

CNN model used for first part of Experiment 1

CNNmodel_1

April 28, 2021

```
[ ]: !pip install tensorflow-addons
```

```
Collecting tensorflow-addons
  Downloading https://files.pythonhosted.org/packages/74/e3/56d2fe76f0bb7c
  88ed9b2a6a557e25e83e252aec08f13de34369cd850a0b/tensorflow_addons-0.12.1-cp37-cp3
  7m-manylinux2010_x86_64.whl (703kB)
    || 706kB 15.7MB/s
Requirement already satisfied: typeguard>=2.7 in
/usr/local/lib/python3.7/dist-packages (from tensorflow-addons) (2.7.1)
Installing collected packages: tensorflow-addons
Successfully installed tensorflow-addons-0.12.1
```

```
[ ]: import keras
import seaborn as sn
import pandas as pd
import sklearn.metrics as metrics
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import image
from keras.preprocessing.image import ImageDataGenerator
import tensorflow as tf
from keras.models import Sequential
from keras.models import load_model
from keras.optimizers import Adam
from keras.initializers import TruncatedNormal
from keras.layers import Input, Dense, Dropout, Flatten, Conv2D, MaxPooling2D
from keras.callbacks import ReduceLROnPlateau, EarlyStopping, TensorBoard
from sklearn.metrics import roc_curve, auc
from sklearn.utils import class_weight
import os
from keras.layers.normalization import BatchNormalization
from keras.layers import Dropout
import tensorflow_addons as tfa
from keras.callbacks import ReduceLROnPlateau, EarlyStopping
from sklearn.metrics import roc_curve
from sklearn.metrics import auc
```

```
[ ]: # im=image.imread('/content/drive/MyDrive/MLPH582/finalProject/test/merger/
→1237648721763041373.jpeg')

[ ]: # im.shape

[ ]: # plt.imshow(im)

[ ]: from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
[ ]: train_path = '/content/drive/MyDrive/MLPH582/finalProject/train1'
valid_path = '/content/drive/MyDrive/MLPH582/finalProject/valid1'
test_path = '/content/drive/MyDrive/MLPH582/finalProject/test1'

[ ]: train_datagen = ImageDataGenerator(rescale = 1/255., rotation_range= 40,
→width_shift_range= 0.3, height_shift_range= 0.3, zoom_range= 0.3)

valid_datagen = ImageDataGenerator(rescale= 1/255.)

test_datagen = ImageDataGenerator(rescale= 1/255.)

train_BATCH = 100
valid_BATCH = 100
test_BATCH = 20

train_generator = train_datagen.flow_from_directory(train_path, classes =
→['MergerGalaxies', 'NormalGalaxiesAllInclination'], batch_size =
→train_BATCH, class_mode = 'binary')
valid_generator = valid_datagen.flow_from_directory(valid_path, classes =
→['MergerGalaxies', 'NormalGalaxiesAllInclination'], batch_size =
→valid_BATCH, class_mode = 'binary')
test_generator = test_datagen.flow_from_directory(test_path, classes =
→['MergerGalaxies', 'NormalGalaxiesAllInclination'], batch_size = test_BATCH,
→class_mode = 'binary', shuffle=False)
```

Found 3000 images belonging to 2 classes.
 Found 400 images belonging to 2 classes.
 Found 200 images belonging to 2 classes.

```
[ ]: # data augmentation of train dataset

train_generator_1 = train_datagen.flow_from_directory(train_path, classes =
→['MergerGalaxies', 'NormalGalaxiesLowInclination'], batch_size = 1,
→class_mode = 'binary')

fig = plt.gcf()
```

```

fig.set_size_inches(15, 15)

# fig, ax = plt.subplots(nrows=2, ncols=4, figsize=(15,15))
for i in range(10):

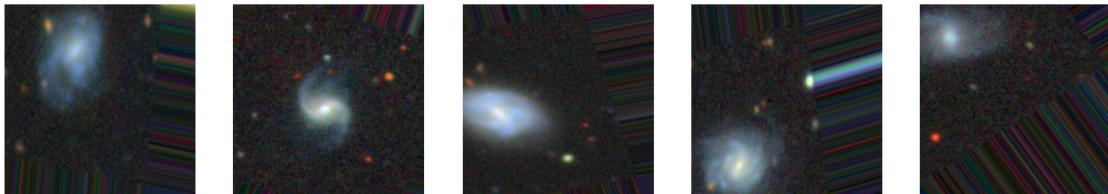
    # convert to unsigned integers for plotting
    image = next(train_generator_1)[0]

    # changing size from (1, 256, 256, 3) to (256, 256, 3) for plotting the image
    image = np.squeeze(image)

    sp = plt.subplot(2, 5 , i+1) #subplot indices start at 1
    sp.axis('Off')
    plt.imshow(np.asarray(image))

```

Found 3000 images belonging to 2 classes.



```
[ ]: print(train_generator.class_indices)
print(valid_generator.class_indices)
print(test_generator.class_indices)
```

```
{'MergerGalaxies': 0, 'NormalGalaxiesAllInclination': 1}  
{'MergerGalaxies': 0, 'NormalGalaxiesAllInclination': 1}  
{'MergerGalaxies': 0, 'NormalGalaxiesAllInclination': 1}
```

```
[ ]: # model
```

```
FILTERS = 128  
ACTIVATION = 'relu'  
KERNEL_SIZE = 5  
INITIALIZER = 'he_normal'  
convLayers = [64, 32]  
DenseLayers = [64, 32]  
POOL_SIZE = 5  
dropout = 0.5  
LR_INIT = 0.001  
  
def build_model():  
    model = Sequential()  
  
    model.add(Conv2D(filters = FILTERS, activation = ACTIVATION, kernel_size = KERNEL_SIZE, padding='same', kernel_initializer = INITIALIZER , input_shape = (256,256,3)))  
    model.add(MaxPooling2D(pool_size= POOL_SIZE))  
  
    for conv in convLayers:  
        model.add(Conv2D(conv, activation = ACTIVATION, kernel_size= KERNEL_SIZE, padding = 'same', kernel_initializer=INITIALIZER))  
        model.add(MaxPooling2D(pool_size= POOL_SIZE))  
  
    model.add(Flatten())  
  
    for dense in DenseLayers:  
        model.add(Dense(dense, activation= ACTIVATION, kernel_initializer = INITIALIZER))  
        model.add(Dropout(dropout))  
  
    model.add(Dense(1, activation='sigmoid'))  
    model.compile(loss=tfa.losses.SigmoidFocalCrossEntropy(), optimizer=Adam(lr=LR_INIT), metrics= [tf.keras.metrics.AUC(name='auc')])  
    # model.compile(loss= tf.keras.losses.BinaryCrossentropy(),  
    # optimizer=Adam(lr= LR_INIT), metrics= [tf.keras.metrics.AUC(name='auc')])  
  
    return model
```

```
[ ]: model = build_model()  
model.summary()
```

```
Model: "sequential_2"
```

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 256, 256, 128)	9728
max_pooling2d_6 (MaxPooling2D)	(None, 51, 51, 128)	0
conv2d_7 (Conv2D)	(None, 51, 51, 64)	204864
max_pooling2d_7 (MaxPooling2D)	(None, 10, 10, 64)	0
conv2d_8 (Conv2D)	(None, 10, 10, 32)	51232
max_pooling2d_8 (MaxPooling2D)	(None, 2, 2, 32)	0
flatten_2 (Flatten)	(None, 128)	0
dense_6 (Dense)	(None, 64)	8256
dropout_4 (Dropout)	(None, 64)	0
dense_7 (Dense)	(None, 32)	2080
dropout_5 (Dropout)	(None, 32)	0
dense_8 (Dense)	(None, 1)	33

Total params: 276,193
Trainable params: 276,193
Non-trainable params: 0

```
[ ]: # import os
# root_logdir = os.path.join('/content/drive/MyDrive/MLPH582/finalProject',
#                            'my_logs')
# ExperimentName = 'Expt1_02'
```

```
# def get_run_logdir():
#     import time
#     run_id = time.strftime("run_%Y_%m_%d-%H_%M_%S_")
#     return os.path.join(root_logdir, ExperimentName)
```

```
[ ]: # weighing classes
class_weights = class_weight.compute_class_weight('balanced', np.unique(train_generator.classes), train_generator.classes)
class_weight_dict = {0:class_weights[0], 1:class_weights[1]}
```

```

[ ]: class_weight_dict
[ ]: {0: 3.0, 1: 0.6}

[ ]: # tensorboard_cb = TensorBoard(get_run_logdir())
    Reducelr_cb = ReduceLROnPlateau(monitor='val_auc', factor =0.5, patience = 3, u
        ↪verbose=1, mode='max', min_lr = 1e-8)
    early_cb = EarlyStopping(monitor='val_auc', patience=5, mode='max')
    # filepath = '/content/drive/MyDrive/MLPH582/finalProject/model.{epoch:
        ↪02d}-{auc:.2f}.h5'
    # checkpoint_cb = tf.keras.callbacks.ModelCheckpoint(filepath, monitor = 'auc', u
        ↪save_best_only = True, verbose = 1, mode = 'max' )
    EPOCHS_NUM = 20
    history = model.fit(train_generator,
        epochs= EPOCHS_NUM,
        class_weight = class_weight_dict,
        validation_data = valid_generator,
        verbose=1, callbacks = [Reducelr_cb, early_cb])

    # model.save('/content/drive/MyDrive/MLPH582/finalProject/model_Expt1.h5')

```

```

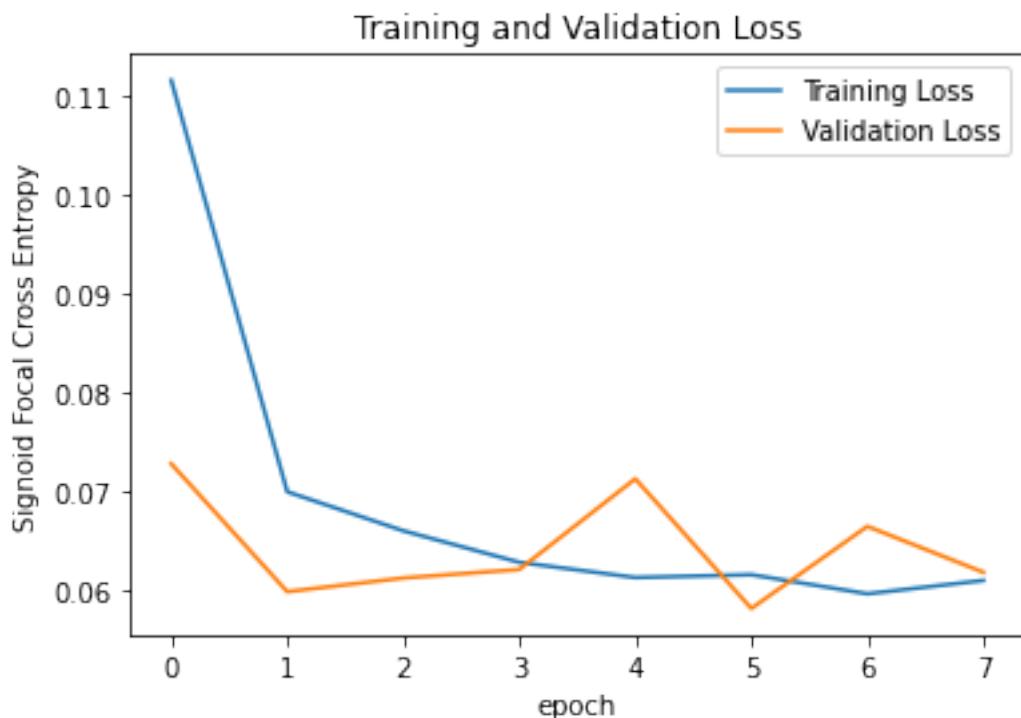
Epoch 1/20
30/30 [=====] - 49s 2s/step - loss: 0.1717 - auc: 0.5678 - val_loss: 0.0728 - val_auc: 0.7468
Epoch 2/20
30/30 [=====] - 47s 2s/step - loss: 0.0707 - auc: 0.6030 - val_loss: 0.0599 - val_auc: 0.7976
Epoch 3/20
30/30 [=====] - 47s 2s/step - loss: 0.0653 - auc: 0.7155 - val_loss: 0.0613 - val_auc: 0.7999
Epoch 4/20
30/30 [=====] - 47s 2s/step - loss: 0.0626 - auc: 0.7597 - val_loss: 0.0621 - val_auc: 0.7821
Epoch 5/20
30/30 [=====] - 47s 2s/step - loss: 0.0601 - auc: 0.8093 - val_loss: 0.0713 - val_auc: 0.7424
Epoch 6/20
30/30 [=====] - 47s 2s/step - loss: 0.0607 - auc: 0.7752 - val_loss: 0.0582 - val_auc: 0.7965

Epoch 00006: ReduceLROnPlateau reducing learning rate to 0.0005000000237487257.
Epoch 7/20
30/30 [=====] - 47s 2s/step - loss: 0.0596 - auc: 0.7995 - val_loss: 0.0665 - val_auc: 0.7502
Epoch 8/20
30/30 [=====] - 47s 2s/step - loss: 0.0611 - auc: 0.7944 - val_loss: 0.0618 - val_auc: 0.7763

```

```
[ ]: # learning curve
loss = history.history['loss']
val_loss = history.history['val_loss']

plt.plot(loss, label='Training Loss')
plt.plot(val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.ylabel('Sigmoid Focal Cross Entropy')
# plt.ylim([0,0.15])
plt.title('Training and Validation Loss')
plt.xlabel('epoch')
plt.show()
# plt.savefig('/content/drive/MyDrive/MLPH582/finalProject/
→LearningCurve_expt1_02.png')
```



