In [ ]:

**import** glob  
**import** matplotlib.pyplot **as** plt  
**import** numpy **as** np  
**import** tifffile **as** tiff  
**import** cv2  
**from** PIL **import** Image  
**import** os  
  
**import** tensorflow **as** tf  
**from** tensorflow.keras.models **import** **\***  
**from** tensorflow.keras.layers **import** **\***  
**from** tensorflow.keras **import** backend **as** K  
**from** tensorflow.keras.optimizers **import** **\***  
**from** tensorflow.keras.callbacks **import** EarlyStopping, ModelCheckpoint, Callback  
**from** tensorflow.keras.metrics **import** MeanIoU, Accuracy

In [ ]:

**from** google.colab **import** drive  
drive**.**mount('/content/drive')

Mounted at /content/drive

In [ ]:

**from** keras **import** backend **as** K  
**def** adjustData(img, mask):  
 onehot**=**mask**/**255 *###pixel normalization*  
 **return**(img, onehot)  
  
**def** TrainGenerator(aug\_dict, image\_folder, mask\_folder, batch\_size, save\_to\_dir, seed **=** 1, image1\_save\_prefix **=** "image", mask\_save\_prefix **=** "mask"):  
 image\_datagen **=** ImageDataGenerator(**\*\***aug\_dict)  
 mask\_datagen **=** ImageDataGenerator(**\*\***aug\_dict)  
  
 image\_generator **=** image\_datagen**.**flow\_from\_directory(  
 directory **=** image\_folder,  
 target\_size **=** (512, 512),  
 color\_mode **=** 'rgb',  
 class\_mode **=** **None**,  
 batch\_size **=** batch\_size,  
 save\_to\_dir **=** save\_to\_dir,  
 save\_prefix **=** image1\_save\_prefix,  
 shuffle **=** **True**,  
 seed **=** seed  
 )  
 mask\_generator **=** mask\_datagen**.**flow\_from\_directory(  
 directory **=** mask\_folder,  
 target\_size **=** (512, 512),  
 color\_mode **=** 'grayscale',  
 class\_mode **=** **None**,  
 batch\_size **=** batch\_size,  
 save\_to\_dir **=** save\_to\_dir,  
 save\_prefix **=** mask\_save\_prefix,  
 shuffle **=** **True**,  
 seed **=** seed  
 )  
 train\_generator **=** zip(image\_generator, mask\_generator)  
 **for** (img, mask) **in** train\_generator:  
 X**=**[]  
 Y**=**[]  
 **for** d **in** range(img**.**shape[0]):  
 x, y **=** adjustData(img[d], mask[d])  
 X**.**append(x)  
 Y**.**append(y)  
 X **=** np**.**reshape(np**.**array(X),(img**.**shape[0],512, 512,3))  
 Y **=** np**.**reshape(np**.**array(Y),(img**.**shape[0],512, 512,1))  
 **yield** X, Y  
  
**def** ValGenerator(aug\_dict, image\_folder, mask\_folder, batch\_size, save\_to\_dir, seed **=** 1, image\_save\_prefix **=** "image", mask\_save\_prefix **=** "mask"):  
 image\_datagen **=** ImageDataGenerator(**\*\***aug\_dict)  
 mask\_datagen **=** ImageDataGenerator(**\*\***aug\_dict)  
  
 image\_generator **=** image\_datagen**.**flow\_from\_directory(  
 directory **=** image\_folder,  
 target\_size **=** (512, 512),  
 color\_mode **=** 'rgb',  
 class\_mode **=** **None**,  
 batch\_size **=** batch\_size,  
 save\_to\_dir **=** save\_to\_dir,  
 save\_prefix **=** image\_save\_prefix,  
 shuffle **=** **True**,  
 seed **=** seed  
 )  
 mask\_generator **=** mask\_datagen**.**flow\_from\_directory(  
 directory **=** mask\_folder,  
 target\_size **=** (512, 512),  
 color\_mode **=** 'grayscale',  
 class\_mode **=** **None**,  
 batch\_size **=** batch\_size,  
 save\_to\_dir **=** save\_to\_dir,  
 save\_prefix **=** mask\_save\_prefix,  
 shuffle **=** **True**,  
 seed **=** seed  
 )  
  
 valid\_generator **=** zip(image\_generator, mask\_generator)  
 **for** (img, mask) **in** valid\_generator:  
 X**=**[]  
 Y**=**[]  
 **for** d **in** range(img**.**shape[0]):  
 x, y **=** adjustData(img[d], mask[d])  
 X**.**append(x)  
 Y**.**append(y)  
 X **=** np**.**reshape(np**.**array(X),(img**.**shape[0],512, 512,3))  
 Y **=** np**.**reshape(np**.**array(Y),(img**.**shape[0],512, 512,1))  
 **yield** X, Y

In [ ]:

**from** tensorflow.keras.preprocessing.image **import** ImageDataGenerator  
data\_gen\_args **=** dict()  
Generator **=** TrainGenerator(aug\_dict**=**data\_gen\_args,image\_folder**=**'/content/drive/Shareddrives/Untitled shared drive/Nitk Final Project/Train/I/',  
 mask\_folder**=**'/content/drive/Shareddrives/Untitled shared drive/Nitk Final Project/Train/G',batch\_size**=**4,save\_to\_dir**=None**)  
Val **=** ValGenerator(aug\_dict**=**data\_gen\_args,image\_folder**=**'/content/drive/Shareddrives/Untitled shared drive/Nitk Final Project/Val/I',  
 mask\_folder**=**'/content/drive/Shareddrives/Untitled shared drive/Nitk Final Project/Val/G',batch\_size**=**4,save\_to\_dir**=None**)

In [ ]:

IMG\_WIDTH **=** 512  
IMG\_HEIGHT **=** 512  
IMG\_CHANNELS **=** 3  
  
inputs **=** Input((IMG\_HEIGHT, IMG\_WIDTH, IMG\_CHANNELS))  
  
