3D U-Net Based Brain Tumor Semantic Segmentation Using Various Evaluation Metrics And Data Augmentation

Wagaye Tadele Kussa[1], Yeabsira Mengistu Dana[1], Prof. Hitesh Kag[1]

[1] Department of Computer Engineering – Artificial Intelligence, Marwadi University, India

Abstract- Numerous people worldwide lose their lives each year as a result of brain tumors. Glioma is a form of brain tumor that can have dangerous side effects and symptoms. The biological subtypes of glioblastoma, a high-grade glioma tumor, are Whole Tumor (WT), Tumor Core (TC), and Enhancing Tumor (ET). This study presents 3D U-Net semantic segmentation for accurately segmenting tumor sections from multimodal magnetic resonance data (MRI). In this study, we applied on-the-fly data augmentation (Rotation in random angles between 0, 90, 180, and 270 and Flipping in three axes with different probabilities) to the Multimodal Brain Tumor Segmentation Challenge (BraTS 2020) dataset and trained multiple 3D U-Net models using the IOU score, F-Score, and Dice Score with dense volumetric learning mechanisms. Without employing transfer learning, we trained six alternative models from beginning to end using the following particular assessment metrics: Model 1 was trained using Dice and IOU score, followed by Model 2 which used Dice, F-Score, and IOU score, Model 3 which used IOU score and Dice for each class (WT, TC, and ET), Model 4 which used Dice and F-Score, Model 5 which was trained similarly to Model 4 but with the addition of on-the-fly data augmentation, and Model 6 which was trained using Dice and IOU as the evaluation metric but with gradually increasing dropout and kernel initializer included. Our approach in all of the models achieved Dice scores of 90.13%, 88.83%, 85.81%, 90.25%, 87.35%, and 89.69% respectively.

Index Terms- 3D U-Net, Data augmentation, 3D brain tumor segmentation, Deep learning, Semantic Segmentation, BraTS

1. Introduction

One of the most fatal brain cancers is glioma, which may seriously harm the neurological system and put patients at risk[1]. Due to the high frequency of glioma, early diagnosis, treatment, and intervention are popular study subjects[2]. For the purpose of diagnosing a brain tumor, many MRI sequences are recorded. T1-weighted (T1), T1-weighted with contrast enhancement (T1C), T2-weighted (T2), and T2-weighted with fluid-attenuated inversion recovery (FLAIR)[3]. An example of a multimodal MR scan with a corresponding segmented glioma is shown below in **Figure 1**. Edema (ED), enhancing tumor (ET), and necrotic core and non-enhancing tumor (NCR/NET) are the three non-overlapping subregions of glioma which exhibit many biological characteristics with three further regions of interest, whole tumor (WT), tumor core (TC), and extra tumor (ET)[4].

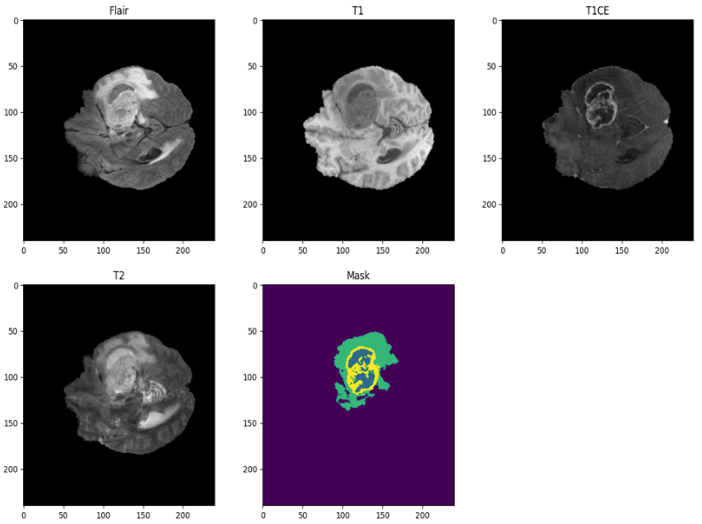

Figure 1: An example of multimodal MR images (Flair: First image, T1: Second image, T1CE: Third image, T2: Fourth image, and Mask: Last image); where the mask shows three partially overlapping interest areas (WT, TC, and ET) represented by TC: Blue, ET: Yellow, WT: Green, and the background Purple.