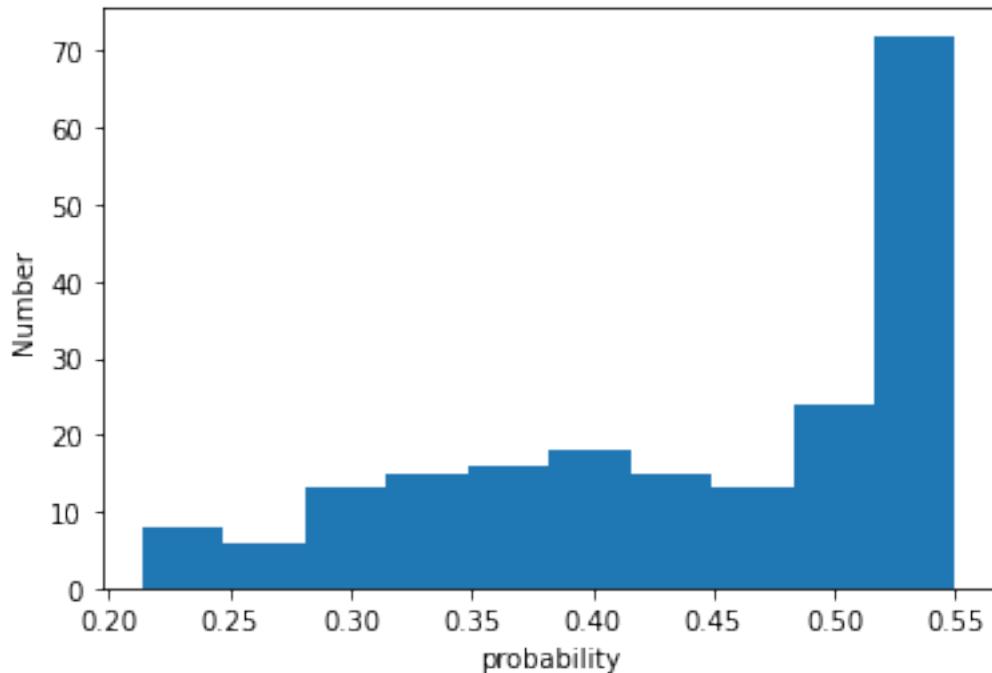
Due to the stochastic nature of the training process, I get a different model each time (I forgot to set the seed). However, the model I refer to in the report and presentation was saved to my drive and can be loaded to reproduce my results.

```
[ ]: # load model
# model_Expt1 = load_model('/content/drive/MyDrive/MLPH582/finalProject/
→model_Expt1.h5')
```

```
[ ]: # predict on test
predictions = model.predict(test_generator, verbose =1)
```

```
10/10 [=====] - 1s 74ms/step
```

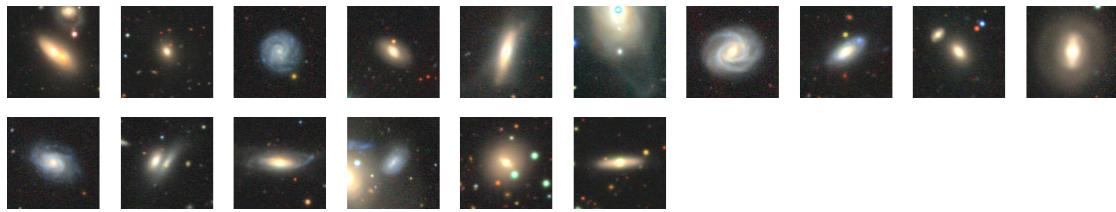
```
[ ]: plt.hist(predictions)
plt.xlabel('probability')
plt.ylabel('Number')
plt.show()
# plt.savefig('/content/drive/MyDrive/MLPH582/finalProject/
→TestPredictions_expt1.png')
```



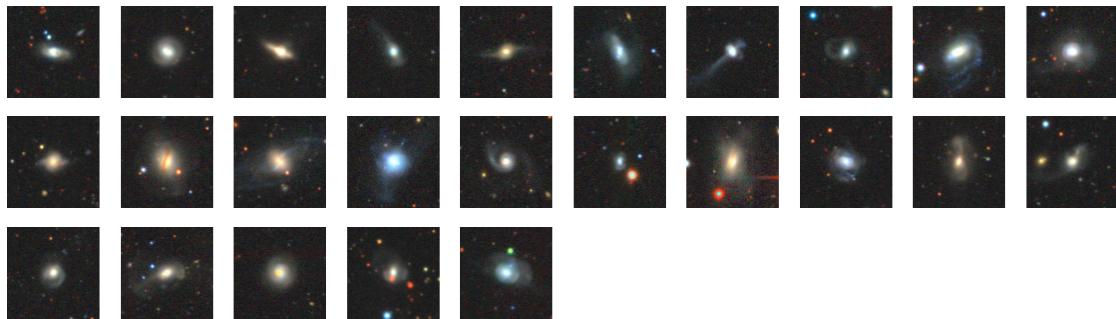
```
[ ]: predictedClasses = np.where(predictions>0.45, 1, 0)
[ ]: predictedClass_=predictedClasses.ravel()
[ ]: normal_classified_normal = []
for idx in range(len(predictedClass_)):
    if (test_generator.classes[idx]==1) & (predictedClass_[idx]==1):
        normal_classified_normal.append(idx)
[ ]: merger_classified_merger = []
for idx in range(len(predictedClass_)):
    if (test_generator.classes[idx]==0) & (predictedClass_[idx]==0):
        merger_classified_merger.append(idx)
[ ]: normal_classified_merger = []
for idx in range(len(predictedClass_)):
    if (test_generator.classes[idx]==1) & (predictedClass_[idx]==0):
        normal_classified_merger.append(idx)
```

```
[ ]: merger_classified_normal = []
for idx in range(len(predictedClass_)):
    if (test_generator.classes[idx]==0) & (predictedClass_[idx]==1):
        merger_classified_normal.append(idx)

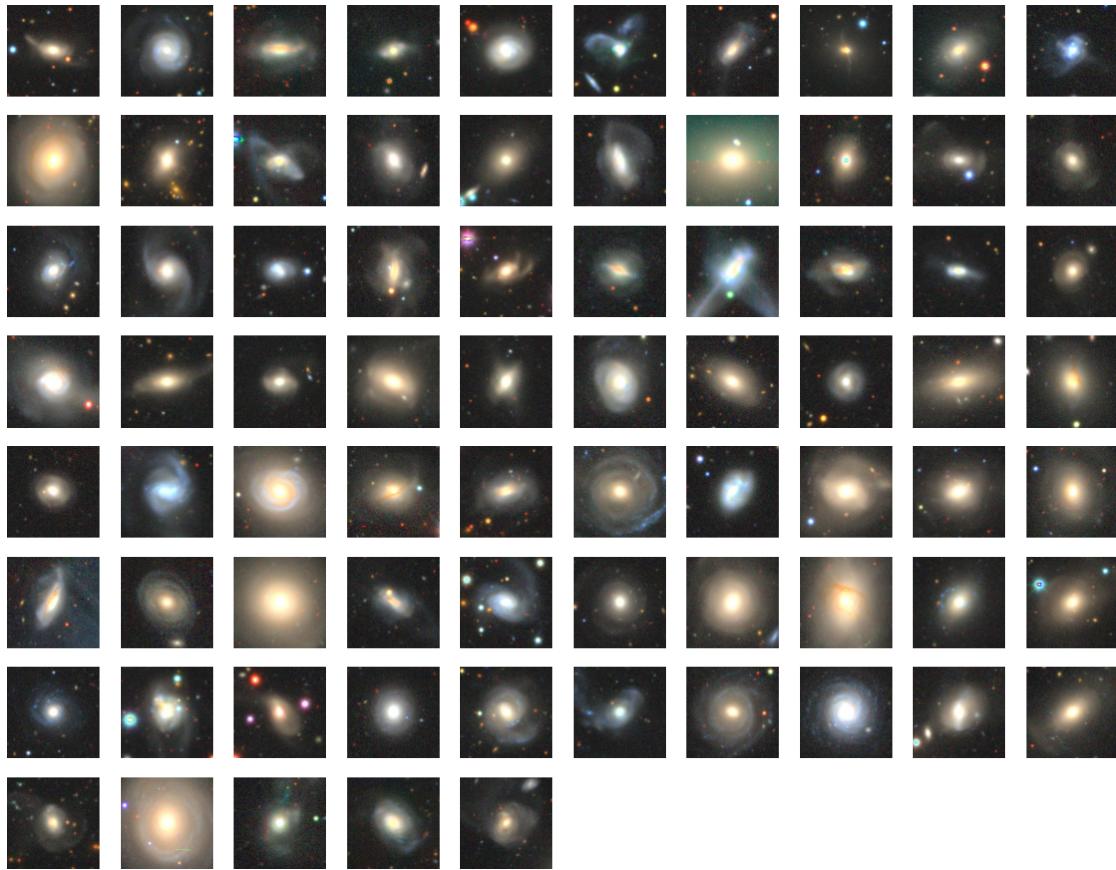
[ ]: # NORMAL CLASSIFIED AS MERGER
fig = plt.gcf()
fig.set_size_inches(20, 20)
nrows = 10
ncols = 10
fnames = test_generator.filenames
for i, idx in enumerate(normal_classified_merger):
    sp = plt.subplot(nrows, ncols, i+1)
    sp.axis('Off')
    img = image.imread(os.path.join('/content/drive/MyDrive/MLPH582/finalProject/' + test1, fnames[idx]))
    plt.imshow(img)
```



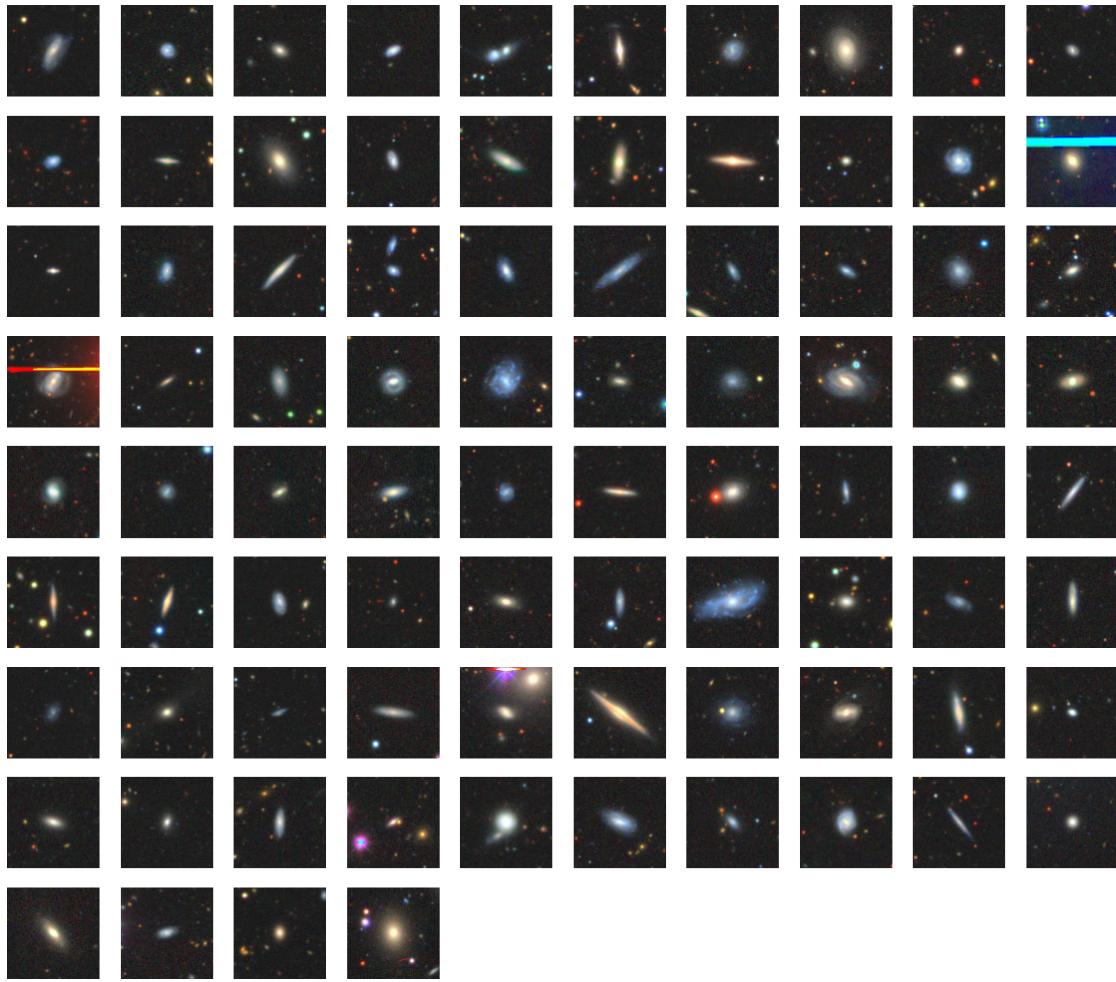
```
[ ]: # MERGER CLASSIFIED AS NORMAL
fig = plt.gcf()
fig.set_size_inches(20, 20)
nrows = 10
ncols = 10
fnames = test_generator.filenames
for i, idx in enumerate(merger_classified_normal):
    sp = plt.subplot(nrows, ncols, i+1)
    sp.axis('Off')
    img = image.imread(os.path.join('/content/drive/MyDrive/MLPH582/finalProject/' + test1, fnames[idx]))
    plt.imshow(img)
```



```
[ ]: # MERGER CLASSIFIED AS MERGER
fig = plt.gcf()
fig.set_size_inches(20, 20)
nrows = 10
ncols = 10
fnames = test_generator.filenames
for i, idx in enumerate(merger_classified_merger):
    sp = plt.subplot(nrows, ncols, i+1)
    sp.axis('Off')
    img = image.imread(os.path.join('/content/drive/MyDrive/MLPH582/finalProject/
→test1', fnames[idx]))
    plt.imshow(img)
```

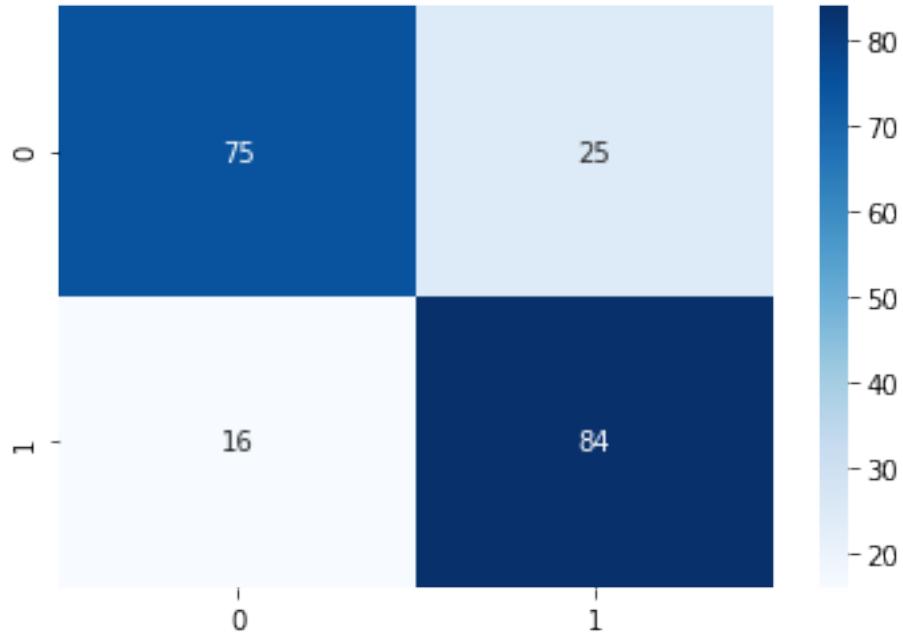


```
[ ]: # NORMAL CLASSIFIED AS NORMAL
fig = plt.gcf()
fig.set_size_inches(20, 20)
nrows = 10
ncols = 10
fnames = test_generator.filenames
for i, idx in enumerate(normal_classified_normal):
    sp = plt.subplot(nrows, ncols, i+1)
    sp.axis('Off')
    img = image.imread(os.path.join('/content/drive/MyDrive/MLPH582/finalProject/
→test1', fnames[idx]))
    plt.imshow(img)
```



```
[ ]: #Confusion Matrix
cm = metrics.confusion_matrix(test_generator.classes, predictedClasses)
df_cm = pd.DataFrame(cm, index = ['0', '1'], columns=['0','1'])
sn.heatmap(df_cm, annot=True, cmap = plt.cm.Blues)
plt.show()
# plt.savefig('/content/drive/MyDrive/MLPH582/finalProject/
→ConfusionMatrix_expt1.png')

cm_report = metrics.classification_report(test_generator.classes, predictedClasses)
print (cm_report)
```

