in the confidence it imposes on its predictions (column 2).

4.4.4. Model Uncertainty. Daily scenarios involve decision-making influenced by the level of uncertainties/certainties prevailing at the time. Depending on the field under consideration, uncertainty estimation can be a critical part of the decision-making process. For instance, the reliability and efficacy of a deep learning model for medical applications

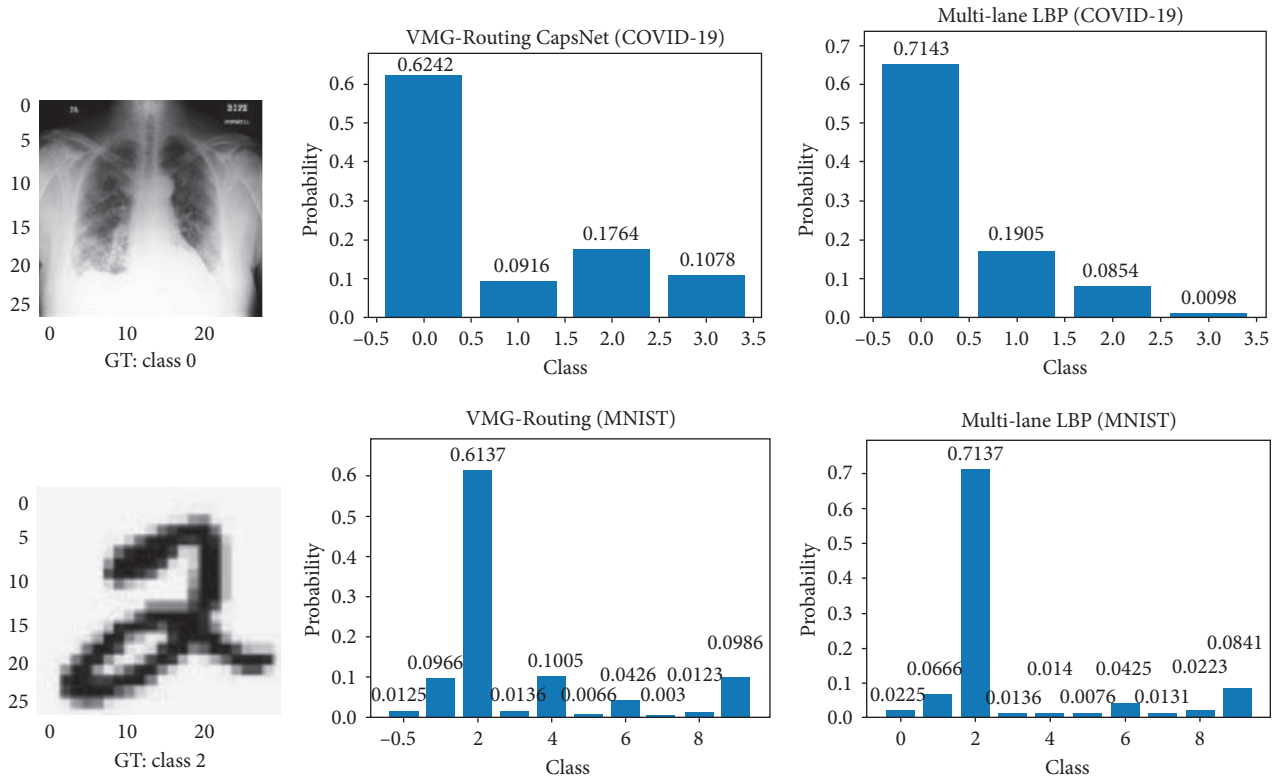

FIGURE 3: Comparison of the prediction probabilities of the VMG-Routing CapsNet and the Multi-lane LBP-Gabor Capsule model on in-distribution COVID-19 and MNIST test images. These results express the certainty of each model in its prediction of the test image. Notice that the deterministic model produces higher probabilities as a way of expressing overconfidence.

such as Artificial Intelligence (AI) assisted surgery depends on the uncertainty with which it identifies the medical condition correctly. Bayesian methods have advantages over other neural networks as they provide the avenue to effectively model uncertainty [12]. The inability of machine learning applications to provide reliable uncertainty estimates is a potential limiting factor in their acceptability and widespread adoption for critical tasks.