*# Encode module*  
c1 **=** Conv2D(16, (3, 3), activation**=**'relu', kernel\_initializer**=**'he\_normal', padding**=**'same') (inputs)  
c1 **=** Conv2D(16, (3, 3), activation**=**'relu', kernel\_initializer**=**'he\_normal', padding**=**'same') (c1)  
p1 **=** MaxPooling2D((2, 2)) (c1)  
  
c2 **=** Conv2D(32, (3, 3), activation**=**'relu', kernel\_initializer**=**'he\_normal', padding**=**'same') (p1)  
c2 **=** Conv2D(32, (3, 3), activation**=**'relu', kernel\_initializer**=**'he\_normal', padding**=**'same') (c2)  
p2 **=** MaxPooling2D((2, 2)) (c2)  
  
c3 **=** Conv2D(64, (3, 3), activation**=**'relu', kernel\_initializer**=**'he\_normal', padding**=**'same') (p2)  
c3 **=** Conv2D(64, (3, 3), activation**=**'relu', kernel\_initializer**=**'he\_normal', padding**=**'same') (c3)  
p3 **=** MaxPooling2D((2, 2)) (c3)  
  
c4 **=** Conv2D(128, (3, 3), activation**=**'relu', kernel\_initializer**=**'he\_normal', padding**=**'same') (p3)  
c4 **=** Dropout(0.2) (c4)  
c4 **=** Conv2D(128, (3, 3), activation**=**'relu', kernel\_initializer**=**'he\_normal', padding**=**'same') (c4)  
p4 **=** MaxPooling2D(pool\_size**=**(2, 2)) (c4)  
  
c5 **=** Conv2D(256, (3, 3), activation**=**'relu', kernel\_initializer**=**'he\_normal', padding**=**'same') (p4)  
c5 **=** Dropout(0.4) (c5)  
c5 **=** Conv2D(256, (3, 3), activation**=**'relu', kernel\_initializer**=**'he\_normal', padding**=**'same') (c5)  
  
*# Decoder module*  
u6 **=** Conv2DTranspose(128, (2, 2), strides**=**(2, 2), padding**=**'same') (c5)  
u6 **=** concatenate([u6, c4])  
c6 **=** Conv2D(128, (3, 3), activation**=**'relu', kernel\_initializer**=**'he\_normal', padding**=**'same') (u6)  
c6 **=** Dropout(0.2) (c6)  
c6 **=** Conv2D(128, (3, 3), activation**=**'relu', kernel\_initializer**=**'he\_normal', padding**=**'same') (c6)  
  
u7 **=** Conv2DTranspose(64, (2, 2), strides**=**(2, 2), padding**=**'same') (c6)  
u7 **=** concatenate([u7, c3])  
c7 **=** Conv2D(64, (3, 3), activation**=**'relu', kernel\_initializer**=**'he\_normal', padding**=**'same') (u7)  
c7 **=** Conv2D(64, (3, 3), activation**=**'relu', kernel\_initializer**=**'he\_normal', padding**=**'same') (c7)  
  
u8 **=** Conv2DTranspose(32, (2, 2), strides**=**(2, 2), padding**=**'same') (c7)  
u8 **=** concatenate([u8, c2])  
c8 **=** Conv2D(32, (3, 3), activation**=**'relu', kernel\_initializer**=**'he\_normal', padding**=**'same') (u8)  
c8 **=** Conv2D(32, (3, 3), activation**=**'relu', kernel\_initializer**=**'he\_normal', padding**=**'same') (c8)  
  
u9 **=** Conv2DTranspose(16, (2, 2), strides**=**(2, 2), padding**=**'same') (c8)  
u9 **=** concatenate([u9, c1], axis**=**3)  
c9 **=** Conv2D(16, (3, 3), activation**=**'relu', kernel\_initializer**=**'he\_normal', padding**=**'same') (u9)  
c9 **=** Conv2D(16, (3, 3), activation**=**'relu', kernel\_initializer**=**'he\_normal', padding**=**'same') (c9)  
outputs **=** Conv2D(1, (1, 1), activation**=**'sigmoid') (c9)  
  
model **=** Model(inputs**=**[inputs], outputs**=**[outputs])

In [ ]:

*# define loss function*  
opt **=** Adam(learning\_rate**=**0.0001)  
model**.**compile(optimizer**=**opt, loss**=**'binary\_crossentropy', metrics**=**['accuracy'])  
print(model**.**summary())

**Model: "functional"**

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┓  
┃ **Layer (type)** ┃ **Output Shape** ┃ **Param #** ┃ **Connected to** ┃  
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━┩  
│ input\_layer (InputLayer) │ (None, 512, 512, 3) │ 0 │ - │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d (Conv2D) │ (None, 512, 512, 16) │ 448 │ input\_layer[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_1 (Conv2D) │ (None, 512, 512, 16) │ 2,320 │ conv2d[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ max\_pooling2d │ (None, 256, 256, 16) │ 0 │ conv2d\_1[0][0] │  
│ (MaxPooling2D) │ │ │ │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_2 (Conv2D) │ (None, 256, 256, 32) │ 4,640 │ max\_pooling2d[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_3 (Conv2D) │ (None, 256, 256, 32) │ 9,248 │ conv2d\_2[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ max\_pooling2d\_1 │ (None, 128, 128, 32) │ 0 │ conv2d\_3[0][0] │  
│ (MaxPooling2D) │ │ │ │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_4 (Conv2D) │ (None, 128, 128, 64) │ 18,496 │ max\_pooling2d\_1[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_5 (Conv2D) │ (None, 128, 128, 64) │ 36,928 │ conv2d\_4[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ max\_pooling2d\_2 │ (None, 64, 64, 64) │ 0 │ conv2d\_5[0][0] │  
│ (MaxPooling2D) │ │ │ │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_6 (Conv2D) │ (None, 64, 64, 128) │ 73,856 │ max\_pooling2d\_2[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ dropout (Dropout) │ (None, 64, 64, 128) │ 0 │ conv2d\_6[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_7 (Conv2D) │ (None, 64, 64, 128) │ 147,584 │ dropout[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ max\_pooling2d\_3 │ (None, 32, 32, 128) │ 0 │ conv2d\_7[0][0] │  
│ (MaxPooling2D) │ │ │ │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_8 (Conv2D) │ (None, 32, 32, 256) │ 295,168 │ max\_pooling2d\_3[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ dropout\_1 (Dropout) │ (None, 32, 32, 256) │ 0 │ conv2d\_8[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_9 (Conv2D) │ (None, 32, 32, 256) │ 590,080 │ dropout\_1[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_transpose │ (None, 64, 64, 128) │ 131,200 │ conv2d\_9[0][0] │  
│ (Conv2DTranspose) │ │ │ │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ concatenate (Concatenate) │ (None, 64, 64, 256) │ 0 │ conv2d\_transpose[0][0… │  
│ │ │ │ conv2d\_7[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_10 (Conv2D) │ (None, 64, 64, 128) │ 295,040 │ concatenate[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ dropout\_2 (Dropout) │ (None, 64, 64, 128) │ 0 │ conv2d\_10[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_11 (Conv2D) │ (None, 64, 64, 128) │ 147,584 │ dropout\_2[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_transpose\_1 │ (None, 128, 128, 64) │ 32,832 │ conv2d\_11[0][0] │  
│ (Conv2DTranspose) │ │ │ │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ concatenate\_1 │ (None, 128, 128, 128) │ 0 │ conv2d\_transpose\_1[0]… │  
│ (Concatenate) │ │ │ conv2d\_5[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_12 (Conv2D) │ (None, 128, 128, 64) │ 73,792 │ concatenate\_1[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_13 (Conv2D) │ (None, 128, 128, 64) │ 36,928 │ conv2d\_12[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_transpose\_2 │ (None, 256, 256, 32) │ 8,224 │ conv2d\_13[0][0] │  
│ (Conv2DTranspose) │ │ │ │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ concatenate\_2 │ (None, 256, 256, 64) │ 0 │ conv2d\_transpose\_2[0]… │  
│ (Concatenate) │ │ │ conv2d\_3[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_14 (Conv2D) │ (None, 256, 256, 32) │ 18,464 │ concatenate\_2[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_15 (Conv2D) │ (None, 256, 256, 32) │ 9,248 │ conv2d\_14[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_transpose\_3 │ (None, 512, 512, 16) │ 2,064 │ conv2d\_15[0][0] │  
│ (Conv2DTranspose) │ │ │ │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ concatenate\_3 │ (None, 512, 512, 32) │ 0 │ conv2d\_transpose\_3[0]… │  
│ (Concatenate) │ │ │ conv2d\_1[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_16 (Conv2D) │ (None, 512, 512, 16) │ 4,624 │ concatenate\_3[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_17 (Conv2D) │ (None, 512, 512, 16) │ 2,320 │ conv2d\_16[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_18 (Conv2D) │ (None, 512, 512, 1) │ 17 │ conv2d\_17[0][0] │  
└───────────────────────────┴────────────────────────┴────────────────┴────────────────────────┘

