Inverted Cone Convolutional Neural Network for Deboning MRIs

Oliver Palumbo, Dimah Dera and Nidhal C. Bouaynaya
Department of Electrical and Computer Engineering,
Rowan University
Glassboro, NJ

Hassan M. Fathallah-Shaykh
Department of Neurology
University of Alabama at Birmingham School of Medicine
Birmingham, AL

***Abstract*—Data plenitude is the bottleneck for data-driven approaches, including neural networks. In particular, Convolutional Neural Networks (CNNs) require an abundant database of training images to achieve a desired high accuracy. Current techniques employed for boosting small datasets are data augmentation and synthetic data generation, which suffer from computational complexity and imprecision compared to original datasets. In this paper, we intercalate prior knowledge based on spatial relation between images in the third dimension by computing the gradient of subsequent images in the dataset to remove extraneous information and highlight subtle variations between images. The approach is coined "Inverted Cone" because the volume of brain images below the level of the eyes is ordered to form an inverted cone geometry.**

The application explored in this work is deboning, or brain extraction, in brain magnetic resonance imaging (MRI) scans. The difficulty of obtaining ground truth for this application prevents the ability of obtaining a large quantity of training images to train the CNN. We considered a limited dataset of 23 patients with and without malignant glioblastoma. Deboning was performed by employing an optimized CNN architecture with and without the Inverted Cone processing. The classic CNN without prior knowledge achieved a validation accuracy of 77%, while the Inverted Cone CNN model achieved a validation accuracy of 86% in a dataset of 451 brain MRI slices.

I. Introduction

Deep learning has been widely utilized in object detection and recognition. Convolutional neural networks (CNNs) allow processing and analyzing large sets of image data into classification of predefined classes. Beginning with the AlexNet architecture developed in 2012 by the SuperVision group [1], CNNs have been proven to outperform classical modeling for the purpose of object detection. AlexNet ranked in the top-5 models for the *ImageNet Large Scale Visual Recognition Challenge* with only 15.3% error by classifying 1.2 million images into 1,000 categories [1]. Since then, deep learning for object recognition has been expanded into many different applications, such as pothole detection for intelligent transportation systems [2] and medical brain tumor segmentation as an aide for medical diagnoses [3], [4]. Image segmentation is a concentrated application of object detection that distills an image into a series of patches. Each patch is then processed by the network in order to classify the central pixel of that patch. Ultimately, every pixel in an image could be classified to divide a single image into multiple classes.

The Multimodal Brain Tumor Segmentation (BRATS) competition [5] strives to improve the brain tumor segmentation application by evaluating a set of image segmentation models, both classical and deep learning, to determine which technique produces the highest accuracy. The task is to develop a model that detects and classifies 5 distinct regions in a brain magnetic resonance imaging (MRI) scan (normal tissue, necrosis,edema, non-enhancing, and enhancing tumor). A CNN model was awarded first place in the 2015 BRATS challenge with Dice Similarity Coefficients of 0.88, 0.83, and 0.77 in the complete, core, and enhanced regions, respectively, as computed by the BRATS organization [4].

A vital issue that arises in specific applications such as brain MRI segmentation or pothole detection is the requirement of a large database of images to train the network. The BRATS competition employed a training dataset comprising 276 patients' four modalities MRIs with each MRI modality containing approximately 150 images. Often times, especially for medical image applications, image datasets are limited with a small number of images available for training the network, which can result in overfitting of the model to the images in the training database and not being able to generalize well on unseen images.

