Bald_Test_n6

May 21, 2023

1 Bald BCNN (Active Learning)

2 IndianPines

3 Date: 5 2023

[ ]:
```
from sklearn.decomposition import PCA

from __future__ import print_function
from keras.datasets import mnist
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Activation, Flatten
from keras.layers.convolutional import Convolution2D, MaxPooling2D
from keras.optimizers import SGD, Adadelta, Adagrad, Adam
from keras.utils import np_utils, generic_utils
from six.moves import range
import numpy as np
import scipy as sp
from keras import backend as K
import random
import scipy.io
import matplotlib.pyplot as plt
from keras.regularizers import l2
```

[ ]:
```

```

DATA

[ ]:
```
## VARIABLES

test_ratio = 0.3
test_val_ratio=0.7

train_ratio = 1-test_ratio
#train_val_ratio = 0.8
```

```
windowSize = 7  #  25
dimReduction = 80 # dimReduction

drop = 0.4
```

[ ]:
```
# Read data
from sklearn.model_selection import train_test_split

from scipy.io import loadmat

def read_HSI():
  X = loadmat('Indian_pines.mat')['indian_pines']
  y = loadmat('Indian_pines_gt.mat')['indian_pines_gt']
  print(f"X shape: {X.shape}\ny shape: {y.shape}")
  return X, y

X, y = read_HSI()


# PCA
def applyPCA(X, numComponents): # numComponents=64
    newX = np.reshape(X, (-1, X.shape[2]))
    print(newX.shape)
    pca = PCA(n_components=numComponents, whiten=True)
    newX = pca.fit_transform(newX)
    newX = np.reshape(newX, (X.shape[0],X.shape[1], numComponents))
    return newX, pca, pca.explained_variance_ratio_

# pading With Zeros
def padWithZeros(X, margin=2):
    newX = np.zeros((X.shape[0] + 2 * margin, X.shape[1] + 2* margin, X.
 ↪shape[2]),dtype="float16")
    x_offset = margin
    y_offset = margin
    newX[x_offset:X.shape[0] + x_offset, y_offset:X.shape[1] + y_offset, :] = X
    return newX

# Split Data

def splitTrainTestSet(X, y, testRatio, randomState=345):
    X_train, X_test, y_train, y_test = train_test_split(X, y,␣
 ↪test_size=testRatio, random_state=randomState,stratify=y)
    return X_train, X_test, y_train, y_test
```

```
X shape: (145, 145, 220)
y shape: (145, 145)
```

[ ]:
```
# Split the hyperspectral image into patches of size windowSize-by-windowSize
 ↪pixels
def Patches_Creating(X, y, windowSize, removeZeroLabels = True):  #
 ↪windowSize=15, 25
    margin = int((windowSize - 1) / 2)
    zeroPaddedX = padWithZeros(X, margin=margin)
    # split patches
    patchesData = np.zeros((X.shape[0] * X.shape[1], windowSize, windowSize, X.
 ↪shape[2]),dtype="float16")
    patchesLabels = np.zeros((X.shape[0] * X.shape[1]),dtype="float16")
    patchIndex = 0
    for r in range(margin, zeroPaddedX.shape[0] - margin):
        for c in range(margin, zeroPaddedX.shape[1] - margin):
            patch = zeroPaddedX[r - margin:r + margin + 1, c - margin:c +
 ↪margin + 1]
            patchesData[patchIndex, :, :, :] = patch
            patchesLabels[patchIndex] = y[r-margin, c-margin]
            patchIndex = patchIndex + 1
    if removeZeroLabels:
        patchesData = patchesData[patchesLabels>0,:,:,:]
        patchesLabels = patchesLabels[patchesLabels>0]
        patchesLabels -= 1
    return patchesData, patchesLabels
# channel_wise_shift
def channel_wise_shift(X,numComponents):
    X_copy = np.zeros((X.shape[0] , X.shape[1], X.shape[2]))
    half = int(numComponents/2)
    for i in range(0,half-1):
        X_copy[:,:,i] = X[:,:,(half-i)*2-1]
    for i in range(half,numComponents):
        X_copy[:,:,i] = X[:,:,(i-half)*2]
    X = X_copy
    return X
```

[ ]:
```

```

[ ]:
```
#(X_train_All, y_train_All), (X_test, y_test) = mnist.load_data()

Experiments = 3

batch_size = 128
nb_classes = 16

#use a large number of epochs
nb_epoch = 20

# input image dimensions
```

```
img_rows, img_cols = 28, 28
# number of convolutional filters to use
nb_filters = 32
# size of pooling area for max pooling
nb_pool = 1

# Original Kernel Size used in all other experiments: nb_conv = 3

# convolution kernel size
nb_conv = 1

score=0
all_accuracy = 0
acquisition_iterations = 98

#use a large number of dropout iterations
dropout_iterations = 50

Queries = 16


Experiments_All_Accuracy = np.zeros(shape=(acquisition_iterations+1))
```

[ ]:
```
X0, y0 = read_HSI()
#X=X0
#y=y0

InputShape=(windowSize, windowSize, dimReduction)

#X, y = loadData(dataset) channel_wise_shift
X1,pca,ratio = applyPCA(X0,numComponents=dimReduction)
X2_shifted = channel_wise_shift(X1,dimReduction) # channel-wise shift
#X2=X1

#print(f"X0 shape: {X0.shape}\ny0 shape: {y0.shape}")
#print(f"X1 shape: {X1.shape}\nX2 shape: {X2.shape}")

X3, y3 = Patches_Creating(X2_shifted, y0, windowSize=windowSize)  # 5 for PAvia␣
↪Center

Xtrain, Xtest, ytrain, ytest = splitTrainTestSet(X3, y3, test_ratio)


Xtest0=Xtest
ytest0=ytest

Xtrain = Xtrain.reshape(-1, windowSize,windowSize, dimReduction)
```

```
#ytrain = np_utils.to_categorical(ytrain)

Xvalid, Xtest, yvalid, ytest = splitTrainTestSet(Xtest, ytest, test_val_ratio)

#ytest = np_utils.to_categorical(ytest)


Xvalid = Xvalid.reshape(-1, windowSize,windowSize, dimReduction)
#yvalid = np_utils.to_categorical(yvalid)

print(f"Xtrain shape: {Xtrain.shape}\nytrain shape : {ytrain.shape}")

# Xtrain ytrain   Xtest  ytest    Xvalid yvalid
```

```
X shape: (145, 145, 220)
y shape: (145, 145)
(21025, 220)
Xtrain shape: (7174, 7, 7, 80)
ytrain shape : (7174, 16)
```

[ ]:
```
# Testing Model_ NO2
# Basian deep neural network (BCNN)
import tensorflow as tf
import tensorflow_probability as tfp
from tensorflow.keras.layers import Dense, Conv2D, MaxPooling2D, Flatten,␣
 ↪Dropout
from tensorflow.keras.layers import Input, Dense, Conv1D, MaxPooling1D,␣
 ↪Dropout, Flatten
from tensorflow.keras.models import Sequential
from tensorflow.keras.utils import to_categorical

tfd = tfp.distributions
tfpl = tfp.layers
# Testing Model_ NO2
# Basian deep neural network (BCNN)

divergence_fn = lambda q,p,_:tfd.kl_divergence(q,p)/len(Xtrain)   #3457

# BCNN model
#
model_bayes = Sequential([
    # Statistical 2D conv
    tfpl.Convolution2DReparameterization(input_shape=InputShape, filters=32,␣
 ↪kernel_size=3, activation='relu',
```

```
                                        kernel_prior_fn = tfpl.
↪default_multivariate_normal_fn,
                                        kernel_posterior_fn=tfpl.
↪default_mean_field_normal_fn(is_singular=False),
                                        kernel_divergence_fn = divergence_fn,
                                        bias_prior_fn = tfpl.
↪default_multivariate_normal_fn,
                                        bias_posterior_fn=tfpl.
↪default_mean_field_normal_fn(is_singular=False),
                                        bias_divergence_fn = divergence_fn),
    MaxPooling2D(2,1),
    Conv2D(32, (3,3), activation='relu'),
    MaxPooling2D(2,1),

    Flatten(),
    Dense(512, activation='relu'),
    Dropout(0.2),
    # Statistical Dense-
    tfpl.DenseReparameterization(units=tfpl.OneHotCategorical.params_size(16),␣
↪activation=None,
                                 kernel_prior_fn = tfpl.
↪default_multivariate_normal_fn,
                                 kernel_posterior_fn=tfpl.
↪default_mean_field_normal_fn(is_singular=False),
                                 kernel_divergence_fn = divergence_fn,
                                 bias_prior_fn = tfpl.
↪default_multivariate_normal_fn,
                                 bias_posterior_fn=tfpl.
↪default_mean_field_normal_fn(is_singular=False),
                                 bias_divergence_fn = divergence_fn
                               ),
    # output-
    tfpl.OneHotCategorical(16)

])
model_bayes.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_reparameterization (  (None, 5, 5, 32)         46144
 Conv2DReparameterization)

 max_pooling2d (MaxPooling2D  (None, 4, 4, 32)         0
```

```
c:\Users\kifah\AppData\Local\Programs\Python\Python310\lib\site-
```

```
packages\tensorflow_probability\python\layers\util.py:95: UserWarning:
`layer.add_variable` is deprecated and will be removed in a future version.
Please use the `layer.add_weight()` method instead.
  loc = add_variable_fn(
c:\Users\kifah\AppData\Local\Programs\Python\Python310\lib\site-
packages\tensorflow_probability\python\layers\util.py:105: UserWarning:
`layer.add_variable` is deprecated and will be removed in a future version.
Please use the `layer.add_weight()` method instead.
  untransformed_scale = add_variable_fn(
 )

 conv2d (Conv2D)             (None, 2, 2, 32)          9248

 max_pooling2d_1 (MaxPooling  (None, 1, 1, 32)         0
 2D)

 flatten (Flatten)           (None, 32)                0

 dense (Dense)               (None, 512)               16896

 dropout (Dropout)           (None, 512)               0

 conv2d (Conv2D)             (None, 2, 2, 32)          9248

 max_pooling2d_1 (MaxPooling  (None, 1, 1, 32)         0
 2D)

 flatten (Flatten)           (None, 32)                0

 dense (Dense)               (None, 512)               16896

 dropout (Dropout)           (None, 512)               0

 dense_reparameterization (D  (None, 16)               16416
 enseReparameterization)

 one_hot_categorical (OneHot  ((None, 16),             0
 Categorical)                 (None, 16))

=================================================================
Total params: 88,704
Trainable params: 88,704
Non-trainable params: 0
_________________________________________________________________
```

[ ]:
```
#Testing Model_ NO1
from tensorflow.keras.optimizers import Adam
```

```
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint,␣
 ↪TensorBoard

def negative_log_likelihood(y_true, y_pred):
    return -y_pred.log_prob(y_true)

# Testing Model_ N02
# Comiple

model_bayes.compile(loss = negative_log_likelihood,
              optimizer = Adam(learning_rate=0.001), #0.005
              metrics = ['accuracy'],
              experimental_run_tf_function = False)
```