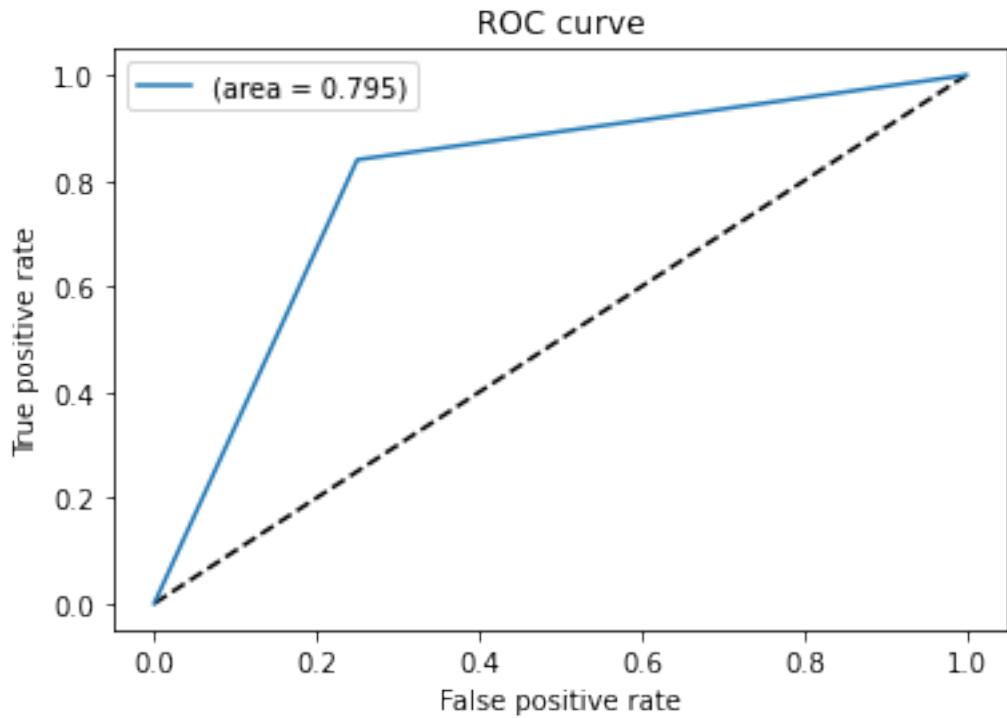


	precision	recall	f1-score	support
0	0.82	0.75	0.79	100
1	0.77	0.84	0.80	100
accuracy			0.80	200
macro avg	0.80	0.79	0.79	200
weighted avg	0.80	0.80	0.79	200

```
[ ]: # ROC-AUC curve
fpr, tpr, threshold = roc_curve(test_generator.classes, predictedClasses)

# AUC
auc_ = auc(fpr, tpr)

# Plot
plt.plot([0, 1], [0, 1], 'k--')
plt.plot(fpr, tpr, label='(area = {:.3f})'.format(auc_))
plt.xlabel('False positive rate')
plt.ylabel('True positive rate')
plt.title('ROC curve')
plt.legend(loc='best')
plt.show()
# plt.savefig('/content/drive/MyDrive/MLPH582/finalProject/ROC_AUC_expt1_02.
→png')
```



CNN used for second part of Experiment 1

CNNmodel

April 28, 2021

```
[1]: !pip install tensorflow-addons
```

```
Requirement already satisfied: tensorflow-addons in  
/usr/local/lib/python3.7/dist-packages (0.12.1)  
Requirement already satisfied: typeguard>=2.7 in /usr/local/lib/python3.7/dist-  
packages (from tensorflow-addons) (2.7.1)
```

```
[2]: import keras  
import sklearn.metrics as metrics  
import matplotlib.pyplot as plt  
import numpy as np  
from matplotlib import image  
from keras.preprocessing.image import ImageDataGenerator  
import tensorflow as tf  
from keras.models import Sequential  
from keras.optimizers import Adam  
from keras.initializers import TruncatedNormal  
from keras.layers import Input, Dense, Dropout, Flatten, Conv2D, MaxPooling2D  
from keras.callbacks import ReduceLROnPlateau, EarlyStopping, TensorBoard  
from sklearn.metrics import roc_curve, auc  
from sklearn.utils import class_weight  
import os  
from keras.layers.normalization import BatchNormalization  
from keras.layers import Dropout  
import tensorflow_addons as tfa  
from keras.callbacks import ReduceLROnPlateau, EarlyStopping  
from sklearn.metrics import roc_curve  
from sklearn.metrics import auc
```

```
[ ]: # im=image.imread('/content/drive/MyDrive/MLPH582/finalProject/test/merger/  
↪1237648721763041373.jpeg')
```

```
[ ]: # im.shape
```

```
[ ]: # plt.imshow(im)
```

```
[3]: from google.colab import drive  
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call  
drive.mount("/content/drive", force_remount=True).
```

```
[4]: train_path = '/content/drive/MyDrive/finalProject/train1'  
      valid_path = '/content/drive/MyDrive/finalProject/valid1'  
      test_path = '/content/drive/MyDrive/finalProject/test1'
```

```
[5]: train_datagen = ImageDataGenerator(rescale = 1/255., rotation_range= 10,  
                                      width_shift_range= 0.1, height_shift_range= 0.1, zoom_range= 0.1)  
  
      valid_datagen = ImageDataGenerator(rescale= 1/255.)  
  
      test_datagen = ImageDataGenerator(rescale= 1/255.)  
  
      train_BATCH = 100  
      valid_BATCH = 100  
      test_BATCH = 20  
  
      train_generator = train_datagen.flow_from_directory(train_path, classes =  
                                                       ['MergerGalaxies', 'NormalGalaxiesLowInclination'], batch_size =  
                                                       train_BATCH, class_mode = 'binary')  
      valid_generator = valid_datagen.flow_from_directory(valid_path, classes =  
                                                       ['MergerGalaxies', 'NormalGalaxiesLowInclination'], batch_size =  
                                                       valid_BATCH, class_mode = 'binary')  
      test_generator = test_datagen.flow_from_directory(test_path, classes =  
                                                       ['MergerGalaxies', 'NormalGalaxiesLowInclination'], batch_size = test_BATCH,  
                                                       class_mode = 'binary', shuffle=False)
```

```
Found 3000 images belonging to 2 classes.
```

```
Found 400 images belonging to 2 classes.
```

```
Found 200 images belonging to 2 classes.
```

```
[6]: # data augmentation of train dataset
```

```
train_generator_1 = train_datagen.flow_from_directory(train_path, classes =  
                                                       ['MergerGalaxies', 'NormalGalaxiesLowInclination'], batch_size = 1,  
                                                       class_mode = 'binary')  
  
fig = plt.gcf()  
fig.set_size_inches(15, 15)  
  
# fig, ax = plt.subplots(nrows=2, ncols=4, figsize=(15,15))  
for i in range(10):  
  
    # convert to unsigned integers for plotting  
    image = next(train_generator_1)[0]
```

```
# changing size from (1, 256, 256, 3) to (256, 256, 3) for plotting the image
image = np.squeeze(image)

sp = plt.subplot(2, 5 , i+1) #subplot indices start at 1
sp.axis('Off')
plt.imshow(np.asarray(image))
```

Found 3000 images belonging to 2 classes.



[7]:

```
print(train_generator.class_indices)
print(valid_generator.class_indices)
print(test_generator.class_indices)
```

```
{'MergerGalaxies': 0, 'NormalGalaxiesLowInclination': 1}
{'MergerGalaxies': 0, 'NormalGalaxiesLowInclination': 1}
{'MergerGalaxies': 0, 'NormalGalaxiesLowInclination': 1}
```

[8]:

```
# model

FILTERS = 128
ACTIVATION = 'relu'
```

```

KERNEL_SIZE = 5
INITIALIZER = 'he_normal'
convLayers = [128, 64, 64]
DenseLayers = [256, 128, 64]
POOL_SIZE = 3
dropout = 0.5
LR_INIT = 0.001

def build_model():
    model = Sequential()

    model.add(Conv2D(filters=FILTERS, activation = ACTIVATION, kernel_size = KERNEL_SIZE, padding='same', kernel_initializer = INITIALIZER , input_shape=(256,256,3)))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size= 9))

    for conv in convLayers:
        model.add(Conv2D(conv, activation = ACTIVATION, kernel_size= KERNEL_SIZE, padding = 'same', kernel_initializer=INITIALIZER))
        model.add(BatchNormalization())
        model.add(MaxPooling2D(pool_size= POOL_SIZE))

    model.add(Flatten())

    for dense in DenseLayers:
        model.add(Dense(dense, activation= ACTIVATION, kernel_initializer = INITIALIZER))
        model.add(BatchNormalization())
        model.add(Dropout(dropout))

    model.add(Dense(1, activation='sigmoid'))
    # model.compile(loss=tfa.losses.SigmoidFocalCrossEntropy(), optimizer=Adam(lr=LR_INIT), metrics= [tf.keras.metrics.AUC(name='auc')])
    model.compile(loss= tf.keras.losses.BinaryCrossentropy() , optimizer=Adam(lr=LR_INIT), metrics= [tf.keras.metrics.AUC(name='auc')])

    return model

```

[9]: model = build_model()
model.summary()

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 256, 256, 128)	9728
batch_normalization (BatchNo	(None, 256, 256, 128)	512
max_pooling2d (MaxPooling2D)	(None, 28, 28, 128)	0
conv2d_1 (Conv2D)	(None, 28, 28, 128)	409728
batch_normalization_1 (Batch	(None, 28, 28, 128)	512
max_pooling2d_1 (MaxPooling2	(None, 9, 9, 128)	0
conv2d_2 (Conv2D)	(None, 9, 9, 64)	204864
batch_normalization_2 (Batch	(None, 9, 9, 64)	256
max_pooling2d_2 (MaxPooling2	(None, 3, 3, 64)	0
conv2d_3 (Conv2D)	(None, 3, 3, 64)	102464
batch_normalization_3 (Batch	(None, 3, 3, 64)	256
max_pooling2d_3 (MaxPooling2	(None, 1, 1, 64)	0
flatten (Flatten)	(None, 64)	0
dense (Dense)	(None, 256)	16640
batch_normalization_4 (Batch	(None, 256)	1024
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32896
batch_normalization_5 (Batch	(None, 128)	512
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8256
batch_normalization_6 (Batch	(None, 64)	256
dropout_2 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 1)	65

```
Total params: 787,969  
Trainable params: 786,305  
Non-trainable params: 1,664
```

```
[10]: import os  
root_logdir = os.path.join('/content/drive/MyDrive/project', 'my_logs')  
ExperimentName = 'Expt1_02'
```

```
def get_run_logdir():  
    # import time  
    # run_id = time.strftime("run_%Y_%m_%d-%H_%M_%S_")  
    return os.path.join(root_logdir, ExperimentName)
```

```
[11]: # weighing classes  
class_weights = class_weight.compute_class_weight('balanced', np.  
    unique(train_generator.classes), train_generator.classes)  
class_weight_dict = {0:class_weights[0], 1:class_weights[1]}
```

```
[12]: class_weight_dict
```

```
[12]: {0: 3.0, 1: 0.6}
```

ADD CALLBACKS BEFORE RUNNING NEXT TIME

```
[ ]: tensorboard_cb = TensorBoard(get_run_logdir())  
Reducelr_cb = ReduceLROnPlateau(monitor='loss', factor =0.2, patience =2,  
    verbose=1, mode='min', min_lr = 1e-5)  
early_cb = EarlyStopping(monitor='loss', patience=5, mode='min')  
filepath = '/content/drive/MyDrive/project/model_{epoch:02d}-{auc:.2f}.h5'  
checkpoint_cb = tf.keras.callbacks.ModelCheckpoint(filepath, monitor = 'auc',  
    save_best_only = True, verbose = 1, mode = 'max' )  
EPOCHS_NUM = 20  
history = model.fit(train_generator,  
    epochs= EPOCHS_NUM,  
    class_weight = class_weight_dict,  
    validation_data = valid_generator,  
    verbose=1)  
  
model.save('/content/drive/MyDrive/project/model_Expt1.h5')
```

```
[ ]: # learning curve  
loss = history.history['loss']  
val_loss = history.history['val_loss']  
  
plt.plot(loss, label='Training Loss')  
plt.plot(val_loss, label='Validation Loss')  
plt.legend(loc='upper right')
```

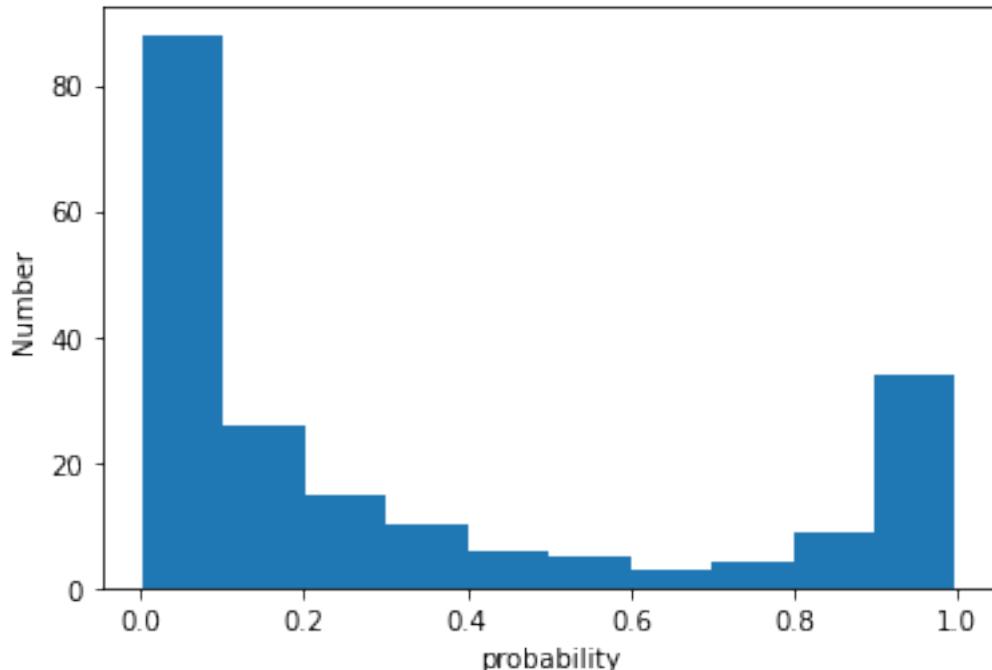
```
plt.ylabel('Sigmoid Focal Cross Entropy')
# plt.ylim([0.1,2])
plt.title('Training and Validation Loss')
plt.xlabel('epoch')
plt.savefig('/content/drive/MyDrive/project/LearningCurve_expt1_02.png')
plt.show()
```