Several preprocessing techniques have been developed to alleviate some of the issues that arise with the limited datasets. Data augmentation is one way to artificially increase the size of a database by duplicating and performing transformations on the original dataset [6], [7] and [8]. For example, one could perform a series of $90°$, $180°$ and $270°$ rotations on each image to effectively quadruple the size of their database [4]. Furthermore, these transformations would make the model rotationally invariant, allowing accurate object detection regardless of how the object is oriented within the test images. However, data augmentation increases the computational complexity, which is undesirable especially for medical image analysis intended for diagnosis purposes. Additionally, a database can be expanded through synthetic data generation [9]. Originally proposed as a solution to imbalanced classes, the Synthetic Minority Over-sampling technique has been utilized to increase the amount of training data in an underrepresented class [10], [11]. Each training class could be manipulated through this technique until all classes contain an equal quantity of training images. This concept could then be expanded to the dataset

as a whole; generating synthetic data to increase the total size of the database. However, medical images are usually very difficult to imitate and include critical information that need to be extracted. By synthesizing these medical images, we may be loosing some information that might help in diagnosis and treatment.

Although CNN architectures are robust for general object recognition with large and diverse training data, faults arise when specific applications lack a suitable database. The standard approach to dealing with this deficiency would be to artificially expand the training data through data augmentation and synthetic data generation. Both of these techniques involve adding additional data points to the database in the preprocessing step that do not truly exist in the reality of the problem. The solution proposed in this paper is to impose a constraint on the dataset in both preprocessing and during the testing of the CNN architecture based upon prior knowledge of the dataset.

The application explored in this paper is the deboning process for brain MRI scans. Prior to diagnoses, skull structures that are unrelated to the illness under evaluation must be removed from each image. Currently, this process is performed either by hand or by manually adjusting a single variable within preprocessing software, such as *FSL* [12]. This variable must be readjusted for every image in each MRI set, thus increasing the time and labor for each diagnosis. The proposed modification, named the Inverted Cone Method, utilizes the known order of images in an MRI dataset to remove the most complex skull structures prior to processing by the CNN.

II. The Inverted Cone CNN

A. The Inverted Cone Method

The Inverted Cone method relies on a sequentially ordered dataset, such as in a stream of video or a series of MRI scans, in order to isolate relevant information in more complex images. In the deboning application, all images in an individual patient's MRI scan are ordered in layers from the base to the top of the skull and separated by a constant thickness per slice.

MRI scan slices that are taken from the top to the middle of the skull are easily segmented by both visual inspection and through CNN processing (Fig. 1(a)). At these locations, the skull is present in a well-defined ring around the brain. As the scans descend further into the skull, sinus cavities begin to appear in the skull structure as the area occupied by the brain shrinks (Fig. 1(b)). MRI slices that have been taken closer to the base of the skull introduce highly irregular areas and deviate greatly from the typical central slice in Fig. 1(a). In these lower images, the sinus cavities and eye sockets create more complex structures to classify. An example of a complex MRI slice containing eye sockets can be seen in Fig. 1(b). The ordered and related nature of the brain MRI images allows the use of preprocessing to remove the most difficult to classify sections of the skull. By working from the central slices outward, the MRI images with the largest area of brain can be leveraged to remove extraneous skull structures in the more complex scans.

The difference is taken between each image with a larger area of brain and the next image in the dataset to highlight the relevant area of analysis in that next image. The skull structures that result from this difference can be removed from the subsequent images to reduce presence of the most difficult to classify areas.

B. The Inverted Cone CNN

To construct the Inverted Cone CNN, a preprocessing system was created that must be applied during both the training and the testing phases of the CNN. The training set is prepared with this system prior to training the CNN, and then applied in a feedback loop during the testing of the network model. A block diagram depicting the overall process for the Inverted Cone CNN can be seen in Fig. 2.

First, the training set of images must be preprocessed through the Inverted Cone method. The preprocessing was performed by finding the MRI slice for a specific patient with the largest area of brain in the ground truth. The resultant image would be from the central area of the MRI volume, and the difference would then be taken between this image and the subsequent image for that patient's MRI. This difference would include all irrelevant skull structures that fall outside the immediate area surrounding the brain. This process was then performed upon all of the following images; thereby, the MRI sets for each patient were simplified based on a central slice for each patient. A CNN architecture would then be trained upon this modified dataset.

