

Brain Tumor Classification

Janet Yu

basic overview:

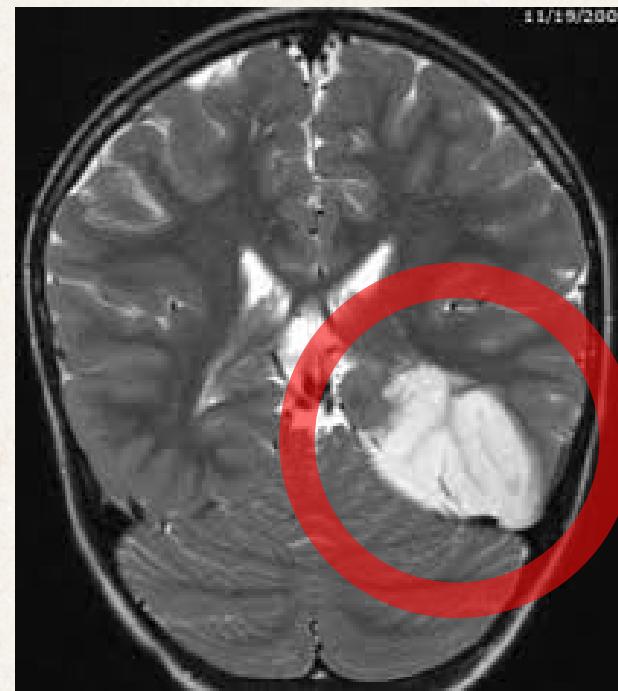
take brain MRIs and
classify as: “Yes Tumor” vs. “No Tumor”

my goals ⚡:

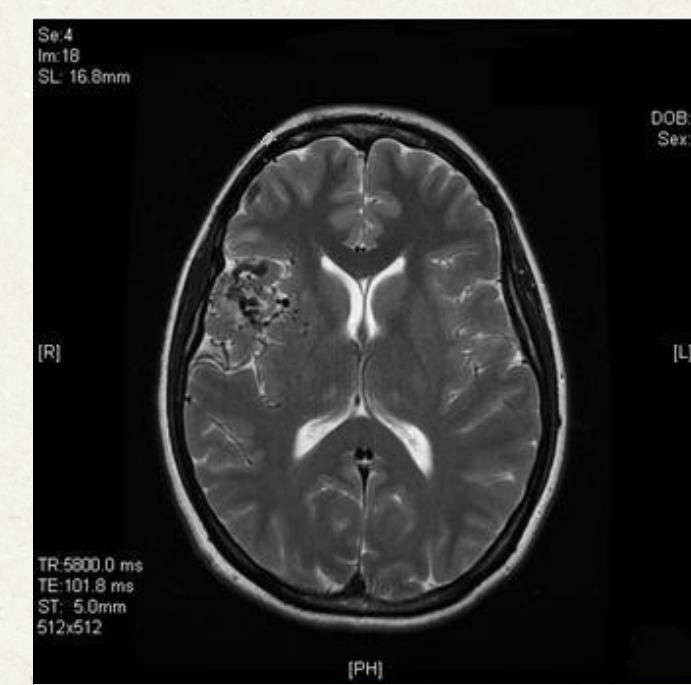
- ~ learn pytorch
- ~ understand more about CNNs and image classification
- ~ do something medical-related to ML

dataset from kaggle (tygirlie)

“yes”

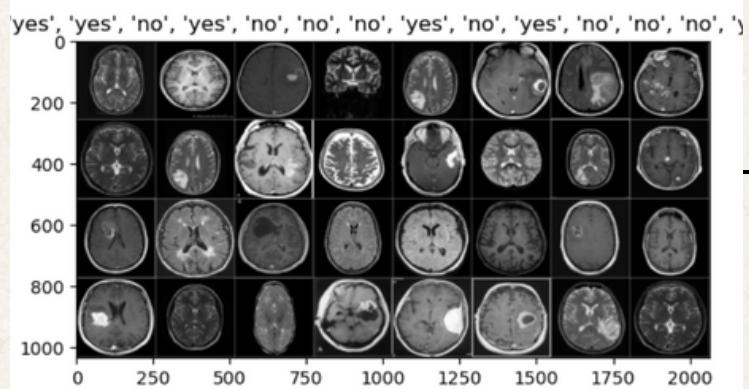


“no”