[13]: # predict on test

```
model = keras.models.load_model('/content/drive/MyDrive/project/low/model_Expt1.
→h5')
predictions = model.predict(test_generator, verbose =1)
```

10/10 [=====] - 4s 71ms/step

[15]: plt.hist(predictions)
plt.xlabel('probability')
plt.ylabel('Number')
plt.savefig('/content/drive/MyDrive/project/TestPredictions_expt1_02.png')
plt.show()



[16]: predictedClasses = np.where(predictions>0.5, 1, 0)

[17]: predictedClass_=predictedClasses.ravel()

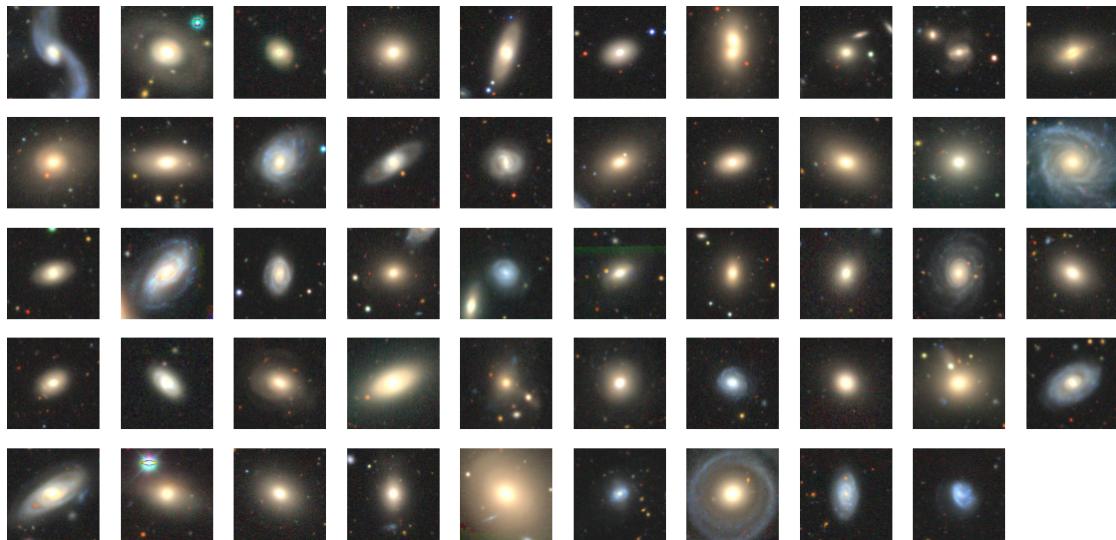
```
[18]: normal_classified_normal = []
for idx in range(len(predictedClass_)):
    if (test_generator.classes[idx]==1) & (predictedClass_[idx]==1):
        normal_classified_normal.append(idx)

[19]: merger_classified_merger = []
for idx in range(len(predictedClass_)):
    if (test_generator.classes[idx]==0) & (predictedClass_[idx]==0):
        merger_classified_merger.append(idx)

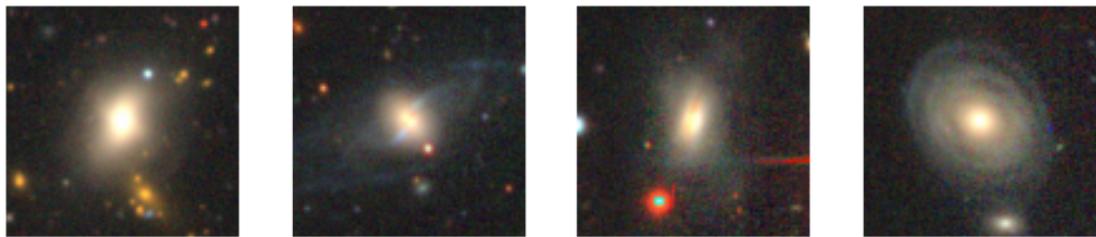
[20]: normal_classified_merger = []
for idx in range(len(predictedClass_)):
    if (test_generator.classes[idx]==1) & (predictedClass_[idx]==0):
        normal_classified_merger.append(idx)

[21]: merger_classified_normal = []
for idx in range(len(predictedClass_)):
    if (test_generator.classes[idx]==0) & (predictedClass_[idx]==1):
        merger_classified_normal.append(idx)

[22]: # NORMAL CLASSIFIED AS MERGER
from matplotlib import image
fig = plt.gcf()
fig.set_size_inches(20, 20)
nrows = 10
ncols = 10
fnames = test_generator.filenames
for i, idx in enumerate(normal_classified_merger):
    sp = plt.subplot(nrows, ncols, i+1)
    sp.axis('Off')
    img = image.imread(os.path.join('/content/drive/MyDrive/finalProject/test1', fnames[idx]))
    plt.imshow(img)
```



```
[23]: # MERGER CLASSIFIED AS NORMAL
fig = plt.gcf()
fig.set_size_inches(20, 20)
nrows = 10
ncols = 10
fnames = test_generator.filenames
for i, idx in enumerate(merger_classified_normal):
    sp = plt.subplot(nrows, ncols, i+1)
    sp.axis('Off')
    img = image.imread(os.path.join('/content/drive/MyDrive/finalProject/test1', fnames[idx]))
    plt.imshow(img)
```

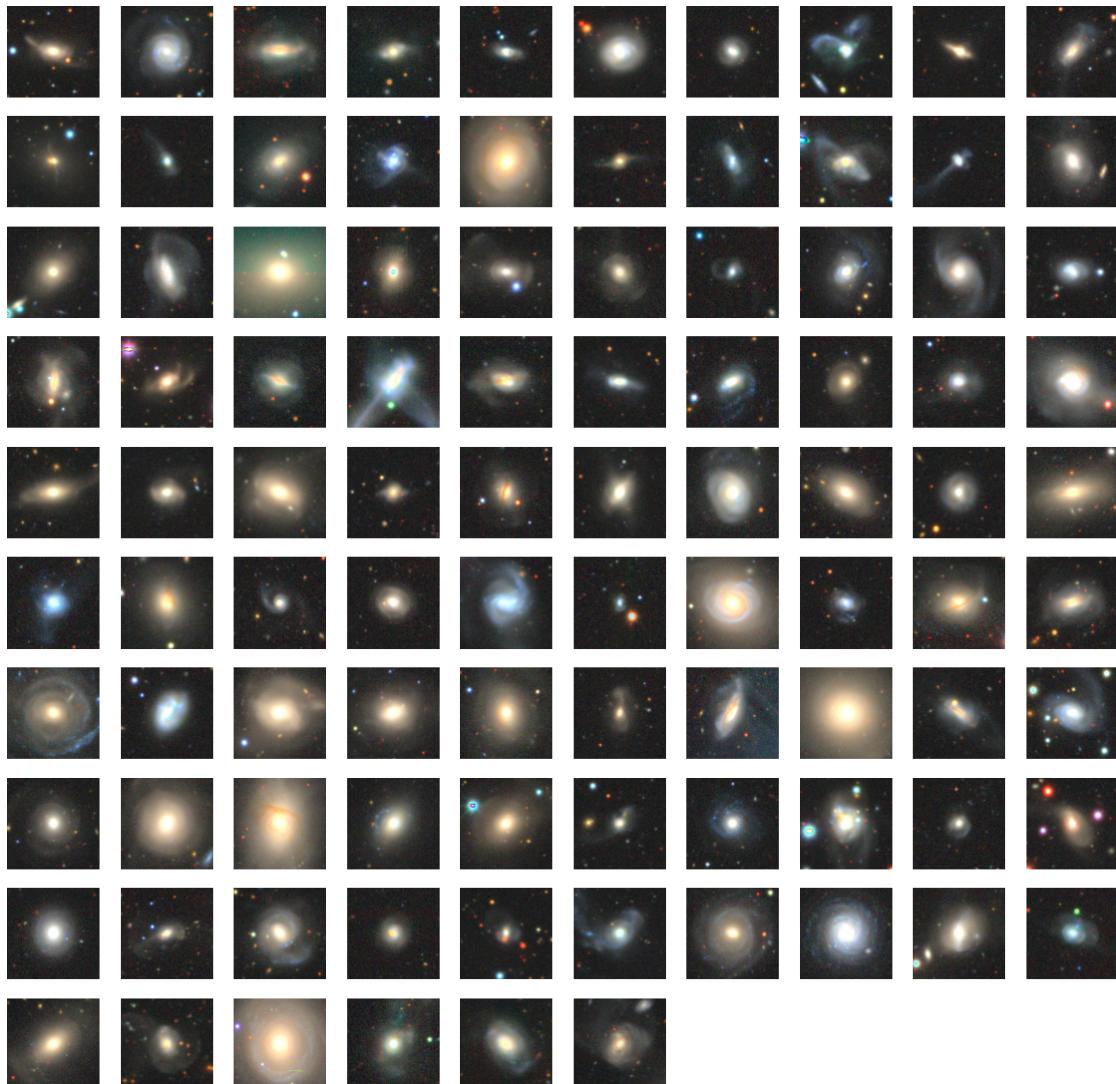


```
[24]: # MERGER CLASSIFIED AS MERGER
fig = plt.gcf()
fig.set_size_inches(20, 20)
nrows = 10
ncols = 10
```

```

fnames = test_generator.filenames
for i, idx in enumerate(merger_classified_merger):
    sp = plt.subplot(nrows, ncols, i+1)
    sp.axis('Off')
    img = image.imread(os.path.join('/content/drive/MyDrive/finalProject/test1', fnames[idx]))
    plt.imshow(img)

```

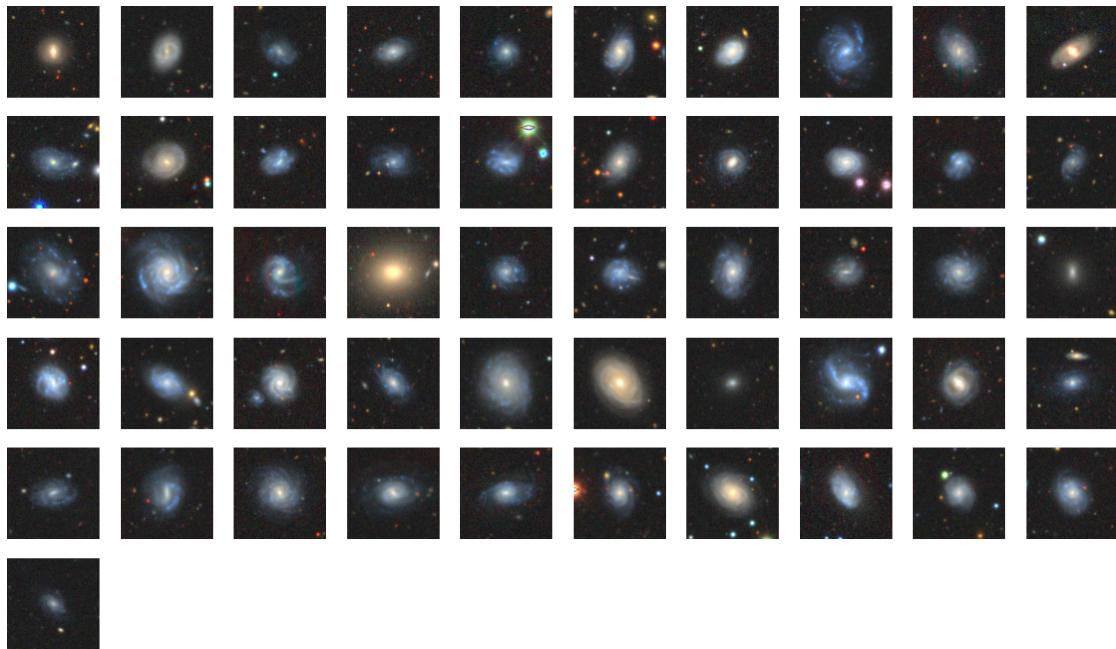


```
[25]: # NORMAL CLASSIFIED AS NORMAL
fig = plt.gcf()
fig.set_size_inches(20, 20)
nrows = 10
ncols = 10
```

```

fnames = test_generator.filenames
for i, idx in enumerate(normal_classified_normal):
    sp = plt.subplot(nrows, ncols, i+1)
    sp.axis('Off')
    img = image.imread(os.path.join('/content/drive/MyDrive/finalProject/test1', fnames[idx]))
    plt.imshow(img)

```



```

[26]: print('Confusion Matrix')
cm = metrics.confusion_matrix(test_generator.classes, predictedClasses)
print(cm)

cm_report = metrics.classification_report(test_generator.classes, predictedClasses)
print(cm_report)

```

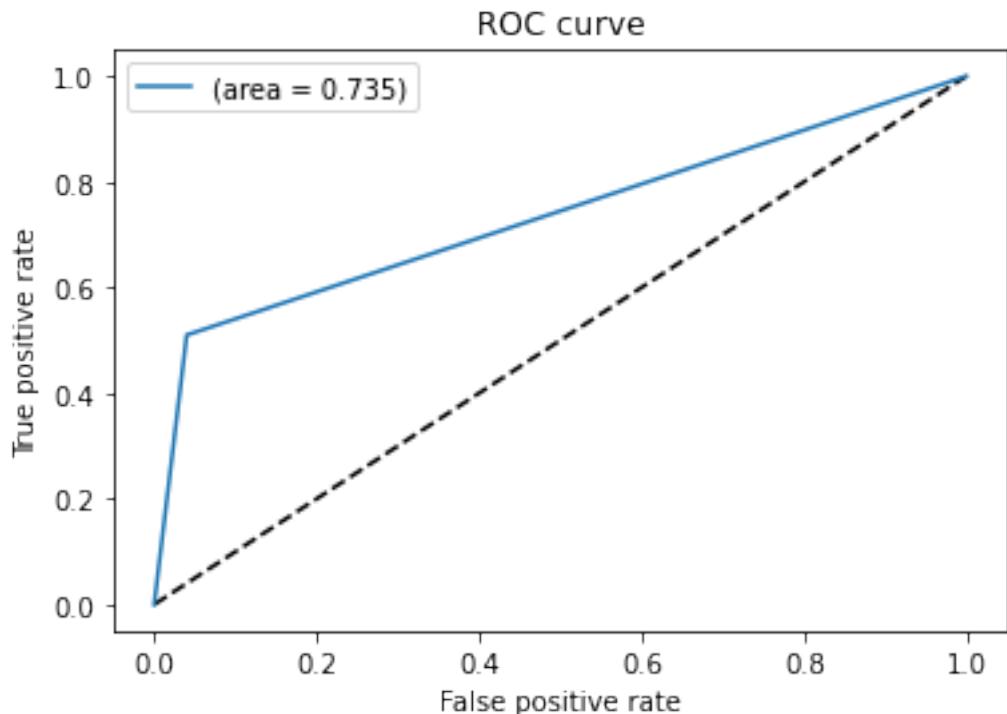
Confusion Matrix				
[[96 4]				
[49 51]]				
	precision	recall	f1-score	support
0	0.66	0.96	0.78	100
1	0.93	0.51	0.66	100
accuracy			0.73	200
macro avg	0.79	0.73	0.72	200

weighted avg	0.79	0.73	0.72	200
--------------	------	------	------	-----

```
[27]: # ROC-AUC curve
fpr, tpr, threshold = roc_curve(test_generator.classes, predictedClasses)
#predictedClasses predictions

# AUC
auc_ = auc(fpr, tpr)

# Plot
plt.plot([0, 1], [0, 1], 'k--')
plt.plot(fpr, tpr, label='(area = {:.3f})'.format(auc_))
plt.xlabel('False positive rate')
plt.ylabel('True positive rate')
plt.title('ROC curve')
plt.legend(loc='best')
plt.savefig('/content/drive/MyDrive/project/ROC_AUC_expt1_02.png')
plt.show()
```