[ ]:
```
# Bald
for e in range(3):  # Experiments

        print('Experiment Number ', e)

        # the data, shuffled and split between tran and test sets

        #X_train_All = X_train_All.reshape(X_train_All.shape[0], 1, img_rows,␣
 ↪img_cols)
        #X_test = X_test.reshape(X_test.shape[0], 1, img_rows, img_cols)

        X_train_All=Xtrain
        y_train_All=ytrain
        y_test=ytest
        X_test=Xtest

        ## Xtrain ytrain   Xtest  ytest    Xvalid yvalid


        #X_valid = X_train_All[6000:7174, :, :, :]
        #y_valid = y_train_All[6000:7174]
        X_valid=Xvalid
        y_valid=yvalid


        X_Pool = X_train_All[5000:7174, :, :, :]
        y_Pool = y_train_All[5000:7174]


        X_train_All = X_train_All[0:5000, :, :, :]
        y_train_All = y_train_All[0:5000]
```

```
ssz=20  # Sample for every class


idx_0 = np.array( np.where(y_train_All==0)  ).T
idx_0 = idx_0[0:ssz,0]
X_0 = X_train_All[idx_0, :, :, :]
y_0 = y_train_All[idx_0]

idx_1 = np.array( np.where(y_train_All==1)  ).T
idx_1 = idx_1[0:ssz,0]
X_1 = X_train_All[idx_1, :, :, :]
y_1 = y_train_All[idx_1]

idx_2 = np.array( np.where(y_train_All==2)  ).T
idx_2 = idx_2[0:ssz,0]
X_2 = X_train_All[idx_2, :, :, :]
y_2 = y_train_All[idx_2]

idx_3 = np.array( np.where(y_train_All==3)  ).T
idx_3 = idx_3[0:ssz,0]
X_3 = X_train_All[idx_3, :, :, :]
y_3 = y_train_All[idx_3]

idx_4 = np.array( np.where(y_train_All==4)  ).T
idx_4 = idx_4[0:ssz,0]
X_4 = X_train_All[idx_4, :, :, :]
y_4 = y_train_All[idx_4]

idx_5 = np.array( np.where(y_train_All==5)  ).T
idx_5 = idx_5[0:ssz,0]
X_5 = X_train_All[idx_5, :, :, :]
y_5 = y_train_All[idx_5]

idx_6 = np.array( np.where(y_train_All==6)  ).T
idx_6 = idx_6[0:ssz,0]
X_6 = X_train_All[idx_6, :, :, :]
y_6 = y_train_All[idx_6]

idx_7 = np.array( np.where(y_train_All==7)  ).T
idx_7 = idx_7[0:ssz,0]
X_7 = X_train_All[idx_7, :, :, :]
y_7 = y_train_All[idx_7]

idx_8 = np.array( np.where(y_train_All==8)  ).T
idx_8 = idx_8[0:ssz,0]
X_8 = X_train_All[idx_8, :, :, :]
y_8 = y_train_All[idx_8]
```

```
        idx_9 = np.array( np.where(y_train_All==9)  ).T
        idx_9 = idx_9[0:ssz,0]
        X_9 = X_train_All[idx_9, :, :, :]
        y_9 = y_train_All[idx_9]

        idx_10 = np.array( np.where(y_train_All==10)  ).T
        idx_10 = idx_10[0:ssz,0]
        X_10 = X_train_All[idx_10, :, :, :]
        y_10 = y_train_All[idx_10]

        idx_11 = np.array( np.where(y_train_All==11)  ).T
        idx_11 = idx_11[0:ssz,0]
        X_11 = X_train_All[idx_11, :, :, :]
        y_11 = y_train_All[idx_11]

        idx_12 = np.array( np.where(y_train_All==12)  ).T
        idx_12 = idx_12[0:ssz,0]
        X_12 = X_train_All[idx_12, :, :, :]
        y_12 = y_train_All[idx_12]


        idx_13 = np.array( np.where(y_train_All==13)  ).T
        idx_13 = idx_13[0:ssz,0]
        X_13 = X_train_All[idx_13, :, :, :]
        y_13 = y_train_All[idx_13]

        idx_14 = np.array( np.where(y_train_All==14)  ).T
        idx_14 = idx_14[0:ssz,0]
        X_14 = X_train_All[idx_14, :, :, :]
        y_14 = y_train_All[idx_14]

        idx_15 = np.array( np.where(y_train_All==15)  ).T
        idx_15 = idx_15[0:ssz,0]
        X_15 = X_train_All[idx_15, :, :, :]
        y_15 = y_train_All[idx_15]


    # New tarin data
        X_train = np.concatenate(( X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8,␣
↪X_9,X_10,X_11,X_12,X_13,X_14,X_15), axis=0 )
        y_train = np.concatenate(( y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8,␣
↪y_9,y_10,y_11,y_12,y_13,y_14,y_15), axis=0 )


        print('X_train shape:', X_train.shape)
        print(X_train.shape[0], 'train samples')
```

```
    # here to put
        X_train = X_train.astype('float32')
        X_test = X_test.astype('float32')
        X_valid = X_valid.astype('float32')
        X_Pool = X_Pool.astype('float32')
        X_train /= 255
        X_valid /= 255
        X_Pool /= 255
        X_test /= 255


        Y_test = np_utils.to_categorical(y_test, nb_classes)
        Y_valid = np_utils.to_categorical(y_valid, nb_classes)
        Y_Pool = np_utils.to_categorical(y_Pool, nb_classes)


        #loss values in each experiment
        Pool_Valid_Loss = np.zeros(shape=(nb_epoch, 1))
        Pool_Train_Loss = np.zeros(shape=(nb_epoch, 1))
        Pool_Valid_Acc = np.zeros(shape=(nb_epoch, 1))
        Pool_Train_Acc = np.zeros(shape=(nb_epoch, 1))
        x_pool_All = np.zeros(shape=(1))

        Y_train = np_utils.to_categorical(y_train, nb_classes)

        print('Training Model Without Acquisitions in Experiment', e)

    #Xvalid, yvalid

        #model_bayes.compile(loss=negative_log_likelihood, optimizer='adam',␣
↪metrics='accuracy')
        hist = model_bayes.fit(X_train, Y_train, batch_size=batch_size, epochs␣
↪=nb_epoch, validation_data=(X_valid, Y_valid))

    #loss, accuracy = model.evaluate(test_X, test_y_ohe, verbose=0)

        Train_Result_Optimizer = hist.history
        Train_Loss = np.asarray(Train_Result_Optimizer.get('loss'))
        Train_Loss = np.array([Train_Loss]).T
        Valid_Loss = np.asarray(Train_Result_Optimizer.get('val_loss'))
        Valid_Loss = np.asarray([Valid_Loss]).T
        Train_Acc = np.asarray(Train_Result_Optimizer.get('acc'))
        Train_Acc = np.array([Train_Acc]).T
        Valid_Acc = np.asarray(Train_Result_Optimizer.get('val_acc'))
        Valid_Acc = np.asarray([Valid_Acc]).T
```

```
        Pool_Train_Loss = Train_Loss
        Pool_Valid_Loss = Valid_Loss
        Pool_Train_Acc = Train_Acc
        Pool_Valid_Acc = Valid_Acc

    # Xtrain ytrain   Xtest  ytest    Xvalid yvalid
        print('Evaluating Test Accuracy Without Acquisition')
         # X_test  y_test Xtest ytest
    score, acc = model_bayes.evaluate(X_test, y_test,  verbose=0)

        all_accuracy = acc

        print('Starting Active Learning in Experiment ', e)



        for i in range(3): # acquisition_iterations

                print('POOLING ITERATION', i)

                #take subset of Pool Points for Test Time Dropout
                #and do acquisition from there
                pool_subset = 500
                pool_subset_dropout = np.asarray(random.sample(range(0,X_Pool.
↪shape[0]), pool_subset))
                X_Pool_Dropout = X_Pool[pool_subset_dropout, :, :, :]
                y_Pool_Dropout = y_Pool[pool_subset_dropout]

                score_All = np.zeros(shape=(X_Pool_Dropout.shape[0],␣
↪nb_classes))
                All_Entropy_Dropout = np.zeros(shape=X_Pool_Dropout.shape[0])

                for d in range(dropout_iterations):
                        print ('Dropout Iteration', d)
                        dropout_score = model_bayes.
↪predict(X_Pool_Dropout,batch_size=batch_size, verbose=1)
                        #dropout_score = np_utils.to_categorical(y_Pool,␣
↪nb_classes)


                        #computing G_X
                        score_All = score_All + dropout_score

                        #computing F_X
                        dropout_score_log = np.log2(dropout_score+2)
```

```
                        Entropy_Compute = - np.multiply(dropout_score,␣
↪dropout_score_log)
                        Entropy_Per_Dropout = np.sum(Entropy_Compute, axis=1)

                        All_Entropy_Dropout = All_Entropy_Dropout +␣
↪Entropy_Per_Dropout


                Avg_Pi = np.divide(score_All, dropout_iterations)
                Log_Avg_Pi = np.log2(Avg_Pi+2)
                Entropy_Avg_Pi = - np.multiply(Avg_Pi, Log_Avg_Pi)
                Entropy_Average_Pi = np.sum(Entropy_Avg_Pi, axis=1)

                G_X = Entropy_Average_Pi

                Average_Entropy = np.divide(All_Entropy_Dropout,␣
↪dropout_iterations)

                F_X = Average_Entropy

                U_X = G_X - F_X

                # find min index
                # a_1d = U_X.flatten()
                # x_pool_index = a_1d.argsort()[-Queries:]

                a_1d = U_X.flatten()
                x_pool_index = a_1d.argsort()[-Queries:][::-1]


                #store all the pooled images indexes
                x_pool_All = np.append(x_pool_All, x_pool_index)

                #saving

                Pooled_X = X_Pool_Dropout[x_pool_index, :, :, :]
                Pooled_Y = y_Pool_Dropout[x_pool_index]

                # delete the random subset used for test time dropout from␣
↪X_Pool
                # Delete the pooled point from this pool set

                delete_Pool_X = np.delete(X_Pool, (pool_subset_dropout), axis=0)
                delete_Pool_Y = np.delete(y_Pool, (pool_subset_dropout), axis=0)

                delete_Pool_X_Dropout = np.delete(X_Pool_Dropout,␣
↪(x_pool_index), axis=0)
```

```
                delete_Pool_Y_Dropout = np.delete(y_Pool_Dropout,␣
↪(x_pool_index), axis=0)

                X_Pool = np.concatenate((delete_Pool_X, delete_Pool_X_Dropout),␣
↪axis=0)
                y_Pool = np.concatenate((delete_Pool_Y, delete_Pool_Y_Dropout),␣
↪axis=0)



                print('Acquised Points added to training set')
                X_train = np.concatenate((X_train, Pooled_X), axis=0)
                y_train = np.concatenate((y_train, Pooled_Y), axis=0)

                # convert class vectors to binary class matrices

                Y_train = np_utils.to_categorical(y_train, nb_classes)

                #Xtrain, ytrain, ytest,Xtest   Xvalid, yvalid


                #model_bayes.compile(loss=negative_log_likelihood,␣
↪optimizer='adam',metrics='accuracy')
                hist = model_bayes.fit(X_train, Y_train, batch_size=batch_size,␣
↪epochs=nb_epoch, verbose=1, validation_data=(X_valid, Y_valid))

                Train_Result_Optimizer = hist.history
                Train_Loss = np.asarray(Train_Result_Optimizer.get('loss'))
                Train_Loss = np.array([Train_Loss]).T
                Valid_Loss = np.asarray(Train_Result_Optimizer.get('val_loss'))
                Valid_Loss = np.asarray([Valid_Loss]).T
                Train_Acc = np.asarray(Train_Result_Optimizer.get('acc'))
                Train_Acc = np.array([Train_Acc]).T
                Valid_Acc = np.asarray(Train_Result_Optimizer.get('val_acc'))
                Valid_Acc = np.asarray([Valid_Acc]).T

                #Accumulate the training and validation/test loss after every␣
↪pooling iteration - for plotting

                Pool_Valid_Loss = np.append(Pool_Valid_Loss, Valid_Loss, axis=1)
                Pool_Train_Loss = np.append(Pool_Train_Loss, Train_Loss, axis=1)

                print('Evaluate Model Test Accuracy with pooled points')

                score, acc = model_bayes.evaluate(X_test, Y_test , verbose=1)

                print('Test score:', score)
```

```
                print('Test accuracy:', acc)
                all_accuracy = np.append(all_accuracy, acc)

                print('Use this trained model with pooled points for Dropout␣
↪again')


        print('Storing Accuracy Values over experiments')
        Experiments_All_Accuracy =  all_accuracy
```

```
Experiment Number  0
X_train shape: (40, 7, 7, 80)
40 train samples
Training Model Without Acquisitions in Experiment 0
Epoch 1/20
57/57 [==============================] - 2s 18ms/step - loss: 12.9481 -
accuracy: 0.2774 - val_loss: 11.9594 - val_accuracy: 0.5358
Epoch 2/20
57/57 [==============================] - 1s 18ms/step - loss: 11.6220 -
accuracy: 0.6583 - val_loss: 11.3386 - val_accuracy: 0.7516
Epoch 3/20
57/57 [==============================] - 1s 13ms/step - loss: 11.1555 -
accuracy: 0.8173 - val_loss: 11.0267 - val_accuracy: 0.8557
Epoch 4/20
57/57 [==============================] - 1s 12ms/step - loss: 10.9267 -
accuracy: 0.8793 - val_loss: 10.8309 - val_accuracy: 0.9002
Epoch 5/20
57/57 [==============================] - 1s 15ms/step - loss: 10.7345 -
accuracy: 0.9235 - val_loss: 10.6655 - val_accuracy: 0.9295
Epoch 6/20
57/57 [==============================] - 1s 13ms/step - loss: 10.5775 -
accuracy: 0.9501 - val_loss: 10.5577 - val_accuracy: 0.9338
Epoch 7/20
57/57 [==============================] - 1s 12ms/step - loss: 10.4641 -
accuracy: 0.9540 - val_loss: 10.4156 - val_accuracy: 0.9447
Epoch 8/20
57/57 [==============================] - 1s 13ms/step - loss: 10.3317 -
accuracy: 0.9684 - val_loss: 10.2832 - val_accuracy: 0.9685
Epoch 9/20
57/57 [==============================] - 1s 12ms/step - loss: 10.2107 -
accuracy: 0.9732 - val_loss: 10.1703 - val_accuracy: 0.9631
Epoch 10/20
57/57 [==============================] - 1s 13ms/step - loss: 10.0811 -
accuracy: 0.9796 - val_loss: 10.0641 - val_accuracy: 0.9707
Epoch 11/20
57/57 [==============================] - 1s 13ms/step - loss: 9.9616 - accuracy:
0.9823 - val_loss: 9.9551 - val_accuracy: 0.9642
Epoch 12/20
```

```
57/57 [==============================] - 1s 17ms/step - loss: 9.8442 - accuracy:
0.9791 - val_loss: 9.8061 - val_accuracy: 0.9685
Epoch 13/20
57/57 [==============================] - 1s 18ms/step - loss: 9.7258 - accuracy:
0.9819 - val_loss: 9.6774 - val_accuracy: 0.9740
Epoch 14/20
57/57 [==============================] - 1s 15ms/step - loss: 9.5926 - accuracy:
0.9854 - val_loss: 9.5677 - val_accuracy: 0.9794
Epoch 15/20
57/57 [==============================] - 1s 16ms/step - loss: 9.4769 - accuracy:
0.9837 - val_loss: 9.4739 - val_accuracy: 0.9588
Epoch 16/20
57/57 [==============================] - 1s 13ms/step - loss: 9.3460 - accuracy:
0.9872 - val_loss: 9.3223 - val_accuracy: 0.9740
Epoch 17/20
57/57 [==============================] - 1s 12ms/step - loss: 9.2228 - accuracy:
0.9849 - val_loss: 9.1944 - val_accuracy: 0.9751
Epoch 18/20
57/57 [==============================] - 1s 15ms/step - loss: 9.1019 - accuracy:
0.9837 - val_loss: 9.0459 - val_accuracy: 0.9826
Epoch 19/20
57/57 [==============================] - 1s 13ms/step - loss: 8.9745 - accuracy:
0.9851 - val_loss: 8.9526 - val_accuracy: 0.9707
Epoch 20/20
57/57 [==============================] - 1s 13ms/step - loss: 8.8540 - accuracy:
0.9862 - val_loss: 8.8254 - val_accuracy: 0.9707
Evaluating Test Accuracy Without Acquisition
Starting Active Learning in Experiment  0
POOLING ITERATION 0
Dropout Iteration 0
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 1
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 2
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 3
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 4
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 5
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 6
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 7
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 8
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 9
```

```
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 10
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 11
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 12
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 13
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 14
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 15
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 16
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 17
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 18
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 19
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 20
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 21
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 22
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 23
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 24
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 25
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 26
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 27
4/4 [==============================] - 0s 7ms/step
Dropout Iteration 28
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 29
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 30
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 31
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 32
4/4 [==============================] - 0s 6ms/step
Dropout Iteration 33
```

```
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 34
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 35
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 36
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 37
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 38
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 39
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 40
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 41
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 42
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 43
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 44
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 45
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 46
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 47
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 48
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 49
4/4 [==============================] - 0s 3ms/step
Acquised Points added to training set
Epoch 1/20
57/57 [==============================] - 1s 21ms/step - loss: 8.7290 - accuracy:
0.9856 - val_loss: 8.6654 - val_accuracy: 0.9805
Epoch 2/20
57/57 [==============================] - 1s 18ms/step - loss: 8.6131 - accuracy:
0.9855 - val_loss: 8.5689 - val_accuracy: 0.9837
Epoch 3/20
57/57 [==============================] - 1s 15ms/step - loss: 8.4908 - accuracy:
0.9859 - val_loss: 8.4808 - val_accuracy: 0.9729
Epoch 4/20
57/57 [==============================] - 1s 14ms/step - loss: 8.3727 - accuracy:
0.9866 - val_loss: 8.3348 - val_accuracy: 0.9761
Epoch 5/20
57/57 [==============================] - 1s 14ms/step - loss: 8.2779 - accuracy:
```

```
0.9817 - val_loss: 8.2284 - val_accuracy: 0.9794
Epoch 6/20
57/57 [==============================] - 1s 13ms/step - loss: 8.1376 - accuracy:
0.9861 - val_loss: 8.1146 - val_accuracy: 0.9848
Epoch 7/20
57/57 [==============================] - 1s 13ms/step - loss: 8.0266 - accuracy:
0.9855 - val_loss: 7.9683 - val_accuracy: 0.9837
Epoch 8/20
57/57 [==============================] - 1s 13ms/step - loss: 7.9232 - accuracy:
0.9849 - val_loss: 7.9216 - val_accuracy: 0.9696
Epoch 9/20
57/57 [==============================] - 1s 13ms/step - loss: 7.8040 - accuracy:
0.9869 - val_loss: 7.7741 - val_accuracy: 0.9751
Epoch 10/20
57/57 [==============================] - 1s 15ms/step - loss: 7.6947 - accuracy:
0.9880 - val_loss: 7.7065 - val_accuracy: 0.9718
Epoch 11/20
57/57 [==============================] - 1s 17ms/step - loss: 7.6045 - accuracy:
0.9858 - val_loss: 7.5840 - val_accuracy: 0.9826
Epoch 12/20
57/57 [==============================] - 1s 25ms/step - loss: 7.4857 - accuracy:
0.9872 - val_loss: 7.4243 - val_accuracy: 0.9881
Epoch 13/20
57/57 [==============================] - 2s 32ms/step - loss: 7.3756 - accuracy:
0.9891 - val_loss: 7.3575 - val_accuracy: 0.9751
Epoch 14/20
57/57 [==============================] - 1s 13ms/step - loss: 7.2909 - accuracy:
0.9837 - val_loss: 7.2444 - val_accuracy: 0.9870
Epoch 15/20
57/57 [==============================] - 1s 26ms/step - loss: 7.1825 - accuracy:
0.9880 - val_loss: 7.1589 - val_accuracy: 0.9826
Epoch 16/20
57/57 [==============================] - 1s 21ms/step - loss: 7.0898 - accuracy:
0.9869 - val_loss: 7.1233 - val_accuracy: 0.9631
Epoch 17/20
57/57 [==============================] - 1s 13ms/step - loss: 6.9870 - accuracy:
0.9876 - val_loss: 7.0066 - val_accuracy: 0.9729
Epoch 18/20
57/57 [==============================] - 1s 14ms/step - loss: 6.8957 - accuracy:
0.9859 - val_loss: 6.9428 - val_accuracy: 0.9653
Epoch 19/20
57/57 [==============================] - 1s 15ms/step - loss: 6.8111 - accuracy:
0.9859 - val_loss: 6.7492 - val_accuracy: 0.9870
Epoch 20/20
57/57 [==============================] - 1s 13ms/step - loss: 6.7138 - accuracy:
0.9873 - val_loss: 6.6872 - val_accuracy: 0.9794
Evaluate Model Test Accuracy with pooled points
68/68 [==============================] - 0s 2ms/step - loss: 6.7375 - accuracy:
```

```
0.9763
Test score: 6.73746395111084
Test accuracy: 0.9763121008872986
Use this trained model with pooled points for Dropout again
POOLING ITERATION 1
Dropout Iteration 0
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 1
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 2
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 3
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 4
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 5
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 6
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 7
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 8
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 9
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 10
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 11
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 12
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 13
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 14
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 15
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 16
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 17
4/4 [==============================] - 0s 10ms/step
Dropout Iteration 18
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 19
4/4 [==============================] - 0s 6ms/step
Dropout Iteration 20
4/4 [==============================] - 0s 6ms/step
Dropout Iteration 21
```

```
4/4 [==============================] - 0s 15ms/step
Dropout Iteration 22
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 23
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 24
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 25
4/4 [==============================] - 0s 6ms/step
Dropout Iteration 26
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 27
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 28
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 29
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 30
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 31
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 32
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 33
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 34
4/4 [==============================] - 0s 7ms/step
Dropout Iteration 35
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 36
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 37
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 38
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 39
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 40
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 41
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 42
4/4 [==============================] - 0s 13ms/step
Dropout Iteration 43
4/4 [==============================] - 0s 9ms/step
Dropout Iteration 44
4/4 [==============================] - 0s 26ms/step
Dropout Iteration 45
```

```
4/4 [==============================] - 0s 6ms/step
Dropout Iteration 46
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 47
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 48
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 49
4/4 [==============================] - 0s 5ms/step
Acquised Points added to training set
Epoch 1/20
57/57 [==============================] - 1s 24ms/step - loss: 6.6435 - accuracy:
0.9845 - val_loss: 6.6113 - val_accuracy: 0.9783
Epoch 2/20
57/57 [==============================] - 1s 18ms/step - loss: 6.5474 - accuracy:
0.9848 - val_loss: 6.5240 - val_accuracy: 0.9794
Epoch 3/20
57/57 [==============================] - 1s 19ms/step - loss: 6.4663 - accuracy:
0.9849 - val_loss: 6.4540 - val_accuracy: 0.9740
Epoch 4/20
57/57 [==============================] - 1s 17ms/step - loss: 6.3767 - accuracy:
0.9869 - val_loss: 6.3966 - val_accuracy: 0.9761
Epoch 5/20
57/57 [==============================] - 1s 17ms/step - loss: 6.2927 - accuracy:
0.9863 - val_loss: 6.2861 - val_accuracy: 0.9805
Epoch 6/20
57/57 [==============================] - 1s 19ms/step - loss: 6.2166 - accuracy:
0.9880 - val_loss: 6.1999 - val_accuracy: 0.9816
Epoch 7/20
57/57 [==============================] - 1s 19ms/step - loss: 6.1332 - accuracy:
0.9863 - val_loss: 6.1119 - val_accuracy: 0.9826
Epoch 8/20
57/57 [==============================] - 1s 16ms/step - loss: 6.0745 - accuracy:
0.9838 - val_loss: 6.0542 - val_accuracy: 0.9816
Epoch 9/20
57/57 [==============================] - 1s 16ms/step - loss: 6.0009 - accuracy:
0.9840 - val_loss: 6.0236 - val_accuracy: 0.9707
Epoch 10/20
57/57 [==============================] - 1s 25ms/step - loss: 5.9238 - accuracy:
0.9844 - val_loss: 5.8852 - val_accuracy: 0.9870
Epoch 11/20
57/57 [==============================] - 2s 42ms/step - loss: 5.8584 - accuracy:
0.9833 - val_loss: 5.8682 - val_accuracy: 0.9707
Epoch 12/20
57/57 [==============================] - 1s 16ms/step - loss: 5.7778 - accuracy:
0.9868 - val_loss: 5.7728 - val_accuracy: 0.9816
Epoch 13/20
57/57 [==============================] - 1s 21ms/step - loss: 5.7204 - accuracy:
```

```
0.9833 - val_loss: 5.7189 - val_accuracy: 0.9751
Epoch 14/20
57/57 [==============================] - 1s 15ms/step - loss: 5.6551 - accuracy:
0.9856 - val_loss: 5.6427 - val_accuracy: 0.9751
Epoch 15/20
57/57 [==============================] - 1s 20ms/step - loss: 5.5788 - accuracy:
0.9886 - val_loss: 5.6209 - val_accuracy: 0.9729
Epoch 16/20
57/57 [==============================] - 1s 24ms/step - loss: 5.6929 - accuracy:
0.9578 - val_loss: 5.5538 - val_accuracy: 0.9740
Epoch 17/20
57/57 [==============================] - 1s 16ms/step - loss: 5.4840 - accuracy:
0.9842 - val_loss: 5.5078 - val_accuracy: 0.9696
Epoch 18/20
57/57 [==============================] - 1s 23ms/step - loss: 5.4301 - accuracy:
0.9877 - val_loss: 5.4214 - val_accuracy: 0.9837
Epoch 19/20
57/57 [==============================] - 2s 28ms/step - loss: 5.3789 - accuracy:
0.9882 - val_loss: 5.3505 - val_accuracy: 0.9870
Epoch 20/20
57/57 [==============================] - 1s 18ms/step - loss: 5.3425 - accuracy:
0.9829 - val_loss: 5.3218 - val_accuracy: 0.9794
Evaluate Model Test Accuracy with pooled points
68/68 [==============================] - 0s 2ms/step - loss: 5.3013 - accuracy:
0.9847
Test score: 5.3012590408325195
Test accuracy: 0.9846725463867188
Use this trained model with pooled points for Dropout again
POOLING ITERATION 2
Dropout Iteration 0
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 1
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 2
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 3
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 4
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 5
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 6
4/4 [==============================] - 0s 8ms/step
Dropout Iteration 7
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 8
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 9
```

```
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 10
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 11
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 12
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 13
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 14
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 15
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 16
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 17
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 18
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 19
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 20
4/4 [==============================] - 0s 13ms/step
Dropout Iteration 21
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 22
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 23
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 24
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 25
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 26
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 27
4/4 [==============================] - 0s 6ms/step
Dropout Iteration 28
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 29
4/4 [==============================] - 0s 7ms/step
Dropout Iteration 30
4/4 [==============================] - 0s 7ms/step
Dropout Iteration 31
4/4 [==============================] - 0s 8ms/step
Dropout Iteration 32
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 33
```

```
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 34
4/4 [==============================] - 0s 7ms/step
Dropout Iteration 35
4/4 [==============================] - 0s 6ms/step
Dropout Iteration 36
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 37
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 38
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 39
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 40
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 41
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 42
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 43
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 44
4/4 [==============================] - 0s 7ms/step
Dropout Iteration 45
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 46
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 47
4/4 [==============================] - 0s 7ms/step
Dropout Iteration 48
4/4 [==============================] - 0s 8ms/step
Dropout Iteration 49
4/4 [==============================] - 0s 4ms/step
Acquised Points added to training set
Epoch 1/20
57/57 [==============================] - 2s 28ms/step - loss: 5.2777 - accuracy:
0.9847 - val_loss: 5.2991 - val_accuracy: 0.9783
Epoch 2/20
57/57 [==============================] - 1s 21ms/step - loss: 5.2371 - accuracy:
0.9861 - val_loss: 5.2439 - val_accuracy: 0.9859
Epoch 3/20
57/57 [==============================] - 1s 20ms/step - loss: 5.1775 - accuracy:
0.9880 - val_loss: 5.1946 - val_accuracy: 0.9783
Epoch 4/20
57/57 [==============================] - 1s 19ms/step - loss: 5.1305 - accuracy:
0.9884 - val_loss: 5.1243 - val_accuracy: 0.9826
Epoch 5/20
57/57 [==============================] - 1s 22ms/step - loss: 5.0744 - accuracy:
```

```
0.9877 - val_loss: 5.0669 - val_accuracy: 0.9881
Epoch 6/20
57/57 [==============================] - 1s 20ms/step - loss: 5.0482 - accuracy:
0.9845 - val_loss: 5.0716 - val_accuracy: 0.9805
Epoch 7/20
57/57 [==============================] - 1s 15ms/step - loss: 5.0024 - accuracy:
0.9833 - val_loss: 4.9749 - val_accuracy: 0.9870
Epoch 8/20
57/57 [==============================] - 1s 21ms/step - loss: 4.9591 - accuracy:
0.9884 - val_loss: 4.9676 - val_accuracy: 0.9794
Epoch 9/20
57/57 [==============================] - 1s 17ms/step - loss: 4.9362 - accuracy:
0.9847 - val_loss: 4.9469 - val_accuracy: 0.9783
Epoch 10/20
57/57 [==============================] - 1s 16ms/step - loss: 4.9033 - accuracy:
0.9806 - val_loss: 4.8860 - val_accuracy: 0.9805
Epoch 11/20
57/57 [==============================] - 1s 18ms/step - loss: 4.8408 - accuracy:
0.9852 - val_loss: 4.8781 - val_accuracy: 0.9772
Epoch 12/20
57/57 [==============================] - 1s 19ms/step - loss: 4.7924 - accuracy:
0.9868 - val_loss: 4.7660 - val_accuracy: 0.9848
Epoch 13/20
57/57 [==============================] - 1s 16ms/step - loss: 4.7513 - accuracy:
0.9861 - val_loss: 4.7999 - val_accuracy: 0.9740
Epoch 14/20
57/57 [==============================] - 1s 15ms/step - loss: 4.7124 - accuracy:
0.9873 - val_loss: 4.7581 - val_accuracy: 0.9805
Epoch 15/20
57/57 [==============================] - 1s 17ms/step - loss: 4.6969 - accuracy:
0.9855 - val_loss: 4.6672 - val_accuracy: 0.9859
Epoch 16/20
57/57 [==============================] - 1s 15ms/step - loss: 4.6451 - accuracy:
0.9844 - val_loss: 4.6746 - val_accuracy: 0.9783
Epoch 17/20
57/57 [==============================] - 1s 17ms/step - loss: 4.5987 - accuracy:
0.9866 - val_loss: 4.5817 - val_accuracy: 0.9837
Epoch 18/20
57/57 [==============================] - 1s 16ms/step - loss: 4.5588 - accuracy:
0.9873 - val_loss: 4.5670 - val_accuracy: 0.9837
Epoch 19/20
57/57 [==============================] - 1s 15ms/step - loss: 4.5209 - accuracy:
0.9876 - val_loss: 4.5315 - val_accuracy: 0.9859
Epoch 20/20
57/57 [==============================] - 1s 16ms/step - loss: 4.4784 - accuracy:
0.9880 - val_loss: 4.5074 - val_accuracy: 0.9859
Evaluate Model Test Accuracy with pooled points
68/68 [==============================] - 0s 3ms/step - loss: 4.4768 - accuracy:
```

```
0.9847
Test score: 4.47678804397583
Test accuracy: 0.9846725463867188
Use this trained model with pooled points for Dropout again
Storing Accuracy Values over experiments
Experiment Number  1
X_train shape: (40, 7, 7, 80)
40 train samples
Training Model Without Acquisitions in Experiment 1
Epoch 1/20
57/57 [==============================] - 1s 17ms/step - loss: 4.4707 - accuracy:
0.9852 - val_loss: 4.4413 - val_accuracy: 0.9751
Epoch 2/20
57/57 [==============================] - 1s 15ms/step - loss: 4.4381 - accuracy:
0.9824 - val_loss: 4.4446 - val_accuracy: 0.9805
Epoch 3/20
57/57 [==============================] - 1s 17ms/step - loss: 4.3715 - accuracy:
0.9866 - val_loss: 4.4212 - val_accuracy: 0.9772
Epoch 4/20
57/57 [==============================] - 1s 15ms/step - loss: 4.3371 - accuracy:
0.9880 - val_loss: 4.3384 - val_accuracy: 0.9794
Epoch 5/20
57/57 [==============================] - 1s 15ms/step - loss: 4.2966 - accuracy:
0.9861 - val_loss: 4.3175 - val_accuracy: 0.9805
Epoch 6/20
57/57 [==============================] - 1s 16ms/step - loss: 4.2662 - accuracy:
0.9869 - val_loss: 4.2550 - val_accuracy: 0.9794
Epoch 7/20
57/57 [==============================] - 1s 15ms/step - loss: 4.2311 - accuracy:
0.9868 - val_loss: 4.2635 - val_accuracy: 0.9837
Epoch 8/20
57/57 [==============================] - 1s 14ms/step - loss: 4.2297 - accuracy:
0.9834 - val_loss: 4.2547 - val_accuracy: 0.9664
Epoch 9/20
57/57 [==============================] - 1s 13ms/step - loss: 4.1745 - accuracy:
0.9841 - val_loss: 4.1753 - val_accuracy: 0.9826
Epoch 10/20
57/57 [==============================] - 1s 13ms/step - loss: 4.1505 - accuracy:
0.9868 - val_loss: 4.2115 - val_accuracy: 0.9664
Epoch 11/20
57/57 [==============================] - 1s 14ms/step - loss: 4.1186 - accuracy:
0.9834 - val_loss: 4.1529 - val_accuracy: 0.9772
Epoch 12/20
57/57 [==============================] - 1s 13ms/step - loss: 4.1004 - accuracy:
0.9812 - val_loss: 4.0811 - val_accuracy: 0.9848
Epoch 13/20
57/57 [==============================] - 1s 14ms/step - loss: 4.0443 - accuracy:
0.9851 - val_loss: 4.0336 - val_accuracy: 0.9881
```

```
Epoch 14/20
57/57 [==============================] - 1s 13ms/step - loss: 4.0464 - accuracy:
0.9803 - val_loss: 4.0178 - val_accuracy: 0.9816
Epoch 15/20
57/57 [==============================] - 1s 12ms/step - loss: 4.0279 - accuracy:
0.9787 - val_loss: 3.9820 - val_accuracy: 0.9805
Epoch 16/20
57/57 [==============================] - 1s 12ms/step - loss: 3.9529 - accuracy:
0.9863 - val_loss: 3.9925 - val_accuracy: 0.9772
Epoch 17/20
57/57 [==============================] - 1s 12ms/step - loss: 3.9367 - accuracy:
0.9858 - val_loss: 3.9557 - val_accuracy: 0.9783
Epoch 18/20
57/57 [==============================] - 1s 12ms/step - loss: 3.9233 - accuracy:
0.9844 - val_loss: 3.9321 - val_accuracy: 0.9761
Epoch 19/20
57/57 [==============================] - 1s 12ms/step - loss: 3.8763 - accuracy:
0.9851 - val_loss: 3.8599 - val_accuracy: 0.9826
Epoch 20/20
57/57 [==============================] - 1s 13ms/step - loss: 3.8407 - accuracy:
0.9870 - val_loss: 3.8488 - val_accuracy: 0.9805
Evaluating Test Accuracy Without Acquisition
Starting Active Learning in Experiment  1
POOLING ITERATION 0
Dropout Iteration 0
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 1
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 2
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 3
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 4
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 5
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 6
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 7
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 8
4/4 [==============================] - 0s 6ms/step
Dropout Iteration 9
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 10
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 11
4/4 [==============================] - 0s 4ms/step
```

```
Dropout Iteration 12
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 13
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 14
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 15
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 16
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 17
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 18
