COMPANY NAME: MAH | June 6, 2025

AI Cancer Detection Project Using CNN

We (MAH) will be training an AI model to differentiate between healthy brains and one's with cancer through images of brain scans

Our team

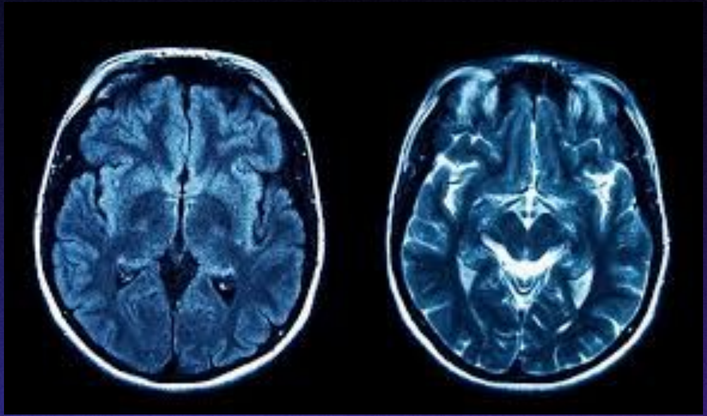
Amanda Loeung

Hamza Harb

Melodie Cornelly

What did we explore & What research did we make?

Team MAH wanted to aid medical professionals in detecting brain tumors from MRI scans using deep learning.



We explored the use of CNNs for automated detection of abnormalities in MRI scans.

About the Dataset

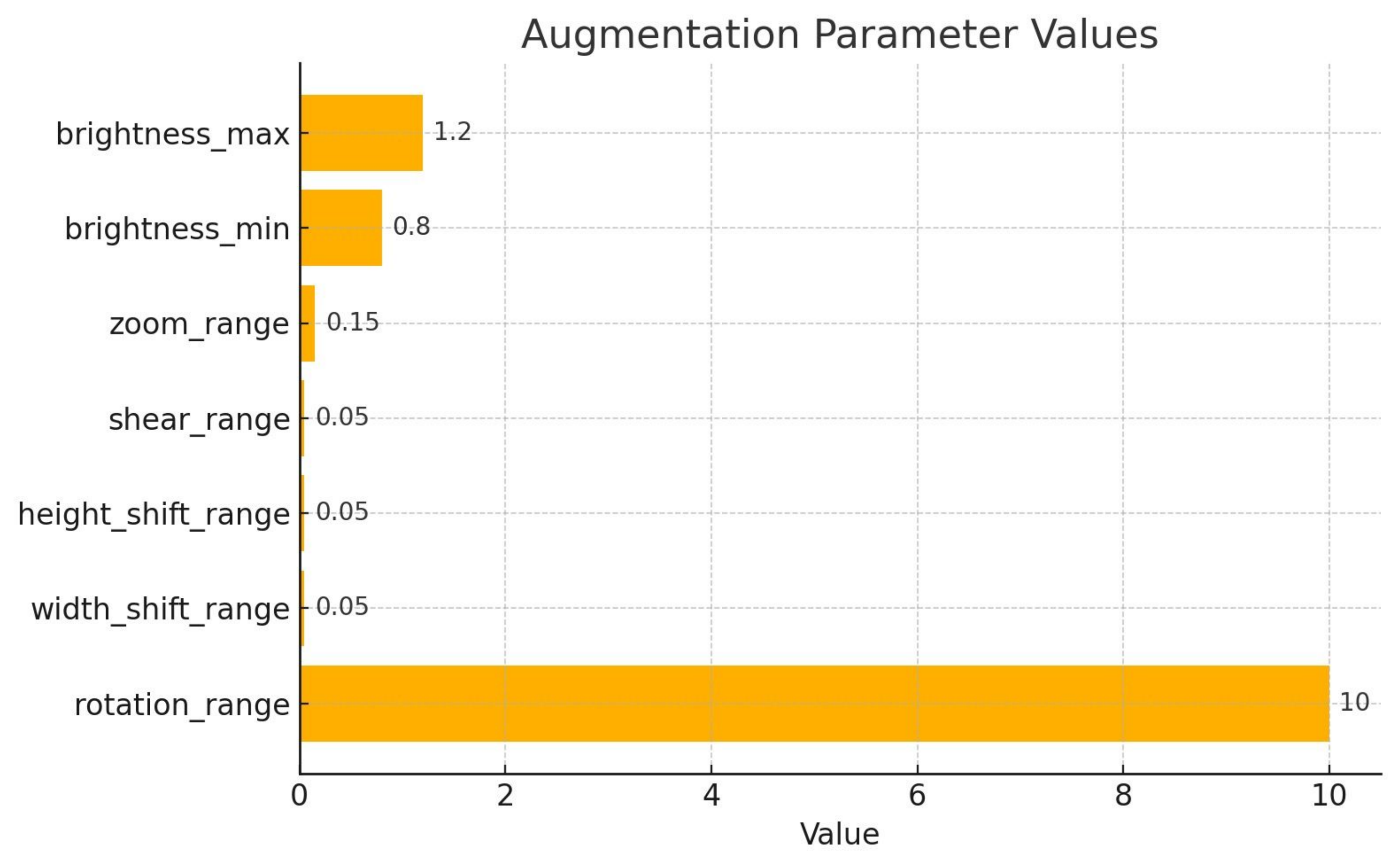
URL:
Brain Tumor Dataset: Segmentation & Classification