**Total params:** 1,941,105 (7.40 MB)

**Trainable params:** 1,941,105 (7.40 MB)

**Non-trainable params:** 0 (0.00 B)

None

In [ ]:

**import** kagglehub  
  
*# Download latest version*  
path **=** kagglehub**.**dataset\_download("balraj98/massachusetts-buildings-dataset")  
  
print("Path to dataset files:", path)

---------------------------------------------------------------------------  
KeyboardInterrupt Traceback (most recent call last)  
<ipython-input-7-bf09de22071c> in <cell line: 0>()  
 **2**   
 **3** # Download latest version  
----> 4 path = kagglehub.dataset\_download("balraj98/massachusetts-buildings-dataset")  
 **5**   
 **6** print("Path to dataset files:", path)  
  
/usr/local/lib/python3.11/dist-packages/kagglehub/datasets.py in dataset\_download(handle, path, force\_download)  
 **38** h = parse\_dataset\_handle(handle)  
 **39** logger.info(f"Downloading Dataset: {h.to\_url()} ...", extra={\*\*EXTRA\_CONSOLE\_BLOCK})  
---> 40 return registry.dataset\_resolver(h, path, force\_download=force\_download)  
 **41**   
 **42**   
  
/usr/local/lib/python3.11/dist-packages/kagglehub/registry.py in \_\_call\_\_(self, \*args, \*\*kwargs)  
 **21** for impl in reversed(self.\_impls):  
 **22** if impl.is\_supported(\*args, \*\*kwargs):  
---> 23 return impl(\*args, \*\*kwargs)  
 **24** else:  
 **25** fails.append(type(impl).\_\_name\_\_)  
  
/usr/local/lib/python3.11/dist-packages/kagglehub/http\_resolver.py in \_\_call\_\_(self, h, path, force\_download)  
 **126**   
 **127** # First, we download the archive.  
--> 128 api\_client.download\_file(url\_path, archive\_path, h)  
 **129**   
 **130** # Create the directory to extract the archive to.  
  
/usr/local/lib/python3.11/dist-packages/kagglehub/clients.py in download\_file(self, path, out\_file, resource\_handle, cached\_path, extract\_auto\_compressed\_file)  
 **151** """  
 **152** url = self.\_build\_url(path)  
--> 153 with requests.get(  
 **154** url,  
 **155** headers={"User-Agent": get\_user\_agent()},  
  
/usr/local/lib/python3.11/dist-packages/requests/api.py in get(url, params, \*\*kwargs)  
 **71** """  
 **72**   
---> 73 return request("get", url, params=params, \*\*kwargs)  
 **74**   
 **75**   
  
/usr/local/lib/python3.11/dist-packages/requests/api.py in request(method, url, \*\*kwargs)  
 **57** # cases, and look like a memory leak in others.  
 **58** with sessions.Session() as session:  
---> 59 return session.request(method=method, url=url, \*\*kwargs)  
 **60**   
 **61**   
  
/usr/local/lib/python3.11/dist-packages/requests/sessions.py in request(self, method, url, params, data, headers, cookies, files, auth, timeout, allow\_redirects, proxies, hooks, stream, verify, cert, json)  
 **587** }  
 **588** send\_kwargs.update(settings)  
--> 589 resp = self.send(prep, \*\*send\_kwargs)  
 **590**   
 **591** return resp  
  
/usr/local/lib/python3.11/dist-packages/requests/sessions.py in send(self, request, \*\*kwargs)  
 **722** # Redirect resolving generator.  
 **723** gen = self.resolve\_redirects(r, request, \*\*kwargs)  
--> 724 history = [resp for resp in gen]  
 **725** else:  
 **726** history = []  
  
/usr/local/lib/python3.11/dist-packages/requests/sessions.py in <listcomp>(.0)  
 **722** # Redirect resolving generator.  
 **723** gen = self.resolve\_redirects(r, request, \*\*kwargs)  
--> 724 history = [resp for resp in gen]  
 **725** else:  
 **726** history = []  
  
