The Strategy of Generalization Ability Improvement for Brain Tumor Classification Based on CNNs Model

Yuze Hou

Computer and Informational Science, The Ohio State University, Columbus, the United States
hou.445@buckeyemail.osu.edu

Abstract. Brain tumor is a serious disease that affects lots of people. Traditional methods of tumor detection are time-consuming and subjective. Many studies have demonstrated Convolutional Neural Networks (CNNs) can classify brain tumors with a high accuracy, but they did not focus on the generalization of the model. This study proposes a way to enhance the generalization ability of CNN model for brain tumor classification. Two distinct sets were used in the study, one was designed to be the original dataset, the other was the external dataset. The testing set from the external dataset was split into different proportions (0%, 10%, 20%, 30%, 40%) then incorporated into the training set of original datasets to form 5 different training and testing sets. Five models with the same architecture were trained based on these training sets. The validation set used for training was from the original dataset in order to keep models align with the original distribution. Models were being tested on both the testing set of original datasets and the testing set been split. The 30% model turns out to have the best balance, which indicates that by incorporating a proper amount of external data into the training set, the model's generalization ability could improve with a tradeoff of some accuracy.

Keywords: CNNs, generalization ability, brain tumor

1 Introduction

In recent years, the advent of Convolutional Neural Networks (CNNs) has revolutionized the field of medical image analysis. For instance, researchers have investigated CNN's utilization on Covid19 [1], chest radiology [2], Ophthalmology [3] and rehabilitation training [4]. CNN is suitable for many medical fields, especially for tumor detection and segmentation. In 2023, in the US alone, an estimated 24,810 adults will be diagnosed with tumors of brain or spinal cord, while brain tumors account for 85% to 90% of it [5]. Traditional methods of tumor detection often rely on manual interpretation of radiological images by expert clinicians, which can be time-consuming and subjective. However, the integration of CNNs into medical imaging workflows offers a promising solution to enhance the accuracy and efficiency of brain tumor detection.

CNN is a class of deep learning algorithms designed to automatically learn features from visual data. It's well-suited for tasks such as object recognition, segmentation, and

Y. Wang (ed.), *Proceedings of the 2024 2nd International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2024)*, Advances in Computer Science Research 115,
https://doi.org/10.2991/978-94-6463-540-9_63

classification. In the context of brain tumor detection, CNN can analyze Magnetic Resonance Imaging (MRI) scans with remarkable precision, aiding clinicians in identifying tumor presence, location, and type. Researchers have achieved high accuracy for brain tumor classification task and segmentation task with variations of CNN.

Eldin et al. develop a CNN model based on pre-trained Inception-ResnetV2, which employs adaptive dynamic sine cosine fitness grey wolf optimizer (ADSCFGWO). The researchers name it Brain Tumor Classification Model (BCM-CNN). BCM-CNN came out to be the top in detecting brain tumor among multiple algorithms, including regular CNN, Support Vector Machine (SVM), Decision Tree, Linear Discriminant and K-nearest Neighbor. The study shows that BCM-CNN achieves 0.9998 in terms of accuracy [6]. Ayadi et al. presented a model architecture and tested the model on three distinct sources of brain tumor images, each having a different classification task. The model architecture is made of convolution layer, batch normalization layer pooling layer and fully connected layer. The researchers trained a model on each set of images. In overall, models achieve a fairly well performance in all the task, achieves in an accuracy above 90% in all of the tasks [7].

However, while CNN could be reliable in certain data sets as shown in previous studies, images given to the model might not have the same distribution as the training dataset in practice. A cross-data-set prediction could have a much lower accuracy due to the differences in characteristics such as camera angles, image resolution, brightness or even invisible noises. Furthermore, there might not be sufficient data for training another model. Thus, it is crucial to adapt the original model to new data with diverse distributions, increasing its generalization ability.

