Behind the Pixels: An Empirical Study of Novel Machine Learning Preprocessing Techniques for Image Classification

Tejas R. Annapareddy, Jacob E. Lee, Minghui Zhang

I. Abstract

Image classification plays a crucial role in medical diagnostics, where convolutional neural networks (CNNs) are commonly used to detect various cancers accurately. However, the quality and characteristics of input images significantly impact CNN performance. This study explores the effects of novel preprocessing techniques, particularly edge detection, on CNN reliability and accuracy. Using a dataset from Kaggle, which includes 10986 images of three brain cancer subtypes, we trained a CNN model via PyTorch on Google Colab. Preliminary results indicate that edge detection preprocessing marginally improves classification accuracy compared to raw images, suggesting that preprocessing techniques can enhance CNN performance. Future research will refine these techniques and evaluate their impact on different CNN architectures to improve medical image classification further.

II. Introduction

CONVOLUTIONAL Neural Networks (CNNs) have revolutionized image classification tasks. It has many applications, such as automatically detecting street signs to reduce the number of tasks the driver needs to do while driving or detecting rust on steel that can compromise structural integrity [1, 2]. However, CNN performance can be significantly impacted by the quality and characteristics of input images (i.e. training data). In this paper, we investigate the effects of various image preprocessing techniques on the accuracy of CNN models for image classification.

III. Literature Review

A. Background

There has been much effort put into constructing CNN models that aim to aid in the medical process of detecting various cancers. It has been shown that, among 12 breast cancer detecting models, the accuracies of those models range from 85.5% to 97.8% [3]. Other common cancers, such as brain cancer, likely have models with similar ranges of accuracies. With the hope of employing these models on a large scale to aid doctors, improving the accuracies of these cancer detecting models would allow them to be used with higher confidence and lower the possibility of misdiagnoses.

B. Edge Detection

The first image preprocessing technique examined in this study is edge detection. The edge detection technique is a computer algorithm that targets boundary information of objects in images by analyzing pixel mutations of images [4]. That is, the algorithm detects discontinuities of brightness in images and reflects the information gathered in a new image, as shown in Figure 1.

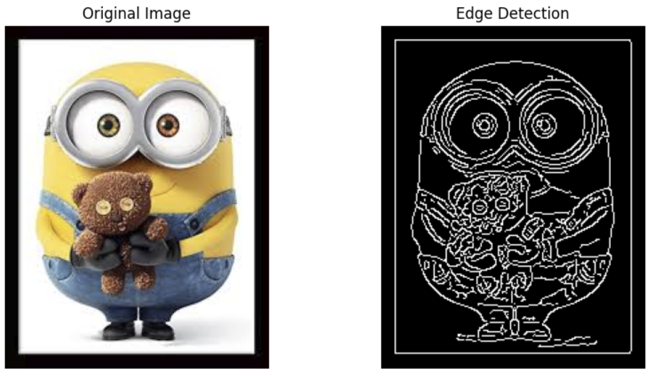

Fig. 1: An image of a Minion before and after edge detection image processing

We have chosen to use the Canny edge detection algorithm due to its resistance to noise and random variations in color or brightness and is capable of detecting weak edges [5]. Since medical images are likely to contain very weak edges of body tissues and potentially noise, the Canny edge detection algorithm is appropriate for this task. We hypothesize that applying image-processing techniques such as edge detection will improve the accuracy of a CNN model to classify various types of brain cancer scans. In accordance to our hypothesis, the independent variable of this experiment is the image-processing techniques we apply during the pre-processing stage of our CNN models, whereas the dependent variable of this experiment is the classification accuracy of these models.

IV. Methods and Materials

Kaggle is a data science platform that houses various data designated for the public to train machine learning models. We downloaded 10986 images of brain cancer from the Multi Cancer Dataset consisting of three subtypes: glioma (glio.), meningioma (mening.), and pituitary tumor (PitNET). We stored these images in Google Drive. We utilized Google Colab for its integration with Google Drive and its cloud computing features, allowing us to run our model on enterprise level equipment in the cloud in addition to an access to a free GPU instead of locally on our machines. In addition to regular brain cancer images being stored in Google Drive, after processing regular images using the edge detection described above, these images were also stored in Google Drive. Figure 2 shows a sample comparison between a regular brain cancer image and the image after edge detection processing.

After obtaining cancer images, we began processing these images and coding the CNN models. We utilized the PyTorch library to develop our models. We allocated 8788 images for training data and 2198 images for testing data and converted them to a tensor-readable format (Tensor objects). Having these images being converted to tensors allowed for these images to be passed to the models for training and testing. Tensors in the Google Colab environment were collectively using Pandas, a Python library for structured data analysis and processing, DataFrame objects. We ended up with four Dataframe objects filled with tensors: training tensor dataset for regular models, testing tensor dataset for regular models, training tensor dataset for edge detection models, and testing tensor dataset for edge detection models. These DataFrame objects were then saved as pt (PyTorch Tensors) files in our Google Drive.