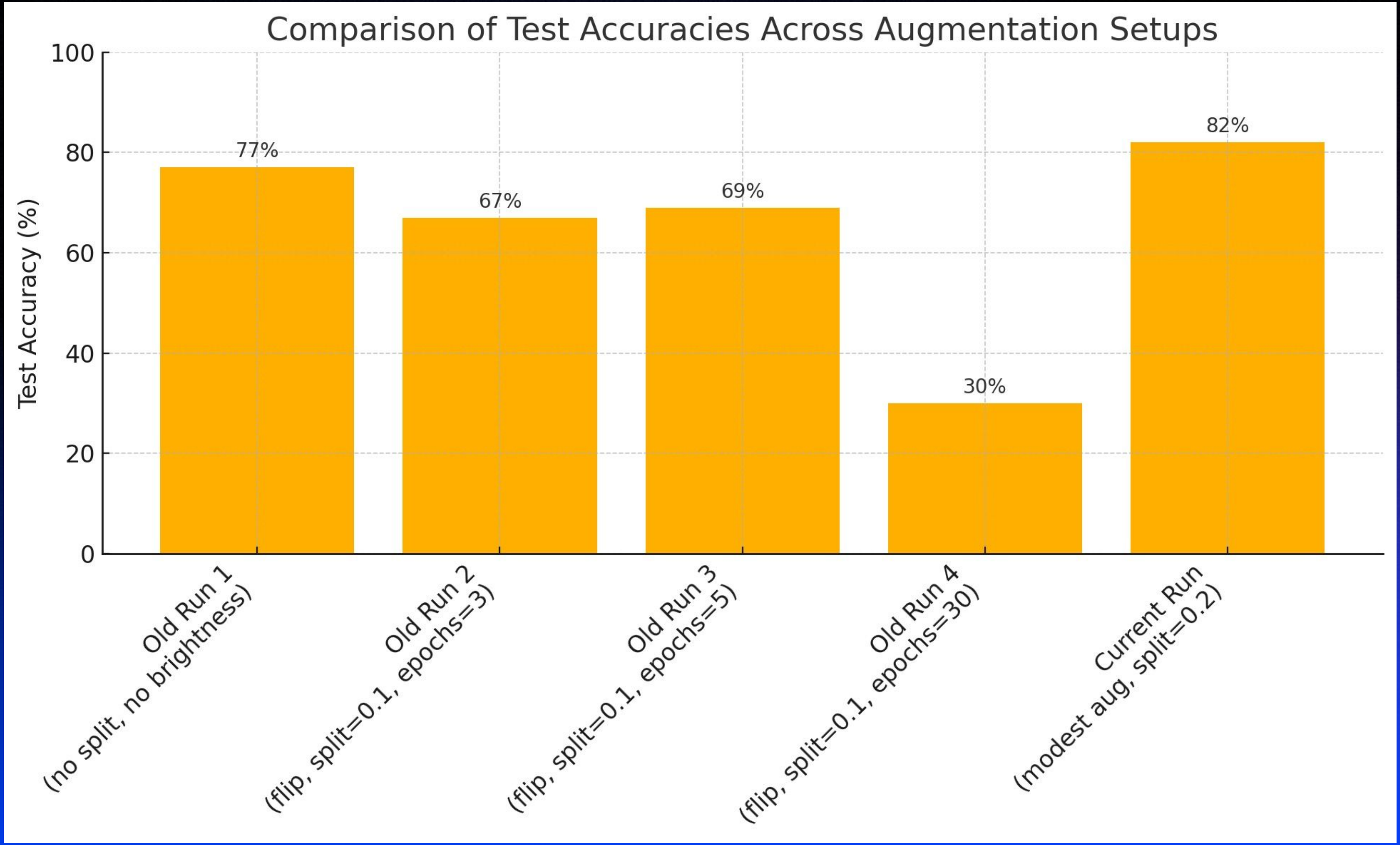
5712 images for training:
Pituitary: 1457 images
No Tumor: 1595 images
Meningioma: 1339 images
Glioma: 1321 images