To demonstrate the reliability of the uncertainty estimates of our VMG-Routing model, we present a comparison of experimental results from the prediction of both in-distribution (Figure 4) and out-of-distribution (Figure 5) images for the VMG-Routing model and the baseline deterministic ML-LBP capsule model.

We use p_{out} to express the aleatoric uncertainty shown by the distribution across the classes for the deterministic model. This uncertainty assumes a value of zero if a class gets a probability of one and all other classes obtain a probability of zero. Since deterministic CapsNets have fixed weights, they cannot express epistemic uncertainties [25] and will produce the same output when inference is carried on the same input image N times. The output of the SoftMax layer p_{out} (see Figure 3) sums up to one and measures the certainty (certainty = p_{out}) of the model in its predictions. We obtain the aleatoric uncertainty of the deterministic CapsNet from the same quantity p_{out} by computing the negative log likelihood (NLL) or the entropy of the predictions.

$$\begin{aligned}\text{entropy} &= -\sum_{i=0}^{\#c} p_i \log(p_i),\\ \text{aleatoric uncertainty} &= NLL\\ &= -\log(p_{\text{out}}),\end{aligned} \tag{34}$$

where $0 \le i < \#c$ and $\#c$ is the number of classes in the dataset under consideration.

On the other hand, our VMG-Routing CapsNet replaces the fixed weights with Gaussian distributions giving it the ability to express both epistemic and aleatoric uncertainties in its predictions. The aleatoric uncertainty is expressed in the distributions similar to the deterministic CapsNets, except that it is based on average prediction probabilities. Meanwhile, the epistemic uncertainty is measured in the spread of the inference probabilities and is zero for a zero spread. For this scenario, N different multinomial conditional probability distribution $p(\hat{y} \mid x, w_n)$ conditioned on the weight distribution w_n are obtained out of N predictions on the same input image. The mean probability p^*_{out} is computed for each class i and the maximum mean conditional probability is chosen as the predicted class of the input image.

