

Article

A Bayesian Neo-Normal Mixture Model (Nenomimo) for MRI-Based Brain Tumor Segmentation

Anindya Apriliyanti Pravitasari ^{1,2}, Nur Iriawan ^{2,*}, Kartika Fithriasari ², Santi Wulan Purnami ², Irhamah ² and Widiana Ferriastuti ³

¹ Department of Statistics, Faculty of Mathematics and Natural Sciences, Universitas Padjadjaran, Jl. Raya Bandung-Sumedang KM. 21, Bandung 45363, Indonesia; anindya.apriliyanti@unpad.ac.id

² Department of Statistics, Faculty of Science and Data Analytics, Institut Teknologi Sepuluh Nopember, Jl. Arif Rahman Hakim Surabaya 60111, Indonesia; kartika_f@statistika.its.ac.id (K.F.); santiwulan08@gmail.com (S.W.P.); irhamah@statistika.its.ac.id (I.)

³ Department of Radiology, Faculty of Medicine, Universitas Airlangga, Surabaya Indonesia 60132, Indonesia; widiana_ferriastuti@yahoo.com

* Correspondence: nur_i@statistika.its.ac.id

Received: 9 June 2020; Accepted: 13 July 2020; Published: 16 July 2020

Abstract: The detection of a brain tumor through magnetic resonance imaging (MRI) is still challenging when the image is in low quality. Image segmentation could be done to provide a clear brain tumor area as the region of interest. In this study, we propose an improved model-based clustering approach for MRI-based image segmentation. The main contribution is the use of the adaptive neo-normal distributions in the form of a finite mixture model that could handle both symmetrical and asymmetrical patterns in an MRI image. The neo-normal mixture model (Nenomimo) also resolves the limitation of the Gaussian mixture model (GMM) and the generalized GMM (GGMM), which are limited by the short-tailed form of their distributions and their sensitivity against noise. Model estimation is done through an optimization process using the Bayesian method coupled with a Markov chain Monte Carlo (MCMC) approach, and it employs a silhouette coefficient to find the optimum number of clusters. The performance of the Nenomimo was evaluated against the GMM and the GGMM using the misclassification ratio (MCR). Finally, this study discovered that the Nenomimo provides better segmentation results for both simulated and real data sets, with an average MCR for MRI brain tumor image segmentation of less than 3%.

Keywords: MRI; image segmentation; neo-normal; mixture model; Bayesian; MCMC

1. Introduction

We chose brain tumor detection as the topic for this study since a brain tumor is the 15th most deadly disease in Indonesia, a comparison that includes all types of cancer. The World Health Organization noted that during 2018, Indonesia had 5323 cases of brain and nervous system tumors, with 4229 cases of mortality [1]. This high mortality rate also occurs in Surabaya, where cases of brain tumors are increasing from year to year. The Dr. Soetomo General Hospital (RSUD) in Surabaya provides radiological examination services for brain tumor patients, including brain scanning with the 3 Tesla and 1.5 Tesla magnetic resonance imaging (MRI) scanners. The higher Tesla power provides better quality MRI images. However, the 1.5 Tesla service is still in high demand due to its low cost; moreover, this service is also covered by the Indonesian government's social security (BPJS). A high-quality MRI image is needed to provide an accurate tumor location for medical surgery. This study tried to employ the image segmentation technique to enhance brain tumor images from the 1.5 Tesla MRI scanner. Segmentation is done by separating the tumor area (which is known as the region

of interest—ROI) from the healthy brain. This helps medical surgery by providing a clear boundary to the parts of the tumor area that must be taken away without damaging healthy parts of the brain.

Various image segmentation techniques have been proposed in the form of clustering algorithms. Traditional methods include hierarchical clustering and the partitioning algorithm [2]. In hierarchical clustering, objects are grouped by forming a cluster tree or dendrogram, while partitioning algorithms are based on objective functions that are optimized to obtain clustering and cluster centers. These methods are mostly heuristics, so they are not based on formal models. The hierarchical clustering for image segmentation was implemented by Bruse et al. [3] and Pestunov et al. [4]. Several partitioning algorithm techniques used in image segmentation are K-means cluster [5], K-medoids [6], and fuzzy-based clustering techniques such as Fuzzy C-means (FCM) [7]. Raftery [8] stated that traditional methods are in great demand because of their simplicity. However, Greeve et al. [9] evaluated the traditional methods and found that those methods were unable to represent real data patterns. Model-based clustering is the most appropriate tool to overcome this problem, especially when facing a limited number of data. One of the methods is known as the Gaussian mixture model (GMM) [10].

In image segmentation, the GMM has several limitations. One of them is the usage of Gaussian distributions, which have symmetry and short-tailed properties. They make the GMM inflexible to overcome several cases of image segmentation that tend to have diverse patterns [11]. Some methods developed to overcome this problem include the use of distributions other than Gaussian, e.g., the Student's t mixture model (SMM) [12] and the Laplace mixture model (LMM) [13]. A similar idea is also seen in the generalized GMM (GGMM), which utilizes the flexibility of form parameters from the Gaussian distribution [14]. However, the use of a Gaussian distribution still leaves problems regarding the short-tailed form of distribution. The existence of a pattern with a long tail must be approached with more Gaussian distribution components that must compile the Gaussian mixture so that the parsimony of the model is less intact. Some of the mentioned models still apply distributions with rules of symmetry and set the mean as the cluster center; therefore, their sensitivity to noise is still high.

Another problem that often arises in image segmentation is the existence of different distributions in the mixture model. Therefore, using a mixture approach from a similar distribution is not adequate. Motivated by a previous study by Rasmussen [10] and Deledalle et al. [14], we propose the use of mixture models with adaptive distributions that can adaptively accommodate a variety of symmetrical and asymmetrical patterns. Despite those facts, these distributions could also handle the limitations related to the short tail form and the lack of robustness of a Gaussian distribution. These adaptive distributions—which are called nearly normal or neo-normal—have symmetrical properties, as with the Gaussian distribution, but can be more pointed or tilted. Some of the distributions in the neo-normal family are Fernandez–Steel skew normal (FSSN) [15] and modified stable to normal from Burr (MSNBurr) [16]. This study used the FSSN and MSNBurr distributions to develop our mixture model, and we named it the Nenomimo. The used FSSN mixture model followed the research of Iriawan et al. [17], which also applied it to MRI image segmentation. The Bayesian method coupled with the Markov chain Monte Carlo (MCMC) approach was employed in model estimation, since the analytical solutions were considered complicated. The proposed models were compared with the GMM and the GGMM for segmenting the simulated and MRI brain tumor image sets. The cluster validation was done by calculating the silhouette coefficient, and the best model was decided by the minimum value of the misclassification ratio.

In the next section, we discuss the Nenomimo, and in Section 3, we continue with the estimation of the purposed models (FSSN and MSNBurr mixture models). Section 4 discusses the cluster validation and model comparison tools. Section 5 shows the discussion of segmentation results performed for the normal model and the Nenomimo in simulated and real data sets. Finally, Section 6 presents some conclusions and directions for future work.

2. Neo-Normal Mixture Model

In this section, we present the mixture model based on neo-normal distributions that were applied in segmenting the MRI brain tumor images. This is explained, as are the finite mixture model and the neo-normal family distributions.

2.1. Finite Mixture Model

This study used MRI images as the input data. MRI images are grayscale digitized image, in which each pixel has gray color intensities. The grayscale intensities in the pixels describe the level of color from black to white. The lower the intensity value, the darker the gray level, and vice versa.

Suppose we have random variable $\mathbf{Y} = Y_1, Y_2, \dots, Y_N$, where y_i denotes the grayscale intensity of i -th pixel observation, and we assume that there are N pixels in an image that groups into K clusters; then, the finite mixture model is a superposition of K specific distributions whose form is given by [11]:

$$f(y_i|\boldsymbol{\theta}, \mathbf{p}) = \sum_{j=1}^K p_j f_j(y_i|\boldsymbol{\theta}_j), \quad (1)$$

where $f(y_i|\boldsymbol{\theta}, \mathbf{p})$ is the mixture densities with parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_K)$ and $\mathbf{p} = (p_1, p_2, \dots, p_K)$, and $f_j(y_i|\boldsymbol{\theta}_j)$ is the density of specific univariate distribution with parameter $\boldsymbol{\theta}_j$. The p_j terms are the proportions of the mixture component density that must satisfy the following constraints:

$$\sum_{j=1}^K p_j = 1 \text{ and } 0 \leq p_j \leq 1, j = 1, 2, \dots, K.$$

The likelihood of the N data can be written as:

$$L(\mathbf{y}|\boldsymbol{\theta}, \mathbf{p}) = \prod_{i=1}^N f(y_i|\boldsymbol{\theta}, \mathbf{p}), \quad (2)$$

where Y_i and Y_n are assumed to be conditional independence $\forall i \neq n$, and come from identical distributions. This formulation is a general finite K components mixture model. In a specific way, we can provide a certain distribution in the form of $f_j(y_i|\boldsymbol{\theta}_j)$. If it is a normal density, then the model becomes a normal or Gaussian mixture model. Likewise, if $f_j(y_i|\boldsymbol{\theta}_j)$ is the neo-normal distribution, then the model becomes a neo-normal mixture model or the Nenomimo.

2.2. Neo-Normal Distribution

The neo-normal distribution family consists of the adaptive distributions that has symmetrical properties, but it can also accommodate skewed or asymmetrical patterns; moreover, it can also be sharper or gentler [18]. These distributions slightly deviate from normal or Gaussian distributions but set their mode as the location parameter. Some of the distributions included in the neo-normal model are the exponential power distribution and Azalini skew normal distribution. The exponential power distribution is similar to the Gaussian distribution, but it can be more pointed and has a thick tail [19]. The Azalini skew normal distribution can accommodate symmetrical or skewed characteristics of a distribution, but this distribution's mode is not stable in its location parameters [20].

Fernandez and Steel [15] overcame the weakness of the Azalini's distribution by introducing another skew normal distribution. This distribution is able to maintain the stability of modes in its location parameters—whether it is symmetrical or skewed—by adding the transformation parameters. This distribution is called the FSSN. Another distribution that is also stable in mode is the MSNBurr by Iriawan [16]. The advantage of using the MSNBurr is that the distribution can accommodate long-tail data, as in an MRI image's pattern. This study used the FSSN and MSNBurr distributions to build the mixture models. The cluster center was more robust since these distributions are stable in their modes.

