**MRI Brain Classification Interim Report**

Arnab Sagar and Nicholas Baker

1088717, 1100494

arnab, nbaker05

University of Guelph

CIS\*4780: Computational intelligence

Dr. Rozita Dara

November 1st, 2023

# **Objective Overview**

Our project is geared towards a thorough comparison of the performance of existing algorithms across diverse applications, primarily focusing on the benchmarking of different algorithms for object detection in images—specifically utilizing MRI images of brains with and without tumors.

The project encompasses several key objectives. Initially, we aim to locate a suitable dataset containing brain images, encompassing both instances with tumors and those without. Following dataset acquisition, the data will undergo a meticulous preprocessing phase to ensure its cleanliness and suitability for the models, with a focus on mitigating potential noise within the dataset. Subsequently, we will shift our attention to the selection and understanding of various models employed in object detection. This involves gaining insights into the operational mechanisms of each model, especially concerning the different dynamics of the chosen dataset. The next phase would involve the actual training of each selected model on the prepared dataset. We will be conducting validation and testing, during which crucial metrics such as processing time, training time, accuracy, F1 score, loss, and p-value will be recorded. Once we have all the results we can then perform an extensive analysis, potentially finding improvements in the models if possible. Finally, we will compare the results of each of the models.

As for the expected outcome of the comparison we have some thoughts. The anticipated outcomes include the identification of a superior model based on specified metrics, as well as an acknowledgment of the possibility that, when pushed to their limits, all models may exhibit similar performance.

# **Tools**

The tools that we are using for the project are not far off from the proposal. Firstly, we are using Google Collab for pair programming which allows us to implement in the same environment at the same time. Google Collab is the same as Jupyter Notebook just within the Google cloud. We are running all code within Python. The main libraries we are running are Pytorch, Numpy, and Matplotlib. The other tools would be the models themselves and that too is mostly done with Pytorch with an outside source for the YOLO model **[1]**.

The data we are using has a total of 253 images. 155 tumor, 98 no\_tumor **[2]**. The sizes of the images are not consistent, and neither is the framing. Hence some preprocessing is required. Furthermore, each model has different requirements that we will need to meet when preprocessing.

# **Fig 1. Sample images from the dataset**

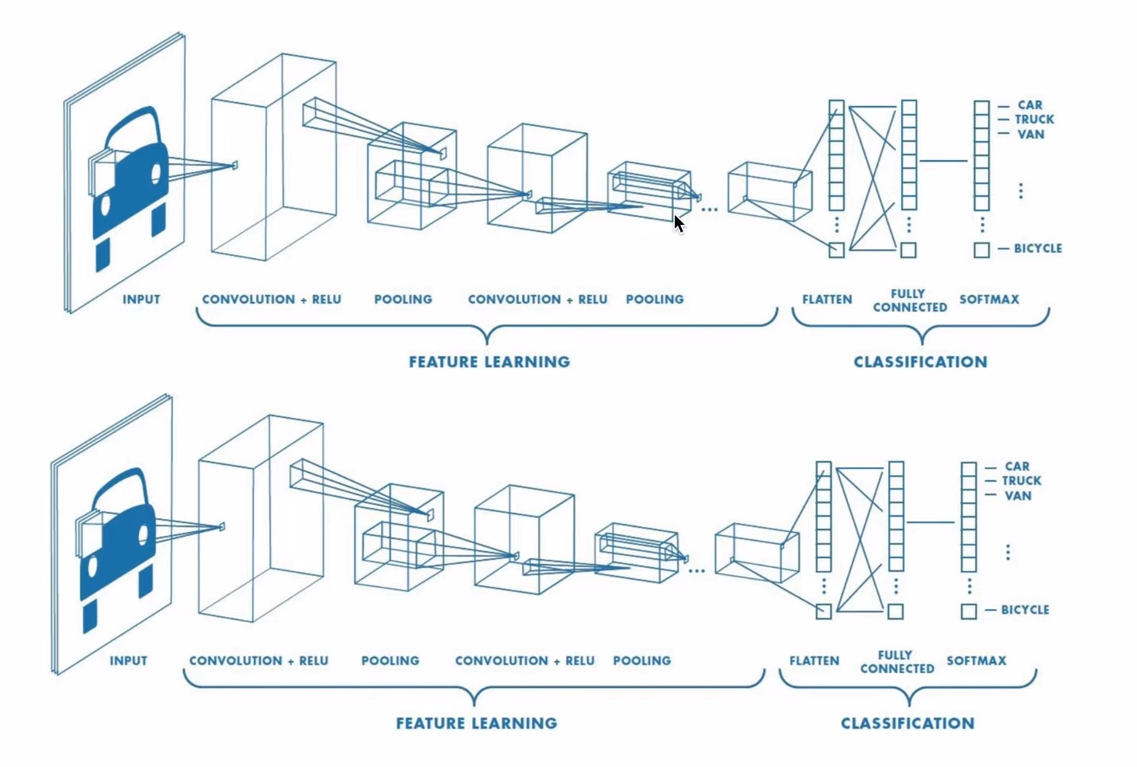


# **Approach**

Our approach to this project is the same as our list of objectives. The plan is to first find and preprocess the dataset to limit the noise the models have to deal with. That preprocessing involves contouring the images to clearly show tumours to the model. Also cropping the image as some MRI images contain text and symbols other than the brain itself. From there we plan to split the data into test and train sets, then perform transformations and then finally convert them into tensors. We also normalize the data after preprocessing.