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 19
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 20
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 21
4/4 [==============================] - 0s 7ms/step
Dropout Iteration 22
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 23
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 24
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 25
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 26
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 27
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 28
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 29
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 30
4/4 [==============================] - 0s 2ms/step
Dropout Iteration 31
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 32
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 33
4/4 [==============================] - 0s 2ms/step
Dropout Iteration 34
4/4 [==============================] - 0s 2ms/step
Dropout Iteration 35
4/4 [==============================] - 0s 4ms/step
```

```
Dropout Iteration 36
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 37
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 38
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 39
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 40
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 41
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 42
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 43
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 44
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 45
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 46
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 47
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 48
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 49
4/4 [==============================] - 0s 3ms/step
Acquised Points added to training set
Epoch 1/20
57/57 [==============================] - 1s 12ms/step - loss: 3.8244 - accuracy:
0.9841 - val_loss: 3.8367 - val_accuracy: 0.9848
Epoch 2/20
57/57 [==============================] - 1s 11ms/step - loss: 3.8242 - accuracy:
0.9806 - val_loss: 3.8301 - val_accuracy: 0.9783
Epoch 3/20
57/57 [==============================] - 1s 13ms/step - loss: 3.7529 - accuracy:
0.9870 - val_loss: 3.7790 - val_accuracy: 0.9848
Epoch 4/20
57/57 [==============================] - 1s 13ms/step - loss: 3.7285 - accuracy:
0.9869 - val_loss: 3.7826 - val_accuracy: 0.9837
Epoch 5/20
57/57 [==============================] - 1s 14ms/step - loss: 3.7302 - accuracy:
0.9824 - val_loss: 3.7382 - val_accuracy: 0.9805
Epoch 6/20
57/57 [==============================] - 1s 13ms/step - loss: 3.7222 - accuracy:
0.9806 - val_loss: 3.7909 - val_accuracy: 0.9675
Epoch 7/20
```

```
57/57 [==============================] - 1s 14ms/step - loss: 3.7461 - accuracy:
0.9710 - val_loss: 3.7006 - val_accuracy: 0.9805
Epoch 8/20
57/57 [==============================] - 1s 13ms/step - loss: 3.6358 - accuracy:
0.9856 - val_loss: 3.6631 - val_accuracy: 0.9816
Epoch 9/20
57/57 [==============================] - 1s 11ms/step - loss: 3.6548 - accuracy:
0.9848 - val_loss: 3.6474 - val_accuracy: 0.9848
Epoch 10/20
57/57 [==============================] - 1s 14ms/step - loss: 3.6234 - accuracy:
0.9822 - val_loss: 3.6316 - val_accuracy: 0.9794
Epoch 11/20
57/57 [==============================] - 1s 13ms/step - loss: 3.6115 - accuracy:
0.9791 - val_loss: 3.5475 - val_accuracy: 0.9816
Epoch 12/20
57/57 [==============================] - 1s 10ms/step - loss: 3.5777 - accuracy:
0.9824 - val_loss: 3.5863 - val_accuracy: 0.9729
Epoch 13/20
57/57 [==============================] - 1s 12ms/step - loss: 3.5467 - accuracy:
0.9783 - val_loss: 3.5459 - val_accuracy: 0.9783
Epoch 14/20
57/57 [==============================] - 1s 13ms/step - loss: 3.5109 - accuracy:
0.9840 - val_loss: 3.4628 - val_accuracy: 0.9913
Epoch 15/20
57/57 [==============================] - 1s 12ms/step - loss: 3.5017 - accuracy:
0.9851 - val_loss: 3.4705 - val_accuracy: 0.9826
Epoch 16/20
57/57 [==============================] - 1s 10ms/step - loss: 3.4727 - accuracy:
0.9844 - val_loss: 3.4241 - val_accuracy: 0.9902
Epoch 17/20
57/57 [==============================] - 1s 12ms/step - loss: 3.4362 - accuracy:
0.9863 - val_loss: 3.4652 - val_accuracy: 0.9805
Epoch 18/20
57/57 [==============================] - 1s 12ms/step - loss: 3.4299 - accuracy:
0.9854 - val_loss: 3.4390 - val_accuracy: 0.9816
Epoch 19/20
57/57 [==============================] - 1s 12ms/step - loss: 3.3874 - accuracy:
0.9879 - val_loss: 3.3888 - val_accuracy: 0.9837
Epoch 20/20
57/57 [==============================] - 1s 12ms/step - loss: 3.3636 - accuracy:
0.9869 - val_loss: 3.3759 - val_accuracy: 0.9859
Evaluate Model Test Accuracy with pooled points
68/68 [==============================] - 0s 2ms/step - loss: 3.4274 - accuracy:
0.9824
Test score: 3.427382230758667
Test accuracy: 0.98235023021698
Use this trained model with pooled points for Dropout again
POOLING ITERATION 1
```

```
Dropout Iteration 0
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 1
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 2
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 3
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 4
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 5
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 6
4/4 [==============================] - 0s 2ms/step
Dropout Iteration 7
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 8
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 9
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 10
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 11
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 12
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 13
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 14
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 15
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 16
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 17
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 18
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 19
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 20
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 21
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 22
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 23
4/4 [==============================] - 0s 3ms/step
```

```
Dropout Iteration 24
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 25
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 26
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 27
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 28
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 29
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 30
4/4 [==============================] - 0s 2ms/step
Dropout Iteration 31
4/4 [==============================] - 0s 2ms/step
Dropout Iteration 32
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 33
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 34
4/4 [==============================] - 0s 2ms/step
Dropout Iteration 35
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 36
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 37
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 38
4/4 [==============================] - 0s 2ms/step
Dropout Iteration 39
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 40
4/4 [==============================] - 0s 2ms/step
Dropout Iteration 41
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 42
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 43
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 44
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 45
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 46
4/4 [==============================] - 0s 2ms/step
Dropout Iteration 47
4/4 [==============================] - 0s 3ms/step
```

```
Dropout Iteration 48
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 49
4/4 [==============================] - 0s 3ms/step
Acquised Points added to training set
Epoch 1/20
57/57 [==============================] - 1s 9ms/step - loss: 3.3389 - accuracy:
0.9875 - val_loss: 3.3546 - val_accuracy: 0.9826
Epoch 2/20
57/57 [==============================] - 1s 9ms/step - loss: 3.3215 - accuracy:
0.9879 - val_loss: 3.3540 - val_accuracy: 0.9751
Epoch 3/20
57/57 [==============================] - 1s 9ms/step - loss: 3.3257 - accuracy:
0.9837 - val_loss: 3.3249 - val_accuracy: 0.9783
Epoch 4/20
57/57 [==============================] - 1s 9ms/step - loss: 3.3019 - accuracy:
0.9842 - val_loss: 3.2829 - val_accuracy: 0.9805
Epoch 5/20
57/57 [==============================] - 1s 9ms/step - loss: 3.2620 - accuracy:
0.9859 - val_loss: 3.2666 - val_accuracy: 0.9794
Epoch 6/20
57/57 [==============================] - 1s 9ms/step - loss: 3.2520 - accuracy:
0.9837 - val_loss: 3.2394 - val_accuracy: 0.9761
Epoch 7/20
57/57 [==============================] - 1s 9ms/step - loss: 3.2287 - accuracy:
0.9844 - val_loss: 3.2340 - val_accuracy: 0.9783
Epoch 8/20
57/57 [==============================] - 1s 9ms/step - loss: 3.2196 - accuracy:
0.9833 - val_loss: 3.2371 - val_accuracy: 0.9848
Epoch 9/20
57/57 [==============================] - 1s 9ms/step - loss: 3.1877 - accuracy:
0.9859 - val_loss: 3.2757 - val_accuracy: 0.9805
Epoch 10/20
57/57 [==============================] - 1s 9ms/step - loss: 3.1719 - accuracy:
0.9865 - val_loss: 3.1931 - val_accuracy: 0.9805
Epoch 11/20
57/57 [==============================] - 1s 9ms/step - loss: 3.1363 - accuracy:
0.9861 - val_loss: 3.1673 - val_accuracy: 0.9859
Epoch 12/20
57/57 [==============================] - 1s 9ms/step - loss: 3.1337 - accuracy:
0.9837 - val_loss: 3.0734 - val_accuracy: 0.9946
Epoch 13/20
57/57 [==============================] - 1s 9ms/step - loss: 3.0877 - accuracy:
0.9880 - val_loss: 3.1689 - val_accuracy: 0.9794
Epoch 14/20
57/57 [==============================] - 1s 10ms/step - loss: 3.0814 - accuracy:
0.9869 - val_loss: 3.1179 - val_accuracy: 0.9772
Epoch 15/20
```

```
57/57 [==============================] - 1s 9ms/step - loss: 3.0835 - accuracy:
0.9813 - val_loss: 3.0649 - val_accuracy: 0.9783
Epoch 16/20
57/57 [==============================] - 1s 10ms/step - loss: 3.0479 - accuracy:
0.9847 - val_loss: 3.0309 - val_accuracy: 0.9892
Epoch 17/20
57/57 [==============================] - 1s 11ms/step - loss: 3.0235 - accuracy:
0.9856 - val_loss: 3.0687 - val_accuracy: 0.9805
Epoch 18/20
57/57 [==============================] - 1s 10ms/step - loss: 3.0245 - accuracy:
0.9816 - val_loss: 3.1469 - val_accuracy: 0.9729
Epoch 19/20
57/57 [==============================] - 1s 9ms/step - loss: 3.0327 - accuracy:
0.9787 - val_loss: 2.9933 - val_accuracy: 0.9740
Epoch 20/20
57/57 [==============================] - 1s 9ms/step - loss: 2.9857 - accuracy:
0.9819 - val_loss: 3.0102 - val_accuracy: 0.9837
Evaluate Model Test Accuracy with pooled points
68/68 [==============================] - 0s 2ms/step - loss: 2.9754 - accuracy:
0.9814
Test score: 2.975449562072754
Test accuracy: 0.9814212918281555
Use this trained model with pooled points for Dropout again
POOLING ITERATION 2
Dropout Iteration 0
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 1
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 2
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 3
4/4 [==============================] - 0s 2ms/step
Dropout Iteration 4
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 5
4/4 [==============================] - 0s 2ms/step
Dropout Iteration 6
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 7
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 8
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 9
4/4 [==============================] - 0s 2ms/step
Dropout Iteration 10
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 11
4/4 [==============================] - 0s 2ms/step
```

```
Dropout Iteration 12
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 13
4/4 [==============================] - 0s 2ms/step
Dropout Iteration 14
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 15
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 16
4/4 [==============================] - 0s 2ms/step
Dropout Iteration 17
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 18
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 19
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 20
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 21
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 22
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 23
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 24
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 25
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 26
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 27
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 28
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 29
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 30
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 31
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 32
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 33
4/4 [==============================] - 0s 2ms/step
Dropout Iteration 34
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 35
4/4 [==============================] - 0s 3ms/step
```

```
Dropout Iteration 36
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 37
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 38
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 39
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 40
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 41
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 42
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 43
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 44
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 45
4/4 [==============================] - 0s 5ms/step
Dropout Iteration 46
4/4 [==============================] - 0s 2ms/step
Dropout Iteration 47
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 48
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 49
4/4 [==============================] - 0s 3ms/step
Acquised Points added to training set
Epoch 1/20
57/57 [==============================] - 1s 10ms/step - loss: 2.9460 - accuracy:
0.9869 - val_loss: 2.9561 - val_accuracy: 0.9848
Epoch 2/20
57/57 [==============================] - 1s 10ms/step - loss: 2.9226 - accuracy:
0.9852 - val_loss: 3.0633 - val_accuracy: 0.9837
Epoch 3/20
57/57 [==============================] - 1s 9ms/step - loss: 2.9287 - accuracy:
0.9858 - val_loss: 2.9495 - val_accuracy: 0.9837
Epoch 4/20
57/57 [==============================] - 1s 10ms/step - loss: 2.9400 - accuracy:
0.9784 - val_loss: 2.9647 - val_accuracy: 0.9772
Epoch 5/20
57/57 [==============================] - 1s 9ms/step - loss: 2.9028 - accuracy:
0.9791 - val_loss: 2.9284 - val_accuracy: 0.9751
Epoch 6/20
57/57 [==============================] - 1s 10ms/step - loss: 2.8599 - accuracy:
0.9858 - val_loss: 2.8875 - val_accuracy: 0.9761
Epoch 7/20
```

```
57/57 [==============================] - 1s 9ms/step - loss: 2.8439 - accuracy:
0.9859 - val_loss: 2.8768 - val_accuracy: 0.9696
Epoch 8/20
57/57 [==============================] - 1s 9ms/step - loss: 2.8739 - accuracy:
0.9744 - val_loss: 2.9005 - val_accuracy: 0.9718
Epoch 9/20
57/57 [==============================] - 1s 9ms/step - loss: 2.8337 - accuracy:
0.9817 - val_loss: 2.8877 - val_accuracy: 0.9761
Epoch 10/20
57/57 [==============================] - 1s 9ms/step - loss: 2.8071 - accuracy:
0.9838 - val_loss: 2.8728 - val_accuracy: 0.9772
Epoch 11/20
57/57 [==============================] - 1s 10ms/step - loss: 2.7883 - accuracy:
0.9845 - val_loss: 2.7621 - val_accuracy: 0.9837
Epoch 12/20
57/57 [==============================] - 1s 9ms/step - loss: 2.7719 - accuracy:
0.9816 - val_loss: 2.8818 - val_accuracy: 0.9816
Epoch 13/20
57/57 [==============================] - 1s 10ms/step - loss: 2.7911 - accuracy:
0.9809 - val_loss: 2.7767 - val_accuracy: 0.9783
Epoch 14/20
57/57 [==============================] - 1s 9ms/step - loss: 2.7477 - accuracy:
0.9820 - val_loss: 2.7456 - val_accuracy: 0.9772
Epoch 15/20
57/57 [==============================] - 1s 9ms/step - loss: 2.7500 - accuracy:
0.9803 - val_loss: 2.7111 - val_accuracy: 0.9892
Epoch 16/20
57/57 [==============================] - 1s 10ms/step - loss: 2.7127 - accuracy:
0.9796 - val_loss: 2.8352 - val_accuracy: 0.9642
Epoch 17/20
57/57 [==============================] - 1s 10ms/step - loss: 2.6922 - accuracy:
0.9823 - val_loss: 2.6853 - val_accuracy: 0.9761
Epoch 18/20
57/57 [==============================] - 1s 9ms/step - loss: 2.6679 - accuracy:
0.9823 - val_loss: 2.7742 - val_accuracy: 0.9816
Epoch 19/20
57/57 [==============================] - 1s 9ms/step - loss: 2.6495 - accuracy:
0.9868 - val_loss: 2.7399 - val_accuracy: 0.9620
Epoch 20/20
57/57 [==============================] - 1s 9ms/step - loss: 2.6692 - accuracy:
0.9794 - val_loss: 2.7060 - val_accuracy: 0.9740
Evaluate Model Test Accuracy with pooled points
68/68 [==============================] - 0s 2ms/step - loss: 2.6236 - accuracy:
0.9856
Test score: 2.623633861541748
Test accuracy: 0.9856014847755432
Use this trained model with pooled points for Dropout again
Storing Accuracy Values over experiments
```

```
Experiment Number  2
X_train shape: (40, 7, 7, 80)
40 train samples
Training Model Without Acquisitions in Experiment 2
Epoch 1/20
57/57 [==============================] - 1s 12ms/step - loss: 2.6033 - accuracy:
0.9841 - val_loss: 2.6369 - val_accuracy: 0.9816
Epoch 2/20
57/57 [==============================] - 1s 11ms/step - loss: 2.7407 - accuracy:
0.9702 - val_loss: 2.6528 - val_accuracy: 0.9761
Epoch 3/20
57/57 [==============================] - 1s 10ms/step - loss: 2.6621 - accuracy:
0.9721 - val_loss: 2.7111 - val_accuracy: 0.9783
Epoch 4/20
57/57 [==============================] - 1s 12ms/step - loss: 2.6014 - accuracy:
0.9819 - val_loss: 2.7151 - val_accuracy: 0.9751
Epoch 5/20
57/57 [==============================] - 1s 14ms/step - loss: 2.5543 - accuracy:
0.9854 - val_loss: 2.5886 - val_accuracy: 0.9805
Epoch 6/20
57/57 [==============================] - 1s 14ms/step - loss: 2.5991 - accuracy:
0.9796 - val_loss: 2.6262 - val_accuracy: 0.9707
Epoch 7/20
57/57 [==============================] - 1s 12ms/step - loss: 2.5329 - accuracy:
0.9806 - val_loss: 2.5770 - val_accuracy: 0.9772
Epoch 8/20
57/57 [==============================] - 1s 12ms/step - loss: 2.5380 - accuracy:
0.9794 - val_loss: 2.5043 - val_accuracy: 0.9902
Epoch 9/20
57/57 [==============================] - 1s 14ms/step - loss: 2.5133 - accuracy:
0.9829 - val_loss: 2.5577 - val_accuracy: 0.9761
Epoch 10/20
57/57 [==============================] - 1s 13ms/step - loss: 2.5414 - accuracy:
0.9703 - val_loss: 2.5364 - val_accuracy: 0.9794
Epoch 11/20
57/57 [==============================] - 1s 13ms/step - loss: 2.5089 - accuracy:
0.9806 - val_loss: 2.4738 - val_accuracy: 0.9848
Epoch 12/20
57/57 [==============================] - 1s 13ms/step - loss: 2.4613 - accuracy:
0.9862 - val_loss: 2.4761 - val_accuracy: 0.9837
Epoch 13/20
57/57 [==============================] - 1s 13ms/step - loss: 2.4689 - accuracy:
0.9830 - val_loss: 2.5522 - val_accuracy: 0.9740
Epoch 14/20
57/57 [==============================] - 1s 13ms/step - loss: 2.4599 - accuracy:
0.9794 - val_loss: 2.5334 - val_accuracy: 0.9740
Epoch 15/20
57/57 [==============================] - 1s 14ms/step - loss: 2.4461 - accuracy:
```

```
0.9824 - val_loss: 2.5042 - val_accuracy: 0.9816
Epoch 16/20
57/57 [==============================] - 1s 16ms/step - loss: 2.4278 - accuracy:
0.9847 - val_loss: 2.4618 - val_accuracy: 0.9696
Epoch 17/20
57/57 [==============================] - 1s 16ms/step - loss: 2.4276 - accuracy:
0.9822 - val_loss: 2.4236 - val_accuracy: 0.9751
Epoch 18/20
57/57 [==============================] - 1s 13ms/step - loss: 2.4217 - accuracy:
0.9824 - val_loss: 2.4711 - val_accuracy: 0.9729
Epoch 19/20
57/57 [==============================] - 1s 13ms/step - loss: 2.4085 - accuracy:
0.9792 - val_loss: 2.3882 - val_accuracy: 0.9816
Epoch 20/20
57/57 [==============================] - 1s 15ms/step - loss: 2.3813 - accuracy:
0.9803 - val_loss: 2.5484 - val_accuracy: 0.9794
Evaluating Test Accuracy Without Acquisition
Starting Active Learning in Experiment  2
POOLING ITERATION 0
Dropout Iteration 0
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 1
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 2
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 3
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 4
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 5
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 6
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 7
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 8
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 9
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 10
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 11
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 12
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 13
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 14
```

```
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 15
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 16
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 17
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 18
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 19
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 20
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 21
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 22
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 23
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 24
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 25
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 26
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 27
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 28
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 29
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 30
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 31
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 32
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 33
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 34
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 35
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 36
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 37
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 38
```

```
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 39
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 40
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 41
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 42
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 43
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 44
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 45
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 46
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 47
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 48
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 49
4/4 [==============================] - 0s 3ms/step
Acquised Points added to training set
Epoch 1/20
57/57 [==============================] - 1s 13ms/step - loss: 2.3859 - accuracy:
0.9820 - val_loss: 2.4821 - val_accuracy: 0.9783
Epoch 2/20
57/57 [==============================] - 1s 13ms/step - loss: 2.3601 - accuracy:
0.9816 - val_loss: 2.4373 - val_accuracy: 0.9805
Epoch 3/20
57/57 [==============================] - 1s 12ms/step - loss: 2.3668 - accuracy:
0.9820 - val_loss: 2.4364 - val_accuracy: 0.9675
Epoch 4/20
57/57 [==============================] - 1s 13ms/step - loss: 2.3428 - accuracy:
0.9826 - val_loss: 2.3794 - val_accuracy: 0.9751
Epoch 5/20
57/57 [==============================] - 1s 13ms/step - loss: 2.3037 - accuracy:
0.9834 - val_loss: 2.3186 - val_accuracy: 0.9881
Epoch 6/20
57/57 [==============================] - 1s 13ms/step - loss: 2.3352 - accuracy:
0.9802 - val_loss: 2.3695 - val_accuracy: 0.9783
Epoch 7/20
57/57 [==============================] - 1s 12ms/step - loss: 2.2975 - accuracy:
0.9809 - val_loss: 2.3484 - val_accuracy: 0.9751
Epoch 8/20
57/57 [==============================] - 1s 12ms/step - loss: 2.2776 - accuracy:
0.9802 - val_loss: 2.3723 - val_accuracy: 0.9707
```

```
Epoch 9/20
57/57 [==============================] - 1s 12ms/step - loss: 2.2799 - accuracy:
0.9815 - val_loss: 2.3267 - val_accuracy: 0.9729
Epoch 10/20
57/57 [==============================] - 1s 12ms/step - loss: 2.2764 - accuracy:
0.9802 - val_loss: 2.2635 - val_accuracy: 0.9794
Epoch 11/20
57/57 [==============================] - 1s 12ms/step - loss: 2.2589 - accuracy:
0.9769 - val_loss: 2.2644 - val_accuracy: 0.9794
Epoch 12/20
57/57 [==============================] - 1s 13ms/step - loss: 2.2634 - accuracy:
0.9796 - val_loss: 2.3856 - val_accuracy: 0.9707
Epoch 13/20
57/57 [==============================] - 1s 12ms/step - loss: 2.2018 - accuracy:
0.9854 - val_loss: 2.2328 - val_accuracy: 0.9848
Epoch 14/20
57/57 [==============================] - 1s 13ms/step - loss: 2.2422 - accuracy:
0.9795 - val_loss: 2.2627 - val_accuracy: 0.9794
Epoch 15/20
57/57 [==============================] - 1s 12ms/step - loss: 2.1948 - accuracy:
0.9855 - val_loss: 2.3134 - val_accuracy: 0.9783
Epoch 16/20
57/57 [==============================] - 1s 12ms/step - loss: 2.1940 - accuracy:
0.9791 - val_loss: 2.1941 - val_accuracy: 0.9816
Epoch 17/20
57/57 [==============================] - 1s 12ms/step - loss: 2.1836 - accuracy:
0.9820 - val_loss: 2.3102 - val_accuracy: 0.9783
Epoch 18/20
57/57 [==============================] - 1s 13ms/step - loss: 2.1914 - accuracy:
0.9813 - val_loss: 2.2735 - val_accuracy: 0.9555
Epoch 19/20
57/57 [==============================] - 1s 12ms/step - loss: 2.2076 - accuracy:
0.9778 - val_loss: 2.2456 - val_accuracy: 0.9772
Epoch 20/20
57/57 [==============================] - 1s 13ms/step - loss: 2.1862 - accuracy:
0.9778 - val_loss: 2.1907 - val_accuracy: 0.9718
Evaluate Model Test Accuracy with pooled points
68/68 [==============================] - 0s 2ms/step - loss: 2.1713 - accuracy:
0.9749
Test score: 2.1712892055511475
Test accuracy: 0.9749187231063843
Use this trained model with pooled points for Dropout again
POOLING ITERATION 1
Dropout Iteration 0
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 1
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 2
```

```
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 3
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 4
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 5
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 6
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 7
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 8
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 9
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 10
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 11
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 12
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 13
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 14
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 15
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 16
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 17
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 18
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 19
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 20
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 21
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 22
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 23
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 24
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 25
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 26
```

```
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 27
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 28
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 29
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 30
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 31
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 32
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 33
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 34
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 35
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 36
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 37
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 38
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 39
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 40
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 41
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 42
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 43
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 44
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 45
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 46
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 47
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 48
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 49
4/4 [==============================] - 0s 4ms/step
Acquised Points added to training set
```

```
Epoch 1/20
57/57 [==============================] - 1s 14ms/step - loss: 2.1347 - accuracy:
0.9815 - val_loss: 2.2045 - val_accuracy: 0.9718
Epoch 2/20
57/57 [==============================] - 1s 12ms/step - loss: 2.1826 - accuracy:
0.9752 - val_loss: 2.1503 - val_accuracy: 0.9751
Epoch 3/20
57/57 [==============================] - 1s 12ms/step - loss: 2.1326 - accuracy:
0.9792 - val_loss: 2.1252 - val_accuracy: 0.9783
Epoch 4/20
57/57 [==============================] - 1s 12ms/step - loss: 2.0981 - accuracy:
0.9788 - val_loss: 2.1668 - val_accuracy: 0.9718
Epoch 5/20
57/57 [==============================] - 1s 13ms/step - loss: 2.1073 - accuracy:
0.9753 - val_loss: 2.0837 - val_accuracy: 0.9696
Epoch 6/20
57/57 [==============================] - 1s 12ms/step - loss: 2.1079 - accuracy:
0.9806 - val_loss: 2.0733 - val_accuracy: 0.9859
Epoch 7/20
57/57 [==============================] - 1s 12ms/step - loss: 2.0947 - accuracy:
0.9774 - val_loss: 2.1717 - val_accuracy: 0.9729
Epoch 8/20
57/57 [==============================] - 1s 13ms/step - loss: 2.1074 - accuracy:
0.9749 - val_loss: 2.0567 - val_accuracy: 0.9696
Epoch 9/20
57/57 [==============================] - 1s 13ms/step - loss: 2.1192 - accuracy:
0.9762 - val_loss: 2.0679 - val_accuracy: 0.9740
Epoch 10/20
57/57 [==============================] - 1s 13ms/step - loss: 2.0643 - accuracy:
0.9806 - val_loss: 2.2039 - val_accuracy: 0.9772
Epoch 11/20
57/57 [==============================] - 1s 12ms/step - loss: 2.0476 - accuracy:
0.9813 - val_loss: 2.1564 - val_accuracy: 0.9794
Epoch 12/20
57/57 [==============================] - 1s 13ms/step - loss: 2.0319 - accuracy:
0.9784 - val_loss: 2.0928 - val_accuracy: 0.9805
Epoch 13/20
57/57 [==============================] - 1s 12ms/step - loss: 2.0213 - accuracy:
0.9792 - val_loss: 2.3484 - val_accuracy: 0.9761
Epoch 14/20
57/57 [==============================] - 1s 12ms/step - loss: 2.0155 - accuracy:
0.9796 - val_loss: 2.0214 - val_accuracy: 0.9761
Epoch 15/20
57/57 [==============================] - 1s 13ms/step - loss: 2.0113 - accuracy:
0.9816 - val_loss: 2.0134 - val_accuracy: 0.9848
Epoch 16/20
57/57 [==============================] - 1s 12ms/step - loss: 1.9897 - accuracy:
0.9844 - val_loss: 2.1146 - val_accuracy: 0.9740
```

```
Epoch 17/20
57/57 [==============================] - 1s 12ms/step - loss: 1.9784 - accuracy:
0.9833 - val_loss: 1.9608 - val_accuracy: 0.9848
Epoch 18/20
57/57 [==============================] - 1s 13ms/step - loss: 1.9804 - accuracy:
0.9790 - val_loss: 1.9933 - val_accuracy: 0.9805
Epoch 19/20
57/57 [==============================] - 1s 12ms/step - loss: 1.9443 - accuracy:
0.9841 - val_loss: 1.9740 - val_accuracy: 0.9794
Epoch 20/20
57/57 [==============================] - 1s 12ms/step - loss: 1.9659 - accuracy:
0.9798 - val_loss: 1.9687 - val_accuracy: 0.9772
Evaluate Model Test Accuracy with pooled points
68/68 [==============================] - 0s 2ms/step - loss: 1.9715 - accuracy:
0.9782
Test score: 1.9714635610580444
Test accuracy: 0.9781699776649475
Use this trained model with pooled points for Dropout again
POOLING ITERATION 2
Dropout Iteration 0
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 1
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 2
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 3
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 4
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 5
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 6
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 7
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 8
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 9
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 10
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 11
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 12
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 13
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 14
```

```
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 15
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 16
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 17
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 18
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 19
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 20
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 21
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 22
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 23
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 24
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 25
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 26
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 27
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 28
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 29
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 30
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 31
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 32
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 33
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 34
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 35
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 36
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 37
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 38
```

```
4/4 [==============================] - 0s 4ms/step
Dropout Iteration 39
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 40
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 41
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 42
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 43
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 44
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 45
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 46
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 47
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 48
4/4 [==============================] - 0s 3ms/step
Dropout Iteration 49
4/4 [==============================] - 0s 3ms/step
Acquised Points added to training set
Epoch 1/20
57/57 [==============================] - 1s 13ms/step - loss: 2.0499 - accuracy:
0.9745 - val_loss: 2.0217 - val_accuracy: 0.9718
Epoch 2/20
57/57 [==============================] - 1s 12ms/step - loss: 1.9752 - accuracy:
0.9756 - val_loss: 1.9899 - val_accuracy: 0.9620
Epoch 3/20
57/57 [==============================] - 1s 12ms/step - loss: 1.9337 - accuracy:
0.9790 - val_loss: 2.0001 - val_accuracy: 0.9696
Epoch 4/20
57/57 [==============================] - 1s 13ms/step - loss: 1.9327 - accuracy:
0.9799 - val_loss: 1.9565 - val_accuracy: 0.9740
Epoch 5/20
57/57 [==============================] - 1s 12ms/step - loss: 1.9059 - accuracy:
0.9829 - val_loss: 1.8899 - val_accuracy: 0.9870
Epoch 6/20
57/57 [==============================] - 1s 12ms/step - loss: 1.9248 - accuracy:
0.9834 - val_loss: 1.9545 - val_accuracy: 0.9707
Epoch 7/20
57/57 [==============================] - 1s 13ms/step - loss: 1.8938 - accuracy:
0.9826 - val_loss: 1.9087 - val_accuracy: 0.9826
Epoch 8/20
57/57 [==============================] - 1s 12ms/step - loss: 1.8891 - accuracy:
0.9796 - val_loss: 1.8815 - val_accuracy: 0.9805
```