Manual brain tumor segmentation from MRI images requires a lot of effort and is subject to human-made errors. Researchers are working to create automated segmentation techniques that can precisely and effectively segment brain tumors from MRI data. Convolutional Neural Networks (CNN) is one such approach, where deep learning models have been demonstrated to have amazing performance in medical picture interpretation. We have used basic U-Net Convolutional Neural Network (CNN) architecture and tweaked it to create our 3D U-Net. Olaf Ronneberger, Philipp Fischer, and Thomas Brox introduced the very first U-Net architecture in 2015[5]. Due to its capacity to manage tiny, noisy structures and preserve spatial resolution during the segmentation process, the U-Net architecture is particularly well-suited for medical picture segmentation applications[6].

2. Dataset, Preprocessing, and Data augmentation

The MR images used in this project are from the BraTS 2020 challenge[7]. The dataset includes two folders: one training dataset and another validation dataset. There are 369 folders in total, with five Tii files in each. Each of these Tii files represents T1, T1CE, T2, FLAIR, and segmentation (Mask). We imported the split folder library to split the training dataset into training and validation data. The validation folder provided by BraTS was only for evaluation purposes, and the segmentation files were missing[8]. We performed the following preprocessing steps before we fed these data to our model.

We utilized the BraTS MR images that were originally 244x244 in size. However, to eliminate the non-informative regions surrounding the tumor, we performed patch extraction and cropped the images to 128x128. The resulting data consisted of 155 slices of 128x128 images per subject. To reduce the dimensionality of the data, we further discarded a few slices with no relevant information. To ensure the uniformity of the features, we applied a scaler.fit_transform() function to normalize the data and made the mean zero and variance one. The final shape of each modality was 128x128x128.

For the subsequent analysis, we loaded the data using the nibabel library and converted the data into arrays using the numpy library. The resulting arrays were saved as a single npy file. Finally, we split the preprocessed data into a 75% training dataset and a 25% validation dataset. The training dataset consisted of 258 npy images with the shape of (128,128,128,4) which included 4 modalities (T1, T1CE, T2, FLAIR) and 258 npy masks with the shape of (128,128,128,4) representing 4 tumor classes. The validation dataset consisted of 86 npy images with the shape (128,128,128,4), containing 4 modalities, and 86 npy mask files with the shape (128,128,128,4). The preprocessed data consisted of 344 folders in total.

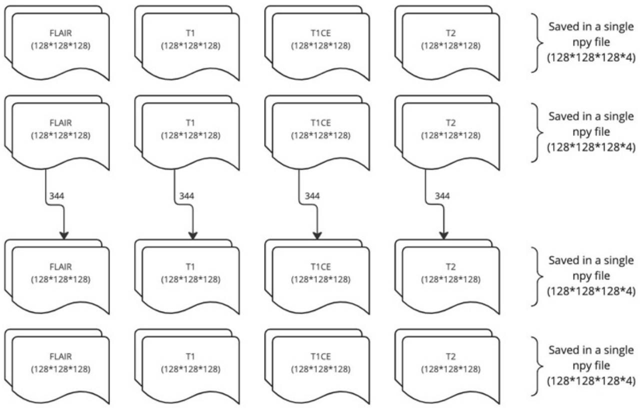

Figure 1: *This is a representation of preprocessed training and validation images only (masks are not included here). Each line corresponds to a training image folder containing a single npy file which includes the four modalities in it. The same representation works for masks too, but the modalities are replaced by classes (WT, ET, TC, and Background).*

In order to reduce overfitting and improve one of the six model's robustness. We were able to broaden the variety of our training data by implementing data augmentations on the fly during training, which improved the model's performance, and reduced overfitting[9]. Additionally, we were able to train the model with less data and still get outstanding results because of the usage of data augmentation. The detailed algorithm for the data augmentation algorithm is shown below.