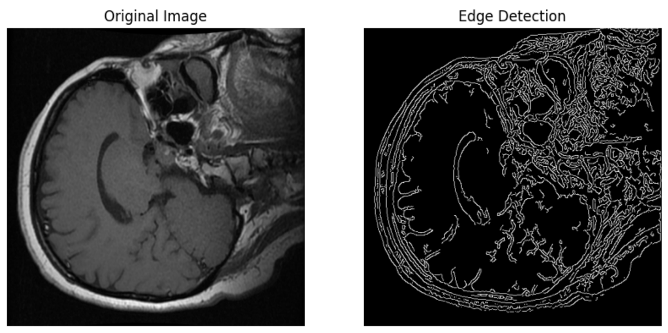

Fig. 2: A sample of medical image of brain tumor used in Google Colab machine learning training

After finishing the code for the CNN models (see Listing 1 for the layers construction of the model), the data was loaded into Colab once again, beginning the training and testing phase of this project.

In the training stage of CNN models, there various hyperparameters that we modified for both normal and edge models that affected how these models learned from the given training data. The hyperparameters that we varied were the learning rate (a value that controls how the model update its weights), the number of epochs (how many times does the model go through the dataset), and the batch size (how many images does it take for the model to update its weights). Modifying these hyperparameters contributes to the structure of the data gathered. Five trials were performed for each set of these three hyperparameters (i.e., a new model was initialized for each trial of one of the two types [normal or edge]), and the data collected from the model during each trial included the overall accuracy and the F1 scores for all three types of brain cancer. After collecting these data, a t-test was performed for each configuration of the hyperparameters in Excel to examine the presence of statistically significant results. Since this project is fully computational, no significant safety risks were present in this project.

V. Results

As mentioned earlier, the data collected from each configuration of hyperparameters included the overall accuracy (OA) and the F1 scores for all three types of brain cancer. Table I provides an overview of the values of the hyperparameters, and Table II summarizes the average of the data collected for each configuration. A total of ten configurations was tested.

As mentioned earlier in the Methods and Materials section, t-tests were performed on the collected data in examining statistically significant differences. Table III presents the results of t-tests in a concise way. For each configuration, a t-test was performed on the OA, F1 scores of glioma, meningioma, and

```
class CNN(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(1, 4, 5, 2)
        self.conv2 = nn.Conv2d(4, 16, 5, 2)
        self.fc1 = nn.Linear(576, 550)
        self.fc2 = nn.Linear(550, 540)
        self.fc3 = nn.Linear(540, 470)
        self.fc4 = nn.Linear(470, 300)
        self.fc5 = nn.Linear(300, 220)
        self.fc6 = nn.Linear(220, 130)
        self.fc7 = nn.Linear(130, 70)
        self.fc8 = nn.Linear(70, 40)
        self.fc9 = nn.Linear(40, 3)

    def forward(self, x):
        x = F.relu(self.conv1(x))
        x = F.max_pool2d(x, 3, 3)
        x = F.relu(self.conv2(x))
        x = F.max_pool2d(x, 3, 3)
        x = x.flatten(start_dim=1)
        x = F.relu(self.fc1(x))
        x = self.fc2(x)
        x = self.fc3(x)
        x = self.fc4(x)
        x = self.fc5(x)
        x = self.fc6(x)
        x = self.fc7(x)
        x = self.fc8(x)
        x = self.fc9(x)
        return x
```

Listing 1: Layers construction for our CNN models in Google Colab.

Config	Learning Rate	Number of Epochs	Batch Size
1	0.001	3	64
2	0.001	5	64
3	0.01	5	64
4	0.001	10	64
5	0.1	10	64
6	0.01	10	96
7	0.01	15	96
8	0.01	15	32
9	0.001	15	32
10	0.001	15	16

TABLE I: An Overview of the Configurations Used

			Mean F1 Score of 5 Trials		
Config.	Type	OA	Glio.	Mening.	PitNET
1	Normal	0.4329	0	0.142	0.53
1	Edge	0.4155	0	0.28	0.354
2	Normal	0.4332	0.004	0.41	0.29
2	Edge	0.4518	0	0.254	0.37
3	Normal	0.4657	0	0.512	0.122
3	Edge	0.4718	0	0.64	0
4	Normal	0.4196	0	0.188	0.434
4	Edge	0.4115	0.006	0.324	0.38
5	Normal	0.4535	0	0.256	0.366
5	Edge	0.4596	0	0.384	0.244
6	Normal	0.3765	0	0.346	0.304
6	Edge	0.4535	0	0.256	0.366
7	Normal	0.4492	0	0.362	0.304
7	Edge	0.4412	0	0.242	0.386
8	Normal	0.4474	0	0.128	0.488
8	Edge	0.4596	0	0.384	0.244
9	Normal	0.4149	0.016	0.34	0.39
9	Edge	0.4338	0	0.34	0.3
10	Normal	0.4657	0	0.512	0.122
10	Edge	0.4596	0	0.384	0.244

TABLE II: Averaged data of five trials of collected data for each configuration set of hyperparameters.

pituitary tumor from the five trials, examining the normal and edge models' differences.