During the testing phase, the Inverted Cone method was applied in a similar fashion. When classifying a new set of MRI scans, the scans were input into the system starting with the slices that are taken at the top of the skull. The area of the brain that is identified in each slice will be stored and compared to the area in the subsequent slice, until the central slice with the largest area of brain is discovered. This central slice would then be utilized to identify extraneous skull structures in the following image. Once these structures are removed from the slice, the image would be processed as an input to the network. The segmented result is then fed back as an input to the network to compare with the next image in the set, thus creating a gradient of change around the immediate area of the brain. This process is performed on each subsequent image until the entire MRI set for a patient is classified and deboned.

III. Application To Brain MRI Deboning

A. Database

The Inverted Cone CNN was applied to a database of anonymized gadoliniumenhanced T1-Weighted MRI images of human brain with and without malignant glioblastoma multiforme, a malignant brain tumor. The database, provided by the University of Alabama at Birmingham School of Medicine, was comprised of patient files containing a series of MRI scan slices for each patient. Patient files which held MRI scans that did not equal the common size of 256x256 were removed from the database. The resulting size of the

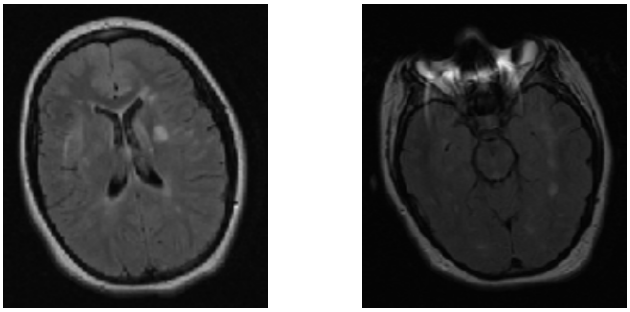

Fig. 1: (a) Central slice of MRI scan; (b) Lower slice of MRI scan

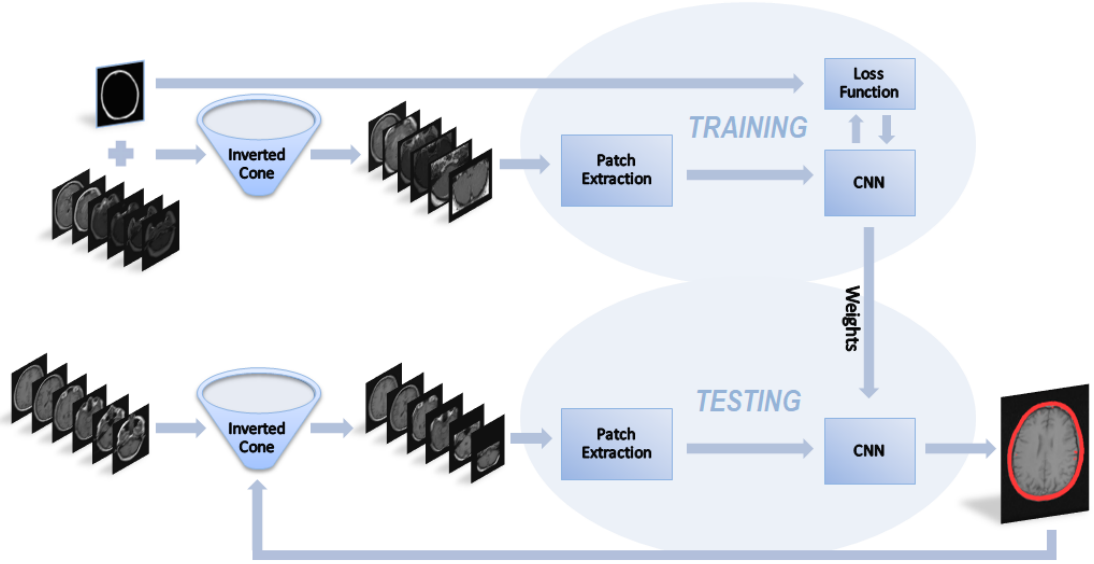

Fig. 2: Block diagram of Inverted Cone CNN

dataset that was utilized in the training of this system was 23 patients with 17-38 slices per MRI scan. Ultimately, the system was trained with 451 MRI slices.

