Using CNNs to Detect Brain Tumors in MRIs

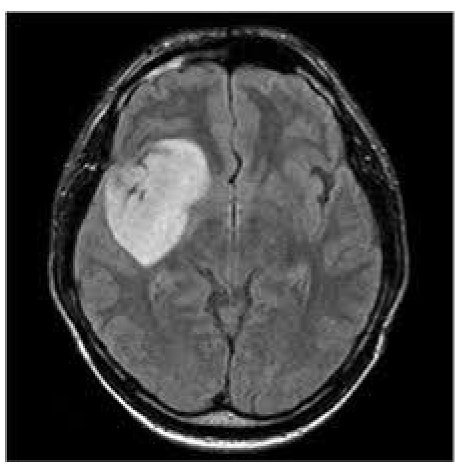
Objective:

The goal of this project is to determine how well neural networks can detect brain tumors when given an MRI. The application for this network could be an automated pre-screening of MRIs to determine whether or not there is a possibility of a tumor being present. A convolutional neural network is used, as it is adept at classifying images. Additionally, a standard dense neural network is used as a baseline to compare the CNN.

Data:

The set of images used for this project were found on Kaggle, and contain 99 images of MRIs of brains without tumors and 156 images of brains with tumors. All images were converted from RGB to grayscale and downsampled to 256 pixels by 256 pixels. The dataset was split into an 80%/20% train/validation split. The training dataset contained 85 images of no tumors and 118 images with tumors. The validation dataset included 13 images of no tumors and 37 images with tumors.

Negative



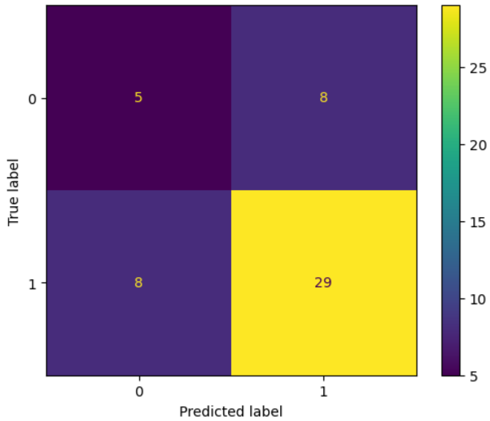
Positive

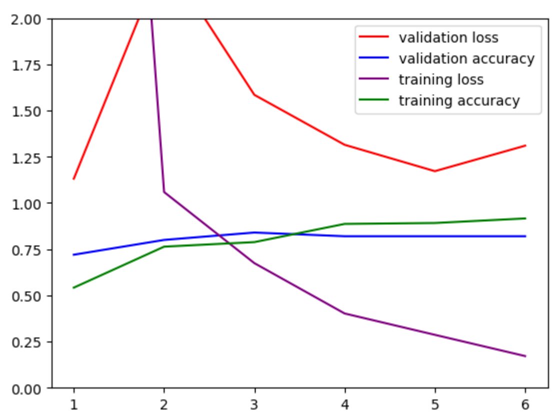
Models:

Various configurations of CNNs with different hyperparameters and structures were tested. All models used binary crossentropy for the loss function, Adam for the optimizer, accuracy for the metrics the model evaluates, an early stopping call back that monitors loss, and all layers used the ReLU activation function except for a final dense layer with 1 neuron and the sigmoid activation function activation function. This last layer is used to classify each image as containing a tumor (1) or no tumor (0).

Dense NN:

```
layers = [
   Rescaling(1./255),
   Flatten(),
   Dense(128, activation='relu'),
   Dense(1, activation='sigmoid')
]
```



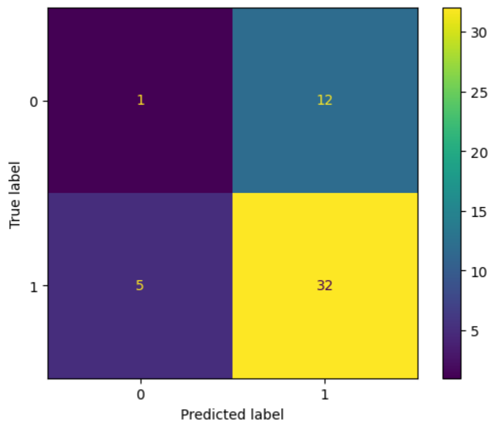


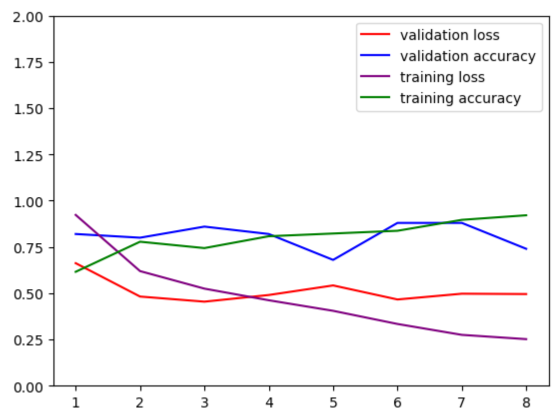
.

As a baseline, a neural network with 1 hidden layer was used. This model performed fairly well, correctly labeling 5 out of 13 images as no tumor and 29 out of 37 images as yes tumor. The benefit of using this model is that it trains and predicts quickly as it is a small network.

