landCov_semSeg

May 3, 2022

1 Land-use classification using Semantic Segmentation

Johnathan Clementi & Gianluca Mangiapane
Remote Sensing: MUSA 650, Spring 2022

1.0.1 Problem [UPDATE]:

In image analysis, traditional classification methods return a single output label for each image. In the context of land-use classification from satellite/aerial imagery, a traditional classification model would only return a single label for an image that has many land-use types within it. However, there are often many different types of land-use held within a single satellite or aerial image. Therefore, an issue may arise when trying to understand the distribution of certain types of land use within a single image. If there are multiple land-use types and the the classifier will assign that image a single label, the remaining land-use data that doesn't fall under the single-classification is then hidden from the user, which could be useful or valuable information. Thus, we want to utilize semantic segmentation, or the classification of individual pixels (and therefore all the different classes) within an image. NEW: The problem we aim to answer is being able to identify all different landcover types in a satellite image, effectively through a single neural network model.

1.0.2 Methods

The primary method that we aim to use is that of Semantic Segmentation, a form of dense prediction where every pixel in an image is given a label of a corresponding class. Semantic Segmentation has become prevalent in both the medical field as a way of identifying harmful cells from healthy cells in brain and organ scans, and autonomous driving cars being able to idenfity free road spcae from other vehicles, pedestrians, and road signs. However, there has also been a growing application of Semantic Segmentation to the field of Remote Sensing, and identifying different classes of land use and objects from satellite imagery.

Our segmentation method will consist of applying a U-net on our image dataset. A U-net initially follows a general CNN architecture of convolutional and pooling layers, but it starts with encoding the input image down to a simplied features map, and then decodes that map back up to the input image, through deconvolutional layers, and therefore becomes a fully convolutional network. A defining feature of the U-net is that at every up-sampling of the decoder layer (the upwards of the U), information is sent from its respective down-sampling of the encoder layer. The encoder layer has more defined information, and therefore helps the decoder layer have more accurate outputs.

Considering Image Segmentation is a very complicated process in and of itself, if we run into difficulties with the image dataset's current form, one back up proposal is that we can split individual images into squares of equal size, and decrease the number of different classes in each sqaure. From there, we can perform a more mangeable multi-classification approach, such as a traditional CNN with convulutional and pooling layers instead of a U-NET.

1.0.3 Data and Image Feature Engineering

We used Earth Imagery collected by the DigitialGlobe satellite in 2018, and presented as a challenge to "Parse the Earth through Satellite Images" by Cornell University. The dataset was also provided on Kaggle.

-Challenge Information https://arxiv.org/abs/1805.06561

-Kaggle Link https://www.kaggle.com/datasets/balraj98/deepglobe-land-cover-classification-dataset

The training data contains 803 satellite imagery in RGB, size 2448 x 2448, with pixel resolution of 50cm. The dataset also contains 171 validation and 172 test images, with no masks. We realized the validation and test images do not have masks due to the challenge/competition aspect of this dataset when first presented. Since there are 803 unique images in the train dataset, we will split our own test/train dataset just using the 803 unique images.

In the masks for the training set, each of the 7 classes has a corresponding RBG value as follows: Urban (0,255,255), Agriculture land (255,255,0), Rangeland (255,0,255), Forest land (0,255,0), Water (0,0,255), Barren land (255,255,255), and Unknown (0,0,0) i.e., clouds and others.

We identified three feature engineering steps that were necessary before the modeling stage,

1. For an image to be properly fed through the structure of a U-net, Ronneberger et Al (2015) explained that the dimension sizes need to be divisible by 32. Since our image dimensions are sqaures of 2448 H x W, and thus not divisible by 32, we need to trim them to the correct U-net dimensions, along with the masks as well.

2. Each image has pixel dimension of 2448 x 2448, which are extremely high resolution and detailed images. Running images of this resolution through a neural network will require large computational power. Thus, we need to crop individual images into smaller sizes, along with their respective mask.

3. Each mask has pixels that are assigned to different classes of RGB channel configuration, as outlined above. We need to one-hot encode every pixel in each mask since this is a multi-class categorical problem, and then reshape the masks so that their RGB channel class changes from ([...], [...], 3) to ([...], [...], 7), with 7 being the number of possible classes a pixel could be labled.

For the first step, we trimmed 200 pixels from each edge of an image and its respective mask, so that a 2448 x 2448 image (and mask) was then 2048 x 2048, which solved our first requirement that the dimensions needeed to be divisibile by 32. We then cropped each image into 4 equal sized images of 512 x 512 pixels. Images of this size are much more feasible to run through our google collab GPU. An example of how an image and its mask was cropped is illustrated below,

1.0.4 Methods

2 Progress Report 4/22/2022

3 Current Status:

We have our total dataset divided into a 90/10 test and train split of 723 train images, and 80 test images. Each image has a corresponding mask associated with it. We have sampled a random image to assess image shape and features of the dataset. We've identified two issues in this current set up, 1. Each image has pixel dimension of 2448 x 2448, which are extremely high resolution and detailed images. Running images of this resolution through a neural network will require large computational power 2. Total dataset is only 803 images, and a model trained on too little data can potentially underperform. In addition, too little test data can present over-optimitic and results with a high variance.

In order to address these issues, we've decided to crop each existing image into smaller sized images of 153 x 153 pixels, so from a single image of size 2448 x 2448, we now have 64 new images for each original image. This resulted in a new dataset with over 200,000 images to work with, along with their respective cropped masks. This will also make it easier to manage and build neural networks with images of smaller dimensionality, and help make sure computational power isn't reduced. We're using an amended version of Jerin Paul's Skeyenet step2_build_data.py script to perform the cropping operations.

4 Hurdles to overcome

Something we've realized in this method however, is dealing with cropped images that will now be overwhelmingly just a single class, i.e., entirely barren_land, or forest_land. This will result in their respective cropped masks being just a single class value. We will need to decide on the percentage threshold of class value in masks that will make the most sense. For example if a mask is composed of 95%, 99% or 100% of a single class, they will need to set aside from the model.

5 Next Steps

Once we've filtered out the cropped images with a single class, we will look at keras tuners and grid Search CV in order to tune the hyperparameters of our U-net model. GridSearch CV is the brute force method, where you give it a list of hyperparameters to test, and it does that for you.

Keras Tuners will adjust hyperparamters it is tuning on, based on defined optimization functions.

Considering a U-net's architecture is defined through keras, finding the right hyperparamters will be crucial to making an accurate and well-performing network.

Keras tuners have three different types to use, 1. Hyperband - combo of random search with adaptation and early stopping 2. Random 3. Bayesian optimization - gaussian method

Keras Tuner's allow for exisitng weights from pre-trained U-net models to be added in for hyperparamter shaping, as a foundation to build a new model upon. Some pre-exisiting U-Net architecture that we will investigate further is that of Attention U-Net, U-Net Plus Plus, and R2-U-Net. These U-net models seem to be active in the medical field, so we are keen to see if the same practicality can be applied to land cover identification.

5.0.1 Sources

https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47

https://nanonets.com/blog/semantic-image-segmentation-2020/

https://keras.io/examples/vision/oxford_pets_image_segmentation/

https://github.com/MUSA-650/Spring2022-Week13/blob/master/DataPrepExample_Skeyenet/Src/step2_build_

https://neptune.ai/blog/keras-tuner-tuning-hyperparameters-deep-learning-model

https://github.com/LeeJunHyun/Image_Segmentation

[ ]:
```
gpu_info = !nvidia-smi
gpu_info = '\n'.join(gpu_info)
if gpu_info.find('failed') >= 0:
  print('Not connected to a GPU')
else:
  print(gpu_info)
```

[ ]:
```
from psutil import virtual_memory
ram_gb = virtual_memory().total / 1e9
print('Your runtime has {:.1f} gigabytes of available RAM\n'.format(ram_gb))

if ram_gb < 20:
  print('Not using a high-RAM runtime')
else:
  print('You are using a high-RAM runtime!')
```

[ ]:
```
#!pip install tensorflow
```

[ ]:
```
## import packages
import os
import numpy as np
import pandas as pd
import math
import random, tqdm
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
%matplotlib inline
```

```
import PIL
import PIL.Image
import cv2

import warnings
warnings.filterwarnings("ignore")
```

[ ]:
```
# Import functions from Kaggle example code:

# helper function for data visualization
def visualize(**images):
    """
    Plot images in one row
    """
    n_images = len(images)
    plt.figure(figsize=(16,16))
    for idx, (name, image) in enumerate(images.items()):
        plt.subplot(1, n_images, idx + 1)
        # plt.xticks([]);
        # plt.yticks([])
        # get title from the parameter names
        plt.title(name.replace('_',' ').title(), fontsize=20)

        if type(image) == str:
            image = mpimg.imread(image)

        plt.imshow(image)
    plt.show()


# See https://stackoverflow.com/questions/43884463/
 ↪how-to-convert-rgb-image-to-one-hot-encoded-3d-array-based-on-color-using-numpy
def rgb_to_onehot(rgb_arr, color_dict):
    num_classes = len(color_dict)
    shape = rgb_arr.shape[:2]+(num_classes,)
    arr = np.zeros( shape, dtype=np.int8 )
    for i, cls in enumerate(color_dict):
        arr[:,:,i] = np.all(rgb_arr.reshape( (-1,3) ) == color_dict[i], axis=1).
 ↪reshape(shape[:2])
    return arr


def onehot_to_rgb(onehot, color_dict):
    single_layer = np.argmax(onehot, axis=-1)
    output = np.zeros( onehot.shape[:2]+(3,) )
    for k in color_dict.keys():
        output[single_layer==k] = color_dict[k]
```

```
    return np.uint8(output)
```

[1]:
```
## Mount drive folder
from google.colab import drive

drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[ ]:
```
## Create path to data

# Gianluca's Path:
# path = "/content/drive/MyDrive/LULC_FinalProject/data"

## Johnathan's Path:
path = "/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/
 ↪650_RemoteSensing/LULC_FinalProject"


## Local path
# path = os.getcwd()

## Define path to the data
path = "{}/data".format(path)
```

[ ]:
```
## Classes and their respective pixel values
class_dict_pd = pd.read_csv('{}/class_dict.csv'.format(path))
class_names = class_dict_pd['name'].values.tolist()
class_rgb_vals = class_dict_pd[['r', 'g', 'b']].values.tolist()

# Useful to shortlist specific classes in datasets with large number of classes
select_classes = class_names

# Get RGB values of required classes
select_class_indices = [class_names.index(cls.lower()) for cls in␣
 ↪select_classes]
select_class_rgb_values =  np.array(class_rgb_vals)[select_class_indices]

print('Selected classes and their corresponding RGB values in labels:')
print('Class Names: ', class_names)
print('Class RGB values: ', class_rgb_vals)
```

```
Selected classes and their corresponding RGB values in labels:
Class Names:  ['urban_land', 'agriculture_land', 'rangeland', 'forest_land',
'water', 'barren_land', 'unknown']
Class RGB values:  [[0, 255, 255], [255, 255, 0], [255, 0, 255], [0, 255, 0],
[0, 0, 255], [255, 255, 255], [0, 0, 0]]
```

[ ]:
```
# Create a dictionary for One-Hot Encoding
class_dict =  dict(zip(select_class_indices, class_rgb_vals))
class_dict
```

[ ]:
```
{0: [0, 255, 255],
 1: [255, 255, 0],
 2: [255, 0, 255],
 3: [0, 255, 0],
 4: [0, 0, 255],
 5: [255, 255, 255],
 6: [0, 0, 0]}
```

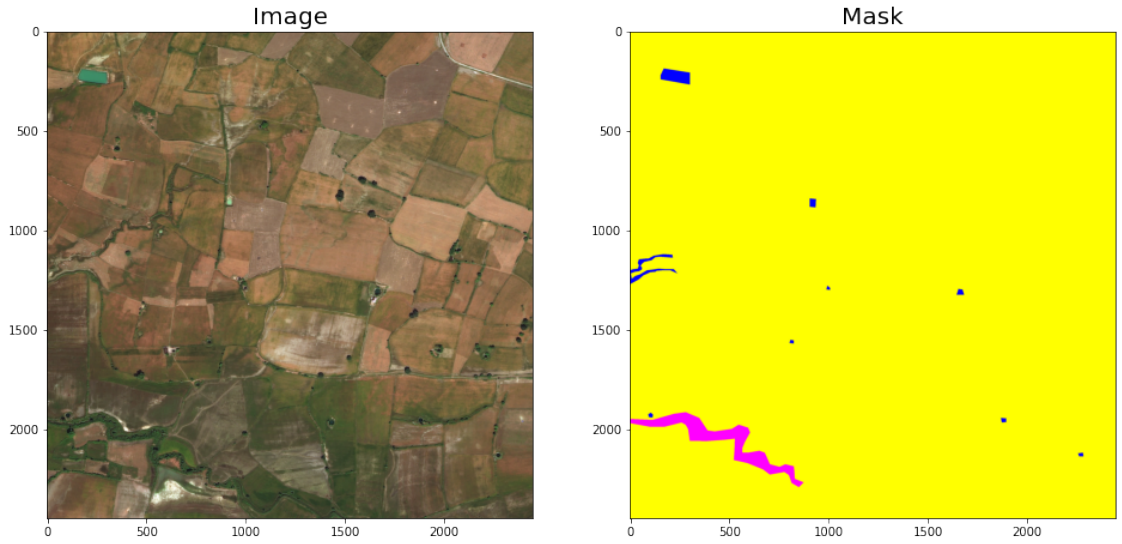
[ ]:
```
# Load metadata file - contains relative paths for images and masks
metadata = pd.read_csv('{}/metadata.csv'.format(path))

# Preparing metadata for use
metadata = metadata[metadata['split']=='train'] # Filter out images that do not␣
 ↪have masks (those images are part of the challenge set)
metadata = metadata[['image_id', 'sat_image_path', 'mask_path']] # Remove the␣
 ↪image status column

# set paths to absolute paths rather than relative paths:
metadata['sat_image_path'] = metadata['sat_image_path'].apply(lambda img_pth:␣
 ↪os.path.join(path, img_pth))
metadata['mask_path'] = metadata['mask_path'].apply(lambda img_pth: os.path.
 ↪join(path, img_pth))
```

[ ]:
```
metadata.sort_values('image_id')
```

[ ]:
```
     image_id                                    sat_image_path  \
15        119  /content/drive/MyDrive/Grad School/Penn_MUSA/S…
144       266  /content/drive/MyDrive/Grad School/Penn_MUSA/S…
456       606  /content/drive/MyDrive/Grad School/Penn_MUSA/S…
676       855  /content/drive/MyDrive/Grad School/Penn_MUSA/S…
120      2334  /content/drive/MyDrive/Grad School/Penn_MUSA/S…
..        …                                                 …
798    992507  /content/drive/MyDrive/Grad School/Penn_MUSA/S…
799    994520  /content/drive/MyDrive/Grad School/Penn_MUSA/S…
800    995492  /content/drive/MyDrive/Grad School/Penn_MUSA/S…
801    997521  /content/drive/MyDrive/Grad School/Penn_MUSA/S…
802    998002  /content/drive/MyDrive/Grad School/Penn_MUSA/S…

                                          mask_path
15   /content/drive/MyDrive/Grad School/Penn_MUSA/S…
144  /content/drive/MyDrive/Grad School/Penn_MUSA/S…
456  /content/drive/MyDrive/Grad School/Penn_MUSA/S…
676  /content/drive/MyDrive/Grad School/Penn_MUSA/S…
```

```
120  /content/drive/MyDrive/Grad School/Penn_MUSA/S…
..                                                 …
798  /content/drive/MyDrive/Grad School/Penn_MUSA/S…
799  /content/drive/MyDrive/Grad School/Penn_MUSA/S…
800  /content/drive/MyDrive/Grad School/Penn_MUSA/S…
801  /content/drive/MyDrive/Grad School/Penn_MUSA/S…
802  /content/drive/MyDrive/Grad School/Penn_MUSA/S…

[803 rows x 3 columns]
```

[ ]:
```
img_Loc = metadata['sat_image_path'][1]
mask_Loc = metadata['mask_path'][1]

img = cv2.imread(img_Loc)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

visualize(Image = img, Mask = mask_Loc)
```



[ ]:
```
# img = PIL.Image.open(mask_Loc)
# uni, freq = np.unique(np.asarray(img), return_counts=True)

# print('Unique Vals: {}'.format(uni))
# print('Freq of vals: {}'.format(freq))
```

6 Data Ingestion and Augmentation

As mentioned above, it is advantageous to use smaller images when modeling. The code below allows us to crop our images and is borrowed and ammended from Jerin Paul. When run, the code iterates through images and corresponding masks in two separate directories, for each image-mask pair, the image and mask are opened, then a cropping filter iterates across each row and column to extract a subset image.

[ ]:
```
# Indentify the number of satellite images and corresponding masks
numSatImg = len(metadata['sat_image_path'])
numMask = len(metadata['mask_path'])
print("Number of original satellite images: " + str(numSatImg))
print("Number of original masks: " + str(numMask))

# Open example image & mask
img = PIL.Image.open(metadata['sat_image_path'][0])
mask = PIL.Image.open(metadata['mask_path'][0])
# Get dimensionality of that image & mask
numPixels = np.asarray(img).shape
numMaskPx = np.asarray(mask).shape
print('Number of pixels in each image: {}'.format(numPixels))
print('Number of pixels in each masks: {}'.format(numMaskPx))
```

```
Number of original satellite images: 803
Number of original masks: 803
Number of pixels in each image: (2448, 2448, 3)
Number of pixels in each masks: (2448, 2448, 3)
```

[ ]:
```
# Initialize arrays for cropped sat images and masks
# First we will need to trim the 2448 x 2448 image to 2048 x 2048 image so that␣
 ↪it can be used with U-net
# which needs to have images with sizes divisible by 32
# We will then crop each image down to sections of 128 x 128 x 3, which␣
 ↪produces 256 cropped images per original sat image.
# If you want to change the size of the cropped image, change the denominator␣
 ↪for cropimg width / height below
# cropImg_height = int((numPixels[0]-400) / 16) # 2048 / 16 = 153
# cropImg_width = int((numPixels[1]-400) / 16) # 2048 / 16 = 153


# X = np.zeros([(numSatImg * 256), cropImg_height, cropImg_width,␣
 ↪numPixels[2]], dtype='uint8')
# y = np.zeros([(numMask * 256), cropImg_height, cropImg_width, numPixels[2]],␣
 ↪dtype='uint8')
# print('Shape of cropped sat image dataset: {}'.format(X.shape))
# print('Shape of cropped mask dataset: {}'.format(y.shape))
```

```
cropImg_height = int((numPixels[0]-400) / 4) # 2048 / 4 = 512
cropImg_width = int((numPixels[1]-400) / 4) # 2048 / 4 = 512

X = np.zeros([(numSatImg * 16), cropImg_height, cropImg_width, numPixels[2]],␣
 ↪dtype='uint8')
y = np.zeros([(numMask * 16), cropImg_height, cropImg_width, len(class_dict)],␣
 ↪dtype='uint8')
print('Shape of cropped sat image dataset: {}'.format(X.shape))
print('Shape of cropped mask dataset: {}'.format(y.shape))
```

```
Shape of cropped sat image dataset: (12848, 512, 512, 3)
Shape of cropped mask dataset: (12848, 512, 512, 7)
```

[ ]:
```
# trimm = np.array(img[200:2248, 200:2248, :])
# visualize(og_img = img, trimmed = trimm)
```

[ ]:
```
# Extract paths from pandas df to np array for iteration purposes - don't want␣
 ↪to use iterrows b/c its slow!
Xrows = np.asarray(metadata['sat_image_path'])
yrows = np.asarray(metadata['mask_path'])


# Step 0: Trim images from 2448x2448 to 2048x24048 to make the image size␣
 ↪divisible by 32 for Unet purposes
# Step 1: Iterate through image & corresponding mask paths and read images into␣
 ↪memory
# Step 2: iterate through original img matrix and crop to predefined crop␣
 ↪height & width
# Step 3: Save cropped matrix to working dataset

cropImgIdx = 0

# for i in range(0, len(Xrows)):
for i in range(0, 200):

  # Read and normalize image
  img = np.asarray(cv2.imread(Xrows[i]))
  mask = np.asarray(cv2.imread(yrows[i]))

  # Trim image to [2048,2048,3] by trimming extra 200px off from the border
  img = np.array(img[200:2248, 200:2248, :])
  mask = np.array(mask[200:2248, 200:2248, :])

  # Iterate through each row
  for r in range(0, img.shape[0], cropImg_height):
    # Iterate through each column
    for c in range(0, img.shape[1], cropImg_width):
```

```
      # Slice mask by cropping window first
      # That way we can check if if we're going to use the image or not
      newMask = np.array(mask[r:r+cropImg_height, c:c+cropImg_width, :])
      newMask = cv2.cvtColor(newMask, cv2.COLOR_BGR2RGB)

      # Convert mask to grayscale to find distribution of classes
      grayMask = cv2.cvtColor(newMask, cv2.COLOR_BGR2GRAY)
      # Get frequency of each classification in the cropped mask
      unique, frequency = np.unique(grayMask, return_counts= True)
      frequency = frequency / (len(grayMask.flatten()))

      # Check if any classes are represent 99% of image
      # If that is the case, throw it out
      if (frequency >= 0.99).any():
        continue

      # Try one-hot encoding of mask here
      newMask = rgb_to_onehot(newMask, class_dict)

      # Crop image if we make it past the majority class checker
      newImg = np.array(img[r:r+cropImg_height, c:c+cropImg_width, :])
      newImg = cv2.cvtColor(newImg, cv2.COLOR_BGR2RGB)