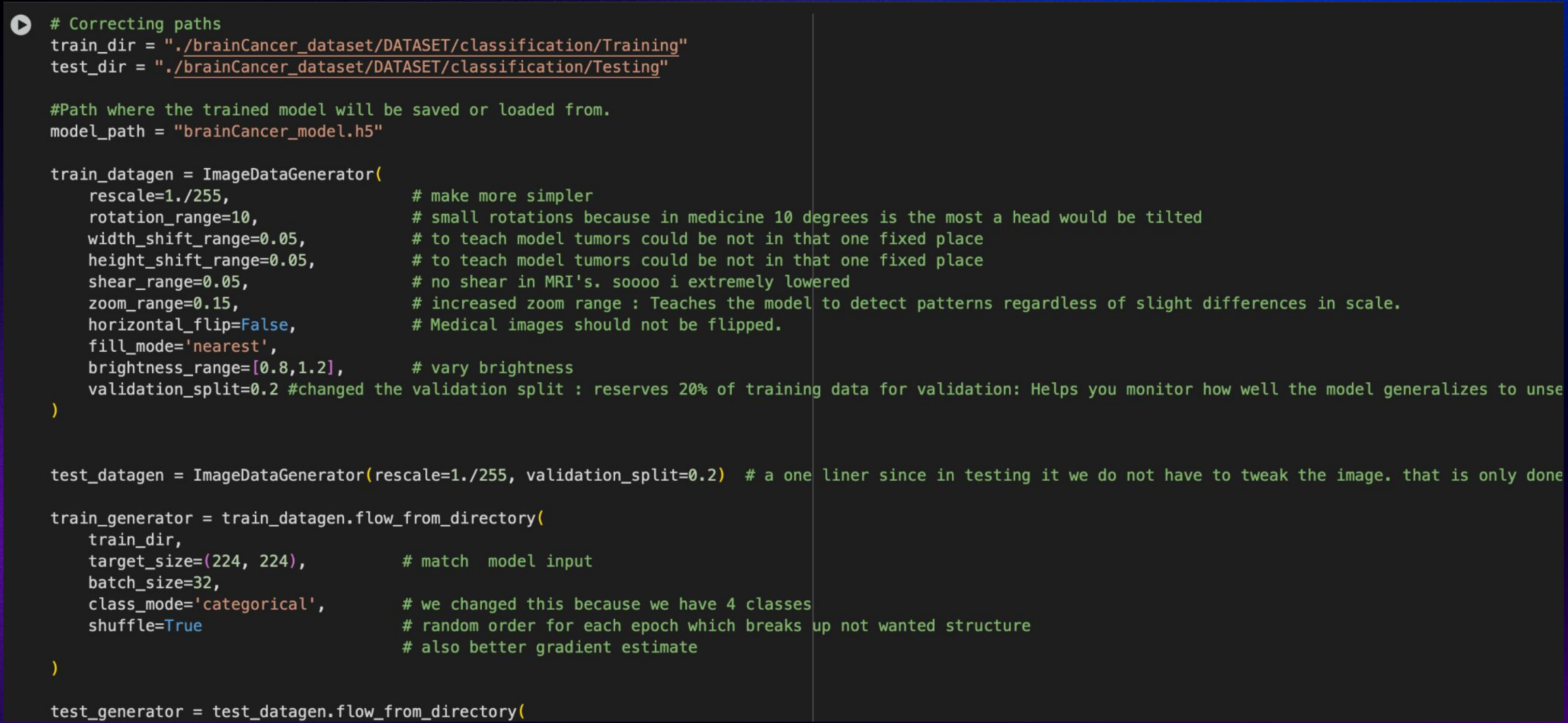
1311 images for testing:
Pituitary: 300 images
No Tumor: 405 images
Meningioma: 306 images
Glioma: 300 images