CNN 1:

```
layers = [
    Rescaling(1./255),
    Conv2D(filters=64, kernel_size=16, strides=4, padding='same',
        activation='relu'),
    MaxPooling2D(),
    Flatten(),
    Dense(64, activation='relu'),
    Dense(1, activation='sigmoid')
]
```

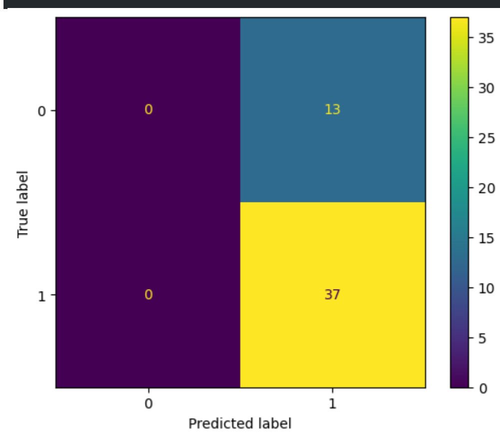


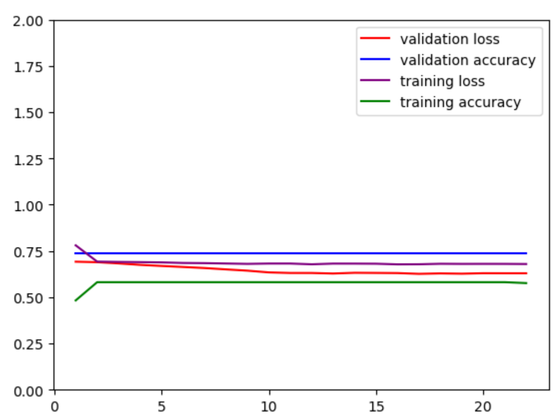


This convolutional neural network performed slightly better than the neural network at detecting true positives, however it performed poorly when classifying non-tumor scans in the validation set.

CNN 2:

```
layers = [
   Rescaling(1./255),
   Conv2D(filters=64, kernel_size=32, strides=4, padding='same',
      activation='relu'),
   MaxPooling2D(),
   Conv2D(filters=128, kernel_size=32, strides=4, padding='same',
      activation='relu'),
   MaxPooling2D(),
   Conv2D(filters=256, kernel_size=8, strides=1, padding='same',
      activation='relu'),
   MaxPooling2D(),
   Flatten(),
   Dense(256, activation='relu'),
   Dropout(0.5),
   Dense(128, activation='relu'),
   Dropout(0.3),
   Dense(32, activation='relu'),
   Dense(1, activation='sigmoid')
]
```

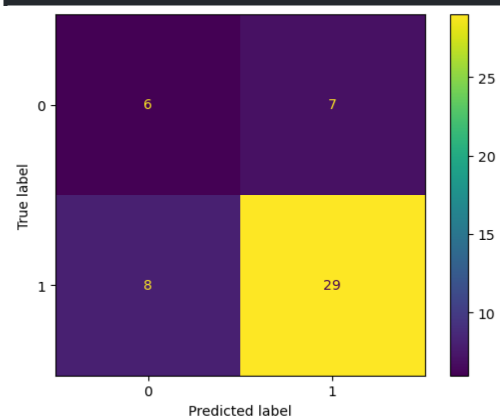


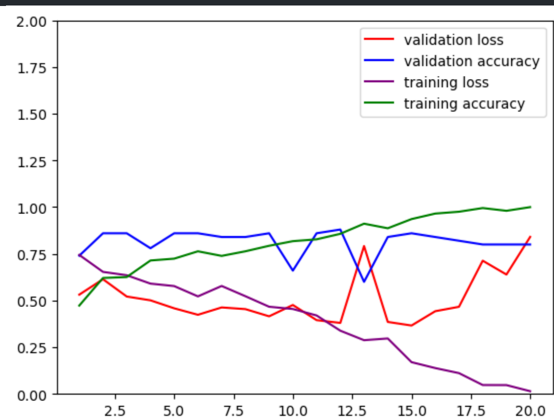


This model builds on the previous model, adding more convolutional and dense layers as well as dropout. This model predicted all images in the validation set as containing a tumor.

CNN 3:

```
layers = [
   Rescaling(1./255),
   Conv2D(filters=128, kernel_size=16, strides=4, padding='same',
      activation='relu'),
   MaxPooling2D(),
   Conv2D(filters=256, kernel_size=8, strides=2, padding='same',
      activation='relu'),
   MaxPooling2D(),
   Flatten(),
   Dense(128, activation='relu'),
   Dropout(0.5),
   Dense(32, activation='relu'),
   Dense(1, activation='sigmoid')
]
```





This model omits one dense and one convolutional layer from the previous model, and performs better at classifying positives and negatives. This model is the most balanced and optimized for detecting tumors when compared to the other models.

Conclusion:

While this model works well for detecting true positives, it is not very accurate for true negatives. Since this model is not more accurate than a trained human, this cannot be relied on for diagnoses. It may however be useful as an automatic prescreener. Future work could be done to help doctors in finding the location of a tumor in an MRI by highlighting the area the network believes a tumor is as well as improving accuracy. Interestingly, the neural network with only one fully connected hidden layer seems to perform almost as well as the most optimized convolutional model, so further testing could be done to see which is more efficient.