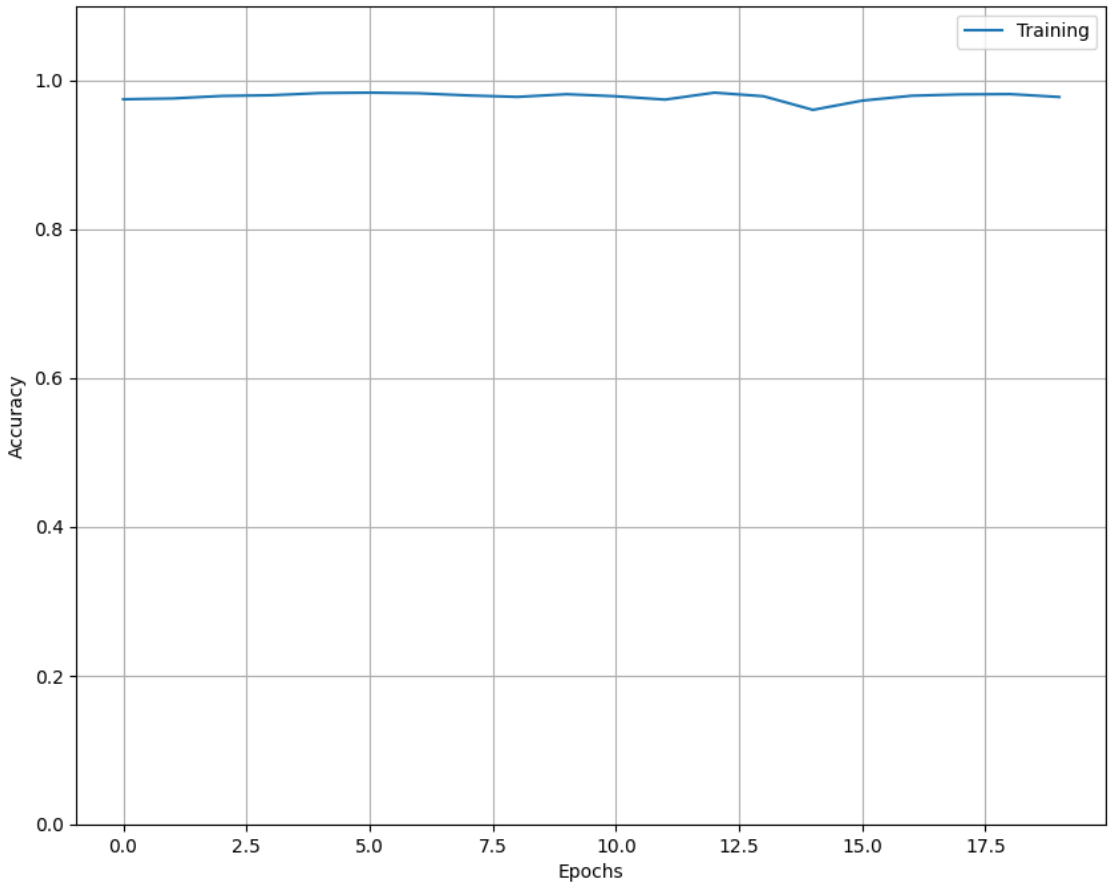
```
Epoch 9/20
57/57 [==============================] - 1s 12ms/step - loss: 1.9008 - accuracy:
0.9777 - val_loss: 2.0556 - val_accuracy: 0.9653
Epoch 10/20
57/57 [==============================] - 1s 13ms/step - loss: 1.8731 - accuracy:
0.9813 - val_loss: 1.9135 - val_accuracy: 0.9729
Epoch 11/20
57/57 [==============================] - 1s 12ms/step - loss: 1.8945 - accuracy:
0.9785 - val_loss: 1.9662 - val_accuracy: 0.9610
Epoch 12/20
57/57 [==============================] - 1s 12ms/step - loss: 1.9023 - accuracy:
0.9741 - val_loss: 1.8885 - val_accuracy: 0.9783
Epoch 13/20
57/57 [==============================] - 1s 12ms/step - loss: 1.8525 - accuracy:
0.9834 - val_loss: 1.8499 - val_accuracy: 0.9794
Epoch 14/20
57/57 [==============================] - 1s 12ms/step - loss: 1.8556 - accuracy:
0.9785 - val_loss: 1.9183 - val_accuracy: 0.9631
Epoch 15/20
57/57 [==============================] - 1s 12ms/step - loss: 2.0090 - accuracy:
0.9603 - val_loss: 1.9392 - val_accuracy: 0.9761
Epoch 16/20
57/57 [==============================] - 1s 13ms/step - loss: 1.8534 - accuracy:
0.9727 - val_loss: 1.9033 - val_accuracy: 0.9783
Epoch 17/20
57/57 [==============================] - 1s 12ms/step - loss: 1.8198 - accuracy:
0.9792 - val_loss: 1.8648 - val_accuracy: 0.9783
Epoch 18/20
57/57 [==============================] - 1s 13ms/step - loss: 1.8138 - accuracy:
0.9810 - val_loss: 1.8288 - val_accuracy: 0.9794
Epoch 19/20
57/57 [==============================] - 1s 13ms/step - loss: 1.8299 - accuracy:
0.9815 - val_loss: 1.8943 - val_accuracy: 0.9707
Epoch 20/20
57/57 [==============================] - 1s 12ms/step - loss: 1.8350 - accuracy:
0.9776 - val_loss: 1.8630 - val_accuracy: 0.9751
Evaluate Model Test Accuracy with pooled points
68/68 [==============================] - 0s 2ms/step - loss: 1.7971 - accuracy:
0.9754
Test score: 1.7970836162567139
Test accuracy: 0.9753831624984741
Use this trained model with pooled points for Dropout again
Storing Accuracy Values over experiments
```