Resized images to 224x224

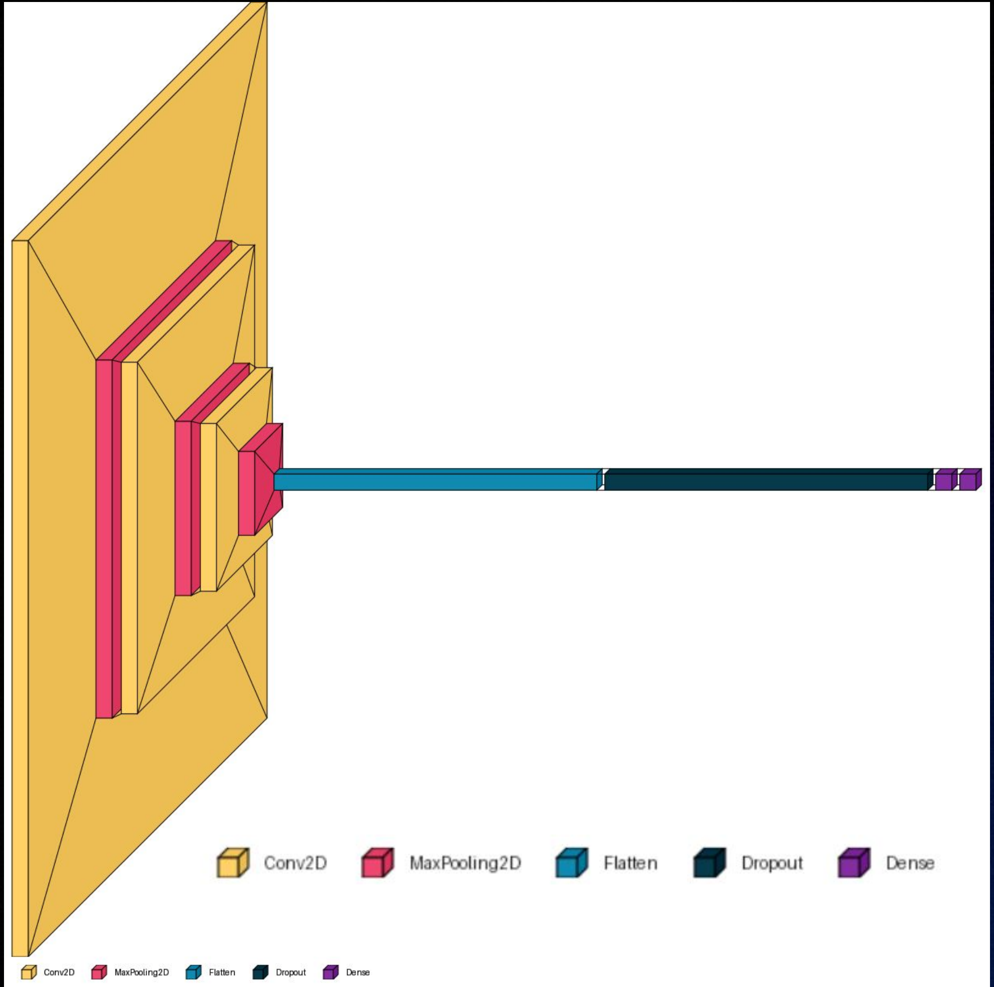
Experimentation



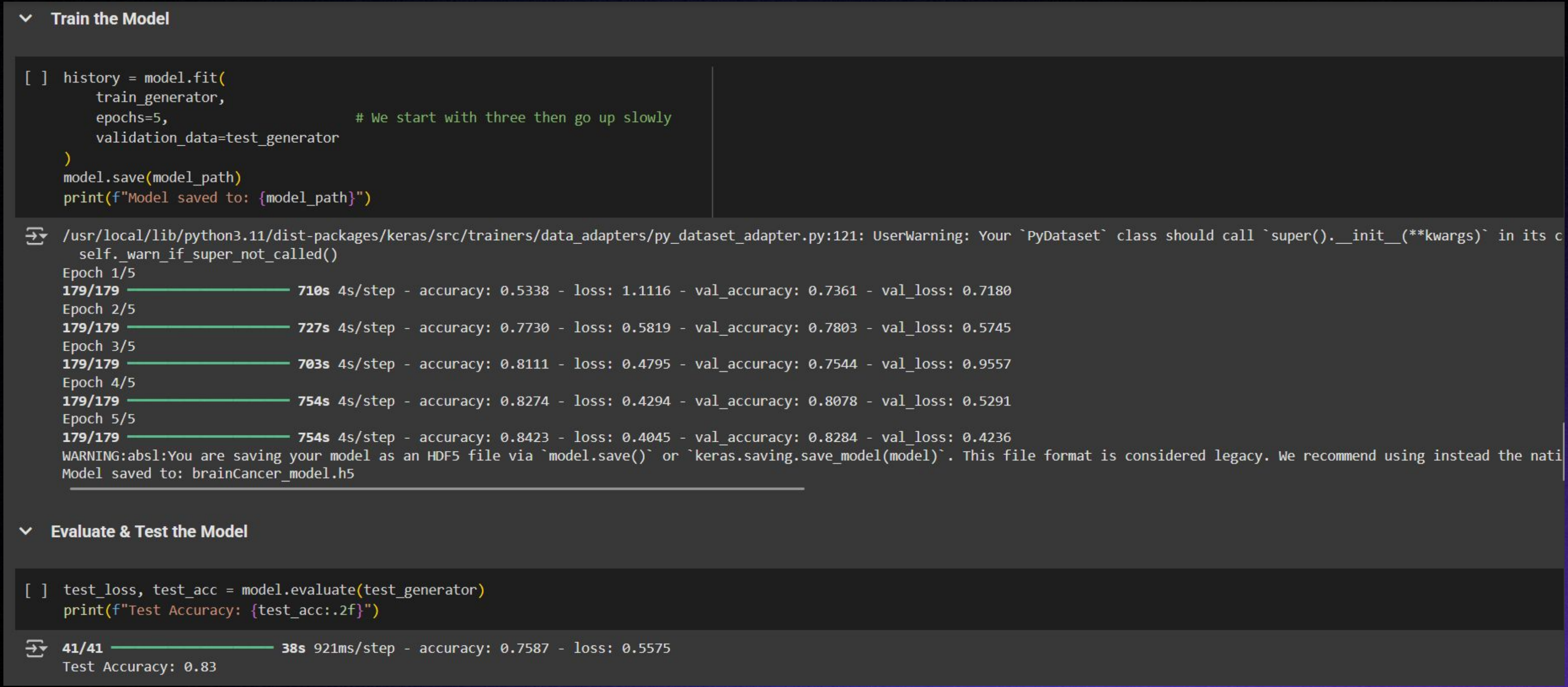

Comparison of Test Accuracies Across Augmentation Setups
Test Accuracy (%)
100
80
60
40
20
0
77%
67%
69%
30%
82%
Old Run 1
(no split, no brightness)
Old Run 2
(flip, split=0.1, epochs=3)
Old Run 3
(flip, split=0.1, epochs=5)
Old Run 4
(flip, split=0.1, epochs=30)
Current Run
(modest aug, split=0.2)



```
# Correcting paths
train_dir = "./brainCancer_dataset/DATASET/classification/Training"
test_dir = "./brainCancer_dataset/DATASET/classification/Testing"

#Path where the trained model will be saved or loaded from.
model_path = "brainCancer_model.h5"

train_datagen = ImageDataGenerator(
    rescale=1./255,                   # make more simpler
    rotation_range=10,                # small rotations because in medicine 10 degrees is the most a head would be tilted
    width_shift_range=0.05,           # to teach model tumors could be not in that one fixed place
    height_shift_range=0.05,          # to teach model tumors could be not in that one fixed place
    shear_range=0.05,                 # no shear in MRI's. soooo i extremely lowered
    zoom_range=0.15,                  # increased zoom range : Teaches the model to detect patterns regardless of slight differences in scale.
    horizontal_flip=False,            # Medical images should not be flipped.