/usr/local/lib/python3.11/dist-packages/requests/sessions.py in resolve\_redirects(self, resp, req, stream, timeout, verify, cert, proxies, yield\_requests, \*\*adapter\_kwargs)  
 **263** yield req  
 **264** else:  
--> 265 resp = self.send(  
 **266** req,  
 **267** stream=stream,  
  
/usr/local/lib/python3.11/dist-packages/requests/sessions.py in send(self, request, \*\*kwargs)  
 **701**   
 **702** # Send the request  
--> 703 r = adapter.send(request, \*\*kwargs)  
 **704**   
 **705** # Total elapsed time of the request (approximately)  
  
/usr/local/lib/python3.11/dist-packages/requests/adapters.py in send(self, request, stream, timeout, verify, cert, proxies)  
 **665**   
 **666** try:  
--> 667 resp = conn.urlopen(  
 **668** method=request.method,  
 **669** url=url,  
  
/usr/local/lib/python3.11/dist-packages/urllib3/connectionpool.py in urlopen(self, method, url, body, headers, retries, redirect, assert\_same\_host, timeout, pool\_timeout, release\_conn, chunked, body\_pos, preload\_content, decode\_content, \*\*response\_kw)  
 **785**   
 **786** # Make the request on the HTTPConnection object  
--> 787 response = self.\_make\_request(  
 **788** conn,  
 **789** method,  
  
/usr/local/lib/python3.11/dist-packages/urllib3/connectionpool.py in \_make\_request(self, conn, method, url, body, headers, retries, timeout, chunked, response\_conn, preload\_content, decode\_content, enforce\_content\_length)  
 **532** # Receive the response from the server  
 **533** try:  
--> 534 response = conn.getresponse()  
 **535** except (BaseSSLError, OSError) as e:  
 **536** self.\_raise\_timeout(err=e, url=url, timeout\_value=read\_timeout)  
  
/usr/local/lib/python3.11/dist-packages/urllib3/connection.py in getresponse(self)  
 **514**   
 **515** # Get the response from http.client.HTTPConnection  
--> 516 httplib\_response = super().getresponse()  
 **517**   
 **518** try:  
  
/usr/lib/python3.11/http/client.py in getresponse(self)  
 **1393** try:  
 **1394** try:  
-> 1395 response.begin()  
 **1396** except ConnectionError:  
 **1397** self.close()  
  
/usr/lib/python3.11/http/client.py in begin(self)  
 **323** # read until we get a non-100 response  
 **324** while True:  
--> 325 version, status, reason = self.\_read\_status()  
 **326** if status != CONTINUE:  
 **327** break  
  
/usr/lib/python3.11/http/client.py in \_read\_status(self)  
 **284**   
 **285** def \_read\_status(self):  
--> 286 line = str(self.fp.readline(\_MAXLINE + 1), "iso-8859-1")  
 **287** if len(line) > \_MAXLINE:  
 **288** raise LineTooLong("status line")  
  
/usr/lib/python3.11/socket.py in readinto(self, b)  
 **716** while True:  
 **717** try:  
--> 718 return self.\_sock.recv\_into(b)  
 **719** except timeout:  
 **720** self.\_timeout\_occurred = True  
  
/usr/lib/python3.11/ssl.py in recv\_into(self, buffer, nbytes, flags)  
 **1312** "non-zero flags not allowed in calls to recv\_into() on %s" %  
 **1313** self.\_\_class\_\_)  
-> 1314 return self.read(nbytes, buffer)  
 **1315** else:  
 **1316** return super().recv\_into(buffer, nbytes, flags)  
  
/usr/lib/python3.11/ssl.py in read(self, len, buffer)  
 **1164** try:  
 **1165** if buffer is not None:  
-> 1166 return self.\_sslobj.read(len, buffer)  
 **1167** else:  
 **1168** return self.\_sslobj.read(len)  
  
KeyboardInterrupt:

In [ ]:

reduce\_lr **=** tf**.**keras**.**callbacks**.**ReduceLROnPlateau(monitor**=**'val\_loss',mode**=**'min',factor**=**0.5,patience**=**10,verbose**=**1,cooldown**=**1,min\_delta**=**0.0001)  
early\_stop **=** tf**.**keras**.**callbacks**.**EarlyStopping(monitor**=**'val\_loss',min\_delta**=**0.0001,patience**=**20,verbose**=**1,mode**=**'min',restore\_best\_weights**=True**)

In [ ]:

check\_path **=** '/content/drive/MyDrive/FDP\_Lab/MultiSeg\_Python/DL\_Building/UNet\_Building.weights.h5'  
checkpoint **=** tf**.**keras**.**callbacks**.**ModelCheckpoint(check\_path,monitor**=**'val\_loss',verbose**=**1,save\_best\_only**=True**,save\_weights\_only**=True**,mode**=**'min')  
  
history **=** model**.**fit(Generator,validation\_data **=**Val, validation\_steps**=** 9,steps\_per\_epoch**=**308,shuffle**=True**,verbose**=**1,epochs**=**10,callbacks**=**[reduce\_lr,early\_stop,checkpoint])

Found 1233 images belonging to 1 classes.  
Found 1233 images belonging to 1 classes.  
Epoch 1/10

/usr/local/lib/python3.11/dist-packages/keras/src/models/functional.py:237: UserWarning: The structure of `inputs` doesn't match the expected structure.  
Expected: ['keras\_tensor']  
Received: inputs=Tensor(shape=(None, 512, 512, 3))  
 warnings.warn(msg)

**308/308** ━━━━━━━━━━━━━━━━━━━━ **0s** 4s/step - accuracy: 0.7769 - loss: 4.7105Found 36 images belonging to 1 classes.  
Found 36 images belonging to 1 classes.  
  