In this study, two datasets will be considered. Each dataset has been divided into train set and test set by the publisher. Both comprise MRI scans from patients diagnosed with four types of brain tumors, including glioma, meningioma, pituitary, and samples with no tumor. Each class represents a distinct pathological entity with unique imaging characteristics, posing challenges for accurate classification. The model scratch from MobileNetV2 pre-trained with ImageNet. The model intends to achieve a higher accuracy on the dataset with a different domain while keeping an around or above accuracy on the same domain.

2 Method

2.1 Dataset Description

Both datasets used in this study come from Kaggle. The first dataset is published by a user named "Shreya Gupta" and her collaborator "VishalAg11" [8]. The dataset comprises 1311 images in 4 categories: glioma, meningioma, no tumor and pituitary, each has 300, 306, 405, 300 images. This dataset will be named "Set1" below. The other dataset is published by a user called "Sartaj" [9]. The dataset is also divided into 4 classes, same as above. Also, this dataset has already been divided into the training set and testing set. However, those sets seem to have a different domain, causing models trained by the training set to have a low performance on the testing set. And there's only

394 images in the testing set, far from sufficient for training a model on it. This dataset will be named "Set2" below.

In this study, Set1 will be divided into training set, validation set, and testing set in the scale of 0.7, 0.1, 0.2. The original training set from Set2 will not be used. Instead, the testing set from Set2 will be divided into training and testing set by different scale: 0%, 10%, 20%,30%,40%. The scale indicates the proportion of training set among the whole set. Then, the training set from Set1 will incorporate with training set from Set2 and formed a new set that will be used for training the model. The validation set is the one from Set1 in order to keep track of model's performance on its own domain. Both testing sets will be used for testing. The goal is to achieve a higher accuracy on Set2's testing set while not significantly drop the accuracy on Set1's testing set.

ImageDataGenerator from Tensorflow was used for data preprocessing. Images were normalized by dividing 255. Moreover, this study enables ImageDataGenerator to augment images by horizontal flip or rotation in a range of 40 degrees. Since the dataset is not rich, these augmentations are crucial for the model to avoid over-fitting.

2.2 Proposed Approach

MobileNet [10], proposed by Google, is exceptionally light weight compared to most of CNN. Through usage of Depthwise Separable Convolution, a technique that significantly lowers the number of operations within model, MobileNet achieves high efficiency while only losing minor amount of accuracy. In this study, the model was built on MobileNetV2 in Tensorflow library. This network is a pre-trained model that has trained on ImageNet, a huge dataset that provides labeled data for machine learning.

2.3 Implementation Details

The model in this study was composed of the backbone of MobileNetV2, an average pooling layer, a flatten layer, a batch normalization layer, a dense layer with 256 nodes and an output layer. The network takes in images in size 224 × 224. After gathering features with mobile net, average pooling is applied or reducing width and height of the feature map. Then the flatten layer will resize the output into one dimension and send it to the fully connected layer after batch normalization layer normalizes the data. Dense layers then will be responsible for classifying the features into classes.

In the study, the model was compiled with Adam optimizer [11] and categorical cross-entropy as its loss. Adam is a widely used optimizer for deep learning. It has an adaptive learning rate and can possibly avoid local minima with momentum. Categorical cross-entropy computes probabilities of wrong outcomes and sums them up. The epochs were set to 50 with an early stopping callback of 10 patience. Input images were batched with a size of 16 to ensure consistent processing during training. Five networks were trained based on this model, each trained with different training data and tested by corresponding test set as stated before. However, the hyper-parameter and model architecture were the same, and they were all tested by testing set of Set1.

3 Results and Discussion

This session shows the five networks' predictions on their testing sets. Furthermore, the number of epochs each network runs before triggering early stopping callback might reveal how the alien data affect the training process. In Table 1 presented below, the testing set from Set1 is labeled as TestingSet1, and the corresponding testing set for each model after splitting is labeled as TestingSet2.

Table 1. Network Performance on Testing Sets

Proportion of Training Set split	epochs	Accuracy on TestingSet1	Accuracy on TestingSet2
Original	32	0.8787	0.7081
10%	26	0.8598	0.7303
20%	25	0.8295	0.7793
30%	25	0.8598	0.8086
40%	25	0.8598	0.8059

Among all these networks, the one that trained on original training set keeps the highest accuracy on its TestingSet1. The one that trained with training set that corporate 30% split test set achieve highest accuracy on its TestingSet2 with a 0.0189 accuracy reduction on TestingSet1.