    fill_mode='nearest',
    brightness_range=[0.8,1.2],       # vary brightness
    validation_split=0.2 #changed the validation split : reserves 20% of training data for validation: Helps you monitor how well the model generalizes to unse
)


test_datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2)  # a one liner since in testing it we do not have to tweak the image. that is only done

train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(224, 224),          # match  model input
    batch_size=32,
    class_mode='categorical',        # we changed this because we have 4 classes
    shuffle=True                     # random order for each epoch which breaks up not wanted structure
                                     # also better gradient estimate
)

test_generator = test_datagen.flow_from_directory(
```



```
#first convolutional layer
model.add(Conv2D(32,(3,3), activation = 'relu', input_shape
=(224,224,3)))

#MaxPooling to reduce spatial size
model.add(MaxPooling2D(pool_size=(2,2)))

#second convolutional layer
model.add(Conv2D(64,(3,3), activation = 'relu'))

#MaxPooling
model.add(MaxPooling2D(pool_size=(2,2)))

#third convolutional layer
model.add(Conv2D(128,(3,3),activation = 'relu'))

#MaxPooling
model.add(MaxPooling2D(pool_size = (2,2)))

#Flatten the 2D feature maps into 1D vector
model.add(Flatten())

 #Dropout to reduce overfitting
model.add(Dropout(0.5))

# Fully Connected (Dense) layer
model.add(Dense(128, activation='relu'))

# Output layer: 4 neurons with softmax for categorical classification
model.add(Dense(4, activation='softmax'))
```



Train the Model

```
history = model.fit(
    train_generator,
    epochs=5,                         # We start with three then go up slowly
    validation_data=test_generator
)
model.save(model_path)
print(f"Model saved to: {model_path}")
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset` class should call `super().__init__(**kwargs)` in its c
  self._warn_if_super_not_called()
Epoch 1/5
179/179 ━━━━━━━━━━━━━━━━━━━━ 710s 4s/step - accuracy: 0.5338 - loss: 1.1116 - val_accuracy: 0.7361 - val_loss: 0.7180
Epoch 2/5
179/179 ━━━━━━━━━━━━━━━━━━━━ 727s 4s/step - accuracy: 0.7730 - loss: 0.5819 - val_accuracy: 0.7803 - val_loss: 0.5745
Epoch 3/5
179/179 ━━━━━━━━━━━━━━━━━━━━ 703s 4s/step - accuracy: 0.8111 - loss: 0.4795 - val_accuracy: 0.7544 - val_loss: 0.9557
Epoch 4/5
179/179 ━━━━━━━━━━━━━━━━━━━━ 754s 4s/step - accuracy: 0.8274 - loss: 0.4294 - val_accuracy: 0.8078 - val_loss: 0.5291
Epoch 5/5
179/179 ━━━━━━━━━━━━━━━━━━━━ 754s 4s/step - accuracy: 0.8423 - loss: 0.4045 - val_accuracy: 0.8284 - val_loss: 0.4236
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend using instead the nati
Model saved to: brainCancer_model.h5
```