Supposing the random variable Y_i follows the FSSN distribution, Iriawan et al. [17] provided the density as:

$$f(y_i|\mu, \sigma, \gamma) = \frac{2}{\gamma + \frac{1}{\gamma}} \left(\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{y_i-\mu}{\sigma}\right)^2} \left(\frac{1}{\gamma^2} I_{[\mu, 255)}(y_i) + \gamma^2 I_{(0, \mu)}(y_i) \right) \right), \quad (3)$$

where μ is a location parameter, σ is a dispersion parameter, and γ is a skewness parameter. This distribution transforms the symmetrical normal distribution using inverse scale factors in the positive and negative axes. It loses its symmetry when $\gamma \neq 1$, skews to the left when $\gamma < 1$, and skews to the right when $\gamma > 1$. By substituting Equation (3) as $f_j(y_i|\theta_j)$ in Equation (1), we were able to obtain the FSSN mixture model.

The MSNBurr distribution idea was designed for the skewed distribution, which ensures that the modes of the Burr II distribution for all values of its skewness parameter, γ , remain at its location parameter. Iriawan [16] defined the density function (p.d.f) for the MSBurr family as:

$$f(y_i|\mu, \sigma, \gamma, \kappa) = \frac{\kappa}{\sigma} \exp\left(-\kappa\left(\frac{y_i-\mu}{\sigma}\right)\right) \left(1 + \frac{\exp\left(-\kappa\left(\frac{y_i-\mu}{\sigma}\right)\right)}{\gamma}\right)^{-(\gamma+1)}, \quad (4)$$

where μ and σ are location and dispersion parameters, respectively, and κ is a scale factor. The distribution in Equation (5) is called the MSNBurr(γ, μ, σ), with the value of κ is as Equation (5):

$$\kappa = \frac{1}{\sqrt{2\pi}} \left(1 + \frac{1}{\gamma}\right)^{(\gamma+1)}, \kappa > 0 \quad (5)$$

The MSNBurr mixture model is built by substituting Equation (4) as $f_j(y_i|\theta_j)$ in Equations (1) and (2) with parameter $\theta_j = (\gamma_j, \mu_j, \sigma_j, \kappa_j)$.

3. Bayesian Coupled with Markov Chain Monte Carlo Approach

The fully Bayesian method coupled with MCMC approach was employed to estimate the model parameters and conduct inferences. Bayesian MCMC estimates parameters using full conditional posterior distribution through a sample average approach that is stochastically optimized [21]. In this section, we present the inference with Bayesian for the FSSN and MSNBurr mixture models, and we construct the algorithm.

Suppose N denotes the number of pixels of the MRI image and K denotes the number of the mixture component; then, we assume that each pixel i belongs to a single cluster that is indexed by label $\mathbf{z}_i$. For each observation, label $\mathbf{z}_i$ is assumed to be independent, and the value is given by:

$$\mathbf{z}_i = (z_{i1}, z_{i2}, \dots, z_{iK}) \quad (6)$$

where:

$$z_{ij} = \begin{cases} 1, & \text{if } y_i \text{ belongs to } j^{th} \text{ mixture component,} \\ 0, & \text{otherwise.} \end{cases}$$

For each $\mathbf{z}_i$, there is only one of z_{ij} that equals to 1, and the rest are 0. For a certain i , therefore, $\sum_{j=1}^K z_{ij} = 1$. Single-trial results for each $\mathbf{z}_i$ in exactly one of K possible mixture components have the probabilities $p_1, p_2, \dots, p_K$, $p_j \in (0,1)$, $j = 1, 2, \dots, K$. Since $\mathbf{z}_i$'s are independent, the joint density of the variable is:

$$f(\mathbf{z}|\mathbf{p}) = \prod_{i=1}^N f(\mathbf{z}_i|\mathbf{p}) = \prod_{i=1}^N \prod_{j=1}^K p_j^{z_{ij}}. \quad (7)$$

Let $\mathbf{Y} = Y_1, Y_2, \dots, Y_N$ be an i.i.d random sample drawn from neo-normal mixture density. The likelihood of the mixture density is given by Equation (2). By integrating the label $\mathbf{z}_i$ to the likelihood, we can rewrite the likelihood as:

$$L(\mathbf{y}, \mathbf{z}|\theta, \mathbf{p}) = f(\mathbf{y}|\mathbf{z}, \theta, \mathbf{p}) f(\mathbf{z}|\mathbf{p}) = \prod_{i=1}^N \prod_{j=1}^K \left(p_j f_j(y_i|\theta_j) \right)^{z_{ij}}, \quad (8)$$

where θ_j is the set of the neo-normal parameters.

3.1. Bayesian Approach for FSSN Mixture Model

The Bayesian approach was employed to perform the estimation, owing to the non-close form of the posterior distribution that acts as the multiplication between the likelihood function in Equation (8) and the prior distribution. We designed the prior distribution for the FSSN mixture model (FSSN-MM) following Iriawan et al. [17]. The prior distribution of mixture parameters were discussed by Gelman et al. [22].

$$\begin{aligned}\mu_j &\sim \text{Gaussian}(\eta_j, \varphi_j^2), \\ \sigma_j &\sim \text{Invers Gamma}(\alpha_j, \beta_j), \\ \gamma_j &\sim \text{Gamma}(a_j, b_j), \\ p_j &\sim \text{Dirichlet}(\delta_1, \delta_2, \dots, \delta_K).\end{aligned}$$

The value of hyperparameters were calculated from the data. The hyperparameter $\eta_j = \text{mean}(y_{i \in j})$ and $\varphi_j = \text{var}(y_{i \in j})$, where $y_{i \in j}$ represents all data that belongs to the j^{th} cluster. We also set $\alpha_j = \varphi_j$, $\beta_j = \eta_j/(\alpha_j - 1)$, $a_j = \eta_j^2/\varphi_j$, and $b_j = \eta_j/\varphi_j$. By employing these priors as independent priors and contaminating them with the likelihood, the joint posterior can be constructed easily by multiplying them together. This derivation and the Gibbs sampling algorithm for the FSSN mixture model were adopted from Iriawan et al. [17].

The full conditional posterior was conducted to generate the parameter values. It was derived from multiplying the likelihood and the prior distribution of parameters, which were estimated [23]. The full conditional posterior of each parameter is given below (for simplicity, all equations are written in logarithm and all parts of equation that do not contain the estimated parameter are summed as a constant).

- a) The full conditional posterior for the location parameter $\mu_j, j = 1, 2, \dots, K$ is given by:

$$\ln f(\mu_j | \sigma_j, \gamma_j, p_j, \mathbf{z}, \mathbf{y}) = \text{constant} - \frac{1}{2} \left(\frac{\mu_j - \eta_j}{\varphi_j} \right)^2 - \frac{1}{2} \sum_{i=1}^N z_{ij} \left(\frac{y_{i \in j} - \mu_j}{\sigma_j} \right)^2. \quad (9)$$

- b) The full conditional posterior for parameter $\sigma_j, j = 1, 2, \dots, K$ is given by Equation (10):

$$\begin{aligned}\ln f(\sigma_j | \mu_j, \gamma_j, p_j, \mathbf{z}, \mathbf{y}) &= \text{constant} - (\alpha_j + 1) \ln(\sigma_j) + \frac{\beta_j}{\sigma_j} \\ &+ -n \log \left(\sigma_j \gamma_j \sqrt{2\pi} + \frac{\sigma_j \sqrt{2\pi}}{\gamma_j} \right) - \frac{1}{2} \sum_{i=1}^N z_{ij} \left(\frac{y_{i \in j} - \mu_j}{\sigma_j} \right)^2.\end{aligned} \quad (10)$$

- c) Equation (11) shows the full conditional posterior for parameter $\gamma_j, j = 1, 2, \dots, K$:

$$\begin{aligned}\ln f(\gamma_j | \mu_j, \sigma_j, p_j, \mathbf{z}, \mathbf{y}) &= \text{constant} + (a_j - 1) \ln(\gamma_j) - b_j \gamma_j \\ &+ \sum_{i=1}^N z_{ij} \log \frac{1}{\gamma_j^2} I_{(\mu, 255)}(y_{i \in j}) + \gamma_j^2 I_{(0, \mu)}(y_{i \in j}) \\ &- n \log \left(\sigma_j \gamma_j \sqrt{2\pi} + \frac{\sigma_j \sqrt{2\pi}}{\gamma_j} \right)\end{aligned} \quad (11)$$

- d) Equation (12) shows the full conditional posterior distribution parameter $p_j, j = 1, 2, \dots, K$:

$$f(p_j | \sigma_j, \gamma_j, \mu_j, \mathbf{z}, \mathbf{y}) \propto \prod_{j=1}^K p_j^{\sum_{i=1}^N z_{ij}} = \prod_{j=1}^K p_j^{(\sum_{i=1}^N z_{ij} + 1) - 1}. \quad (12)$$

This is a Dirichlet $(1 + \sum_{i=1}^N z_{i1}, \dots, 1 + \sum_{i=1}^N z_{iK})$ distribution.

- e) The label $z_{ij}, i = 1, 2, \dots, N$ and $j = 1, 2, \dots, K$, only has two possible values, i.e., 0 or 1. For each $\mathbf{z}_i$, there are K possible mixture components, so $\mathbf{z}_i = (z_{i1}, z_{i2}, \dots, z_{iK})$ follows the multinomial $(1, \omega_{i1}, \dots, \omega_{iK})$ distribution where:

$$\omega_{ij} = \frac{f(y_{i \in j} | \mu_j, \sigma_j, \gamma_j) p_j}{f(y_{i \in j})} \quad (13)$$

To estimate the parameter of the FSSN-MM, the Gibbs sampling framework as the process of Bayesian approach was applied. Algorithm 1 states the steps of the Gibbs sampling algorithm.

Algorithm 1: Gibbs sampling for the Fernandez–Steel skew normal-mixture model (FSSN-MM).

