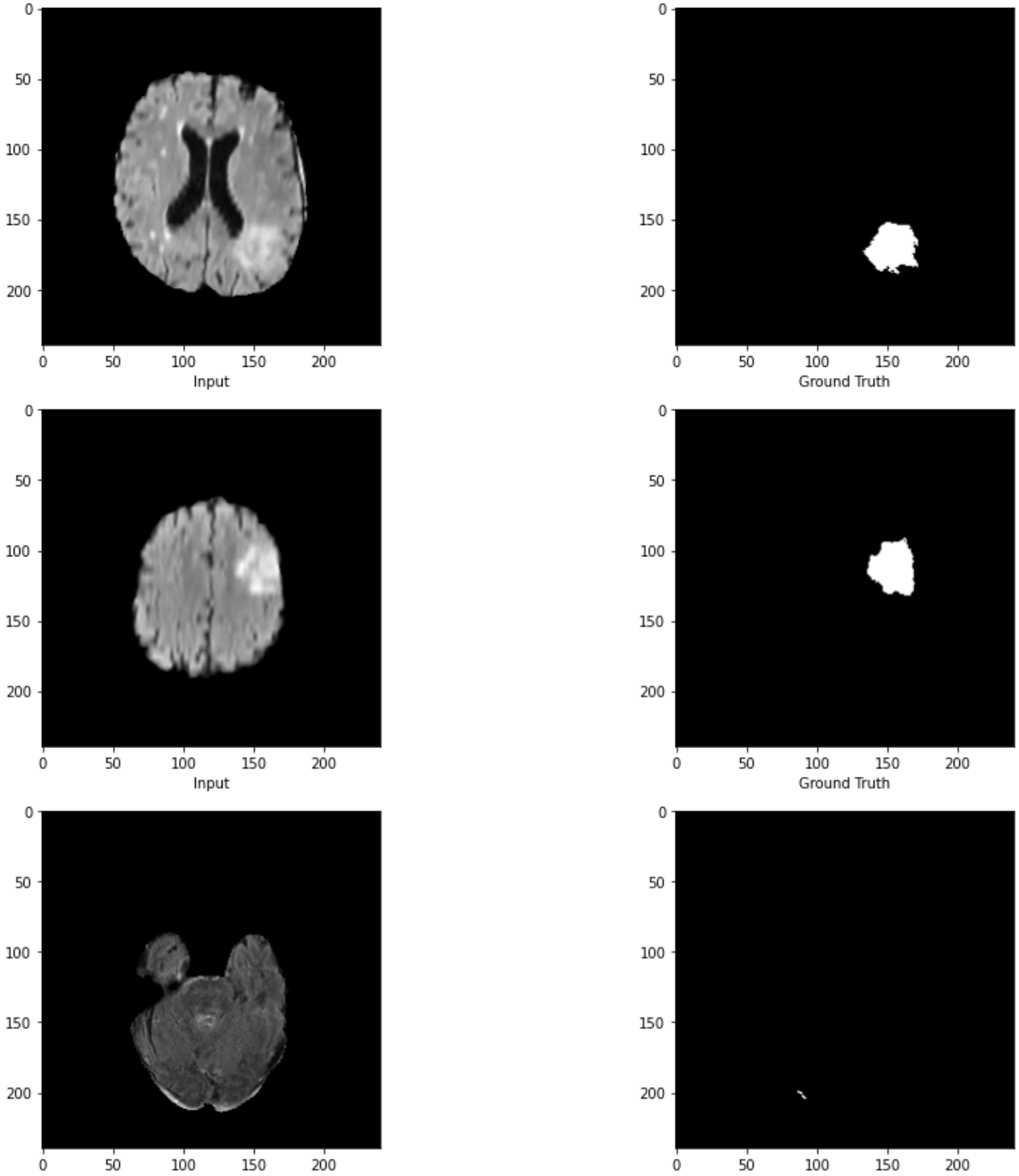
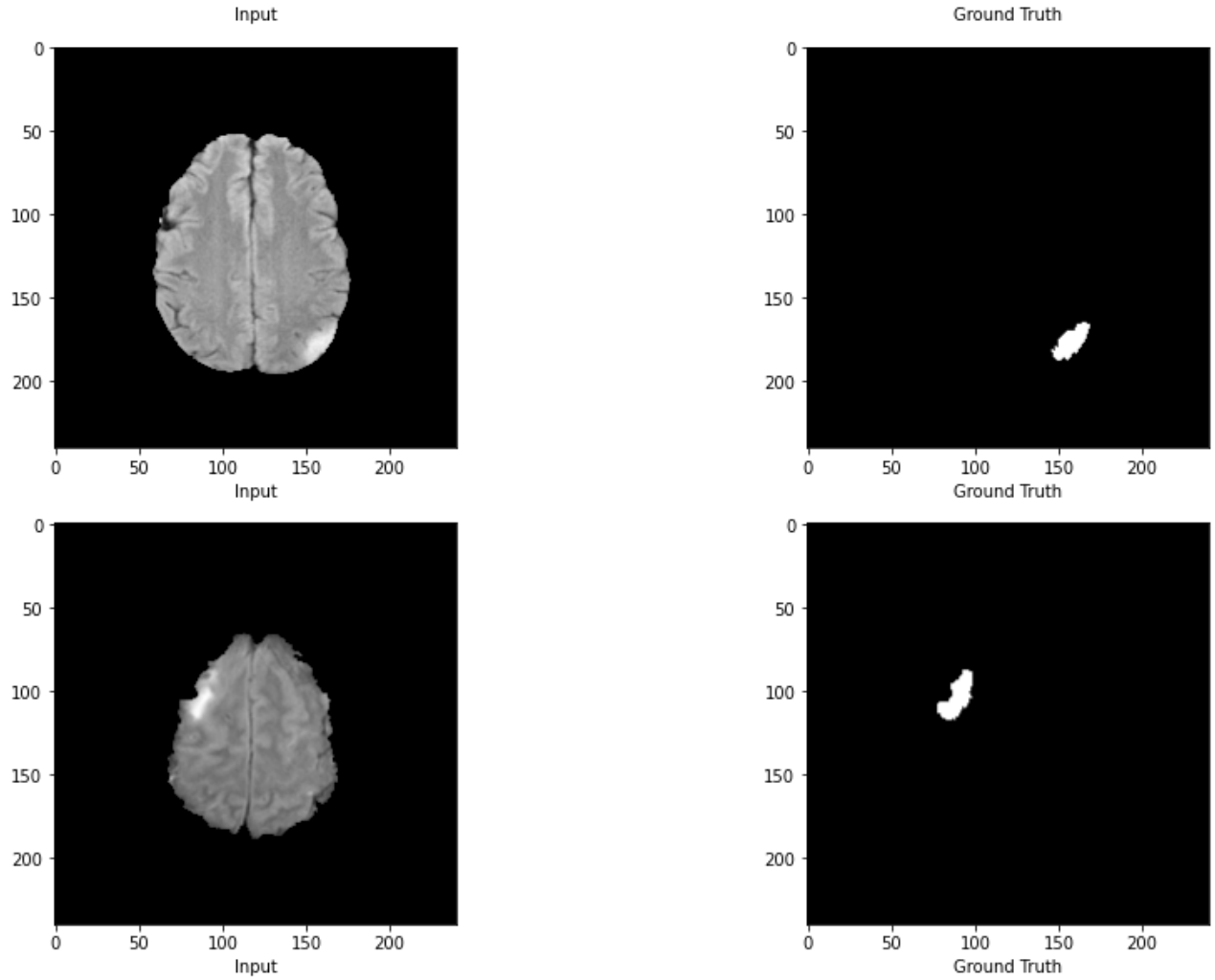
In [1]:
```
import numpy as np
from tqdm import tqdm
import os
import random
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score
import shutil
import tensorflow as tf
import tensorflow.keras.backend as K

from tensorflow.keras import layers
from tensorflow.keras.layers import *
from tensorflow.keras.models import *
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.models import load_model
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.compat.v1 import ConfigProto
from tensorflow.compat.v1 import InteractiveSession

config = ConfigProto()
config.gpu_options.allow_growth = True
session = InteractiveSession(config=config)
```

In [2]:
```
#uncomment if using linux/macos
!rm -rf TrainSeg ValSeg
!mkdir TrainSeg ValSeg TrainSeg/Yes ValSeg/Yes

#uncomment if using windows
#!rmdir TrainSeg ValSeg /s /q
#!md TrainSeg ValSeg TrainSeg\Yes ValSeg\Yes


img_path = 'Dataset/'
train_list = []
val_list = []
CLASS = 'Yes'
all_files = os.listdir(img_path + CLASS)
files = [item for item in all_files if "img" in item]
random.shuffle(files)
img_num = len(files)
for (n, file_name) in enumerate(files):
    img = os.path.join(img_path,CLASS,file_name)
    seg = os.path.join(img_path,CLASS,file_name.split('_')[0]+'_seg.npy')