The result demonstrates that blending external data into the training set can enhance the model's performance on the external test data. Meanwhile it also results in a sacrifice of performance on its own set. As shown in Table 1, the 20% model gets a large reduction in accuracy on TestingSet1. It could be due to the variance the newly joined data brings, prohibiting the model to learn about the pattern of original train data while unable to learn a pattern that fits in both dataset since because there's not too much information about the new data yet. However, as more data joined in, the 30% model achieves higher accuracy on both testing data than the 20% one, suggesting that incorporating more data could help the model to learn a pattern that catches the common points between two datasets. Furthermore, more data does not always mean better accuracy. Increasing split proportion to 40% did not improve the performance of model, it dropped slightly instead. It indicates that instead of keeping increasing the proportion of external data, it's more helpful to figure out the right amount. Thus, by incorporating a proper amount of external data, the model's generalization ability has improved with a tradeoff of a minor amount of accuracy.

One concern about this experiment is that models were tested on different testing sets. Due to the splitting, there's still a great variance between TestingSet2. The choice that splitting training data from the testing set of Set2 instead of using its own training set comes from a reason. There's a model that trained with the training set of Set2. it only achieves 0.7360 accuracy on its testing set, while its accuracy on TestingSet1 achieves 0.9204. It indicates that there might be a different distribution between Set2's training set and testing set. Thus, using the training set of Set2 might not improve models' performance on the test set.

4 Conclusion

This study suggests an approach to improve the generalization ability of CNN model for brain tumor classification. In the study, external data is split to different proportions and incorporated into the original training data. The model trained by 30% split data achieves the best balance among 0%-40%, enhance accuracy on external test set by 0.1005 while reduce accuracy on original test set by 0.0189. Indicating that by incorporating a proper amount of external training data, the model's accuracy can greatly improve external test data in the cost of an acceptable reduction on the accuracy of original test data. This approach might also work for other fields; however, more study needs to be done in order to solidify its effectiveness on other datasets.

References

1. Samee, N.A.; El-Kenawy, E.S.M.; Atteia, G.; Jamjoom, M.M.; Ibrahim, A.; Abdelhamid, A.A.; El-Attar, N.E.; Gaber, T.; Slowik, A.; Shams, M.Y.: Metaheuristic Optimization Through Deep Learning Classification of COVID-19 in Chest X-Ray Images. Comput. Mater. Contin. 73, 4193–4210 (2022).
2. Lakhani, P., Sundaram, B.: Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. Radiology 284(2), 574-582 (2017).
3. Gulshan, V., et al.: Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. JAMA 316(22), 2402-2410 (2016).
4. Qiu, Y., Wang, J., Jin, Z., Chen, H., Zhang, M., Guo, L.: Pose-guided matching based on deep learning for assessing quality of action on rehabilitation training. Biomedical Signal Processing and Control 72, 103323 (2022).
5. Cancer.Net: Brain Tumor - Statistics. Available at: https://www.cancer.net/cancer-types/brain-tumor/statistics (Accessed on 31 May 2023).
6. ZainEldin, H., et al.: Brain tumor detection and classification using deep learning and sine-cosine fitness grey wolf optimization. Bioengineering 10(1), 18 (2022).
7. Ayadi, W., Elhamzi, W., Charfi, I., Atri, M.: Deep CNN for brain tumor classification. Neural Processing Letters 53, 671-700 (2021).
8. Gupta, S.: Brain MRI Scans for Brain Tumor Classification. Kaggle (2023). Available at: https://www.kaggle.com/datasets/shreyag1103/brain-mri-scans-for-brain-tumor-classification (Accessed on 23 Sept. 2023).
9. Sartaj: Brain Tumor Classification (MRI). Kaggle (2020). Available at: https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri (Accessed on 24 May 2020).
10. Howard, A.G., et al.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861 (2017).
11. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980 (2014).