- 1: **Initialize** parameters $\mu_1^{(0)}, \dots, \mu_K^{(0)}, \sigma_1^{(0)}, \dots, \sigma_K^{(0)}, \gamma_1^{(0)}, \dots, \gamma_K^{(0)}, p_1^{(0)}, \dots, p_K^{(0)}, \mathbf{z}_1^{(0)}, \dots, \mathbf{z}_N^{(0)}$, set $t = 1$, and determine T .
 - 2: **While** $t \leq T$ **do**
 - 3: Update value $\mu_j, j = 1, 2, \dots, K$ by generating $\mu_j^{(t)}$ according to Equation (9).
 - 4: Update value $\sigma_j, j = 1, 2, \dots, K$ by generating $\sigma_j^{(t)}$ according to Equation (10).
 - 5: Update value $\gamma_j, j = 1, 2, \dots, K$ by generating $\gamma_j^{(t)}$ according to Equation (11).
 - 6: Update value $p_j, j = 1, 2, \dots, K$ by generating $p_j^{(t)}$ that follow the Dirichlet distribution with parameters $(1 + \sum_{i=1}^N z_{i1}, \dots, 1 + \sum_{i=1}^N z_{iK})$.
 - 7: Update value $\mathbf{z}_i$ by generating $\mathbf{z}_i^{(t)}$ that follows the multinomial $(1, \omega_{i1}, \dots, \omega_{iK})$ distribution, where $\omega_{ij}, i = 1, 2, \dots, N, j = 1, 2, \dots, K$ as in Equation (13).
 - 8: Set $t = t + 1$.
 - 9: **Return** $\mu_1^{(T)}, \dots, \mu_K^{(T)}, \sigma_1^{(T)}, \dots, \sigma_K^{(T)}, \gamma_1^{(T)}, \dots, \gamma_K^{(T)}, p_1^{(T)}, \dots, p_K^{(T)}, \mathbf{z}_1^{(T)}, \dots, \mathbf{z}_N^{(T)}$.
-

3.2. Bayesian Approach for MSNBurr Mixture Model

The prior distribution for MSNBurr was discussed by Iriawan [16]. The proportion of the mixture components follows the Dirichlet distribution, as in the FSSN-MM. The following are prior distributions for each parameter of the MSNBurr mixture model (MSNBurr-MM):

$$\begin{aligned}\mu_j &\sim \text{Gaussian}(\eta_j, \varphi_j^2), \\ \sigma_j &\sim \text{Invers Gamma}(\alpha_j, \beta_j), \\ \gamma_j &\sim \text{Generalised Symmetrical Beta}(\tau_j, \tau_j, l_j, u_j), \\ p_j &\sim \text{Dirichlet}(\delta_1, \delta_2, \dots, \delta_K).\end{aligned}$$

The hyperparameters value are given as in the FSSN-MM; however, we set the hyperparameter for prior γ_j by $\tau_j = \varphi_j/2$, $l_j = 0$, and $u_j = 2$. By setting these priors as independent priors and multiplying them with the likelihood, its joint posterior could be found easily (see Appendix A). The full conditional posterior distributions for each parameter are discussed below. For simplicity, the formulas are shown as logarithms (some derivations are shown in Appendix B) and all parts of the equation that do not contain the estimated parameter are assembled as a constant.

a) The full conditional posterior for $\mu_j, j = 1, 2, \dots, K$ is shown in Equation(14):

$$\begin{aligned}\ln f(\mu_j | \sigma_j, \gamma_j, p_j, \kappa_j, \mathbf{z}, \mathbf{y}) = \text{constant} &- \sum_{i=1}^N z_{ij} \left(\kappa_j \left(\frac{y_{i \in j} - \mu_j}{\sigma_j} \right) \right) - \frac{1}{2} \left(\frac{\mu_j - \eta_j}{\varphi_j} \right)^2 \\ &+ \sum_{i=1}^N z_{ij} \left(-(\gamma_j + 1) \ln \left(1 + \frac{\exp \left(-\kappa_j \left(\frac{y_{i \in j} - \mu_j}{\sigma_j} \right) \right)}{\gamma_j} \right) \right)\end{aligned}\quad (14)$$

b) The following Equation (15) shows the full conditional posterior for $\sigma_j, j = 1, 2, \dots, K$:

$$\begin{aligned}\ln f(\sigma_j | \mu_j, \gamma_j, p_j, \kappa_j, \mathbf{z}, \mathbf{y}) = \text{constant} &- (\alpha_j + 1) \ln \sigma_j - \frac{\beta_j}{\sigma_j} + \sum_{i=1}^N z_{ij} \ln \left(\frac{\kappa_j}{\sigma_j} \right) + \\ &\sum_{i=1}^N z_{ij} \left(-(\gamma_j + 1) \ln \left(1 + \frac{\exp \left(-\kappa_j \left(\frac{y_{i \in j} - \mu_j}{\sigma_j} \right) \right)}{\gamma_j} \right) \right) + \sum_{i=1}^N z_{ij} \left(-\kappa_j \left(\frac{y_{i \in j} - \mu_j}{\sigma_j} \right) \right)\end{aligned}\quad (15)$$

c) The full conditional posterior of $\gamma_j, j = 1, 2, \dots, K$ is given bellow:

$$\ln f(\gamma_j | \mu_j, \sigma_j, p_j, \kappa_j, \mathbf{z}, \mathbf{y}) = \text{constant} - (\tau_j - 1) \ln((\gamma_j - l_j)(u_j - \gamma_j)) + \sum_{i=1}^N z_{ij} \left(-(\gamma_j + 1) \ln \left(1 + \frac{\exp\left(-\kappa_j \left(\frac{y_{i \in j} - \mu_j}{\sigma_j} \right)\right)}{\gamma_j} \right) \right) + \sum_{i=1}^N z_{ij} \left(-\kappa_j \left(\frac{y_{i \in j} - \mu_j}{\sigma_j} \right) \right). \quad (16)$$

d) The full conditional posterior for $p_j, j = 1, 2, \dots, K$ is as shown in Equation (12).

e) The full conditional posterior of the label z_{ij} follows the multinomial $(1, \omega_{i1}, \dots, \omega_{iK})$ distribution, where $\omega_{ij}, i = 1, 2, \dots, N, j = 1, 2, \dots, K$, as in Equation (13).

The Gibbs sampling algorithm was constructed with a full conditional posterior for each parameter. The step of Gibbs sampling for the MSNBurr-MM is presented in Algorithm 2.

Algorithm 2: Gibbs sampling for the modified stable to normal from Burr-mixture model (MSNBurr-MM)

- 1: **Initialize** parameters $\mu_1^{(0)}, \dots, \mu_K^{(0)}, \sigma_1^{(0)}, \dots, \sigma_K^{(0)}, \gamma_1^{(0)}, \dots, \gamma_K^{(0)}, p_1^{(0)}, \dots, p_K^{(0)}, \mathbf{z}_1^{(0)}, \dots, \mathbf{z}_N^{(0)}$, set $t = 1$, and determine T .
 - 2: **While** $t \leq T$ **do**
 - 3: Update value $\mu_j, j = 1, 2, \dots, K$ by generating $\mu_j^{(t)}$ according to Equation(14).
 - 4: Update value $\sigma_j, j = 1, 2, \dots, K$ by generating $\sigma_j^{(t)}$ according to Equation (15).
 - 5: Update value $\gamma_j, j = 1, 2, \dots, K$ by generating $\gamma_j^{(t)}$ according to Equation (16).
 - 6: Update value $p_j, j = 1, 2, \dots, K$ by generating $p_j^{(t)}$ as in Equation (12).
 - 7: Update value $\kappa_j, j = 1, 2, \dots, K$ as in Equation (6).
 - 8: Update value $\mathbf{z}_i$ by generating $\mathbf{z}_i^{(t)}$ that follows the multinomial $(1, \omega_{i1}, \dots, \omega_{iK})$ distribution, where $\omega_{ij}, i = 1, 2, \dots, N, j = 1, 2, \dots, K$, as in Equation (13).
 - 9: Set $t = t + 1$.
 - 10: **Return** $\mu_1^{(T)}, \dots, \mu_K^{(T)}, \sigma_1^{(T)}, \dots, \sigma_K^{(T)}, \gamma_1^{(T)}, \dots, \gamma_K^{(T)}, p_1^{(T)}, \dots, p_K^{(T)}, \mathbf{z}_1^{(T)}, \dots, \mathbf{z}_N^{(T)}$.
-

Both Algorithm 1 and Algorithm 2 were used as a basis for programming via MATLAB. The generation of samples in both algorithms, we used the acceptance rejection sampling method. All programs were run under the computer specification of Intel Core i7, 8GB RAM, 256 GB SSD, without GPU and VRAM. For the sake of reproducibility of our results, we made our MATLAB code for the MSNBurr-MM available online at <https://github.com/anindya364/nenomimo>.

4. Cluster Validation and Comparison Tools

Cluster validation was done by employing the silhouette coefficient (SC). The value of SC determines the optimum number of clusters. SC_i^j is formulated in Equation (17) [24]:

$$SC_i^j = \frac{\xi_i^j - \zeta_i^j}{\max(\zeta_i^j, \xi_i^j)} \quad (17)$$

where SC_i^j is the SC value for all i^{th} data at the j^{th} cluster, ζ_i^j is the average squared Euclidean distance that denotes how similar the i^{th} data are to their cluster, and ξ_i^j indicates how similar the i^{th} data are to the other clusters. The SC_j of a cluster in Equation (18) is the average value of all the data included in the j^{th} cluster, SC_i^j . The overall SC (OSC) of the image is the average of all SC_j of all clusters formulated in Equation (19):

$$SC_j = \frac{1}{n_j} \sum_{i=1}^{n_j} SC_i^j, \quad (18)$$

$$OSC = \frac{1}{K} \sum_{j=1}^K SC_j. \quad (19)$$

The *OSC* has a range of values from -1 to 1 . The closer the *OSC* is to 1 , the more likely the data should be in the cluster.

The comparison measurement tool used for the segmentation result was the misclassification ratio (*MCR*). The *MCR* calculates the differences between the segmentation results with the ground truth (GT) image. The smaller the value of *MCR*, the better the segmentation results [17]. The formula of *MCR* is given by Equation (20):

$$MCR = \frac{\sum_{i=1}^N |\text{pixel cluster output}_i - \text{pixel ground truth}_i|}{N} \quad (20)$$

5. Result and Discussion

In this section, we compare the proposed models with the GMM by Rasmussen [10] and the GGMM by Deledalle et al. [14]. The GMM and the GGMM represented the model-based normal distribution, while the FSSN-MM and the MSNBurr-MM represented the Nenomimo.