    # 80% of images will be used for training, change the number here
    # to use different number of images for training your model.
    if n < 0.8*img_num:
        shutil.copy(img, os.path.join('TrainSeg/',CLASS,file_name))
        train_list.append(os.path.join('TrainSeg/',CLASS,file_name))
        shutil.copy(seg, os.path.join('TrainSeg/',CLASS,file_name.split('_')[0]+
    else:
        shutil.copy(img, os.path.join('ValSeg/',CLASS,file_name))
        val_list.append(os.path.join('ValSeg/',CLASS,file_name))
        shutil.copy(seg, os.path.join('ValSeg/',CLASS,file_name.split('_')[0]+'_
```

In [3]:
```
def plot_samples(x,n=10):
    i = n
    j = 2
    plt.figure(figsize=(15,20))
    k = 1
```

```
    idx_nums = np.random.randint(len(x),size=n)
    for idx in idx_nums:
        plt.subplot(i,j,k)
        while k%2 != 0:
            plt.imshow(np.load(x[idx])[:,:,0], cmap='gray')
            plt.xlabel("Input")
            k += 1
        plt.subplot(i,j,k)
        plt.imshow(np.load(x[idx].split('_')[0]+'_seg.npy')[:,:], cmap='gray')
        plt.xlabel("Ground Truth")
        k += 1
    plt.tight_layout()
    plt.show()

plot_samples(train_list, n=5)
```





In [4]:
```
class DataGenerator(tf.keras.utils.Sequence):
    def __init__(self, list_IDs, batch_size=2, dim=(240,240), n_channels=3,
                 n_classes=2, shuffle=True):
        self.dim = dim
        self.batch_size = batch_size
        self.list_IDs = list_IDs
        self.n_channels = n_channels
        self.n_classes = n_classes
        self.shuffle = shuffle
        self.on_epoch_end()

    def __len__(self):
        return int(np.floor(len(self.list_IDs) / self.batch_size))

    def __getitem__(self, index):
        # Generate indexes of the batch
        indexes = self.indexes[index*self.batch_size:(index+1)*self.batch_size]

        # Find list of IDs
        list_IDs_temp = [self.list_IDs[k] for k in indexes]

        # Generate data
        X, y = self.__data_generation(list_IDs_temp)

        return X, y

    def on_epoch_end(self):
```

```
        self.indexes = np.arange(len(self.list_IDs))
        if self.shuffle == True:
            np.random.shuffle(self.indexes)

    def __data_generation(self, list_IDs_temp):
        # X : (n_samples, *dim, n_channels)
        # Initialization
        X = np.empty((self.batch_size, *self.dim, self.n_channels))
        y = np.empty((self.batch_size, *self.dim))

        # Generate data
        for i, ID in enumerate(list_IDs_temp):
            # Store sample
            # Add data augmentation here
            X[i,] = np.load(ID)

            # Store segmentation map
            y[i] = np.load(ID[:-8] +'_seg.npy')

        return X, y
```

In [5]:
```
train_generator = DataGenerator(train_list)
validation_generator = DataGenerator(val_list)
IMG_SIZE = (240,240)
RANDOM_SEED = 100
```

In [6]:
```
def dice_score(y_true, y_pred, smooth=1):
    y_true_f = K.flatten(y_true)
    y_pred_f = K.flatten(y_pred)
    intersection = K.sum(y_true_f * y_pred_f)
    return (2. * intersection + smooth) / (K.sum(y_true_f) + K.sum(y_pred_f) + s
```

In [7]:
```
def unet(input_size = (240,240,3),base_filter_num=64):
    inputs = Input(input_size)
    conv0_0 = Conv2D(base_filter_num, 3, activation = 'relu', padding = 'same',
    conv0_0 = Conv2D(base_filter_num, 3, activation = 'relu', padding = 'same',
    pool1 = MaxPooling2D(pool_size=(2, 2))(conv0_0)

    conv1_0 = Conv2D(base_filter_num*2, 3, activation = 'relu', padding = 'same'
    conv1_0 = Conv2D(base_filter_num*2, 3, activation = 'relu', padding = 'same'
    pool2 = MaxPooling2D(pool_size=(2, 2))(conv1_0)

    up1_0 = Conv2DTranspose(base_filter_num, (2, 2), strides=(2, 2), padding='sa
    merge00_10 = concatenate([conv0_0,up1_0], axis=-1)
    conv0_1 = Conv2D(base_filter_num, 3, activation = 'relu', padding = 'same',
    conv0_1 = Conv2D(base_filter_num, 3, activation = 'relu', padding = 'same',

    conv2_0 = Conv2D(base_filter_num*4, 3, activation = 'relu', padding = 'same'
    conv2_0 = Conv2D(base_filter_num*4, 3, activation = 'relu', padding = 'same'
    pool3 = MaxPooling2D(pool_size=(2, 2))(conv2_0)

    up2_0 = Conv2DTranspose(base_filter_num*2, (2, 2), strides=(2, 2), padding='
    merge10_20 = concatenate([conv1_0,up2_0], axis=-1)
    conv1_1 = Conv2D(base_filter_num*2, 3, activation = 'relu', padding = 'same'
    conv1_1 = Conv2D(base_filter_num*2, 3, activation = 'relu', padding = 'same'

    up1_1 = Conv2DTranspose(base_filter_num, (2, 2), strides=(2, 2), padding='sa
    merge01_11 = concatenate([conv0_0,conv0_1,up1_1], axis=-1)
    conv0_2 = Conv2D(base_filter_num, 3, activation = 'relu', padding = 'same',