AutoEncoder

AutoEncoderFinal

April 21, 2021

```
[2]: import os
from google.colab import drive
drive.mount('/content/drive', force_remount=True)
```

Mounted at /content/drive

```
[3]: import keras
from keras.models import Sequential
from keras.optimizers import Adam
from keras.layers import Dense, Activation, Flatten, Input, Reshape
from keras.layers import Conv2D, MaxPooling2D, UpSampling2D, BatchNormalization
import matplotlib.pyplot as plt
from keras import backend as K
import numpy as np
from keras.preprocessing.image import ImageDataGenerator, array_to_img,
    img_to_array, load_img
from PIL import Image, ImageChops
from sklearn.neighbors import KernelDensity
import random
import tensorflow as tf
from tensorflow.python.keras.backend import set_session
config = tf.compat.v1.ConfigProto()
config.gpu_options.allow_growth=True
set_session(tf.compat.v1.Session(config=config))
from sklearn.model_selection import train_test_split
```

```
[4]: batch_size = 84
train_datagen = ImageDataGenerator(rescale=1./255, data_format='channels_last')
train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/normalgal',
    batch_size=batch_size,
    target_size = (192,192),
    class_mode='input'
)
```

Found 5980 images belonging to 2 classes.

```
[5]: test_datagen = ImageDataGenerator(rescale=1./255, data_format='channels_last')
validation_generator = test_datagen.flow_from_directory(
    '/content/drive/MyDrive/NormalVal/',
    batch_size=batch_size,
    target_size = (192,192),
    class_mode='input'
)
```

Found 505 images belonging to 2 classes.

```
[6]: anomaly_datagen=ImageDataGenerator(rescale=1./255, data_format='channels_last')
anomaly_generator = anomaly_datagen.flow_from_directory(
    '/content/drive/MyDrive/PostMergers/',
    target_size=(192, 192),
    batch_size=batch_size,
    class_mode='input'
)
```

Found 1121 images belonging to 2 classes.

```
[35]: encoder = Sequential()

encoder.add(Conv2D(48, (3, 3), padding='same',activation='selu',  

    ↪kernel_initializer='lecun_normal', input_shape=(192, 192, 3)))
encoder.add(MaxPooling2D(pool_size=(4,4), padding='same'))

encoder.add(Conv2D(32,(3, 3),activation='selu',  

    ↪kernel_initializer='lecun_normal', padding='same'))
encoder.add(MaxPooling2D(pool_size=(4,4), padding='same'))

encoder.add(Conv2D(12,(3, 3),activation='selu',  

    ↪kernel_initializer='lecun_normal', padding='same'))
encoder.add(MaxPooling2D(pool_size=(2,2), padding='same'))

encoder.add(Conv2D(6,(3, 3),activation='selu',  

    ↪kernel_initializer='lecun_normal', padding='same'))
encoder.add(MaxPooling2D(pool_size=(2,2), padding='same'))

#-----#
decoder = Sequential()

decoder.add(Conv2D(6,(3, 3),activation='selu',  

    ↪kernel_initializer='lecun_normal', padding='same',input_shape=(3,3,6)))
decoder.add(UpSampling2D((2,2),interpolation='bilinear'))

decoder.add(Conv2D(12,(3, 3),activation='selu',  

    ↪kernel_initializer='lecun_normal', padding='same'))
```

```

decoder.add(UpSampling2D((2,2),interpolation='bilinear'))

decoder.add(Conv2D(32,(3, 3),activation='selu',  

    ↪kernel_initializer='lecun_normal', padding='same'))  

decoder.add(UpSampling2D((4,4),interpolation='bilinear'))

decoder.add(Conv2D(48,(3, 3),activation='selu',  

    ↪kernel_initializer='lecun_normal', padding='same'))  

decoder.add(UpSampling2D((4,4),interpolation='bilinear'))

decoder.add(Conv2D(3,(3, 3), activation='sigmoid', padding='same'))  

#-----  

model = keras.models.Sequential([encoder, decoder])

model.compile(optimizer=keras.optimizers.SGD(lr=1*10**-3,momentum= 0.9,  

    ↪nesterov=True), loss='mse')

```

[9]: encoder.summary()

```

Model: "sequential_3"

-----  

Layer (type)           Output Shape        Param #  

-----  

conv2d_9 (Conv2D)      (None, 192, 192, 48)   1344  

-----  

max_pooling2d_4 (MaxPooling2 (None, 48, 48, 48)   0  

-----  

conv2d_10 (Conv2D)     (None, 48, 48, 32)      13856  

-----  

max_pooling2d_5 (MaxPooling2 (None, 12, 12, 32)   0  

-----  

conv2d_11 (Conv2D)     (None, 12, 12, 12)       3468  

-----  

max_pooling2d_6 (MaxPooling2 (None, 6, 6, 12)      0  

-----  

conv2d_12 (Conv2D)     (None, 6, 6, 6)          654  

-----  

max_pooling2d_7 (MaxPooling2 (None, 3, 3, 6)       0  

-----  

Total params: 19,322  

Trainable params: 19,322  

Non-trainable params: 0
-----
```

[10]: decoder.summary()

```

Model: "sequential_4"

-----
```

Layer (type)	Output Shape	Param #
conv2d_13 (Conv2D)	(None, 3, 3, 6)	330
up_sampling2d_4 (UpSampling2D)	(None, 6, 6, 6)	0
conv2d_14 (Conv2D)	(None, 6, 6, 12)	660
up_sampling2d_5 (UpSampling2D)	(None, 12, 12, 12)	0
conv2d_15 (Conv2D)	(None, 12, 12, 32)	3488
up_sampling2d_6 (UpSampling2D)	(None, 48, 48, 32)	0
conv2d_16 (Conv2D)	(None, 48, 48, 48)	13872
up_sampling2d_7 (UpSampling2D)	(None, 192, 192, 48)	0
conv2d_17 (Conv2D)	(None, 192, 192, 3)	1299
<hr/>		
Total params: 19,649		
Trainable params: 19,649		
Non-trainable params: 0		
<hr/>		

```
[36]: es = [keras.callbacks.EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=30),
          keras.callbacks.ReduceLROnPlateau(monitor='val_loss', mode='min', factor=0.2, patience=5, min_lr=1.e-6)]
model_filepath = 'models/model1.keras'
save_best = keras.callbacks.ModelCheckpoint(model_filepath, monitor='val_loss', save_best_only=True, mode='min')
model.fit(
    train_generator,
    steps_per_epoch=1000 // batch_size,
    epochs=300,
    batch_size=batch_size,
    validation_data=validation_generator,
    shuffle = True,
    callbacks=[es, save_best])
```