[ ]:
```
Average_Accuracy = np.average(Experiments_All_Accuracy)
print(Average_Accuracy)
```

```
#np.save('/home/ri258/Documents/Project/MPhil_Thesis_Cluster_Experiments/
 ↪ConvNets/Cluster_Experiments/Final_Experiments/Results/
 ↪'+'Dropout_Bald_Q10_N1000_Average_Accuracy'+'.npy', Average_Accuracy)
```

```
0.9775893986225128
```

[ ]:
```
# Plot accuracy
plt.figure(figsize=(10,8))
plt.ylim(0,1.1)
plt.grid()
plt.plot(hist.history['accuracy'])
plt.ylabel('Accuracy')
plt.xlabel('Epochs')
plt.legend(['Training','Validation'])
plt.savefig("acc_curve.pdf")
plt.show()
```



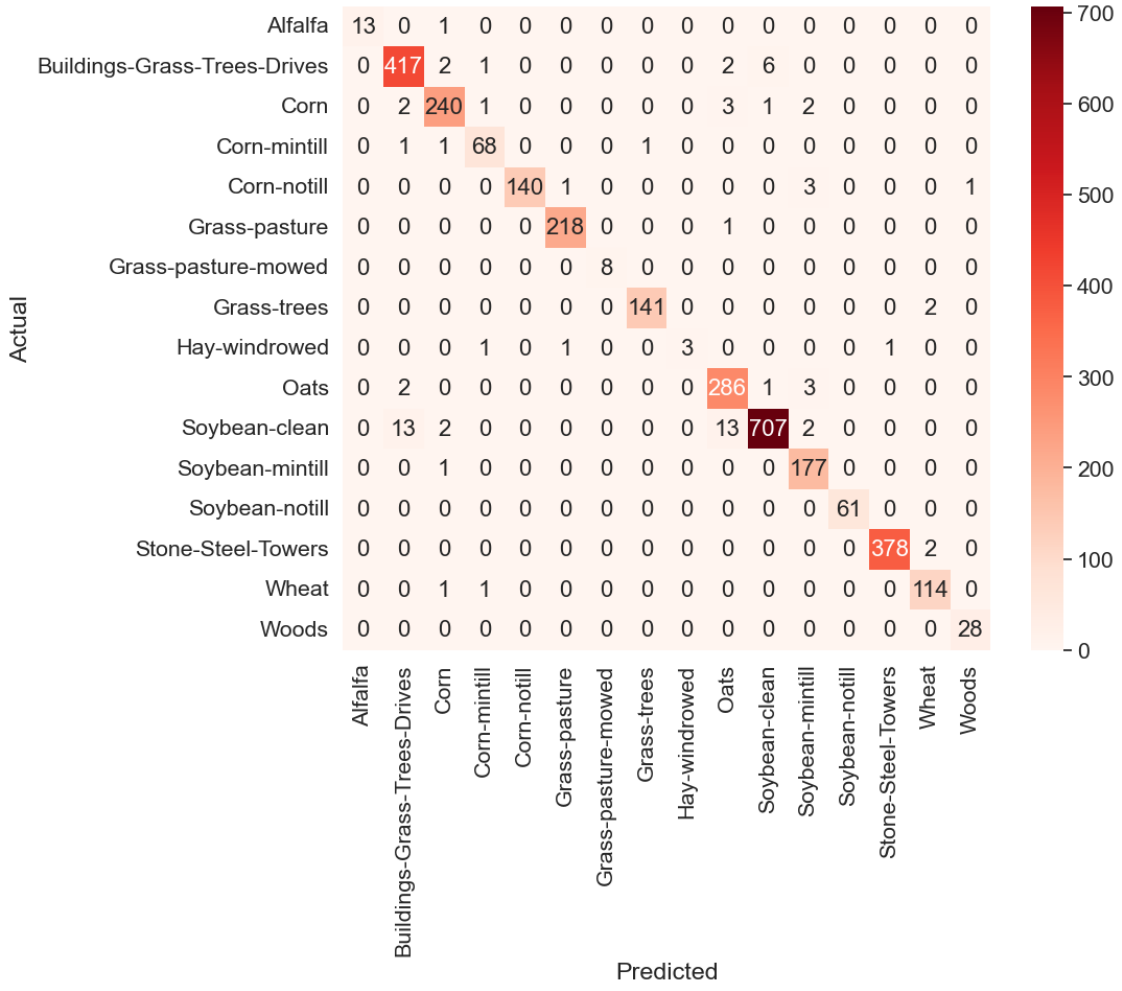
[ ]:
```
# 16 classes
```

```
names = ['Alfalfa', 'Corn-notill', 'Corn-mintill', 'Corn', 'Grass-pasture',␣
 ↪'Grass-trees',
        'Grass-pasture-mowed', 'Hay-windrowed', 'Oats', 'Soybean-notill',␣
 ↪'Soybean-mintill',
          'Soybean-clean', 'Wheat', 'Woods', 'Buildings-Grass-Trees-Drives',␣
 ↪'Stone-Steel-Towers']
```