    conv0_2 = Conv2D(base_filter_num, 3, activation = 'relu', padding = 'same',
```

```
    conv3_0 = Conv2D(base_filter_num*8, 3, activation = 'relu', padding = 'same'
    conv3_0 = Conv2D(base_filter_num*8, 3, activation = 'relu', padding = 'same'
    pool4 = MaxPooling2D(pool_size=(2, 2))(conv3_0)

    up3_0 = Conv2DTranspose(base_filter_num*4, (2, 2), strides=(2, 2), padding='
    merge20_30 = concatenate([conv2_0,up3_0], axis=-1)
    conv2_1 = Conv2D(base_filter_num*4, 3, activation = 'relu', padding = 'same'
    conv2_1 = Conv2D(base_filter_num*4, 3, activation = 'relu', padding = 'same'

    up2_1 = Conv2DTranspose(base_filter_num*2, (2, 2), strides=(2, 2), padding='
    merge11_21 = concatenate([conv1_0,conv1_1,up2_1], axis=-1)
    conv1_2 = Conv2D(base_filter_num*2, 3, activation = 'relu', padding = 'same'
    conv1_2 = Conv2D(base_filter_num*2, 3, activation = 'relu', padding = 'same'

    up1_2 = Conv2DTranspose(base_filter_num, (2, 2), strides=(2, 2), padding='sa
    merge02_12 = concatenate([conv0_0,conv0_1,conv0_2,up1_2], axis=-1)
    conv0_3 = Conv2D(base_filter_num, 3, activation = 'relu', padding = 'same',
    conv0_3 = Conv2D(base_filter_num, 3, activation = 'relu', padding = 'same',

    conv4_0 = Conv2D(base_filter_num*16, 3, activation = 'relu', padding = 'same
    conv4_0 = Conv2D(base_filter_num*16, 3, activation = 'relu', padding = 'same

    up4_0 = Conv2DTranspose(base_filter_num*8, (2, 2), strides=(2, 2), padding='
    merge30_40 = concatenate([conv3_0,up4_0], axis = -1)
    conv3_1 = Conv2D(base_filter_num*8, 3, activation = 'relu', padding = 'same'
    conv3_1 = Conv2D(base_filter_num*8, 3, activation = 'relu', padding = 'same'

    up3_1 = Conv2DTranspose(base_filter_num*4, (2, 2), strides=(2, 2), padding='
    merge21_31 = concatenate([conv2_0,conv2_1,up3_1], axis = -1)
    conv2_2 = Conv2D(base_filter_num*4, 3, activation = 'relu', padding = 'same'
    conv2_2 = Conv2D(base_filter_num*4, 3, activation = 'relu', padding = 'same'

    up2_2 = Conv2DTranspose(base_filter_num*2, (2, 2), strides=(2, 2), padding='
    merge12_22 = concatenate([conv1_0,conv1_1,conv1_2,up2_2], axis = -1)
    conv1_3 = Conv2D(base_filter_num*2, 3, activation = 'relu', padding = 'same'
    conv1_3 = Conv2D(base_filter_num*2, 3, activation = 'relu', padding = 'same'

    up1_3 = Conv2DTranspose(base_filter_num, (2, 2), strides=(2, 2), padding='sa
    merge03_13 = concatenate([conv0_0,conv0_1,conv0_2,conv0_3,up1_3], axis = -1)
    conv0_4 = Conv2D(base_filter_num, 3, activation = 'relu', padding = 'same',
    conv0_4 = Conv2D(base_filter_num, 3, activation = 'relu', padding = 'same',
    # 二分类任务
    conv0_4 = Conv2D(1, 1, activation = 'sigmoid')(conv0_4)

    model = Model(inputs = inputs, outputs = conv0_4)
    model.compile(optimizer=Adam(lr = 1e-4), loss='binary_crossentropy', metrics
    model.summary()
    return model
```

In [16]:

```
model = unet()
history = model.fit(
    train_generator,
    epochs=40,
    validation_data=validation_generator,
)
```

```
Model: "model_2"
__________________________________________________________________________________
_________________
Layer (type)                 Output Shape              Param #     Connected to
```

```
==================================================================================================
input_3 (InputLayer)            [(None, 240, 240, 3) 0
__________________________________________________________________________________________________
conv2d_62 (Conv2D)              (None, 240, 240, 64) 1792        input_3[0][0]
__________________________________________________________________________________________________
conv2d_63 (Conv2D)              (None, 240, 240, 64) 36928       conv2d_62[0][0]
__________________________________________________________________________________________________
max_pooling2d_8 (MaxPooling2D)  (None, 120, 120, 64) 0           conv2d_63[0][0]
__________________________________________________________________________________________________
conv2d_64 (Conv2D)              (None, 120, 120, 128 73856       max_pooling2d_8
[0][0]
__________________________________________________________________________________________________
conv2d_65 (Conv2D)              (None, 120, 120, 128 147584      conv2d_64[0][0]
__________________________________________________________________________________________________
max_pooling2d_9 (MaxPooling2D)  (None, 60, 60, 128)  0           conv2d_65[0][0]
__________________________________________________________________________________________________
conv2d_68 (Conv2D)              (None, 60, 60, 256)  295168      max_pooling2d_9