Epoch 1/300
11/11 [=====] - 7s 613ms/step - loss: 0.1224 -
val_loss: 0.0353
Epoch 2/300
11/11 [=====] - 6s 581ms/step - loss: 0.0316 -
val_loss: 0.0282
Epoch 3/300

```
11/11 [=====] - 6s 566ms/step - loss: 0.0279 -
val_loss: 0.0281
Epoch 4/300
11/11 [=====] - 6s 563ms/step - loss: 0.0275 -
val_loss: 0.0273
Epoch 5/300
11/11 [=====] - 6s 564ms/step - loss: 0.0274 -
val_loss: 0.0264
Epoch 6/300
11/11 [=====] - 6s 593ms/step - loss: 0.0254 -
val_loss: 0.0254
Epoch 7/300
11/11 [=====] - 6s 568ms/step - loss: 0.0256 -
val_loss: 0.0246
Epoch 8/300
11/11 [=====] - 6s 571ms/step - loss: 0.0237 -
val_loss: 0.0240
Epoch 9/300
11/11 [=====] - 6s 574ms/step - loss: 0.0242 -
val_loss: 0.0234
Epoch 10/300
11/11 [=====] - 6s 570ms/step - loss: 0.0234 -
val_loss: 0.0230
Epoch 11/300
11/11 [=====] - 6s 572ms/step - loss: 0.0237 -
val_loss: 0.0225
Epoch 12/300
11/11 [=====] - 6s 576ms/step - loss: 0.0236 -
val_loss: 0.0221
Epoch 13/300
11/11 [=====] - 6s 561ms/step - loss: 0.0211 -
val_loss: 0.0217
Epoch 14/300
11/11 [=====] - 6s 574ms/step - loss: 0.0207 -
val_loss: 0.0213
Epoch 15/300
11/11 [=====] - 6s 569ms/step - loss: 0.0212 -
val_loss: 0.0209
Epoch 16/300
11/11 [=====] - 6s 562ms/step - loss: 0.0214 -
val_loss: 0.0206
Epoch 17/300
11/11 [=====] - 6s 566ms/step - loss: 0.0202 -
val_loss: 0.0202
Epoch 18/300
11/11 [=====] - 6s 582ms/step - loss: 0.0214 -
val_loss: 0.0199
Epoch 19/300
```

```
11/11 [=====] - 6s 571ms/step - loss: 0.0199 -
val_loss: 0.0196
Epoch 20/300
11/11 [=====] - 6s 548ms/step - loss: 0.0191 -
val_loss: 0.0192
Epoch 21/300
11/11 [=====] - 6s 592ms/step - loss: 0.0186 -
val_loss: 0.0189
Epoch 22/300
11/11 [=====] - 6s 581ms/step - loss: 0.0189 -
val_loss: 0.0186
Epoch 23/300
11/11 [=====] - 6s 571ms/step - loss: 0.0192 -
val_loss: 0.0183
Epoch 24/300
11/11 [=====] - 6s 570ms/step - loss: 0.0181 -
val_loss: 0.0181
Epoch 25/300
11/11 [=====] - 6s 573ms/step - loss: 0.0171 -
val_loss: 0.0178
Epoch 26/300
11/11 [=====] - 6s 565ms/step - loss: 0.0178 -
val_loss: 0.0176
Epoch 27/300
11/11 [=====] - 6s 567ms/step - loss: 0.0180 -
val_loss: 0.0174
Epoch 28/300
11/11 [=====] - 6s 568ms/step - loss: 0.0168 -
val_loss: 0.0172
Epoch 29/300
11/11 [=====] - 6s 568ms/step - loss: 0.0166 -
val_loss: 0.0170
Epoch 30/300
11/11 [=====] - 6s 573ms/step - loss: 0.0165 -
val_loss: 0.0168
Epoch 31/300
11/11 [=====] - 6s 561ms/step - loss: 0.0168 -
val_loss: 0.0167
Epoch 32/300
11/11 [=====] - 6s 568ms/step - loss: 0.0165 -
val_loss: 0.0165
Epoch 33/300
11/11 [=====] - 6s 573ms/step - loss: 0.0160 -
val_loss: 0.0164
Epoch 34/300
11/11 [=====] - 6s 568ms/step - loss: 0.0164 -
val_loss: 0.0162
Epoch 35/300
```

```
11/11 [=====] - 6s 567ms/step - loss: 0.0157 -
val_loss: 0.0161
Epoch 36/300
11/11 [=====] - 6s 569ms/step - loss: 0.0153 -
val_loss: 0.0160
Epoch 37/300
11/11 [=====] - 6s 578ms/step - loss: 0.0164 -
val_loss: 0.0159
Epoch 38/300
11/11 [=====] - 6s 572ms/step - loss: 0.0156 -
val_loss: 0.0158
Epoch 39/300
11/11 [=====] - 6s 565ms/step - loss: 0.0162 -
val_loss: 0.0156
Epoch 40/300
11/11 [=====] - 6s 573ms/step - loss: 0.0159 -
val_loss: 0.0155
Epoch 41/300
11/11 [=====] - 6s 573ms/step - loss: 0.0150 -
val_loss: 0.0154
Epoch 42/300
11/11 [=====] - 6s 574ms/step - loss: 0.0159 -
val_loss: 0.0153
Epoch 43/300
11/11 [=====] - 6s 564ms/step - loss: 0.0153 -
val_loss: 0.0152
Epoch 44/300
11/11 [=====] - 6s 562ms/step - loss: 0.0152 -
val_loss: 0.0151
Epoch 45/300
11/11 [=====] - 6s 570ms/step - loss: 0.0147 -
val_loss: 0.0151
Epoch 46/300
11/11 [=====] - 6s 563ms/step - loss: 0.0146 -
val_loss: 0.0150
Epoch 47/300
11/11 [=====] - 6s 563ms/step - loss: 0.0138 -
val_loss: 0.0149
Epoch 48/300
11/11 [=====] - 6s 568ms/step - loss: 0.0144 -
val_loss: 0.0148
Epoch 49/300
11/11 [=====] - 6s 566ms/step - loss: 0.0144 -
val_loss: 0.0147
Epoch 50/300
11/11 [=====] - 6s 573ms/step - loss: 0.0142 -
val_loss: 0.0147
Epoch 51/300
```

```
11/11 [=====] - 6s 557ms/step - loss: 0.0147 -
val_loss: 0.0146
Epoch 52/300
11/11 [=====] - 6s 567ms/step - loss: 0.0146 -
val_loss: 0.0145
Epoch 53/300
11/11 [=====] - 6s 574ms/step - loss: 0.0146 -
val_loss: 0.0144
Epoch 54/300
11/11 [=====] - 6s 563ms/step - loss: 0.0146 -
val_loss: 0.0144
Epoch 55/300
11/11 [=====] - 6s 560ms/step - loss: 0.0145 -
val_loss: 0.0143
Epoch 56/300
11/11 [=====] - 6s 565ms/step - loss: 0.0142 -
val_loss: 0.0142
Epoch 57/300
11/11 [=====] - 6s 560ms/step - loss: 0.0138 -
val_loss: 0.0142
Epoch 58/300
11/11 [=====] - 6s 575ms/step - loss: 0.0138 -
val_loss: 0.0141
Epoch 59/300
11/11 [=====] - 6s 555ms/step - loss: 0.0137 -
val_loss: 0.0140
Epoch 60/300
11/11 [=====] - 6s 556ms/step - loss: 0.0139 -
val_loss: 0.0140
Epoch 61/300
11/11 [=====] - 6s 560ms/step - loss: 0.0137 -
val_loss: 0.0139
Epoch 62/300
11/11 [=====] - 6s 555ms/step - loss: 0.0138 -
val_loss: 0.0139
Epoch 63/300
11/11 [=====] - 6s 564ms/step - loss: 0.0138 -
val_loss: 0.0138
Epoch 64/300
11/11 [=====] - 6s 559ms/step - loss: 0.0142 -
val_loss: 0.0138
Epoch 65/300
11/11 [=====] - 6s 559ms/step - loss: 0.0130 -
val_loss: 0.0137
Epoch 66/300
11/11 [=====] - 6s 564ms/step - loss: 0.0137 -
val_loss: 0.0137
Epoch 67/300
```

```
11/11 [=====] - 6s 558ms/step - loss: 0.0135 -
val_loss: 0.0136
Epoch 68/300
11/11 [=====] - 6s 572ms/step - loss: 0.0133 -
val_loss: 0.0136
Epoch 69/300
11/11 [=====] - 6s 568ms/step - loss: 0.0129 -
val_loss: 0.0135
Epoch 70/300
11/11 [=====] - 6s 562ms/step - loss: 0.0139 -
val_loss: 0.0135
Epoch 71/300
11/11 [=====] - 6s 569ms/step - loss: 0.0134 -
val_loss: 0.0134
Epoch 72/300
11/11 [=====] - 6s 568ms/step - loss: 0.0133 -
val_loss: 0.0134
Epoch 73/300
11/11 [=====] - 6s 565ms/step - loss: 0.0134 -
val_loss: 0.0133
Epoch 74/300
11/11 [=====] - 6s 564ms/step - loss: 0.0133 -
val_loss: 0.0133
Epoch 75/300
11/11 [=====] - 6s 567ms/step - loss: 0.0129 -
val_loss: 0.0132
Epoch 76/300
11/11 [=====] - 6s 561ms/step - loss: 0.0132 -
val_loss: 0.0132
Epoch 77/300
11/11 [=====] - 6s 567ms/step - loss: 0.0130 -
val_loss: 0.0132
Epoch 78/300
11/11 [=====] - 6s 561ms/step - loss: 0.0128 -
val_loss: 0.0131
Epoch 79/300
11/11 [=====] - 6s 573ms/step - loss: 0.0129 -
val_loss: 0.0131
Epoch 80/300
11/11 [=====] - 6s 565ms/step - loss: 0.0136 -
val_loss: 0.0130
Epoch 81/300
11/11 [=====] - 6s 569ms/step - loss: 0.0126 -
val_loss: 0.0130
Epoch 82/300
11/11 [=====] - 6s 564ms/step - loss: 0.0129 -
val_loss: 0.0130
Epoch 83/300
```

```
11/11 [=====] - 6s 555ms/step - loss: 0.0127 -
val_loss: 0.0129
Epoch 84/300
11/11 [=====] - 6s 562ms/step - loss: 0.0127 -
val_loss: 0.0129
Epoch 85/300
11/11 [=====] - 6s 561ms/step - loss: 0.0126 -
val_loss: 0.0128
Epoch 86/300
11/11 [=====] - 6s 559ms/step - loss: 0.0124 -
val_loss: 0.0128
Epoch 87/300
11/11 [=====] - 6s 557ms/step - loss: 0.0129 -
val_loss: 0.0128
Epoch 88/300
11/11 [=====] - 6s 555ms/step - loss: 0.0124 -
val_loss: 0.0127
Epoch 89/300
11/11 [=====] - 6s 594ms/step - loss: 0.0123 -
val_loss: 0.0127
Epoch 90/300
11/11 [=====] - 6s 574ms/step - loss: 0.0123 -
val_loss: 0.0127
Epoch 91/300
11/11 [=====] - 6s 560ms/step - loss: 0.0120 -
val_loss: 0.0127
Epoch 92/300
11/11 [=====] - 6s 550ms/step - loss: 0.0124 -
val_loss: 0.0126
Epoch 93/300
11/11 [=====] - 6s 555ms/step - loss: 0.0123 -
val_loss: 0.0126
Epoch 94/300
11/11 [=====] - 6s 566ms/step - loss: 0.0122 -
val_loss: 0.0126
Epoch 95/300
11/11 [=====] - 6s 570ms/step - loss: 0.0128 -
val_loss: 0.0125
Epoch 96/300
11/11 [=====] - 6s 567ms/step - loss: 0.0123 -
val_loss: 0.0125
Epoch 97/300
11/11 [=====] - 6s 556ms/step - loss: 0.0122 -
val_loss: 0.0125
Epoch 98/300
11/11 [=====] - 6s 552ms/step - loss: 0.0125 -
val_loss: 0.0125
Epoch 99/300
```

```
11/11 [=====] - 6s 561ms/step - loss: 0.0126 -
val_loss: 0.0124
Epoch 100/300
11/11 [=====] - 6s 568ms/step - loss: 0.0121 -
val_loss: 0.0124
Epoch 101/300
11/11 [=====] - 6s 555ms/step - loss: 0.0125 -
val_loss: 0.0124
Epoch 102/300
11/11 [=====] - 6s 554ms/step - loss: 0.0117 -
val_loss: 0.0124
Epoch 103/300
11/11 [=====] - 6s 552ms/step - loss: 0.0120 -
val_loss: 0.0123
Epoch 104/300
11/11 [=====] - 6s 559ms/step - loss: 0.0122 -
val_loss: 0.0123
Epoch 105/300
11/11 [=====] - 6s 558ms/step - loss: 0.0122 -
val_loss: 0.0123
Epoch 106/300
11/11 [=====] - 6s 559ms/step - loss: 0.0125 -
val_loss: 0.0123
Epoch 107/300
11/11 [=====] - 6s 550ms/step - loss: 0.0120 -
val_loss: 0.0123
Epoch 108/300
11/11 [=====] - 6s 555ms/step - loss: 0.0120 -
val_loss: 0.0122
Epoch 109/300
11/11 [=====] - 6s 558ms/step - loss: 0.0124 -
val_loss: 0.0122
Epoch 110/300
11/11 [=====] - 6s 556ms/step - loss: 0.0118 -
val_loss: 0.0122
Epoch 111/300
11/11 [=====] - 6s 566ms/step - loss: 0.0121 -
val_loss: 0.0122
Epoch 112/300
11/11 [=====] - 6s 552ms/step - loss: 0.0115 -
val_loss: 0.0122
Epoch 113/300
11/11 [=====] - 6s 553ms/step - loss: 0.0121 -
val_loss: 0.0121
Epoch 114/300
11/11 [=====] - 6s 551ms/step - loss: 0.0114 -
val_loss: 0.0121
Epoch 115/300
```

```
11/11 [=====] - 6s 557ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 116/300
11/11 [=====] - 6s 559ms/step - loss: 0.0116 -
val_loss: 0.0121
Epoch 117/300
11/11 [=====] - 6s 564ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 118/300
11/11 [=====] - 6s 564ms/step - loss: 0.0121 -
val_loss: 0.0121
Epoch 119/300
11/11 [=====] - 6s 565ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 120/300
11/11 [=====] - 6s 563ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 121/300
11/11 [=====] - 6s 565ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 122/300
11/11 [=====] - 6s 577ms/step - loss: 0.0122 -
val_loss: 0.0121
Epoch 123/300
11/11 [=====] - 6s 563ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 124/300
11/11 [=====] - 6s 554ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 125/300
11/11 [=====] - 6s 553ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 126/300
11/11 [=====] - 6s 563ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 127/300
11/11 [=====] - 6s 549ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 128/300
11/11 [=====] - 6s 552ms/step - loss: 0.0116 -
val_loss: 0.0121
Epoch 129/300
11/11 [=====] - 6s 557ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 130/300
11/11 [=====] - 6s 557ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 131/300
```

```
11/11 [=====] - 6s 558ms/step - loss: 0.0123 -
val_loss: 0.0121
Epoch 132/300
11/11 [=====] - 6s 554ms/step - loss: 0.0115 -
val_loss: 0.0121
Epoch 133/300
11/11 [=====] - 6s 544ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 134/300
11/11 [=====] - 6s 554ms/step - loss: 0.0115 -
val_loss: 0.0121
Epoch 135/300
11/11 [=====] - 6s 562ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 136/300
11/11 [=====] - 6s 557ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 137/300
11/11 [=====] - 6s 567ms/step - loss: 0.0124 -
val_loss: 0.0121
Epoch 138/300
11/11 [=====] - 6s 554ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 139/300
11/11 [=====] - 6s 567ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 140/300
11/11 [=====] - 6s 559ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 141/300
11/11 [=====] - 6s 556ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 142/300
11/11 [=====] - 6s 569ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 143/300
11/11 [=====] - 6s 558ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 144/300
11/11 [=====] - 6s 564ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 145/300
11/11 [=====] - 6s 557ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 146/300
11/11 [=====] - 6s 578ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 147/300
```

```
11/11 [=====] - 6s 559ms/step - loss: 0.0123 -
val_loss: 0.0121
Epoch 148/300
11/11 [=====] - 6s 563ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 149/300
11/11 [=====] - 6s 547ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 150/300
11/11 [=====] - 6s 550ms/step - loss: 0.0123 -
val_loss: 0.0121
Epoch 151/300
11/11 [=====] - 6s 551ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 152/300
11/11 [=====] - 6s 554ms/step - loss: 0.0121 -
val_loss: 0.0121
Epoch 153/300
11/11 [=====] - 6s 561ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 154/300
11/11 [=====] - 6s 557ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 155/300
11/11 [=====] - 6s 551ms/step - loss: 0.0121 -
val_loss: 0.0121
Epoch 156/300
11/11 [=====] - 6s 554ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 157/300
11/11 [=====] - 6s 552ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 158/300
11/11 [=====] - 6s 562ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 159/300
11/11 [=====] - 6s 557ms/step - loss: 0.0121 -
val_loss: 0.0121
Epoch 160/300
11/11 [=====] - 6s 555ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 161/300
11/11 [=====] - 6s 552ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 162/300
11/11 [=====] - 6s 549ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 163/300
```

```
11/11 [=====] - 6s 559ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 164/300
11/11 [=====] - 6s 570ms/step - loss: 0.0116 -
val_loss: 0.0121
Epoch 165/300
11/11 [=====] - 6s 554ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 166/300
11/11 [=====] - 6s 555ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 167/300
11/11 [=====] - 6s 562ms/step - loss: 0.0123 -
val_loss: 0.0121
Epoch 168/300
11/11 [=====] - 6s 552ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 169/300
11/11 [=====] - 6s 564ms/step - loss: 0.0123 -
val_loss: 0.0121
Epoch 170/300
11/11 [=====] - 6s 545ms/step - loss: 0.0121 -
val_loss: 0.0121
Epoch 171/300
11/11 [=====] - 6s 550ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 172/300
11/11 [=====] - 6s 552ms/step - loss: 0.0116 -
val_loss: 0.0121
Epoch 173/300
11/11 [=====] - 6s 547ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 174/300
11/11 [=====] - 6s 567ms/step - loss: 0.0121 -
val_loss: 0.0121
Epoch 175/300
11/11 [=====] - 6s 554ms/step - loss: 0.0127 -
val_loss: 0.0121
Epoch 176/300
11/11 [=====] - 6s 562ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 177/300
11/11 [=====] - 6s 557ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 178/300
11/11 [=====] - 6s 547ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 179/300
```

```
11/11 [=====] - 6s 557ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 180/300
11/11 [=====] - 6s 558ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 181/300
11/11 [=====] - 6s 553ms/step - loss: 0.0122 -
val_loss: 0.0121
Epoch 182/300
11/11 [=====] - 6s 555ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 183/300
11/11 [=====] - 6s 563ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 184/300
11/11 [=====] - 6s 542ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 185/300
11/11 [=====] - 6s 554ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 186/300
11/11 [=====] - 6s 553ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 187/300
11/11 [=====] - 6s 556ms/step - loss: 0.0115 -
val_loss: 0.0121
Epoch 188/300
11/11 [=====] - 6s 555ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 189/300
11/11 [=====] - 6s 556ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 190/300
11/11 [=====] - 6s 566ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 191/300
11/11 [=====] - 6s 550ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 192/300
11/11 [=====] - 6s 551ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 193/300
11/11 [=====] - 6s 560ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 194/300
11/11 [=====] - 6s 547ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 195/300
```

```
11/11 [=====] - 6s 552ms/step - loss: 0.0121 -
val_loss: 0.0121
Epoch 196/300
11/11 [=====] - 6s 561ms/step - loss: 0.0115 -
val_loss: 0.0121
Epoch 197/300
11/11 [=====] - 6s 546ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 198/300
11/11 [=====] - 6s 552ms/step - loss: 0.0122 -
val_loss: 0.0121
Epoch 199/300
11/11 [=====] - 6s 540ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 200/300
11/11 [=====] - 6s 557ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 201/300
11/11 [=====] - 6s 555ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 202/300
11/11 [=====] - 6s 556ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 203/300
11/11 [=====] - 6s 558ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 204/300
11/11 [=====] - 6s 546ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 205/300
11/11 [=====] - 6s 550ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 206/300
11/11 [=====] - 6s 563ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 207/300
11/11 [=====] - 6s 549ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 208/300
11/11 [=====] - 6s 555ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 209/300
11/11 [=====] - 6s 551ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 210/300
11/11 [=====] - 6s 555ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 211/300
```

```
11/11 [=====] - 6s 562ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 212/300
11/11 [=====] - 6s 544ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 213/300
11/11 [=====] - 6s 559ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 214/300
11/11 [=====] - 6s 552ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 215/300
11/11 [=====] - 6s 555ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 216/300
11/11 [=====] - 6s 557ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 217/300
11/11 [=====] - 6s 552ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 218/300
11/11 [=====] - 6s 551ms/step - loss: 0.0121 -
val_loss: 0.0121
Epoch 219/300
11/11 [=====] - 6s 550ms/step - loss: 0.0122 -
val_loss: 0.0121
Epoch 220/300
11/11 [=====] - 6s 552ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 221/300
11/11 [=====] - 6s 541ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 222/300
11/11 [=====] - 6s 565ms/step - loss: 0.0116 -
val_loss: 0.0121
Epoch 223/300
11/11 [=====] - 6s 552ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 224/300
11/11 [=====] - 6s 552ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 225/300
11/11 [=====] - 6s 560ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 226/300
11/11 [=====] - 6s 545ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 227/300
```

```
11/11 [=====] - 6s 565ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 228/300
11/11 [=====] - 6s 550ms/step - loss: 0.0121 -
val_loss: 0.0121
Epoch 229/300
11/11 [=====] - 6s 546ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 230/300
11/11 [=====] - 6s 549ms/step - loss: 0.0115 -
val_loss: 0.0121
Epoch 231/300
11/11 [=====] - 6s 535ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 232/300
11/11 [=====] - 6s 560ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 233/300
11/11 [=====] - 6s 556ms/step - loss: 0.0122 -
val_loss: 0.0121
Epoch 234/300
11/11 [=====] - 6s 549ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 235/300
11/11 [=====] - 6s 551ms/step - loss: 0.0125 -
val_loss: 0.0121
Epoch 236/300
11/11 [=====] - 6s 550ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 237/300
11/11 [=====] - 6s 549ms/step - loss: 0.0116 -
val_loss: 0.0121
Epoch 238/300
11/11 [=====] - 6s 561ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 239/300
11/11 [=====] - 6s 556ms/step - loss: 0.0122 -
val_loss: 0.0121
Epoch 240/300
11/11 [=====] - 6s 554ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 241/300
11/11 [=====] - 6s 560ms/step - loss: 0.0121 -
val_loss: 0.0121
Epoch 242/300
11/11 [=====] - 6s 547ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 243/300
```

```
11/11 [=====] - 6s 563ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 244/300
11/11 [=====] - 6s 542ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 245/300
11/11 [=====] - 6s 553ms/step - loss: 0.0122 -
val_loss: 0.0121
Epoch 246/300
11/11 [=====] - 6s 545ms/step - loss: 0.0123 -
val_loss: 0.0121
Epoch 247/300
11/11 [=====] - 6s 542ms/step - loss: 0.0116 -
val_loss: 0.0121
Epoch 248/300
11/11 [=====] - 6s 546ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 249/300
11/11 [=====] - 6s 549ms/step - loss: 0.0121 -
val_loss: 0.0121
Epoch 250/300
11/11 [=====] - 6s 549ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 251/300
11/11 [=====] - 6s 559ms/step - loss: 0.0116 -
val_loss: 0.0121
Epoch 252/300
11/11 [=====] - 6s 543ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 253/300
11/11 [=====] - 6s 546ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 254/300
11/11 [=====] - 6s 547ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 255/300
11/11 [=====] - 6s 543ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 256/300
11/11 [=====] - 6s 546ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 257/300
11/11 [=====] - 6s 543ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 258/300
11/11 [=====] - 6s 546ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 259/300
```

```
11/11 [=====] - 6s 548ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 260/300
11/11 [=====] - 6s 551ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 261/300
11/11 [=====] - 6s 558ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 262/300
11/11 [=====] - 6s 543ms/step - loss: 0.0116 -
val_loss: 0.0121
Epoch 263/300
11/11 [=====] - 6s 552ms/step - loss: 0.0114 -
val_loss: 0.0121
Epoch 264/300
11/11 [=====] - 6s 547ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 265/300
11/11 [=====] - 6s 563ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 266/300
11/11 [=====] - 6s 537ms/step - loss: 0.0121 -
val_loss: 0.0121
Epoch 267/300
11/11 [=====] - 6s 541ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 268/300
11/11 [=====] - 6s 529ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 269/300
11/11 [=====] - 6s 542ms/step - loss: 0.0115 -
val_loss: 0.0121
Epoch 270/300
11/11 [=====] - 6s 567ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 271/300
11/11 [=====] - 6s 545ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 272/300
11/11 [=====] - 6s 541ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 273/300
11/11 [=====] - 6s 551ms/step - loss: 0.0121 -
val_loss: 0.0121
Epoch 274/300
11/11 [=====] - 6s 554ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 275/300
```

```
11/11 [=====] - 6s 555ms/step - loss: 0.0122 -
val_loss: 0.0121
Epoch 276/300
11/11 [=====] - 6s 546ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 277/300
11/11 [=====] - 6s 545ms/step - loss: 0.0123 -
val_loss: 0.0121
Epoch 278/300
11/11 [=====] - 6s 541ms/step - loss: 0.0116 -
val_loss: 0.0121
Epoch 279/300
11/11 [=====] - 6s 543ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 280/300
11/11 [=====] - 6s 577ms/step - loss: 0.0121 -
val_loss: 0.0121
Epoch 281/300
11/11 [=====] - 6s 547ms/step - loss: 0.0114 -
val_loss: 0.0121
Epoch 282/300
11/11 [=====] - 6s 542ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 283/300
11/11 [=====] - 6s 543ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 284/300
11/11 [=====] - 6s 545ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 285/300
11/11 [=====] - 6s 545ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 286/300
11/11 [=====] - 6s 554ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 287/300
11/11 [=====] - 6s 546ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 288/300
11/11 [=====] - 6s 539ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 289/300
11/11 [=====] - 6s 547ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 290/300
11/11 [=====] - 6s 557ms/step - loss: 0.0115 -
val_loss: 0.0121
Epoch 291/300
```

```
11/11 [=====] - 6s 542ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 292/300
11/11 [=====] - 6s 543ms/step - loss: 0.0117 -
val_loss: 0.0121
Epoch 293/300
11/11 [=====] - 6s 549ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 294/300
11/11 [=====] - 6s 550ms/step - loss: 0.0116 -
val_loss: 0.0121
Epoch 295/300
11/11 [=====] - 6s 548ms/step - loss: 0.0119 -
val_loss: 0.0121
Epoch 296/300
11/11 [=====] - 6s 549ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 297/300
11/11 [=====] - 6s 552ms/step - loss: 0.0120 -
val_loss: 0.0121
Epoch 298/300
11/11 [=====] - 6s 544ms/step - loss: 0.0121 -
val_loss: 0.0121
Epoch 299/300
11/11 [=====] - 6s 547ms/step - loss: 0.0118 -
val_loss: 0.0121
Epoch 300/300
11/11 [=====] - 6s 558ms/step - loss: 0.0119 -
val_loss: 0.0121
```