Epoch 1: val\_loss improved from inf to 0.86465, saving model to /content/drive/MyDrive/FDP\_Lab/MultiSeg\_Python/DL\_Building/UNet\_Building.weights.h5  
**308/308** ━━━━━━━━━━━━━━━━━━━━ **1236s** 4s/step - accuracy: 0.7770 - loss: 4.7034 - val\_accuracy: 0.8384 - val\_loss: 0.8646 - learning\_rate: 1.0000e-04  
Epoch 2/10  
**308/308** ━━━━━━━━━━━━━━━━━━━━ **0s** 136ms/step - accuracy: 0.8212 - loss: 0.9062  
Epoch 2: val\_loss improved from 0.86465 to 0.55104, saving model to /content/drive/MyDrive/FDP\_Lab/MultiSeg\_Python/DL\_Building/UNet\_Building.weights.h5  
**308/308** ━━━━━━━━━━━━━━━━━━━━ **84s** 229ms/step - accuracy: 0.8213 - loss: 0.9058 - val\_accuracy: 0.8430 - val\_loss: 0.5510 - learning\_rate: 1.0000e-04  
Epoch 3/10  
**308/308** ━━━━━━━━━━━━━━━━━━━━ **0s** 135ms/step - accuracy: 0.8505 - loss: 0.5421  
Epoch 3: val\_loss improved from 0.55104 to 0.40299, saving model to /content/drive/MyDrive/FDP\_Lab/MultiSeg\_Python/DL\_Building/UNet\_Building.weights.h5  
**308/308** ━━━━━━━━━━━━━━━━━━━━ **82s** 266ms/step - accuracy: 0.8505 - loss: 0.5420 - val\_accuracy: 0.8749 - val\_loss: 0.4030 - learning\_rate: 1.0000e-04  
Epoch 4/10  
**308/308** ━━━━━━━━━━━━━━━━━━━━ **0s** 135ms/step - accuracy: 0.8609 - loss: 0.4367  
Epoch 4: val\_loss improved from 0.40299 to 0.31278, saving model to /content/drive/MyDrive/FDP\_Lab/MultiSeg\_Python/DL\_Building/UNet\_Building.weights.h5  
**308/308** ━━━━━━━━━━━━━━━━━━━━ **44s** 142ms/step - accuracy: 0.8609 - loss: 0.4366 - val\_accuracy: 0.8989 - val\_loss: 0.3128 - learning\_rate: 1.0000e-04  
Epoch 5/10  
**308/308** ━━━━━━━━━━━━━━━━━━━━ **0s** 133ms/step - accuracy: 0.8789 - loss: 0.3548  
Epoch 5: val\_loss did not improve from 0.31278  
**308/308** ━━━━━━━━━━━━━━━━━━━━ **81s** 262ms/step - accuracy: 0.8789 - loss: 0.3548 - val\_accuracy: 0.8867 - val\_loss: 0.3209 - learning\_rate: 1.0000e-04  
Epoch 6/10  
**308/308** ━━━━━━━━━━━━━━━━━━━━ **0s** 136ms/step - accuracy: 0.8890 - loss: 0.3097  
Epoch 6: val\_loss improved from 0.31278 to 0.27864, saving model to /content/drive/MyDrive/FDP\_Lab/MultiSeg\_Python/DL\_Building/UNet\_Building.weights.h5  
**308/308** ━━━━━━━━━━━━━━━━━━━━ **83s** 270ms/step - accuracy: 0.8890 - loss: 0.3097 - val\_accuracy: 0.9009 - val\_loss: 0.2786 - learning\_rate: 1.0000e-04  
Epoch 7/10  
**308/308** ━━━━━━━━━━━━━━━━━━━━ **0s** 135ms/step - accuracy: 0.8903 - loss: 0.2917  
Epoch 7: val\_loss improved from 0.27864 to 0.22940, saving model to /content/drive/MyDrive/FDP\_Lab/MultiSeg\_Python/DL\_Building/UNet\_Building.weights.h5  
**308/308** ━━━━━━━━━━━━━━━━━━━━ **82s** 266ms/step - accuracy: 0.8904 - loss: 0.2916 - val\_accuracy: 0.9124 - val\_loss: 0.2294 - learning\_rate: 1.0000e-04  
Epoch 8/10  
**308/308** ━━━━━━━━━━━━━━━━━━━━ **0s** 135ms/step - accuracy: 0.8988 - loss: 0.2654  
Epoch 8: val\_loss improved from 0.22940 to 0.21368, saving model to /content/drive/MyDrive/FDP\_Lab/MultiSeg\_Python/DL\_Building/UNet\_Building.weights.h5  
**308/308** ━━━━━━━━━━━━━━━━━━━━ **82s** 266ms/step - accuracy: 0.8988 - loss: 0.2654 - val\_accuracy: 0.9226 - val\_loss: 0.2137 - learning\_rate: 1.0000e-04  
Epoch 9/10  
**308/308** ━━━━━━━━━━━━━━━━━━━━ **0s** 135ms/step - accuracy: 0.8996 - loss: 0.2584  
Epoch 9: val\_loss improved from 0.21368 to 0.20427, saving model to /content/drive/MyDrive/FDP\_Lab/MultiSeg\_Python/DL\_Building/UNet\_Building.weights.h5  
**308/308** ━━━━━━━━━━━━━━━━━━━━ **43s** 141ms/step - accuracy: 0.8996 - loss: 0.2584 - val\_accuracy: 0.9253 - val\_loss: 0.2043 - learning\_rate: 1.0000e-04  
Epoch 10/10  
**308/308** ━━━━━━━━━━━━━━━━━━━━ **0s** 133ms/step - accuracy: 0.9047 - loss: 0.2408  
Epoch 10: val\_loss did not improve from 0.20427  
**308/308** ━━━━━━━━━━━━━━━━━━━━ **42s** 136ms/step - accuracy: 0.9047 - loss: 0.2408 - val\_accuracy: 0.9217 - val\_loss: 0.2070 - learning\_rate: 1.0000e-04  
Restoring model weights from the end of the best epoch: 9.

In [ ]:

plt**.**plot(history**.**history['accuracy'] , label **=** 'train\_acc')  
plt**.**plot(history**.**history['val\_accuracy'] ,'--', label **=** 'val\_acc')  
plt**.**legend()  
plt**.**xlabel("No. of epochs")  
plt**.**ylabel("Accuracy")  
plt**.**title("Training and Validation Accuracy")  
plt**.**show()  
  
plt**.**plot(history**.**history['loss'] , label **=** 'train\_loss')  
plt**.**plot(history**.**history['val\_loss'] ,'--', label **=** 'val\_loss')  
plt**.**legend()  
plt**.**xlabel("No. of epochs")  
plt**.**ylabel("Loss")  
plt**.**title("Training and Validation Loss")  
plt**.**show()

In [ ]:

*# Verify the file path and ensure it exists in your Google Drive.*  
*# !ls '/content/drive/MyDrive/FDP\_Lab/MultiSeg\_Python/DL\_Building/'*  
  