[0][0]
__________________________________________________________________________________________________
conv2d_69 (Conv2D)              (None, 60, 60, 256)  590080      conv2d_68[0][0]
__________________________________________________________________________________________________
max_pooling2d_10 (MaxPooling2D) (None, 30, 30, 256)  0           conv2d_69[0][0]
__________________________________________________________________________________________________
conv2d_74 (Conv2D)              (None, 30, 30, 512)  1180160     max_pooling2d_1
0[0][0]
__________________________________________________________________________________________________
conv2d_75 (Conv2D)              (None, 30, 30, 512)  2359808     conv2d_74[0][0]
__________________________________________________________________________________________________
max_pooling2d_11 (MaxPooling2D) (None, 15, 15, 512)  0           conv2d_75[0][0]
__________________________________________________________________________________________________
conv2d_82 (Conv2D)              (None, 15, 15, 1024) 4719616     max_pooling2d_1
1[0][0]
__________________________________________________________________________________________________
conv2d_83 (Conv2D)              (None, 15, 15, 1024) 9438208     conv2d_82[0][0]
__________________________________________________________________________________________________
conv2d_transpose_26 (Conv2DTran (None, 30, 30, 512)  2097664     conv2d_83[0][0]
__________________________________________________________________________________________________
conv2d_transpose_23 (Conv2DTran (None, 60, 60, 256)  524544      conv2d_75[0][0]
__________________________________________________________________________________________________
concatenate_26 (Concatenate)    (None, 30, 30, 1024) 0           conv2d_75[0][0]
                                                                 conv2d_transpos
e_26[0][0]
__________________________________________________________________________________________________
conv2d_transpose_21 (Conv2DTran (None, 120, 120, 128 131200      conv2d_69[0][0]
__________________________________________________________________________________________________
```

```
concatenate_23 (Concatenate)    (None, 60, 60, 512)  0           conv2d_69[0][0]
                                                                 conv2d_transpos
e_23[0][0]
__________________________________________________________________________________
_____________________
conv2d_84 (Conv2D)              (None, 30, 30, 512)  4719104     concatenate_26
[0][0]
__________________________________________________________________________________
_____________________
conv2d_transpose_20 (Conv2DTran (None, 240, 240, 64) 32832       conv2d_65[0][0]
__________________________________________________________________________________
_____________________
concatenate_21 (Concatenate)    (None, 120, 120, 256 0           conv2d_65[0][0]
                                                                 conv2d_transpos
e_21[0][0]
__________________________________________________________________________________
_____________________
conv2d_76 (Conv2D)              (None, 60, 60, 256)  1179904     concatenate_23
[0][0]
__________________________________________________________________________________
_____________________
conv2d_85 (Conv2D)              (None, 30, 30, 512)  2359808     conv2d_84[0][0]
__________________________________________________________________________________
_____________________
concatenate_20 (Concatenate)    (None, 240, 240, 128 0           conv2d_63[0][0]
                                                                 conv2d_transpos
e_20[0][0]
__________________________________________________________________________________
_____________________
conv2d_70 (Conv2D)              (None, 120, 120, 128 295040      concatenate_21
[0][0]
__________________________________________________________________________________
_____________________
conv2d_77 (Conv2D)              (None, 60, 60, 256)  590080      conv2d_76[0][0]
__________________________________________________________________________________
_____________________
conv2d_transpose_27 (Conv2DTran (None, 60, 60, 256)  524544      conv2d_85[0][0]
__________________________________________________________________________________
_____________________
conv2d_66 (Conv2D)              (None, 240, 240, 64) 73792       concatenate_20
[0][0]
__________________________________________________________________________________
_____________________
conv2d_71 (Conv2D)              (None, 120, 120, 128 147584      conv2d_70[0][0]