$$p^*_{\text{out}} = \frac{1}{N}\sum_{i=0}^{N-1} p_{\text{out}_i}, \tag{35}$$

and

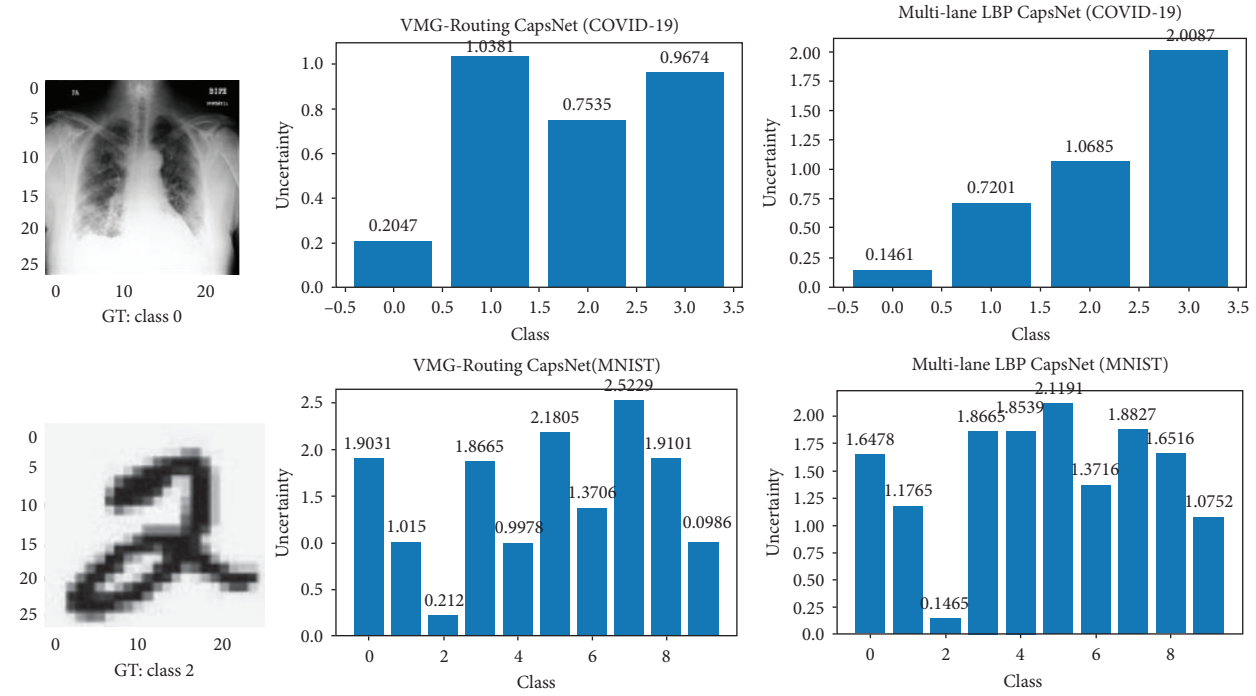

Figure 4: Comparison of the uncertainty between the VMG-Routing and Multi-lane CapsNets on in-distribution test images. The VMG-Routing CapsNet shows uncertainty over the ML model on the MNIST dataset. On the COVID-19 dataset, the uncertainty for both models is approximately the same.