Now that the data is ready we test and train the models and save the best models. We are closely following the approach of a report found in the proposal **[3]**. We leverage Transfer Learning with the pre-trained models to fit our use case. We only use the Convolutional Layers (w/ ReLu activation functions) and Pooling layers from the pre-trained CNN models for the feature learning part, and then we add our own classification layer (w/ softmax) at the end.

**Fig 2. Transfer Learning**



**[4]**

For the addition of the classification layers, our first choice is to do some model fine-tuning. This method trains the model all over again, but only a little bit, as we fine-tune all the weights based on the new data and for the newly added last layer (for the classification). The alternative is freezing all the layers in the beginning and only training the very last layer which we added. This is done by looping over all the model parameters. The latter approach is much faster in training but leads to less accuracy. This can however change with a higher number of epochs. We ended up using the latter method.

# **Results**

# Right now we only have one model working so far. That model is ResNet-50, and the results so far are promising. When run through a few epochs we can get a validation accuracy of 92% with a loss of 0.33. Now right now we only have one model working but plan to finish the rest of the models soon. Having an accuracy so high makes us believe that we are on the right track so far.

# **Remaining Objectives**

The list below shows the objectives that we have completed and what we plan to complete. Specifically, we have completed all the preprocessing and can test and train models. We currently have one model completed, that model being ResNet-50. We plan to complete YOLOv5, VGG16, and Inception V3. Once all the testing and optimizing are done we will then compare the models against each other.

**Objectives**

1. ~~Find a dataset of MRI images of brains~~
2. ~~Preprocess the dataset for training and testing~~
3. ~~Download and run models~~
4. Save trained models on the dataset
5. Test and compare models on accuracy, f1-score, loss, etc.
6. See if we can boost the scores of models potentially
7. Conclude comparison with results

# **Risks**

The only risk we have when it comes to our objective would be a heavy computing load. Now we don't have this issue but that could have been one. The problem is that if we had too much data our model could have taken too long to train up. But as our data set is not as large as 3000 images we are fine.

**Fig 3. Preliminary Results with ResNet50 model**



**References**

Ultralytics. "YOLOv5: YOLOv5 in PyTorch > ONNX > CoreML > TFLite." GitHub. Available: <https://github.com/ultralytics/yolov5>. Accessed: November 22, 2023. **[1]**

Navoneel. "Brain MRI Images for Brain Tumor Detection." Kaggle. Available: <https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection>. Accessed: November 22, 2023. **[2]**

Md Ishtyaq Mahmud, Muntasir Mamun, Ahmed Abdelgawad. "A Deep Analysis of Brain Tumor Detection from MR Images Using Deep Learning Networks" Sensors, vol. 16, no. 4, pp. 176, Apr. 2016. Available: <https://www.mdpi.com/1999-4893/16/4/176>. **[3]**

<https://www.youtube.com/watch?v=K0lWSB2QoIQ> **[4]**

PyTorch. "PyTorch Vision - Pre-trained models for Image Classification," PyTorch. Available: <https://pytorch.org/vision/stable/models.html#classification>. **[5]**

Ritika singh. "Dish Classification using ResNet50 Model with PyTorch," Medium. Available: <https://medium.com/@wularitz/image-classifier-for-dish-classification-using-resnet50-with-pytorch-5d11c02067b5>. **[6]**

Arnab Kumar Roy. "Brain Tumor Classification using ResNet50 with Intel oneAPI Toolkit," Medium. Available: <https://medium.com/@arnabroy02/brain-tumor-classification-using-resnet50-3f8648958799>. **[7]**

<https://learn.microsoft.com/en-us/windows/ai/windows-ml/tutorials/pytorch-train-model> **[8]**

MindSpore. "Image Classification Using ResNet-50 Network," MindSpore. Available: <https://www.mindspore.cn/tutorial/training/en/r1.1/advanced_use/cv_resnet50.html#:~:text=ResNet%20is%20recommended.,Second%2C%20it%20is%20learnable>. **[9]**

Piotr Migdał. "Transfer learning with ResNet-50 in PyTorch," Kaggle. Available: <https://www.kaggle.com/code/pmigdal/transfer-learning-with-resnet-50-in-pytorch/notebook>. **[10]**

<https://medium.com/analytics-vidhya/cnn-transfer-learning-with-vgg-16-and-resnet-50-feature-extraction-for-image-retrieval-with-keras-53320c580853> **[11]**

PyTorch. "Training a Classifier," PyTorch. Available: <https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html> **[12]**