[36]: <tensorflow.python.keras.callbacks.History at 0x7f9bf5bf4950>

[37]: model.evaluate(validation_generator)

```
7/7 [=====] - 2s 270ms/step - loss: 0.0121
```

[37]: 0.012070969678461552

[38]: model.evaluate(anomaly_generator)

```
14/14 [=====] - 4s 304ms/step - loss: 0.0137
```

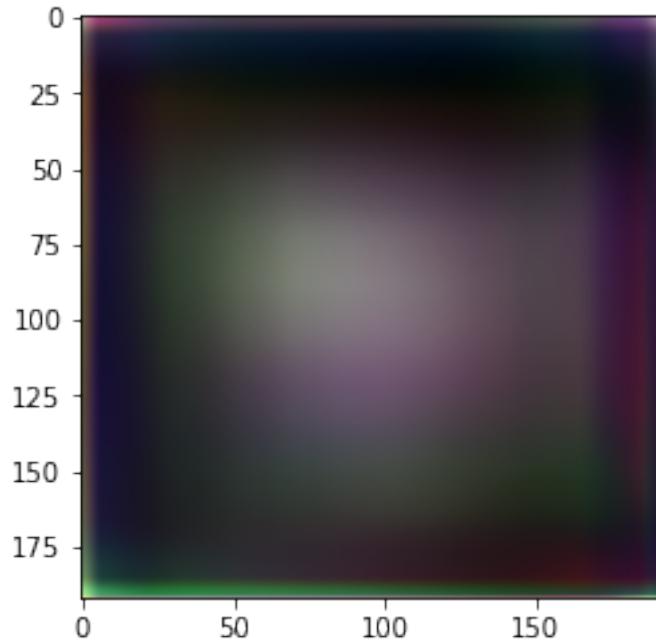
[38]: 0.013714466243982315

```
[39]: from keras.preprocessing.image import load_img, img_to_array, array_to_img

img = load_img('/content/drive/MyDrive/normalgal/normal/normal/
→1237648672922206835.jpeg', target_size=(192,192,3))
x = img_to_array(img)*1./255
```

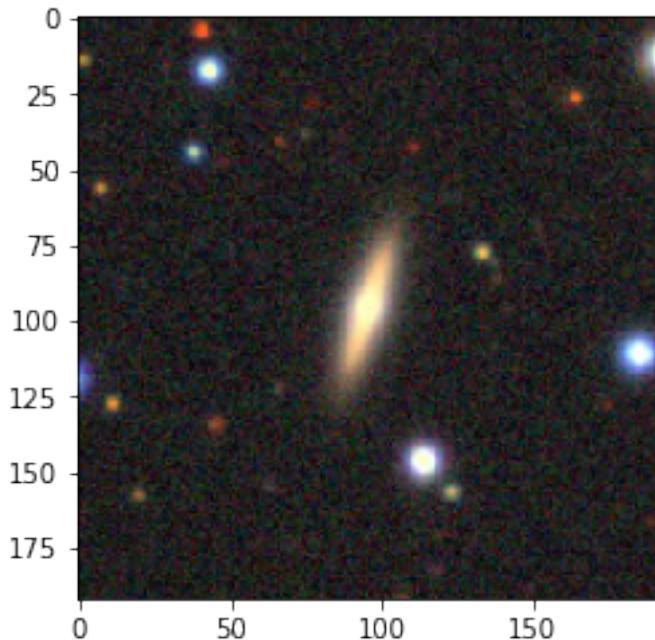
```
x = x.reshape((1,) + x.shape)
xp = model.predict(x)
plt.imshow(array_to_img(xp[0,:,:,:]))
```

[39]: <matplotlib.image.AxesImage at 0x7f9bf68f8a10>



[]: plt.imshow(array_to_img(x[0,:,:,:]))

[]: <matplotlib.image.AxesImage at 0x7fe546e17c50>



```
[40]: encoded_images = encoder.predict(train_generator)
encoded_images_flat = [np.reshape(img, (3*3*6)) for img in encoded_images]

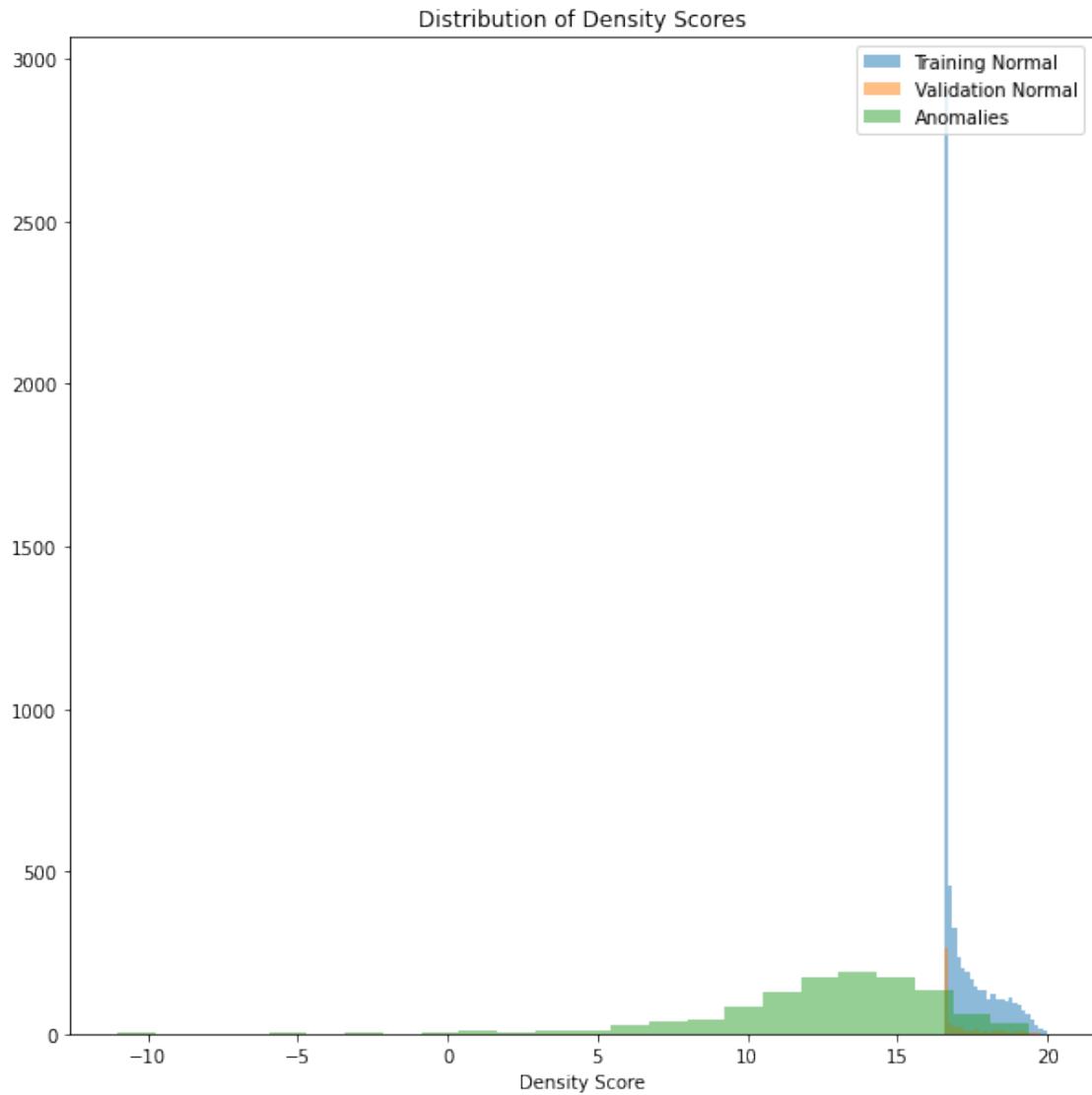
validation_encoded = encoder.predict(validation_generator)
val_enc_flat = [np.reshape(img, (3*3*6)) for img in validation_encoded]
print(np.sum(val_enc_flat[0]))
anom_encoded = encoder.predict(anomaly_generator)
anom_enc_flat = [np.reshape(img, (3*3*6)) for img in anom_encoded]
```

31.76124

```
[209]: kde = KernelDensity(kernel='gaussian', bandwidth=0.25).fit(encoded_images_flat)
training_density_scores = kde.score_samples(encoded_images_flat)
validation_density_scores = kde.score_samples(val_enc_flat)
anomaly_density_scores = kde.score_samples(anom_enc_flat)
```