5.1. Application for Data Simulation

The data simulation used in this study were synthetic images and natural images. For synthetic images, original images were also set for the GT. To add more challenge in segmentation, we provided noise (Gaussian noise, salt-paper noise, and speckle noise) to each image. The synthetic images (SI01 and SI02) and natural images (NI01 and NI02) with their GT are visualized in Figure 1.

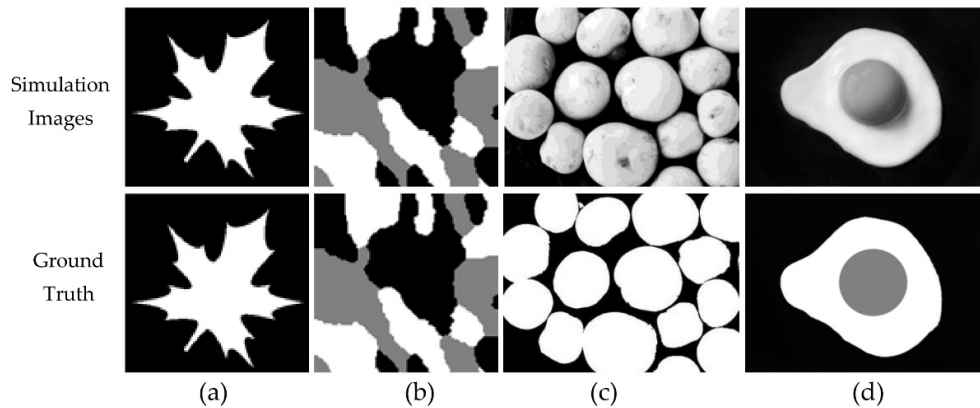


Figure 1. Synthetic images: (a) SI01 and (b) SI01. Natural images: (c) NI01 and (d) NI02.

The segmentation results of each method for SI01, SI02, NI01, and NI02 are shown in Figures 2–5. The methods run under two clusters for SI01 and NI01, while SI02 and NI02 run under three clusters.

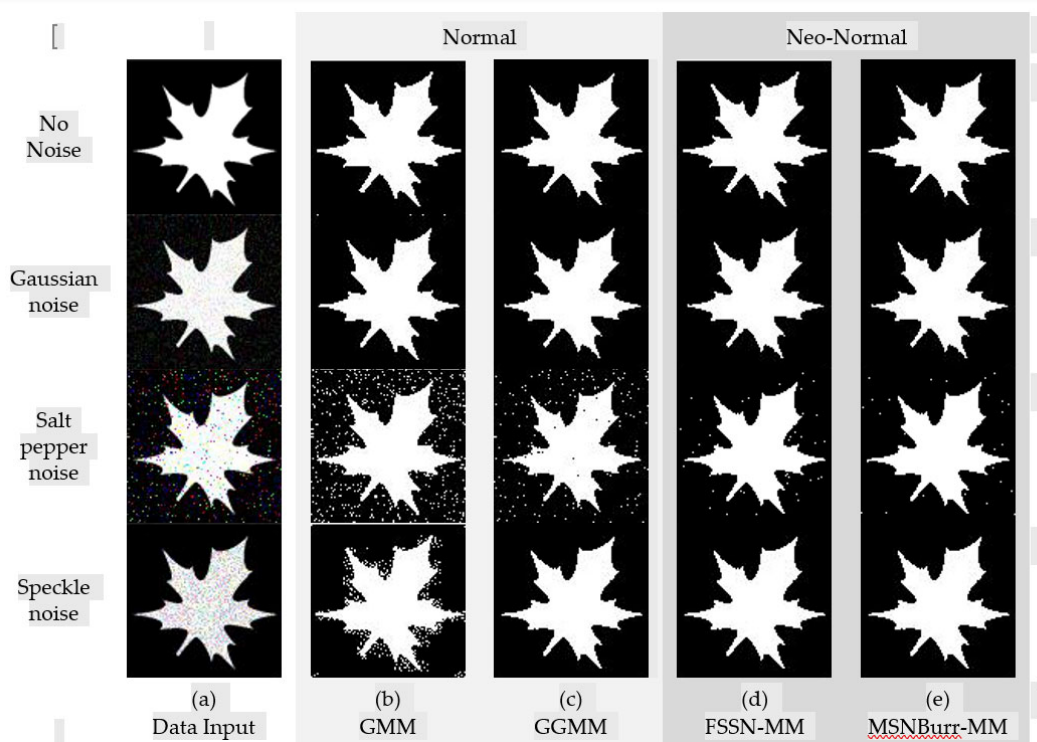


Figure 2. Segmentation result of SI01 for different methods.

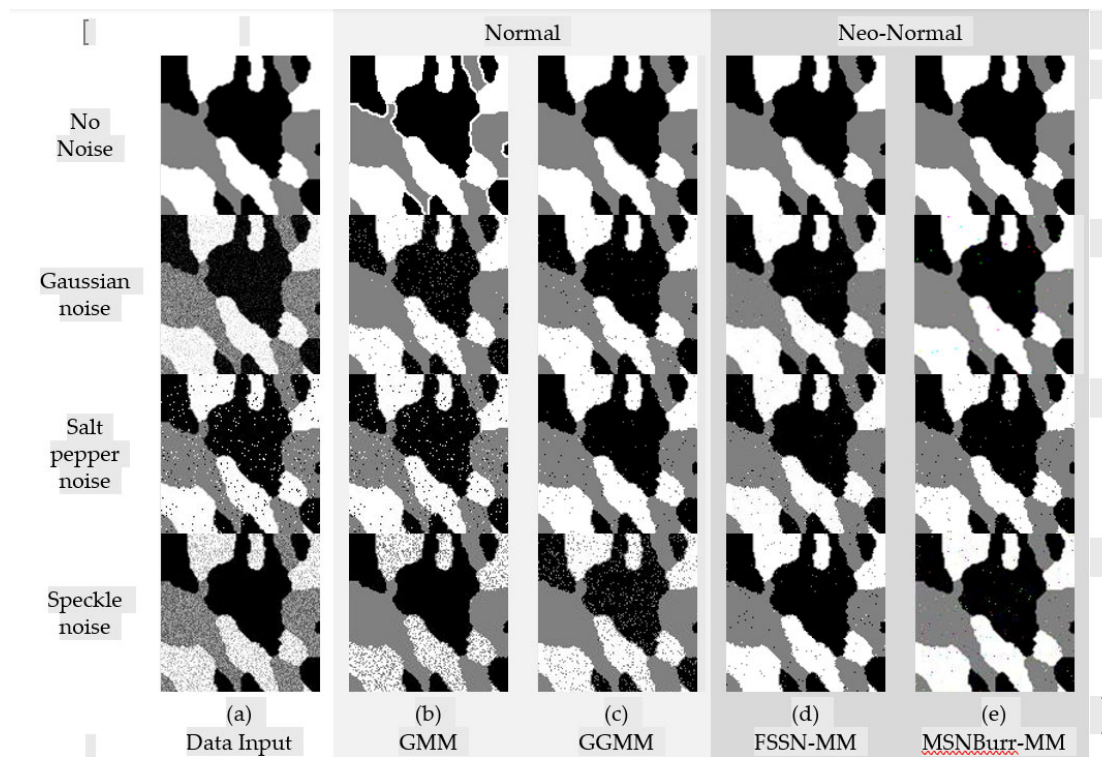


Figure 3. Segmentation result of SI02 for different methods.

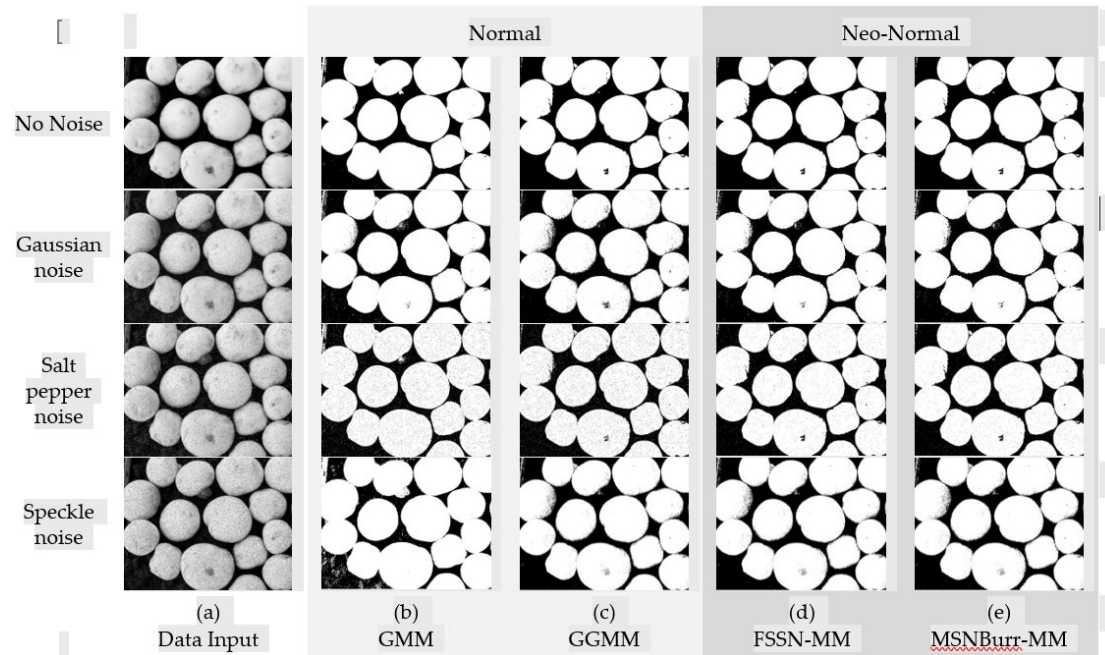


Figure 4. Segmentation result of NI01 for different methods.