__________________________________________________________________________________
_____________________
conv2d_transpose_24 (Conv2DTran (None, 120, 120, 128 131200      conv2d_77[0][0]
__________________________________________________________________________________
_____________________
concatenate_27 (Concatenate)    (None, 60, 60, 768)  0           conv2d_69[0][0]
                                                                 conv2d_77[0][0]
                                                                 conv2d_transpos
e_27[0][0]
__________________________________________________________________________________
_____________________
conv2d_67 (Conv2D)              (None, 240, 240, 64) 36928       conv2d_66[0][0]
__________________________________________________________________________________
_____________________
conv2d_transpose_22 (Conv2DTran (None, 240, 240, 64) 32832       conv2d_71[0][0]
__________________________________________________________________________________
_____________________
concatenate_24 (Concatenate)    (None, 120, 120, 384 0           conv2d_65[0][0]
                                                                 conv2d_71[0][0]
                                                                 conv2d_transpos
e_24[0][0]
```

```
__________________________________________________________________________________________________
____________________
conv2d_86 (Conv2D)              (None, 60, 60, 256)  1769728     concatenate_27
[0][0]
__________________________________________________________________________________________________
____________________
concatenate_22 (Concatenate)    (None, 240, 240, 192 0           conv2d_63[0][0]
                                                                 conv2d_67[0][0]
                                                                 conv2d_transpos
e_22[0][0]
__________________________________________________________________________________________________
____________________
conv2d_78 (Conv2D)              (None, 120, 120, 128 442496      concatenate_24
[0][0]
__________________________________________________________________________________________________
____________________
conv2d_87 (Conv2D)              (None, 60, 60, 256)  590080      conv2d_86[0][0]
__________________________________________________________________________________________________
____________________
conv2d_72 (Conv2D)              (None, 240, 240, 64) 110656      concatenate_22
[0][0]
__________________________________________________________________________________________________
____________________
conv2d_79 (Conv2D)              (None, 120, 120, 128 147584      conv2d_78[0][0]
__________________________________________________________________________________________________
____________________
conv2d_transpose_28 (Conv2DTran (None, 120, 120, 128 131200      conv2d_87[0][0]
__________________________________________________________________________________________________
____________________
conv2d_73 (Conv2D)              (None, 240, 240, 64) 36928       conv2d_72[0][0]
__________________________________________________________________________________________________
____________________
conv2d_transpose_25 (Conv2DTran (None, 240, 240, 64) 32832       conv2d_79[0][0]
__________________________________________________________________________________________________
____________________
concatenate_28 (Concatenate)    (None, 120, 120, 512 0           conv2d_65[0][0]
                                                                 conv2d_71[0][0]
                                                                 conv2d_79[0][0]
                                                                 conv2d_transpos
e_28[0][0]
__________________________________________________________________________________________________
____________________
concatenate_25 (Concatenate)    (None, 240, 240, 256 0           conv2d_63[0][0]
                                                                 conv2d_67[0][0]
                                                                 conv2d_73[0][0]
                                                                 conv2d_transpos
e_25[0][0]
__________________________________________________________________________________________________
____________________
conv2d_88 (Conv2D)              (None, 120, 120, 128 589952      concatenate_28
[0][0]
__________________________________________________________________________________________________
____________________
conv2d_80 (Conv2D)              (None, 240, 240, 64) 147520      concatenate_25
[0][0]
__________________________________________________________________________________________________
____________________
conv2d_89 (Conv2D)              (None, 120, 120, 128 147584      conv2d_88[0][0]
__________________________________________________________________________________________________
____________________
conv2d_81 (Conv2D)              (None, 240, 240, 64) 36928       conv2d_80[0][0]
__________________________________________________________________________________________________
____________________
conv2d_transpose_29 (Conv2DTran (None, 240, 240, 64) 32832       conv2d_89[0][0]
__________________________________________________________________________________________________
```

```
__________________
concatenate_29 (Concatenate)    (None, 240, 240, 320 0           conv2d_63[0][0]
                                                                 conv2d_67[0][0]
                                                                 conv2d_73[0][0]
                                                                 conv2d_81[0][0]
                                                                 conv2d_transpos
e_29[0][0]
__________________________________________________________________________________
__________________
conv2d_90 (Conv2D)              (None, 240, 240, 64) 184384      concatenate_29
[0][0]
__________________________________________________________________________________
__________________
conv2d_91 (Conv2D)              (None, 240, 240, 64) 36928       conv2d_90[0][0]
__________________________________________________________________________________
__________________
conv2d_92 (Conv2D)              (None, 240, 240, 1)  65          conv2d_91[0][0]
==================================================================================
==================
Total params: 36,157,953
Trainable params: 36,157,953
Non-trainable params: 0
__________________________________________________________________________________
__________________
Epoch 1/40
182/182 [==============================] - 21s 115ms/step - loss: 0.0599 - dice_
score: 0.5202 - val_loss: 0.0344 - val_dice_score: 0.5549
Epoch 2/40
182/182 [==============================] - 20s 108ms/step - loss: 0.0343 - dice_
score: 0.6164 - val_loss: 0.0330 - val_dice_score: 0.5898
Epoch 3/40
182/182 [==============================] - 20s 108ms/step - loss: 0.0282 - dice_
score: 0.6650 - val_loss: 0.0239 - val_dice_score: 0.7227
Epoch 4/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0250 - dice_
score: 0.7033 - val_loss: 0.0245 - val_dice_score: 0.7010
Epoch 5/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0200 - dice_
score: 0.7430 - val_loss: 0.0245 - val_dice_score: 0.6933
Epoch 6/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0187 - dice_
score: 0.7505 - val_loss: 0.0187 - val_dice_score: 0.7558
Epoch 7/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0163 - dice_
score: 0.7845 - val_loss: 0.0175 - val_dice_score: 0.7507
Epoch 8/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0148 - dice_
score: 0.7901 - val_loss: 0.0178 - val_dice_score: 0.7862
Epoch 9/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0147 - dice_
score: 0.8046 - val_loss: 0.0223 - val_dice_score: 0.7491
Epoch 10/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0160 - dice_
score: 0.7874 - val_loss: 0.0157 - val_dice_score: 0.7908
Epoch 11/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0141 - dice_
score: 0.8055 - val_loss: 0.0135 - val_dice_score: 0.8117
Epoch 12/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0138 - dice_
score: 0.8197 - val_loss: 0.0148 - val_dice_score: 0.7822
Epoch 13/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0111 - dice_
score: 0.8510 - val_loss: 0.0114 - val_dice_score: 0.8206
Epoch 14/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0099 - dice_
```

```
score: 0.8657 - val_loss: 0.0104 - val_dice_score: 0.8397
Epoch 15/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0093 - dice_
score: 0.8629 - val_loss: 0.0107 - val_dice_score: 0.8510
Epoch 16/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0087 - dice_
score: 0.8858 - val_loss: 0.0176 - val_dice_score: 0.7586
Epoch 17/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0088 - dice_
score: 0.8681 - val_loss: 0.0096 - val_dice_score: 0.8517
Epoch 18/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0076 - dice_
score: 0.8860 - val_loss: 0.0093 - val_dice_score: 0.8789
Epoch 19/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0104 - dice_
score: 0.8613 - val_loss: 0.0152 - val_dice_score: 0.7911
Epoch 20/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0091 - dice_