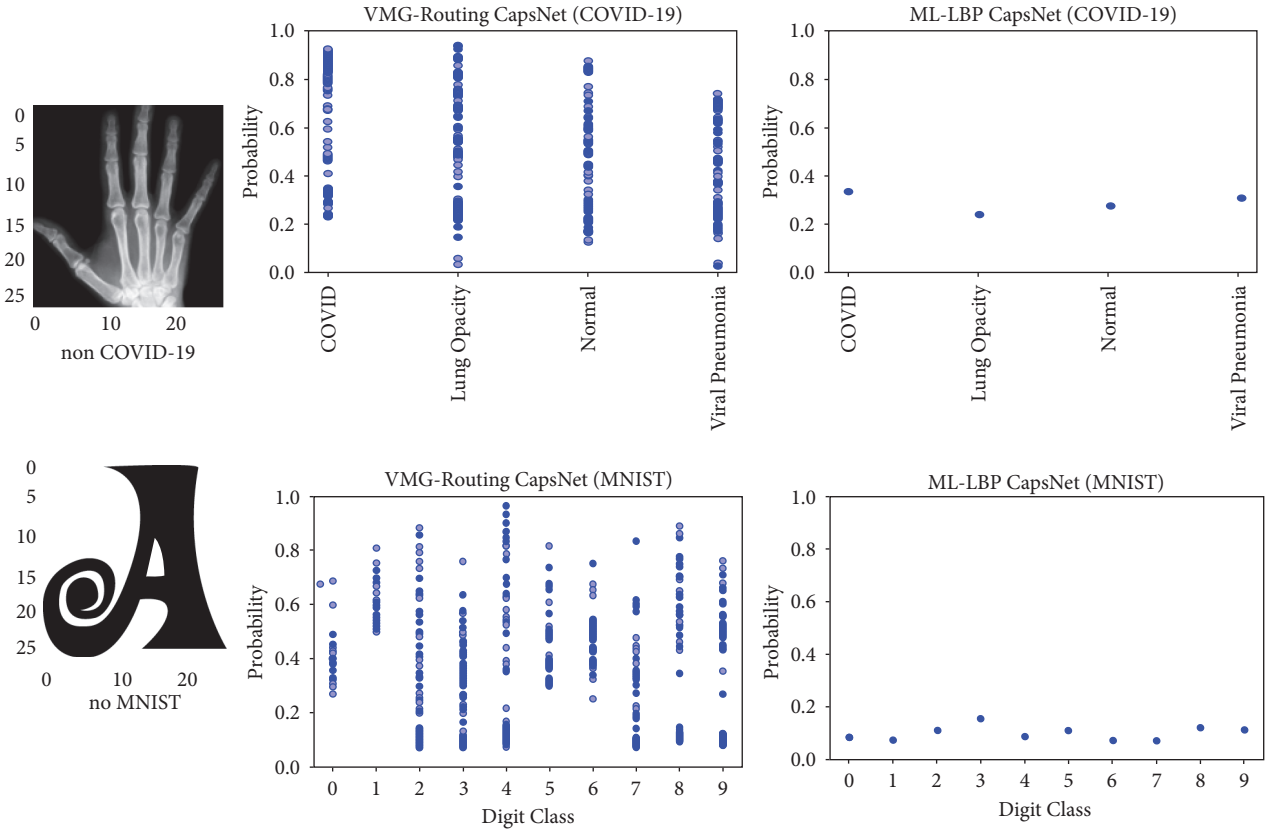

Figure 5: Uncertainty plots for (column 2) the VMG CapsNet on the out-of-distribution images, and (column 3) the deterministic Multilane LBP CapsNet on the same two images. The spread of the VMG CapsNet predictions demonstrates its high epistemic uncertainty on the images. On the other hand, the Multi-lane LBP CapsNet shows false confidence in its predictions as shown by the consistency with which it produces the same probabilities for the $N = 100$ predictive runs.

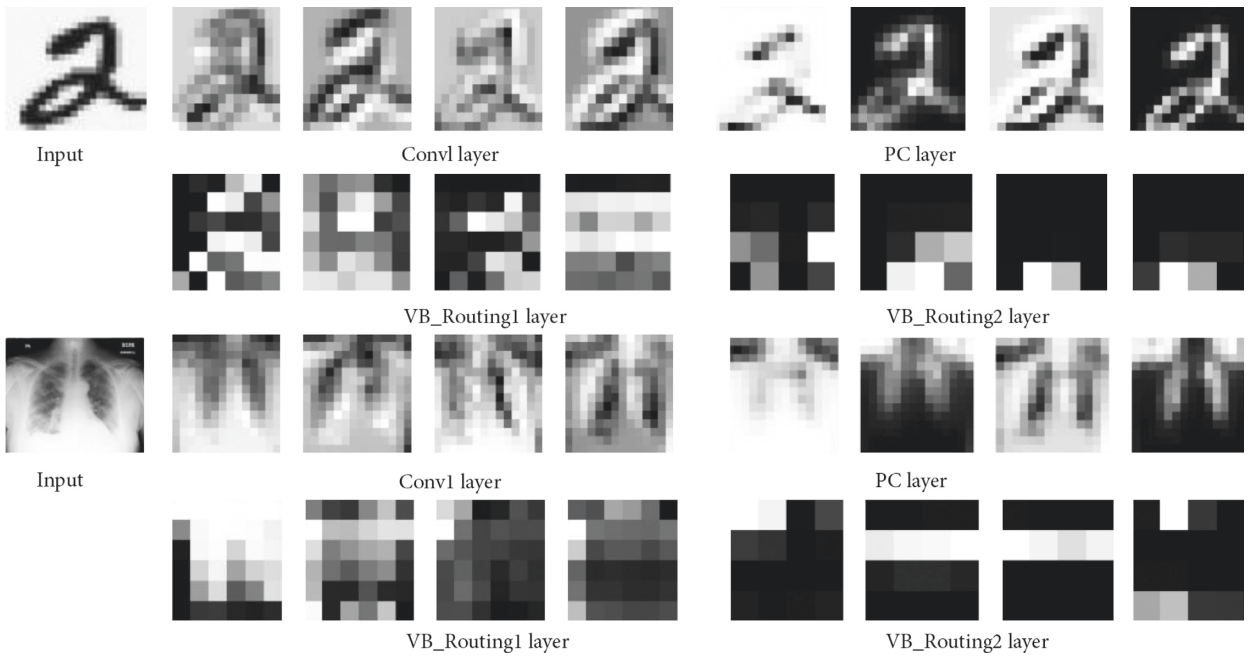

Figure 6: Visualization of the feature maps for the layers in the VMG-Routing model. The first two rows show the MNIST input image and the outputs of each layer. The second two rows show the input from a COVID-19 Radiography image and the corresponding outputs of the filters for each layer.