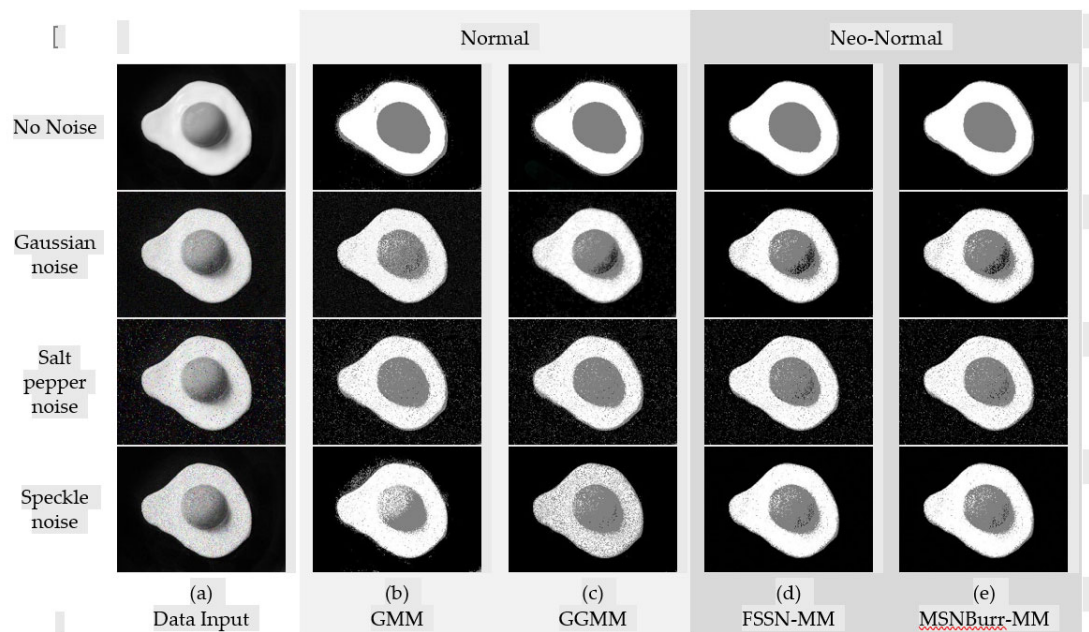


Figure 5. Segmentation result of NI02 for different methods.

The accuracy of segmentation results was shown by the *MCR*. As visualized in Figure 6, the minimum *MCR* was shown by the Nenomimo (FSSN-MM and MSNBurr-MM). For different types of noise, the Nenomimo gave a better performance in segmenting both the synthetic images and natural images. This indicated that the Nenomimo is more robust in handling noise than the GMM and the GGMM. Overall, for the simulation dataset, the Nenomimo segmented the images more precisely than the model-based normal distribution.

The next application was for a real dataset. The MRI-based brain tumor segmentation was also done under the Nenomimo, the GMM, and the GGMM.

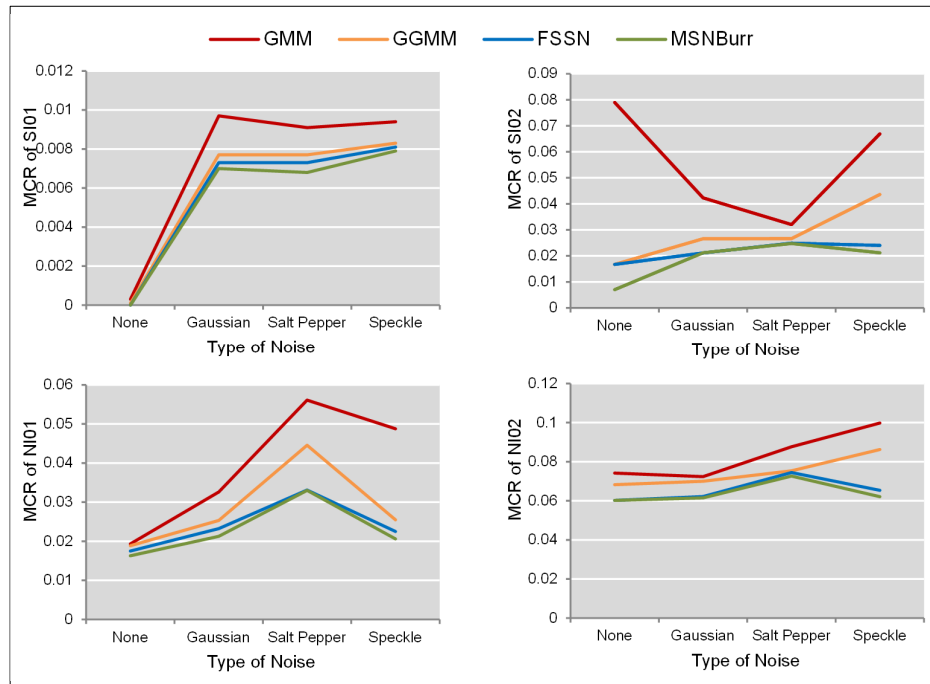


Figure 6. The misclassification ratio (MCR) comparison for SI01, SI02, NI01, and NI02 segmentation results.

5.2. Application for MRI-Based Brain Tumor

The real datasets were from the Dr. Soetomo General Hospital (RSUD), Surabaya. The MRI sequences in this study were from T1 MEMP + C and T2 FLAIR; both were from the axial point of view. T1 MEMP + C was the MRI sequence in which the tumor area was enhanced by contrast media, which is the material given to patients, either orally or injected, before doing MRI scanning. T2 FLAIR was the MRI sequence without contrast media; this sequence visualized the swelling or edema of the brain tumor more visibly; therefore, the ROI of T2 FLAIR was usually larger than T1 MEMP + C.

We tried to segment about 75 images from the T1 MEMP + C and T2 FLAIR sequences. The tumor area was set as the ROI, while another area was set as a non-ROI. All sequences used in this study passed medical approval, and the GT was also conducted under medical judgment. All MRI images were analyzed under normal and neo-normal models. This paper only displays four sample images. The sample datasets are visualized in Figure 7, while the histograms are shown in Figure 8.

Figure 7a,c includes the sample from the T1 MEMP + C sequence, while Figure 7b,d is the sample from the T2 FLAIR sequence. The sample data images are named IM01, IM02, IM03, and IM04. The first applied procedure was preprocessing. It was done to eliminate the fat around the skull, which could have interfered with the segmentation process. Figure 8 shows the image histogram after preprocessing. Unlike Figure 8b–d, the multimodality in Figure 8a is not clearly visible since the frequency of the other modes was very small. Due to the multimodalities present in the MRI data pattern, applying the mixture model framework was considered appropriate for this case.

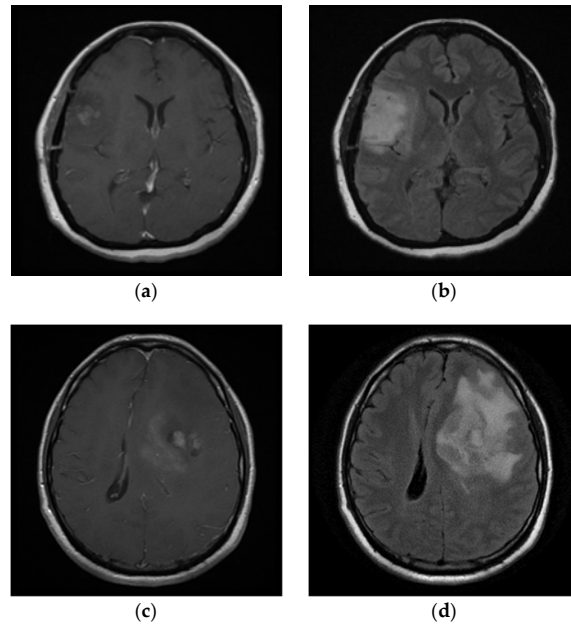


Figure 7. The real dataset from medical consent: (a) IM01, (b) IM02, (c) IM03, and (d) IM04.

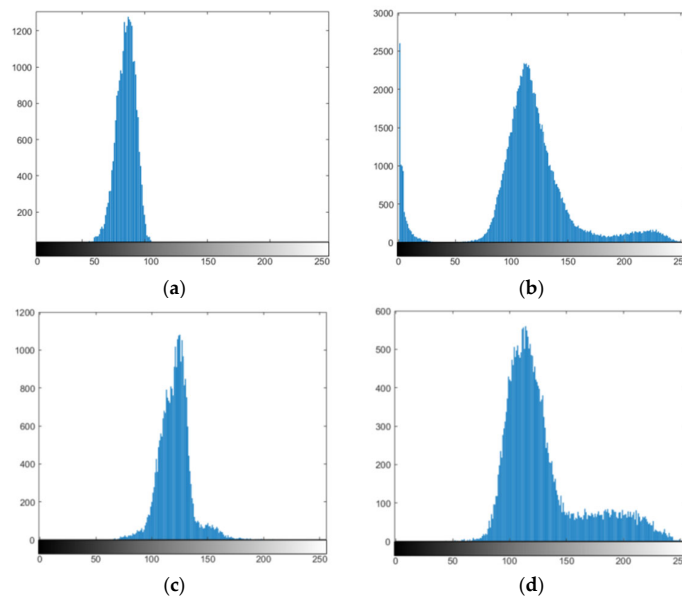


Figure 8. The histograms of (a) IM01, (b) IM02, (c) IM03, and (d) IM04 after preprocessing.

The 75 datasets were segmented under the GMM, the GGMM, the FSSN-MM, and the MSNBurr-MM. The OSC was calculated to provide the optimum number of clusters. Table 1 shows the sample of the OSC for IM01–04. The optimum value of the OSC led to the optimum number of clusters. The optimum number of clusters for the Nenomimo for IM01 and IM03 was $K = 3$, while IM02 and IM04 reached the optimum number at $K = 4$. Various results were found with the GMM; for IM01, the optimum number of clusters was $K = 7$, while IM02 and IM04 reached the optimum number at $K = 4$, and IM03 reached the optimum number at $K = 5$. The GGMM had the cluster optimum at $K = 4$ for IM01 and IM04, $K = 5$ for IM02, and $K = 3$ for IM03. The segmentation results for the optimum clusters are visualized in Figure 9.

Table 1. The summary of overall silhouette coefficients (OSCs) for the sample image. GMM: Gaussian mixture model; GGMM: generalized Gaussian mixture model.