B. Preprocessing

Segmentation is performed to classify multiple objects or classes within a single image. In the deboning application, segmentation could be formulated as a binary classification problem of skull vs. brain. The process of segmentation involves patching to be applied to the dataset to extract local regions of a specified size from a variety of areas. An equal amount of patches are extracted from the training set of images for each of the two classes. Each patch is classified by the central pixel in that patch. Once a balanced library of patches is derived from the dataset, the network is then trained on that library. The weights in the network are adjusted after each pass through the library of patches until all epochs are completed.

After training has been completed, testing can be performed on images that the network has not yet seen in the training set. The input images to the system are decomposed into patches of a specified size and input into the network. Each patch would then be analyzed by the network to classify the central pixel of that patch into one of the predefined classes of the training set. Image segmentation is complete once all pixels have been classified by the network.

For the application of deboning, a library of 300,000 patches were created for a data set consisting of 23 patients and between 17-38 slices per MRI scan. The total number of images in the dataset was 451 MRI slices. Patch sizes of 33x33 and 15x15 were investigated in this model. Larger patch sizes allow for a larger region of features to be analyzed around the pixel being classified which makes the network more robust to local structures, yet increases computational complexity and overfitting due to a loss of resolution. Ultimately, a patch size of 15x15 was chosen for the final model.

The Inverted Cone method was applied to the deboning dataset during preprocessing to remove complex skull structures at the base of the skull. The ground truth for each image following the central slice was used to filter the subsequent slice for that patient. Once the difference was taken between the area of the brain in the previous image with the second

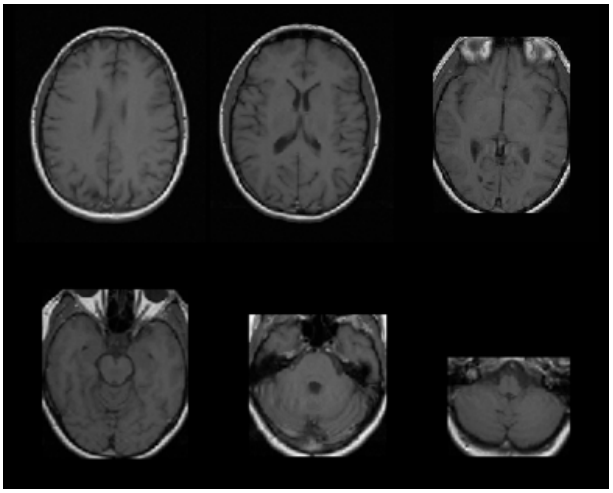

Fig. 3: Brain MRI scans preprocessed by Inverted Cone

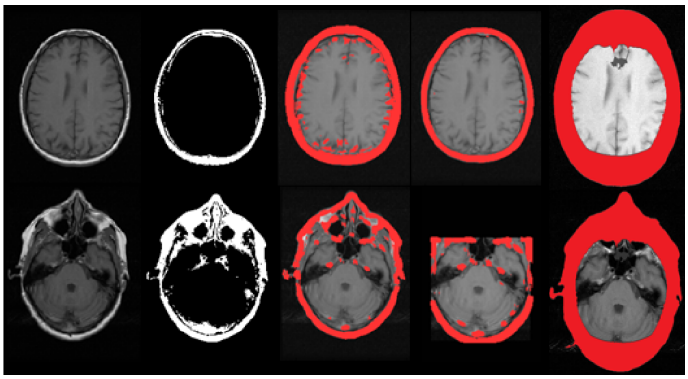

Fig. 4: Segmentation Results: 1st column: Original MRI scans; 2nd column: Ground truth deboning; 3rd column: Segmentation of the standard CNN; 4th column: Segmentation of the Inverted Cone CNN; 5th column: Deboning of the FSL software.

image, that difference could then be removed from the second image as extraneous skull structures. A series of brain MRI slices preprocessed using the Inverted Cone method can be seen in Fig. 3.