Data augmentation steps applied

Step 1:	Rotate in 0, 90, 180 and 270 P=0.5
Step 2:	Flip in axis 0 P=1/3
Step 3:	Flip in axis 1 P=1/3
Step 4:	Flip in axis 2 P=1/3

Table 1: Illustration of details implemented during data augmentation in one of the six models trained

3. Related work

Deep learning has recently made incredible strides in a variety of medical picture segmentation tasks, especially when there is a wealth of training data available. Numerous pre-operative multi-modal MR images and related manual annotations of brain tumors have been made available as part of the multi-modal Brain Tumor Segmentation (BraTS) challenge[10]. Either 2D slice-based or 3D patch-based CNN techniques are possible. In 2D CNN techniques, a brain tumor is independently predicted for each slice of a 3D volume that has been partitioned into several 2D slices[11]. For instance, Caver et al. used three 2D U-Nets to slice-by-slice segment WT, TC, and ET individually[12].

Another common neural network approach for segmenting images is 3D U Net, which has an encoder/decoder structure and multiple connected convolutional layers in the encoder module. This approach aims to gradually reduce the spatial dimension of feature maps and capture more high-level semantic features that have been trained to be very effective at classifying images at a pixel level. The decoder module performs upsampling to layers in order to restore the spatial information and object representation[13]. J. Long et al. [14] described an inherent conflict between collecting semantic and spatial information due to the challenge of finding a model that can make local predictions while taking general structure into account. Some works solve this issue by combining both feature maps and adding "skip connections" across layers from the encoder and decoder modules. Combining local and spatial information is the basic concept. The advantage of skip connections was first utilized by Drozdzal et al. [15]. The memory and processing performance requirements for 3D U-Net are quite high. Therefore, using the complete 3D volume as the input and output may not be practical. One solution to this problem is to construct label maps for the smaller 3D patches by extracting them from the network input[16]. Many publishers indicated the use of an ensemble of 3D U Net models performs better than a single trained 3D U Net model. A. Myronenko[17] suggested adding a variational autoencoder branch to an encoder/decoder model, In order to recreate the input picture while segmenting it. This would force the network's previous layers to extract more useful features. In order to show that a well-trained U-Net network may produce competitive results, Isensee et al.[18] proposed the adoption of a 3D U-Net architecture with small but significant alterations, such as the introduction of instance normalization [19] and leaky ReLU. In another work by Zhou et al. [20] ensemble of several models was presented to segment the three distinct tumor regions in a cascade while taking into account multiscale context data.

In a recent study by Xue Feng et al.[21], a deep learning model for segmenting brain tumors was constructed utilizing a 3D U-Net with modifications to training and testing methods, network architecture, and model parameters. In order to eliminate random mistakes and enhance performance, they used a group of different models, and as a result, they placed ninth in the 2018 Multimodal Brain Tumor Segmentation (BraTS) competition. According to reports, the mean Dice scores for the tumor core (TC), whole tumor (WT), and enhancing tumor (ET) were 75.40%, 87.80%, and 79.90%, respectively.

The Multimodal Brain Tumor Segmentation Challenge (BraTS) 2020 training dataset was used in another study by Theophraste Henry et al. [22] to train several U-net-like neural networks to automate and standardize brain tumor segmentation. Techniques like stochastic weight averaging and deep supervision were used in the training. Brain tumor segmentation maps were produced using two distinct ensembles of models from various training pipelines, and the performance of each ensemble for certain tumor subregions was utilized to determine how the two ensembles should be combined. The final test dataset generated a Dice score for the solution of 79%, 89%, and 84%, placing it in the top 10 teams in BraTS 2020 challenge. The study also looked at more intricate training plans and neural network designs but found that they did not significantly boost performance and required longer training periods.

Here we have presented the performance result of the 3D U-Net segmentation task[23] by Theophraste Henry et al. in a tabular form.

Metric (mean)	ET	WT	TC
Dice	0.78507	0.88595	0.84273
Sensitivity	0.81308	0.91690	0.85934
Specificity	0.99967	0.99905	0.99964
Hausdorff (95%)	20.36071	6.66665	19.54915

Table 1: Performance on BraTS 2020 dataset (by Theophraste Henry et al.)