Evaluate & Test the Model

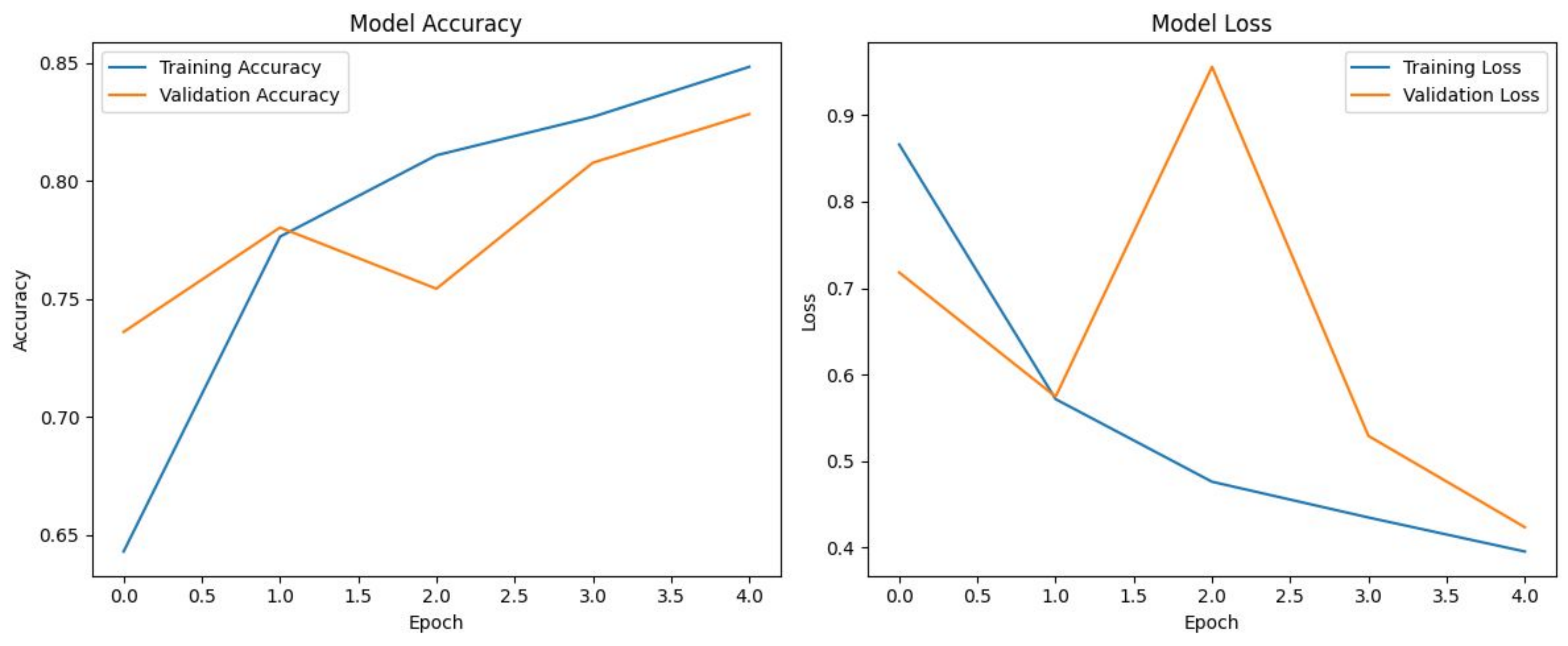
```
test_loss, test_acc = model.evaluate(test_generator)
print(f"Test Accuracy: {test_acc:.2f}")
```

```
41/41 ━━━━━━━━━━━━━━━━━━━━ 38s 921ms/step - accuracy: 0.7587 - loss: 0.5575
Test Accuracy: 0.83
```

The dataset consists of labeled MRI brain images that we categorized into 4 classes. Glioma, Meningioma, Pituitary, No Tumor.

We prepared the MRI images by Rescaling, rotation, zoom, and brightness adjustment.