It can be first observed from Table II, in terms of OA, models trained using edge detection processed images slightly outperformed normal models in Configurations 2, 3, 5, 6, 8, and 9. It is also noted that both types of models struggle to correctly identify and classify glioma brain cancer images as indicated by the low F1 scores. Continuing to the models' ability to correctly recognize images

		F1 Score of 5 Trials		
Config.	OA	Glio.	Mening.	PitNET
1	0.56		0.53	0.38
2	0.44	0.37	0.58	0.78
3	0.37		0.37	0.37
4	0.87	0.37	0.45	0.75
5	0.62		0.62	0.62
6	0.04		0.67	0.75
7	0.65		0.63	0.74
8	0.37		0.37	0.37
9	0.51	0.31	1	0.47
10	0.37		0.37	0.37

TABLE III: A summary of t-tests performed for each configuration set of hyper-parameters. Empty entries are the result of having all zero values in the calculated arrays.

		F1 Score of All Configs.		
Type	OA	Glio.	Mening.	PitNET
Normal	0.44	0.002	0.3196	0.335
Edge	0.45	0.0006	0.3488	0.2888

TABLE IV: Average of Collected Data of All 10 Configurations

of meningioma, edge detection models performed better than regular models in Configurations 1, 3, 4, 5, and 8. Additionally, with regards to correctly identify images of pituitary tumor, edge detection models only outperformed regular models in Configurations 2, 6, 7, and 10 (less than half of the total configurations).

Both types of models' poor performance on recognizing glioma brain cancer images could be attributed to the little amount of images of this type of brain cancer. Out of the 10986 total images, only 954 images were glioma. Not to mention that these 954 images were also split into training and testing data randomly, which further drives down the number of images available for the both types of models to develop robust parameters to correctly classify glioma images. The total images for meningioma and pituitary tumor were 5022 and 5010, respectively. This comparatively larger dataset likely contributed to the models performing better on these two types of cancer images than glioma. Moreover, this similar amount of total images of meningioma and pituitary tumor could play a role in explaining how edge detection models outperformed regular models in around half of the total configurations for these two types of brain cancer images.

As one can see from Table IV, edge detection models outperformed normal models in OA and F1 score of meningioma overall while normal models did better in recognizing giloma and pituitary tumor images. This suggests the potential in the future where researchers combined normal and edge models together, assigning the tasks of identifying giloma and pituitary tumor images to the normal model and the task of recognizing meningioma to the edge model. Doing so could potentially further improve the OA and performance.

Examining Table III, out of the 40 t-tests performed, only one of which is statistically significant p ¡ 0.05: the overall accuracy of Config. 6. This result indicates that the better performance of edge models with regards to OA using the hyperparameters of Config. 6 has less than 5% chance of occurring at random.

VI. Conclusion

Our research relates to the optimization of CNN-based image classification models. By investigating the effects of preprocessing techniques, such as edge detection, we aim to enhance the accuracy and reliability of these models. The use of the Canny edge detection algorithm, known for its resistance to noise and sensitivity to weak edges, holds promise for improving medical image analysis.

Our future work will focus on refining the preprocessing techniques and evaluating their impact on different CNN models. We will also explore the scalability of our proposed methods to other medical imaging applications. Additionally, in this research, we kept our layers construction constant, which limited the possibility of the models perhaps performing better with more complex layers construction (i.e., more layers) or simpler construction (i.e., less layers). Therefore, in the future, researchers could also have multiple configurations of layers construction in combination with different preprocessing techniques to examine whether it could be a direction to improve the performance of these models. Researchers should also consider to expand this examination to other types of cancer, including breast and lung cancers, to see whether similar results would be obtained.

References

Jmour, N., Zayen, S., & Abdelkrim, A. (2018). Convolutional neural networks for image classification. *2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*, 397–402. https://doi.org/10.1109/ASET.2018.8379889

Lyu, H. (2023). Research on corrosion recognition method of steel based on convolutional neural network. *2023 IEEE 6th International Conference on Information Systems and Computer Aided Education (ICISCAE)*, 507–511. https://doi.org/10.1109/ICISCAE59047.2023.10393077

Patel, K., Huang, S., Rashid, A., Varghese, B., & Gholamrezanezhad, A. (2023). A narrative review of the use of artificial intelligence in breast, lung, and prostate cancer. *Life (Basel)*, *13*(10).

Sun, R., Lei, T., Chen, Q., Wang, Z., Du, X., Zhao, W., & Nandi, A. K. (2022). Survey of image edge detection. *Frontiers in Signal Processing*, *2*. https://doi.org/10.3389/frsip.2022.826967

Tian, B., & Wei, W. (2022). Research overview on edge detection algorithms based on deep learning and image fusion. *Security and Communication Networks*, *2022*, 1155814. https://doi.org/10.1155/2022/1155814