4. Method

The 3D U-Net model used in this paper is based on the well-known U-Net architecture which is suitable for image segmentation tasks[24]. The 3D U-Net model is intended to process image segmentation to remove the background from the target items or regions of interest after receiving 3D medical pictures as input. A number of Convolutional Transpose layers and Max Pooling layers are used in the model to extract features, which are then refined and up-sampled by a number of Convolutional Transpose layers and Up-sampling layers. Additionally, the model has Batch Normalization layers to enhance performance and Dropout layers to avoid overfitting. To create the final segmentation result, the model includes a convolutional layer with a softmax activation function. The softmax activation function layer $[S(o(x))]_i$ explained by Jun Liu et al.[25] forces the output to be a probability.

$$[S(o(x))]_i = \frac{e^{[(o_i(x))]}}{\Sigma_i^n e^{[(o_i(x))]}} \qquad (1)$$

4.1. Network Architecture

The 3D U-Net architecture shown below in **Figure 3** is composed of multiple blocks of convolutional, max pooling, and transpose convolutional layers, as well as batch normalization, activation, and dropout layers. The convolutional block consists of two 3D convolutional layers, each with a kernel size of (3,3,3) and a stride of (1,1,1). The number of filters used in each convolutional layer can be specified using the number of filters argument. The output of each convolutional layer is passed through a batch normalization layer if batch normalization is specified as True, and an activation layer with a rectified linear unit (ReLU) activation function. The encoder network uses max pooling layers to gradually down-sample the spatial dimensions of the input data. Then the max pooling layers use a pool size of (2,2,2) and a stride of 2. Dropout layers are also added after each max pooling layer to help prevent overfitting. The dropout rate can be specified using the dropout argument.

The decoder network uses transpose convolutional layers to up-sample the feature maps from the encoder[13]. The transpose convolutional layers use a kernel size of (3,3,3) and a stride of (2,2,2). After each transpose convolutional layer, skip connections are used to concatenate the decoder feature maps with corresponding feature maps from the encoder. This helps the network retain important information from the encoder while generating the final segmentation as explained by Drozdzal et al. [15]. The final layer in the network is a 1x1x1 convolutional layer with a softmax activation function, which generates the segmentation mask with 4 classes[25]. The 3D U-Net model takes a 3D input tensor and outputs a 3D tensor of segmentation masks with 4 classes. The architecture is flexible and can be adjusted based on the specific application requirements by modifying the number of filters, dropout rate, batch normalization, and other hyperparameters.