C. Deboning Application

The architecture utilized in the Inverted Cone CNN was a three-layer convolutional network followed by two densely connected layers and a classification layer at the output. Three hidden layers in this network were able to extract a better feature-set from the data then a shallow network with only a single layer. An increase in the number of layers past three resulted in a decrease of overall accuracy due to overfitting. A standard ReLU activation function was used in each layer along with batch normalization to prevent overfitting. Batch normalization ensures that the mean activation of the previous layer is close to zero and the standard deviation is close to one. The full list of hyperparameters for this network is displayed in Table 1.

Kernel sizes were chosen to be cascading in size from 7x7 for the first layer, 5x5 for the second layer, and 3x3 for the third layer. Smaller kernel size allowed for a three hidden layer design with a small patch size of 15x15. Larger kernels were employed at the outer layers to extract features with more locality information. The kernels decrease in size in the two subsequent layers in order to reduce the number of weights

TABLE I: Hyperparameters for the Inverted Cone CNN

Hyperparamter	**Value**
Patch dimensions	15x15
Patch quantity	300000
Kernel dimensions	7x7, 5x5, 3x3
Kernel quantity	8, 8, 8
Learning rate	0.001
Decay	0.01
Momentum	0.9
Epochs	10
Batch size	256
Database size	451

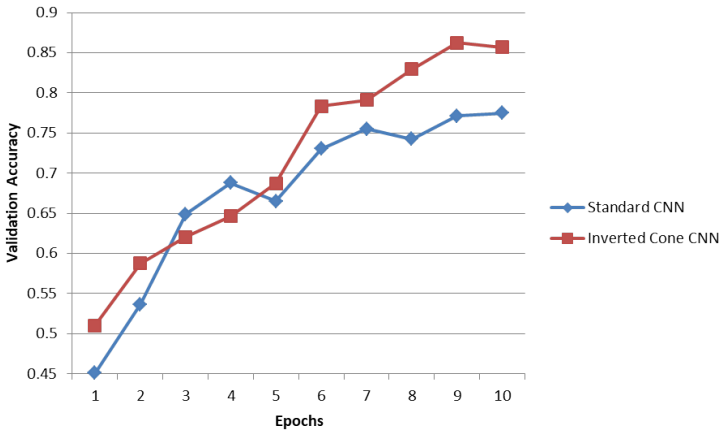

Fig. 5: Validation accuracy for standard and inverted cone CNN

TABLE II: Resultant Loss, Accuracy, and Validation Accuracy for Standard and Inverted Cone CNN

Measurement	Standard CNN	Inverted Cone CNN
Loss	0.5112	0.4333
Accuracy	0.8204	0.8431
Validation Accuracy	0.7748	0.8567

and deter overfitting.

The architecture described in Table 1 was trained with and without using the Inverted Cone method to process the inputs. The results from the Inverted Cone CNN deboning were compared to a standard CNN as well as the widely used *FSL* deboning software. The accuracy measurements for the *FSL* software was acquired by manually adjusting the parameter for the central slice of each test MRI set and using this parameter for all other slices in the set. Accuracy, validation accuracy, and loss measurements for the two CNN techniques are shown in Table 2. The Inverted Cone CNN outperformed the standard CNN model and the *FSL* software on this dataset. Accuracy and validation accuracy were plotted for both CNN models over the 10 epochs, as seen in figure 5.

The Inverted Cone CNN outperformed the standard model in validation accuracy. The results show that utilizing the ordered nature of the brain MRI scans during preprocessing can reduce the complexity of the dataset and provide the system with a greater capacity to learn the data. Figure 4 contains both simple and complex images segmented using the standard CNN model and the modified Inverted Cone CNN. As can be observed, the Inverted Cone CNN was able to segment the validation set more accurately as compared to the ground truth than the standard CNN architecture for both complex and simple MRI slices. In the complex image, the Inverted Cone model was able to identify skull structure immediately bordering the brain more readily than the standard model.