Algorithm	Dataset	Number of Clusters (K)							
		3	4	5	6	7	8	9	
Normal	GMM	IM01	0.5498	0.3488	0.539	0.4553	0.6949*	0.6936	0.654
		IM02	0.904	0.9133*	0.748	0.6909	0.758	0.795	0.748
		IM03	0.7924	0.803	0.8356*	0.8034	0.7654	0.7497	0.7138
		IM04	0.8875	0.9275*	0.8764	0.8759	0.8437	0.7868	0.7745
	GGMM	IM01	0.8021	0.8263*	0.7412	0.7406	0.6824	0.6211	0.5488
		IM02	0.7682	0.8487	0.8935*	0.8204	0.7581	0.6328	0.7004
		IM03	0.9002*	0.8955	0.7324	0.7101	0.6048	0.5545	0.6266
		IM04	0.8333	0.8859*	0.8479	0.8022	0.7643	0.7614	0.7321
Neo-Normal	FSSN-MM	IM01	0.8725*	0.6412	0.6752	0.6351	0.5897	0.6325	0.6606
		IM02	0.8109	0.8636*	0.8464	0.812	0.8467	0.7929	0.8292
		IM03	0.9145*	0.8014	0.7512	0.7049	0.6854	0.7046	0.6699
		IM04	0.8347	0.9010*	0.9009	0.8794	0.8695	0.8529	0.7558
	MSNBurr-MM	IM01	0.9300*	0.8349	0.7412	0.7218	0.714	0.7331	0.7435
		IM02	0.8428	0.9119*	0.9106	0.7729	0.7322	0.6731	0.7091
		IM03	0.9157*	0.8014	0.7491	0.7072	0.6938	0.6931	0.6912
		IM04	0.8364	0.9230*	0.8647	0.8009	0.801	0.7647	0.7577

* indicates the maximum OSC value.

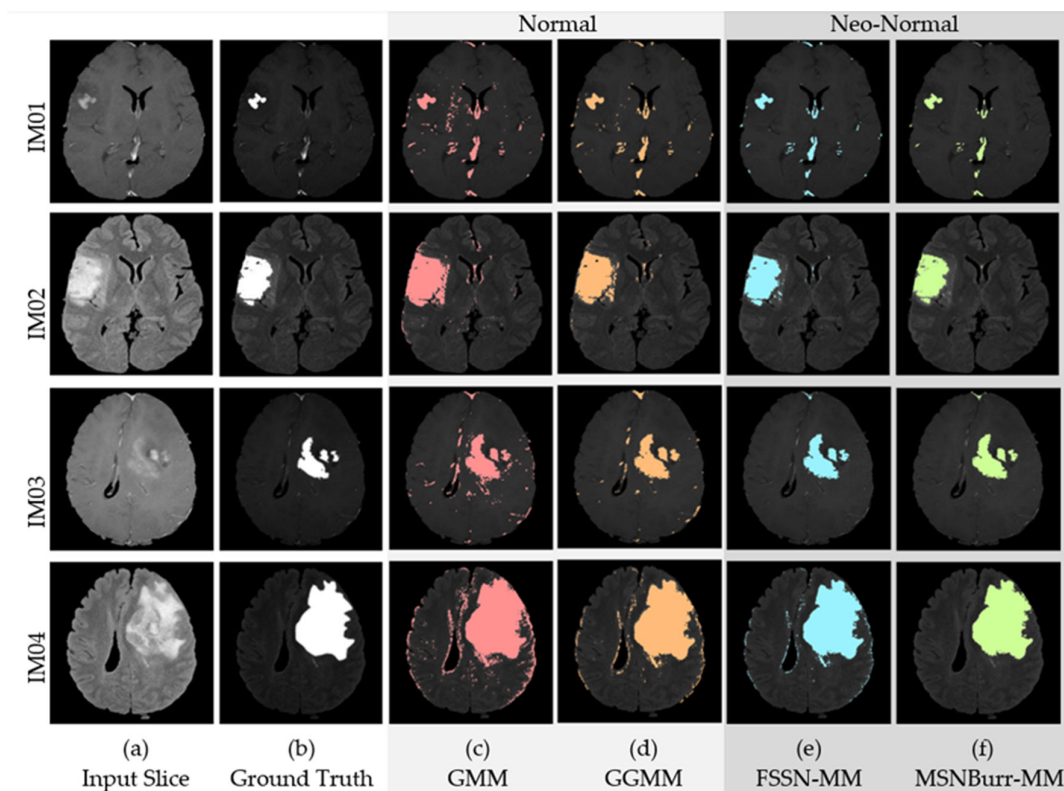


Figure 9. The region of interest (ROI) of (a) input slice in different layers compared with (b) the ground truth (GT) for the (c) GMM, (d) GGMM, (e) FSSN-MM, and (f) MSNBurr-MM algorithms.

Based on segmentation results in Figure 9, we can see that the ROI from the GMM still contained more noise than the GGMM, the FSSN-MM, and the MSNBurr-MM. For empirical comparison, the *MCR* from 75 dataset was found. Figure 10 shows the comparison of the *MCR* for all dataset using

different methods. The *MCR* from Figure 10 shows that the minimum value was reached by the MSNBurr-MM, while the FSSN-MM gave similar results. This indicated that the Nenomimo have a better performance than the model-based normal distribution. The average value of the *MCR* for the FSSN-MM was about 0.02761, while the MSNBurr-MM had an average *MCR* value of about 0.02644. This value showed that the Nenomimo could segment the real dataset with a misclassification ratio of less than 3%.

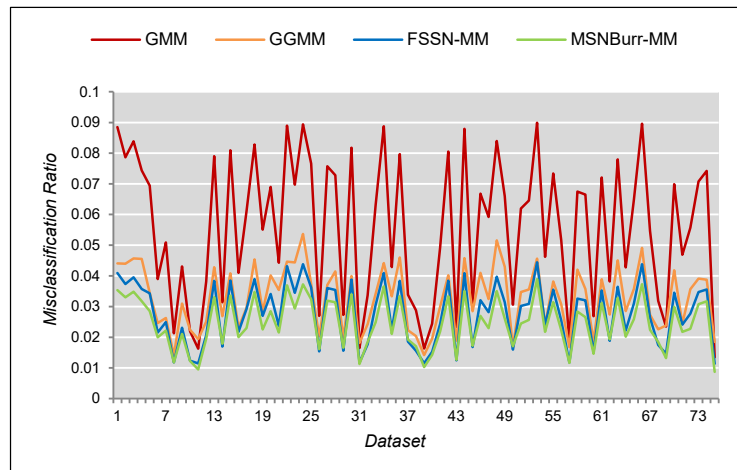


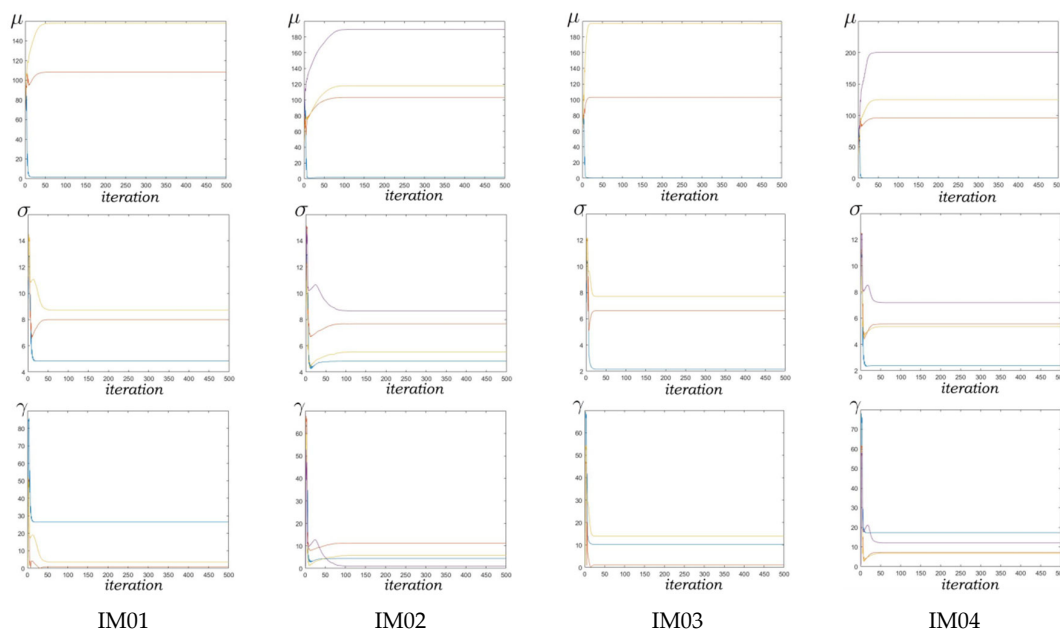
Figure 10. The comparison of the *MCR* for each image and each method.

The parameter of the Nenomimo for optimum clusters is shown in Table 2. In Table 2, the ROI for each dataset is denoted by the last cluster. For example, in IM01, for the FSSN-MM, the ROI had a cluster center at a grayscale intensity of 165 with the range of intensity dispersion at about 6.719. The pixel in the ROI was about 5.4% from all MRI areas. Different results were provided by the MSNBurr-MM, which had a cluster center at a grayscale intensity of about 163, an intensity dispersion of 7.138, and the ROI pixel just 6.3% from the MRI areas. Both the FSSN-MM and the MSNBurr-MM had skewness parameters $\gamma > 1$ in the first and third clusters, indicating that the distribution of the grayscale intensities was right skewed. The skewness parameter $\gamma < 1$ in the second cluster of IM01 indicated the left skew of the distribution.

Table 2. The parameter of the Nenomimo for the sample image.