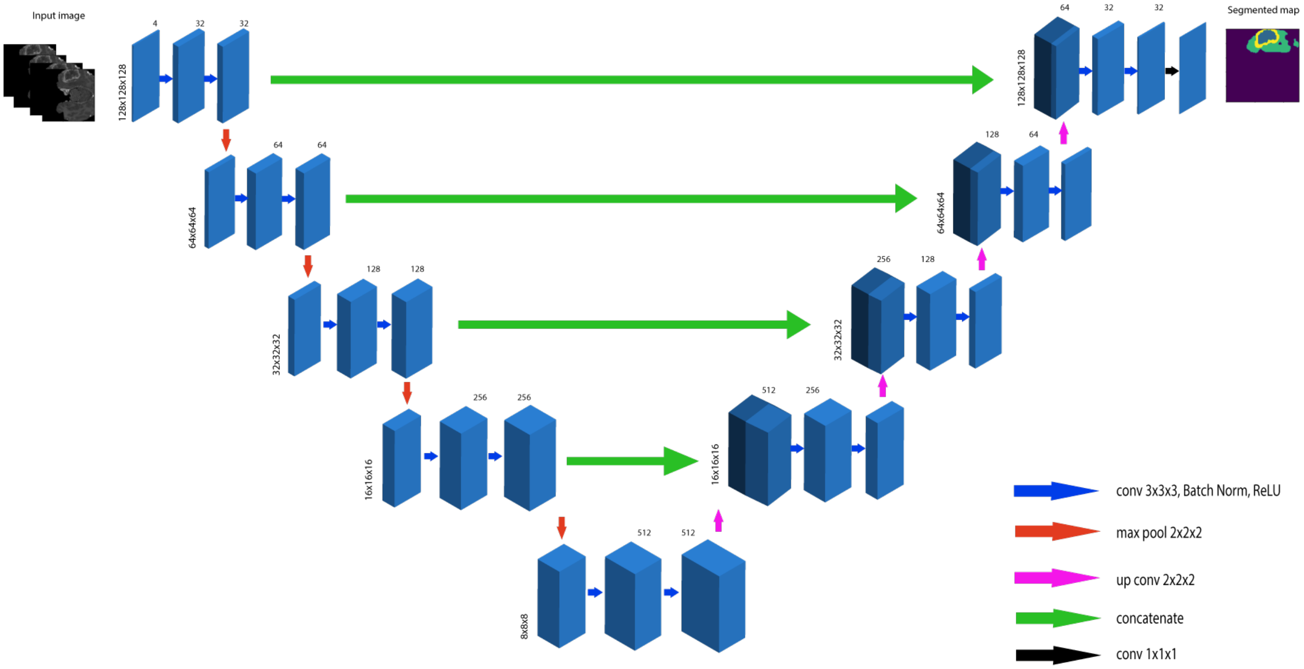

Figure 2: A 3D U-Net architecture with encoders and decoders

4.2. Evaluation metrics

The predictions are processed through a soft-max layer that outputs the likelihood that each voxel belongs to the foreground or the background. It is normal for the anatomy of interest to occupy only a relatively tiny portion of the scan in medical volumes like the ones we are analyzing in our work. As a result, the learning process frequently becomes stuck in the local minima of the loss function, producing a network with predictions that are heavily skewed in favor of the background[26]. In order to measure the similarity between the predicted segmentation and the actual segmentation, we applied the Dice coefficient, a commonly used assessment metric in medical image segmentation. In this work, the effectiveness of our 3D U-Net based brain tumor segmentation model was evaluated using the Dice coefficient. The Dice coefficient is defined as the ratio of the number of voxels in the intersection of the predicted and ground truth masks, to the total number of voxels in both masks. Mathematically, the Dice coefficient per each voxel is represented as:

$$Dice\ Score = \frac{2\Sigma P_i G_i}{\Sigma P_i^2 + \Sigma G_i^2} \qquad (2)$$

Where the sums run over each voxel of predicted segmentation P_i and ground truth G_i[26]. In this implementation, the dice coefficient is calculated as the ratio of the sum of the product of the predicted and ground truth masks to the sum of the squares of both masks, with a small epsilon value added to the numerator and denominator to prevent division by zero. The mean Dice coefficient over all slices is returned as the final evaluation metric. In addition to the Dice coefficient, we have also used the dice coefficient loss as an optimization objective to train the 3D U-Net model. The dice coefficient loss is defined as 1 minus the Dice coefficient and it measures the difference between the predicted and ground truth masks. Minimizing the dice coefficient loss during training would result in a model that produces masks with a higher degree of similarity to the ground truth masks. Mathematically, the Dice coefficient loss per each voxel is represented as:

$$Dice\ Loss = 1 - \frac{2\Sigma P_i G_i}{\Sigma P_i^2 + \Sigma G_i^2} \qquad (3)$$

4.3. Data Augmentation and Data Generator

Due to ethics and data protection laws (such as GDPR), it is more challenging for medical imaging tasks than for other computer vision tasks to acquire a large number of training images. For this reason, data augmentation is even more crucial to increase the number of images for training and testing. There are surprisingly few studies that discuss the value of various augmentation techniques (such as rotations, random flips, scaling, and elastic deformations) for convolutional neural networks (CNNs) training, particularly for 3D CNNs, Marco

