Radiology AI Project Documentation

BATCH_SIZE = 64

Here I have utilized the transfer learning with VGG19 model trained on 1000 class ‘Imagenet’ dataset and the top is removed to customize according to the number of output I want. Below image is for VGG19 transfer learning model bottom.

```
pretrained_model = tf.keras.applications.VGG19(weights = 'imagenet',
                                               classes = 1000,
                                               input_shape = (224, 224, 3),
                                               include_top = False,
                                               pooling = 'max')
```

In below image, I’m using 2 dense layers with top layer/output layer with 3 neurones and softmax activation function since I’m classifying as 3 categories(3 outputs).

And also used Adam optimiser instead of SGD. Since Adam performed well on current task. Also used learning rate of 0.0001.

```
# Adding a prediction layer. It takes input from the last layer (global_max_pooling2d) of MobileNet
# It has 2 dense units, as it is a binary classification problem
predictions1 = Dense(128, activation = 'relu')(pretrained_model.output)
predictions2 = Dense(3, activation = 'softmax')(predictions1)


# Defining new model's input and output layers
# Input layer of the new model will be the same as MobileNet
# But the output of the new model will be the output of final dense layer, i.e., 2 units

model = Model(inputs = pretrained_model.input, outputs = predictions2)


# We use the SGD optimiser, with a very low learning rate, and loss function which is specific to two class classification

model.compile(optimizer = tf.keras.optimizers.Adam(0.0001),
              loss = "binary_crossentropy",
              metrics = ["accuracy"])
```

Above model was trained to 25 epoches which gave best accuracy and performance.

```
history = model.fit(train_datagen,
                    epochs = 25,
                    steps_per_epoch = (len(train_datagen)),
                    validation_data = val_datagen,
                    validation_steps = (len(val_datagen)),
                    shuffle = False,
                    callbacks = callback)
```