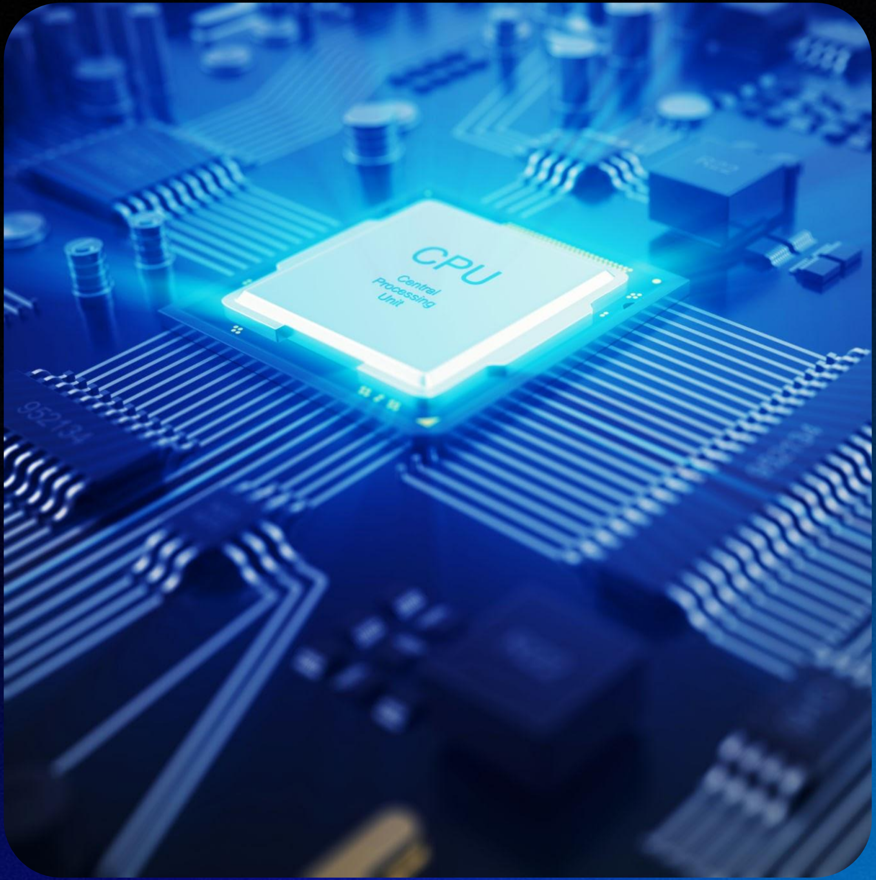
Graph of Model Training



Final Model Performance

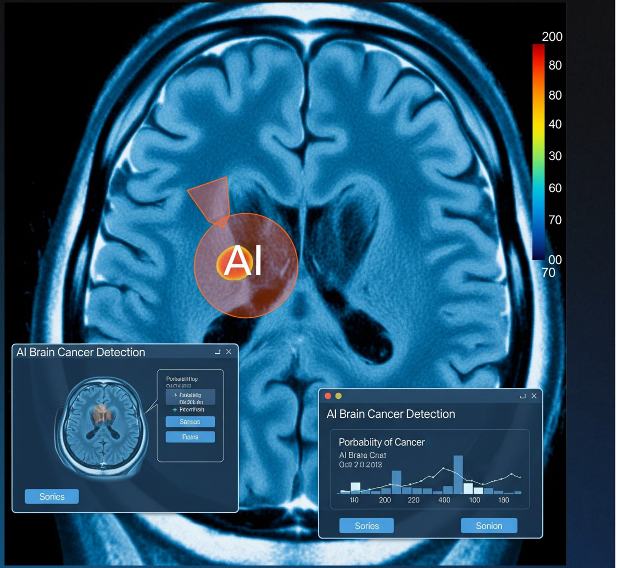
Metric	Value
Accuracy	83%
Epochs	5
Loss	0.4236%
Layers	10

The model showed steady improvement in accuracy over epochs and was able to classify the MRI images with 83% accuracy on the test set.



Demo: Brain Cancer Detection

LEARNING EXPERIENCE



What did our team learn

We learned how to train a CNN, work with MRI data, and improve accuracy using image preprocessing. It also helped us build real experience applying AI to a healthcare problem.

What was difficult

One of the challenges we faced during the project was deciding the optimal number of epochs and figuring out how to improve our data augmentation to boost model performance.

Overall is it ethical to rely on AI for medical diagnoses when mistakes could affect someone's health?

AI can support doctors by improving speed and accuracy, but it shouldn't replace human judgment especially in high-risk situations. Ethical use means AI should be a **tool**, not a decision-maker

Amanda Loeung

Hamza Harb

Melodie Cornelly

Thank you!