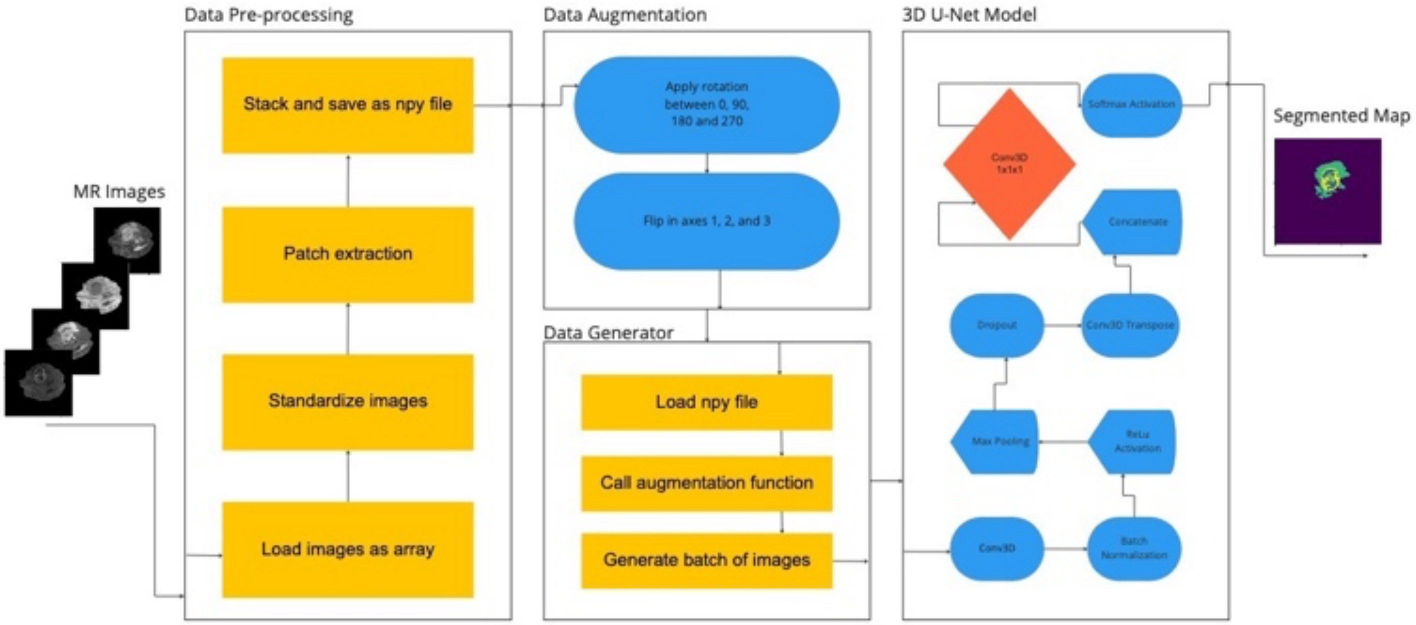

Figure 3: Flowchart of the whole segmentation implementation (step 1: Data preprocessing, step 2: Data augmentation step 3: Data generator, step 4: 3D U-Net model)

Domenico Cirillo et al.[27]. The goal of data augmentation is to artificially increase the size of the training dataset by creating modified versions of the original images. This can help reduce overfitting and improve the generalization ability of the model. In the case of brain tumor semantic segmentation, data augmentation can help the model handle variations in imaging acquisition and variability in the appearance of the tumor. In this study, we applied several data augmentation techniques, including rotations at random angles of 0, 90, 180, or 270 with a probability of one, and flipping at three axes with a one-third probability each, to the 3D MR images of the brain tumors used for training. This was done using an open-source custom library, Volumentations by Solovyev [28] and Albumentations by Buslaev et al.[29]. The rotation angle and flipping axes were chosen to increase the diversity of the augmented images and avoid over-augmentation.

Additionally, a random seed was set for each image to ensure that the augmentation was unique to each image. To efficiently handle the large size of the 3D MR images, a data generator was implemented. A custom function was designed to load the images in batches and augment them on-the-fly during training.
This allows for efficient memory usage and faster training times, as the augmented images are not stored in memory. The data augmentation and data generator methods used in this study play a crucial role in improving the performance of the 3D U-Net model for brain tumor semantic segmentation. The implementation of these techniques allows for efficient use of resources and faster training times, while also improving the generalization ability of the model[30].

5. Training and Results

REFERENCES

[1] G. O. Young, “Synthetic structure of industrial plastics (Book style with paper title and editor),” in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.

[2] W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

[3] H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985, ch. 4.

[4] B. Smith, “An approach to graphs of linear forms (Unpublished work style),” unpublished.

[5] E. H. Miller, “A note on reflector arrays (Periodical style—Accepted for publication),” *IEEE Trans. Antennas Propagat.*, to be published.

[6] J. Wang, “Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication),” *IEEE J. Quantum Electron.*, submitted for publication.

AUTHORS

First Author – Author name, qualifications, associated institute (if any) and email address.
Second Author – Author name, qualifications, associated institute (if any) and email address.
Third Author – Author name, qualifications, associated institute (if any) and email address.

Correspondence Author – Author name, email address, alternate email address (if any), contact number.