$$p^*_{\text{inf}} = \max\left(p^*_{\text{out}}\right). \tag{36}$$

The averaging in the measure in equation (35) ensures that the epistemic uncertainty in the model is captured. Subsequently, $NLL^* = -\log\left(p^*_{\text{inf}}\right)$ is possible to compute. In addition, the uncertainty based on the entropy and total variance obtained from the averaging naturally follows from the following expressions:

$$\begin{aligned} \text{entropy}^* &= -\sum_{i=0}^{\#c} p^*_{\text{out}} \log\left(p^*_{\text{out}}\right), \\ \sigma_T^2 &= \sum_{i=0}^{\#c} \sigma^2\left(p_i\right) = \sum_{i=0}^{\#c} \frac{1}{N} \sum_{n=0}^{N-1} \left(p_{in} - p_i^*\right)^2. \end{aligned} \tag{37}$$

Figure 5 shows the uncertainty of both models on the respective out-of-distribution images. The spread of the prediction probabilities of a given class expresses the epistemic uncertainty while the distribution across the different classes epitomizes the aleatoric uncertainty of the models [25].

Even though both models produce wrong predictions for the out-of-distribution images, the VMG-Routing CapsNet produces predictive probabilities (Figure 5, column 2) that significantly vary in the distribution and spread of the $N = 100$ predictive runs. The VMG-Routing CapsNet, therefore, can express both uncertainties. On the contrary, the deterministic model cannot express epistemic uncertainty since performing $N = 100$ predictive runs on the same input image produces the same probabilities (Figure 5, column 3). The ability of a model to express its uncertainty is a desirable property since it can be shown that models that produce higher uncertainties are likely to produce accurate predictions [25]. Finally, the shape of the VMG-Routing CapsNet's predictive probability distribution has some semblance to that of the Gaussian distribution which may be attributed to the model being driven by a variational mixture of Gaussians.

4.4.5. Model's Ability to Extract Relevant Features. To enable us to understand and tune the VMG-Routing model for further performance improvement, we investigated the ability of the layers in the model to extract the relevant features. Through experimentation via this approach, redundant layers were eliminated, resulting in a reduction in the model size/complexity, convergence time, and excessive oscillations during training. More specifically, we visualized the output (feature maps) of the layers by feeding an input image into the trained (best saved) model. The feature maps for the various layers are shown in Figure 6. It can be observed that the layers of the model can extract the most relevant features from the input images.

4.4.6. Threats to Validity. Deep Learning (DL) is capable of learning and modeling real-life scenarios when extreme care is taken, during the design and development stages, to consider all the factors that have the potential to prevent the model from achieving optimal performance. For instance, the choice of hyperparameters and their values is an important exercise that has a direct impact on the validity of the model outputs. For stochastic gradient descent (SGD)-based methods and their variants, a fraction of the dataset used for training are organized into batches whose size is relevant to the computation of the gradient. Practically, larger batch sizes reduce the quality of the model during generalization

[35]. This work, therefore, sampled from 16–32 data points for the experiments as batch sizes. We also avoided the sorting of the dataset and introduced randomization of batches in a bid to prevent the possibility that a given batch will have the same labels. In addition, the learning rate controls the rate at which the model should be modified in response to the error anytime there is an update in the model weights. We chose a smaller learning rate to allow the model to learn the optimal set of weights even though this has the potential to increase training time and the risk of overfitting. Other methods for solving this include implementing a learning rate decay function which returns an updated learning rate value that drops by half every n number of epochs. Furthermore, nonlinear activation functions are useful for DL to effectively model real-life scenarios which are nonlinear. The choice of the appropriate activation function determines the speed of computations necessary to speed up the training process as well as the ability to reduce the likelihood of generating vanishing gradients and improve performance [36]. To introduce nonlinearity and activate the capsule, we adopted the Sigmoid activation function since it encourages unambiguous predictions with 1 or 0, plus the fact that it can return a value between 0 and 1 when used with $(-\infty, +\infty)$.

Another scenario that poses a threat to the validity of the Bayesian model outputs is the covariate shift, where the distributions of training and target data are different [37]. Covariate shift may also occur due to pixelate-corrupted test data, spurious correlations, and domain shift. This problem is well pronounced with Bayesian models that make use of unconstrained Λ (covariance matrix) and is worsened when there exists linear independence in the features. In this work, we employed mean-field variational inference (MFVI) which constraints the Λ to be a diagonal matrix, limiting the effect of linear dependence in the features [38] and hence the impact of covariate shift.

5. Conclusion and Future Work

In this work, we proposed a capsule network based on a variational mixture of Gaussian routing to express the uncertainties associated with performing predictions on out-of-distribution data. The results show that a Bayesian capsule can be less computationally complex, converge faster, and outperform both the state-of-the-art deterministic and probabilistic models during inference. Furthermore, our work demonstrates that Bayesian capsules may have advantages over their deterministic counterparts since they have a bigger potential to exhibit transparency, credibility, reliability, and interpretability required to gain the confidence of industry players.

In the future, we intend to carry out a full investigation into Bayesian capsule interpretability in a quest to unravel the "black box" concept.

Data Availability

The data used to support the findings of this study can be accessed in the following repositories: 1. http://yann.lecun.com/exdb/mnist/ 2. https://www.cs.toronto.edu/~kriz/cifar.html 3. https://www.kaggle.com/datasets/zalando-research/fashionmnist 4. https://www.kaggle.com/datasets/preetviradiya/covid19-radiography-dataset.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Z. Bai, Y. Li, M. Woźniak, M. Zhou, and D. Li, "DecomVQANet: decomposing visual question answering deep network via tensor decomposition and regression," *Pattern Recognition*, vol. 110, Article ID 107538, 2021.

[2] P. Mensah Kwabena, B. A. Weyori, and A. Abra Mighty, "Exploring the performance of LBP-capsule networks with K-Means routing on complex images," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 2574–2588, 2022.

[3] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, vol. 2017, pp. 3857–3867, 2017.

[4] G. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with em routing," *ICLR*, pp. 1–15, 2018.

[5] A. F. M. Saif, C. Shahnaz, W. P. Zhu, and M. O. Ahmad, "Abnormality detection in musculoskeletal radiographs using capsule network," *IEEE Access*, vol. 7, pp. 81494–81503, 2019.

[6] K. R. Kruthika, Rajeswari, and H. D. Maheshappa, "CBIR system using Capsule Networks and 3D CNN for Alzheimer's disease diagnosis," *Informatics in Medicine Unlocked*, vol. 14, pp. 59–68, 2019.

[7] R. V. Kurup, M. A. Anupama, R. Vinayakumar, and V. Sowmya, "Capsule network for plant disease and plant species classification," *ICCVBIC -Advances in Intelligent Systems and Computing*, vol. 186, pp. 413–421, 2019.

[8] P. K. Mensah, B. A. Weyori, and M. A. Ayidzoe, "Capsule network with K-Means routing for plant disease recognition," *Journal of Intelligent and Fuzzy Systems*, vol. 40, no. 1, pp. 1025–1036, 2021.

[9] J. Gawlikowski, "A survey of uncertainty in deep neural networks," pp. 1–41, 2021, https://arxiv.org/pdf/2107.03342.pdf.

[10] M. Christopher, *Pattern Recognition and Machine Learning*, Springer Science+Business Media, LLC, Cambridge, UK, 2006.

[11] F. De Sousa Ribeiro, G. Leontidis, and S. Kollias, "Capsule routing via variational bayes," in *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, pp. 3749–3756, 2020.

[12] T. Pearce, F. Leibfried, A. Brintrup, M. Zaki, and A. Neely, "Uncertainty in neural networks: approximately bayesian ensembling," *23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 108, 2020.

[13] C. M. Bishop, "Bayesian neural networks," *Journal of the Brazilian Computer Society*, vol. 4, no. 1, pp. 361–370, 1997.

[14] L. Smith, L. Schut, Y. Gal, and M. Van Der Wilk, "Capsule networks—a probabilistic perspective," 2021, https://arxiv.org/abs/2004.03553, Article ID 03553v3.

[15] Y. Guo, G. Camporese, W. Yang, A. Sperduti, and L. Ballan, "Conditional variational capsule network for open set recognition," in *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision*, pp. 103–111, Iccv, Montreal, BC, Canada, October 2021.

[16] Q. Hua, L. Wei, C. Dong, and F. Zhang, "Improved variational inference with dynamic routing flow," *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 2, pp. 301–312, Article ID 0123456789, 2019.

[17] F. D. S. Ribeiro, "Introducing routing uncertainty in capsule networks," in *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, NeurIPS, Vancouver, Canada, December 2020.

[18] X. Chu, N. Xu, X. Liu, and X. Yao, "Research on capsule network optimization structure by variable route planning," in *Proceedings of the 2019 IEEE International Conference on Real-time Computing and Robotics*, pp. 858–861, Irkutsk, Russia, August 2019.

[19] H. RaviPrakash, S. M. Anwar, and U. Bagci, "Variational capsule encoder," *Proc. - Int. Conf. Pattern Recognit.*, pp. 5820–5827, 2020.

[20] H. Huang, L. Song, R. He, Z. Sun, and T. Tan, "Variational capsules for image analysis and synthesis," pp. 1–10, 2018, https://arxiv.org/abs/1807.04099.

[21] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods," *Machine Learning*, vol. 110, no. 3, pp. 457–506, 2021.

[22] S. Depeweg, J. M. Hernández-Lobato, S. Udluft, and T. Runkler, "Sensitivity analysis for predictive uncertainty in Bayesian neural networks," in *ESANN 2018 - Proceedings, European Symposium on Artificial Neural Networks*, pp. 279–284, Computational Intelligence and Machine Learning, 2018.

[23] A. Ganguly and S. W. F. Earp, "An introduction to variational inference," 2021, https://arxiv.org/abs/2108.13083.

[24] A. Kushwaha, *Variational Inference: Gaussian Mixture Model*, https://ashkush.medium.com/variational-inference-gaussian-mixture-model-52595074247b, 2020.

[25] O. Durr, B. Sick, and E. Murina, *Probabilistic Deep Learning with Python*, Manning Publications Co, NY, USA, 2020.

[26] F. D. S. Ribeiro, "Variational capsule routing," GitHub, 2020, https://github.com/fabio-deep/Variational-Capsule-Routing.

[27] Y. LeCun, C. Cortes, and C. J. C. Burges, "The MNIST database of handwritten digits," 2012, http://yann.lecun.com/exdb/mnist/2012.

[28] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," pp. 1–6, 2017, https://arxiv.org/abs/1708.07747.

[29] A. Krizhevsky and G. Hinton, *Learning Multiple Layers of Features from Tiny Images*, 2009.

[30] M. E. H. Chowdhury, T. Rahman, A. Khandakar et al., "Can AI help in screening viral and COVID-19 pneumonia?" *IEEE Access*, vol. 8, pp. 132665–132676, 2020.

[31] T. Rahman, A. Khandakar, Y. Qiblawey 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, Article ID 104319, 2021.

[32] P. K. Mensah, A. A. Amponsah, K. B. Agyemang et al., "Multi-lane LBP-gabor capsule network with K-means routing for medical image analysis," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 10, pp. 282–294, 2021.

[33] W. Dong, M. Wozniak, J. Wu, W. Li, and Z. Bai, "De-noising aggregation of graph neural networks by using principal component analysis," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, 2022.

[34] C. Zhang, B. Recht, S. Bengio, M. Hardt, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *Proceedings of the 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, Toulon, France, April 2017.

[35] N. S. Keskar, J. Nocedal, P. T. P. Tang, D. Mudigere, and M. Smelyanskiy, "On large-batch training for deep learning: generalization gap and sharp minima," in *Proceedings of the 5th International Conference on Learning Representations ICLR 2017 - Conference Track Proceedings*, pp. 1–16, Toulon, France, April 2017.

[36] B. Gagana, H. A. U. Athri, and S. Natarajan, "Activation function optimizations for capsule networks," in *Proceedings of the 2018 International Conference on Advances in Computing, Communications and Informatics ICACCI 2018*, no. 3, pp. 1172–1178, Bangalore, India, September 2018.

[37] M. Arjovsky, "Out of distribution generalization in machine learning," 2021, https://arxiv.org/abs/2103.02667.

[38] P. Izmailov, P. Nicholson, S. Lotfi, and A. G. Wilson, "Dangers of bayesian model averaging under covariate shift," in *Advances in Neural Information Processing Systems*, vol. 5, pp. 3309–3322, NeurIPS, 2021.