      X[cropImgIdx,:] = newImg
      y[cropImgIdx,:] = newMask

      cropImgIdx += 1

print('There were {} images cropped from {} original images'.
 ↪format((cropImgIdx-1), numSatImg))
```

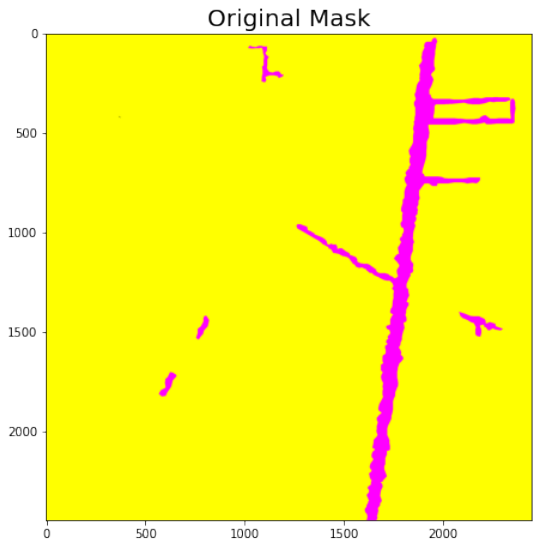
```
There were 1923 images cropped from 803 original images
```

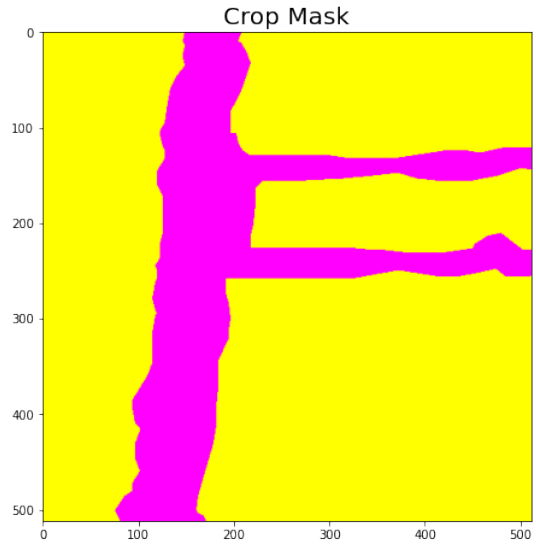
[ ]:
```
cropImgIdx -= 1

X = X[0:cropImgIdx, :,:,:]
y = y[0:cropImgIdx, :,:,:]

print(X.shape)
print(y.shape)
```

```
(1923, 512, 512, 3)
(1923, 512, 512, 7)
```

[ ]:
```
ogImg = Xrows[0]
ogmask = yrows[0]
```

```
cropImg = X[0]
cropMsk = onehot_to_rgb(y[0], class_dict)

visualize(original_Image = ogImg, original_Mask = ogmask)
visualize(crop_Image = cropImg, crop_Mask = cropMsk)
```





[ ]:
```
## Check that values are reading correctly
# print("original Image:")
# print(np.asarray(cv2.imread(Xrows[0]))[:153,:153,:])
# print("cropped Image:")
```

```
# print(X[0,:,:,:])

# print("original mask:")
# print(np.asarray(cv2.imread(yrows[0]))[:153,:153,:])
# print("cropped mask:")
# print(y[0,:,:,:])
```

7 U-Net model building

[ ]:
```
from sklearn.model_selection import train_test_split

import tensorflow as tf
from tensorflow import math
from tensorflow.keras.layers import Input, Conv2D, BatchNormalization,␣
↪Activation, MaxPool2D, UpSampling2D, Concatenate
from keras.models import Sequential, Model, load_model
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam
from keras.utils.vis_utils import plot_model
from keras.callbacks import EarlyStopping, ReduceLROnPlateau, ModelCheckpoint


# Pre-trained U-net in Keras: https://pypi.org/project/keras-unet/
!pip install keras-unet
from keras_unet.models import vanilla_unet, custom_unet
from keras_unet.metrics import iou, iou_thresholded
from keras_unet.losses import jaccard_distance
from keras_unet.utils import plot_segm_history

# KerasTuner info: https://keras.io/guides/keras_tuner/getting_started/
!pip install keras-tuner
import keras_tuner as kt
from kerastuner.applications import HyperResNet
from kerastuner.tuners import Hyperband, RandomSearch

# Segmentation Models info: https://segmentation-models.readthedocs.io/en/
↪latest/install.html
#!pip install segmentation-models
import segmentation_models as sm
```

```
Requirement already satisfied: keras-unet in /usr/local/lib/python3.7/dist-
packages (0.1.2)
----------------------------------------
keras-unet init: TF version is >= 2.0.0 - using `tf.keras` instead of `Keras`
----------------------------------------
```

```
Requirement already satisfied: keras-tuner in /usr/local/lib/python3.7/dist-
packages (1.1.2)
Requirement already satisfied: ipython in /usr/local/lib/python3.7/dist-packages
(from keras-tuner) (5.5.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages
(from keras-tuner) (1.21.6)
Requirement already satisfied: kt-legacy in /usr/local/lib/python3.7/dist-
packages (from keras-tuner) (1.0.4)
Requirement already satisfied: tensorboard in /usr/local/lib/python3.7/dist-
packages (from keras-tuner) (2.8.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-
packages (from keras-tuner) (21.3)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-
packages (from keras-tuner) (2.23.0)
Requirement already satisfied: pygments in /usr/local/lib/python3.7/dist-
packages (from ipython->keras-tuner) (2.6.1)
Requirement already satisfied: pexpect in /usr/local/lib/python3.7/dist-packages
(from ipython->keras-tuner) (4.8.0)
Requirement already satisfied: setuptools>=18.5 in
/usr/local/lib/python3.7/dist-packages (from ipython->keras-tuner) (57.4.0)
Requirement already satisfied: prompt-toolkit<2.0.0,>=1.0.4 in
/usr/local/lib/python3.7/dist-packages (from ipython->keras-tuner) (1.0.18)
Requirement already satisfied: simplegeneric>0.8 in
/usr/local/lib/python3.7/dist-packages (from ipython->keras-tuner) (0.8.1)
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.7/dist-
packages (from ipython->keras-tuner) (5.1.1)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.7/dist-
packages (from ipython->keras-tuner) (0.7.5)
Requirement already satisfied: decorator in /usr/local/lib/python3.7/dist-
packages (from ipython->keras-tuner) (4.4.2)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.7/dist-packages
(from prompt-toolkit<2.0.0,>=1.0.4->ipython->keras-tuner) (0.2.5)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.7/dist-
packages (from prompt-toolkit<2.0.0,>=1.0.4->ipython->keras-tuner) (1.15.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/usr/local/lib/python3.7/dist-packages (from packaging->keras-tuner) (3.0.8)
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.7/dist-
packages (from pexpect->ipython->keras-tuner) (0.7.0)
Requirement already satisfied: chardet<4,>=3.0.2 in
/usr/local/lib/python3.7/dist-packages (from requests->keras-tuner) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.7/dist-packages (from requests->keras-tuner) (1.24.3)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-
packages (from requests->keras-tuner) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.7/dist-packages (from requests->keras-tuner) (2021.10.8)
Requirement already satisfied: wheel>=0.26 in /usr/local/lib/python3.7/dist-
packages (from tensorboard->keras-tuner) (0.37.1)
```

```
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in
/usr/local/lib/python3.7/dist-packages (from tensorboard->keras-tuner) (0.4.6)
Requirement already satisfied: grpcio>=1.24.3 in /usr/local/lib/python3.7/dist-
packages (from tensorboard->keras-tuner) (1.44.0)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/usr/local/lib/python3.7/dist-packages (from tensorboard->keras-tuner) (1.35.0)
Requirement already satisfied: protobuf>=3.6.0 in /usr/local/lib/python3.7/dist-
packages (from tensorboard->keras-tuner) (3.17.3)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
/usr/local/lib/python3.7/dist-packages (from tensorboard->keras-tuner) (1.8.1)
Requirement already satisfied: absl-py>=0.4 in /usr/local/lib/python3.7/dist-
packages (from tensorboard->keras-tuner) (1.0.0)
Requirement already satisfied: werkzeug>=0.11.15 in
/usr/local/lib/python3.7/dist-packages (from tensorboard->keras-tuner) (1.0.1)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in
/usr/local/lib/python3.7/dist-packages (from tensorboard->keras-tuner) (0.6.1)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-
packages (from tensorboard->keras-tuner) (3.3.6)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.7/dist-
packages (from google-auth<3,>=1.6.3->tensorboard->keras-tuner) (4.8)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in
/usr/local/lib/python3.7/dist-packages (from google-
auth<3,>=1.6.3->tensorboard->keras-tuner) (4.2.4)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/usr/local/lib/python3.7/dist-packages (from google-
auth<3,>=1.6.3->tensorboard->keras-tuner) (0.2.8)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/usr/local/lib/python3.7/dist-packages (from google-auth-
oauthlib<0.5,>=0.4.1->tensorboard->keras-tuner) (1.3.1)
Requirement already satisfied: importlib-metadata>=4.4 in
/usr/local/lib/python3.7/dist-packages (from
markdown>=2.6.8->tensorboard->keras-tuner) (4.11.3)
Requirement already satisfied: typing-extensions>=3.6.4 in
/usr/local/lib/python3.7/dist-packages (from importlib-
metadata>=4.4->markdown>=2.6.8->tensorboard->keras-tuner) (4.2.0)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-
packages (from importlib-metadata>=4.4->markdown>=2.6.8->tensorboard->keras-
tuner) (3.8.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in
/usr/local/lib/python3.7/dist-packages (from pyasn1-modules>=0.2.1->google-
auth<3,>=1.6.3->tensorboard->keras-tuner) (0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-
packages (from requests-oauthlib>=0.7.0->google-auth-
oauthlib<0.5,>=0.4.1->tensorboard->keras-tuner) (3.2.0)
Segmentation Models: using `keras` framework.
```

[ ]:
```
# Build UNet Architecture

def conv_block(inputs, filters, pool=True):
    x = Conv2D(filters, 3, padding="same")(inputs)
    x = BatchNormalization()(x)
    x = Activation("relu")(x)

    x = Conv2D(filters, 3, padding="same")(x)
    x = BatchNormalization()(x)
    x = Activation("relu")(x)

    if pool == True:
        p = MaxPool2D((2, 2))(x)
        return x, p
    else:
        return x

def build_unet(shape, num_classes):
    inputs = Input(shape)

    """ Encoder """
    x1, p1 = conv_block(inputs, 16, pool=True)
    x2, p2 = conv_block(p1, 32, pool=True)
    x3, p3 = conv_block(p2, 48, pool=True)
    x4, p4 = conv_block(p3, 64, pool=True)

    """ Bridge """
    b1 = conv_block(p4, 128, pool=False)

    """ Decoder """
    u1 = UpSampling2D((2, 2), interpolation="bilinear")(b1)
    c1 = Concatenate()([u1, x4])
    x5 = conv_block(c1, 64, pool=False)

    u2 = UpSampling2D((2, 2), interpolation="bilinear")(x5)
    c2 = Concatenate()([u2, x3])
    x6 = conv_block(c2, 48, pool=False)

    u3 = UpSampling2D((2, 2), interpolation="bilinear")(x6)
    c3 = Concatenate()([u3, x2])
    x7 = conv_block(c3, 32, pool=False)

    u4 = UpSampling2D((2, 2), interpolation="bilinear")(x7)
    c4 = Concatenate()([u4, x1])
    x8 = conv_block(c4, 16, pool=False)

    """ Output layer """
```

```
    output = Conv2D(num_classes, 1, padding="same", activation="softmax")(x8)

    return Model(inputs, output)

if __name__ == "__main__":
    model = build_unet((512, 512, 3), 7)
    model.summary()
```

```
Model: "model"
__________________________________________________________________________________
__________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================
==================
 input_1 (InputLayer)           [(None, 512, 512, 3  0           []
                                )]

 conv2d (Conv2D)                (None, 512, 512, 16  448
['input_1[0][0]']
                                )

 batch_normalization (BatchNorm  (None, 512, 512, 16  64
['conv2d[0][0]']
 alization)                     )

 activation (Activation)        (None, 512, 512, 16  0
['batch_normalization[0][0]']
                                )

 conv2d_1 (Conv2D)              (None, 512, 512, 16  2320
['activation[0][0]']
                                )

 batch_normalization_1 (BatchNo  (None, 512, 512, 16  64
['conv2d_1[0][0]']
 rmalization)                   )

 activation_1 (Activation)      (None, 512, 512, 16  0
['batch_normalization_1[0][0]']
                                )

 max_pooling2d (MaxPooling2D)   (None, 256, 256, 16  0
['activation_1[0][0]']
                                )

 conv2d_2 (Conv2D)              (None, 256, 256, 32  4640
['max_pooling2d[0][0]']
```

```
                                )

 batch_normalization_2 (BatchNo  (None, 256, 256, 32  128
['conv2d_2[0][0]']
 rmalization)                   )

 activation_2 (Activation)      (None, 256, 256, 32  0
['batch_normalization_2[0][0]']
                                )

 conv2d_3 (Conv2D)              (None, 256, 256, 32  9248
['activation_2[0][0]']
                                )

 batch_normalization_3 (BatchNo  (None, 256, 256, 32  128
['conv2d_3[0][0]']
 rmalization)                   )

 activation_3 (Activation)      (None, 256, 256, 32  0
['batch_normalization_3[0][0]']
                                )

 max_pooling2d_1 (MaxPooling2D)  (None, 128, 128, 32  0
['activation_3[0][0]']
                                )

 conv2d_4 (Conv2D)              (None, 128, 128, 48  13872
['max_pooling2d_1[0][0]']
                                )

 batch_normalization_4 (BatchNo  (None, 128, 128, 48  192
['conv2d_4[0][0]']
 rmalization)                   )

 activation_4 (Activation)      (None, 128, 128, 48  0
['batch_normalization_4[0][0]']
                                )

 conv2d_5 (Conv2D)              (None, 128, 128, 48  20784
['activation_4[0][0]']
                                )

 batch_normalization_5 (BatchNo  (None, 128, 128, 48  192
['conv2d_5[0][0]']
 rmalization)                   )

 activation_5 (Activation)      (None, 128, 128, 48  0
['batch_normalization_5[0][0]']
```

```
                                )

 max_pooling2d_2 (MaxPooling2D)  (None, 64, 64, 48)  0
['activation_5[0][0]']

 conv2d_6 (Conv2D)              (None, 64, 64, 64)   27712
['max_pooling2d_2[0][0]']

 batch_normalization_6 (BatchNo  (None, 64, 64, 64)  256
['conv2d_6[0][0]']
 rmalization)

 activation_6 (Activation)      (None, 64, 64, 64)   0
['batch_normalization_6[0][0]']

 conv2d_7 (Conv2D)              (None, 64, 64, 64)   36928
['activation_6[0][0]']

 batch_normalization_7 (BatchNo  (None, 64, 64, 64)  256
['conv2d_7[0][0]']
 rmalization)

 activation_7 (Activation)      (None, 64, 64, 64)   0
['batch_normalization_7[0][0]']

 max_pooling2d_3 (MaxPooling2D)  (None, 32, 32, 64)  0
['activation_7[0][0]']

 conv2d_8 (Conv2D)              (None, 32, 32, 128)  73856
['max_pooling2d_3[0][0]']

 batch_normalization_8 (BatchNo  (None, 32, 32, 128)  512
['conv2d_8[0][0]']
 rmalization)

 activation_8 (Activation)      (None, 32, 32, 128)  0
['batch_normalization_8[0][0]']

 conv2d_9 (Conv2D)              (None, 32, 32, 128)  147584
['activation_8[0][0]']

 batch_normalization_9 (BatchNo  (None, 32, 32, 128)  512
['conv2d_9[0][0]']
 rmalization)

 activation_9 (Activation)      (None, 32, 32, 128)  0
['batch_normalization_9[0][0]']
```

```
 up_sampling2d (UpSampling2D)   (None, 64, 64, 128)  0
['activation_9[0][0]']

 concatenate (Concatenate)      (None, 64, 64, 192)  0
['up_sampling2d[0][0]',
'activation_7[0][0]']

 conv2d_10 (Conv2D)             (None, 64, 64, 64)   110656
['concatenate[0][0]']

 batch_normalization_10 (BatchN  (None, 64, 64, 64)  256
['conv2d_10[0][0]']
 ormalization)

 activation_10 (Activation)     (None, 64, 64, 64)   0
['batch_normalization_10[0][0]']

 conv2d_11 (Conv2D)             (None, 64, 64, 64)   36928
['activation_10[0][0]']

 batch_normalization_11 (BatchN  (None, 64, 64, 64)  256
['conv2d_11[0][0]']
 ormalization)

 activation_11 (Activation)     (None, 64, 64, 64)   0
['batch_normalization_11[0][0]']

 up_sampling2d_1 (UpSampling2D)  (None, 128, 128, 64  0
['activation_11[0][0]']
                                )

 concatenate_1 (Concatenate)    (None, 128, 128, 11  0
['up_sampling2d_1[0][0]',
                                2)
'activation_5[0][0]']

 conv2d_12 (Conv2D)             (None, 128, 128, 48  48432
['concatenate_1[0][0]']
                                )

 batch_normalization_12 (BatchN  (None, 128, 128, 48  192
['conv2d_12[0][0]']
 ormalization)                  )

 activation_12 (Activation)     (None, 128, 128, 48  0
['batch_normalization_12[0][0]']
                                )
```

```
 conv2d_13 (Conv2D)             (None, 128, 128, 48  20784
['activation_12[0][0]']
                                )

 batch_normalization_13 (BatchN  (None, 128, 128, 48  192
['conv2d_13[0][0]']
 ormalization)                  )

 activation_13 (Activation)     (None, 128, 128, 48  0
['batch_normalization_13[0][0]']
                                )

 up_sampling2d_2 (UpSampling2D)  (None, 256, 256, 48  0
['activation_13[0][0]']
                                )

 concatenate_2 (Concatenate)    (None, 256, 256, 80  0
['up_sampling2d_2[0][0]',
                                )
'activation_3[0][0]']

 conv2d_14 (Conv2D)             (None, 256, 256, 32  23072
['concatenate_2[0][0]']
                                )

 batch_normalization_14 (BatchN  (None, 256, 256, 32  128
['conv2d_14[0][0]']
 ormalization)                  )

 activation_14 (Activation)     (None, 256, 256, 32  0
['batch_normalization_14[0][0]']
                                )

 conv2d_15 (Conv2D)             (None, 256, 256, 32  9248
['activation_14[0][0]']
                                )

 batch_normalization_15 (BatchN  (None, 256, 256, 32  128
['conv2d_15[0][0]']
 ormalization)                  )

 activation_15 (Activation)     (None, 256, 256, 32  0
['batch_normalization_15[0][0]']
                                )

 up_sampling2d_3 (UpSampling2D)  (None, 512, 512, 32  0
['activation_15[0][0]']
                                )
```

```
 concatenate_3 (Concatenate)    (None, 512, 512, 48  0
['up_sampling2d_3[0][0]',
                                )
'activation_1[0][0]']

 conv2d_16 (Conv2D)             (None, 512, 512, 16  6928
['concatenate_3[0][0]']
                                )

 batch_normalization_16 (BatchN  (None, 512, 512, 16  64
['conv2d_16[0][0]']
 ormalization)                  )

 activation_16 (Activation)     (None, 512, 512, 16  0
['batch_normalization_16[0][0]']
                                )

 conv2d_17 (Conv2D)             (None, 512, 512, 16  2320
['activation_16[0][0]']
                                )

 batch_normalization_17 (BatchN  (None, 512, 512, 16  64
['conv2d_17[0][0]']
 ormalization)                  )

 activation_17 (Activation)     (None, 512, 512, 16  0
['batch_normalization_17[0][0]']
                                )

 conv2d_18 (Conv2D)             (None, 512, 512, 7)  119
['activation_17[0][0]']

==================================================================================
==================
Total params: 599,463
Trainable params: 597,671
Non-trainable params: 1,792
----------------------------------------------------------------------------------
------------------
```

[ ]:
```
## Shuffle DataFrame
## For function origin, check this stack overflow: https://stackoverflow.com/
 ↪questions/4601373/better-way-to-shuffle-two-numpy-arrays-in-unison
# def unison_shuffled(a, b):
#     assert len(a) == len(b)
#     p = np.random.permutation(len(a))
```

```
#     return a[p], b[p]

# X_shuf, y_shuf = unison_shuffled(X, y)
```

[ ]:
```
## Downsample so that we don't break our computers
X_down = X[0:500, :,:,:]
y_down = y[0:500, :,:,:]
print(X_down.shape)
print(y_down.shape)
```

```
(500, 512, 512, 3)
(500, 512, 512, 7)
```

[ ]:
```
X_train, X_test, y_train, y_test = train_test_split(X_down, y_down, test_size=0.
 ↪25, random_state=42)
print("X_train: ", X_train.shape)
print("y_train: ", y_train.shape)
print("X_test: ", X_test.shape)
print("y_test: ", y_test.shape)
```

```
X_train:  (375, 512, 512, 3)
y_train:  (375, 512, 512, 7)
X_test:  (125, 512, 512, 3)
y_test:  (125, 512, 512, 7)
```

[ ]:
```
## Here is a helpful guide on using Keras ImageDataGenerator to scale images
## https://machinelearningmastery.com/
 ↪how-to-normalize-center-and-standardize-images-with-the-imagedatagenerator-in-keras/
 ↪
# datagen = ImageDataGenerator(rescale=1.0/255.0)

# Instead of dynamically scaling with the ImageDataGenerator, scale manually:
# X_train = X_train / 255
# X_test = X_test / 255

# Initialize ImageDataGenerator
datagen = ImageDataGenerator()

# prepare an iterators to scale images
train_iterator = datagen.flow(X_train, y_train, batch_size=5)
test_iterator = datagen.flow(X_test, y_test, batch_size=5)
print('Batches train=%d, test=%d' % (len(train_iterator), len(test_iterator)))
```

```
Batches train=75, test=25
```

[ ]:
```
#hyperparamters
shape = (512, 512, 3)
```

```
num_classes = 7
lr = 1e-7
batch_size = 5 # johnathan testing more than 1 image in a batch
epochs = 50 # expanding epochs to see if accuracy improves over time

train_steps = len(X_train)//batch_size
test_steps = len(X_test)//batch_size
print(train_steps)
print(test_steps)
```

```
75
25
```

[ ]:
```
# save model here
path = "/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/
 ↪650_RemoteSensing/LULC_FinalProject"
model_path = '{}/models'.format(path)
model_file = '{}/custom_unet_v2.h5'.format(model_path)

checkpoint = ModelCheckpoint(filepath = model_file,
                             monitor = 'accuracy',
                             save_best_only = True,
                             verbose=1
                             )

early_stop = EarlyStopping(monitor = 'accuracy',
                           patience = 15,
                           restore_best_weights = True)

reduce_LR = ReduceLROnPlateau(monitor = 'accuracy',
                              factor = 0.1,
                              patience = 5)

callback_list = [reduce_LR, early_stop, checkpoint]
```

[ ]:
```
model = build_unet(shape, num_classes)
model.compile(loss="categorical_crossentropy",
              optimizer=Adam(),  # Johnathan removed lr from Adam parameter␣
 ↪because it was keeping lr stagnant, whereas Adam can adjust lr dynamically
              # See: https://www.youtube.com/watch?v=mdKjMPmcWjY
              metrics="accuracy") #Johnathan added accuracy metric morning of 5/
 ↪2
```

[ ]:
```
# Run the model
history = model.fit(train_iterator,
        steps_per_epoch=train_steps,
        validation_data=(X_test, y_test),
```

```
        validation_steps=test_steps,
        epochs=epochs,
        callbacks=callback_list,
        # verbose=1
    )
```

```
Epoch 1/50
75/75 [==============================] - ETA: 0s - loss: 1.4053 - accuracy:
0.5179
Epoch 1: accuracy improved from -inf to 0.51785, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_v2.h5
75/75 [==============================] - 14s 157ms/step - loss: 1.4053 -
accuracy: 0.5179 - val_loss: 6.8787 - val_accuracy: 0.1233 - lr: 0.0010
Epoch 2/50
75/75 [==============================] - ETA: 0s - loss: 1.2243 - accuracy:
0.5629
Epoch 2: accuracy improved from 0.51785 to 0.56287, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_v2.h5
75/75 [==============================] - 11s 143ms/step - loss: 1.2243 -
accuracy: 0.5629 - val_loss: 1.3544 - val_accuracy: 0.4843 - lr: 0.0010
Epoch 3/50
75/75 [==============================] - ETA: 0s - loss: 1.1939 - accuracy:
0.5665
Epoch 3: accuracy improved from 0.56287 to 0.56648, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_v2.h5
75/75 [==============================] - 11s 144ms/step - loss: 1.1939 -
accuracy: 0.5665 - val_loss: 1.4056 - val_accuracy: 0.5224 - lr: 0.0010
Epoch 4/50
75/75 [==============================] - ETA: 0s - loss: 1.1453 - accuracy:
0.5776
Epoch 4: accuracy improved from 0.56648 to 0.57760, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_v2.h5
75/75 [==============================] - 11s 144ms/step - loss: 1.1453 -
accuracy: 0.5776 - val_loss: 1.6836 - val_accuracy: 0.5097 - lr: 0.0010
Epoch 5/50
75/75 [==============================] - ETA: 0s - loss: 1.1243 - accuracy:
0.5857
Epoch 5: accuracy improved from 0.57760 to 0.58566, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_v2.h5
75/75 [==============================] - 11s 143ms/step - loss: 1.1243 -
accuracy: 0.5857 - val_loss: 1.1828 - val_accuracy: 0.5381 - lr: 0.0010
Epoch 6/50
```

```
75/75 [==============================] - ETA: 0s - loss: 1.1053 - accuracy:
0.5895
Epoch 6: accuracy improved from 0.58566 to 0.58947, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_v2.h5
75/75 [==============================] - 11s 144ms/step - loss: 1.1053 -
accuracy: 0.5895 - val_loss: 1.2576 - val_accuracy: 0.5315 - lr: 0.0010
Epoch 7/50
75/75 [==============================] - ETA: 0s - loss: 1.0825 - accuracy:
0.6042
Epoch 7: accuracy improved from 0.58947 to 0.60420, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_v2.h5
75/75 [==============================] - 11s 144ms/step - loss: 1.0825 -
accuracy: 0.6042 - val_loss: 1.0807 - val_accuracy: 0.5893 - lr: 0.0010
Epoch 8/50
75/75 [==============================] - ETA: 0s - loss: 1.0792 - accuracy:
0.6024
Epoch 8: accuracy did not improve from 0.60420
75/75 [==============================] - 10s 138ms/step - loss: 1.0792 -
accuracy: 0.6024 - val_loss: 0.9668 - val_accuracy: 0.6448 - lr: 0.0010
Epoch 9/50
75/75 [==============================] - ETA: 0s - loss: 1.0940 - accuracy:
0.5934
Epoch 9: accuracy did not improve from 0.60420
75/75 [==============================] - 10s 138ms/step - loss: 1.0940 -
accuracy: 0.5934 - val_loss: 1.1998 - val_accuracy: 0.5661 - lr: 0.0010
Epoch 10/50
75/75 [==============================] - ETA: 0s - loss: 1.0993 - accuracy:
0.5999
Epoch 10: accuracy did not improve from 0.60420
75/75 [==============================] - 10s 138ms/step - loss: 1.0993 -
accuracy: 0.5999 - val_loss: 1.0009 - val_accuracy: 0.6288 - lr: 0.0010
Epoch 11/50
75/75 [==============================] - ETA: 0s - loss: 1.0395 - accuracy:
0.6151
Epoch 11: accuracy improved from 0.60420 to 0.61506, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_v2.h5
75/75 [==============================] - 11s 150ms/step - loss: 1.0395 -
accuracy: 0.6151 - val_loss: 1.2453 - val_accuracy: 0.5788 - lr: 0.0010
Epoch 12/50
75/75 [==============================] - ETA: 0s - loss: 1.0411 - accuracy:
0.6216
Epoch 12: accuracy improved from 0.61506 to 0.62163, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_v2.h5
75/75 [==============================] - 11s 143ms/step - loss: 1.0411 -
```

```
accuracy: 0.6216 - val_loss: 1.2313 - val_accuracy: 0.6063 - lr: 0.0010
Epoch 13/50
75/75 [==============================] - ETA: 0s - loss: 1.0322 - accuracy:
0.6198
Epoch 13: accuracy did not improve from 0.62163
75/75 [==============================] - 10s 138ms/step - loss: 1.0322 -
accuracy: 0.6198 - val_loss: 0.9759 - val_accuracy: 0.6565 - lr: 0.0010
Epoch 14/50
75/75 [==============================] - ETA: 0s - loss: 1.0161 - accuracy:
0.6225
Epoch 14: accuracy improved from 0.62163 to 0.62252, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_v2.h5
75/75 [==============================] - 11s 150ms/step - loss: 1.0161 -
accuracy: 0.6225 - val_loss: 0.8878 - val_accuracy: 0.6857 - lr: 0.0010
Epoch 15/50
75/75 [==============================] - ETA: 0s - loss: 1.0262 - accuracy:
0.6229
Epoch 15: accuracy improved from 0.62252 to 0.62292, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_v2.h5
75/75 [==============================] - 11s 143ms/step - loss: 1.0262 -
accuracy: 0.6229 - val_loss: 1.1501 - val_accuracy: 0.5714 - lr: 0.0010
Epoch 16/50
75/75 [==============================] - ETA: 0s - loss: 1.0470 - accuracy:
0.6155
Epoch 16: accuracy did not improve from 0.62292
75/75 [==============================] - 10s 138ms/step - loss: 1.0470 -
accuracy: 0.6155 - val_loss: 1.0670 - val_accuracy: 0.6321 - lr: 0.0010
Epoch 17/50
75/75 [==============================] - ETA: 0s - loss: 0.9934 - accuracy:
0.6431
Epoch 17: accuracy improved from 0.62292 to 0.64314, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_v2.h5
75/75 [==============================] - 11s 151ms/step - loss: 0.9934 -
accuracy: 0.6431 - val_loss: 1.3231 - val_accuracy: 0.4959 - lr: 0.0010
Epoch 18/50
75/75 [==============================] - ETA: 0s - loss: 1.0002 - accuracy:
0.6330
Epoch 18: accuracy did not improve from 0.64314
75/75 [==============================] - 10s 139ms/step - loss: 1.0002 -
accuracy: 0.6330 - val_loss: 1.4236 - val_accuracy: 0.3780 - lr: 0.0010
Epoch 19/50
75/75 [==============================] - ETA: 0s - loss: 0.9818 - accuracy:
0.6415
Epoch 19: accuracy did not improve from 0.64314
75/75 [==============================] - 10s 139ms/step - loss: 0.9818 -
```

```
accuracy: 0.6415 - val_loss: 1.1419 - val_accuracy: 0.5972 - lr: 0.0010
Epoch 20/50
75/75 [==============================] - ETA: 0s - loss: 0.9832 - accuracy:
0.6389
Epoch 20: accuracy did not improve from 0.64314
75/75 [==============================] - 10s 138ms/step - loss: 0.9832 -
accuracy: 0.6389 - val_loss: 0.9628 - val_accuracy: 0.6495 - lr: 0.0010
Epoch 21/50
75/75 [==============================] - ETA: 0s - loss: 0.9713 - accuracy:
0.6416
Epoch 21: accuracy did not improve from 0.64314
75/75 [==============================] - 10s 138ms/step - loss: 0.9713 -
accuracy: 0.6416 - val_loss: 0.9922 - val_accuracy: 0.6523 - lr: 0.0010
Epoch 22/50
75/75 [==============================] - ETA: 0s - loss: 0.9542 - accuracy:
0.6487
Epoch 22: accuracy improved from 0.64314 to 0.64866, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_v2.h5
75/75 [==============================] - 11s 151ms/step - loss: 0.9542 -
accuracy: 0.6487 - val_loss: 1.1020 - val_accuracy: 0.5573 - lr: 0.0010
Epoch 23/50
75/75 [==============================] - ETA: 0s - loss: 0.9622 - accuracy:
0.6408
Epoch 23: accuracy did not improve from 0.64866
75/75 [==============================] - 10s 139ms/step - loss: 0.9622 -
accuracy: 0.6408 - val_loss: 0.9000 - val_accuracy: 0.6912 - lr: 0.0010
Epoch 24/50
75/75 [==============================] - ETA: 0s - loss: 0.9532 - accuracy:
0.6561
Epoch 24: accuracy improved from 0.64866 to 0.65606, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_v2.h5
75/75 [==============================] - 11s 149ms/step - loss: 0.9532 -
accuracy: 0.6561 - val_loss: 1.3536 - val_accuracy: 0.4540 - lr: 0.0010
Epoch 25/50
75/75 [==============================] - ETA: 0s - loss: 0.9595 - accuracy:
0.6505
Epoch 25: accuracy did not improve from 0.65606
75/75 [==============================] - 10s 139ms/step - loss: 0.9595 -
accuracy: 0.6505 - val_loss: 2.9048 - val_accuracy: 0.2801 - lr: 0.0010
Epoch 26/50
75/75 [==============================] - ETA: 0s - loss: 0.9547 - accuracy:
0.6499
Epoch 26: accuracy did not improve from 0.65606
75/75 [==============================] - 10s 138ms/step - loss: 0.9547 -
accuracy: 0.6499 - val_loss: 1.0165 - val_accuracy: 0.6505 - lr: 0.0010
Epoch 27/50
```

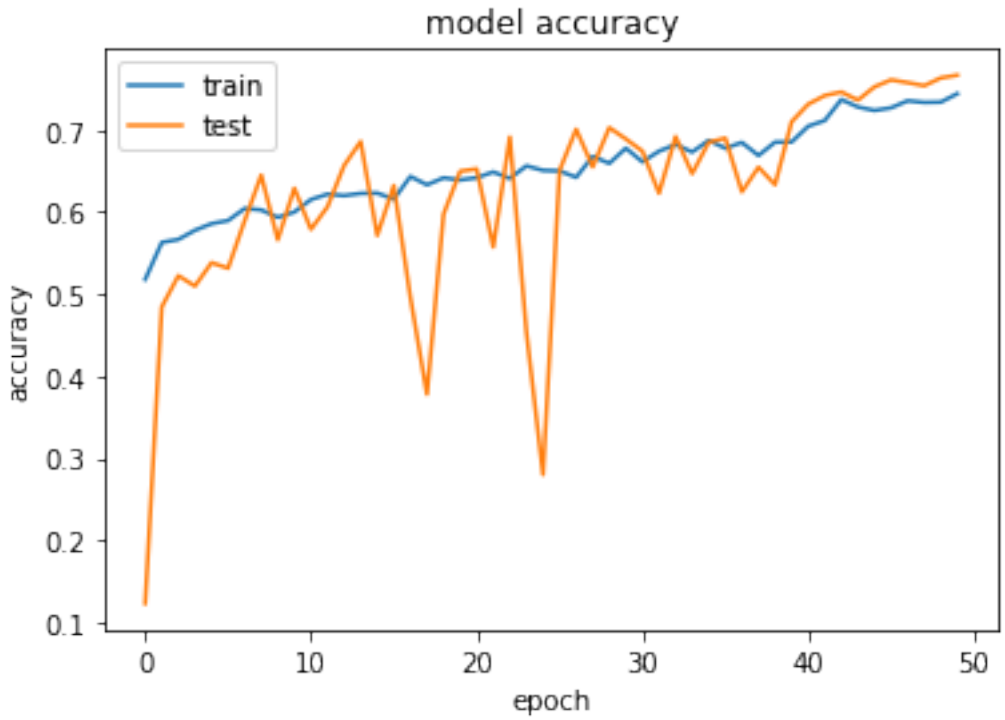
```
75/75 [==============================] - ETA: 0s - loss: 0.9830 - accuracy:
0.6421
Epoch 27: accuracy did not improve from 0.65606
75/75 [==============================] - 10s 138ms/step - loss: 0.9830 -
accuracy: 0.6421 - val_loss: 0.8156 - val_accuracy: 0.7006 - lr: 0.0010
Epoch 28/50
75/75 [==============================] - ETA: 0s - loss: 0.9141 - accuracy:
0.6676
Epoch 28: accuracy improved from 0.65606 to 0.66759, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_v2.h5
75/75 [==============================] - 11s 150ms/step - loss: 0.9141 -
accuracy: 0.6676 - val_loss: 0.9292 - val_accuracy: 0.6548 - lr: 0.0010
Epoch 29/50
75/75 [==============================] - ETA: 0s - loss: 0.9294 - accuracy:
0.6591
Epoch 29: accuracy did not improve from 0.66759
75/75 [==============================] - 10s 138ms/step - loss: 0.9294 -
accuracy: 0.6591 - val_loss: 0.8636 - val_accuracy: 0.7026 - lr: 0.0010
Epoch 30/50
75/75 [==============================] - ETA: 0s - loss: 0.8959 - accuracy:
0.6775
Epoch 30: accuracy improved from 0.66759 to 0.67753, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_v2.h5
75/75 [==============================] - 11s 149ms/step - loss: 0.8959 -
accuracy: 0.6775 - val_loss: 0.9344 - val_accuracy: 0.6888 - lr: 0.0010
Epoch 31/50
75/75 [==============================] - ETA: 0s - loss: 0.9172 - accuracy:
0.6610
Epoch 31: accuracy did not improve from 0.67753
75/75 [==============================] - 10s 139ms/step - loss: 0.9172 -
accuracy: 0.6610 - val_loss: 0.9353 - val_accuracy: 0.6740 - lr: 0.0010
Epoch 32/50
75/75 [==============================] - ETA: 0s - loss: 0.8975 - accuracy:
0.6739
Epoch 32: accuracy did not improve from 0.67753
75/75 [==============================] - 10s 139ms/step - loss: 0.8975 -
accuracy: 0.6739 - val_loss: 1.0878 - val_accuracy: 0.6227 - lr: 0.0010
Epoch 33/50
75/75 [==============================] - ETA: 0s - loss: 0.8774 - accuracy:
0.6821
Epoch 33: accuracy improved from 0.67753 to 0.68206, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_v2.h5
75/75 [==============================] - 11s 150ms/step - loss: 0.8774 -
accuracy: 0.6821 - val_loss: 0.8612 - val_accuracy: 0.6917 - lr: 0.0010
Epoch 34/50
```

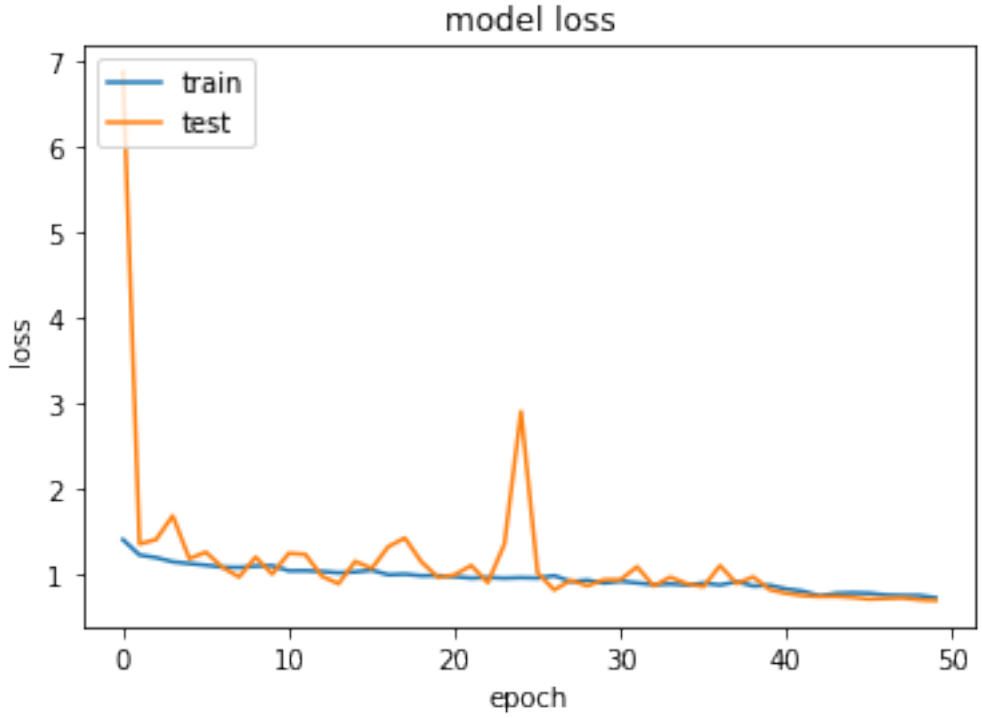
```
75/75 [==============================] - ETA: 0s - loss: 0.8847 - accuracy:
0.6729
Epoch 34: accuracy did not improve from 0.68206
75/75 [==============================] - 10s 139ms/step - loss: 0.8847 -
accuracy: 0.6729 - val_loss: 0.9647 - val_accuracy: 0.6466 - lr: 0.0010
Epoch 35/50
75/75 [==============================] - ETA: 0s - loss: 0.8736 - accuracy:
0.6871
Epoch 35: accuracy improved from 0.68206 to 0.68714, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_v2.h5
75/75 [==============================] - 11s 149ms/step - loss: 0.8736 -
accuracy: 0.6871 - val_loss: 0.8883 - val_accuracy: 0.6848 - lr: 0.0010
Epoch 36/50
75/75 [==============================] - ETA: 0s - loss: 0.8960 - accuracy:
0.6775
Epoch 36: accuracy did not improve from 0.68714
75/75 [==============================] - 10s 138ms/step - loss: 0.8960 -
accuracy: 0.6775 - val_loss: 0.8546 - val_accuracy: 0.6898 - lr: 0.0010
Epoch 37/50
75/75 [==============================] - ETA: 0s - loss: 0.8729 - accuracy:
0.6846
Epoch 37: accuracy did not improve from 0.68714
75/75 [==============================] - 10s 139ms/step - loss: 0.8729 -
accuracy: 0.6846 - val_loss: 1.1000 - val_accuracy: 0.6245 - lr: 0.0010
Epoch 38/50
75/75 [==============================] - ETA: 0s - loss: 0.9126 - accuracy:
0.6687
Epoch 38: accuracy did not improve from 0.68714
75/75 [==============================] - 10s 139ms/step - loss: 0.9126 -
accuracy: 0.6687 - val_loss: 0.8906 - val_accuracy: 0.6544 - lr: 0.0010
Epoch 39/50
75/75 [==============================] - ETA: 0s - loss: 0.8615 - accuracy:
0.6849
Epoch 39: accuracy did not improve from 0.68714
75/75 [==============================] - 10s 138ms/step - loss: 0.8615 -
accuracy: 0.6849 - val_loss: 0.9682 - val_accuracy: 0.6331 - lr: 0.0010
Epoch 40/50
75/75 [==============================] - ETA: 0s - loss: 0.8702 - accuracy:
0.6851
Epoch 40: accuracy did not improve from 0.68714
75/75 [==============================] - 10s 138ms/step - loss: 0.8702 -
accuracy: 0.6851 - val_loss: 0.8182 - val_accuracy: 0.7098 - lr: 0.0010
Epoch 41/50
75/75 [==============================] - ETA: 0s - loss: 0.8262 - accuracy:
0.7044
Epoch 41: accuracy improved from 0.68714 to 0.70436, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
```

```
inalProject/models/custom_unet_v2.h5
75/75 [==============================] - 11s 150ms/step - loss: 0.8262 -
accuracy: 0.7044 - val_loss: 0.7750 - val_accuracy: 0.7311 - lr: 1.0000e-04
Epoch 42/50
75/75 [==============================] - ETA: 0s - loss: 0.8008 - accuracy:
0.7113
Epoch 42: accuracy improved from 0.70436 to 0.71129, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_v2.h5
75/75 [==============================] - 11s 144ms/step - loss: 0.8008 -
accuracy: 0.7113 - val_loss: 0.7494 - val_accuracy: 0.7417 - lr: 1.0000e-04
Epoch 43/50
75/75 [==============================] - ETA: 0s - loss: 0.7461 - accuracy:
0.7368
Epoch 43: accuracy improved from 0.71129 to 0.73677, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_v2.h5
75/75 [==============================] - 11s 144ms/step - loss: 0.7461 -
accuracy: 0.7368 - val_loss: 0.7381 - val_accuracy: 0.7460 - lr: 1.0000e-04
Epoch 44/50
75/75 [==============================] - ETA: 0s - loss: 0.7750 - accuracy:
0.7277
Epoch 44: accuracy did not improve from 0.73677
75/75 [==============================] - 10s 138ms/step - loss: 0.7750 -
accuracy: 0.7277 - val_loss: 0.7402 - val_accuracy: 0.7361 - lr: 1.0000e-04
Epoch 45/50
75/75 [==============================] - ETA: 0s - loss: 0.7814 - accuracy:
0.7235
Epoch 45: accuracy did not improve from 0.73677
75/75 [==============================] - 10s 139ms/step - loss: 0.7814 -
accuracy: 0.7235 - val_loss: 0.7290 - val_accuracy: 0.7523 - lr: 1.0000e-04
Epoch 46/50
75/75 [==============================] - ETA: 0s - loss: 0.7765 - accuracy:
0.7267
Epoch 46: accuracy did not improve from 0.73677
75/75 [==============================] - 10s 138ms/step - loss: 0.7765 -
accuracy: 0.7267 - val_loss: 0.7081 - val_accuracy: 0.7608 - lr: 1.0000e-04
Epoch 47/50
75/75 [==============================] - ETA: 0s - loss: 0.7559 - accuracy:
0.7353
Epoch 47: accuracy did not improve from 0.73677
75/75 [==============================] - 10s 138ms/step - loss: 0.7559 -
accuracy: 0.7353 - val_loss: 0.7153 - val_accuracy: 0.7574 - lr: 1.0000e-04
Epoch 48/50
75/75 [==============================] - ETA: 0s - loss: 0.7506 - accuracy:
0.7333
Epoch 48: accuracy did not improve from 0.73677
75/75 [==============================] - 10s 139ms/step - loss: 0.7506 -
```

```
accuracy: 0.7333 - val_loss: 0.7214 - val_accuracy: 0.7536 - lr: 1.0000e-04
Epoch 49/50
75/75 [==============================] - ETA: 0s - loss: 0.7510 - accuracy:
0.7337
Epoch 49: accuracy did not improve from 0.73677
75/75 [==============================] - 10s 138ms/step - loss: 0.7510 -
accuracy: 0.7337 - val_loss: 0.7004 - val_accuracy: 0.7631 - lr: 1.0000e-05
Epoch 50/50
75/75 [==============================] - ETA: 0s - loss: 0.7236 - accuracy:
0.7440
Epoch 50: accuracy improved from 0.73677 to 0.74395, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_v2.h5
75/75 [==============================] - 11s 150ms/step - loss: 0.7236 -
accuracy: 0.7440 - val_loss: 0.6921 - val_accuracy: 0.7665 - lr: 1.0000e-05
```

[ ]:
```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

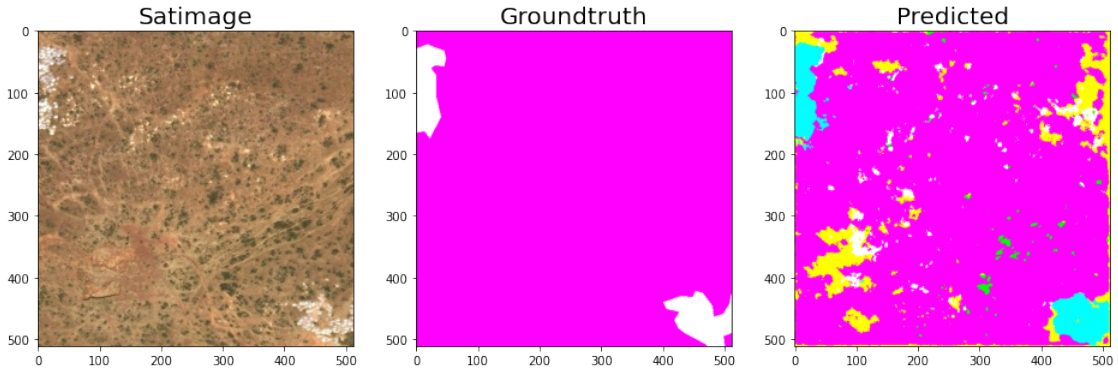

model accuracy
train
test
accuracy
epoch
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
10
20
30
40
50

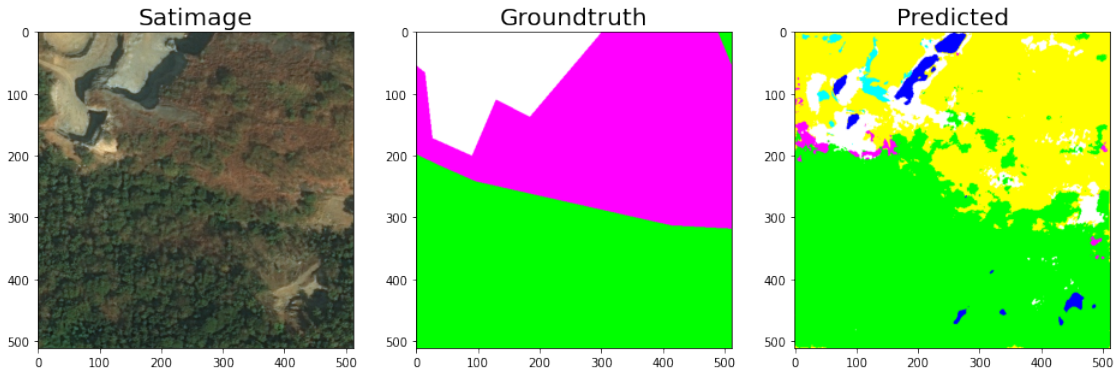

model loss
train
test
loss
epoch
1
2
3
4
5
6
7
0
10
20
30
40
50

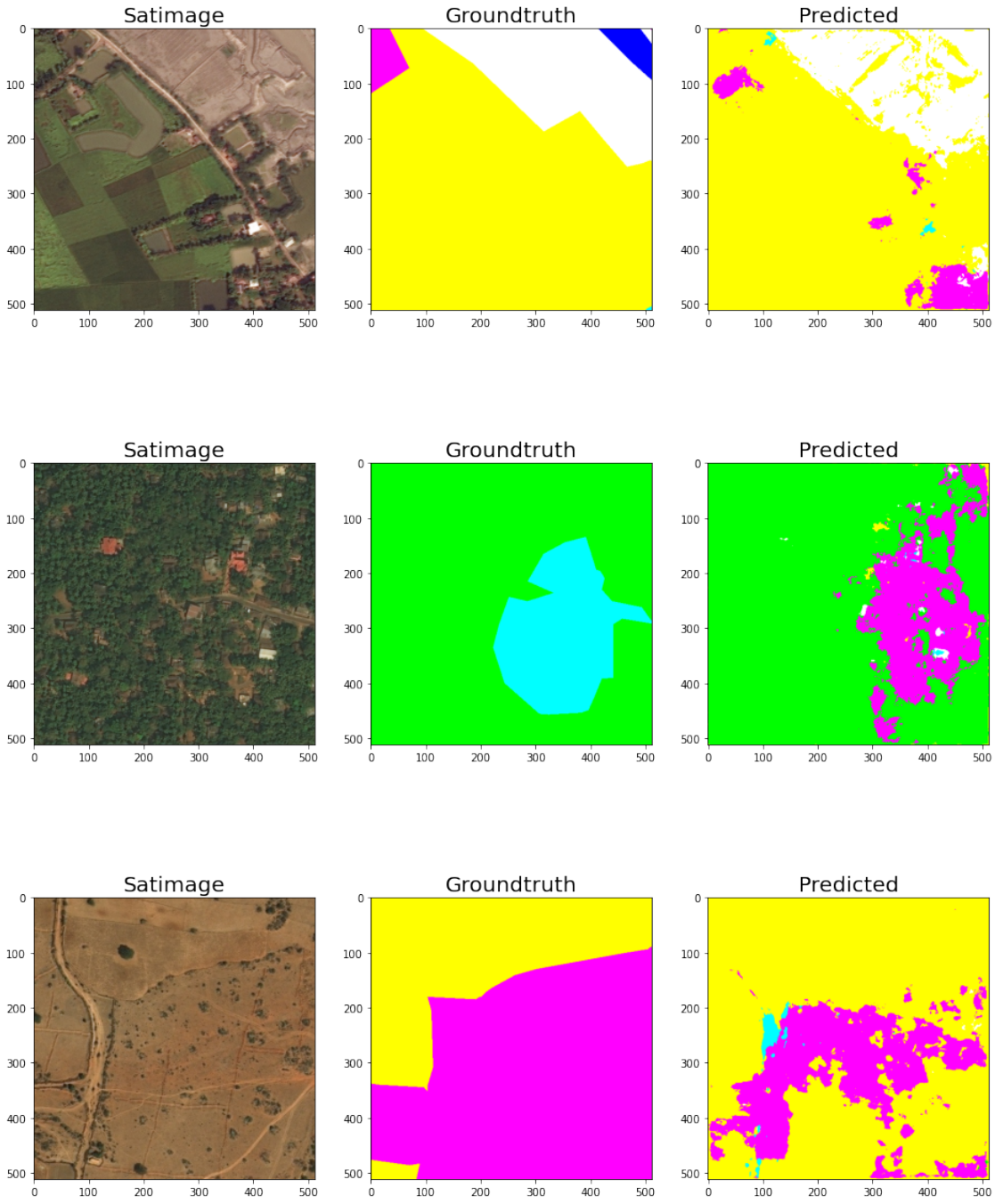
[ ]:
```
score = model.evaluate(test_iterator, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
Test loss: 0.6920950412750244
Test accuracy: 0.7664816379547119
```

[ ]:
```
y_pred = model.predict(X_test)

for i in range(5):
  visualize(satImage = X_test[i],
            groundTruth=(onehot_to_rgb(y_test[i], class_dict)),
            predicted=(onehot_to_rgb(y_pred[i], class_dict)))
```







Testing a higher batch size

[ ]:
```
# Initialize ImageDataGenerator
datagen = ImageDataGenerator()

# prepare an iterators to scale images
train_iterator = datagen.flow(X_train, y_train, batch_size=25)
```

```
test_iterator = datagen.flow(X_test, y_test, batch_size=25)
print('Batches train=%d, test=%d' % (len(train_iterator), len(test_iterator)))
```

```
Batches train=15, test=5
```

[ ]:
```
#hyperparamters
shape = (512, 512, 3)
num_classes = 7
lr = 1e-7
batch_size = 25 # johnathan testing more than 1 image in a batch
epochs = 75 # expanding epochs to see if accuracy improves over time

train_steps = len(X_train)//batch_size
test_steps = len(X_test)//batch_size
print(train_steps)
print(test_steps)
```

```
15
5
```

[ ]:
```
# save model here
path = "/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/
 ↪650_RemoteSensing/LULC_FinalProject"
model_path = '{}/models'.format(path)
model_file = '{}/custom_unet_highbatchsize.h5'.format(model_path)

checkpoint = ModelCheckpoint(filepath = model_file,
                             monitor = 'accuracy',
                             save_best_only = True,
                             verbose=1
                             )

early_stop = EarlyStopping(monitor = 'accuracy',
                           patience = 15,
                           restore_best_weights = True)

reduce_LR = ReduceLROnPlateau(monitor = 'accuracy',
                              factor = 0.1,
                              patience = 5)

callback_list = [reduce_LR, early_stop, checkpoint]
```

[ ]:
```
model = build_unet(shape, num_classes)
model.compile(loss="categorical_crossentropy",
              optimizer=Adam(),
              metrics="accuracy")
```

[ ]:
```
# Run the model
history = model.fit(train_iterator,
        steps_per_epoch=train_steps,
        validation_data=(X_test, y_test),
        validation_steps=test_steps,
        epochs=epochs,
        callbacks=callback_list,
        # verbose=1
    )
```

```
Epoch 1/75
15/15 [==============================] - ETA: 0s - loss: 1.6280 - accuracy:
0.4980
Epoch 1: accuracy improved from -inf to 0.49798, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 17s 734ms/step - loss: 1.6280 -
accuracy: 0.4980 - val_loss: 50.2014 - val_accuracy: 0.0969 - lr: 0.0010
Epoch 2/75
15/15 [==============================] - ETA: 0s - loss: 1.3943 - accuracy:
0.5951
Epoch 2: accuracy improved from 0.49798 to 0.59509, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 689ms/step - loss: 1.3943 -
accuracy: 0.5951 - val_loss: 54.5566 - val_accuracy: 0.0970 - lr: 0.0010
Epoch 3/75
15/15 [==============================] - ETA: 0s - loss: 1.2752 - accuracy:
0.6274
Epoch 3: accuracy improved from 0.59509 to 0.62738, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 655ms/step - loss: 1.2752 -
accuracy: 0.6274 - val_loss: 3.4702 - val_accuracy: 0.2164 - lr: 0.0010
Epoch 4/75
15/15 [==============================] - ETA: 0s - loss: 1.1963 - accuracy:
0.6477
Epoch 4: accuracy improved from 0.62738 to 0.64774, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 659ms/step - loss: 1.1963 -
accuracy: 0.6477 - val_loss: 2.1919 - val_accuracy: 0.2051 - lr: 0.0010
Epoch 5/75
15/15 [==============================] - ETA: 0s - loss: 1.1442 - accuracy:
0.6562
Epoch 5: accuracy improved from 0.64774 to 0.65617, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
```

```
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 690ms/step - loss: 1.1442 -
accuracy: 0.6562 - val_loss: 2.0737 - val_accuracy: 0.1812 - lr: 0.0010
Epoch 6/75
15/15 [==============================] - ETA: 0s - loss: 1.0593 - accuracy:
0.6831
Epoch 6: accuracy improved from 0.65617 to 0.68314, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 654ms/step - loss: 1.0593 -
accuracy: 0.6831 - val_loss: 2.0044 - val_accuracy: 0.2465 - lr: 0.0010
Epoch 7/75
15/15 [==============================] - ETA: 0s - loss: 1.0441 - accuracy:
0.6799
Epoch 7: accuracy did not improve from 0.68314
15/15 [==============================] - 9s 625ms/step - loss: 1.0441 -
accuracy: 0.6799 - val_loss: 1.4962 - val_accuracy: 0.4870 - lr: 0.0010
Epoch 8/75
15/15 [==============================] - ETA: 0s - loss: 0.9981 - accuracy:
0.6859
Epoch 8: accuracy improved from 0.68314 to 0.68595, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 679ms/step - loss: 0.9981 -
accuracy: 0.6859 - val_loss: 1.7285 - val_accuracy: 0.4971 - lr: 0.0010
Epoch 9/75
15/15 [==============================] - ETA: 0s - loss: 0.9763 - accuracy:
0.6925
Epoch 9: accuracy improved from 0.68595 to 0.69250, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 654ms/step - loss: 0.9763 -
accuracy: 0.6925 - val_loss: 1.5832 - val_accuracy: 0.4369 - lr: 0.0010
Epoch 10/75
15/15 [==============================] - ETA: 0s - loss: 0.9819 - accuracy:
0.6867
Epoch 10: accuracy did not improve from 0.69250
15/15 [==============================] - 9s 625ms/step - loss: 0.9819 -
accuracy: 0.6867 - val_loss: 1.4995 - val_accuracy: 0.4558 - lr: 0.0010
Epoch 11/75
15/15 [==============================] - ETA: 0s - loss: 0.9399 - accuracy:
0.6899
Epoch 11: accuracy did not improve from 0.69250
15/15 [==============================] - 9s 627ms/step - loss: 0.9399 -
accuracy: 0.6899 - val_loss: 1.1364 - val_accuracy: 0.6530 - lr: 0.0010
Epoch 12/75
15/15 [==============================] - ETA: 0s - loss: 0.9378 - accuracy:
0.6896
```

```
Epoch 12: accuracy did not improve from 0.69250
15/15 [==============================] - 9s 625ms/step - loss: 0.9378 -
accuracy: 0.6896 - val_loss: 1.2405 - val_accuracy: 0.5737 - lr: 0.0010
Epoch 13/75
15/15 [==============================] - ETA: 0s - loss: 0.9240 - accuracy:
0.6998
Epoch 13: accuracy improved from 0.69250 to 0.69981, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 683ms/step - loss: 0.9240 -
accuracy: 0.6998 - val_loss: 1.5121 - val_accuracy: 0.6063 - lr: 0.0010
Epoch 14/75
15/15 [==============================] - ETA: 0s - loss: 0.8889 - accuracy:
0.7058
Epoch 14: accuracy improved from 0.69981 to 0.70579, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 652ms/step - loss: 0.8889 -
accuracy: 0.7058 - val_loss: 1.1434 - val_accuracy: 0.5979 - lr: 0.0010
Epoch 15/75
15/15 [==============================] - ETA: 0s - loss: 0.8891 - accuracy:
0.7078
Epoch 15: accuracy improved from 0.70579 to 0.70785, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 655ms/step - loss: 0.8891 -
accuracy: 0.7078 - val_loss: 0.9796 - val_accuracy: 0.6816 - lr: 0.0010
Epoch 16/75
15/15 [==============================] - ETA: 0s - loss: 0.8749 - accuracy:
0.7032
Epoch 16: accuracy did not improve from 0.70785
15/15 [==============================] - 9s 626ms/step - loss: 0.8749 -
accuracy: 0.7032 - val_loss: 1.3106 - val_accuracy: 0.6094 - lr: 0.0010
Epoch 17/75
15/15 [==============================] - ETA: 0s - loss: 0.8096 - accuracy:
0.7371
Epoch 17: accuracy improved from 0.70785 to 0.73713, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 688ms/step - loss: 0.8096 -
accuracy: 0.7371 - val_loss: 0.8236 - val_accuracy: 0.7354 - lr: 0.0010
Epoch 18/75
15/15 [==============================] - ETA: 0s - loss: 0.8729 - accuracy:
0.7043
Epoch 18: accuracy did not improve from 0.73713
15/15 [==============================] - 9s 626ms/step - loss: 0.8729 -
accuracy: 0.7043 - val_loss: 1.5677 - val_accuracy: 0.5757 - lr: 0.0010
Epoch 19/75
```

```
15/15 [==============================] - ETA: 0s - loss: 0.8780 - accuracy:
0.7016
Epoch 19: accuracy did not improve from 0.73713
15/15 [==============================] - 9s 623ms/step - loss: 0.8780 -
accuracy: 0.7016 - val_loss: 1.4229 - val_accuracy: 0.5868 - lr: 0.0010
Epoch 20/75
15/15 [==============================] - ETA: 0s - loss: 0.7918 - accuracy:
0.7320
Epoch 20: accuracy did not improve from 0.73713
15/15 [==============================] - 9s 625ms/step - loss: 0.7918 -
accuracy: 0.7320 - val_loss: 0.8795 - val_accuracy: 0.6921 - lr: 0.0010
Epoch 21/75
15/15 [==============================] - ETA: 0s - loss: 0.7906 - accuracy:
0.7303
Epoch 21: accuracy did not improve from 0.73713
15/15 [==============================] - 9s 625ms/step - loss: 0.7906 -
accuracy: 0.7303 - val_loss: 0.9689 - val_accuracy: 0.6650 - lr: 0.0010
Epoch 22/75
15/15 [==============================] - ETA: 0s - loss: 0.8204 - accuracy:
0.7189
Epoch 22: accuracy did not improve from 0.73713
15/15 [==============================] - 9s 627ms/step - loss: 0.8204 -
accuracy: 0.7189 - val_loss: 0.7912 - val_accuracy: 0.7376 - lr: 0.0010
Epoch 23/75
15/15 [==============================] - ETA: 0s - loss: 0.7518 - accuracy:
0.7395
Epoch 23: accuracy improved from 0.73713 to 0.73952, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 690ms/step - loss: 0.7518 -
accuracy: 0.7395 - val_loss: 0.7588 - val_accuracy: 0.7362 - lr: 1.0000e-04
Epoch 24/75
15/15 [==============================] - ETA: 0s - loss: 0.7207 - accuracy:
0.7508
Epoch 24: accuracy improved from 0.73952 to 0.75081, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 652ms/step - loss: 0.7207 -
accuracy: 0.7508 - val_loss: 0.7592 - val_accuracy: 0.7373 - lr: 1.0000e-04
Epoch 25/75
15/15 [==============================] - ETA: 0s - loss: 0.7143 - accuracy:
0.7550
Epoch 25: accuracy improved from 0.75081 to 0.75503, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 652ms/step - loss: 0.7143 -
accuracy: 0.7550 - val_loss: 0.7768 - val_accuracy: 0.7342 - lr: 1.0000e-04
Epoch 26/75
```

```
15/15 [==============================] - ETA: 0s - loss: 0.7354 - accuracy:
0.7488
Epoch 26: accuracy did not improve from 0.75503
15/15 [==============================] - 9s 624ms/step - loss: 0.7354 -
accuracy: 0.7488 - val_loss: 0.7791 - val_accuracy: 0.7334 - lr: 1.0000e-04
Epoch 27/75
15/15 [==============================] - ETA: 0s - loss: 0.6976 - accuracy:
0.7596
Epoch 27: accuracy improved from 0.75503 to 0.75962, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 691ms/step - loss: 0.6976 -
accuracy: 0.7596 - val_loss: 0.7809 - val_accuracy: 0.7349 - lr: 1.0000e-04
Epoch 28/75
15/15 [==============================] - ETA: 0s - loss: 0.6986 - accuracy:
0.7632
Epoch 28: accuracy improved from 0.75962 to 0.76322, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 656ms/step - loss: 0.6986 -
accuracy: 0.7632 - val_loss: 0.7608 - val_accuracy: 0.7413 - lr: 1.0000e-04
Epoch 29/75
15/15 [==============================] - ETA: 0s - loss: 0.6904 - accuracy:
0.7644
Epoch 29: accuracy improved from 0.76322 to 0.76439, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 653ms/step - loss: 0.6904 -
accuracy: 0.7644 - val_loss: 0.7520 - val_accuracy: 0.7436 - lr: 1.0000e-04
Epoch 30/75
15/15 [==============================] - ETA: 0s - loss: 0.6720 - accuracy:
0.7737
Epoch 30: accuracy improved from 0.76439 to 0.77367, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 654ms/step - loss: 0.6720 -
accuracy: 0.7737 - val_loss: 0.7388 - val_accuracy: 0.7492 - lr: 1.0000e-04
Epoch 31/75
15/15 [==============================] - ETA: 0s - loss: 0.7013 - accuracy:
0.7635
Epoch 31: accuracy did not improve from 0.77367
15/15 [==============================] - 9s 626ms/step - loss: 0.7013 -
accuracy: 0.7635 - val_loss: 0.7128 - val_accuracy: 0.7635 - lr: 1.0000e-04
Epoch 32/75
15/15 [==============================] - ETA: 0s - loss: 0.6842 - accuracy:
0.7662
Epoch 32: accuracy did not improve from 0.77367
15/15 [==============================] - 9s 628ms/step - loss: 0.6842 -
```

```
accuracy: 0.7662 - val_loss: 0.7224 - val_accuracy: 0.7592 - lr: 1.0000e-04
Epoch 33/75
15/15 [==============================] - ETA: 0s - loss: 0.6925 - accuracy:
0.7652
Epoch 33: accuracy did not improve from 0.77367
15/15 [==============================] - 9s 626ms/step - loss: 0.6925 -
accuracy: 0.7652 - val_loss: 0.7415 - val_accuracy: 0.7462 - lr: 1.0000e-04
Epoch 34/75
15/15 [==============================] - ETA: 0s - loss: 0.7010 - accuracy:
0.7613
Epoch 34: accuracy did not improve from 0.77367
15/15 [==============================] - 9s 626ms/step - loss: 0.7010 -
accuracy: 0.7613 - val_loss: 0.7185 - val_accuracy: 0.7559 - lr: 1.0000e-04
Epoch 35/75
15/15 [==============================] - ETA: 0s - loss: 0.6596 - accuracy:
0.7748
Epoch 35: accuracy improved from 0.77367 to 0.77478, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 686ms/step - loss: 0.6596 -
accuracy: 0.7748 - val_loss: 0.6993 - val_accuracy: 0.7696 - lr: 1.0000e-04
Epoch 36/75
15/15 [==============================] - ETA: 0s - loss: 0.6591 - accuracy:
0.7710
Epoch 36: accuracy did not improve from 0.77478
15/15 [==============================] - 9s 625ms/step - loss: 0.6591 -
accuracy: 0.7710 - val_loss: 0.7236 - val_accuracy: 0.7541 - lr: 1.0000e-04
Epoch 37/75
15/15 [==============================] - ETA: 0s - loss: 0.6797 - accuracy:
0.7651
Epoch 37: accuracy did not improve from 0.77478
15/15 [==============================] - 9s 627ms/step - loss: 0.6797 -
accuracy: 0.7651 - val_loss: 0.6872 - val_accuracy: 0.7683 - lr: 1.0000e-04
Epoch 38/75
15/15 [==============================] - ETA: 0s - loss: 0.6739 - accuracy:
0.7697
Epoch 38: accuracy did not improve from 0.77478
15/15 [==============================] - 9s 626ms/step - loss: 0.6739 -
accuracy: 0.7697 - val_loss: 0.6996 - val_accuracy: 0.7653 - lr: 1.0000e-04
Epoch 39/75
15/15 [==============================] - ETA: 0s - loss: 0.6547 - accuracy:
0.7759
Epoch 39: accuracy improved from 0.77478 to 0.77586, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 691ms/step - loss: 0.6547 -
accuracy: 0.7759 - val_loss: 0.7144 - val_accuracy: 0.7589 - lr: 1.0000e-04
Epoch 40/75
```

```
15/15 [==============================] - ETA: 0s - loss: 0.6740 - accuracy:
0.7696
Epoch 40: accuracy did not improve from 0.77586
15/15 [==============================] - 9s 627ms/step - loss: 0.6740 -
accuracy: 0.7696 - val_loss: 0.6729 - val_accuracy: 0.7768 - lr: 1.0000e-04
Epoch 41/75
15/15 [==============================] - ETA: 0s - loss: 0.6693 - accuracy:
0.7703
Epoch 41: accuracy did not improve from 0.77586
15/15 [==============================] - 9s 624ms/step - loss: 0.6693 -
accuracy: 0.7703 - val_loss: 0.6954 - val_accuracy: 0.7684 - lr: 1.0000e-04
Epoch 42/75
15/15 [==============================] - ETA: 0s - loss: 0.6571 - accuracy:
0.7755
Epoch 42: accuracy did not improve from 0.77586
15/15 [==============================] - 9s 626ms/step - loss: 0.6571 -
accuracy: 0.7755 - val_loss: 0.6681 - val_accuracy: 0.7812 - lr: 1.0000e-04
Epoch 43/75
15/15 [==============================] - ETA: 0s - loss: 0.6734 - accuracy:
0.7695
Epoch 43: accuracy did not improve from 0.77586
15/15 [==============================] - 9s 625ms/step - loss: 0.6734 -
accuracy: 0.7695 - val_loss: 0.6732 - val_accuracy: 0.7779 - lr: 1.0000e-04
Epoch 44/75
15/15 [==============================] - ETA: 0s - loss: 0.6659 - accuracy:
0.7721
Epoch 44: accuracy did not improve from 0.77586
15/15 [==============================] - 9s 625ms/step - loss: 0.6659 -
accuracy: 0.7721 - val_loss: 0.6705 - val_accuracy: 0.7800 - lr: 1.0000e-04
Epoch 45/75
15/15 [==============================] - ETA: 0s - loss: 0.6457 - accuracy:
0.7790
Epoch 45: accuracy improved from 0.77586 to 0.77905, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 685ms/step - loss: 0.6457 -
accuracy: 0.7790 - val_loss: 0.6615 - val_accuracy: 0.7829 - lr: 1.0000e-05
Epoch 46/75
15/15 [==============================] - ETA: 0s - loss: 0.6373 - accuracy:
0.7828
Epoch 46: accuracy improved from 0.77905 to 0.78279, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 654ms/step - loss: 0.6373 -
accuracy: 0.7828 - val_loss: 0.6529 - val_accuracy: 0.7862 - lr: 1.0000e-05
Epoch 47/75
15/15 [==============================] - ETA: 0s - loss: 0.6435 - accuracy:
0.7797
```

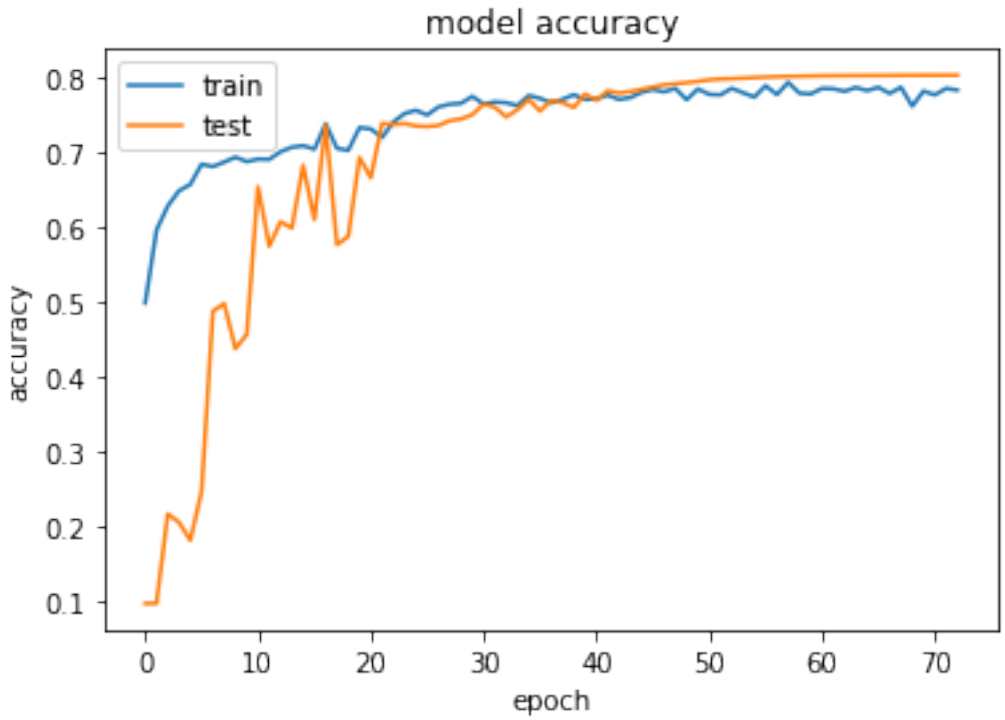
```
Epoch 47: accuracy did not improve from 0.78279
15/15 [==============================] - 9s 626ms/step - loss: 0.6435 -
accuracy: 0.7797 - val_loss: 0.6438 - val_accuracy: 0.7891 - lr: 1.0000e-05
Epoch 48/75
15/15 [==============================] - ETA: 0s - loss: 0.6304 - accuracy:
0.7844
Epoch 48: accuracy improved from 0.78279 to 0.78441, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 677ms/step - loss: 0.6304 -
accuracy: 0.7844 - val_loss: 0.6378 - val_accuracy: 0.7905 - lr: 1.0000e-05
Epoch 49/75
15/15 [==============================] - ETA: 0s - loss: 0.6702 - accuracy:
0.7691
Epoch 49: accuracy did not improve from 0.78441
15/15 [==============================] - 9s 624ms/step - loss: 0.6702 -
accuracy: 0.7691 - val_loss: 0.6341 - val_accuracy: 0.7919 - lr: 1.0000e-05
Epoch 50/75
15/15 [==============================] - ETA: 0s - loss: 0.6331 - accuracy:
0.7834
Epoch 50: accuracy did not improve from 0.78441
15/15 [==============================] - 9s 624ms/step - loss: 0.6331 -
accuracy: 0.7834 - val_loss: 0.6293 - val_accuracy: 0.7938 - lr: 1.0000e-05
Epoch 51/75
15/15 [==============================] - ETA: 0s - loss: 0.6558 - accuracy:
0.7764
Epoch 51: accuracy did not improve from 0.78441
15/15 [==============================] - 9s 627ms/step - loss: 0.6558 -
accuracy: 0.7764 - val_loss: 0.6248 - val_accuracy: 0.7957 - lr: 1.0000e-05
Epoch 52/75
15/15 [==============================] - ETA: 0s - loss: 0.6536 - accuracy:
0.7757
Epoch 52: accuracy did not improve from 0.78441
15/15 [==============================] - 9s 626ms/step - loss: 0.6536 -
accuracy: 0.7757 - val_loss: 0.6219 - val_accuracy: 0.7969 - lr: 1.0000e-05
Epoch 53/75
15/15 [==============================] - ETA: 0s - loss: 0.6306 - accuracy:
0.7844
Epoch 53: accuracy improved from 0.78441 to 0.78444, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 688ms/step - loss: 0.6306 -
accuracy: 0.7844 - val_loss: 0.6200 - val_accuracy: 0.7972 - lr: 1.0000e-05
Epoch 54/75
15/15 [==============================] - ETA: 0s - loss: 0.6472 - accuracy:
0.7785
Epoch 54: accuracy did not improve from 0.78444
15/15 [==============================] - 9s 624ms/step - loss: 0.6472 -
```

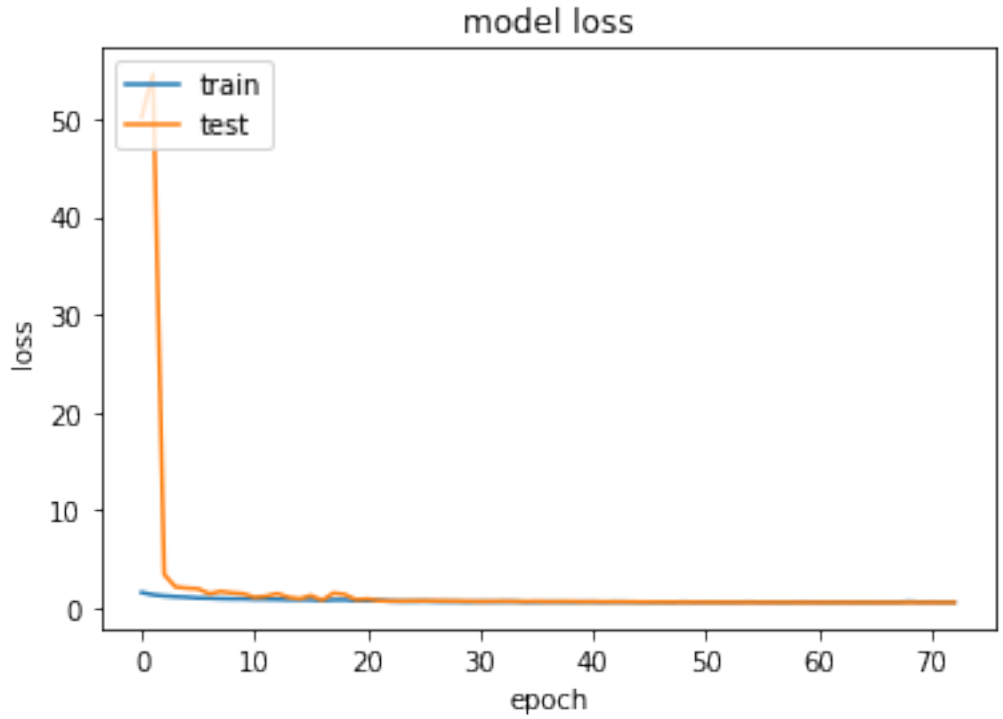
```
accuracy: 0.7785 - val_loss: 0.6176 - val_accuracy: 0.7980 - lr: 1.0000e-06
Epoch 55/75
15/15 [==============================] - ETA: 0s - loss: 0.6636 - accuracy:
0.7727
Epoch 55: accuracy did not improve from 0.78444
15/15 [==============================] - 9s 625ms/step - loss: 0.6636 -
accuracy: 0.7727 - val_loss: 0.6153 - val_accuracy: 0.7987 - lr: 1.0000e-06
Epoch 56/75
15/15 [==============================] - ETA: 0s - loss: 0.6258 - accuracy:
0.7879
Epoch 56: accuracy improved from 0.78444 to 0.78787, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 681ms/step - loss: 0.6258 -
accuracy: 0.7879 - val_loss: 0.6137 - val_accuracy: 0.7994 - lr: 1.0000e-06
Epoch 57/75
15/15 [==============================] - ETA: 0s - loss: 0.6535 - accuracy:
0.7760
Epoch 57: accuracy did not improve from 0.78787
15/15 [==============================] - 9s 625ms/step - loss: 0.6535 -
accuracy: 0.7760 - val_loss: 0.6120 - val_accuracy: 0.7999 - lr: 1.0000e-06
Epoch 58/75
15/15 [==============================] - ETA: 0s - loss: 0.6137 - accuracy:
0.7924
Epoch 58: accuracy improved from 0.78787 to 0.79241, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_highbatchsize.h5
15/15 [==============================] - 10s 690ms/step - loss: 0.6137 -
accuracy: 0.7924 - val_loss: 0.6110 - val_accuracy: 0.8003 - lr: 1.0000e-06
Epoch 59/75
15/15 [==============================] - ETA: 0s - loss: 0.6558 - accuracy:
0.7779
Epoch 59: accuracy did not improve from 0.79241
15/15 [==============================] - 9s 628ms/step - loss: 0.6558 -
accuracy: 0.7779 - val_loss: 0.6100 - val_accuracy: 0.8005 - lr: 1.0000e-06
Epoch 60/75
15/15 [==============================] - ETA: 0s - loss: 0.6532 - accuracy:
0.7769
Epoch 60: accuracy did not improve from 0.79241
15/15 [==============================] - 9s 624ms/step - loss: 0.6532 -
accuracy: 0.7769 - val_loss: 0.6092 - val_accuracy: 0.8008 - lr: 1.0000e-06
Epoch 61/75
15/15 [==============================] - ETA: 0s - loss: 0.6398 - accuracy:
0.7842
Epoch 61: accuracy did not improve from 0.79241
15/15 [==============================] - 9s 624ms/step - loss: 0.6398 -
accuracy: 0.7842 - val_loss: 0.6083 - val_accuracy: 0.8011 - lr: 1.0000e-06
Epoch 62/75
```

```
15/15 [==============================] - ETA: 0s - loss: 0.6309 - accuracy:
0.7838
Epoch 62: accuracy did not improve from 0.79241
15/15 [==============================] - 9s 626ms/step - loss: 0.6309 -
accuracy: 0.7838 - val_loss: 0.6079 - val_accuracy: 0.8012 - lr: 1.0000e-06
Epoch 63/75
15/15 [==============================] - ETA: 0s - loss: 0.6500 - accuracy:
0.7804
Epoch 63: accuracy did not improve from 0.79241
15/15 [==============================] - 9s 624ms/step - loss: 0.6500 -
accuracy: 0.7804 - val_loss: 0.6074 - val_accuracy: 0.8013 - lr: 1.0000e-06
Epoch 64/75
15/15 [==============================] - ETA: 0s - loss: 0.6314 - accuracy:
0.7855
Epoch 64: accuracy did not improve from 0.79241
15/15 [==============================] - 9s 627ms/step - loss: 0.6314 -
accuracy: 0.7855 - val_loss: 0.6069 - val_accuracy: 0.8014 - lr: 1.0000e-07
Epoch 65/75
15/15 [==============================] - ETA: 0s - loss: 0.6415 - accuracy:
0.7814
Epoch 65: accuracy did not improve from 0.79241
15/15 [==============================] - 9s 624ms/step - loss: 0.6415 -
accuracy: 0.7814 - val_loss: 0.6068 - val_accuracy: 0.8015 - lr: 1.0000e-07
Epoch 66/75
15/15 [==============================] - ETA: 0s - loss: 0.6297 - accuracy:
0.7857
Epoch 66: accuracy did not improve from 0.79241
15/15 [==============================] - 9s 623ms/step - loss: 0.6297 -
accuracy: 0.7857 - val_loss: 0.6065 - val_accuracy: 0.8016 - lr: 1.0000e-07
Epoch 67/75
15/15 [==============================] - ETA: 0s - loss: 0.6466 - accuracy:
0.7775
Epoch 67: accuracy did not improve from 0.79241
15/15 [==============================] - 9s 624ms/step - loss: 0.6466 -
accuracy: 0.7775 - val_loss: 0.6061 - val_accuracy: 0.8016 - lr: 1.0000e-07
Epoch 68/75
15/15 [==============================] - ETA: 0s - loss: 0.6276 - accuracy:
0.7859
Epoch 68: accuracy did not improve from 0.79241
15/15 [==============================] - 9s 625ms/step - loss: 0.6276 -
accuracy: 0.7859 - val_loss: 0.6060 - val_accuracy: 0.8018 - lr: 1.0000e-07
Epoch 69/75
15/15 [==============================] - ETA: 0s - loss: 0.6965 - accuracy:
0.7606
Epoch 69: accuracy did not improve from 0.79241
15/15 [==============================] - 9s 624ms/step - loss: 0.6965 -
accuracy: 0.7606 - val_loss: 0.6056 - val_accuracy: 0.8018 - lr: 1.0000e-08
Epoch 70/75
```

```
15/15 [==============================] - ETA: 0s - loss: 0.6359 - accuracy:
0.7808
Epoch 70: accuracy did not improve from 0.79241
15/15 [==============================] - 9s 626ms/step - loss: 0.6359 -
accuracy: 0.7808 - val_loss: 0.6056 - val_accuracy: 0.8018 - lr: 1.0000e-08
Epoch 71/75
15/15 [==============================] - ETA: 0s - loss: 0.6581 - accuracy:
0.7756
Epoch 71: accuracy did not improve from 0.79241
15/15 [==============================] - 9s 630ms/step - loss: 0.6581 -
accuracy: 0.7756 - val_loss: 0.6055 - val_accuracy: 0.8019 - lr: 1.0000e-08
Epoch 72/75
15/15 [==============================] - ETA: 0s - loss: 0.6323 - accuracy:
0.7843
Epoch 72: accuracy did not improve from 0.79241
15/15 [==============================] - 9s 624ms/step - loss: 0.6323 -
accuracy: 0.7843 - val_loss: 0.6054 - val_accuracy: 0.8020 - lr: 1.0000e-08
Epoch 73/75
15/15 [==============================] - ETA: 0s - loss: 0.6456 - accuracy:
0.7821
Epoch 73: accuracy did not improve from 0.79241
15/15 [==============================] - 10s 632ms/step - loss: 0.6456 -
accuracy: 0.7821 - val_loss: 0.6051 - val_accuracy: 0.8020 - lr: 1.0000e-08
```

[ ]:
```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

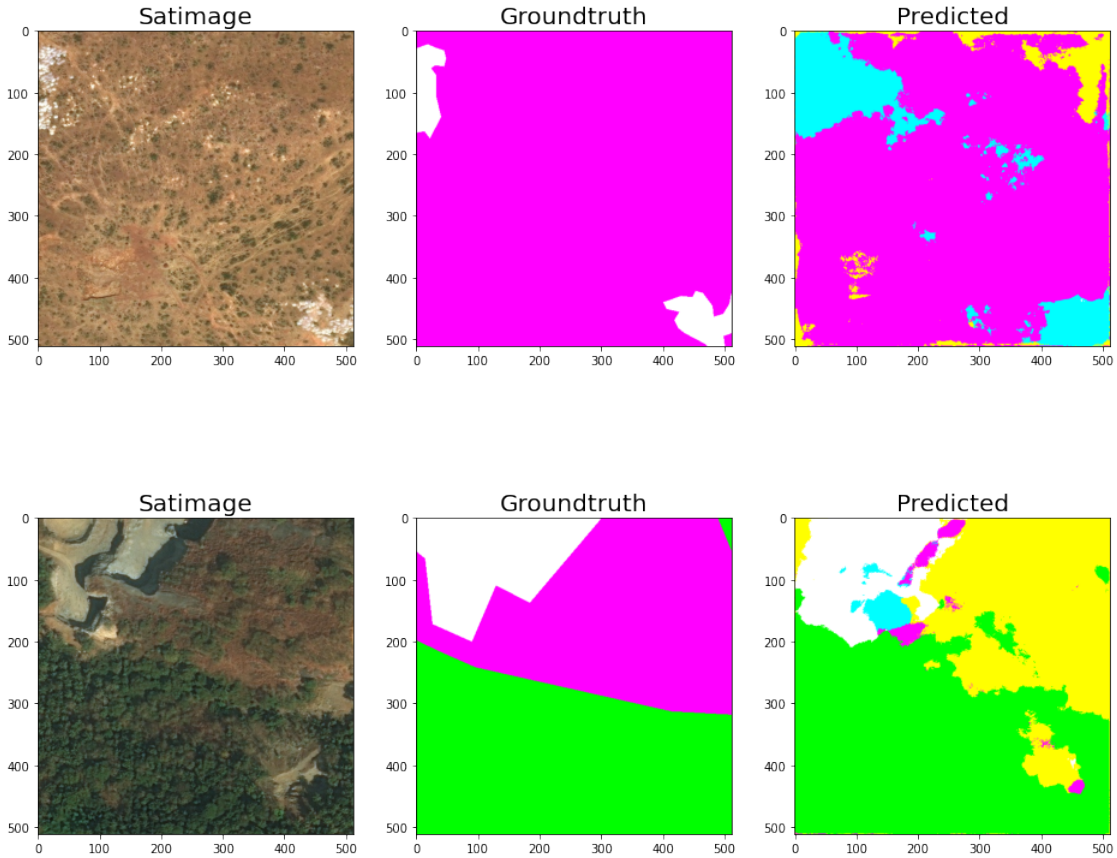

model accuracy
train
test
accuracy
epoch
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
10
20
30
40
50
60
70

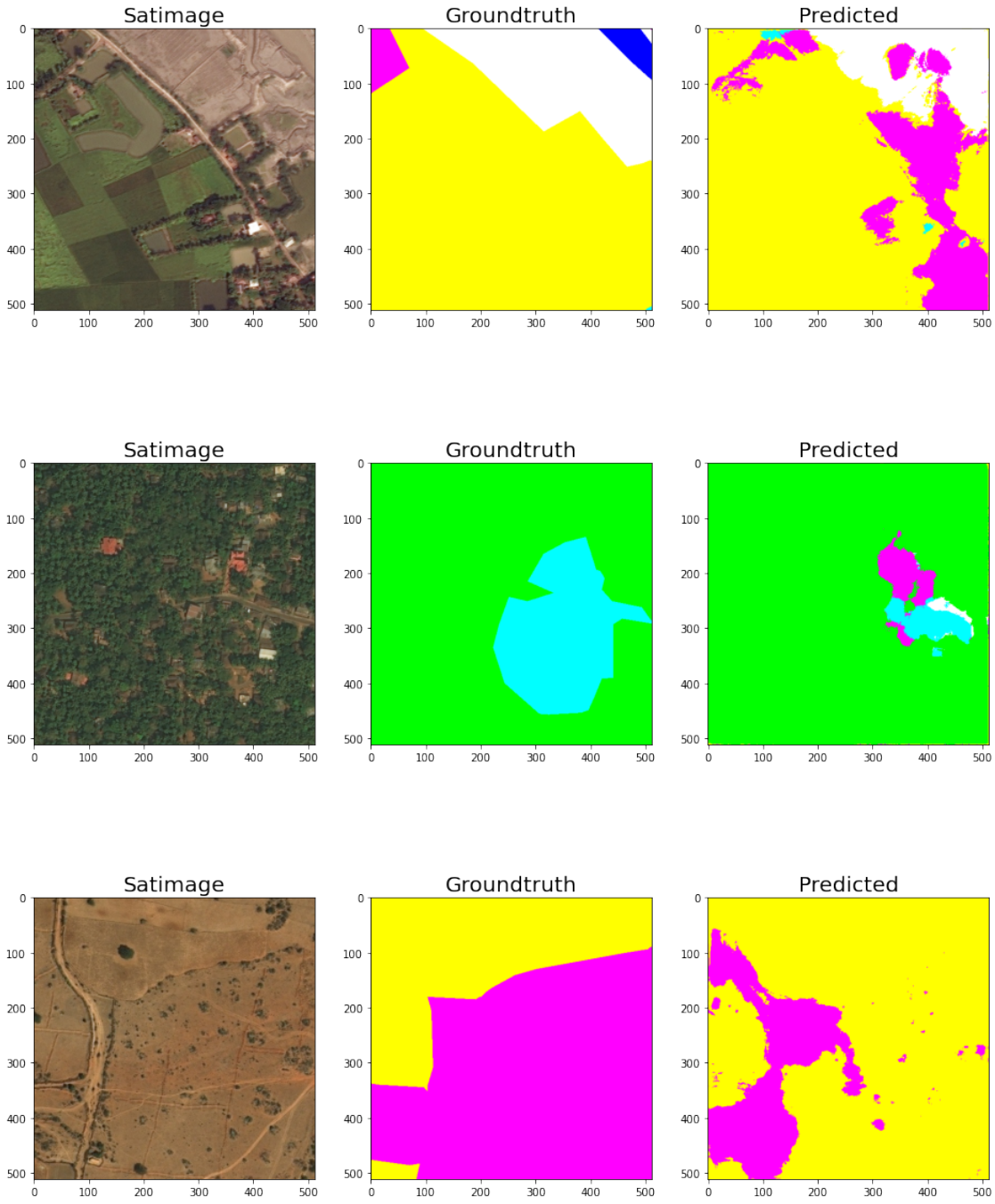

model loss
train
test
loss
epoch
0
10
20
30
40
50
0
10
20
30
40
50
60
70

[ ]:
```
score = model.evaluate(test_iterator, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
Test loss: 0.6110191941261292
Test accuracy: 0.8003209829330444
```

[ ]:
```
y_pred = model.predict(X_test)

for i in range(5):
  visualize(satImage = X_test[i],
            groundTruth=(onehot_to_rgb(y_test[i], class_dict)),
            predicted=(onehot_to_rgb(y_pred[i], class_dict)))
```





Testing different loss functions and optimizers

[ ]:
```
# Initialize ImageDataGenerator
datagen = ImageDataGenerator()

Y_train = X_train.astype('float')
X_test = X_test.astype('float')
```

```
y_train = y_train.astype('float')
y_test = y_test.astype('float')


#X_train = X_train.astype('uint8')
#X_test = X_test.astype('uint8')

#y_train = y_train.astype('uint8')
#y_test = y_test.astype('uint8')



# prepare an iterators to scale images
train_iterator = datagen.flow(X_train, y_train, batch_size=25)
test_iterator = datagen.flow(X_test, y_test, batch_size=25)
print('Batches train=%d, test=%d' % (len(train_iterator), len(test_iterator)))
```

```
Batches train=15, test=5
```

[ ]:
```
#hyperparamters
shape = (512, 512, 3)
num_classes = 7
lr = 1e-7
batch_size = 25 # johnathan testing more than 1 image in a batch
epochs = 75 # expanding epochs to see if accuracy improves over time

train_steps = len(X_train)//batch_size
test_steps = len(X_test)//batch_size
print(train_steps)
print(test_steps)
```

```
15
5
```

[ ]:
```
# save model here
path = "/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/
 ↪650_RemoteSensing/LULC_FinalProject"
model_path = '{}/models'.format(path)
model_file = '{}/custom_unet_Dice.h5'.format(model_path)

checkpoint = ModelCheckpoint(filepath = model_file,
                             monitor = 'accuracy',
                             save_best_only = True,
                             verbose=1
                             )
```

```
early_stop = EarlyStopping(monitor = 'accuracy',
                           patience = 15,
                           restore_best_weights = True)

reduce_LR = ReduceLROnPlateau(monitor = 'accuracy',
                              factor = 0.1,
                              patience = 5)

callback_list = [reduce_LR, early_stop, checkpoint]
```

[ ]:
```
!pip install focal-loss
dice_loss = sm.losses.DiceLoss(class_weights=np.ones(7)) #number of classes
focal_loss = sm.losses.CategoricalFocalLoss()
total_loss = dice_loss + (1 * focal_loss)

#metrics = [sm.metrics.IOUScore(threshold=0.5), sm.metrics.FScore(threshold=0.
↪5)]
```

```
Requirement already satisfied: focal-loss in /usr/local/lib/python3.7/dist-
packages (0.0.7)
Requirement already satisfied: tensorflow>=2.2 in /usr/local/lib/python3.7/dist-
packages (from focal-loss) (2.8.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/usr/local/lib/python3.7/dist-packages (from tensorflow>=2.2->focal-loss)
(4.2.0)
Requirement already satisfied: absl-py>=0.4.0 in /usr/local/lib/python3.7/dist-
packages (from tensorflow>=2.2->focal-loss) (1.0.0)
Requirement already satisfied: tensorboard<2.9,>=2.8 in
/usr/local/lib/python3.7/dist-packages (from tensorflow>=2.2->focal-loss)
(2.8.0)
Requirement already satisfied: termcolor>=1.1.0 in
/usr/local/lib/python3.7/dist-packages (from tensorflow>=2.2->focal-loss)
(1.1.0)
Requirement already satisfied: libclang>=9.0.1 in /usr/local/lib/python3.7/dist-
packages (from tensorflow>=2.2->focal-loss) (14.0.1)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.7/dist-
packages (from tensorflow>=2.2->focal-loss) (1.21.6)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/usr/local/lib/python3.7/dist-packages (from tensorflow>=2.2->focal-loss)
(0.25.0)
Requirement already satisfied: gast>=0.2.1 in /usr/local/lib/python3.7/dist-
packages (from tensorflow>=2.2->focal-loss) (0.5.3)
Requirement already satisfied: keras-preprocessing>=1.1.1 in
/usr/local/lib/python3.7/dist-packages (from tensorflow>=2.2->focal-loss)
(1.1.2)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.7/dist-
packages (from tensorflow>=2.2->focal-loss) (1.15.0)
```

```
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.7/dist-
packages (from tensorflow>=2.2->focal-loss) (1.14.0)
Requirement already satisfied: opt-einsum>=2.3.2 in
/usr/local/lib/python3.7/dist-packages (from tensorflow>=2.2->focal-loss)
(3.3.0)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.7/dist-
packages (from tensorflow>=2.2->focal-loss) (3.1.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/usr/local/lib/python3.7/dist-packages (from tensorflow>=2.2->focal-loss)
(1.44.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/usr/local/lib/python3.7/dist-packages (from tensorflow>=2.2->focal-loss)
(0.2.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-
packages (from tensorflow>=2.2->focal-loss) (57.4.0)
Requirement already satisfied: protobuf>=3.9.2 in /usr/local/lib/python3.7/dist-
packages (from tensorflow>=2.2->focal-loss) (3.17.3)
Requirement already satisfied: flatbuffers>=1.12 in
/usr/local/lib/python3.7/dist-packages (from tensorflow>=2.2->focal-loss) (2.0)
Requirement already satisfied: astunparse>=1.6.0 in
/usr/local/lib/python3.7/dist-packages (from tensorflow>=2.2->focal-loss)
(1.6.3)
Requirement already satisfied: tf-estimator-nightly==2.8.0.dev2021122109 in
/usr/local/lib/python3.7/dist-packages (from tensorflow>=2.2->focal-loss)
(2.8.0.dev2021122109)
Requirement already satisfied: keras<2.9,>=2.8.0rc0 in
/usr/local/lib/python3.7/dist-packages (from tensorflow>=2.2->focal-loss)
(2.8.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/usr/local/lib/python3.7/dist-packages (from
astunparse>=1.6.0->tensorflow>=2.2->focal-loss) (0.37.1)
Requirement already satisfied: cached-property in /usr/local/lib/python3.7/dist-
packages (from h5py>=2.9.0->tensorflow>=2.2->focal-loss) (1.5.2)
Requirement already satisfied: werkzeug>=0.11.15 in
/usr/local/lib/python3.7/dist-packages (from
tensorboard<2.9,>=2.8->tensorflow>=2.2->focal-loss) (1.0.1)
Requirement already satisfied: requests<3,>=2.21.0 in
/usr/local/lib/python3.7/dist-packages (from
tensorboard<2.9,>=2.8->tensorflow>=2.2->focal-loss) (2.23.0)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-
packages (from tensorboard<2.9,>=2.8->tensorflow>=2.2->focal-loss) (3.3.6)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in
/usr/local/lib/python3.7/dist-packages (from
tensorboard<2.9,>=2.8->tensorflow>=2.2->focal-loss) (0.4.6)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/usr/local/lib/python3.7/dist-packages (from
tensorboard<2.9,>=2.8->tensorflow>=2.2->focal-loss) (1.35.0)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in
```

```
/usr/local/lib/python3.7/dist-packages (from
tensorboard<2.9,>=2.8->tensorflow>=2.2->focal-loss) (0.6.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
/usr/local/lib/python3.7/dist-packages (from
tensorboard<2.9,>=2.8->tensorflow>=2.2->focal-loss) (1.8.1)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in
/usr/local/lib/python3.7/dist-packages (from google-
auth<3,>=1.6.3->tensorboard<2.9,>=2.8->tensorflow>=2.2->focal-loss) (4.2.4)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/usr/local/lib/python3.7/dist-packages (from google-
auth<3,>=1.6.3->tensorboard<2.9,>=2.8->tensorflow>=2.2->focal-loss) (0.2.8)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.7/dist-
packages (from google-
auth<3,>=1.6.3->tensorboard<2.9,>=2.8->tensorflow>=2.2->focal-loss) (4.8)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/usr/local/lib/python3.7/dist-packages (from google-auth-
oauthlib<0.5,>=0.4.1->tensorboard<2.9,>=2.8->tensorflow>=2.2->focal-loss)
(1.3.1)
Requirement already satisfied: importlib-metadata>=4.4 in
/usr/local/lib/python3.7/dist-packages (from
markdown>=2.6.8->tensorboard<2.9,>=2.8->tensorflow>=2.2->focal-loss) (4.11.3)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-
packages (from importlib-
metadata>=4.4->markdown>=2.6.8->tensorboard<2.9,>=2.8->tensorflow>=2.2->focal-
loss) (3.8.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in
/usr/local/lib/python3.7/dist-packages (from pyasn1-modules>=0.2.1->google-
auth<3,>=1.6.3->tensorboard<2.9,>=2.8->tensorflow>=2.2->focal-loss) (0.4.8)
Requirement already satisfied: chardet<4,>=3.0.2 in
/usr/local/lib/python3.7/dist-packages (from
requests<3,>=2.21.0->tensorboard<2.9,>=2.8->tensorflow>=2.2->focal-loss) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.7/dist-packages (from
requests<3,>=2.21.0->tensorboard<2.9,>=2.8->tensorflow>=2.2->focal-loss)
(2021.10.8)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-
packages (from
requests<3,>=2.21.0->tensorboard<2.9,>=2.8->tensorflow>=2.2->focal-loss) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.7/dist-packages (from
requests<3,>=2.21.0->tensorboard<2.9,>=2.8->tensorflow>=2.2->focal-loss)
(1.24.3)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-
packages (from requests-oauthlib>=0.7.0->google-auth-
oauthlib<0.5,>=0.4.1->tensorboard<2.9,>=2.8->tensorflow>=2.2->focal-loss)
(3.2.0)
```

[ ]:
```
model = build_unet(shape, num_classes)
model.compile(loss=total_loss, metrics=["accuracy"], optimizer=Adam())
```

[ ]:
```
# Run the model
history = model.fit(train_iterator,
        steps_per_epoch=train_steps,
        validation_data=(X_test, y_test),
        validation_steps=test_steps,
        epochs=epochs,
        callbacks=callback_list,
        # verbose=1
    )
```

```
Epoch 1/75
15/15 [==============================] - ETA: 0s - loss: 0.8742 - accuracy:
0.4161
Epoch 1: val_accuracy did not improve from 0.77216
15/15 [==============================] - 18s 963ms/step - loss: 0.8742 -
accuracy: 0.4161 - val_loss: 1.1263 - val_accuracy: 0.0972 - lr: 0.0010
Epoch 2/75
15/15 [==============================] - ETA: 0s - loss: 0.7912 - accuracy:
0.5781
Epoch 2: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 695ms/step - loss: 0.7912 -
accuracy: 0.5781 - val_loss: 1.3338 - val_accuracy: 0.0973 - lr: 0.0010
Epoch 3/75
15/15 [==============================] - ETA: 0s - loss: 0.7506 - accuracy:
0.6028
Epoch 3: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 693ms/step - loss: 0.7506 -
accuracy: 0.6028 - val_loss: 1.4549 - val_accuracy: 0.1019 - lr: 0.0010
Epoch 4/75
15/15 [==============================] - ETA: 0s - loss: 0.7322 - accuracy:
0.6088
Epoch 4: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 691ms/step - loss: 0.7322 -
accuracy: 0.6088 - val_loss: 1.0800 - val_accuracy: 0.1556 - lr: 0.0010
Epoch 5/75
15/15 [==============================] - ETA: 0s - loss: 0.7114 - accuracy:
0.6226
Epoch 5: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 698ms/step - loss: 0.7114 -
accuracy: 0.6226 - val_loss: 0.9844 - val_accuracy: 0.2101 - lr: 0.0010
Epoch 6/75
15/15 [==============================] - ETA: 0s - loss: 0.7038 - accuracy:
0.6156
Epoch 6: val_accuracy did not improve from 0.77216
```

```
15/15 [==============================] - 11s 694ms/step - loss: 0.7038 -
accuracy: 0.6156 - val_loss: 0.8669 - val_accuracy: 0.5014 - lr: 0.0010
Epoch 7/75
15/15 [==============================] - ETA: 0s - loss: 0.6861 - accuracy:
0.6276
Epoch 7: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 693ms/step - loss: 0.6861 -
accuracy: 0.6276 - val_loss: 0.9080 - val_accuracy: 0.3618 - lr: 0.0010
Epoch 8/75
15/15 [==============================] - ETA: 0s - loss: 0.6701 - accuracy:
0.6335
Epoch 8: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 690ms/step - loss: 0.6701 -
accuracy: 0.6335 - val_loss: 0.9678 - val_accuracy: 0.2566 - lr: 0.0010
Epoch 9/75
15/15 [==============================] - ETA: 0s - loss: 0.6772 - accuracy:
0.6097
Epoch 9: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 689ms/step - loss: 0.6772 -
accuracy: 0.6097 - val_loss: 0.8381 - val_accuracy: 0.5001 - lr: 0.0010
Epoch 10/75
15/15 [==============================] - ETA: 0s - loss: 0.6462 - accuracy:
0.6578
Epoch 10: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 692ms/step - loss: 0.6462 -
accuracy: 0.6578 - val_loss: 0.7310 - val_accuracy: 0.6281 - lr: 0.0010
Epoch 11/75
15/15 [==============================] - ETA: 0s - loss: 0.6319 - accuracy:
0.6553
Epoch 11: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 689ms/step - loss: 0.6319 -
accuracy: 0.6553 - val_loss: 0.8676 - val_accuracy: 0.4290 - lr: 0.0010
Epoch 12/75
15/15 [==============================] - ETA: 0s - loss: 0.6134 - accuracy:
0.6662
Epoch 12: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 694ms/step - loss: 0.6134 -
accuracy: 0.6662 - val_loss: 0.7179 - val_accuracy: 0.6428 - lr: 0.0010
Epoch 13/75
15/15 [==============================] - ETA: 0s - loss: 0.6096 - accuracy:
0.6676
Epoch 13: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 688ms/step - loss: 0.6096 -
accuracy: 0.6676 - val_loss: 0.8491 - val_accuracy: 0.3796 - lr: 0.0010
Epoch 14/75
15/15 [==============================] - ETA: 0s - loss: 0.6081 - accuracy:
0.6783
Epoch 14: val_accuracy did not improve from 0.77216
```

```
15/15 [==============================] - 11s 690ms/step - loss: 0.6081 -
accuracy: 0.6783 - val_loss: 0.7100 - val_accuracy: 0.5924 - lr: 0.0010
Epoch 15/75
15/15 [==============================] - ETA: 0s - loss: 0.5898 - accuracy:
0.6763
Epoch 15: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 694ms/step - loss: 0.5898 -
accuracy: 0.6763 - val_loss: 0.9285 - val_accuracy: 0.3851 - lr: 0.0010
Epoch 16/75
15/15 [==============================] - ETA: 0s - loss: 0.5876 - accuracy:
0.6867
Epoch 16: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 690ms/step - loss: 0.5876 -
accuracy: 0.6867 - val_loss: 0.8100 - val_accuracy: 0.4866 - lr: 0.0010
Epoch 17/75
15/15 [==============================] - ETA: 0s - loss: 0.5827 - accuracy:
0.6875
Epoch 17: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 692ms/step - loss: 0.5827 -
accuracy: 0.6875 - val_loss: 0.8440 - val_accuracy: 0.3234 - lr: 0.0010
Epoch 18/75
15/15 [==============================] - ETA: 0s - loss: 0.5653 - accuracy:
0.7116
Epoch 18: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 694ms/step - loss: 0.5653 -
accuracy: 0.7116 - val_loss: 0.7072 - val_accuracy: 0.4865 - lr: 1.0000e-04
Epoch 19/75
15/15 [==============================] - ETA: 0s - loss: 0.5560 - accuracy:
0.7055
Epoch 19: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 691ms/step - loss: 0.5560 -
accuracy: 0.7055 - val_loss: 0.6676 - val_accuracy: 0.5357 - lr: 1.0000e-04
Epoch 20/75
15/15 [==============================] - ETA: 0s - loss: 0.5366 - accuracy:
0.7274
Epoch 20: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 691ms/step - loss: 0.5366 -
accuracy: 0.7274 - val_loss: 0.6268 - val_accuracy: 0.6302 - lr: 1.0000e-04
Epoch 21/75
15/15 [==============================] - ETA: 0s - loss: 0.5578 - accuracy:
0.7173
Epoch 21: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 692ms/step - loss: 0.5578 -
accuracy: 0.7173 - val_loss: 0.6241 - val_accuracy: 0.6526 - lr: 1.0000e-04
Epoch 22/75
15/15 [==============================] - ETA: 0s - loss: 0.5284 - accuracy:
0.7284
Epoch 22: val_accuracy did not improve from 0.77216
```

```
15/15 [==============================] - 11s 695ms/step - loss: 0.5284 -
accuracy: 0.7284 - val_loss: 0.6107 - val_accuracy: 0.6676 - lr: 1.0000e-04
Epoch 23/75
15/15 [==============================] - ETA: 0s - loss: 0.5374 - accuracy:
0.7294
Epoch 23: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 692ms/step - loss: 0.5374 -
accuracy: 0.7294 - val_loss: 0.5890 - val_accuracy: 0.7038 - lr: 1.0000e-04
Epoch 24/75
15/15 [==============================] - ETA: 0s - loss: 0.5302 - accuracy:
0.7309
Epoch 24: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 691ms/step - loss: 0.5302 -
accuracy: 0.7309 - val_loss: 0.5766 - val_accuracy: 0.7248 - lr: 1.0000e-04
Epoch 25/75
15/15 [==============================] - ETA: 0s - loss: 0.5338 - accuracy:
0.7320
Epoch 25: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 692ms/step - loss: 0.5338 -
accuracy: 0.7320 - val_loss: 0.5704 - val_accuracy: 0.7415 - lr: 1.0000e-04
Epoch 26/75
15/15 [==============================] - ETA: 0s - loss: 0.5333 - accuracy:
0.7303
Epoch 26: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 689ms/step - loss: 0.5333 -
accuracy: 0.7303 - val_loss: 0.5898 - val_accuracy: 0.7247 - lr: 1.0000e-04
Epoch 27/75
15/15 [==============================] - ETA: 0s - loss: 0.5238 - accuracy:
0.7414
Epoch 27: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 691ms/step - loss: 0.5238 -
accuracy: 0.7414 - val_loss: 0.5658 - val_accuracy: 0.7532 - lr: 1.0000e-04
Epoch 28/75
15/15 [==============================] - ETA: 0s - loss: 0.5148 - accuracy:
0.7429
Epoch 28: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 688ms/step - loss: 0.5148 -
accuracy: 0.7429 - val_loss: 0.5691 - val_accuracy: 0.7493 - lr: 1.0000e-04
Epoch 29/75
15/15 [==============================] - ETA: 0s - loss: 0.5084 - accuracy:
0.7456
Epoch 29: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 693ms/step - loss: 0.5084 -
accuracy: 0.7456 - val_loss: 0.5605 - val_accuracy: 0.7611 - lr: 1.0000e-04
Epoch 30/75
15/15 [==============================] - ETA: 0s - loss: 0.5227 - accuracy:
0.7449
Epoch 30: val_accuracy did not improve from 0.77216
```

```
15/15 [==============================] - 11s 690ms/step - loss: 0.5227 -
accuracy: 0.7449 - val_loss: 0.5656 - val_accuracy: 0.7605 - lr: 1.0000e-04
Epoch 31/75
15/15 [==============================] - ETA: 0s - loss: 0.5387 - accuracy:
0.7292
Epoch 31: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 690ms/step - loss: 0.5387 -
accuracy: 0.7292 - val_loss: 0.5769 - val_accuracy: 0.7520 - lr: 1.0000e-04
Epoch 32/75
15/15 [==============================] - ETA: 0s - loss: 0.5220 - accuracy:
0.7391
Epoch 32: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 690ms/step - loss: 0.5220 -
accuracy: 0.7391 - val_loss: 0.5637 - val_accuracy: 0.7577 - lr: 1.0000e-04
Epoch 33/75
15/15 [==============================] - ETA: 0s - loss: 0.5094 - accuracy:
0.7414
Epoch 33: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 692ms/step - loss: 0.5094 -
accuracy: 0.7414 - val_loss: 0.5455 - val_accuracy: 0.7675 - lr: 1.0000e-04
Epoch 34/75
15/15 [==============================] - ETA: 0s - loss: 0.5096 - accuracy:
0.7461
Epoch 34: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 688ms/step - loss: 0.5096 -
accuracy: 0.7461 - val_loss: 0.5868 - val_accuracy: 0.7419 - lr: 1.0000e-04
Epoch 35/75
15/15 [==============================] - ETA: 0s - loss: 0.5015 - accuracy:
0.7497
Epoch 35: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 689ms/step - loss: 0.5015 -
accuracy: 0.7497 - val_loss: 0.5586 - val_accuracy: 0.7645 - lr: 1.0000e-04
Epoch 36/75
15/15 [==============================] - ETA: 0s - loss: 0.5228 - accuracy:
0.7417
Epoch 36: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 688ms/step - loss: 0.5228 -
accuracy: 0.7417 - val_loss: 0.5644 - val_accuracy: 0.7612 - lr: 1.0000e-04
Epoch 37/75
15/15 [==============================] - ETA: 0s - loss: 0.5022 - accuracy:
0.7539
Epoch 37: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 690ms/step - loss: 0.5022 -
accuracy: 0.7539 - val_loss: 0.5749 - val_accuracy: 0.7477 - lr: 1.0000e-04
Epoch 38/75
15/15 [==============================] - ETA: 0s - loss: 0.5071 - accuracy:
0.7492
Epoch 38: val_accuracy did not improve from 0.77216
```

```
15/15 [==============================] - 11s 688ms/step - loss: 0.5071 -
accuracy: 0.7492 - val_loss: 0.5474 - val_accuracy: 0.7661 - lr: 1.0000e-04
Epoch 39/75
15/15 [==============================] - ETA: 0s - loss: 0.4915 - accuracy:
0.7579
Epoch 39: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 692ms/step - loss: 0.4915 -
accuracy: 0.7579 - val_loss: 0.5443 - val_accuracy: 0.7684 - lr: 1.0000e-05
Epoch 40/75
15/15 [==============================] - ETA: 0s - loss: 0.5028 - accuracy:
0.7533
Epoch 40: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 695ms/step - loss: 0.5028 -
accuracy: 0.7533 - val_loss: 0.5430 - val_accuracy: 0.7697 - lr: 1.0000e-05
Epoch 41/75
15/15 [==============================] - ETA: 0s - loss: 0.5085 - accuracy:
0.7438
Epoch 41: val_accuracy did not improve from 0.77216
15/15 [==============================] - 11s 692ms/step - loss: 0.5085 -
accuracy: 0.7438 - val_loss: 0.5405 - val_accuracy: 0.7709 - lr: 1.0000e-05
Epoch 42/75
15/15 [==============================] - ETA: 0s - loss: 0.4940 - accuracy:
0.7541
Epoch 42: val_accuracy improved from 0.77216 to 0.77248, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_Dice.h5
15/15 [==============================] - 11s 754ms/step - loss: 0.4940 -
accuracy: 0.7541 - val_loss: 0.5370 - val_accuracy: 0.7725 - lr: 1.0000e-05
Epoch 43/75
15/15 [==============================] - ETA: 0s - loss: 0.4981 - accuracy:
0.7571
Epoch 43: val_accuracy improved from 0.77248 to 0.77369, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_Dice.h5
15/15 [==============================] - 11s 721ms/step - loss: 0.4981 -
accuracy: 0.7571 - val_loss: 0.5338 - val_accuracy: 0.7737 - lr: 1.0000e-05
Epoch 44/75
15/15 [==============================] - ETA: 0s - loss: 0.5037 - accuracy:
0.7531
Epoch 44: val_accuracy improved from 0.77369 to 0.77542, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_Dice.h5
15/15 [==============================] - 11s 721ms/step - loss: 0.5037 -
accuracy: 0.7531 - val_loss: 0.5308 - val_accuracy: 0.7754 - lr: 1.0000e-05
Epoch 45/75
15/15 [==============================] - ETA: 0s - loss: 0.5008 - accuracy:
0.7534
Epoch 45: val_accuracy improved from 0.77542 to 0.77715, saving model to
```

```
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_Dice.h5
15/15 [==============================] - 12s 760ms/step - loss: 0.5008 -
accuracy: 0.7534 - val_loss: 0.5314 - val_accuracy: 0.7772 - lr: 1.0000e-05
Epoch 46/75
15/15 [==============================] - ETA: 0s - loss: 0.4929 - accuracy:
0.7600
Epoch 46: val_accuracy improved from 0.77715 to 0.77934, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_Dice.h5
15/15 [==============================] - 11s 719ms/step - loss: 0.4929 -
accuracy: 0.7600 - val_loss: 0.5290 - val_accuracy: 0.7793 - lr: 1.0000e-05
Epoch 47/75
15/15 [==============================] - ETA: 0s - loss: 0.4880 - accuracy:
0.7642
Epoch 47: val_accuracy improved from 0.77934 to 0.78003, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_Dice.h5
15/15 [==============================] - 11s 728ms/step - loss: 0.4880 -
accuracy: 0.7642 - val_loss: 0.5276 - val_accuracy: 0.7800 - lr: 1.0000e-05
Epoch 48/75
15/15 [==============================] - ETA: 0s - loss: 0.4978 - accuracy:
0.7574
Epoch 48: val_accuracy improved from 0.78003 to 0.78068, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_Dice.h5
15/15 [==============================] - 11s 718ms/step - loss: 0.4978 -
accuracy: 0.7574 - val_loss: 0.5259 - val_accuracy: 0.7807 - lr: 1.0000e-05
Epoch 49/75
15/15 [==============================] - ETA: 0s - loss: 0.4813 - accuracy:
0.7645
Epoch 49: val_accuracy improved from 0.78068 to 0.78162, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_Dice.h5
15/15 [==============================] - 11s 749ms/step - loss: 0.4813 -
accuracy: 0.7645 - val_loss: 0.5252 - val_accuracy: 0.7816 - lr: 1.0000e-05
Epoch 50/75
15/15 [==============================] - ETA: 0s - loss: 0.5053 - accuracy:
0.7459
Epoch 50: val_accuracy did not improve from 0.78162
15/15 [==============================] - 11s 692ms/step - loss: 0.5053 -
accuracy: 0.7459 - val_loss: 0.5252 - val_accuracy: 0.7806 - lr: 1.0000e-05
Epoch 51/75
15/15 [==============================] - ETA: 0s - loss: 0.4901 - accuracy:
0.7615
Epoch 51: val_accuracy did not improve from 0.78162
15/15 [==============================] - 11s 694ms/step - loss: 0.4901 -
accuracy: 0.7615 - val_loss: 0.5238 - val_accuracy: 0.7807 - lr: 1.0000e-05
```

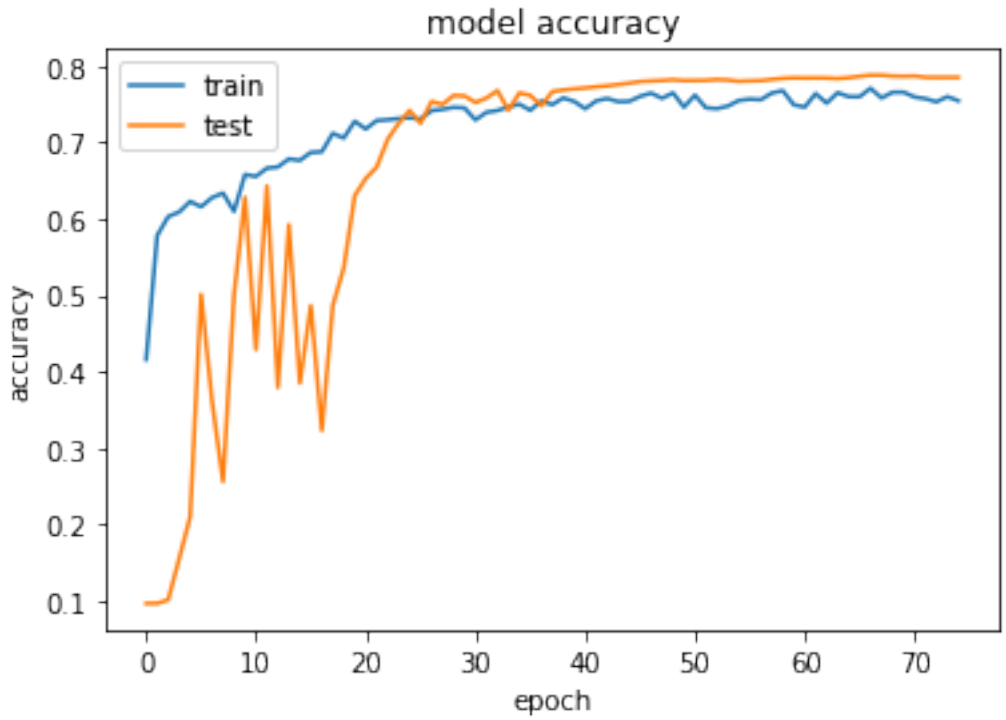
```
Epoch 52/75
15/15 [==============================] - ETA: 0s - loss: 0.5079 - accuracy:
0.7451
Epoch 52: val_accuracy did not improve from 0.78162
15/15 [==============================] - 11s 690ms/step - loss: 0.5079 -
accuracy: 0.7451 - val_loss: 0.5228 - val_accuracy: 0.7809 - lr: 1.0000e-05
Epoch 53/75
15/15 [==============================] - ETA: 0s - loss: 0.5196 - accuracy:
0.7439
Epoch 53: val_accuracy improved from 0.78162 to 0.78190, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_Dice.h5
15/15 [==============================] - 11s 743ms/step - loss: 0.5196 -
accuracy: 0.7439 - val_loss: 0.5219 - val_accuracy: 0.7819 - lr: 1.0000e-05
Epoch 54/75
15/15 [==============================] - ETA: 0s - loss: 0.5200 - accuracy:
0.7473
Epoch 54: val_accuracy did not improve from 0.78190
15/15 [==============================] - 11s 691ms/step - loss: 0.5200 -
accuracy: 0.7473 - val_loss: 0.5227 - val_accuracy: 0.7813 - lr: 1.0000e-05
Epoch 55/75
15/15 [==============================] - ETA: 0s - loss: 0.4902 - accuracy:
0.7546
Epoch 55: val_accuracy did not improve from 0.78190
15/15 [==============================] - 11s 691ms/step - loss: 0.4902 -
accuracy: 0.7546 - val_loss: 0.5266 - val_accuracy: 0.7796 - lr: 1.0000e-05
Epoch 56/75
15/15 [==============================] - ETA: 0s - loss: 0.4991 - accuracy:
0.7563
Epoch 56: val_accuracy did not improve from 0.78190
15/15 [==============================] - 11s 694ms/step - loss: 0.4991 -
accuracy: 0.7563 - val_loss: 0.5259 - val_accuracy: 0.7802 - lr: 1.0000e-05
Epoch 57/75
15/15 [==============================] - ETA: 0s - loss: 0.5025 - accuracy:
0.7556
Epoch 57: val_accuracy did not improve from 0.78190
15/15 [==============================] - 11s 690ms/step - loss: 0.5025 -
accuracy: 0.7556 - val_loss: 0.5239 - val_accuracy: 0.7805 - lr: 1.0000e-05
Epoch 58/75
15/15 [==============================] - ETA: 0s - loss: 0.4885 - accuracy:
0.7643
Epoch 58: val_accuracy improved from 0.78190 to 0.78211, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_Dice.h5
15/15 [==============================] - 11s 749ms/step - loss: 0.4885 -
accuracy: 0.7643 - val_loss: 0.5214 - val_accuracy: 0.7821 - lr: 1.0000e-05
Epoch 59/75
15/15 [==============================] - ETA: 0s - loss: 0.4812 - accuracy:
```

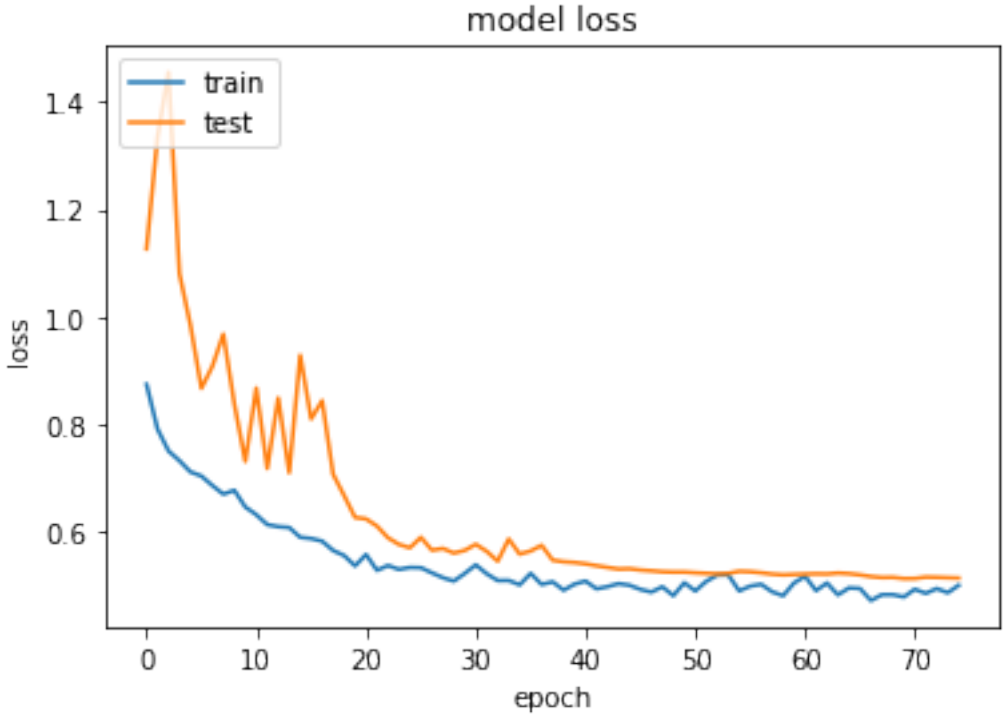
```
0.7673
Epoch 59: val_accuracy improved from 0.78211 to 0.78352, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_Dice.h5
15/15 [==============================] - 11s 722ms/step - loss: 0.4812 -
accuracy: 0.7673 - val_loss: 0.5200 - val_accuracy: 0.7835 - lr: 1.0000e-05
Epoch 60/75
15/15 [==============================] - ETA: 0s - loss: 0.5049 - accuracy:
0.7490
Epoch 60: val_accuracy improved from 0.78352 to 0.78415, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_Dice.h5
15/15 [==============================] - 11s 753ms/step - loss: 0.5049 -
accuracy: 0.7490 - val_loss: 0.5208 - val_accuracy: 0.7841 - lr: 1.0000e-05
Epoch 61/75
15/15 [==============================] - ETA: 0s - loss: 0.5173 - accuracy:
0.7459
Epoch 61: val_accuracy did not improve from 0.78415
15/15 [==============================] - 11s 689ms/step - loss: 0.5173 -
accuracy: 0.7459 - val_loss: 0.5215 - val_accuracy: 0.7839 - lr: 1.0000e-05
Epoch 62/75
15/15 [==============================] - ETA: 0s - loss: 0.4907 - accuracy:
0.7635
Epoch 62: val_accuracy did not improve from 0.78415
15/15 [==============================] - 11s 688ms/step - loss: 0.4907 -
accuracy: 0.7635 - val_loss: 0.5220 - val_accuracy: 0.7841 - lr: 1.0000e-05
Epoch 63/75
15/15 [==============================] - ETA: 0s - loss: 0.5049 - accuracy:
0.7511
Epoch 63: val_accuracy did not improve from 0.78415
15/15 [==============================] - 11s 692ms/step - loss: 0.5049 -
accuracy: 0.7511 - val_loss: 0.5216 - val_accuracy: 0.7839 - lr: 1.0000e-05
Epoch 64/75
15/15 [==============================] - ETA: 0s - loss: 0.4828 - accuracy:
0.7648
Epoch 64: val_accuracy did not improve from 0.78415
15/15 [==============================] - 11s 690ms/step - loss: 0.4828 -
accuracy: 0.7648 - val_loss: 0.5234 - val_accuracy: 0.7833 - lr: 1.0000e-05
Epoch 65/75
15/15 [==============================] - ETA: 0s - loss: 0.4957 - accuracy:
0.7596
Epoch 65: val_accuracy improved from 0.78415 to 0.78435, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_Dice.h5
15/15 [==============================] - 11s 757ms/step - loss: 0.4957 -
accuracy: 0.7596 - val_loss: 0.5223 - val_accuracy: 0.7843 - lr: 1.0000e-05
Epoch 66/75
15/15 [==============================] - ETA: 0s - loss: 0.4947 - accuracy:
```

```
0.7595
Epoch 66: val_accuracy improved from 0.78435 to 0.78615, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_Dice.h5
15/15 [==============================] - 11s 719ms/step - loss: 0.4947 -
accuracy: 0.7595 - val_loss: 0.5200 - val_accuracy: 0.7861 - lr: 1.0000e-05
Epoch 67/75
15/15 [==============================] - ETA: 0s - loss: 0.4724 - accuracy:
0.7706
Epoch 67: val_accuracy improved from 0.78615 to 0.78766, saving model to
/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/650_RemoteSensing/LULC_F
inalProject/models/custom_unet_Dice.h5
15/15 [==============================] - 11s 720ms/step - loss: 0.4724 -
accuracy: 0.7706 - val_loss: 0.5168 - val_accuracy: 0.7877 - lr: 1.0000e-05
Epoch 68/75
15/15 [==============================] - ETA: 0s - loss: 0.4831 - accuracy:
0.7578
Epoch 68: val_accuracy did not improve from 0.78766
15/15 [==============================] - 11s 689ms/step - loss: 0.4831 -
accuracy: 0.7578 - val_loss: 0.5151 - val_accuracy: 0.7876 - lr: 1.0000e-05
Epoch 69/75
15/15 [==============================] - ETA: 0s - loss: 0.4832 - accuracy:
0.7653
Epoch 69: val_accuracy did not improve from 0.78766
15/15 [==============================] - 11s 691ms/step - loss: 0.4832 -
accuracy: 0.7653 - val_loss: 0.5154 - val_accuracy: 0.7865 - lr: 1.0000e-05
Epoch 70/75
15/15 [==============================] - ETA: 0s - loss: 0.4789 - accuracy:
0.7654
Epoch 70: val_accuracy did not improve from 0.78766
15/15 [==============================] - 11s 691ms/step - loss: 0.4789 -
accuracy: 0.7654 - val_loss: 0.5131 - val_accuracy: 0.7861 - lr: 1.0000e-05
Epoch 71/75
15/15 [==============================] - ETA: 0s - loss: 0.4932 - accuracy:
0.7592
Epoch 71: val_accuracy did not improve from 0.78766
15/15 [==============================] - 11s 689ms/step - loss: 0.4932 -
accuracy: 0.7592 - val_loss: 0.5133 - val_accuracy: 0.7865 - lr: 1.0000e-05
Epoch 72/75
15/15 [==============================] - ETA: 0s - loss: 0.4856 - accuracy:
0.7569
Epoch 72: val_accuracy did not improve from 0.78766
15/15 [==============================] - 11s 689ms/step - loss: 0.4856 -
accuracy: 0.7569 - val_loss: 0.5158 - val_accuracy: 0.7849 - lr: 1.0000e-05
Epoch 73/75
15/15 [==============================] - ETA: 0s - loss: 0.4945 - accuracy:
0.7528
Epoch 73: val_accuracy did not improve from 0.78766
```

```
15/15 [==============================] - 11s 692ms/step - loss: 0.4945 -
accuracy: 0.7528 - val_loss: 0.5154 - val_accuracy: 0.7847 - lr: 1.0000e-06
Epoch 74/75
15/15 [==============================] - ETA: 0s - loss: 0.4868 - accuracy:
0.7596
Epoch 74: val_accuracy did not improve from 0.78766
15/15 [==============================] - 11s 688ms/step - loss: 0.4868 -
accuracy: 0.7596 - val_loss: 0.5147 - val_accuracy: 0.7848 - lr: 1.0000e-06
Epoch 75/75
15/15 [==============================] - ETA: 0s - loss: 0.5007 - accuracy:
0.7542
Epoch 75: val_accuracy did not improve from 0.78766
15/15 [==============================] - 11s 692ms/step - loss: 0.5007 -
accuracy: 0.7542 - val_loss: 0.5141 - val_accuracy: 0.7848 - lr: 1.0000e-06
```

[ ]:
```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```


model accuracy
train
test
accuracy
epoch
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
10
20
30
40
50
60
70


model loss
train
test
loss
epoch
0.6
0.8
1.0
1.2
1.4
0
10
20
30
40
50
60
70

[ ]:
```
score = model.evaluate(test_iterator, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
Test loss: 0.506648063659668
Test accuracy: 0.7847950458526611
```

[ ]:
```
from keras.models import load_model
path = "/content/drive/MyDrive/Grad School/Penn_MUSA/Spring2022/
 ↪650_RemoteSensing/LULC_FinalProject"
model_path = '{}/models'.format(path)
model_file = '{}/custom_unet_Dice.h5'.format(model_path)
model = load_model(model_file, compile=False)
```

[ ]:
```
model.compile(loss=total_loss, metrics=["accuracy"], optimizer=Adam())
```

[ ]:
```
y_pred = model.predict(X_test)

for i in range(5):
  visualize(satImage = X_test[i],
          groundTruth=(onehot_to_rgb(y_test[i], class_dict)),
          predicted=(onehot_to_rgb(y_pred[i], class_dict)))
# visualize(satImage = cv2.cvtColor(X_test[0], cv2.COLOR_BGR2RGB))
```

```
Output hidden; open in https://colab.research.google.com to view.
```

[ ]:
```
!apt-get install texlive texlive-xetex texlive-latex-extra pandoc
!pip install pypandoc

!jupyter nbconvert --to PDF '/content/drive/MyDrive/Grad School/Penn_MUSA/
 ↪Spring2022/650_RemoteSensing/LULC_FinalProject/landCov_semSeg.ipynb'
```

```
Reading package lists… Done
Building dependency tree
Reading state information… Done
pandoc is already the newest version (1.19.2.4~dfsg-1build4).
pandoc set to manually installed.
The following packages were automatically installed and are no longer required:
  libnvidia-common-460 nsight-compute-2020.2.0
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-texgyre
  javascript-common libcupsfilters1 libcupsimage2 libgs9 libgs9-common
  libijs-0.35 libjbig2dec0 libjs-jquery libkpathsea6 libpotrace0 libptexenc1
  libruby2.5 libsynctex1 libtexlua52 libtexluajit2 libzzip-0-13 lmodern
  poppler-data preview-latex-style rake ruby ruby-did-you-mean ruby-minitest
  ruby-net-telnet ruby-power-assert ruby-test-unit ruby2.5
```

```
  rubygems-integration t1utils tex-common tex-gyre texlive-base
  texlive-binaries texlive-fonts-recommended texlive-latex-base
  texlive-latex-recommended texlive-pictures texlive-plain-generic tipa
Suggested packages:
  fonts-noto apache2 | lighttpd | httpd poppler-utils ghostscript
  fonts-japanese-mincho | fonts-ipafont-mincho fonts-japanese-gothic
  | fonts-ipafont-gothic fonts-arphic-ukai fonts-arphic-uming fonts-nanum ri
  ruby-dev bundler debhelper gv | postscript-viewer perl-tk xpdf-reader
  | pdf-viewer texlive-fonts-recommended-doc texlive-latex-base-doc
  python-pygments icc-profiles libfile-which-perl
  libspreadsheet-parseexcel-perl texlive-latex-extra-doc
  texlive-latex-recommended-doc texlive-pstricks dot2tex prerex ruby-tcltk
  | libtcltk-ruby texlive-pictures-doc vprerex
The following NEW packages will be installed:
  fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-texgyre
  javascript-common libcupsfilters1 libcupsimage2 libgs9 libgs9-common
  libijs-0.35 libjbig2dec0 libjs-jquery libkpathsea6 libpotrace0 libptexenc1
  libruby2.5 libsynctex1 libtexlua52 libtexluajit2 libzzip-0-13 lmodern
  poppler-data preview-latex-style rake ruby ruby-did-you-mean ruby-minitest
  ruby-net-telnet ruby-power-assert ruby-test-unit ruby2.5
  rubygems-integration t1utils tex-common tex-gyre texlive texlive-base
  texlive-binaries texlive-fonts-recommended texlive-latex-base
  texlive-latex-extra texlive-latex-recommended texlive-pictures
  texlive-plain-generic texlive-xetex tipa
0 upgraded, 47 newly installed, 0 to remove and 42 not upgraded.
Need to get 146 MB of archives.
After this operation, 460 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-droid-fallback
all 1:6.0.1r16-1.1 [1,805 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-lato all 2.0-2
[2,698 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic/main amd64 poppler-data all
0.4.8-2 [1,479 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic/main amd64 tex-common all 6.09
[33.0 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-lmodern all
2.004.5-3 [4,551 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-noto-mono all
20171026-2 [75.5 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/universe amd64 fonts-texgyre all
20160520-1 [8,761 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic/main amd64 javascript-common all
11 [6,066 B]
Get:9 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libcupsfilters1
amd64 1.20.2-0ubuntu3.1 [108 kB]
Get:10 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libcupsimage2
amd64 2.2.7-1ubuntu2.8 [18.6 kB]
Get:11 http://archive.ubuntu.com/ubuntu bionic/main amd64 libijs-0.35 amd64
```

```
0.35-13 [15.5 kB]
Get:12 http://archive.ubuntu.com/ubuntu bionic/main amd64 libjbig2dec0 amd64
0.13-6 [55.9 kB]
Get:13 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libgs9-common
all 9.26~dfsg+0-0ubuntu0.18.04.16 [5,093 kB]
Get:14 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libgs9 amd64
9.26~dfsg+0-0ubuntu0.18.04.16 [2,265 kB]
Get:15 http://archive.ubuntu.com/ubuntu bionic/main amd64 libjs-jquery all
3.2.1-1 [152 kB]
Get:16 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libkpathsea6
amd64 2017.20170613.44572-8ubuntu0.1 [54.9 kB]
Get:17 http://archive.ubuntu.com/ubuntu bionic/main amd64 libpotrace0 amd64
1.14-2 [17.4 kB]
Get:18 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libptexenc1
amd64 2017.20170613.44572-8ubuntu0.1 [34.5 kB]
Get:19 http://archive.ubuntu.com/ubuntu bionic/main amd64 rubygems-integration
all 1.11 [4,994 B]
Get:20 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 ruby2.5 amd64
2.5.1-1ubuntu1.11 [48.6 kB]
Get:21 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby amd64 1:2.5.1
[5,712 B]
Get:22 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 rake all
12.3.1-1ubuntu0.1 [44.9 kB]
Get:23 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-did-you-mean all
1.2.0-2 [9,700 B]
Get:24 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-minitest all
5.10.3-1 [38.6 kB]
Get:25 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-net-telnet all
0.1.1-2 [12.6 kB]
Get:26 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-power-assert all
0.3.0-1 [7,952 B]
Get:27 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-test-unit all
3.2.5-1 [61.1 kB]
Get:28 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libruby2.5
amd64 2.5.1-1ubuntu1.11 [3,072 kB]
Get:29 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libsynctex1
amd64 2017.20170613.44572-8ubuntu0.1 [41.4 kB]
Get:30 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libtexlua52
amd64 2017.20170613.44572-8ubuntu0.1 [91.2 kB]
Get:31 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libtexluajit2
amd64 2017.20170613.44572-8ubuntu0.1 [230 kB]
Get:32 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libzzip-0-13
amd64 0.13.62-3.1ubuntu0.18.04.1 [26.0 kB]
Get:33 http://archive.ubuntu.com/ubuntu bionic/main amd64 lmodern all 2.004.5-3
[9,631 kB]
Get:34 http://archive.ubuntu.com/ubuntu bionic/main amd64 preview-latex-style
all 11.91-1ubuntu1 [185 kB]
Get:35 http://archive.ubuntu.com/ubuntu bionic/main amd64 t1utils amd64 1.41-2
```

```
[56.0 kB]
Get:36 http://archive.ubuntu.com/ubuntu bionic/universe amd64 tex-gyre all
20160520-1 [4,998 kB]
Get:37 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 texlive-
binaries amd64 2017.20170613.44572-8ubuntu0.1 [8,179 kB]
Get:38 http://archive.ubuntu.com/ubuntu bionic/main amd64 texlive-base all
2017.20180305-1 [18.7 MB]
Get:39 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-fonts-
recommended all 2017.20180305-1 [5,262 kB]
Get:40 http://archive.ubuntu.com/ubuntu bionic/main amd64 texlive-latex-base all
2017.20180305-1 [951 kB]
Get:41 http://archive.ubuntu.com/ubuntu bionic/main amd64 texlive-latex-
recommended all 2017.20180305-1 [14.9 MB]
Get:42 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive all
2017.20180305-1 [14.4 kB]
Get:43 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-pictures
all 2017.20180305-1 [4,026 kB]
Get:44 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-latex-
extra all 2017.20180305-2 [10.6 MB]
Get:45 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-plain-
generic all 2017.20180305-2 [23.6 MB]
Get:46 http://archive.ubuntu.com/ubuntu bionic/universe amd64 tipa all 2:1.3-20
[2,978 kB]
Get:47 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-xetex all
2017.20180305-1 [10.7 MB]
Fetched 146 MB in 2s (83.8 MB/s)
Extracting templates from packages: 100%
Preconfiguring packages …
Selecting previously unselected package fonts-droid-fallback.
(Reading database … 155202 files and directories currently installed.)
Preparing to unpack …/00-fonts-droid-fallback_1%3a6.0.1r16-1.1_all.deb …
Unpacking fonts-droid-fallback (1:6.0.1r16-1.1) …
Selecting previously unselected package fonts-lato.
Preparing to unpack …/01-fonts-lato_2.0-2_all.deb …
Unpacking fonts-lato (2.0-2) …
Selecting previously unselected package poppler-data.
Preparing to unpack …/02-poppler-data_0.4.8-2_all.deb …
Unpacking poppler-data (0.4.8-2) …
Selecting previously unselected package tex-common.
Preparing to unpack …/03-tex-common_6.09_all.deb …
Unpacking tex-common (6.09) …
Selecting previously unselected package fonts-lmodern.
Preparing to unpack …/04-fonts-lmodern_2.004.5-3_all.deb …
Unpacking fonts-lmodern (2.004.5-3) …
Selecting previously unselected package fonts-noto-mono.
Preparing to unpack …/05-fonts-noto-mono_20171026-2_all.deb …
Unpacking fonts-noto-mono (20171026-2) …
Selecting previously unselected package fonts-texgyre.
```

```
Preparing to unpack …/06-fonts-texgyre_20160520-1_all.deb …
Unpacking fonts-texgyre (20160520-1) …
Selecting previously unselected package javascript-common.
Preparing to unpack …/07-javascript-common_11_all.deb …
Unpacking javascript-common (11) …
Selecting previously unselected package libcupsfilters1:amd64.
Preparing to unpack …/08-libcupsfilters1_1.20.2-0ubuntu3.1_amd64.deb …
Unpacking libcupsfilters1:amd64 (1.20.2-0ubuntu3.1) …
Selecting previously unselected package libcupsimage2:amd64.
Preparing to unpack …/09-libcupsimage2_2.2.7-1ubuntu2.8_amd64.deb …
Unpacking libcupsimage2:amd64 (2.2.7-1ubuntu2.8) …
Selecting previously unselected package libijs-0.35:amd64.
Preparing to unpack …/10-libijs-0.35_0.35-13_amd64.deb …
Unpacking libijs-0.35:amd64 (0.35-13) …
Selecting previously unselected package libjbig2dec0:amd64.
Preparing to unpack …/11-libjbig2dec0_0.13-6_amd64.deb …
Unpacking libjbig2dec0:amd64 (0.13-6) …
Selecting previously unselected package libgs9-common.
Preparing to unpack …/12-libgs9-common_9.26~dfsg+0-0ubuntu0.18.04.16_all.deb
…
Unpacking libgs9-common (9.26~dfsg+0-0ubuntu0.18.04.16) …
Selecting previously unselected package libgs9:amd64.
Preparing to unpack …/13-libgs9_9.26~dfsg+0-0ubuntu0.18.04.16_amd64.deb …
Unpacking libgs9:amd64 (9.26~dfsg+0-0ubuntu0.18.04.16) …
Selecting previously unselected package libjs-jquery.
Preparing to unpack …/14-libjs-jquery_3.2.1-1_all.deb …
Unpacking libjs-jquery (3.2.1-1) …
Selecting previously unselected package libkpathsea6:amd64.
Preparing to unpack …/15-libkpathsea6_2017.20170613.44572-8ubuntu0.1_amd64.deb
…
Unpacking libkpathsea6:amd64 (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package libpotrace0.
Preparing to unpack …/16-libpotrace0_1.14-2_amd64.deb …
Unpacking libpotrace0 (1.14-2) …
Selecting previously unselected package libptexenc1:amd64.
Preparing to unpack …/17-libptexenc1_2017.20170613.44572-8ubuntu0.1_amd64.deb
…
Unpacking libptexenc1:amd64 (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package rubygems-integration.
Preparing to unpack …/18-rubygems-integration_1.11_all.deb …
Unpacking rubygems-integration (1.11) …
Selecting previously unselected package ruby2.5.
Preparing to unpack …/19-ruby2.5_2.5.1-1ubuntu1.11_amd64.deb …
Unpacking ruby2.5 (2.5.1-1ubuntu1.11) …
Selecting previously unselected package ruby.
Preparing to unpack …/20-ruby_1%3a2.5.1_amd64.deb …
Unpacking ruby (1:2.5.1) …
Selecting previously unselected package rake.
```

```
Preparing to unpack …/21-rake_12.3.1-1ubuntu0.1_all.deb …
Unpacking rake (12.3.1-1ubuntu0.1) …
Selecting previously unselected package ruby-did-you-mean.
Preparing to unpack …/22-ruby-did-you-mean_1.2.0-2_all.deb …
Unpacking ruby-did-you-mean (1.2.0-2) …
Selecting previously unselected package ruby-minitest.
Preparing to unpack …/23-ruby-minitest_5.10.3-1_all.deb …
Unpacking ruby-minitest (5.10.3-1) …
Selecting previously unselected package ruby-net-telnet.
Preparing to unpack …/24-ruby-net-telnet_0.1.1-2_all.deb …
Unpacking ruby-net-telnet (0.1.1-2) …
Selecting previously unselected package ruby-power-assert.
Preparing to unpack …/25-ruby-power-assert_0.3.0-1_all.deb …
Unpacking ruby-power-assert (0.3.0-1) …
Selecting previously unselected package ruby-test-unit.
Preparing to unpack …/26-ruby-test-unit_3.2.5-1_all.deb …
Unpacking ruby-test-unit (3.2.5-1) …
Selecting previously unselected package libruby2.5:amd64.
Preparing to unpack …/27-libruby2.5_2.5.1-1ubuntu1.11_amd64.deb …
Unpacking libruby2.5:amd64 (2.5.1-1ubuntu1.11) …
Selecting previously unselected package libsynctex1:amd64.
Preparing to unpack …/28-libsynctex1_2017.20170613.44572-8ubuntu0.1_amd64.deb
…
Unpacking libsynctex1:amd64 (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package libtexlua52:amd64.
Preparing to unpack …/29-libtexlua52_2017.20170613.44572-8ubuntu0.1_amd64.deb
…
Unpacking libtexlua52:amd64 (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package libtexluajit2:amd64.
Preparing to unpack
…/30-libtexluajit2_2017.20170613.44572-8ubuntu0.1_amd64.deb …
Unpacking libtexluajit2:amd64 (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package libzzip-0-13:amd64.
Preparing to unpack …/31-libzzip-0-13_0.13.62-3.1ubuntu0.18.04.1_amd64.deb …
Unpacking libzzip-0-13:amd64 (0.13.62-3.1ubuntu0.18.04.1) …
Selecting previously unselected package lmodern.
Preparing to unpack …/32-lmodern_2.004.5-3_all.deb …
Unpacking lmodern (2.004.5-3) …
Selecting previously unselected package preview-latex-style.
Preparing to unpack …/33-preview-latex-style_11.91-1ubuntu1_all.deb …
Unpacking preview-latex-style (11.91-1ubuntu1) …
Selecting previously unselected package t1utils.
Preparing to unpack …/34-t1utils_1.41-2_amd64.deb …
Unpacking t1utils (1.41-2) …
Selecting previously unselected package tex-gyre.
Preparing to unpack …/35-tex-gyre_20160520-1_all.deb …
Unpacking tex-gyre (20160520-1) …
Selecting previously unselected package texlive-binaries.
```

```
Preparing to unpack …/36-texlive-
binaries_2017.20170613.44572-8ubuntu0.1_amd64.deb …
Unpacking texlive-binaries (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package texlive-base.
Preparing to unpack …/37-texlive-base_2017.20180305-1_all.deb …
Unpacking texlive-base (2017.20180305-1) …
Selecting previously unselected package texlive-fonts-recommended.
Preparing to unpack …/38-texlive-fonts-recommended_2017.20180305-1_all.deb …
Unpacking texlive-fonts-recommended (2017.20180305-1) …
Selecting previously unselected package texlive-latex-base.
Preparing to unpack …/39-texlive-latex-base_2017.20180305-1_all.deb …
Unpacking texlive-latex-base (2017.20180305-1) …
Selecting previously unselected package texlive-latex-recommended.
Preparing to unpack …/40-texlive-latex-recommended_2017.20180305-1_all.deb …
Unpacking texlive-latex-recommended (2017.20180305-1) …
Selecting previously unselected package texlive.
Preparing to unpack …/41-texlive_2017.20180305-1_all.deb …
Unpacking texlive (2017.20180305-1) …
Selecting previously unselected package texlive-pictures.
Preparing to unpack …/42-texlive-pictures_2017.20180305-1_all.deb …
Unpacking texlive-pictures (2017.20180305-1) …
Selecting previously unselected package texlive-latex-extra.
Preparing to unpack …/43-texlive-latex-extra_2017.20180305-2_all.deb …
Unpacking texlive-latex-extra (2017.20180305-2) …
Selecting previously unselected package texlive-plain-generic.
Preparing to unpack …/44-texlive-plain-generic_2017.20180305-2_all.deb …
Unpacking texlive-plain-generic (2017.20180305-2) …
Selecting previously unselected package tipa.
Preparing to unpack …/45-tipa_2%3a1.3-20_all.deb …
Unpacking tipa (2:1.3-20) …
Selecting previously unselected package texlive-xetex.
Preparing to unpack …/46-texlive-xetex_2017.20180305-1_all.deb …
Unpacking texlive-xetex (2017.20180305-1) …
Setting up libgs9-common (9.26~dfsg+0-0ubuntu0.18.04.16) …
Setting up libkpathsea6:amd64 (2017.20170613.44572-8ubuntu0.1) …
Setting up libjs-jquery (3.2.1-1) …
Setting up libtexlua52:amd64 (2017.20170613.44572-8ubuntu0.1) …
Setting up fonts-droid-fallback (1:6.0.1r16-1.1) …
Setting up libsynctex1:amd64 (2017.20170613.44572-8ubuntu0.1) …
Setting up libptexenc1:amd64 (2017.20170613.44572-8ubuntu0.1) …
Setting up tex-common (6.09) …
update-language: texlive-base not installed and configured, doing nothing!
Setting up poppler-data (0.4.8-2) …
Setting up tex-gyre (20160520-1) …
Setting up preview-latex-style (11.91-1ubuntu1) …
Setting up fonts-texgyre (20160520-1) …
Setting up fonts-noto-mono (20171026-2) …
Setting up fonts-lato (2.0-2) …
```

```
Setting up libcupsfilters1:amd64 (1.20.2-0ubuntu3.1) …
Setting up libcupsimage2:amd64 (2.2.7-1ubuntu2.8) …
Setting up libjbig2dec0:amd64 (0.13-6) …
Setting up ruby-did-you-mean (1.2.0-2) …
Setting up t1utils (1.41-2) …
Setting up ruby-net-telnet (0.1.1-2) …
Setting up libijs-0.35:amd64 (0.35-13) …
Setting up rubygems-integration (1.11) …
Setting up libpotrace0 (1.14-2) …
Setting up javascript-common (11) …
Setting up ruby-minitest (5.10.3-1) …
Setting up libzzip-0-13:amd64 (0.13.62-3.1ubuntu0.18.04.1) …
Setting up libgs9:amd64 (9.26~dfsg+0-0ubuntu0.18.04.16) …
Setting up libtexluajit2:amd64 (2017.20170613.44572-8ubuntu0.1) …
Setting up fonts-lmodern (2.004.5-3) …
Setting up ruby-power-assert (0.3.0-1) …
Setting up texlive-binaries (2017.20170613.44572-8ubuntu0.1) …
update-alternatives: using /usr/bin/xdvi-xaw to provide /usr/bin/xdvi.bin
(xdvi.bin) in auto mode
update-alternatives: using /usr/bin/bibtex.original to provide /usr/bin/bibtex
(bibtex) in auto mode
Setting up texlive-base (2017.20180305-1) …
mktexlsr: Updating /var/lib/texmf/ls-R-TEXLIVEDIST…
mktexlsr: Updating /var/lib/texmf/ls-R-TEXMFMAIN…
mktexlsr: Updating /var/lib/texmf/ls-R…
mktexlsr: Done.
tl-paper: setting paper size for dvips to a4:
/var/lib/texmf/dvips/config/config-paper.ps
tl-paper: setting paper size for dvipdfmx to a4:
/var/lib/texmf/dvipdfmx/dvipdfmx-paper.cfg
tl-paper: setting paper size for xdvi to a4: /var/lib/texmf/xdvi/XDvi-paper
tl-paper: setting paper size for pdftex to a4:
/var/lib/texmf/tex/generic/config/pdftexconfig.tex
Setting up texlive-fonts-recommended (2017.20180305-1) …
Setting up texlive-plain-generic (2017.20180305-2) …
Setting up texlive-latex-base (2017.20180305-1) …
Setting up lmodern (2.004.5-3) …
Setting up texlive-latex-recommended (2017.20180305-1) …
Setting up texlive-pictures (2017.20180305-1) …
Setting up tipa (2:1.3-20) …
Regenerating '/var/lib/texmf/fmtutil.cnf-DEBIAN'… done.
Regenerating '/var/lib/texmf/fmtutil.cnf-TEXLIVEDIST'… done.
update-fmtutil has updated the following file(s):
        /var/lib/texmf/fmtutil.cnf-DEBIAN
        /var/lib/texmf/fmtutil.cnf-TEXLIVEDIST
If you want to activate the changes in the above file(s),
you should run fmtutil-sys or fmtutil.
Setting up texlive (2017.20180305-1) …
```

```
Setting up texlive-latex-extra (2017.20180305-2) …
Setting up texlive-xetex (2017.20180305-1) …
Setting up ruby2.5 (2.5.1-1ubuntu1.11) …
Setting up ruby (1:2.5.1) …
Setting up ruby-test-unit (3.2.5-1) …
Setting up rake (12.3.1-1ubuntu0.1) …
Setting up libruby2.5:amd64 (2.5.1-1ubuntu1.11) …
Processing triggers for mime-support (3.60ubuntu1) …
Processing triggers for libc-bin (2.27-3ubuntu1.3) …
/sbin/ldconfig.real: /usr/local/lib/python3.7/dist-
packages/ideep4py/lib/libmkldnn.so.0 is not a symbolic link

Processing triggers for man-db (2.8.3-2ubuntu0.1) …
Processing triggers for fontconfig (2.12.6-0ubuntu2) …
Processing triggers for tex-common (6.09) …
Running updmap-sys. This may take some time… done.
Running mktexlsr /var/lib/texmf … done.
```