[ ]:
```
from sklearn.metrics import confusion_matrix, accuracy_score,␣
 ↪classification_report, cohen_kappa_score
import pandas as pd
import numpy as np
import seaborn as sn
# confusion_matrix
Y_pred = model_bayes.predict(Xtest0)
y_pred = np.argmax(Y_pred, axis=1)

confusion = confusion_matrix(ytest0, y_pred)
df_cm = pd.DataFrame(confusion, columns=np.unique(names), index = np.
 ↪unique(names))
df_cm.index.name = 'Actual'
df_cm.columns.name = 'Predicted'
plt.figure(figsize = (10,8))
sn.set(font_scale=1.4)#for label size
sn.heatmap(df_cm, cmap="Reds", annot=True,annot_kws={"size": 16}, fmt='d')
plt.savefig('cmap.png', dpi=300)
```

```
97/97 [==============================] - 0s 2ms/step
```



[ ]:
```
# average_acc
from operator import itemgetter
def AA_andEachClassAccuracy(confusion_matrix):
    counter = confusion_matrix.shape[0]
    list_diag = np.diag(confusion_matrix)
    list_raw_sum = np.sum(confusion_matrix, axis=1)
    each_acc = np.nan_to_num((list_diag/ list_raw_sum))
    average_acc = np.mean(each_acc)
    return each_acc, average_acc
```

[ ]:
```
# average_acc

each_acc, aa = AA_andEachClassAccuracy(confusion)
print("accuracy for each:")
print (each_acc)
```

```
print("OA accuracy:")
print(aa)
```

```
accuracy for each:
[0.92857143 0.97429907 0.96385542 0.95774648 0.96551724 0.99543379
 1.         0.98601399 0.5        0.97945205 0.95929444 0.99438202
 1.         0.99473684 0.98275862 1.        ]
OA accuracy:
0.9488788368042085
```

[ ]:
```
# classification_report
print(classification_report(ytest0, y_pred, target_names = names, digits = 3))
```

```
                            precision    recall  f1-score   support

                   Alfalfa      1.000     0.929     0.963        14
                Corn-notill      0.959     0.974     0.966       428
               Corn-mintill      0.968     0.964     0.966       249
                       Corn      0.944     0.958     0.951        71
              Grass-pasture      1.000     0.966     0.982       145
                Grass-trees      0.991     0.995     0.993       219
        Grass-pasture-mowed      1.000     1.000     1.000         8
              Hay-windrowed      0.993     0.986     0.989       143
                       Oats      1.000     0.500     0.667         6
             Soybean-notill      0.938     0.979     0.958       292
            Soybean-mintill      0.989     0.959     0.974       737
              Soybean-clean      0.947     0.994     0.970       178
                      Wheat      1.000     1.000     1.000        61
                      Woods      0.997     0.995     0.996       380
Buildings-Grass-Trees-Drives      0.966     0.983     0.974       116
          Stone-Steel-Towers      0.966     1.000     0.982        28

                    accuracy                          0.975      3075
                   macro avg      0.979     0.949     0.958      3075
                weighted avg      0.976     0.975     0.975      3075
```