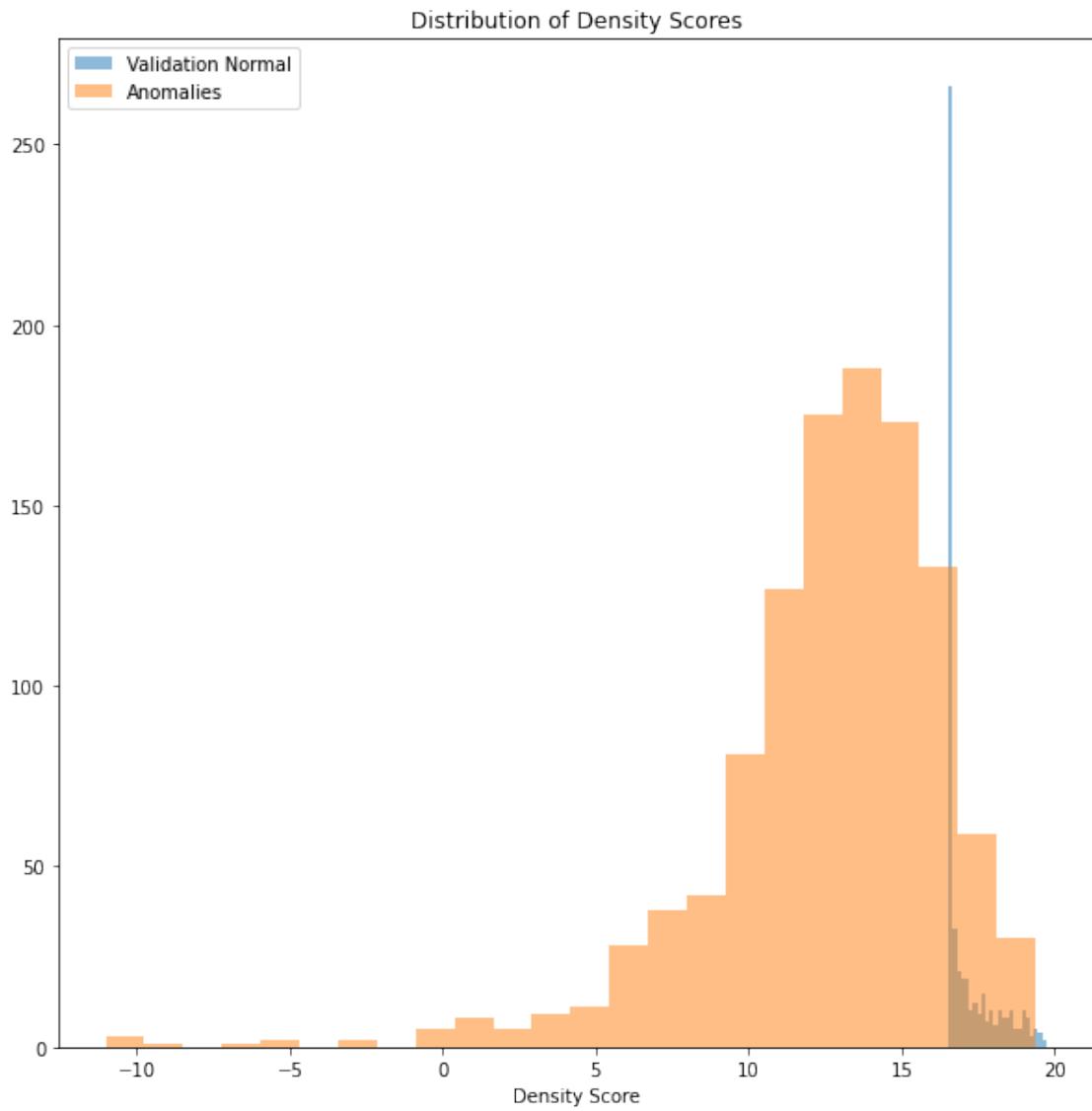
```
[210]: plt.figure(figsize = (10,10))
plt.title('Distribution of Density Scores')
plt.hist(training_density_scores, 24, alpha=0.5, label='Training Normal')
plt.hist(validation_density_scores, 24, alpha=0.5, label='Validation Normal')
plt.hist(anomaly_density_scores, 24, alpha=0.5, label='Anomalies')
plt.legend(loc='upper right')
plt.xlabel('Density Score')

plt.show()
```



```
[211]: plt.figure(figsize = (10,10))
plt.title('Distribution of Density Scores')
plt.hist(validation_density_scores, 24, alpha=0.5, label='Validation Normal')
plt.hist(anomaly_density_scores, 24, alpha=0.5, label='Anomalies')
plt.legend(loc='upper left')
plt.xlabel('Density Score')

plt.show()
```



```
[212]: x = np.zeros_like(anomaly_density_scores)
y = np.zeros_like(validation_density_scores)
z = np.zeros_like(training_density_scores)
for i in range(len(anomaly_density_scores)):
    x[i] = np.abs(anomaly_density_scores[i]-np.mean(training_density_scores))
for i in range(len(validation_density_scores)):
    y[i] = np.abs(validation_density_scores[i]-np.mean(training_density_scores))
for i in range(len(training_density_scores)):
    z[i] = np.abs(training_density_scores[i]-np.mean(training_density_scores))
```

```
[213]: A = training_density_scores[np.argmin(training_density_scores)]
A
```

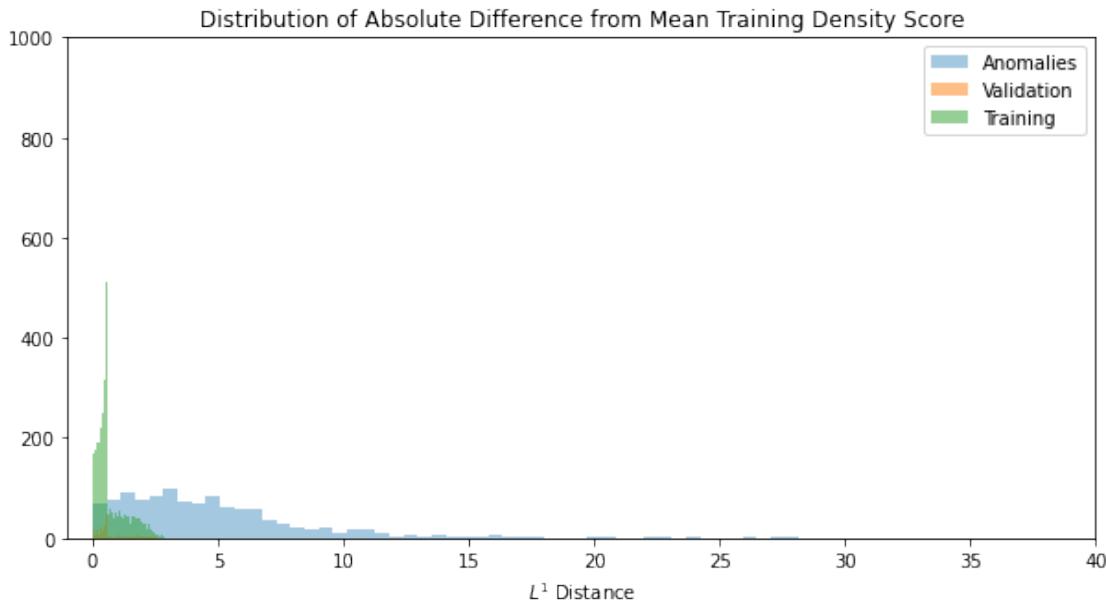
```
[213]: 16.541038860477094
```

```
[214]: anomaly_density_scores[np.argmax(anomaly_density_scores)]
```

```
[214]: 19.386055301241328
```

```
[215]: plt.figure(figsize = (10,5))
plt.title('Distribution of Absolute Difference from Mean Training Density Score')
plt.hist(x, 50, alpha=0.4, label='Anomalies')
plt.hist(y, 50, alpha=0.5, label='Validation')
plt.hist(z, 50, alpha=0.5, label='Training')
plt.ylim(-1,1000)
plt.xlim(-1,40)
plt.legend(loc='upper right')
plt.xlabel('$L^1$ Distance')
plt.show
```

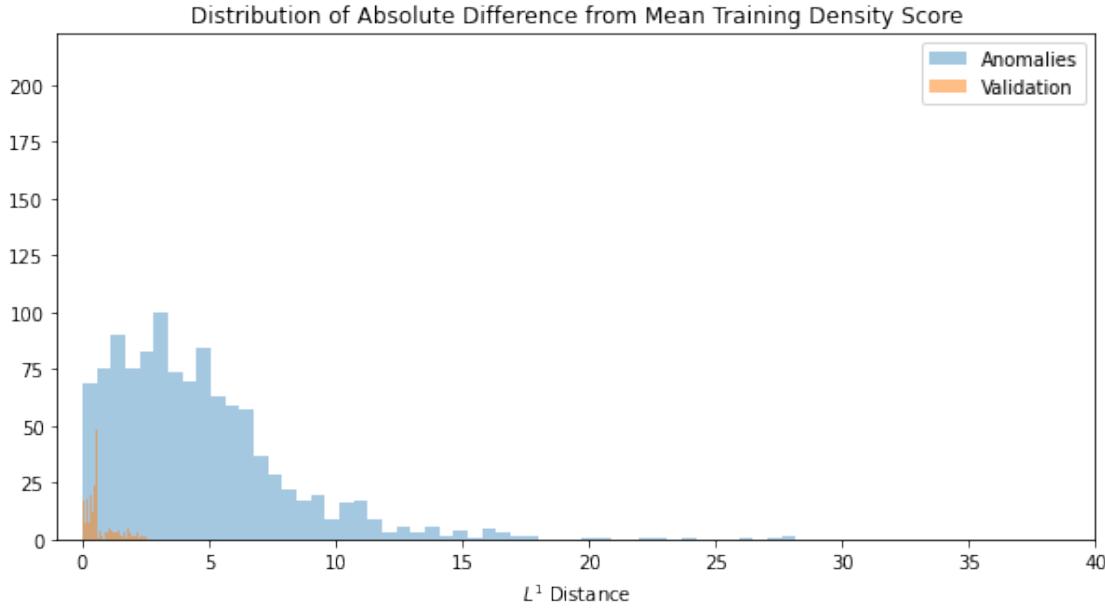
```
[215]: <function matplotlib.pyplot.show>
```



```
[216]: plt.figure(figsize = (10,5))
plt.title('Distribution of Absolute Difference from Mean Training Density Score')
plt.hist(x, 50, alpha=0.4, label='Anomalies')
plt.hist(y, 50, alpha=0.5, label='Validation')
plt.legend(loc='upper right')
plt.xlim(-1,40)
```

```
plt.xlabel('$L^1$ Distance')
plt.show
```

[216]: <function matplotlib.pyplot.show>



```
[217]: def get_mse(original, reconstruction):
    # Returns the mean square error for each image in the array
    return np.mean((original - reconstruction)**2, axis=(1,2,3))
```

```
[218]: for (dirpath, dirnames, filenames) in os.walk('/content/drive/MyDrive/normalgal/
˓→normal/normal'):
    names = [dirpath + "/" + filenames[i] for i in range(len(filenames))]
MSE_train = []
for i in range(len(names)):
    img = load_img(names[i], target_size=(192, 192, 3))
    x = img_to_array(img)*1./255
    x = x.reshape((1,) + x.shape)
    xp = model.predict(x)
    err = get_mse(x, xp)
    MSE_train.append(err)
```

```
[219]: B = np.mean(MSE_train)+2*np.std(MSE_train)
print(np.mean(MSE_train), np.std(MSE_train), MSE_train[np.argmax(MSE_train)], B)
```

0.011866686 0.005334415 [0.10531569] 0.02253551594913006

```
[220]: stuff = list(os.walk('/content/drive/MyDrive/PostMergers/Train/Data/PostMergers/'
→'))
filenames = np.array(stuff[0][2])
names = [stuff[0][0]+filenames[i] for i in range(len(filenames))]
MSE_anomaly = []
for i in range(len(names)):
    img = load_img(names[i],target_size=(192,192,3))
    x = img_to_array(img)*1./255
    x = x.reshape((1,) + x.shape)
    xp = model.predict(x)
    err = get_mse(x,xp)
    MSE_anomaly.append(err)
```

```
[221]: print(np.mean(MSE_anomaly),np.std(MSE_anomaly), MSE_anomaly[np.
→argmin(MSE_anomaly)])
```

0.013842328 0.005423177 [0.00655645]

```
[222]: def evaluate_model(test_file):
    density_threshold = A # This threshold was chosen based on looking at the
    # distribution of the density scores of the normal galaxies (either set)
    reconstruction_error_threshold = B # 2 standard deviations from the mean
    # of the training error
    batch_size = 0
    for (dirpath, dirnames, filenames) in os.walk(test_file):
        batch_size += len([file for file in filenames if file.endswith(".
→jpeg")]) # Batchsize now is total images in folder

    test_img_generator = test_datagen.flow_from_directory(
        test_file,
        target_size=(192,192),
        batch_size=batch_size,
        class_mode='input'
    )

    images = test_img_generator.next()[0]
    reconstructions = model.predict(images)
    MSEs = get_mse(images, reconstructions)
    reconstruction_anomalies = MSEs > reconstruction_error_threshold
    print(f'Reconstruction error anomalies = {sum(reconstruction_anomalies)}')

    encoded_images = encoder.predict(test_img_generator)
    encoded_images_flat = np.array([np.reshape(img, (3*3*6)) for img in
→encoded_images])
    density_scores = kde.score_samples(encoded_images_flat)
    density_anomalies = density_scores < density_threshold
    print(f'Density score anomalies = {sum(density_anomalies)}')
```

```
combined_anomalies = np.maximum(reconstruction_anomalies, density_anomalies)
print(f'Maximum potential anomalies = {sum(combined_anomalies)}')
overall_detection_rate = sum(combined_anomalies) / len(combined_anomalies)
print(f'Overall detection rate = {overall_detection_rate}')
```

[223]: evaluate_model('/content/drive/MyDrive/NormalVal/')

```
Found 505 images belonging to 2 classes.
Reconstruction error anomalies = 20
Density score anomalies = 4
Maximum potential anomalies = 20
Overall detection rate = 0.039603960396039604
```

[224]: evaluate_model('/content/drive/MyDrive/PostMergers/Train/Data/')

```
Found 1001 images belonging to 1 classes.
Reconstruction error anomalies = 49
Density score anomalies = 889
Maximum potential anomalies = 889
Overall detection rate = 0.8881118881118881
```

[225]: evaluate_model('/content/drive/MyDrive/PostMergers/Validation/')

```
Found 120 images belonging to 2 classes.
Reconstruction error anomalies = 2
Density score anomalies = 110
Maximum potential anomalies = 110
Overall detection rate = 0.9166666666666666
```

[226]: evaluate_model('/content/drive/MyDrive/TestNormal/')

```
Found 305 images belonging to 1 classes.
Reconstruction error anomalies = 9
Density score anomalies = 3
Maximum potential anomalies = 9
Overall detection rate = 0.029508196721311476
```

[228]: Total = 505 + 305 + 1001 + 120
TP = 889 + 110
TN = 505-4 + 305-3
FN = 120-110 + 1001-889
FP = 7

```
Accuracy = (TP + TN)/Total
Precision = TP/(TP + FP)
Recall = TP/(TP+FN)
F1_score = 2/(1/Precision+1/Recall)
print("accuracy:", Accuracy, "F1 Score:", F1_score)
```

accuracy: 0.9331952356292077 F1 Score: 0.9393511988716501