Sample Image	j	FSSN-MM				MSNBurr-MM			
		p_j	μ_j	σ_j	γ_j	p_j	μ_j	σ_j	σ_j
IM01	1	0.084	1.569	4.856	26.468	0.076	1.575	3.774	14.024
	2	0.862	89.569	7.992	0.904	0.861	87.891	6.678	0.864
	3	0.054	165.001	6.719	3.566	0.063	163.257	7.138	2.555
IM02	1	0.163	2.684	4.777	4.625	0.186	2.887	4.576	4.128
	2	0.487	113.521	7.562	11.061	0.501	112.372	5.123	10.784
	3	0.196	158.01	5.324	7.225	0.196	154.753	4.852	8.211
	4	0.154	197.731	7.789	0.916	0.117	198.262	5.476	0.913
IM03	1	0.079	0.379	2.149	10.251	0.074	0.375	2.14	9.321
	2	0.813	103.168	6.611	1.083	0.839	103.194	6.62	0.992
	3	0.108	197.113	7.708	13.934	0.087	197.263	7.685	18.858
IM04	1	0.054	0.475	2.379	17.343	0.069	0.445	3.469	16.321
	2	0.665	96.015	5.555	7.236	0.564	122.981	6.278	5.048
	3	0.136	125.176	5.355	6.725	0.245	161.062	3.283	4.769
	4	0.145	200.36	7.18	12.089	0.122	199.908	4.476	8.866

Figure 11 shows the historically generated parameter for the Nenomimo iteration. Under the computer specification with processor Intel Core i7, 8GB RAM, 256GB SSD, without GPU and VRAM, the average running time for clustering the MRI images with the FSSN-MM took about six minutes of computer processing, while the MSNBurr-MM took about four minutes. Figure 11 we can see that parameters from both the FSSN-MM and the MSNBurr-MM reached convergence fast. The steady state for the historical mean plotted in Figure 6 was reached before 100 iterations.

**(a)** FSSN-MM parameter convergence

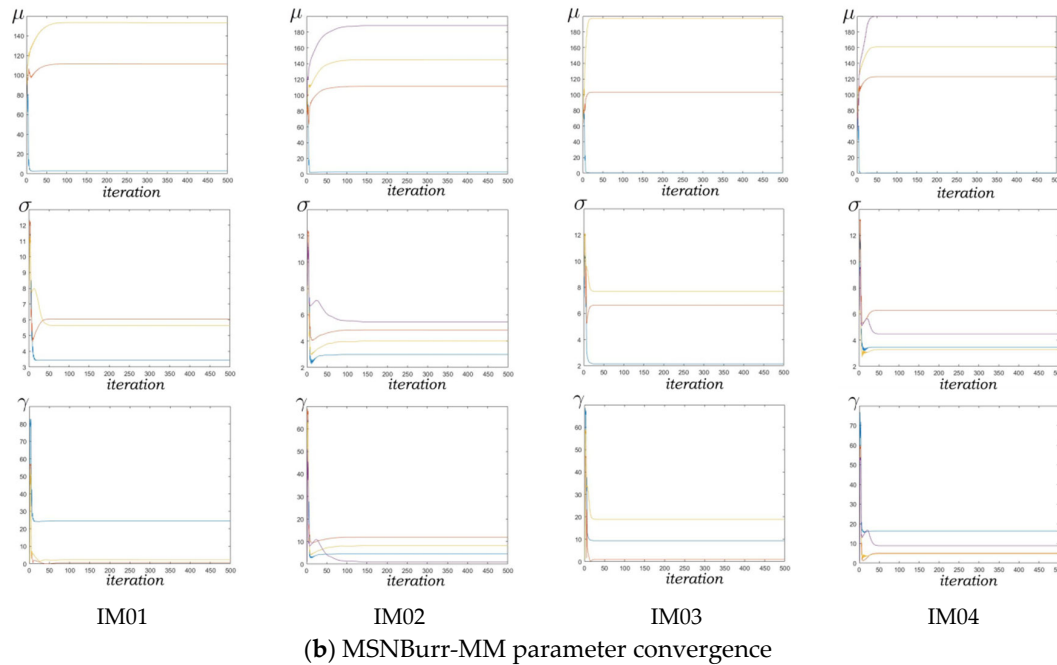


Figure 11. Historically generated the Nenomimo parameter for sample MRI images.

6. Conclusions

In this paper, we presented an improved method for image segmentation, namely the neo-normal mixture model (Nenomimo). The parameter estimation of the Nenomimo employs the Bayesian method coupled with an MCMC framework via Gibbs sampling. The algorithm ran well when using a computer with processor Intel Core i7, 8GB RAM, 256GB SSD, without GPU and VRAM, and it reached convergence quickly. The proposed method successfully overcame the limitation of mixture models based on the normal distribution. The Nenomimo gave a better performance than the GMM and the GGMM on both simulated and real datasets. For segmenting MRI brain tumor images, the Nenomimo showed powerful results, with the value of the misclassification ratio being less than 3%.

In the present work, segmentation results for MRI brain tumors using the Nenomimo still had a little noise, which was identified as the ROI. Therefore, for future research, it is recommended to continue to develop the Nenomimo model that considers pixel spatial dependencies for image segmentation.

Author Contributions: A.A.P., N.I., K.F., S.W.P., I., and W.F. understood and designed the research; A.A.P. collected the data, analyzed the data, and drafted the paper. All authors critically read and revised the draft and approved the final paper. All authors have read and agreed to the published version of the manuscript.

Funding: The authors are grateful to the Directorate for Research and Community Service (DRPM) Ministry of Research, Technology, and Higher Education Indonesia, which supports this research under PT research grant no. 1311/PKS/ITS/2020. This research is also partially funded by the Indonesian Ministry of Research, Technology, and Higher Education under World Class University (WCU) Program managed by Institut Teknologi Bandung.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A Joint Posterior Density for Nenomimo

Assuming that each parameter in the Nenomimo, μ , σ , γ , and $\mathbf{p}$, is independent (where $\mu = (\mu_1, \mu_2, \dots, \mu_j, \dots, \mu_K)$, $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_j, \dots, \sigma_K)$, $\gamma = (\gamma_1, \gamma_2, \dots, \gamma_j, \dots, \gamma_K)$), then $\mathbf{y} = (y_1, y_2, \dots, y_i, \dots, y_N)$ is the pixels in the MRI image. Therefore, the joint prior density for each μ , σ , and γ is given by:

$$g(\boldsymbol{\mu}) = g(\mu_1) \times g(\mu_2) \times \dots \times g(\mu_K) = \prod_{j=1}^K g(\mu_j),$$

$$g(\boldsymbol{\sigma}) = g(\sigma_1) \times g(\sigma_2) \times \dots \times g(\sigma_K) = \prod_{j=1}^K g(\sigma_j),$$

$$g(\boldsymbol{\gamma}) = g(\gamma_1) \times g(\gamma_2) \times \dots \times g(\gamma_K) = \prod_{j=1}^K g(\gamma_j).$$

Via the Bayes theorem, the joint posterior density of the Nenomimo is:

$$f(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\gamma}, \mathbf{p}, \mathbf{z} | \mathbf{y}) = \frac{f(\mathbf{y}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\gamma}, \mathbf{p}, \mathbf{z})}{f(\mathbf{y})} \propto f(\mathbf{y}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\gamma}, \mathbf{p}, \mathbf{z}).$$

Substituting the joint density for each prior in MSNBurr-MM, we then obtain:

$$\begin{aligned} f(\mathbf{y}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\gamma}, \mathbf{p}, \mathbf{z}) &= f(\mathbf{y}, \mathbf{z} | \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\gamma}, \mathbf{p}) g(\boldsymbol{\mu}) g(\boldsymbol{\sigma}) g(\boldsymbol{\gamma}) g(\mathbf{p}) = \\ &= \prod_{i=1}^N \prod_{j=1}^K \left(p_j \frac{\kappa}{\sigma} \exp \left(-\kappa \left(\frac{\gamma - \mu}{\sigma} \right) \right) \left(1 + \frac{\exp \left(-\kappa \left(\frac{\gamma - \mu}{\sigma} \right) \right)}{\gamma} \right)^{-(\gamma+1)} \right)^{z_{ij}} \times \prod_{j=1}^K g(\mu_j) g(\sigma_j) g(\gamma_j) \times (K-1)!. \end{aligned} \quad (\text{A1})$$

Since:

$$\begin{aligned} \mu_j &\sim \text{Gaussian}(\eta_j, \varphi_j^2), \\ \sigma_j &\sim \text{Invers Gamma}(\alpha_j, \beta_j), \\ \gamma_j &\sim \text{Generalised Symmetrical Beta}(\tau_j, \tau_j, l_j, u_j), \\ p_j &\sim \text{Dirichlet}(\delta_1, \delta_2, \dots, \delta_K), \end{aligned}$$

We then get the following in detail.

a) The prior density for $\mu_j, j = 1, 2, \dots, K$ is:

$$g(\mu_j | \eta_j, \varphi_j^2) = \frac{1}{\sqrt{2\pi}\varphi_j} \exp \left(-\frac{1}{2} \left(\frac{\mu_j - \eta_j}{\varphi_j} \right)^2 \right),$$

where η_j and φ_j^2 are the mean and variance from the data, respectively; $0 \leq \eta_j \leq 255$ and $\varphi_j > 0$.

b) The prior density for $\sigma_j, j = 1, 2, \dots, K$ is given by:

$$g(\sigma_j | \alpha_j, \beta_j) = \frac{\beta_j^{\alpha_j}}{\Gamma(\alpha_j)} (\sigma_j)^{-(\alpha_j+1)} \exp \left(-\frac{\beta_j}{\sigma_j} \right).$$

c) The prior density for $\gamma_j, j = 1, 2, \dots, K$ is:

$$g(\gamma_j | \tau_j, \tau_j, l_j, u_j) = \frac{\Gamma(2\tau_j) ((\gamma_j - l_j)(u_j - \gamma_j))^{\tau_j-1}}{\Gamma(\tau_j)^2 R^{(2\tau_j-1)}},$$

where $R = u_j - l_j, u_j > l_j \geq 0, \tau_j \geq 1$, and $\gamma_j > 0$.

d) Prior density for $p_j, j = 1, 2, \dots, K$.