IV. Conclusion

Modifications to training data for a convolutional neural network is vital in situations where the quantity of that data is limited. In the past, data augmentation has been used to multiply the size of the database through duplicating and transforming available images. Synthetic data generation has also been employed to create additional, artificial images to the dataset. Knowledge of intrinsic attributes for a set of images can be leveraged in the training and testing of a network through specific preprocessing operations to increase the overall accuracy of the system. Two such attributes, order and spatial relation, were incorporated into the preprocessing for the application of deboning brain MRI scans.

Since MRI scans are oriented from the top to the base of the skull in sequential order, each scan is a slight gradient from the previous scan. As you descend further into the slices of a scan, the area of the brain occupying the slice grows. Knowledge of the ordered nature of the data along with the relationship between subsequent images allows the Inverted Cone preprocessing method to be employed. Once the slice with the largest area of brain is determined, which will be a central slice in the scan, each slice afterwards can be filtered by the previous slice. The only area of interest – the brain – decreases in size from the central slice to the base of the skull while the size and complexity of the skull structures increase.

By ignoring the skull structures that fall outside of the area occupied by the brain in the previous slice, the classification problem becomes much simpler for the neural network to learn. The most difficult to classify images, which contain sinus cavities, become more similar in shape and relative area of each class when the inverted cone processing is applied; thereby increasing the overall accuracy of the system.

Acknowledgement

This work was supported by the National Science Foundation under Award Numbers NSF DUE-1610911 and NSF ACI-1429467.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks." in *NIPS*, 2012, pp. 1106 – 1114.

[2] Y. J. Cha, W. Choi, and O. Buyukozturk, "Deep learning-based crack damage detection using convolutional neural network," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, pp. 361 – 378, 2017.

[3] P. Moeskops, M. A. Viergever, A. Mendrik, L. S. de Vries, M. J. N. L. Benders, and I. Igum, "Automatic segmentation of mr brain images with a convolutional neural network," *IEEE Transactions on Medical Imaging*, vol. 35, pp. 1252 – 1261, 2016.

[4] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, "Brain tumor segmentation using convolutional neural networks in mri images," *IEEE Transactions on Medical Imaging*, vol. 35, pp. 1240 – 1251, 2016.

[5] B. H. Menze, A. Jakab, and et al, "The multimodal brain tumor image segmentation benchmark (BRATS)," *IEEE Transactions on Medical Imaging*, vol. 34, pp. 1993 – 2024, 2015.

[6] J. Lemley, S. Bazrafkan, and P. Corcoran, "Smart augmentation learning an optimal data augmentation strategy," *IEEE Access*, vol. 5, pp. 5858 – 5869, 2017.

[7] J. Ding, X. Li, and V. N. Gudivada, "Augmentation and evaluation of training data for deep learning," in *IEEE International Conference on Big Data (BIGDATA)*, 2017.

[8] I. Oliveira, J. Medeiros, and V. de Sousa, "A data augmentation methodology to improve age estimation using convolutional neural networks," in *SIBGRAPI Conference on Graphics, Patterns and Images*, October 2016, pp. 88–95.

[9] M. Babaee and A. Nilchi, "Synthetic data generation for X-ray imaging," in *Iranian Conference on Biomedical Engineering (ICBME)*, November 2014, pp. 190–194.

[10] J. W. Anderson, K. E. Kennedy, and L. B. Ngo, "Synthetic data generation for the internet of things," in *IEEE International Conference on Big Data*, October 2014, pp. 171–176.

[11] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell, "Understanding data augmentation for classification: When to warp?" in *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, November 2016, pp. 1–6.

[12] S. Smith, M. Jenkinson, M. Woolrich, C. Beckmann, T. Behrens, H. Johansen-berg, P. Bannister, M. Luca, I. Drobnjak, D. Flitney, R. Niazy, J. Saunders, J. Vickers, Y. Zhang, N. Stefano, J. Brady, and P. Matthews, "Advances in functional and structural MR image analysis and implementation as FSL," *NeuroImage*, vol. 23, pp. 208 – 219, 2004.