the process

i was going for ANY accuracy above 50% :)



load images
transforms,
sizing, colors

```
class SimpleCNN(nn.Module):
    def __init__(self):
        super(SimpleCNN, self).__init__()

        # layers
        self.conv1 = nn.Conv2d(in_channels=3, out_channels=32, kernel_size=3, padding=1) # For RGB images
        self.conv2 = nn.Conv2d(32, 64, kernel_size=3, padding=1)
        self.conv3 = nn.Conv2d(64, 128, kernel_size=3, padding=1)

        # pooling
        self.pool = nn.MaxPool2d(kernel_size=2, stride=2)

        # connect layers
        self.fc1 = nn.Linear(128 * 32 * 32, 512) # Make sure the input size is correct
        self.fc2 = nn.Linear(512, 2) # Assuming binary classification (tumor / no tumor)

        # dropout layer
        self.dropout = nn.Dropout(0.5)

    def forward(self, x):

        # relu -> conv -> pool
        x = self.pool(F.relu(self.conv1(x)))
        x = self.pool(F.relu(self.conv2(x)))
        x = self.pool(F.relu(self.conv3(x)))

        # flatten -> connected layer
        x = x.view(-1, 128 * 32 * 32)

        # connected layer with dropout
        x = self.dropout(F.relu(self.fc1(x)))

        # output layer
        x = self.fc2(x)
        return x
```

define my cnn
layerslayerslayers



```
In [*]: # first training loop  
  
# Example training loop  
for epoch in range(20):
```

Test Loss: 0.031720, Test Accuracy: 72.22%

train the model



output accuracy :3



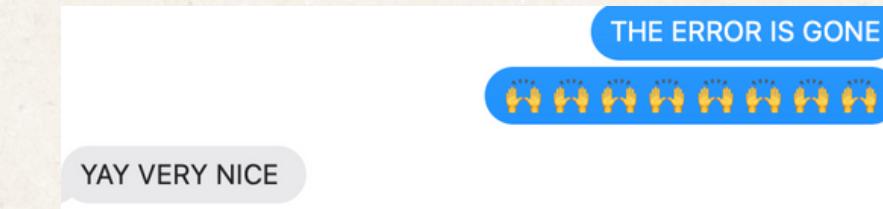
my challenges



~ streamlit.



+ future plans :)



THANK YOU HAIRAN 🙏



```
timeError: Error(s) in loading state_dict for SimpleCNN: size mismatch for fc1.weight: copying
  from checkpoint, the shape is torch.Size([512, 131072]) to current model is
  h.Size([512, 100352]).
```

eback:

```
file "/Users/janet/Desktop/2nd Year - UCLA/winter24/DSU Project/myenv/lib/
      exec(code, module.__dict__)
file "/Users/janet/Desktop/2nd Year - UCLA/winter24/DSU Project/braintumor
      model = load_model('/Users/janet/Desktop/2nd Year - UCLA/winter24/DSU
      ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
file "/Users/janet/Desktop/2nd Year - UCLA/winter24/DSU Project/myenv/lib/
      return get_or_create_cached_value()
      ^^^^^^^^^^^^^^
file "/Users/janet/Desktop/2nd Year - UCLA/winter24/DSU Project/myenv/lib/
      return_value = non_optional_func(*args, **kwargs)
      ^^^^^^^^^^
file "/Users/janet/Desktop/2nd Year - UCLA/winter24/DSU Project/braintumor
      model.load_state_dict(torch.load(model_path, map_location=torch.device
file "/Users/janet/Desktop/2nd Year - UCLA/winter24/DSU Project/myenv/lib/
      raise RuntimeError('Error(s) in loading state_dict for {}:\n\t{}'.format
```