*# If the file exists, load the weights:*  
model**.**load\_weights('/content/drive/MyDrive/FDP\_Lab/MultiSeg\_Python/DL\_Building/UNet\_Building.weights.h5')  
print(model**.**summary())

**Model: "functional"**

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┓  
┃ **Layer (type)** ┃ **Output Shape** ┃ **Param #** ┃ **Connected to** ┃  
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━┩  
│ input\_layer (InputLayer) │ (None, 512, 512, 3) │ 0 │ - │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d (Conv2D) │ (None, 512, 512, 16) │ 448 │ input\_layer[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_1 (Conv2D) │ (None, 512, 512, 16) │ 2,320 │ conv2d[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ max\_pooling2d │ (None, 256, 256, 16) │ 0 │ conv2d\_1[0][0] │  
│ (MaxPooling2D) │ │ │ │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_2 (Conv2D) │ (None, 256, 256, 32) │ 4,640 │ max\_pooling2d[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_3 (Conv2D) │ (None, 256, 256, 32) │ 9,248 │ conv2d\_2[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ max\_pooling2d\_1 │ (None, 128, 128, 32) │ 0 │ conv2d\_3[0][0] │  
│ (MaxPooling2D) │ │ │ │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_4 (Conv2D) │ (None, 128, 128, 64) │ 18,496 │ max\_pooling2d\_1[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_5 (Conv2D) │ (None, 128, 128, 64) │ 36,928 │ conv2d\_4[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ max\_pooling2d\_2 │ (None, 64, 64, 64) │ 0 │ conv2d\_5[0][0] │  
│ (MaxPooling2D) │ │ │ │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_6 (Conv2D) │ (None, 64, 64, 128) │ 73,856 │ max\_pooling2d\_2[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ dropout (Dropout) │ (None, 64, 64, 128) │ 0 │ conv2d\_6[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_7 (Conv2D) │ (None, 64, 64, 128) │ 147,584 │ dropout[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ max\_pooling2d\_3 │ (None, 32, 32, 128) │ 0 │ conv2d\_7[0][0] │  
│ (MaxPooling2D) │ │ │ │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_8 (Conv2D) │ (None, 32, 32, 256) │ 295,168 │ max\_pooling2d\_3[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ dropout\_1 (Dropout) │ (None, 32, 32, 256) │ 0 │ conv2d\_8[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_9 (Conv2D) │ (None, 32, 32, 256) │ 590,080 │ dropout\_1[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_transpose │ (None, 64, 64, 128) │ 131,200 │ conv2d\_9[0][0] │  
│ (Conv2DTranspose) │ │ │ │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ concatenate (Concatenate) │ (None, 64, 64, 256) │ 0 │ conv2d\_transpose[0][0… │  
│ │ │ │ conv2d\_7[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_10 (Conv2D) │ (None, 64, 64, 128) │ 295,040 │ concatenate[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ dropout\_2 (Dropout) │ (None, 64, 64, 128) │ 0 │ conv2d\_10[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_11 (Conv2D) │ (None, 64, 64, 128) │ 147,584 │ dropout\_2[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_transpose\_1 │ (None, 128, 128, 64) │ 32,832 │ conv2d\_11[0][0] │  
│ (Conv2DTranspose) │ │ │ │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ concatenate\_1 │ (None, 128, 128, 128) │ 0 │ conv2d\_transpose\_1[0]… │  
│ (Concatenate) │ │ │ conv2d\_5[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_12 (Conv2D) │ (None, 128, 128, 64) │ 73,792 │ concatenate\_1[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_13 (Conv2D) │ (None, 128, 128, 64) │ 36,928 │ conv2d\_12[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_transpose\_2 │ (None, 256, 256, 32) │ 8,224 │ conv2d\_13[0][0] │  
│ (Conv2DTranspose) │ │ │ │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ concatenate\_2 │ (None, 256, 256, 64) │ 0 │ conv2d\_transpose\_2[0]… │  
│ (Concatenate) │ │ │ conv2d\_3[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_14 (Conv2D) │ (None, 256, 256, 32) │ 18,464 │ concatenate\_2[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_15 (Conv2D) │ (None, 256, 256, 32) │ 9,248 │ conv2d\_14[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_transpose\_3 │ (None, 512, 512, 16) │ 2,064 │ conv2d\_15[0][0] │  
│ (Conv2DTranspose) │ │ │ │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ concatenate\_3 │ (None, 512, 512, 32) │ 0 │ conv2d\_transpose\_3[0]… │  
│ (Concatenate) │ │ │ conv2d\_1[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_16 (Conv2D) │ (None, 512, 512, 16) │ 4,624 │ concatenate\_3[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_17 (Conv2D) │ (None, 512, 512, 16) │ 2,320 │ conv2d\_16[0][0] │  
├───────────────────────────┼────────────────────────┼────────────────┼────────────────────────┤  
│ conv2d\_18 (Conv2D) │ (None, 512, 512, 1) │ 17 │ conv2d\_17[0][0] │  
└───────────────────────────┴────────────────────────┴────────────────┴────────────────────────┘

**Total params:** 5,823,317 (22.21 MB)

**Trainable params:** 1,941,105 (7.40 MB)

**Non-trainable params:** 0 (0.00 B)

**Optimizer params:** 3,882,212 (14.81 MB)

None

In [ ]:

**import** os  
gtappend **=** []  
predappend **=**[]  
imgs **=** []  
addA **=** "/content/drive/Shareddrives/Untitled shared drive/Nitk Final Project/Test/I/x/" *# Added trailing slash*  
addlabel **=** "/content/drive/Shareddrives/Untitled shared drive/Nitk Final Project/Test/G/y/" *# Added trailing slash*  
  
**for** eg **in** os**.**listdir(addA):  
 imageA **=** os**.**path**.**join(addA, eg) *# Use os.path.join to create correct path*  
 GT **=** os**.**path**.**join(addlabel, eg) *# Use os.path.join to create correct path*  
 x1 **=** (plt**.**imread(imageA))**\***255  
 x2 **=** (plt**.**imread(GT))  
 a **=** np**.**reshape(x1,(1,512,512,3))  
 c **=** x2[:,:,0]  
 y **=** model**.**predict(a)  
 y **=** np**.**reshape(y,(512,512))  
 result **=** np**.**where(y**>**0.5,1,0)  
 predappend**.**append(result)  
 gtappend**.**append(c)  
 imgs**.**append(x1)  
  
g **=** np**.**stack(gtappend, axis**=**0)  
p **=** np**.**stack(predappend, axis**=**0)  
imgs **=** np**.**array(imgs)

/usr/local/lib/python3.11/dist-packages/keras/src/models/functional.py:237: UserWarning: The structure of `inputs` doesn't match the expected structure.  
Expected: ['keras\_tensor']  
Received: inputs=Tensor(shape=(1, 512, 512, 3))  
 warnings.warn(msg)

**1/1** ━━━━━━━━━━━━━━━━━━━━ **3s** 3s/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 58ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 44ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 46ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 44ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 44ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 45ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 68ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 46ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 46ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 45ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 45ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 46ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 62ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 44ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 46ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 44ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 45ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 50ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 44ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 57ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 51ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 46ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 49ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 52ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 62ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 44ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 46ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 52ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 45ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 46ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 63ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 60ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 45ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 49ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 48ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 46ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 50ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 62ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 69ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 45ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 45ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 44ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 53ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 44ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 57ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 127ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 45ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 47ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 51ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 46ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 45ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 61ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 45ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 45ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 45ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 44ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 45ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 61ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 47ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 46ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 45ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 45ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 46ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 46ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 76ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 45ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 45ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 46ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 46ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 46ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 62ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 44ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 46ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 49ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 63ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 47ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 61ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 52ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 51ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 49ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 51ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 52ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 69ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 50ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 70ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 49ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 51ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 49ms/step  
**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 67ms/step

In [ ]:

imgs**.**shape

Out[ ]:

(90, 512, 512, 3)

In [ ]:

fig**=**plt**.**figure()  
fig**.**figsize**=**(30,30)  
  
ax**=**fig**.**add\_subplot(1,3,1)  
ax**.**imshow((imgs[2]**/**255))  
plt**.**axis('off')  
plt**.**title("original")  
  
ax**=**fig**.**add\_subplot(1,3,2)  
ax**.**imshow(np**.**reshape(g[2],(512,512)),cmap**=**"gray")  
plt**.**axis('off')  
plt**.**title("Ground-truth")  
  
ax**=**fig**.**add\_subplot(1,3,3)  
ax**.**imshow(np**.**reshape(p[2],(512,512)),cmap**=**"gray")  
plt**.**axis('off')  
plt**.**title("Predicted image")

Out[ ]:

Text(0.5, 1.0, 'Predicted image')

In [ ]:

fig**=**plt**.**figure()  
fig**.**figsize**=**(30,30)  
  
ax**=**fig**.**add\_subplot(1,3,1)  
ax**.**imshow((imgs[19]**/**255))  
plt**.**axis('off')  
plt**.**title("original")  
  
ax**=**fig**.**add\_subplot(1,3,2)  
ax**.**imshow(np**.**reshape(g[19],(512,512)),cmap**=**"gray")  
plt**.**axis('off')  
plt**.**title("Ground-truth")  
  
ax**=**fig**.**add\_subplot(1,3,3)  
ax**.**imshow(np**.**reshape(p[19],(512,512)),cmap**=**"gray")  
plt**.**axis('off')  
plt**.**title("Predicted image")

Out[ ]:

Text(0.5, 1.0, 'Predicted image')

In [ ]:

**from** sklearn.metrics **import** **\***  
gt**=**(g**.**ravel())**.**astype('int')  
pd**=**(p**.**ravel())**.**astype('int')  
f1 **=** f1\_score(gt,pd,average**=**'macro')  
kappa **=** cohen\_kappa\_score(gt,pd)  
accuracy **=** accuracy\_score(gt,pd)  
jaccard **=** jaccard\_score(gt,pd,average**=**'macro')  
precision **=** precision\_score(gt,pd,average**=**'macro')  
recall **=** recall\_score(gt,pd,average**=**'macro')  
print(np**.**unique(gt),np**.**unique(pd))  
print("F1 SCORE:", f1)  
print("Kappa:",kappa)  
print("Accuracy:",accuracy)  
print("Jaccard Score:",jaccard)  
print("Precision:",precision)  
print("Recall:",recall)  
  
*### Dice Coefficient*  
**def** get\_dice(y\_true, y\_pred):  
 intersection **=** np**.**sum(y\_true **\*** y\_pred, axis**=**(0, 1, 2))  
 union **=** np**.**sum(y\_true**\*\***2, axis**=**(0, 1, 2)) **+** np**.**sum(y\_pred**\*\***2, axis**=**(0, 1, 2))  
 dc **=** 2 **\*** intersection **/** union  
 **return** dc  
  
dice\_coeff **=** get\_dice(g, p)  
print("dc = ",np**.**mean(dice\_coeff))

[0 1] [0 1]  
F1 SCORE: 0.76181859558259  
Kappa: 0.5274054734097693  
Accuracy: 0.875938966539171  
Jaccard Score: 0.6444274777766723  
Precision: 0.8238604089908255  
Recall: 0.7284901757855631  
dc = 0.5969509206821589

PART 2

In [ ]:

**import** random  
**import** cv2  
**import** numpy **as** np  
**import** tifffile **as** tiff  
**import** matplotlib.pyplot **as** plt  
**from** skimage **import** measure  
**from** skimage **import** filters  
  
**def** Opening(image, filter\_size):  
 *#Opening is just another name of erosion followed by dilation*  
 img**=**np**.**copy(image)  
 kernel**=**cv2**.**getStructuringElement(cv2**.**MORPH\_ELLIPSE,(filter\_size,filter\_size))  
 opening **=** cv2**.**morphologyEx(img, cv2**.**MORPH\_OPEN, kernel)  
 **return** opening  
  
**def** Closing(image,k):*# closing is useful to detect the overall contour of a figure and opening is suitable to detect subpatterns.*  
 kernel **=** np**.**ones((k, k), np**.**uint8)  
 img**=**np**.**copy(image)  
 img\_close **=** cv2**.**morphologyEx(img, op**=** cv2**.**MORPH\_CLOSE,kernel**=**kernel)  
 **return** img\_close  
  
**def** Denoise(image,k):  
 img**=**np**.**copy(image)  
 struct**=**cv2**.**getStructuringElement(cv2**.**MORPH\_ELLIPSE,(k,k))  
 img**=**cv2**.**morphologyEx(img,cv2**.**MORPH\_OPEN,struct)  
 **return** img  
  
**def** Binary(image, threshold, max\_value **=** 1):  
 img**=**np**.**copy(image)  
 (t,masklayer)**=**cv2**.**threshold(img,threshold,max\_value,cv2**.**THRESH\_BINARY)  
 **return** masklayer  
  
**def** Gaussian\_filter(image, sigma **=**1):  
 img**=**np**.**copy(image)  
 blur **=** filters**.**gaussian(img, sigma**=**sigma)  
 **return** blur  
  
**def** Find\_threshold\_otsu(image):  
 t **=** filters**.**threshold\_otsu(image)  
 **return** t  
  
**def** ExtractObjects(image):  
 img**=**np**.**copy(image)  
 blob\_labels**=**measure**.**label(img,background**=**0)  
 number\_of\_objects**=**np**.**unique(blob\_labels)  
 **return** blob\_labels,number\_of\_objects  
  
**def** post\_processing(img):  
  
 blur **=** Gaussian\_filter(img, sigma**=**1)  
 t **=** Find\_threshold\_otsu(blur)  
 binary\_img **=** Binary(blur,t)  
 opened\_img **=** Opening(binary\_img, filter\_size **=** 3)  
 blob\_labels,number\_of\_objects **=** ExtractObjects(opened\_img)  
  
 **return** opened\_img, number\_of\_objects, blob\_labels

In [ ]:

imgs**.**shape

Out[ ]:

(90, 512, 512, 3)

In [ ]:

number **=** 0 *#56 ,34*  
pred **=** p[number,:,:]  
  
opened\_img, number\_of\_objects, blob\_labels **=** post\_processing(pred)  
print("Number of Buildings in prediction: {}"**.**format(number\_of\_objects[**-**1]))  
  
gt **=** g[number,:,:]  
opened\_img1, number\_of\_objects1, blob\_labels1 **=** post\_processing(gt)  
print("Number of Buildings in Groundtruth: {}"**.**format(number\_of\_objects1[**-**1]))  
  
f, axs **=** plt**.**subplots(1,5, figsize**=**(23,23))  
img **=** imgs[number,:,:,:]**/**255  
img **=** np**.**reshape(img,(img**.**shape[0],img**.**shape[1],3))  
*# img = img.astype('uint8')*  
axs[0]**.**imshow(img)  
axs[0]**.**set\_title("Original")  
  
gt **=** g[number,:,:]  
gt **=** np**.**reshape(gt,(gt**.**shape[0],gt**.**shape[1]))  
axs[1]**.**imshow(gt)  
axs[1]**.**set\_title("Ground Truth")  
  
axs[2]**.**imshow(opened\_img)  
axs[2]**.**set\_title("Prediction")  
  
  
axs[3]**.**imshow(blob\_labels, cmap**=**'gist\_ncar')  
axs[3]**.**set\_title("Number of buildings: {}"**.**format(number\_of\_objects[**-**1]))  
  
axs[4]**.**imshow(blob\_labels1, cmap**=**'gist\_ncar')  
axs[4]**.**set\_title("Number of buildings in gt: {}"**.**format(number\_of\_objects1[**-**1]))  
  
f**.**show()

Number of Buildings in prediction: 229  
Number of Buildings in Groundtruth: 258

In [ ]:

number **=** 10 *#56 ,34*  
pred **=** p[number,:,:]  
  
opened\_img, number\_of\_objects, blob\_labels **=** post\_processing(pred)  
print("Number of Buildings in prediction: {}"**.**format(number\_of\_objects[**-**1]))  
  
gt **=** g[number,:,:]  
opened\_img1, number\_of\_objects1, blob\_labels1 **=** post\_processing(gt)  
print("Number of Buildings in Groundtruth: {}"**.**format(number\_of\_objects1[**-**1]))  
  
f, axs **=** plt**.**subplots(1,5, figsize**=**(23,23))  
img **=** imgs[number,:,:,:]**/**255  
img **=** np**.**reshape(img,(img**.**shape[0],img**.**shape[1],3))  
*# img = img.astype('uint8')*  
axs[0]**.**imshow(img)  
axs[0]**.**set\_title("Original")  
  
gt **=** g[number,:,:]  
gt **=** np**.**reshape(gt,(gt**.**shape[0],gt**.**shape[1]))  
axs[1]**.**imshow(gt)  
axs[1]**.**set\_title("Ground Truth")  
  
axs[2]**.**imshow(opened\_img)  
axs[2]**.**set\_title("Prediction")  
  
  
axs[3]**.**imshow(blob\_labels, cmap**=**'gist\_ncar')  
axs[3]**.**set\_title("Number of buildings: {}"**.**format(number\_of\_objects[**-**1]))  
  
axs[4]**.**imshow(blob\_labels1, cmap**=**'gist\_ncar')  
axs[4]**.**set\_title("Number of buildings in gt: {}"**.**format(number\_of\_objects1[**-**1]))  
  
f**.**show()

Number of Buildings in prediction: 186  
Number of Buildings in Groundtruth: 41

In [31]:

**for** i **in** range(10,20):  
 number **=** i  
 img **=** p[number,:,:]  
  
 opened\_img, number\_of\_objects, blob\_labels **=** post\_processing(img)  
 gt **=** g[number,:,:]  
 opened\_img1, number\_of\_objects1, blob\_labels1 **=** post\_processing(gt)  
  
 f, axs **=** plt**.**subplots(1,5, figsize**=**(23,23))  
 img **=** imgs[number,:,:,:]**/**255  
 img **=** np**.**reshape(img,(img**.**shape[0],img**.**shape[1],3))  
 *# img = img.astype('uint8')*  
 axs[0]**.**imshow(img)  
 axs[0]**.**set\_title("Original")  
  
 gt **=** g[number,:,:]  
 gt **=** np**.**reshape(gt,(gt**.**shape[0],gt**.**shape[1]))  
 axs[1]**.**imshow(gt)  
 axs[1]**.**set\_title("Ground Truth")  
  
 axs[2]**.**imshow(opened\_img)  
 axs[2]**.**set\_title("Prediction")  
  
 axs[3]**.**imshow(blob\_labels, cmap**=**'gist\_ncar')  
 axs[3]**.**set\_title("Number of buildings in prediction: {}"**.**format(number\_of\_objects[**-**1]))  
  
 axs[4]**.**imshow(blob\_labels1, cmap**=**'gist\_ncar')  
 axs[4]**.**set\_title("Number of buildings in gt: {}"**.**format(number\_of\_objects1[**-**1]))  
  
 f**.**show()

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

In [ ]: