Fundamental of ML Final Project: Handwritten Image Classifier

Library imports

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
import tensorflow.keras as keras
import cv2
from sklearn.preprocessing import MinMaxScaler, OneHotEncoder
from sklearn.model_selection import train_test_split
import time
```

Unused Functions

In [947…

```
def vert_det(im, kernel):
    vert = cv2.Sobel(im, ddepth=cv2.CV_64F, dx=0, dy=1, ksize=kernel)
    return im-vert

def hor_det(im, kernel):
    hor = cv2.Sobel(im, ddepth=cv2.CV_64F, dx=1, dy=0, ksize=kernel)
    return im-hor

def binarize(im):
    threshold = np.max(im)-.6
    im[im < threshold] = 0
    im[im >= threshold] = 1
    return im

def otsu(im):
    im = im.astype("uint8")
    _, r = cv2.threshold(im, 0, 255, cv2.THRESH_BINARY+cv2.THRESH_OTSU)
    return r

def bounding_box_transform(im):
    tt = np.argmax(im, axis=0)
    leftbound = np.amin(np.nonzero(tt))
    rightbound = np.amax(np.nonzero(tt))
    t = np.argmax(im, axis=1)
    upbound = np.amin(np.nonzero(t))
    downbound = np.amax(np.nonzero(t))
    boundarybox = np.float32([[leftbound-10,upbound-10],[rightbound+10,upbound-10]
                              ,[leftbound-10,downbound+10],[rightbound+10,downbound+10]
    newbox = np.float32([[0,0],[300,0],[0,300],[300,300]])
    M = cv2.getPerspectiveTransform(boundarybox,newbox)
    dst = cv2.warpPerspective(im,M,(300,300))
    return dst

# takes in a non-inverted image (non min max normalized)
def remove_lines(z):
    z = z.reshape(300,300)
    z = update_type(z)
    #threshold the image
    thresh = cv2.threshold(z, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)[1]

    # Remove horizontal
    horizontal_kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (50,1))
    detected_lines = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, horizontal_kernel, iterat
    cnts = cv2.findContours(detected_lines, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    cnts = cnts[0] if len(cnts) == 2 else cnts[1]
    for c in cnts:
        cv2.drawContours(z, [c], -1, (255,255,255), 2)

    # Repair image
    repair_kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (1,6))
    result = 255 - cv2.morphologyEx(255 - z, cv2.MORPH_CLOSE, repair_kernel, iterations

    #threshold the image
    thresh = cv2.threshold(result, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)[1]

    # Remove vertical
    vertical_kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (1,50))
    detected_lines = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, vertical_kernel, iteratio
    cnts = cv2.findContours(detected_lines, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    cnts = cnts[0] if len(cnts) == 2 else cnts[1]
    for c in cnts:
        cv2.drawContours(result, [c], -1, (255,255,255), 2)

    # Repair image
    repair_kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (6,1))
    result = 255 - cv2.morphologyEx(255 - result, cv2.MORPH_CLOSE, repair_kernel, itera

    return result
```

Used Functions

In [948…

```
def show(im):
    plt.imshow(im, cmap='gray')
    plt.show()

def invert(im):
    return (im*-1)+255

def min_max_scale(im):
    im = MinMaxScaler().fit_transform(im.ravel().reshape(-1,1))
    return im.reshape(300,300)

def brighten(im):
    m = np.max(im)-.3
    im[im >= m] += 0.3
    im[im < m] -= 0.3
    return im

def blur(im, kernel):
    im = im.astype("uint8")
    im = cv2.medianBlur(im, kernel)
    return im

def morph_close(im, kernel, i):
    return cv2.morphologyEx(im, cv2.MORPH_CLOSE, kernel, iterations=i)

def morph_open(im, kernel, i):
    return cv2.morphologyEx(im, cv2.MORPH_OPEN, kernel, iterations=i)

def morph_dilate(im, kernel, i):
    return cv2.dilate(im, kernel, iterations=i)

def morph_erode(im, kernel, i):
    return cv2.erode(im, kernel, iterations=i)

def transform(im):
    dest = np.float32([[0,0],[300,0],[0,300],[300,300]])
    source = np.float32([[25,25],[275,25],[25,275],[275,275]])
    res = cv2.getPerspectiveTransform(source,dest)
    return cv2.warpPerspective(im, res, (300,300))
```

In [991…

```
def preprocess(ims):
    r = []
    for im in ims.T:
        im = im.reshape((300,300))
        im = transform(im)
        im = invert(im)
        im = blur(im, 5)
        im = morph_erode(im, (3,3), 5)
        im = morph_dilate(im, (3,3), 5)
        im = morph_close(im, (3,3), 5)
        im = morph_open(im, (3,3), 5)
        im = min_max_scale(im)
        im = brighten(im)
        im = min_max_scale(im)
        im = cv2.resize(im, (100,100))
        im = np.stack((im,)*3, axis=-1)
        r.append(im)
    return np.array(r)

def augment(ims, labels):
    result = []
    n = []
    o = []
    t = []
    for im in ims:
        im = cv2.rotate(im, cv2.cv2.ROTATE_90_CLOCKWISE)
        n.append(im)
        im = cv2.rotate(im, cv2.cv2.ROTATE_90_CLOCKWISE)
        o.append(im)
        im = cv2.rotate(im, cv2.cv2.ROTATE_90_CLOCKWISE)
        t.append(im)
    result.append(ims)
    result.append(n)
    result.append(o)
    result.append(t)
    labels_new = np.concatenate((labels,labels,labels,labels))
    return np.array(result).reshape(-1,100,100), labels_new
```

Data imports

In [103…

```
data = np.load("data_train.npy")
labels = np.load("labels_train.npy")
```

In [103…

```
cnn_input, labels_new = augment(preprocess(data), labels)
```

In [107…

```
cnn_input = cnn_input.reshape(-1,100,100,3)
```

In [100…

```
# train test split twice to get the splits
# train_x, inter_x, train_y, inter_y = train_test_split(data_rot_augment, labels_augmen
x_train, x_inter, labels_train, labels_inter = train_test_split(cnn_input, labels_new,
```

In [100…

```
x_val, x_test, labels_val, labels_test = train_test_split(x_inter, labels_inter, train_
```

Ensuring that data is in the proper format for input into Xception

In [100…

```
x_train = x_train.reshape(-1, 100, 100, 3)
x_val = x_val.reshape(-1, 100, 100, 3)
x_test = x_test.reshape(-1, 100, 100, 3)
```

In [101…

```
x_train.shape, x_val.shape, x_test.shape
```

Out[101…

```
((18816, 100, 100, 3), (4032, 100, 100, 3), (4032, 100, 100, 3))
```

In [101…

```
labels_train_ohe = OneHotEncoder().fit_transform(labels_train.reshape(-1,1)).toarray()
labels_val_ohe = OneHotEncoder().fit_transform(labels_val.reshape(-1,1)).toarray()
labels_test_ohe = OneHotEncoder().fit_transform(labels_test.reshape(-1,1)).toarray()
```

In [101…

```
# follow the following format
cnn_dataset = (x_train, labels_train_ohe, x_val, labels_val_ohe, x_test, labels_test_oh
```

CNN training and architecture

In [966…

```
# Let's create a custom callback class
class PerfEvalCustomCallback(keras.callbacks.Callback):

    def __init__(self, perf_data):
        self.perf_data = perf_data

    # we define the on_epoch_end callback and save the loss and accuracy in perf_data
    def on_epoch_end(self, epoch, logs=None):
        self.perf_data[epoch,0] = logs['loss']
        self.perf_data[epoch,1] = logs['accuracy']
        self.perf_data[epoch,2] = logs['val_loss']
        self.perf_data[epoch,3] = logs['val_accuracy']

    def get_perf_data():
        return self.perf_data
```

In [967…

```
# Plot the model's performance during training (across epochs)
def plot_training_perf(train_loss, train_acc, val_loss, val_acc, fs=(8,5)):
    plt.figure(figsize=fs)


    assert train_loss.shape == val_loss.shape and train_loss.shape == val_acc.shape and

    # assume we have one measurement per epoch
    num_epochs = train_loss.shape[0]
    epochs = np.arange(0, num_epochs)

    # Can you figure out why this makes sense? Why remove -0.5?
    plt.plot(epochs-0.5, train_loss, 'm', linewidth=2,  label='Loss (Training)')
    plt.plot(epochs-0.5, train_acc, 'r--', linewidth=2, label='Accuracy (Training)')

    plt.plot(epochs, val_loss, 'g', linewidth=2, label='Loss (Validation)')
    plt.plot(epochs, val_acc, 'b:', linewidth=2, label='Accuracy (Validation)')


    plt.xlim([0, num_epochs])
    plt.ylim([0, 1.05])

    plt.legend()

    plt.show()
```

In [101…

```
# Customize this function as you like but makes sure it is implemented correctly.
# Note: If you need to change the method definition to add more arguments, make sure to
# the new arguments optional (and have a sensible default value)

from sklearn.metrics import classification_report, balanced_accuracy_score

def evaluate_model(name, model, eval_data,
                   plot_training=True, evaluate_on_test_set=True):

    # unpack the stuff
    perf_data, dataset = eval_data
    train_x, train_y, val_x, val_y, test_x, test_y = dataset

    # get predictions from the model
    train_preds = model.predict(train_x)
    val_preds = model.predict(val_x)

    # measure the accuracy (as categorical accuracy since we have a softmax layer)
    catacc_metric = keras.metrics.CategoricalAccuracy()
    catacc_metric.update_state(train_y, train_preds)
    train_acc = catacc_metric.result()

    catacc_metric = keras.metrics.CategoricalAccuracy()
    catacc_metric.update_state(val_y, val_preds)
    val_acc = catacc_metric.result()
    print('[{}] Training Accuracy: {:.3f}%, Validation Accuracy: {:.3f}%'.format(name,

    if plot_training:
        plot_training_perf(perf_data[:,0], perf_data[:,1], perf_data[:,2], perf_data[:,

    if evaluate_on_test_set:
        ### Evaluate the model on test data and put the results in 'test_loss', 'test_a
        ###* put your code here (~1-2 lines) *###

        test_preds = model.predict(test_x)

        accuracy_obj = keras.metrics.CategoricalAccuracy()
        accuracy_obj.update_state(test_y, test_preds)
        test_acc = accuracy_obj.result()

        test_loss, _ = model.evaluate(test_x, test_y) #the test_acc could also be extra

        cat_loss_obj = tf.keras.losses.CategoricalCrossentropy()
        test_ce_loss = cat_loss_obj(test_y, test_preds)


        print('[{}]  Test loss: {:.5f}; test accuracy: {:.3f}%'.format(name, test_loss,
        print('[{}]  Test cross entropy loss: {:.5f}'.format(name, test_ce_loss))



    # You can add stuff here if you want
    ###* put your code here (0+ lines) *###


    return
```

In [ ]:

```
evaluate_model("CNN", cnn_model, eval_data)
```

In [108…

```
base_model = keras.applications.Xception(
weights='imagenet',
input_shape=(100,100,3),
include_top=False)

base_model.trainable=False
```

In [109…

```
inputs = keras.Input(shape=(100,100,3))
x = base_model.output
x = keras.layers.GlobalAveragePooling2D()(x)
x = keras.layers.Dense(200, activation='relu')(x)
x = keras.layers.Dense(100, activation='relu')(x)
x = keras.layers.Dense(50, activation='relu')(x)
outputs = keras.layers.Dense(10, activation='softmax')(x)
model = keras.Model(inputs=base_model.inputs, outputs=outputs)
```

In [109…

```
for layer in model.layers:
    layer.trainable = True
```

In [109…

```
model.compile(loss='categorical_crossentropy', optimizer='nadam', metrics=['accuracy'])
```

In [108…

```
perf_data = np.zeros((10, 4))
perf_eval_cb = PerfEvalCustomCallback(perf_data)
early_stop_cb = keras.callbacks.EarlyStopping(monitor='loss', mode='min', patience=4)
```

In [108…

```
model.summary()
```

```
Model: "model_8"
__________________________________________________________________________________________
__________
 Layer (type)                   Output Shape         Param #     Connected to
==========================================================================================
==========
 input_30 (InputLayer)          [(None, 100, 100, 3  0           []
                                )]

 block1_conv1 (Conv2D)          (None, 49, 49, 32)   864         ['input_30[0][0]']

 block1_conv1_bn (BatchNormaliz  (None, 49, 49, 32)  128         ['block1_conv1[0][0]']
 ation)

 block1_conv1_act (Activation)  (None, 49, 49, 32)   0           ['block1_conv1_bn[0]
[0]']

 block1_conv2 (Conv2D)          (None, 47, 47, 64)   18432       ['block1_conv1_act[0]
[0]']

 block1_conv2_bn (BatchNormaliz  (None, 47, 47, 64)  256         ['block1_conv2[0][0]']
 ation)

 block1_conv2_act (Activation)  (None, 47, 47, 64)   0           ['block1_conv2_bn[0]
[0]']

 block2_sepconv1 (SeparableConv  (None, 47, 47, 128)  8768       ['block1_conv2_act[0]
[0]']
 2D)

 block2_sepconv1_bn (BatchNorma  (None, 47, 47, 128)  512        ['block2_sepconv1[0]
[0]']
 lization)

 block2_sepconv2_act (Activatio  (None, 47, 47, 128)  0          ['block2_sepconv1_bn[0]
[0]']
 n)

 block2_sepconv2 (SeparableConv  (None, 47, 47, 128)  17536      ['block2_sepconv2_act
[0][0]']
 2D)

 block2_sepconv2_bn (BatchNorma  (None, 47, 47, 128)  512        ['block2_sepconv2[0]
[0]']
 lization)

 conv2d_16 (Conv2D)             (None, 24, 24, 128)  8192        ['block1_conv2_act[0]
[0]']

 block2_pool (MaxPooling2D)     (None, 24, 24, 128)  0           ['block2_sepconv2_bn[0]
[0]']

 batch_normalization_70 (BatchN  (None, 24, 24, 128)  512        ['conv2d_16[0][0]']
 ormalization)

 add_48 (Add)                   (None, 24, 24, 128)  0           ['block2_pool[0][0]',
                                                                  'batch_normalization_7
0[0][0]']

 block3_sepconv1_act (Activatio  (None, 24, 24, 128)  0          ['add_48[0][0]']
 n)

 block3_sepconv1 (SeparableConv  (None, 24, 24, 256)  33920      ['block3_sepconv1_act
[0][0]']
 2D)

 block3_sepconv1_bn (BatchNorma  (None, 24, 24, 256)  1024       ['block3_sepconv1[0]
[0]']
 lization)

 block3_sepconv2_act (Activatio  (None, 24, 24, 256)  0          ['block3_sepconv1_bn[0]
[0]']
 n)

 block3_sepconv2 (SeparableConv  (None, 24, 24, 256)  67840      ['block3_sepconv2_act
[0][0]']
 2D)

 block3_sepconv2_bn (BatchNorma  (None, 24, 24, 256)  1024       ['block3_sepconv2[0]
[0]']
 lization)

 conv2d_17 (Conv2D)             (None, 12, 12, 256)  32768      ['add_48[0][0]']

 block3_pool (MaxPooling2D)     (None, 12, 12, 256)  0          ['block3_sepconv2_bn[0]
[0]']

 batch_normalization_71 (BatchN  (None, 12, 12, 256)  1024       ['conv2d_17[0][0]']
 ormalization)

 add_49 (Add)                   (None, 12, 12, 256)  0          ['block3_pool[0][0]',
                                                                  'batch_normalization_7
1[0][0]']

 block4_sepconv1_act (Activatio  (None, 12, 12, 256)  0          ['add_49[0][0]']
 n)

 block4_sepconv1 (SeparableConv  (None, 12, 12, 728)  188672     ['block4_sepconv1_act
[0][0]']
 2D)

 block4_sepconv1_bn (BatchNorma  (None, 12, 12, 728)  2912       ['block4_sepconv1[0]
[0]']
 lization)

 block4_sepconv2_act (Activatio  (None, 12, 12, 728)  0          ['block4_sepconv1_bn[0]
[0]']
 n)

 block4_sepconv2 (SeparableConv  (None, 12, 12, 728)  536536     ['block4_sepconv2_act
[0][0]']
 2D)

 block4_sepconv2_bn (BatchNorma  (None, 12, 12, 728)  2912       ['block4_sepconv2[0]
[0]']
 lization)
 conv2d_18 (Conv2D)             (None, 6, 6, 728)    186368      ['add_49[0][0]']

 block4_pool (MaxPooling2D)     (None, 6, 6, 728)    0           ['block4_sepconv2_bn[0]
[0]']

 batch_normalization_72 (BatchN  (None, 6, 6, 728)   2912        ['conv2d_18[0][0]']
 ormalization)

 add_50 (Add)                   (None, 6, 6, 728)    0           ['block4_pool[0][0]',
                                                                  'batch_normalization_7
2[0][0]']

 block5_sepconv1_act (Activatio  (None, 6, 6, 728)   0           ['add_50[0][0]']
 n)

 block5_sepconv1 (SeparableConv  (None, 6, 6, 728)   536536      ['block5_sepconv1_act
[0][0]']
 2D)

 block5_sepconv1_bn (BatchNorma  (None, 6, 6, 728)   2912        ['block5_sepconv1[0]
[0]']
 lization)

 block5_sepconv2_act (Activatio  (None, 6, 6, 728)   0           ['block5_sepconv1_bn[0]
[0]']
 n)

 block5_sepconv2 (SeparableConv  (None, 6, 6, 728)   536536      ['block5_sepconv2_act
[0][0]']
 2D)

 block5_sepconv2_bn (BatchNorma  (None, 6, 6, 728)   2912        ['block5_sepconv2[0]
[0]']
 lization)

 block5_sepconv3_act (Activatio  (None, 6, 6, 728)   0           ['block5_sepconv2_bn[0]
[0]']
 n)

 block5_sepconv3 (SeparableConv  (None, 6, 6, 728)   536536      ['block5_sepconv3_act
[0][0]']
 2D)

 block5_sepconv3_bn (BatchNorma  (None, 6, 6, 728)   2912        ['block5_sepconv3[0]
[0]']
 lization)

 add_51 (Add)                   (None, 6, 6, 728)    0           ['block5_sepconv3_bn[0]
[0]',

                                                                  'add_50[0][0]']

 block6_sepconv1_act (Activatio  (None, 6, 6, 728)   0           ['add_51[0][0]']
 n)

 block6_sepconv1 (SeparableConv  (None, 6, 6, 728)   536536      ['block6_sepconv1_act
[0][0]']
 2D)

 block6_sepconv1_bn (BatchNorma  (None, 6, 6, 728)   2912        ['block6_sepconv1[0]
[0]']
 lization)

 block6_sepconv2_act (Activatio  (None, 6, 6, 728)   0           ['block6_sepconv1_bn[0]
[0]']
 n)

 block6_sepconv2 (SeparableConv  (None, 6, 6, 728)   536536      ['block6_sepconv2_act
[0][0]']
 2D)

 block6_sepconv2_bn (BatchNorma  (None, 6, 6, 728)   2912        ['block6_sepconv2[0]
[0]']
 lization)

 block6_sepconv3_act (Activatio  (None, 6, 6, 728)   0           ['block6_sepconv2_bn[0]
[0]']
 n)

 block6_sepconv3 (SeparableConv  (None, 6, 6, 728)   536536      ['block6_sepconv3_act
[0][0]']
 2D)

 block6_sepconv3_bn (BatchNorma  (None, 6, 6, 728)   2912        ['block6_sepconv3[0]
[0]']
 lization)

 add_52 (Add)                   (None, 6, 6, 728)    0           ['block6_sepconv3_bn[0]
[0]',

                                                                  'add_51[0][0]']

 block7_sepconv1_act (Activatio  (None, 6, 6, 728)   0           ['add_52[0][0]']
 n)

 block7_sepconv1 (SeparableConv  (None, 6, 6, 728)   536536      ['block7_sepconv1_act
[0][0]']
 2D)

 block7_sepconv1_bn (BatchNorma  (None, 6, 6, 728)   2912        ['block7_sepconv1[0]
[0]']
 lization)

 block7_sepconv2_act (Activatio  (None, 6, 6, 728)   0           ['block7_sepconv1_bn[0]
[0]']
 n)

 block7_sepconv2 (SeparableConv  (None, 6, 6, 728)   536536      ['block7_sepconv2_act
[0][0]']
 2D)

 block7_sepconv2_bn (BatchNorma  (None, 6, 6, 728)   2912        ['block7_sepconv2[0]
[0]']
 lization)

 block7_sepconv3_act (Activatio  (None, 6, 6, 728)   0           ['block7_sepconv2_bn[0]
[0]']
 n)

 block7_sepconv3 (SeparableConv  (None, 6, 6, 728)   536536      ['block7_sepconv3_act
[0][0]']
```

```
 2D)

 block7_sepconv3_bn (BatchNorma  (None, 6, 6, 728)   2912        ['block7_sepconv3[0]
[0]']
 lization)

 add_53 (Add)                   (None, 6, 6, 728)    0           ['block7_sepconv3_bn[0]
[0]',
                                                                  'add_52[0][0]']

 block8_sepconv1_act (Activatio  (None, 6, 6, 728)   0           ['add_53[0][0]']
 n)

 block8_sepconv1 (SeparableConv  (None, 6, 6, 728)   536536      ['block8_sepconv1_act
[0][0]']
 2D)

 block8_sepconv1_bn (BatchNorma  (None, 6, 6, 728)   2912        ['block8_sepconv1[0]
[0]']
 lization)

 block8_sepconv2_act (Activatio  (None, 6, 6, 728)   0           ['block8_sepconv1_bn[0]
[0]']
 n)

 block8_sepconv2 (SeparableConv  (None, 6, 6, 728)   536536      ['block8_sepconv2_act
[0][0]']
 2D)

 block8_sepconv2_bn (BatchNorma  (None, 6, 6, 728)   2912        ['block8_sepconv2[0]
[0]']
 lization)

 block8_sepconv3_act (Activatio  (None, 6, 6, 728)   0           ['block8_sepconv2_bn[0]
[0]']
 n)

 block8_sepconv3 (SeparableConv  (None, 6, 6, 728)   536536      ['block8_sepconv3_act
[0][0]']
 2D)

 block8_sepconv3_bn (BatchNorma  (None, 6, 6, 728)   2912        ['block8_sepconv3[0]
[0]']
 lization)

 add_54 (Add)                   (None, 6, 6, 728)    0           ['block8_sepconv3_bn[0]
[0]',
                                                                  'add_53[0][0]']

 block9_sepconv1_act (Activatio  (None, 6, 6, 728)   0           ['add_54[0][0]']
 n)

 block9_sepconv1 (SeparableConv  (None, 6, 6, 728)   536536      ['block9_sepconv1_act
[0][0]']
 2D)

 block9_sepconv1_bn (BatchNorma  (None, 6, 6, 728)   2912        ['block9_sepconv1[0]
[0]']
 lization)
 block9_sepconv2_act (Activatio  (None, 6, 6, 728)   0           ['block9_sepconv1_bn[0]
[0]']
 n)

 block9_sepconv2 (SeparableConv  (None, 6, 6, 728)   536536      ['block9_sepconv2_act
[0][0]']
 2D)

 block9_sepconv2_bn (BatchNorma  (None, 6, 6, 728)   2912        ['block9_sepconv2[0]
[0]']
 lization)

 block9_sepconv3_act (Activatio  (None, 6, 6, 728)   0           ['block9_sepconv2_bn[0]
[0]']
 n)

 block9_sepconv3 (SeparableConv  (None, 6, 6, 728)   536536      ['block9_sepconv3_act
[0][0]']
 2D)

 block9_sepconv3_bn (BatchNorma  (None, 6, 6, 728)   2912        ['block9_sepconv3[0]
[0]']
 lization)

 add_55 (Add)                   (None, 6, 6, 728)    0           ['block9_sepconv3_bn[0]
[0]',
                                                                  'add_54[0][0]']

 block10_sepconv1_act (Activati  (None, 6, 6, 728)   0           ['add_55[0][0]']
 on)

 block10_sepconv1 (SeparableCon  (None, 6, 6, 728)   536536      ['block10_sepconv1_act
[0][0]']
 v2D)

 block10_sepconv1_bn (BatchNorm  (None, 6, 6, 728)   2912        ['block10_sepconv1[0]
[0]']
 alization)

 block10_sepconv2_act (Activati  (None, 6, 6, 728)   0           ['block10_sepconv1_bn
[0][0]']
 on)

 block10_sepconv2 (SeparableCon  (None, 6, 6, 728)   536536      ['block10_sepconv2_act
[0][0]']
 v2D)

 block10_sepconv2_bn (BatchNorm  (None, 6, 6, 728)   2912        ['block10_sepconv2[0]
[0]']
 alization)

 block10_sepconv3_act (Activati  (None, 6, 6, 728)   0           ['block10_sepconv2_bn
[0][0]']
 on)

 block10_sepconv3 (SeparableCon  (None, 6, 6, 728)   536536      ['block10_sepconv3_act
[0][0]']
 v2D)

 block10_sepconv3_bn (BatchNorm  (None, 6, 6, 728)   2912        ['block10_sepconv3[0]
[0]']
 alization)

 add_56 (Add)                   (None, 6, 6, 728)    0           ['block10_sepconv3_bn
[0][0]',
                                                                  'add_55[0][0]']

 block11_sepconv1_act (Activati  (None, 6, 6, 728)   0           ['add_56[0][0]']
 on)

 block11_sepconv1 (SeparableCon  (None, 6, 6, 728)   536536      ['block11_sepconv1_act
[0][0]']
 v2D)

 block11_sepconv1_bn (BatchNorm  (None, 6, 6, 728)   2912        ['block11_sepconv1[0]
[0]']
 alization)

 block11_sepconv2_act (Activati  (None, 6, 6, 728)   0           ['block11_sepconv1_bn
[0][0]']
 on)

 block11_sepconv2 (SeparableCon  (None, 6, 6, 728)   536536      ['block11_sepconv2_act
[0][0]']
 v2D)

 block11_sepconv2_bn (BatchNorm  (None, 6, 6, 728)   2912        ['block11_sepconv2[0]
[0]']
 alization)

 block11_sepconv3_act (Activati  (None, 6, 6, 728)   0           ['block11_sepconv2_bn
[0][0]']
 on)

 block11_sepconv3 (SeparableCon  (None, 6, 6, 728)   536536      ['block11_sepconv3_act
[0][0]']
 v2D)

 block11_sepconv3_bn (BatchNorm  (None, 6, 6, 728)   2912        ['block11_sepconv3[0]
[0]']
 alization)

 add_57 (Add)                   (None, 6, 6, 728)    0           ['block11_sepconv3_bn
[0][0]',
                                                                  'add_56[0][0]']

 block12_sepconv1_act (Activati  (None, 6, 6, 728)   0           ['add_57[0][0]']
 on)

 block12_sepconv1 (SeparableCon  (None, 6, 6, 728)   536536      ['block12_sepconv1_act
[0][0]']
 v2D)

 block12_sepconv1_bn (BatchNorm  (None, 6, 6, 728)   2912        ['block12_sepconv1[0]
[0]']
 alization)

 block12_sepconv2_act (Activati  (None, 6, 6, 728)   0           ['block12_sepconv1_bn
[0][0]']
 on)
 block12_sepconv2 (SeparableCon  (None, 6, 6, 728)   536536      ['block12_sepconv2_act
[0][0]']
 v2D)

 block12_sepconv2_bn (BatchNorm  (None, 6, 6, 728)   2912        ['block12_sepconv2[0]
[0]']
 alization)

 block12_sepconv3_act (Activati  (None, 6, 6, 728)   0           ['block12_sepconv2_bn
[0][0]']
 on)

 block12_sepconv3 (SeparableCon  (None, 6, 6, 728)   536536      ['block12_sepconv3_act
[0][0]']
 v2D)

 block12_sepconv3_bn (BatchNorm  (None, 6, 6, 728)   2912        ['block12_sepconv3[0]
[0]']
 alization)

 add_58 (Add)                   (None, 6, 6, 728)    0           ['block12_sepconv3_bn
[0][0]',
                                                                  'add_57[0][0]']

 block13_sepconv1_act (Activati  (None, 6, 6, 728)   0           ['add_58[0][0]']
 on)

 block13_sepconv1 (SeparableCon  (None, 6, 6, 728)   536536      ['block13_sepconv1_act
[0][0]']
 v2D)

 block13_sepconv1_bn (BatchNorm  (None, 6, 6, 728)   2912        ['block13_sepconv1[0]
[0]']
 alization)

 block13_sepconv2_act (Activati  (None, 6, 6, 728)   0           ['block13_sepconv1_bn
[0][0]']
 on)

 block13_sepconv2 (SeparableCon  (None, 6, 6, 1024)  752024      ['block13_sepconv2_act
[0][0]']
 v2D)

 block13_sepconv2_bn (BatchNorm  (None, 6, 6, 1024)  4096        ['block13_sepconv2[0]
[0]']
 alization)

 conv2d_19 (Conv2D)             (None, 3, 3, 1024)  745472      ['add_58[0][0]']

 block13_pool (MaxPooling2D)    (None, 3, 3, 1024)  0           ['block13_sepconv2_bn
[0][0]']

 batch_normalization_73 (BatchN  (None, 3, 3, 1024)  4096        ['conv2d_19[0][0]']
 ormalization)

 add_59 (Add)                   (None, 3, 3, 1024)  0           ['block13_pool[0][0]',
                                                                  'batch_normalization_7
3[0][0]']
```

```
 block14_sepconv1 (SeparableCon  (None, 3, 3, 1536)  1582080     ['add_59[0][0]']
 v2D)

 block14_sepconv1_bn (BatchNorm  (None, 3, 3, 1536)  6144        ['block14_sepconv1[0]
[0]']
 alization)

 block14_sepconv1_act (Activati  (None, 3, 3, 1536)  0           ['block14_sepconv1_bn
[0][0]']
 on)

 block14_sepconv2 (SeparableCon  (None, 3, 3, 2048)  3159552     ['block14_sepconv1_act
[0][0]']
 v2D)

 block14_sepconv2_bn (BatchNorm  (None, 3, 3, 2048)  8192        ['block14_sepconv2[0]
[0]']
 alization)

 block14_sepconv2_act (Activati  (None, 3, 3, 2048)  0           ['block14_sepconv2_bn
[0][0]']
 on)

 global_average_pooling2d_9 (Gl  (None, 2048)        0           ['block14_sepconv2_act
[0][0]']
 obalAveragePooling2D)

 dense_20 (Dense)               (None, 200)          409800      ['global_average_poolin
g2d_9[0][0

                                                                 ]']

 dense_21 (Dense)               (None, 100)          20100       ['dense_20[0][0]']

 dense_22 (Dense)               (None, 50)           5050        ['dense_21[0][0]']

 dense_23 (Dense)               (None, 10)           510         ['dense_22[0][0]']

==========================================================================================
==========
Total params: 21,296,940
Trainable params: 21,242,412
Non-trainable params: 54,528
__________________________________________________________________________________________
__________
```

In [106…

```
hobj = model.fit(x_train, labels_train_ohe, validation_data=(x_val, labels_val_ohe), ep
         shuffle=True, callbacks=[perf_eval_cb, early_stop_cb], verbose=1)
```

```
Epoch 1/10
377/377 [==============================] - 791s 2s/step - loss: 0.9091 - accuracy: 0.706
8 - val_loss: 1.6019 - val_accuracy: 0.6823
Epoch 2/10
377/377 [==============================] - 765s 2s/step - loss: 0.3833 - accuracy: 0.886
8 - val_loss: 0.6148 - val_accuracy: 0.8199
Epoch 3/10
377/377 [==============================] - 778s 2s/step - loss: 0.2659 - accuracy: 0.919
6 - val_loss: 0.7714 - val_accuracy: 0.8643
Epoch 4/10
```

```
377/377 [==============================] - 783s 2s/step - loss: 0.2030 - accuracy: 0.939
5 - val_loss: 0.7210 - val_accuracy: 0.8584
Epoch 5/10
377/377 [==============================] - 811s 2s/step - loss: 0.1621 - accuracy: 0.950
4 - val_loss: 0.4860 - val_accuracy: 0.8663
Epoch 6/10
377/377 [==============================] - 799s 2s/step - loss: 0.1367 - accuracy: 0.958
1 - val_loss: 0.3396 - val_accuracy: 0.9008
Epoch 7/10
377/377 [==============================] - 785s 2s/step - loss: 0.1059 - accuracy: 0.968
8 - val_loss: 0.3019 - val_accuracy: 0.9194
Epoch 8/10
377/377 [==============================] - 751s 2s/step - loss: 0.1004 - accuracy: 0.969
8 - val_loss: 0.2482 - val_accuracy: 0.9271
Epoch 9/10
377/377 [==============================] - 763s 2s/step - loss: 0.0850 - accuracy: 0.974
9 - val_loss: 0.2475 - val_accuracy: 0.9392
Epoch 10/10
377/377 [==============================] - 756s 2s/step - loss: 0.0812 - accuracy: 0.976
1 - val_loss: 0.2512 - val_accuracy: 0.9382

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_11052/2076790798.py in <module>
      2           shuffle=True, callbacks=[perf_eval_cb, early_stop_cb], verbose=1)
      3 eff_epochs = len(hobj.history['loss'])
----> 4 eval_data = (perf_data[0:eff_epochs,:], dataset)

NameError: name 'dataset' is not defined
```

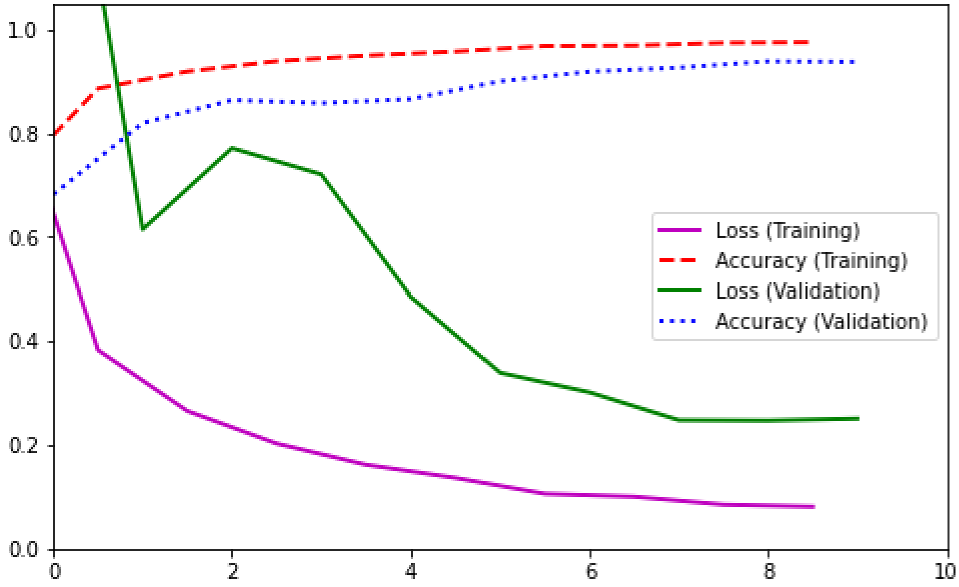
In [106…

```
eff_epochs = len(hobj.history['loss'])
eval_data = (perf_data[0:eff_epochs,:], cnn_dataset)
```

In [106…

```
evaluate_model("test", model, eval_data)
```

```
[test] Training Accuracy: 97.784%, Validation Accuracy: 93.824%
```



```
126/126 [==============================] - 42s 242ms/step - loss: 0.2857 - accuracy: 0.9
335
[test]  Test loss: 0.28567; test accuracy: 93.353%
[test]  Test cross entropy loss: 0.28447
```

FULL TRAINING

In [107…

```
cnn_input.shape
```

Out[107…

```
(26880, 100, 100, 3)
```

In [107…

```
labels_new_ohe = OneHotEncoder().fit_transform(labels_new.reshape(-1,1)).toarray()
```

In [107…

```
labels_new_ohe.shape
```

Out[107…

```
(26880, 10)
```

In [109…

```
hobj = model.fit(cnn_input, labels_new_ohe, epochs=8, batch_size=100,
         shuffle=True, verbose=1)
```

```
Epoch 1/8
269/269 [==============================] - 967s 4s/step - loss: 0.2659 - accuracy: 0.921
7
Epoch 2/8
269/269 [==============================] - 960s 4s/step - loss: 0.1679 - accuracy: 0.948
0
Epoch 3/8
269/269 [==============================] - 975s 4s/step - loss: 0.1285 - accuracy: 0.960
7
Epoch 4/8
269/269 [==============================] - 964s 4s/step - loss: 0.0978 - accuracy: 0.971
8
Epoch 5/8
269/269 [==============================] - 1004s 4s/step - loss: 0.0806 - accuracy: 0.97
66
Epoch 6/8
  1/269 [..............................] - ETA: 16:14 - loss: 0.0712 - accuracy: 0.9600
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_11052/1563957083.py in <module>
----> 1 hobj = model.fit(cnn_input, labels_new_ohe, epochs=8, batch_size=100,
      2          shuffle=True, verbose=1)

~\anaconda3\lib\site-packages\keras\utils\traceback_utils.py in error_handler(*args, **k
wargs)
     62     filtered_tb = None
     63     try:
---> 64       return fn(*args, **kwargs)
     65     except Exception as e:  # pylint: disable=broad-except
     66       filtered_tb = _process_traceback_frames(e.__traceback__)

~\anaconda3\lib\site-packages\keras\engine\training.py in fit(self, x, y, batch_size, ep
ochs, verbose, callbacks, validation_split, validation_data, shuffle, class_weight, samp
le_weight, initial_epoch, steps_per_epoch, validation_steps, validation_batch_size, vali
dation_freq, max_queue_size, workers, use_multiprocessing)
   1382                 _r=1):
   1383               callbacks.on_train_batch_begin(step)
```

```
-> 1384                 tmp_logs = self.train_function(iterator)
   1385                 if data_handler.should_sync:
   1386                   context.async_wait()

~\anaconda3\lib\site-packages\tensorflow\python\util\traceback_utils.py in error_handler
(*args, **kwargs)
    148     filtered_tb = None
    149     try:
--> 150       return fn(*args, **kwargs)
    151     except Exception as e:
    152       filtered_tb = _process_traceback_frames(e.__traceback__)

~\anaconda3\lib\site-packages\tensorflow\python\eager\def_function.py in __call__(self,
 *args, **kwds)
    913 
    914       with OptionalXlaContext(self._jit_compile):
--> 915         result = self._call(*args, **kwds)
    916 
    917       new_tracing_count = self.experimental_get_tracing_count()

~\anaconda3\lib\site-packages\tensorflow\python\eager\def_function.py in _call(self, *ar
gs, **kwds)
    945       # In this case we have created variables on the first call, so we run the
    946       # defunned version which is guaranteed to never create variables.
--> 947       return self._stateless_fn(*args, **kwds)  # pylint: disable=not-callable
    948     elif self._stateful_fn is not None:
    949       # Release the lock early so that multiple threads can perform the call

~\anaconda3\lib\site-packages\tensorflow\python\eager\function.py in __call__(self, *arg
s, **kwargs)
   2954       (graph_function,
   2955        filtered_flat_args) = self._maybe_define_function(args, kwargs)
-> 2956     return graph_function._call_flat(
   2957         filtered_flat_args, captured_inputs=graph_function.captured_inputs)  # p
ylint: disable=protected-access
   2958 

~\anaconda3\lib\site-packages\tensorflow\python\eager\function.py in _call_flat(self, ar
gs, captured_inputs, cancellation_manager)
   1851         and executing_eagerly):
   1852       # No tape is watching; skip to running the function.
-> 1853       return self._build_call_outputs(self._inference_function.call(
   1854           ctx, args, cancellation_manager=cancellation_manager))
   1855     forward_backward = self._select_forward_and_backward_functions(

~\anaconda3\lib\site-packages\tensorflow\python\eager\function.py in call(self, ctx, arg
s, cancellation_manager)
    497       with _InterpolateFunctionError(self):
    498         if cancellation_manager is None:
--> 499           outputs = execute.execute(
    500               str(self.signature.name),
    501               num_outputs=self._num_outputs,

~\anaconda3\lib\site-packages\tensorflow\python\eager\execute.py in quick_execute(op_nam
e, num_outputs, inputs, attrs, ctx, name)
     52   try:
     53     ctx.ensure_initialized()
---> 54     tensors = pywrap_tfe.TFE_Py_Execute(ctx._handle, device_name, op_name,
     55                                         inputs, attrs, num_outputs)
     56   except core._NotOkStatusException as e:
```

```
KeyboardInterrupt:
```

In [109…

```
model.save('final_trained_model')
```

```
INFO:tensorflow:Assets written to: final_trained_model\assets
```

TEST SECTION

In [ ]:

```
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
import tensorflow.keras as keras
import cv2
from sklearn.preprocessing import MinMaxScaler, OneHotEncoder
from sklearn.model_selection import train_test_split
import time
```

In [ ]:

```
test_data = np.load("put_path_here.npy")
test_labels = np.load("put_path_here.npy")

weights = "final_trained_model"
model = keras.models.load_model(weights)
```

Functions used to preprocess data

In [ ]:

```
def show(im):
    plt.imshow(im, cmap='gray')
    plt.show()

def invert(im):
    return (im*-1)+255

def min_max_scale(im):
    im = MinMaxScaler().fit_transform(im.ravel().reshape(-1,1))
    return im.reshape(300,300)

def brighten(im):
    m = np.max(im)-.3
    im[im >= m] += 0.3
    im[im < m] -= 0.3
    return im

def blur(im, kernel):
    im = im.astype("uint8")
    im = cv2.medianBlur(im, kernel)
    return im

def morph_close(im, kernel, i):
    return cv2.morphologyEx(im, cv2.MORPH_CLOSE, kernel, iterations=i)

def morph_open(im, kernel, i):
    return cv2.morphologyEx(im, cv2.MORPH_OPEN, kernel, iterations=i)
```

```
def morph_dilate(im, kernel, i):
    return cv2.dilate(im, kernel, iterations=i)

def morph_erode(im, kernel, i):
    return cv2.erode(im, kernel, iterations=i)

def transform(im):
    dest = np.float32([[0,0],[300,0],[0,300],[300,300]])
    source = np.float32([[25,25],[275,25],[25,275],[275,275]])
    res = cv2.getPerspectiveTransform(source,dest)
    return cv2.warpPerspective(im, res, (300,300))

def preprocess(ims):
    r = []
    for im in ims.T:
        im = im.reshape((300,300))
        im = transform(im)
        im = invert(im)
        im = blur(im, 5)
        im = morph_erode(im, (3,3), 5)
        im = morph_dilate(im, (3,3), 5)
        im = morph_close(im, (3,3), 5)
        im = morph_open(im, (3,3), 5)
        im = min_max_scale(im)
        im = brighten(im)
        im = min_max_scale(im)
        im = cv2.resize(im, (100,100))
        im = np.stack((im,)*3, axis=-1)
        r.append(im)
    return np.array(r)
```

In [ ]:

```
test_data = preprocess(test_data)
test_labels_ohe = OneHotEncoder().fit_transform(test_labels.reshape(-1,1)).toarray()
```

In [ ]:

```
model = keras.models.load_model("put_model_file_here")
```

In [ ]:

```
test_preds = model.predict(test_data)

accuracy_obj = keras.metrics.CategoricalAccuracy()
accuracy_obj.update_state(test_labels_ohe, test_preds)
test_acc = accuracy_obj.result()

test_loss, _ = model.evaluate(test_data, test_labels_ohe)

cat_loss_obj = tf.keras.losses.CategoricalCrossentropy()
test_ce_loss = cat_loss_obj(test_labels_ohe, test_preds)


print('[{}]  Test loss: {:.5f}; test accuracy: {:.3f}%'.format(name, test_loss, 100*tes
print('[{}]  Test cross entropy loss: {:.5f}'.format(name, test_ce_loss))
```