Assume that $\mathbf{p} \sim \text{Dir}(p_1, p_2, \dots, p_K, \delta_1, \delta_2, \dots, \delta_K)$, with $\delta_j = 1, \forall j = 1, 2, \dots, K$; then, we can get the prior density as:

$$\begin{aligned} g(p_1, p_2, \dots, p_K | \delta_1, \delta_2, \dots, \delta_K) &= \frac{1}{B(\boldsymbol{\delta})} \prod_{j=1}^K p_j^{\delta_j-1} = \frac{\Gamma(\sum_{j=1}^K 1)}{\prod_{j=1}^K \Gamma(1)} \prod_{j=1}^K p_j^0 \\ &= \frac{\Gamma(K)}{\prod_{j=1}^K \Gamma(1)} = \frac{(K-1)!}{\prod_{j=1}^K (1)} = (K-1)!, \end{aligned}$$

where the normalized constant $B(\boldsymbol{\delta})$ is a multinomial beta function that could be expressed as the following gamma function:

$$B(\boldsymbol{\delta}) = \frac{\prod_{j=1}^K \Gamma(\delta_j)}{\Gamma(\sum_{j=1}^K \delta_j)}, \delta = (\delta_1, \delta_2, \dots, \delta_K).$$

Appendix B. Some direction for the derivation of Full Conditional Posterior

Appendix B gives an example of a derivation to get the full conditional posterior for one of the MSNBurr-MM parameters. For example, we derived the full conditional posterior for parameter $\mu_j, j = 1, 2, \dots, K$. The full conditional posterior μ_j for certain j^{th} cluster is given by:

$$f(\mu_j | \sigma_j, \gamma_j, p_j, \mathbf{z}, \mathbf{y}) = f(\mathbf{y} | \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\gamma}, \mathbf{z}) g(\mu_j) = \prod_{i=1}^N \left(\frac{\kappa_j}{\sigma_j} \exp \left(-\kappa_j \left(\frac{y_i - \mu_j}{\sigma_j} \right) \right) \left(1 + \frac{\exp \left(-\kappa_j \left(\frac{y_i - \mu_j}{\sigma_j} \right) \right)}{\gamma_j} \right)^{-(\gamma_j + 1)} \right)^{z_{ij}} \times \frac{1}{\sqrt{2\pi}\varphi_j} \exp \left(\frac{1}{2} \left(\frac{\mu_j - \eta_j}{\varphi_j} \right)^2 \right). \quad (A2)$$

To simplify the derivation, we processed Equation (A2) in the form of the logarithm as follows:

$$\begin{aligned} \ln f(\mu_j | \sigma_j, \gamma_j, p_j, \mathbf{z}, \mathbf{y}) &= \sum_{i=1}^N z_{ij} \frac{\kappa_j}{\sigma_j} \exp \left(-\kappa_j \left(\frac{y_i - \mu_j}{\sigma_j} \right) \right) \left(1 + \frac{\exp \left(-\kappa_j \left(\frac{y_i - \mu_j}{\sigma_j} \right) \right)}{\gamma_j} \right)^{-(\gamma_j + 1)} \\ &\quad + \ln \left(\frac{1}{\sqrt{2\pi}\varphi_j} \exp \left(\frac{1}{2} \left(\frac{\mu_j - \eta_j}{\varphi_j} \right)^2 \right) \right) = \sum_{i=1}^N z_{ij} \left(\ln \left(\frac{\kappa_j}{\sigma_j} \right) - \kappa_j \left(\frac{y_i - \mu_j}{\sigma_j} \right) - (\gamma_j + 1) \ln \left(1 + \frac{\exp \left(-\kappa_j \left(\frac{y_i - \mu_j}{\sigma_j} \right) \right)}{\gamma_j} \right) \right) \\ &\quad \times \ln \left(\frac{1}{\sqrt{2\pi}\varphi_j} \right) - \frac{1}{2} \left(\frac{\mu_j - \eta_j}{\varphi_j} \right)^2. \end{aligned} \quad (A3)$$

Equation components that do not contain μ_j can be simplified into constants; therefore, Equation (A3) can be written as:

$$\begin{aligned} \log f(\mu_j | \sigma_j, \gamma_j, p_j, \mathbf{z}, \mathbf{y}) &= \text{constant} - \sum_{i=1}^N z_{ij} \left(\kappa \left(\frac{y_i - \mu_j}{\sigma_j} \right) \right) - \frac{1}{2} \left(\frac{\mu_j - \eta_j}{\varphi_j} \right)^2 + \\ &\quad \sum_{i=1}^N z_{ij} \left(-(\gamma_j + 1) \ln \left(1 + \frac{\exp \left(-\kappa \left(\frac{y_i - \mu_j}{\sigma_j} \right) \right)}{\gamma_j} \right) \right). \end{aligned}$$

In the same way, a full conditional posterior for parameter σ_j and γ_j can be obtained.

References

1. Global Cancer Observatory. Available online: <http://gco.iarc.fr/today/data/factsheets/populations/360-indonesia-fact-sheets.pdf> (accessed on 16 April 2020).
2. Grün, B. Model-based Clustering. Model-based Clustering. In *Handbook of Mixture Analysis*; CRC Press: Boca Raton, FL, USA, 2018; pp 163–198.
3. Bruse, J.L.; Zuluaga, M.A.; Khushnood, A.; McLeod, K.; Ntsinjana, H.N.; Hsia, T.; Sermesant, M.; Pennec, X.; Taylor, A.M.; Schievano, S. Detecting Clinically Meaningful Shape Clusters in Medical Image Data: Metrics Analysis for Hierarchical Clustering Applied to Healthy and Pathological Aortic Arches. *IEEE Trans. Biomed. Eng.* **2017**, *64*, 2373–2383.
4. Pestunov, I.A.; Rylov, S.A.; Berikov, V.B. Hierarchical Clustering Algorithms for Segmentation of Multispectral Images. *Optoelectron. Instrum. Data Process.* **2015**, *51*, 329–338.

5. Rohith, J.; Ramesh, H. Colour Based Segmentation of a Landsat Image Using K-Means Clustering Algorithm. *J. Imag. Process. Pattern Recogn. Progress* **2017**, *4*, 31–38, doi:10.37591%2Fjoipprp.v4i3.1133
6. Muruganandham, S.K.; Soby, D.; Nallusamy, S.; Mandal, D.K.; Chakraborty, P.S. Study on Leaf Segmentation using K-Means and K-Medoid Clustering Algorithm for Identification of Disease. *Indian J. Public Health Res. Dev.* **2018**, *9*, 289–293.
7. Huang, H.; Meng, F.; Zhou, S.; Jiang, F.; Manogaran, G. Brain Image Segmentation based on FCM Clustering Algorithm and Rough Set. *IEEE Access* **2019**, *7*, 12386–12396.
8. Oh, M.S.; Raftery, A.E. Model-Based Clustering with Dissimilarities: A Bayesian Approach. *J. Comput. Graph. Stat.* **2007**, *16*, 559–585.
9. Greve, B.; Pigeot, I.; Huybrechts, I.; Pala, V.; Börnhorst, C.A. Comparison of Heuristic and Model-based Clustering Methods for Dietary Pattern Analysis. *Public Health Nutr.* **2016**, *19*, 255–264. doi:10.1017/S1368980014003243
10. Rasmussen, C.E. The Infinite Gaussian mixture model. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2000; pp. 554–560.
11. Ji, Z.; Huang, Y.; Sun, Q.; Cao, G.A. Spatially Constrained Generative Asymmetric Gaussian mixture model for Image Segmentation. *J. Vis. Commun. Image Represent.* **2016**, *40*, 600–626.
12. Zhu, H.; Pan, S.; Xie, Q. Image Segmentation by Student's-t Mixture Models Based on Markov Random Field and Weighted Mean Template. *Int. J. Signal. Process. Imag. Process. Pattern Recogn.* **2016**, *9*, 313–322.
13. Franczak, B.C.; Browne, P.; McNicholas, P. Mixtures of Shifted Asymmetric Laplace Distribution. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 1149–1157.
14. Deledalle, C.A.; Parameswaran, S.; Nguyen, T.Q. Image denoising with generalized Gaussian mixture model patch priors. *SIAM J. Imag. Sci.* **2018**, *11*, 2568–2609.
15. Fernandez, C.; Steel, M.F.J. On Bayesian Modelling of Fat Tails and Skewness. *J. Am. Stat. Assoc.* **1998**, *93*, 359–371.
16. Iriawan, N. Computationally Intensive Approaches to Inference in Neo-Normal Linear Models. Ph.D. Thesis, Curtin University of Technology, Perth, Australia, 2000.
17. Iriawan, N.; Pravitasari, A.A.; Fithriasari, K.; Irhamah; Purnami, S.W.; Ferriastuti, W. Comparative Study of Brain Tumor Segmentation using Different Segmentation Techniques in Handling Noise. In Proceedings of the 2018 International Conference on Computer Engineering, Network and Intelligent Multimedia (CENIM), Surabaya, Indonesia, 26–27 November 2018; IEEE: Surabaya, Indonesia 2018; pp 289–293.
18. Choir, A.S.; Iriawan, N.; Ulama, B.; Dokhi, M. MSEpBurr Distribution: Properties and Parameter Estimation. *Pakistan J. Stat. Oper. Res.* **2019**, *15*, 179–193.
19. Box, G.P.E.; Tiao, G.C. *Bayesian Inference in Statistical Analysis*, 1st ed.; Addison Wesley Publishing Company: Reading, UK, 1973.
20. Azzalini, A.A. Class of Distribution which Includes the Normal Ones. *Scand. J. Stat.* **1985**, *12*, 171–178.
21. Anderson, E.; Nguyen, H. When can we improve on sample average approximation for stochastic optimization? *Oper. Res. Lett.* **2020**, *48*, 566–572.
22. Gelman, A.; Carlin, J.B.; Stern, H.S.; Dunson, D.B.; Vehtari, A.; Rubin, D.B. *Bayesian Data Analysis*, 3rd ed.; CRC Press: Boca Raton, FL, USA, 2014.
23. Prasetyo, R.B.; Kuswanto, H.; Iriawan, N.; Ulama, B.S.S. Binomial Regression Models with a Flexible Generalized Logit Link Function. *Symmetry* **2020**, *12*, doi:10.3390/sym12020221.
24. Pravitasari, A.A.; Iriawan, N.; Safa, M.A.I.; Irhamah; Fithriasari, K.; Purnami, S.W.; Ferriastuti, W. MRI-Based Brain Tumor Segmentation using Modified Stable Student's t from Burr Mixture Model with Bayesian Approach. *Malays. J. Math. Sci.* **2019**, *13*, 297–310.