score: 0.8635 - val_loss: 0.0096 - val_dice_score: 0.8556
Epoch 21/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0068 - dice_
score: 0.9010 - val_loss: 0.0090 - val_dice_score: 0.8667
Epoch 22/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0061 - dice_
score: 0.9141 - val_loss: 0.0088 - val_dice_score: 0.8818
Epoch 23/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0058 - dice_
score: 0.9104 - val_loss: 0.0096 - val_dice_score: 0.8741
Epoch 24/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0056 - dice_
score: 0.9071 - val_loss: 0.0087 - val_dice_score: 0.8930
Epoch 25/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0051 - dice_
score: 0.9239 - val_loss: 0.0096 - val_dice_score: 0.8966
Epoch 26/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0052 - dice_
score: 0.9195 - val_loss: 0.0105 - val_dice_score: 0.8611
Epoch 27/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0132 - dice_
score: 0.8337 - val_loss: 0.0211 - val_dice_score: 0.7468
Epoch 28/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0125 - dice_
score: 0.8346 - val_loss: 0.0097 - val_dice_score: 0.8828
Epoch 29/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0057 - dice_
score: 0.9142 - val_loss: 0.0090 - val_dice_score: 0.8753
Epoch 30/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0051 - dice_
score: 0.9198 - val_loss: 0.0090 - val_dice_score: 0.8855
Epoch 31/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0043 - dice_
score: 0.9357 - val_loss: 0.0095 - val_dice_score: 0.8993
Epoch 32/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0040 - dice_
score: 0.9423 - val_loss: 0.0096 - val_dice_score: 0.8805
Epoch 33/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0037 - dice_
score: 0.9437 - val_loss: 0.0103 - val_dice_score: 0.8888
Epoch 34/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0034 - dice_
score: 0.9479 - val_loss: 0.0099 - val_dice_score: 0.9037
Epoch 35/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0033 - dice_
score: 0.9478 - val_loss: 0.0108 - val_dice_score: 0.9086
Epoch 36/40
```

```
182/182 [==============================] - 20s 109ms/step - loss: 0.0033 - dice_
score: 0.9466 - val_loss: 0.0099 - val_dice_score: 0.8890
Epoch 37/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0032 - dice_
score: 0.9533 - val_loss: 0.0103 - val_dice_score: 0.8948
Epoch 38/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0033 - dice_
score: 0.9536 - val_loss: 0.0104 - val_dice_score: 0.8786
Epoch 39/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0030 - dice_
score: 0.9493 - val_loss: 0.0105 - val_dice_score: 0.9066
Epoch 40/40
182/182 [==============================] - 20s 109ms/step - loss: 0.0029 - dice_
score: 0.9527 - val_loss: 0.0100 - val_dice_score: 0.8996
```

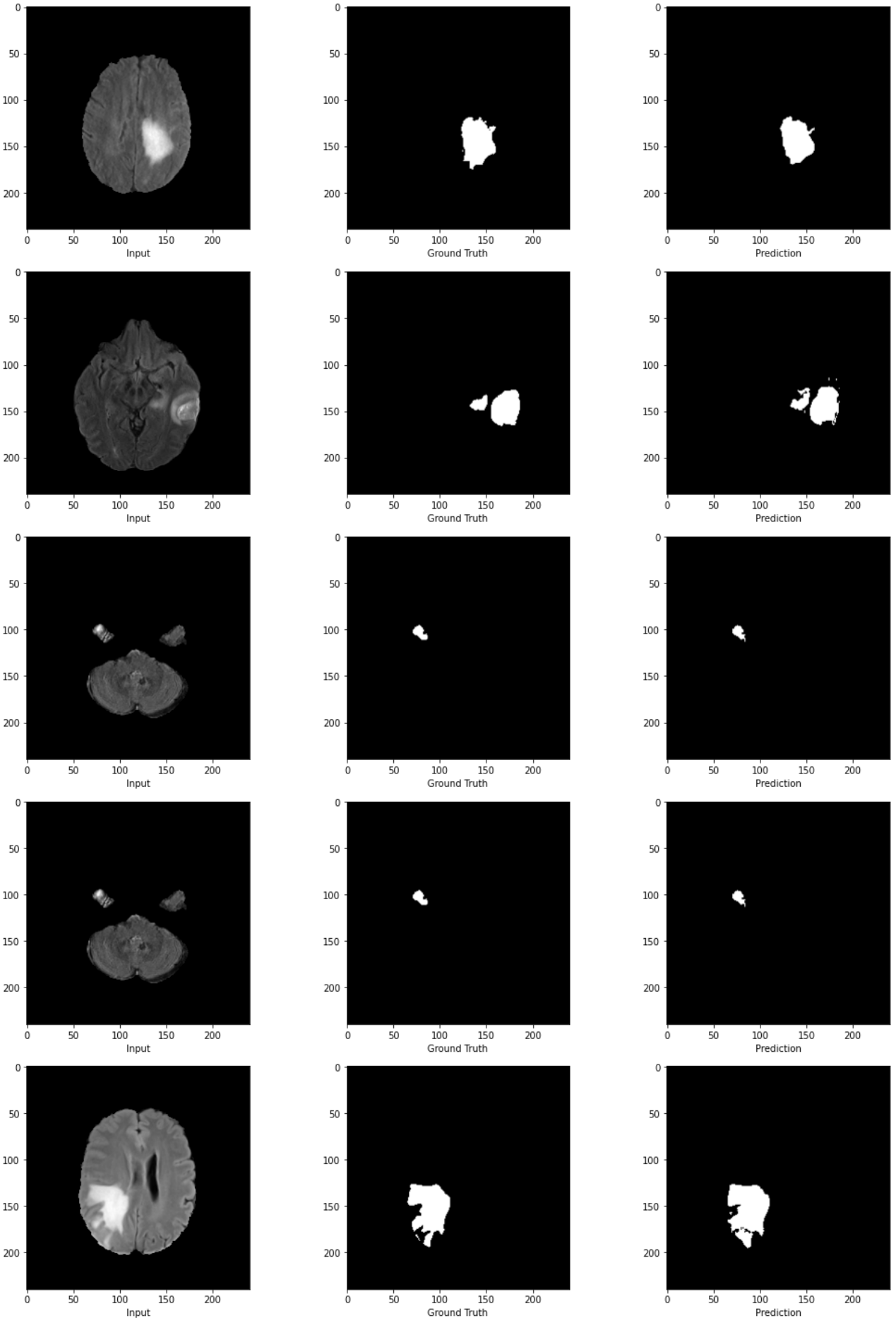
In [17]:
```
model.save('trained_seg_model.h5')
```

In [18]:
```
test_dir = 'ValSeg/'
#load your model here
dependencies = {
    'dice_score': dice_score
}
model = load_model('trained_seg_model.h5', custom_objects=dependencies)
test_list = []
CLASS = 'Yes'
all_files = os.listdir(test_dir + CLASS)
files = [item for item in all_files if "img" in item]
for file_name in files:
    test_list.append(test_dir + CLASS + '/' + file_name)
test_generator = DataGenerator(test_list[:100], batch_size=1)

predictions = []
x_test = []
y_test = []
accuracy = []
for i in range(test_generator.__len__()):
    x, y = test_generator.__getitem__(i)
    x_test.append(x)
    y_test.append(y[0])
    prediction = model.predict(x)
    prediction[prediction>0.5] = 1
    prediction[prediction<=0.5] = 0
    predictions.append(prediction[0])
    accuracy.append(dice_score(y[0], prediction[0].astype('float64')))
print('Test Score = %.2f' % np.mean(accuracy))
```

```
Test Score = 0.85
```

In [19]:
```
def plot_result(x,y,pred,n=10):
    i = n
    j = 3
    plt.figure(figsize=(15,20))
    k = 1
    idx_nums = np.random.randint(len(x),size=n)
    for idx in idx_nums:
        while k%3 != 0:
            plt.subplot(i,j,k)
            if k%3 == 1:
                plt.imshow(x[idx][0,:,:,0], cmap='gray')
                plt.xlabel("Input")
            if k%3 == 2:
```

```
                plt.imshow(y[idx][:,:], cmap='gray')
                plt.xlabel("Ground Truth")
            k += 1
        plt.subplot(i,j,k)
        plt.imshow(pred[idx][:,:,0], cmap='gray')
        plt.xlabel("Prediction")
        k += 1
    plt.tight_layout()
    plt.show()

plot_result(x_test, y_test, predictions, n=5)
```


Input
Ground Truth
Prediction
Input
Ground Truth
Prediction
Input
Ground Truth
Prediction
Input
Ground Truth
Prediction
Input
Ground Truth
Prediction

In [ ]: