Research Article

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

Patrick Kwabena Mensah, Mighty Abra Ayidzoe, Alex Akwasi Opoku, Kwabena Adu, Benjamin Asubam Weyori, Isaac Kofi Nti, and Peter Nimbe

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

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 \( p(\cdot | x) \) 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 \( \theta \) of a distribution on the weights \( p(w|\theta) \) 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...
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 $p(x, z) = p(x | z)p(z)$, $z$ can be marginalized out to give $p(x) = \sum_{z=1}^{K} p(x, z) = \sum_{z=1}^{K} p(z)p(x | z)$. 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):

$$p(x) = \sum_{k=1}^{K} \pi_{x}N(x | \mu_k, \Sigma_k).$$

The Gaussian density (also called component) in the above expression has its own mean $\mu_k$ and covariance $\Sigma_k$. 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:

$$p(\theta | x) = \frac{p(x | \theta)p(\theta)}{p(x)}$$

$$= \frac{p(x | \theta)p(\theta)}{\int p(x, \theta)d\theta}$$

where $\int p(x, \theta)d\theta$ 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 $p(\cdot | x)$. The distance between the variational ("nice") distribution $q$ and the true posterior $p(\cdot | x)$ is measured by the Kullback–Leibler (KL) divergence.

$$KL[q\|p(\cdot | x)] = \int q(\theta)\log\frac{q(\theta)}{p(\theta | x)}d\theta$$

$$= \mathbb{E}_q\log\frac{q(\theta)}{p(\theta | x)}$$

$$= \log p(x) - \int q(\theta)\log\frac{p(x, \theta)}{q(\theta)}d\theta.$$
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.

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

(5)

This mean-field (MF) approximation makes it possible to obtain a free-form optimization of the ELBO \( \mathcal{L}[q] \) with respect to all the distributions \( q_i(\theta_i) \) by optimizing each of the factors in turn. When the \( \mathcal{L}[q] \) 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 \( \mathcal{L}[q] \).

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 \( RD \)-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., \( H_k \sim N(\mu_k, \Delta_k) \), where \( k = 1, \ldots, K, \mu_k \) is the \( 1 \times D \) mean-vector and \( \Delta_k \) the \( D \times D \) covariance matrix. In what follows, \( f_k \) will be written for the density of \( N(\mu_k, \Delta_k) \). 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 \( N(\mu_k, \Delta_k) \), for some \( k = 1, \ldots, K \). 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 \( \pi_n \) being the probability \( Z_nk = 1 \), for some \( k = 1, \ldots, K \). Therefore, \( \pi = (\pi_1, \ldots, \pi_K) \), called the vector of mixing coefficients, is a probability vector and \( N = (N_1, \ldots, N_K) = Z_1 + Z_2 + \ldots + Z_N \) is a random vector with \( K \) non-negative coordinates that sum up to \( N \). In fact, \( Y \) is multinominally distributed with parameters \( N \) and \( \pi \). Observe that for any \( n = 1, \ldots, N \), the probability that \( Z_nk = 1 \) is given by the following equation:

\[
p(z_nk) = \prod_{k=1}^{K} \pi_k^{z_nk}. 
\]

(6)

Putting \( \theta = (Z, \pi, (\mu, \Lambda)) \), with \( Z = (Z_1, \ldots, Z_N) \), \( \pi = (\pi_1, \mu_1, \mu_2, \ldots, \mu_K) \), and \( \Lambda = (\Lambda_1, \Lambda_2, \ldots, \Lambda_K) \), the joint distribution of \( X \) and \( \theta \) can be written as follows:

\[
p(X, \theta) = p(X | Z, \pi, \Lambda)p(Z | \pi, \mu, \Lambda)p(\pi | \mu, \Lambda)p(\mu | \Lambda)p(\Lambda)
\]

\[
\quad = p(X | Z, \mu, \Lambda)p(Z | \pi)p(\pi | \mu)p(\mu | \Lambda)p(\Lambda).
\]

(7)

The second equality of equation (7) uses that \( p(Z | \pi, \mu, \Lambda) = p(Z | \pi) \) and \( p(\pi | \mu, \pi) = p(\pi) \). We assume further that conditioning on \( \theta \), the components of \( X \) are independent. Similarly, given \( \pi \) and \( \Lambda \), the components of \( Z \) and \( \mu \) are respectively independent. Furthermore, the components of \( \Lambda \) 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 \( \Lambda \) and \( \pi \), and the above-given result in

\[
p \sim \text{SymDir}(K, \alpha_0), \quad \Lambda_k \sim \text{Wi}(W, v_0), \quad 
\mu_k \sim N((\mu_0), (\beta_0\Lambda_k)^{-1}), \quad \pi \sim \text{Mult}(N, \pi), \quad X_{i=1}\ldots N \sim N(\mu_k, \Lambda_k^{-1}), 
\]

Therefore,

\[
p(x | z, \mu, \Lambda) = \prod_{n=1}^{N} \prod_{k=1}^{K} f_k(x_n^{z_nk}), 
\]

(9)

\[
p(z | \pi) = \prod_{n=1}^{N} \left( \prod_{k=1}^{K} (\pi_k)^{z_nk} \right), 
\]

(10)

\[
p(\pi) = \frac{\Gamma(K\alpha_0)}{\Gamma(\alpha_0)^K} \prod_{k=1}^{K} \pi_k^{\alpha_0 - 1}, 
\]

(11)

\[
p(\mu | \Lambda) = \prod_{k=1}^{K} f_k^0(\mu_k), 
\]

(12)

\[
p(\Lambda) = \prod_{k=1}^{K} W(\Lambda_k), 
\]

(13)

where

\[
f_k(x_n) = \frac{1}{2\pi^{D/2} |\Lambda_k|^{1/2}} \exp \left\{ -\frac{1}{2} (x_n - \mu_k)^\top \Lambda_k (x_n - \mu_k) \right\}, 
\]

\[
f_k^0(\mu_k) = \frac{1}{2\pi^{D/2} |(\beta_0\Lambda_k)^{-1}|^{1/2}} \exp \left\{ -\frac{1}{2} (\mu_k - \mu)\top \beta_0 \Lambda_k (\mu_k - \mu) \right\}, 
\]

\[
W(\Lambda_k) = B(W_0, v_0) |\Lambda_k|^{(v_0-D)/2} \exp \left\{ -\frac{1}{2} Tr(W_0 \Lambda_k) \right\}, 
\]

\[
B(W_0, v_0) = |W_0|^{-D/2} 2^{v_0-D/2} \prod_{n=1}^{D} \frac{\Gamma\left(\frac{v_n+1}{2}\right)}{\Gamma\left(\frac{v_n}{2}\right)}, 
\]

(14)

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

(15)

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

\[
\log q^* (z|x) = E_{q(\pi, \mu, \Lambda)}[\log(p(x|z, \mu, \Lambda)p(z|\pi))] + \text{const}
\]

(16)

Substituting (9) and (10) into the expression for \(\log q^* (z|x)\), produces

\[
\log q^* (z|x) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \log(\rho_{nk}(x_n)) + \text{const},
\]

(17)

where

\[
\log \rho_{nk}(x_n) = E_{q(\pi)}[\log(\pi_k)] - \frac{1}{2} E_{q(\pi)}[\log(|\Lambda_k|^{1/2})] - \frac{D}{2} \log(2\pi) - \frac{1}{2} E_{q(\pi, \mu_k, \Lambda_k)}[(x_n - \mu_k)^T \Lambda_k (x_n - \mu_k)].
\]

(18)

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

\[
q^*(z|x) = \prod_{n=1}^{N} \prod_{k=1}^{K} r_{nk}^{znk}(x_n),
\]

(19)

where

\[
r_{nk}(x_n) = \frac{\rho_{nk}(x_n)}{\sum_{j=1}^{K} \rho_{nj}(x_n)}.
\]

(20)

The best \(q^* (z|x)\), 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

\[
\log q^*(\pi) = \log p(\pi) + E_{q(\pi)}[\log p(z|x, \pi)] + \text{const}
\]

\[
= (\alpha_0 - 1) \sum_{k=1}^{K} \ln \pi_k + \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \ln \pi_k + \text{const}
\]

\[
= (\alpha_0 - 1) \sum_{k=1}^{K} \ln \pi_k + \sum_{k=1}^{K} y_k \ln \pi_k + \text{const}.
\]

(21)

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

\[
q^*(\pi) = \text{Dir}(K, \alpha_1, \ldots, \alpha_K)
\]

\[
= C(\alpha_1, \ldots, \alpha_K) \prod_{k=1}^{K} \pi_k^{\alpha_k - 1},
\]

(22)

where

\[
\alpha_k = \alpha_0 + y_k \quad \text{and}
\]

\[
y_k = \sum_{n=1}^{N} z_{nk}.
\]

(23)

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

\[
\log q^*(\mu, \Lambda) = \log p(\mu, \Lambda) + E_{q(\pi)}[\log(p(x|z, \mu, \Lambda))] + \text{const}
\]

\[
= \sum_{k=1}^{K} \left[ \log(f_k^0(\mu_k)) + \log(W_i(A_k)) \right] + \text{const},
\]

(24)

where \(f_k^0\) and \(W_i\) are respectively the Gaussian and Wishart densities (see equations (12) and (13)) with parameters \(m_k, \beta_k, W_k, \) and \(v_k\). These parameters are given as follows:

\[
\beta_k = \beta_0 + y_k,
\]

\[
m_k = \frac{1}{\beta_k} (\beta_0 \mu_0 + y_k \mathbf{x}_k),
\]

\[
W_k^{-1} = W_0^{-1} + y_k S_k + \frac{\beta_0 y_k}{\beta_0 + y_k} (\mathbf{x}_k - \mu_0)(\mathbf{x}_k - \mu_0)^T,
\]

\[
v_k = v_0 + y_k,
\]

(25)

\[
y_k = \sum_{n=1}^{N} z_{nk},
\]

\[
\mathbf{x}_k = \frac{1}{y_k} \sum_{n=1}^{N} z_{nk} x_n,
\]

\[
S_k = \frac{1}{y_k} \sum_{n=1}^{N} z_{nk}(x_n - \mathbf{x}_k)(x_n - \mathbf{x}_k)^T.
\]

(26)

To evaluate \(r_{nk}\), the quantities in \(\rho_{nk}\) are expressed as follows:

\[
E_{q^*(\pi)}[(x_n - \mu_k)^T \Lambda_k (x_n - \mu_k)] = D \beta_k^{-1} + v_k (x_n - m_k)^T W_k (x_n - m_k),
\]

\[
\ln \bar{\Lambda}_k = E[\ln \Lambda_k] = \sum_{n=1}^{D} \psi\left(\frac{n+1}{2}\right) + D \ln 2 + \ln |W_k|,
\]

\[
\ln \bar{\mu}_k = E[\ln |\mu_k|] = \psi(\alpha_k) - \psi\left(\sum_{i=1}^{K} \alpha_i\right).
\]

(26)
\[
\ln p_{\alpha_k} = [\psi(\alpha_k) - \psi(\bar{\alpha}_k)] + \frac{1}{2} \sum_{i=1}^{D} \left[ \psi\left( \frac{\bar{y}_k + 1 - i}{2} \right) + D \ln 2 + \ln |W_k| \right] - \frac{D}{2} \ln (2\pi) - \frac{1}{2} D \beta_k^{-1} + \nu_k (x_n - m_k)^T W_k (x_n - m_k),
\]

(27)

where \( \sum_{k=1}^{K} \alpha_k = \bar{\alpha}_k \). 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:

\[
\mathcal{L} = \mathbb{E}[\ln p(X, Z, \pi, \mu, \Lambda)] - \mathbb{E}[\ln q(Z, \pi, \mu, \Lambda)].
\]

(28)

Applying the product rule, we obtain the following equation:

\[
\mathcal{L} = [\mathbb{E}[\ln p(X \mid Z, \mu, \Lambda)] + \mathbb{E}[\ln p(Z \mid \pi)]
\]

\[
+ \mathbb{E}[\ln p(\pi)] + \mathbb{E}[\ln p(\mu, \Lambda)]
\]

\[
- \mathbb{E}[\ln q(Z)] - \mathbb{E}[\ln q(\pi)] - \mathbb{E}[\ln q(\mu, \Lambda)],
\]

(29)

and substituting the following expressions,

\[
\mathbb{E}[\ln p(X \mid Z, \mu, \Lambda)] = \frac{1}{2} \sum_{k=1}^{K} Y_k \left[ \ln \bar{\lambda}_k - D \bar{\beta}_k^{-1} - \nu_k Tr(S_k W_k) - \nu_k (\bar{x}_k - m_k)^T W_k (\bar{x}_k - m_k) - D \ln (2\pi) \right],
\]

\[
\mathbb{E}[\ln p(Z \mid \pi)] = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \ln \bar{\pi}_k,
\]

\[
\mathbb{E}[\ln p(\pi)] = \ln C(\bar{\alpha}_0) + (\bar{\alpha}_0 - 1) \sum_{k=1}^{K} \ln \bar{\pi}_k,
\]

\[
\mathbb{E}[\ln p(\mu, \Lambda)] = \frac{1}{2} \sum_{k=1}^{K} \left[ D \ln \left( \frac{\bar{\bar{\beta}}_k}{2\pi} \right) + \ln \bar{\lambda}_k + D \bar{\beta}_k^{-1} - \bar{\beta}_0 \nu_k (m_k - m_0)^T W_k (m_k - m_0) \right] + \text{KlnB}(W_0, \nu_0) + \frac{v_0 - D - 1}{2} \sum_{k=1}^{K} \nu_k Tr\left( W_0^{-1} W_k \right),
\]

(30)

\[
\mathbb{E}[\ln q(Z)] = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \ln \bar{\alpha}_k,
\]

\[
\mathbb{E}[\ln q(\pi)] = \sum_{k=1}^{K} (\bar{\alpha}_k - 1) \ln \bar{\pi}_k + \ln C(\alpha),
\]

and

\[
\mathbb{E}[\ln q(\mu, \Lambda)] = \sum_{k=1}^{K} \left( \frac{1}{2} \ln \bar{\lambda}_k + \frac{D}{2} \ln \left( \frac{\bar{\bar{\beta}}_k}{2\pi} \right) - D - \mathcal{H}[q(\Lambda_k)] \right),
\]

(31)

where \( \mathcal{H}[q(\Lambda_k)] \) is the entropy of the Wishart distribution. \( \mathcal{L} \) then becomes the objective function to maximize and is given by the following equation:
\[
\mathcal{L} = \left[ \frac{1}{2} \sum_{k=1}^{K} Y_k \left( \ln \bar{\Lambda}_k - D \beta_k^{-1} - v_k T_r (S_k W_k) - v_k (\bar{x}_k - m_k)^\top W_k (\bar{x}_k - m_k) - D \ln(2\pi) \right) \right] \\
+ \left[ \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \ln r_{nk} \right] + \left[ \ln C(a_0) + (a_0 - 1) \sum_{k=1}^{K} \ln \bar{\pi}_k \right] \\
+ \left[ \frac{1}{2} \sum_{k=1}^{K} \left( D \ln \left( \frac{\bar{\beta}_k}{2\pi} \right) + \ln \bar{\Lambda}_k - D \beta_k - \beta_k v_k (m_k - m_0)^\top W_k (m_k - m_0) \right) \right] \\
+ K \ln B(W_0, v_0) + \left( \frac{v_0 - D - 1}{2} \right) \sum_{k=1}^{K} \ln \bar{\Lambda}_k - \frac{1}{2} \sum_{k=1}^{K} v_k T_r (W_0^{-1} W_k) \\
- \left[ \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \ln r_{nk} \right] - \left[ \sum_{k=1}^{K} (\bar{\alpha}_k - 1) \ln \bar{\pi}_k + \ln C(\alpha) \right] \\
- \left[ \frac{1}{2} \ln \bar{\Lambda}_k + \frac{D}{2} \ln \left( \frac{\bar{\beta}_k}{2\pi} \right) - \frac{D}{2} - \mathcal{H}[q(\Lambda_\alpha)] \right],
\]

where 
\[
\ln \bar{\pi}_k = \mathbb{E}[\ln \pi_k] \\
B(W, v) = |W|^{-v/2} \left\{ \Gamma \left( \frac{D+v}{2} \right) \prod_{i=1}^{D} \Gamma \left( \frac{v + 1 - i}{2} \right) \right\}^{-1} \\
C(\alpha) = \frac{\Gamma(\bar{\alpha})}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_k)} \quad \text{and} \\
\mathcal{H}[\Lambda] = -\ln B(W, v) - \left( \frac{v - D - 1}{2} \right) \mathbb{E}[\ln |\Lambda|] + \frac{v D}{2}.
\]

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 $\mathcal{Z}_n$ and $\mathcal{Z}_k$, respectively, represent capsules at the lower and higher-level layers. Let $X_{k|n} \in \mathbb{R}^{D \times k}$ 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 $M_k \in \mathbb{R}^{4 \times k}$ is flattened to obtain capsule $k$’s pose vector $\mu_k \in \mathbb{R}^D$. For ease of computations, we use the precision matrix $\Lambda_k$ instead of the covariance matrix $\Sigma$, and use $\Lambda_k \in \mathbb{R}^{D \times D}$ to represent the diagonal entries of $\Lambda_k$. As mentioned earlier, $r_{nk}$ represents the vector form of the routing responsibilities while $\pi_k$ 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 $a_k$ to represent the likelihood that cluster $k$ is activated by computing the ELBO (equation (32)) and paying a fixed cost of $\beta_k$, 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
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 $2 \times 2$-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 $4 \times 4$ $p_i$ capsule pose matrix and activation. The final layer broadcasts its weight matrices to produce a capsule $p_i$ 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 $28 \times 28$ 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

Algorithm 1: Variational mixture of Gaussians routing.

Figure 1: Architecture of the proposed VMG CapsNet model.
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. We notice that the model converges faster for the images that are less complex compared to CIFAR-10 and COVID-19.

![Figure 2](image-url)
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.

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

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.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>CIFAR-10</th>
<th>Fashion-MNIST</th>
<th>MNIST</th>
<th>COVID-19 radiography</th>
</tr>
</thead>
<tbody>
<tr>
<td>VB-routing {64, 8, 16, 16, #c} [11]</td>
<td>13.10</td>
<td>5.46</td>
<td>0.99</td>
<td>8.01</td>
</tr>
<tr>
<td>VB-routing* {64, 16, 32, 32, #c} [11]</td>
<td>11.2 ± .09</td>
<td>5.61</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>VB-routing+ {64, 16, 16, 16, #c} [11]</td>
<td>12.40</td>
<td>5.2 ± .07</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>EM-routing* {64, 16, 32, 32, #c} [4]</td>
<td>11.2 ± .09</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Multi-lane LBP-Gabor capsule [32]</td>
<td>11.43</td>
<td>5.17</td>
<td>1.00</td>
<td>8.04</td>
</tr>
<tr>
<td>Dynamic routing [3]</td>
<td>35.57</td>
<td>22.45</td>
<td>0.25</td>
<td>8.09</td>
</tr>
<tr>
<td>VMG-routing {32, 4, 8, 8, #c} (ours)</td>
<td>12.19</td>
<td>5.38</td>
<td>1.00</td>
<td>7.01</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>CIFAR-10</th>
<th>Fashion-MNIST</th>
<th>MNIST</th>
<th>COVID-19 radiography</th>
</tr>
</thead>
<tbody>
<tr>
<td>VB-routing {64, 8, 16, 16, #c}</td>
<td>145 K</td>
<td>145 K</td>
<td>145 K</td>
<td>120 K</td>
</tr>
<tr>
<td>VB-routing* {64, 16, 32, 32, #c}</td>
<td>323 K</td>
<td>323 K</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>VB-routing+ {64, 16, 16, 16, #c}</td>
<td>172 K</td>
<td>172 K</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>EM-routing {64, 16, 16, 16, #c}</td>
<td>—</td>
<td>323 K</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>EM-routing* {64, 16, 32, 32, #c}</td>
<td>323 K</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Multi-lane LBP-Gabor capsule</td>
<td>4.10 M</td>
<td>4.10 M</td>
<td>4.10 M</td>
<td>3.70 M</td>
</tr>
<tr>
<td>Dynamic routing</td>
<td>9.3 M</td>
<td>8.2 M</td>
<td>8.2 M</td>
<td>9.8 M</td>
</tr>
<tr>
<td>VMG-routing {32, 4, 8, 8, #c} (ours)</td>
<td>14 K</td>
<td>15.5 K</td>
<td>14 K</td>
<td>10.2 K</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MNIST (%)</th>
<th>CIFAR-10 (%)</th>
<th>Fashion-MNIST (%)</th>
<th>COVID-19 Radiography</th>
<th>Average time</th>
</tr>
</thead>
<tbody>
<tr>
<td>VB-routing {64, 8, 16, 16, #c}</td>
<td>98.53</td>
<td>85.06</td>
<td>92.97</td>
<td>90.92%</td>
<td>30 s, 14 ms</td>
</tr>
<tr>
<td>Multi-lane LBP-Gabor capsule</td>
<td>98.89</td>
<td>86.97</td>
<td>94.00</td>
<td>91.09%</td>
<td>18 s, 25 ms</td>
</tr>
<tr>
<td>Dynamic routing</td>
<td>99.21</td>
<td>65.11</td>
<td>76.32</td>
<td>90.02%</td>
<td>20 s, 11 ms</td>
</tr>
<tr>
<td>VMG-routing {32, 4, 8, 8, #c} (ours)</td>
<td>98.96</td>
<td>86.82</td>
<td>93.71</td>
<td>92.15%</td>
<td>15 s, 23 ms</td>
</tr>
</tbody>
</table>
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 $p_{\text{out}}$ 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 $p_{\text{out}}$ (see Figure 3) sums up to one and measures the certainty (certainty $=p_{\text{out}}$) of the model in its predictions. We obtain the aleatoric uncertainty of the deterministic CapsNet from the same quantity $p_{\text{out}}$ by computing the negative log likelihood (NLL) or the entropy of the predictions.

$$\text{entropy} = -\sum_{i=0}^{\#c} p_i \log(p_i),$$

$$\text{aleatoric uncertainty} = NLL = -\log(p_{\text{out}}),$$

where $0 \leq i < \#c$ 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 $p(\hat{y} | x, w_n)$ conditioned on the weight distribution $w_n$ are obtained out of $N$ predictions on the same input image. The mean probability $p_{\text{out}}^{*}$ is computed for each class $i$ and the maximum mean conditional probability is chosen as the predicted class of the input image.

$$p_{\text{out}}^{*} = \frac{1}{N} \sum_{i=0}^{N-1} p_{\text{out}}^{i},$$

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.
\[ \rho_{\text{inf}}^* = \max(p_{\text{out}}^*). \]  

The averaging in the measure in equation (35) ensures that the epistemic uncertainty in the model is captured. Subsequently, \( \text{NLL}^* = -\log(p_{\text{inf}}^*) \) 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:

\[ \text{entropy}^* = \sum_{i=0}^{\#c} p_{\text{out}}^\star \log(p_{\text{out}}^*), \]  
\[ \sigma^2_{\text{t}} = \sum_{i=0}^{\#c} \sigma^2(p_i) = \sum_{i=0}^{\#c} \left( \frac{1}{N} \sum_{n=0}^{N-1} (p_{\text{in}} - p_i^\star)^2 \right). \]

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.
which constraints the we employed mean-field variational inference (MFVI) there exists linear independence in the features. In this work, unconstrained \( \Lambda \) data, spurious correlations, and domain shift. 

Hin this problem, Covariate shift may also occur due to pixelate-corrupted test distributions of training and target data are different [37]. Bayesian model outputs is the covariate shift, where the uncertainties associated with performing predictions on out-of-distribution data.

- References


