```
# This Python 3 environment comes with many helpful analytics
libraries installed
# It is defined by the kaggle/python Docker image:
https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

# import numpy as np # linear algebra
# import pandas as pd # data processing, CSV file I/O (e.g.
pd.read_csv)

# Input data files are available in the read-only "../input/"
directory
# For example, running this (by clicking run or pressing Shift+Enter)
will list all files under the input directory

# import os
# for dirname, _, filenames in os.walk('/kaggle/input'):
    # for filename in filenames:
        # print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/)
that gets preserved as output when you create a version using "Save &
Run All"
# You can also write temporary files to /kaggle/temp/, but they won't
be saved outside of the current session
```

Import Libraries

```
import os
import torch
import numpy as np
import pandas as pd
import torch.nn as nn
import torch.optim as optim
import matplotlib.pyplot as plt
import torch.nn.functional as F
import torch.utils.checkpoint as C
import torchvision.transforms as T
import torchvision.transforms.functional as fn

from tqdm.notebook import tqdm
from torch.utils.data import Dataset
from torch.utils.data import DataLoader
from torchvision import models

!pip install /kaggle/input/segmentation-models-pytorch/timm-0.6.12-
py3-none-any.whl
!pip install /kaggle/input/segmentation-models-
```

```
pytorch/efficientnet_pytorch-0.7.1-py3-none-any.whl
!pip install /kaggle/input/segmentation-models-pytorch/munch-3.0.0-
py2.py3-none-any.whl
!pip install /kaggle/input/segmentation-models-
pytorch/pretrainedmodels-0.7.4-py3-none-any.whl
!pip install /kaggle/input/segmentation-models-
pytorch/segmentation_models_pytorch-0.3.2-py3-none-any.whl

import segmentation_models_pytorch as smp

!pip install /kaggle/input/torch-summary/torchsummary-1.5.1-py3-none-
any.whl
from torchsummary import summary
```

```
Processing /kaggle/input/segmentation-models-pytorch/timm-0.6.12-py3-
none-any.whl
Requirement already satisfied: torch>=1.7 in
/opt/conda/lib/python3.10/site-packages (from timm==0.6.12) (2.0.0)
Requirement already satisfied: torchvision in
/opt/conda/lib/python3.10/site-packages (from timm==0.6.12) (0.15.1)
Requirement already satisfied: pyyaml in
/opt/conda/lib/python3.10/site-packages (from timm==0.6.12) (5.4.1)
Requirement already satisfied: huggingface-hub in
/opt/conda/lib/python3.10/site-packages (from timm==0.6.12) (0.15.1)
Requirement already satisfied: filelock in
/opt/conda/lib/python3.10/site-packages (from torch>=1.7-
>timm==0.6.12) (3.12.0)
Requirement already satisfied: typing-extensions in
/opt/conda/lib/python3.10/site-packages (from torch>=1.7-
>timm==0.6.12) (4.5.0)
Requirement already satisfied: sympy in
/opt/conda/lib/python3.10/site-packages (from torch>=1.7-
>timm==0.6.12) (1.12)
Requirement already satisfied: networkx in
/opt/conda/lib/python3.10/site-packages (from torch>=1.7-
>timm==0.6.12) (3.1)
Requirement already satisfied: jinja2 in
/opt/conda/lib/python3.10/site-packages (from torch>=1.7-
>timm==0.6.12) (3.1.2)
Requirement already satisfied: fsspec in
/opt/conda/lib/python3.10/site-packages (from huggingface-hub-
>timm==0.6.12) (2023.6.0)
Requirement already satisfied: requests in
/opt/conda/lib/python3.10/site-packages (from huggingface-hub-
>timm==0.6.12) (2.28.2)
Requirement already satisfied: tqdm>=4.42.1 in
/opt/conda/lib/python3.10/site-packages (from huggingface-hub-
>timm==0.6.12) (4.64.1)
Requirement already satisfied: packaging>=20.9 in
/opt/conda/lib/python3.10/site-packages (from huggingface-hub-
>timm==0.6.12) (21.3)
Requirement already satisfied: numpy in
/opt/conda/lib/python3.10/site-packages (from torchvision-
>timm==0.6.12) (1.23.5)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in
/opt/conda/lib/python3.10/site-packages (from torchvision-
>timm==0.6.12) (9.5.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/opt/conda/lib/python3.10/site-packages (from packaging>=20.9-
>huggingface-hub->timm==0.6.12) (3.0.9)
Requirement already satisfied: MarkupSafe>=2.0 in
/opt/conda/lib/python3.10/site-packages (from jinja2->torch>=1.7-
>timm==0.6.12) (2.1.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/opt/conda/lib/python3.10/site-packages (from requests->huggingface-
hub->timm==0.6.12) (2.1.1)
Requirement already satisfied: idna<4,>=2.5 in
/opt/conda/lib/python3.10/site-packages (from requests->huggingface-
hub->timm==0.6.12) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/opt/conda/lib/python3.10/site-packages (from requests->huggingface-
hub->timm==0.6.12) (1.26.15)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/lib/python3.10/site-packages (from requests->huggingface-
hub->timm==0.6.12) (2023.5.7)
Requirement already satisfied: mpmath>=0.19 in
/opt/conda/lib/python3.10/site-packages (from sympy->torch>=1.7-
>timm==0.6.12) (1.3.0)
Installing collected packages: timm
  Attempting uninstall: timm
    Found existing installation: timm 0.9.2
    Uninstalling timm-0.9.2:
      Successfully uninstalled timm-0.9.2
Successfully installed timm-0.6.12
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
Processing
/kaggle/input/segmentation-models-pytorch/efficientnet_pytorch-0.7.1-
py3-none-any.whl
Requirement already satisfied: torch in
/opt/conda/lib/python3.10/site-packages (from efficientnet-
pytorch==0.7.1) (2.0.0)
Requirement already satisfied: filelock in
/opt/conda/lib/python3.10/site-packages (from torch->efficientnet-
pytorch==0.7.1) (3.12.0)
Requirement already satisfied: typing-extensions in
/opt/conda/lib/python3.10/site-packages (from torch->efficientnet-
pytorch==0.7.1) (4.5.0)
Requirement already satisfied: sympy in
/opt/conda/lib/python3.10/site-packages (from torch->efficientnet-
pytorch==0.7.1) (1.12)
Requirement already satisfied: networkx in
/opt/conda/lib/python3.10/site-packages (from torch->efficientnet-
pytorch==0.7.1) (3.1)
Requirement already satisfied: jinja2 in
/opt/conda/lib/python3.10/site-packages (from torch->efficientnet-
pytorch==0.7.1) (3.1.2)
Requirement already satisfied: MarkupSafe>=2.0 in
/opt/conda/lib/python3.10/site-packages (from jinja2->torch-
>efficientnet-pytorch==0.7.1) (2.1.2)
Requirement already satisfied: mpmath>=0.19 in
/opt/conda/lib/python3.10/site-packages (from sympy->torch-
>efficientnet-pytorch==0.7.1) (1.3.0)
Installing collected packages: efficientnet-pytorch
Successfully installed efficientnet-pytorch-0.7.1
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
Processing /kaggle/input/segmentation-models-pytorch/munch-3.0.0-
py2.py3-none-any.whl
Requirement already satisfied: six in /opt/conda/lib/python3.10/site-
packages (from munch==3.0.0) (1.16.0)
Installing collected packages: munch
Successfully installed munch-3.0.0
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
Processing /kaggle/input/segmentation-models-pytorch/pretrainedmodels-
0.7.4-py3-none-any.whl
Requirement already satisfied: torch in
/opt/conda/lib/python3.10/site-packages (from pretrainedmodels==0.7.4)
(2.0.0)
Requirement already satisfied: torchvision in
/opt/conda/lib/python3.10/site-packages (from pretrainedmodels==0.7.4)
(0.15.1)
Requirement already satisfied: munch in
/opt/conda/lib/python3.10/site-packages (from pretrainedmodels==0.7.4)
(3.0.0)
Requirement already satisfied: tqdm in /opt/conda/lib/python3.10/site-
packages (from pretrainedmodels==0.7.4) (4.64.1)
Requirement already satisfied: six in /opt/conda/lib/python3.10/site-
packages (from munch->pretrainedmodels==0.7.4) (1.16.0)
Requirement already satisfied: filelock in
/opt/conda/lib/python3.10/site-packages (from torch-
>pretrainedmodels==0.7.4) (3.12.0)
Requirement already satisfied: typing-extensions in
/opt/conda/lib/python3.10/site-packages (from torch-
>pretrainedmodels==0.7.4) (4.5.0)
Requirement already satisfied: sympy in
/opt/conda/lib/python3.10/site-packages (from torch-
>pretrainedmodels==0.7.4) (1.12)
Requirement already satisfied: networkx in
/opt/conda/lib/python3.10/site-packages (from torch-
>pretrainedmodels==0.7.4) (3.1)
Requirement already satisfied: jinja2 in
/opt/conda/lib/python3.10/site-packages (from torch-
>pretrainedmodels==0.7.4) (3.1.2)
Requirement already satisfied: numpy in
/opt/conda/lib/python3.10/site-packages (from torchvision-
>pretrainedmodels==0.7.4) (1.23.5)
Requirement already satisfied: requests in
/opt/conda/lib/python3.10/site-packages (from torchvision-
>pretrainedmodels==0.7.4) (2.28.2)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in
/opt/conda/lib/python3.10/site-packages (from torchvision-
>pretrainedmodels==0.7.4) (9.5.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/opt/conda/lib/python3.10/site-packages (from jinja2->torch-
>pretrainedmodels==0.7.4) (2.1.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/opt/conda/lib/python3.10/site-packages (from requests->torchvision-
>pretrainedmodels==0.7.4) (2.1.1)
Requirement already satisfied: idna<4,>=2.5 in
/opt/conda/lib/python3.10/site-packages (from requests->torchvision-
>pretrainedmodels==0.7.4) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/opt/conda/lib/python3.10/site-packages (from requests->torchvision-
>pretrainedmodels==0.7.4) (1.26.15)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/lib/python3.10/site-packages (from requests->torchvision-
>pretrainedmodels==0.7.4) (2023.5.7)
Requirement already satisfied: mpmath>=0.19 in
/opt/conda/lib/python3.10/site-packages (from sympy->torch-
>pretrainedmodels==0.7.4) (1.3.0)
Installing collected packages: pretrainedmodels
Successfully installed pretrainedmodels-0.7.4
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
Processing
/kaggle/input/segmentation-models-pytorch/segmentation_models_pytorch-
0.3.2-py3-none-any.whl
```

```
Requirement already satisfied: torchvision>=0.5.0 in
/opt/conda/lib/python3.10/site-packages (from segmentation-models-
pytorch==0.3.2) (0.15.1)
Requirement already satisfied: pretrainedmodels==0.7.4 in
/opt/conda/lib/python3.10/site-packages (from segmentation-models-
pytorch==0.3.2) (0.7.4)
Requirement already satisfied: efficientnet-pytorch==0.7.1 in
/opt/conda/lib/python3.10/site-packages (from segmentation-models-
pytorch==0.3.2) (0.7.1)
Requirement already satisfied: timm==0.6.12 in
/opt/conda/lib/python3.10/site-packages (from segmentation-models-
pytorch==0.3.2) (0.6.12)
Requirement already satisfied: tqdm in /opt/conda/lib/python3.10/site-
packages (from segmentation-models-pytorch==0.3.2) (4.64.1)
Requirement already satisfied: pillow in
/opt/conda/lib/python3.10/site-packages (from segmentation-models-
pytorch==0.3.2) (9.5.0)
Requirement already satisfied: torch in
/opt/conda/lib/python3.10/site-packages (from efficientnet-
pytorch==0.7.1->segmentation-models-pytorch==0.3.2) (2.0.0)
Requirement already satisfied: munch in
/opt/conda/lib/python3.10/site-packages (from pretrainedmodels==0.7.4-
>segmentation-models-pytorch==0.3.2) (3.0.0)
Requirement already satisfied: pyyaml in
/opt/conda/lib/python3.10/site-packages (from timm==0.6.12-
>segmentation-models-pytorch==0.3.2) (5.4.1)
Requirement already satisfied: huggingface-hub in
/opt/conda/lib/python3.10/site-packages (from timm==0.6.12-
>segmentation-models-pytorch==0.3.2) (0.15.1)
Requirement already satisfied: numpy in
/opt/conda/lib/python3.10/site-packages (from torchvision>=0.5.0-
>segmentation-models-pytorch==0.3.2) (1.23.5)
Requirement already satisfied: requests in
/opt/conda/lib/python3.10/site-packages (from torchvision>=0.5.0-
>segmentation-models-pytorch==0.3.2) (2.28.2)
Requirement already satisfied: filelock in
/opt/conda/lib/python3.10/site-packages (from torch->efficientnet-
pytorch==0.7.1->segmentation-models-pytorch==0.3.2) (3.12.0)
Requirement already satisfied: typing-extensions in
/opt/conda/lib/python3.10/site-packages (from torch->efficientnet-
pytorch==0.7.1->segmentation-models-pytorch==0.3.2) (4.5.0)
Requirement already satisfied: sympy in
/opt/conda/lib/python3.10/site-packages (from torch->efficientnet-
pytorch==0.7.1->segmentation-models-pytorch==0.3.2) (1.12)
Requirement already satisfied: networkx in
/opt/conda/lib/python3.10/site-packages (from torch->efficientnet-
pytorch==0.7.1->segmentation-models-pytorch==0.3.2) (3.1)
Requirement already satisfied: jinja2 in
/opt/conda/lib/python3.10/site-packages (from torch->efficientnet-
pytorch==0.7.1->segmentation-models-pytorch==0.3.2) (3.1.2)
Requirement already satisfied: fsspec in
/opt/conda/lib/python3.10/site-packages (from huggingface-hub-
>timm==0.6.12->segmentation-models-pytorch==0.3.2) (2023.6.0)
Requirement already satisfied: packaging>=20.9 in
/opt/conda/lib/python3.10/site-packages (from huggingface-hub-
>timm==0.6.12->segmentation-models-pytorch==0.3.2) (21.3)
Requirement already satisfied: six in /opt/conda/lib/python3.10/site-
packages (from munch->pretrainedmodels==0.7.4->segmentation-models-
pytorch==0.3.2) (1.16.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/opt/conda/lib/python3.10/site-packages (from requests-
>torchvision>=0.5.0->segmentation-models-pytorch==0.3.2) (2.1.1)
Requirement already satisfied: idna<4,>=2.5 in
/opt/conda/lib/python3.10/site-packages (from requests-
>torchvision>=0.5.0->segmentation-models-pytorch==0.3.2) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/opt/conda/lib/python3.10/site-packages (from requests-
>torchvision>=0.5.0->segmentation-models-pytorch==0.3.2) (1.26.15)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/lib/python3.10/site-packages (from requests-
>torchvision>=0.5.0->segmentation-models-pytorch==0.3.2) (2023.5.7)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/opt/conda/lib/python3.10/site-packages (from packaging>=20.9-
>huggingface-hub->timm==0.6.12->segmentation-models-pytorch==0.3.2)
(3.0.9)
Requirement already satisfied: MarkupSafe>=2.0 in
/opt/conda/lib/python3.10/site-packages (from jinja2->torch-
>efficientnet-pytorch==0.7.1->segmentation-models-pytorch==0.3.2)
(2.1.2)
Requirement already satisfied: mpmath>=0.19 in
/opt/conda/lib/python3.10/site-packages (from sympy->torch-
>efficientnet-pytorch==0.7.1->segmentation-models-pytorch==0.3.2)
(1.3.0)
Installing collected packages: segmentation-models-pytorch
Successfully installed segmentation-models-pytorch-0.3.2
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
Processing /kaggle/input/torch-summary/torchsummary-1.5.1-py3-none-
any.whl
Installing collected packages: torchsummary
Successfully installed torchsummary-1.5.1
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
```

```
!mkdir -p /root/.cache/torch/hub/checkpoints/
!cp /kaggle/input/timm-pretrained-resnest26/gluon_resnest26-
50eb607c.pth /root/.cache/torch/hub/checkpoints/gluon_resnest26-
50eb607c.pth
```

Utilities

```
def get_device():
    if torch.cuda.is_available():
        device = torch.device('cuda')
    else:
        device = torch.device('cpu')
    print(f'Using {device}')
    return device

device = get_device()
```

```
Using cuda
```

Import Data

```
class Config:
    # path to data folder
    data_dir = '/kaggle/input/google-research-identify-contrails-
reduce-global-warming'
    train_path = os.path.join(data_dir, 'train')
    val_path = os.path.join(data_dir, 'validation')
    test_path = os.path.join(data_dir, 'test')

    # base image size
    resize_value = 256

    # resize image
    resize = False
    if resize:
        resize_value = 384

    # model settings
    model = 'UNET'
    encoder = 'timm-resnest26d'
    weights = 'imagenet'

    epochs = 40
    batch_size = 16 #16
    lr = 5e-3
    optimizer = 'Adam'
```

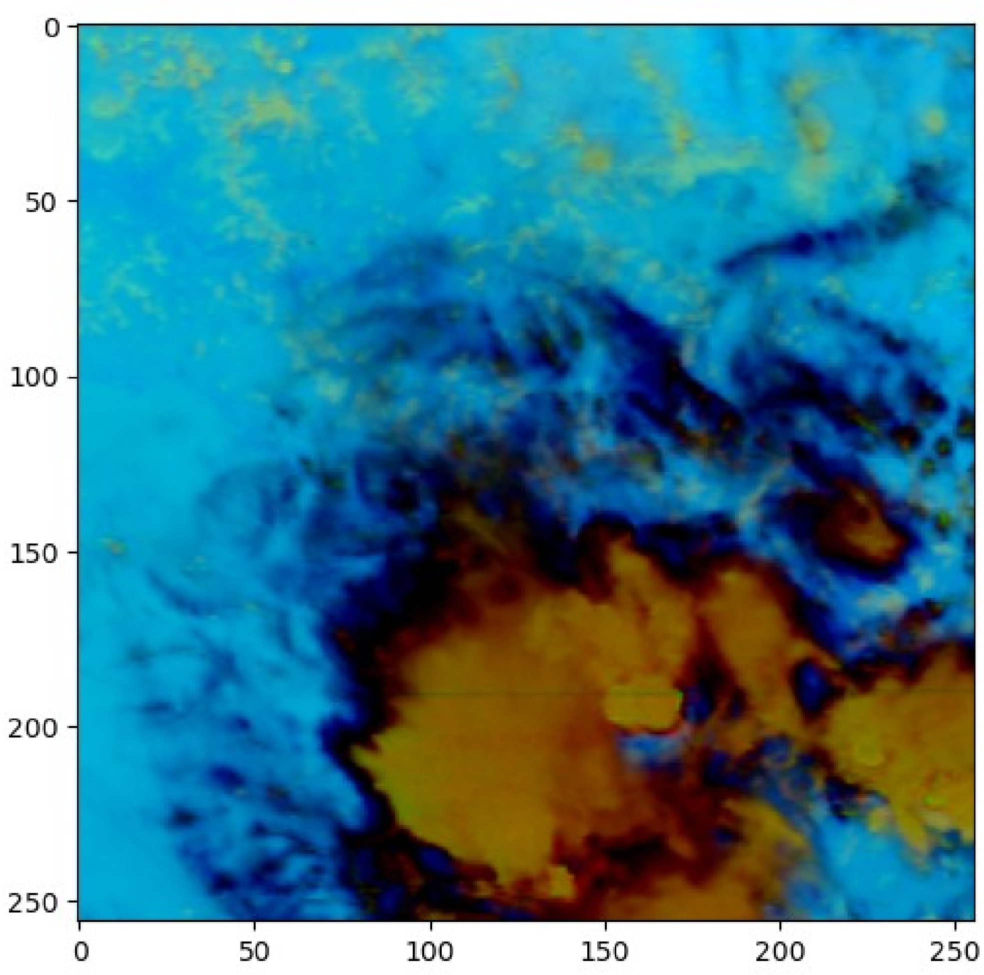
Create the Torch Dataset

```
_T11_BOUNDS = (243, 303)
_CLOUD_TOP_TDIFF_BOUNDS = (-4, 5)
_TDIFF_BOUNDS = (-4, 2)


fraction = 50/100
def normalize_range(data, bounds):
    # maps data to the range[0,1]
    return (data - bounds[0]) / (bounds[1] - bounds[0])

def normalize_std(spec):
    return(spec-np.mean(spec))/np.std(spec)

class ContrailDataset(Dataset):
    def __init__(self, data_dir, mode = 'train'):

        self.data_dir = data_dir
#         self.file_name = os.listdir(data_dir)
        if 'train' in data_dir:
            temp_file_name = os.listdir(data_dir)
            used = int(len(temp_file_name)*fraction)
            print('{:d} records in train, fraction of {:f} loaded,
{:d} recrods loaded '.format(len(temp_file_name), fraction, used))
            self.file_name = os.listdir(data_dir)[:used]
        elif 'validation' in data_dir: # validation
            temp_file_name = os.listdir(data_dir)
            used = int(len(temp_file_name)*fraction)
            print('{:d} records in validation, fraction of {:f}
loaded, {:d} recrods loaded '.format(len(temp_file_name), fraction,
used))
            self.file_name = os.listdir(data_dir)[:used]
        elif 'test' in data_dir: # test
            self.file_name = os.listdir(data_dir)


#         print(self.file_name)
        self.mode = mode

        self.resize_image = T.Resize(Config.resize_value,
interpolation = T.InterpolationMode.BILINEAR,
                                    antialias = True)
        self.resize_mask = T.Resize(Config.resize_value, interpolation
= T.InterpolationMode.NEAREST,
                                   antialias = True)
```

```python
    def __len__(self):
        # returns the number of samples in dataset
        return len(self.file_name)

    def __getitem__(self, i):
        # loads and returns a sample from the dataset at the given
index
        band11 = np.load(os.path.join(self.data_dir,
self.file_name[i], 'band_11.npy'))
        band14 = np.load(os.path.join(self.data_dir,
self.file_name[i], 'band_14.npy'))
        band15 = np.load(os.path.join(self.data_dir,
self.file_name[i], 'band_15.npy'))

        r = normalize_range(band15 - band14, _TDIFF_BOUNDS)
        g = normalize_range(band14 - band11, _CLOUD_TOP_TDIFF_BOUNDS)
        b = normalize_range(band14, _T11_BOUNDS)

        false_color = np.transpose(np.clip(np.stack([r,g,b], axis =
2), 0, 1)[:,:,:,4],(2,0,1))
        false_color = normalize_std(false_color)

        if self.mode == 'train':
            human_pixel_mask = np.load(os.path.join(self.data_dir,
self.file_name[i],

'human_pixel_masks.npy')).astype(np.float32).transpose(2,0,1)

        elif self.mode == 'test':
            human_pixel_mask = self.file_name[i]
        else:
            human_pixel_mask = None

        return false_color, human_pixel_mask

_T11_BOUNDS = (243, 303)
_CLOUD_TOP_TDIFF_BOUNDS = (-4, 5)
_TDIFF_BOUNDS = (-4, 2)

def normalize_range(data, bounds):
    # maps data to the range[0,1]
    return (data - bounds[0]) / (bounds[1] - bounds[0])

def normalize_std(spec):
    return(spec-np.mean(spec))/np.std(spec)

band11_test = np.load('/kaggle/input/google-research-identify-
contrails-reduce-global-warming/train/1000216489776414077/
band_11.npy')
band14_test = np.load('/kaggle/input/google-research-identify-
contrails-reduce-global-warming/train/1000216489776414077/
band_14.npy')
band15_test = np.load('/kaggle/input/google-research-identify-
contrails-reduce-global-warming/train/1000216489776414077/
band_15.npy')

r_test = normalize_range(band15_test - band14_test, _TDIFF_BOUNDS)
g_test = normalize_range(band14_test - band11_test,
_CLOUD_TOP_TDIFF_BOUNDS)
b_test = normalize_range(band14_test, _T11_BOUNDS)

print('band11',band11_test.shape)
print('band14',band14_test.shape)
print('band15',band15_test.shape)

print('r',r_test.shape)
print('g',g_test.shape)
print('b',b_test.shape)

clip_stack = np.clip(np.stack([r_test,g_test,b_test], axis = 2), 0, 1)
print('After clip and stack', clip_stack.shape)

before_transpose = clip_stack[:,:,:,4]
after_transpose = np.transpose(before_transpose, (2,0,1))
print('before transpose: ', before_transpose.shape)
print(before_transpose[0][0])
print('after transpose: ', after_transpose.shape)

false_color_test = normalize_std(after_transpose)
print('false color: ', false_color_test.shape)
# false_color = np.transpose(np.clip(np.stack([r,g,b], axis = 2), 0,
1)[:,:,:,4],(2,0,1))
# false_color = normalize_std(false_color)

plt.figure(figsize=(6,6))
ax = plt.subplot(1,1,1)
ax.imshow(before_transpose)
```

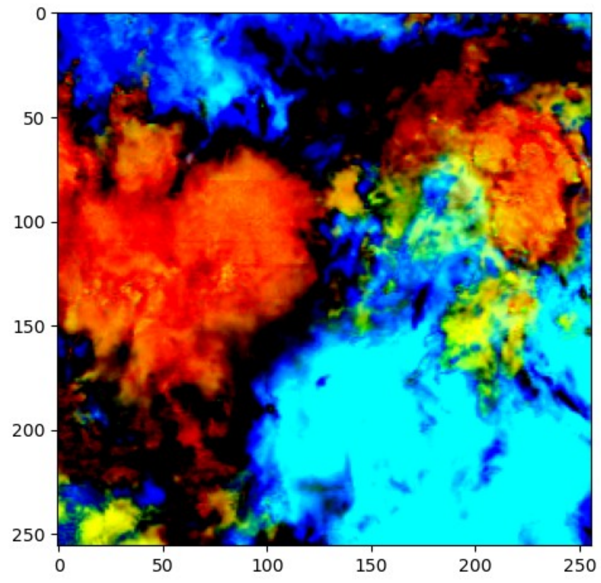
```
band11 (256, 256, 8)
band14 (256, 256, 8)
band15 (256, 256, 8)
r (256, 256, 8)
g (256, 256, 8)
b (256, 256, 8)
After clip and stack (256, 256, 3, 8)
before transpose:  (256, 256, 3)
[0.095637  0.7042304 0.7834071]
after transpose:  (3, 256, 256)
false color:  (3, 256, 256)
```

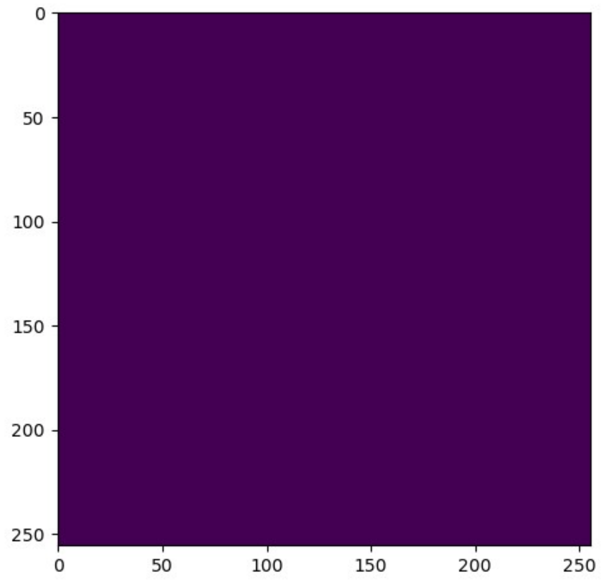
```
<matplotlib.image.AxesImage at 0x7e2e430b8760>
```



```
dataset = ContrailDataset(Config.train_path)
print('dataset: ', len(dataset))
a, b = dataset[1]
print('a: ', a.shape)
print('b: ', b.shape)

plt.figure(figsize=(12,6))
ax = plt.subplot(1,2,1)
a_transpose = np.transpose(a,(1,2,0))
ax.imshow(a_transpose)
print('a_transpose: ', a_transpose.shape)
print(a_transpose[0][0])
ax = plt.subplot(1,2,2)
b_transpose = np.transpose(b,(1,2,0))
ax.imshow(b_transpose, interpolation = 'none')
print('b_transpose: ', b_transpose.shape)
plt.show()
```

```
20529 records in train, fraction of 0.500000 loaded, 10264 recrods
loaded
dataset:  10264
a:  (3, 256, 256)
b:  (1, 256, 256)
a_transpose:  (256, 256, 3)
[-1.317837    0.09456838  1.4377766 ]
b_transpose:  (256, 256, 1)
```





Create the Training and Validation Dataloader

```
training_data = ContrailDataset(data_dir = Config.train_path)
train_dataloader = DataLoader(training_data,
                             batch_size = Config.batch_size,
                             shuffle = True,
                             num_workers = 2 if
torch.cuda.is_available() else 0,
                             pin_memory = True,
                             drop_last = True)

validation_data = ContrailDataset(data_dir = Config.val_path)
validation_dataloader = DataLoader(validation_data,
                                  batch_size = Config.batch_size,
                                  shuffle = False,
                                  num_workers = 2 if
torch.cuda.is_available() else 0,
                                  pin_memory = True,
                                  drop_last = True)
```

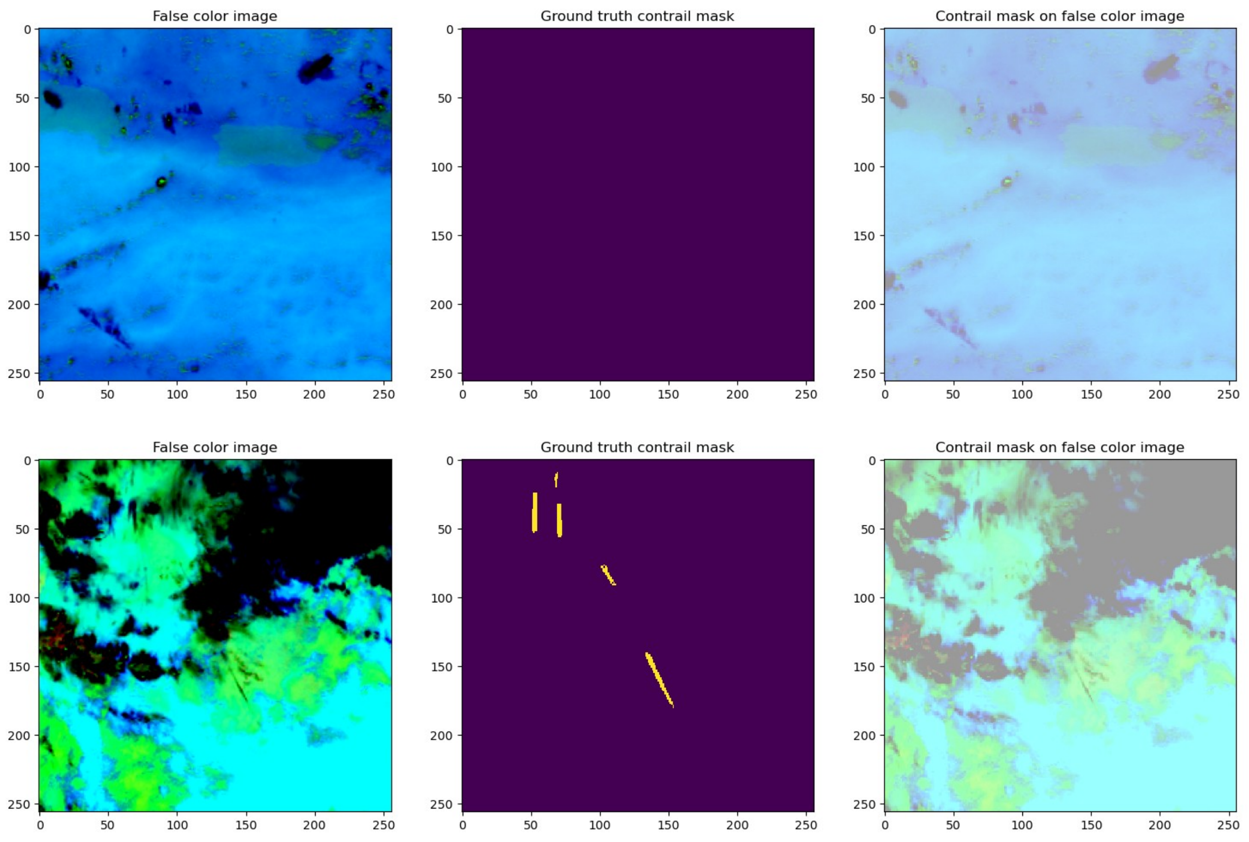
```
print('train_dataloader:', train_dataloader)
print('validation_dataloader:', validation_dataloader )
```

```
20529 records in train, fraction of 0.500000 loaded, 10264 recrods
loaded
1856 records in validation, fraction of 0.500000 loaded, 928 recrods
loaded
train_dataloader: <torch.utils.data.dataloader.DataLoader object at
0x7e2e27117130>
validation_dataloader: <torch.utils.data.dataloader.DataLoader object
at 0x7e2ee8f7da20>
```

Show Some Image from the Dataloaders

```
print('validation dataloader: ', validation_dataloader)
image, mask = next(iter(validation_dataloader))
print('image: ', image.shape)
print('mask: ', mask.shape)

image = torch.moveaxis(image, 1, -1)
mask = torch.moveaxis(mask, 1, -1)
print('after moveaxis')
print('image: ', image.shape)
print('mask: ', mask.shape)

for i in range(2):
    plt.figure(figsize=(18,6))

    ax = plt.subplot(1,3,1)
    ax.imshow(image[i])
    ax.set_title('False color image')

    ax = plt.subplot(1,3,2)
    ax.imshow(mask[i], interpolation = 'none')
    ax.set_title('Ground truth contrail mask')

    ax = plt.subplot(1,3,3)
    ax.imshow(image[i], cmap = 'Reds', alpha = .4, interpolation =
'none')
    ax.set_title('Contrail mask on false color image')
```

```
validation dataloader:  <torch.utils.data.dataloader.DataLoader object
at 0x7e2ee8f7da20>
image:  torch.Size([16, 3, 256, 256])
mask:  torch.Size([16, 1, 256, 256])
after moveaxis
image:  torch.Size([16, 256, 256, 3])
mask:  torch.Size([16, 256, 256, 1])
```


False color image
Ground truth contrail mask
Contrail mask on false color image
False color image
Ground truth contrail mask
Contrail mask on false color image

Create the Model UNET

```
if Config.model == 'UNET':
    model = smp.Unet(encoder_name = Config.encoder, encoder_weights =
Config.weights,
                     in_channels = 3, activation = 'sigmoid')
    model.to(device)
    summary(model,(3, 256, 256))
```

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1        [-1, 32, 128, 128]             864
       BatchNorm2d-2        [-1, 32, 128, 128]              64
              ReLU-3        [-1, 32, 128, 128]               0
            Conv2d-4        [-1, 32, 128, 128]           9,216
       BatchNorm2d-5        [-1, 32, 128, 128]              64
              ReLU-6        [-1, 32, 128, 128]               0
            Conv2d-7        [-1, 64, 128, 128]          18,432
       BatchNorm2d-8        [-1, 64, 128, 128]             128
              ReLU-9        [-1, 64, 128, 128]               0
        MaxPool2d-10          [-1, 64, 64, 64]               0
           Conv2d-11          [-1, 64, 64, 64]           4,096
```

```
      BatchNorm2d-12           [-1, 64, 64, 64]             128
             ReLU-13           [-1, 64, 64, 64]               0
           Conv2d-14          [-1, 128, 64, 64]          36,864
      BatchNorm2d-15          [-1, 128, 64, 64]             256
         Identity-16          [-1, 128, 64, 64]               0
             ReLU-17          [-1, 128, 64, 64]               0
           Conv2d-18             [-1, 32, 1, 1]           2,080
      BatchNorm2d-19             [-1, 32, 1, 1]              64
             ReLU-20             [-1, 32, 1, 1]               0
           Conv2d-21            [-1, 128, 1, 1]           4,224
     RadixSoftmax-22                  [-1, 128]               0
        SplitAttn-23           [-1, 64, 64, 64]               0
         Identity-24           [-1, 64, 64, 64]               0
         Identity-25           [-1, 64, 64, 64]               0
         Identity-26           [-1, 64, 64, 64]               0
           Conv2d-27          [-1, 256, 64, 64]          16,384
      BatchNorm2d-28          [-1, 256, 64, 64]             512
         Identity-29           [-1, 64, 64, 64]               0
           Conv2d-30          [-1, 256, 64, 64]          16,384
      BatchNorm2d-31          [-1, 256, 64, 64]             512
             ReLU-32          [-1, 256, 64, 64]               0
ResNestBottleneck-33          [-1, 256, 64, 64]               0
           Conv2d-34           [-1, 64, 64, 64]          16,384
      BatchNorm2d-35           [-1, 64, 64, 64]             128
             ReLU-36           [-1, 64, 64, 64]               0
           Conv2d-37          [-1, 128, 64, 64]          36,864
      BatchNorm2d-38          [-1, 128, 64, 64]             256
         Identity-39          [-1, 128, 64, 64]               0
             ReLU-40          [-1, 128, 64, 64]               0
           Conv2d-41             [-1, 32, 1, 1]           2,080
      BatchNorm2d-42             [-1, 32, 1, 1]              64
             ReLU-43             [-1, 32, 1, 1]               0
           Conv2d-44            [-1, 128, 1, 1]           4,224
     RadixSoftmax-45                  [-1, 128]               0
        SplitAttn-46           [-1, 64, 64, 64]               0
         Identity-47           [-1, 64, 64, 64]               0
         Identity-48           [-1, 64, 64, 64]               0
         Identity-49           [-1, 64, 64, 64]               0
           Conv2d-50          [-1, 256, 64, 64]          16,384
      BatchNorm2d-51          [-1, 256, 64, 64]             512
             ReLU-52          [-1, 256, 64, 64]               0
ResNestBottleneck-53          [-1, 256, 64, 64]               0
           Conv2d-54          [-1, 128, 64, 64]          32,768
      BatchNorm2d-55          [-1, 128, 64, 64]             256
             ReLU-56          [-1, 128, 64, 64]               0
           Conv2d-57          [-1, 256, 64, 64]         147,456
      BatchNorm2d-58          [-1, 256, 64, 64]             512
         Identity-59          [-1, 256, 64, 64]               0
             ReLU-60          [-1, 256, 64, 64]               0
           Conv2d-61                [-1, 64, 1, 1]            8,256
      BatchNorm2d-62                [-1, 64, 1, 1]              128
             ReLU-63                [-1, 64, 1, 1]                0
           Conv2d-64               [-1, 256, 1, 1]           16,640
     RadixSoftmax-65                     [-1, 256]                0
        SplitAttn-66             [-1, 128, 64, 64]                0
         Identity-67             [-1, 128, 64, 64]                0
         Identity-68             [-1, 128, 64, 64]                0
         Identity-69             [-1, 128, 64, 64]                0
        AvgPool2d-70             [-1, 128, 32, 32]                0
           Conv2d-71             [-1, 512, 32, 32]           65,536
      BatchNorm2d-72             [-1, 512, 32, 32]            1,024
        AvgPool2d-73             [-1, 256, 32, 32]                0
           Conv2d-74             [-1, 512, 32, 32]          131,072
      BatchNorm2d-75             [-1, 512, 32, 32]            1,024
             ReLU-76             [-1, 512, 32, 32]                0
ResNestBottleneck-77             [-1, 512, 32, 32]                0
           Conv2d-78             [-1, 128, 32, 32]           65,536
      BatchNorm2d-79             [-1, 128, 32, 32]              256
             ReLU-80             [-1, 128, 32, 32]                0
           Conv2d-81             [-1, 256, 32, 32]          147,456
      BatchNorm2d-82             [-1, 256, 32, 32]              512
         Identity-83             [-1, 256, 32, 32]                0
             ReLU-84             [-1, 256, 32, 32]                0
           Conv2d-85                [-1, 64, 1, 1]            8,256
      BatchNorm2d-86                [-1, 64, 1, 1]              128
             ReLU-87                [-1, 64, 1, 1]                0
           Conv2d-88               [-1, 256, 1, 1]           16,640
     RadixSoftmax-89                     [-1, 256]                0
        SplitAttn-90             [-1, 128, 32, 32]                0
         Identity-91             [-1, 128, 32, 32]                0
         Identity-92             [-1, 128, 32, 32]                0
         Identity-93             [-1, 128, 32, 32]                0
           Conv2d-94             [-1, 512, 32, 32]           65,536
      BatchNorm2d-95             [-1, 512, 32, 32]            1,024
             ReLU-96             [-1, 512, 32, 32]                0
ResNestBottleneck-97             [-1, 512, 32, 32]                0
           Conv2d-98             [-1, 256, 32, 32]          131,072
      BatchNorm2d-99             [-1, 256, 32, 32]              512
            ReLU-100             [-1, 256, 32, 32]                0
          Conv2d-101             [-1, 512, 32, 32]          589,824
     BatchNorm2d-102             [-1, 512, 32, 32]            1,024
        Identity-103             [-1, 512, 32, 32]                0
            ReLU-104             [-1, 512, 32, 32]                0
          Conv2d-105               [-1, 128, 1, 1]           32,896
     BatchNorm2d-106               [-1, 128, 1, 1]              256
            ReLU-107               [-1, 128, 1, 1]                0
          Conv2d-108               [-1, 512, 1, 1]           66,048
    RadixSoftmax-109                     [-1, 512]                0
```

```
        SplitAttn-110          [-1, 256, 32, 32]               0
         Identity-111          [-1, 256, 32, 32]               0
         Identity-112          [-1, 256, 32, 32]               0
         Identity-113          [-1, 256, 32, 32]               0
        AvgPool2d-114          [-1, 256, 16, 16]               0
           Conv2d-115         [-1, 1024, 16, 16]         262,144
      BatchNorm2d-116         [-1, 1024, 16, 16]           2,048
        AvgPool2d-117          [-1, 512, 16, 16]               0
           Conv2d-118         [-1, 1024, 16, 16]         524,288
      BatchNorm2d-119         [-1, 1024, 16, 16]           2,048
             ReLU-120         [-1, 1024, 16, 16]               0
ResNestBottleneck-121          [-1, 1024, 16, 16]               0
           Conv2d-122          [-1, 256, 16, 16]         262,144
      BatchNorm2d-123          [-1, 256, 16, 16]             512
             ReLU-124          [-1, 256, 16, 16]               0
           Conv2d-125          [-1, 512, 16, 16]         589,824
      BatchNorm2d-126          [-1, 512, 16, 16]           1,024
         Identity-127          [-1, 512, 16, 16]               0
             ReLU-128          [-1, 512, 16, 16]               0
           Conv2d-129            [-1, 128, 1, 1]          32,896
      BatchNorm2d-130            [-1, 128, 1, 1]             256
             ReLU-131            [-1, 128, 1, 1]               0
           Conv2d-132            [-1, 512, 1, 1]          66,048
     RadixSoftmax-133                  [-1, 512]               0
        SplitAttn-134          [-1, 256, 16, 16]               0
         Identity-135          [-1, 256, 16, 16]               0
         Identity-136          [-1, 256, 16, 16]               0
         Identity-137          [-1, 256, 16, 16]               0
           Conv2d-138         [-1, 1024, 16, 16]         262,144
      BatchNorm2d-139         [-1, 1024, 16, 16]           2,048
             ReLU-140         [-1, 1024, 16, 16]               0
ResNestBottleneck-141          [-1, 1024, 16, 16]               0
           Conv2d-142          [-1, 512, 16, 16]         524,288
      BatchNorm2d-143          [-1, 512, 16, 16]           1,024
             ReLU-144          [-1, 512, 16, 16]               0
           Conv2d-145         [-1, 1024, 16, 16]       2,359,296
      BatchNorm2d-146         [-1, 1024, 16, 16]           2,048
         Identity-147         [-1, 1024, 16, 16]               0
             ReLU-148         [-1, 1024, 16, 16]               0
           Conv2d-149            [-1, 256, 1, 1]         131,328
      BatchNorm2d-150            [-1, 256, 1, 1]             512
             ReLU-151            [-1, 256, 1, 1]               0
           Conv2d-152           [-1, 1024, 1, 1]         263,168
     RadixSoftmax-153                 [-1, 1024]               0
        SplitAttn-154          [-1, 512, 16, 16]               0
         Identity-155          [-1, 512, 16, 16]               0
         Identity-156          [-1, 512, 16, 16]               0
         Identity-157          [-1, 512, 16, 16]               0
        AvgPool2d-158            [-1, 512, 8, 8]               0
```

```
          Conv2d-159          [-1, 2048, 8, 8]       1,048,576
     BatchNorm2d-160          [-1, 2048, 8, 8]           4,096
       AvgPool2d-161          [-1, 1024, 8, 8]               0
          Conv2d-162          [-1, 2048, 8, 8]       2,097,152
     BatchNorm2d-163          [-1, 2048, 8, 8]           4,096
            ReLU-164          [-1, 2048, 8, 8]               0
ResNestBottleneck-165          [-1, 2048, 8, 8]               0
          Conv2d-166           [-1, 512, 8, 8]       1,048,576
     BatchNorm2d-167           [-1, 512, 8, 8]           1,024
            ReLU-168           [-1, 512, 8, 8]               0
          Conv2d-169          [-1, 1024, 8, 8]       2,359,296
     BatchNorm2d-170          [-1, 1024, 8, 8]           2,048
        Identity-171          [-1, 1024, 8, 8]               0
            ReLU-172          [-1, 1024, 8, 8]               0
          Conv2d-173           [-1, 256, 1, 1]         131,328
     BatchNorm2d-174           [-1, 256, 1, 1]             512
            ReLU-175           [-1, 256, 1, 1]               0
          Conv2d-176          [-1, 1024, 1, 1]         263,168
    RadixSoftmax-177                [-1, 1024]               0
       SplitAttn-178           [-1, 512, 8, 8]               0
        Identity-179           [-1, 512, 8, 8]               0
        Identity-180           [-1, 512, 8, 8]               0
        Identity-181           [-1, 512, 8, 8]               0
          Conv2d-182          [-1, 2048, 8, 8]       1,048,576
     BatchNorm2d-183          [-1, 2048, 8, 8]           4,096
            ReLU-184          [-1, 2048, 8, 8]               0
ResNestBottleneck-185          [-1, 2048, 8, 8]               0
  ResNestEncoder-186  [[-1, 3, 256, 256], [-1, 64, 128, 128], [-1,
256, 64, 64], [-1, 512, 32, 32], [-1, 1024, 16, 16], [-1, 2048, 8, 8]]
0
        Identity-187          [-1, 2048, 8, 8]               0
        Identity-188        [-1, 3072, 16, 16]               0
       Attention-189        [-1, 3072, 16, 16]               0
          Conv2d-190         [-1, 256, 16, 16]       7,077,888
     BatchNorm2d-191         [-1, 256, 16, 16]             512
            ReLU-192         [-1, 256, 16, 16]               0
          Conv2d-193         [-1, 256, 16, 16]         589,824
     BatchNorm2d-194         [-1, 256, 16, 16]             512
            ReLU-195         [-1, 256, 16, 16]               0
        Identity-196         [-1, 256, 16, 16]               0
       Attention-197         [-1, 256, 16, 16]               0
    DecoderBlock-198         [-1, 256, 16, 16]               0
        Identity-199         [-1, 768, 32, 32]               0
       Attention-200         [-1, 768, 32, 32]               0
          Conv2d-201         [-1, 128, 32, 32]         884,736
     BatchNorm2d-202         [-1, 128, 32, 32]             256
            ReLU-203         [-1, 128, 32, 32]               0
          Conv2d-204         [-1, 128, 32, 32]         147,456
     BatchNorm2d-205         [-1, 128, 32, 32]             256
```

```
           ReLU-206          [-1, 128, 32, 32]                0
       Identity-207          [-1, 128, 32, 32]                0
      Attention-208          [-1, 128, 32, 32]                0
   DecoderBlock-209          [-1, 128, 32, 32]                0
       Identity-210          [-1, 384, 64, 64]                0
      Attention-211          [-1, 384, 64, 64]                0
         Conv2d-212           [-1, 64, 64, 64]          221,184
    BatchNorm2d-213           [-1, 64, 64, 64]              128
           ReLU-214           [-1, 64, 64, 64]                0
         Conv2d-215           [-1, 64, 64, 64]           36,864
    BatchNorm2d-216           [-1, 64, 64, 64]              128
           ReLU-217           [-1, 64, 64, 64]                0
       Identity-218           [-1, 64, 64, 64]                0
      Attention-219           [-1, 64, 64, 64]                0
   DecoderBlock-220           [-1, 64, 64, 64]                0
       Identity-221        [-1, 128, 128, 128]                0
      Attention-222        [-1, 128, 128, 128]                0
         Conv2d-223         [-1, 32, 128, 128]           36,864
    BatchNorm2d-224         [-1, 32, 128, 128]               64
           ReLU-225         [-1, 32, 128, 128]                0
         Conv2d-226         [-1, 32, 128, 128]            9,216
    BatchNorm2d-227         [-1, 32, 128, 128]               64
           ReLU-228         [-1, 32, 128, 128]                0
       Identity-229         [-1, 32, 128, 128]                0
      Attention-230         [-1, 32, 128, 128]                0
   DecoderBlock-231         [-1, 32, 128, 128]                0
         Conv2d-232         [-1, 16, 256, 256]            4,608
    BatchNorm2d-233         [-1, 16, 256, 256]               32
           ReLU-234         [-1, 16, 256, 256]                0
         Conv2d-235         [-1, 16, 256, 256]            2,304
    BatchNorm2d-236         [-1, 16, 256, 256]               32
           ReLU-237         [-1, 16, 256, 256]                0
       Identity-238         [-1, 16, 256, 256]                0
      Attention-239         [-1, 16, 256, 256]                0
   DecoderBlock-240         [-1, 16, 256, 256]                0
    UnetDecoder-241         [-1, 16, 256, 256]                0
         Conv2d-242          [-1, 1, 256, 256]              145
       Identity-243          [-1, 1, 256, 256]                0
        Sigmoid-244          [-1, 1, 256, 256]                0
     Activation-245          [-1, 1, 256, 256]                0
================================================================
Total params: 24,033,521
Trainable params: 24,033,521
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.75
Forward/backward pass size (MB): 629.08
Params size (MB): 91.68
```

```
Estimated Total Size (MB): 721.51
----------------------------------------------------------------
```

Optimizer

```
optimizer = optim.Adam(model.parameters(), lr = Config.lr)
scheduler = torch.optim.lr_scheduler.ReduceLROnPlateau(optimizer, mode
= 'min', patience = 1,
                                                       factor = 0.5,
verbose = True)
print(f'Learning rate: {optimizer.param_groups[0]["lr"]}')

Learning rate: 0.005
```

Loss Function

```
# average dice score for the example in a batch
def dice_avg(y_p, y_t, smooth = 1e-3):
    i = torch.sum(y_p * y_t, dim = (2,3))
#     print('i:',i)
    u = torch.sum(y_p, dim = (2,3)) + torch.sum(y_t, dim = (2, 3))
#     print('u:',u)
    score = (2 * i + smooth) / (u + smooth)
    return torch.mean(score)

def dice_loss_avg(y_p, y_t):
    return 1- dice_score_jan(y_p, y_t)

def dice_global(y_p, y_t, smooth = 1e-3):
    intersection = torch.sum(y_p * y_t)
#     print('intersection:',intersection)
    union = torch.sum(y_p) + torch.sum(y_t)
#     print('union:',union)
    dice = (2.0 * intersection + smooth) / (union + smooth)

    return dice


def dice_loss_global(y_p, y_t):
    return 1 - dice_global(y_p, y_t)
```

Training and Validation Loop

```
train_dice_global_test = []
train_dice_avg_test = []
```

```
eval_dice_global_test = []
eval_dice_avg_test = []


model.train()
bar_test = tqdm(train_dataloader)
tot_loss_global_test = 0
tot_dice_global_test = 0
tot_dice_avg_test = 0
# print(bar_test)
count = 0
for image, mask in bar_test:
#         print('image:',image.shape)
#         print('mask:',mask.shape)
        image = torch.nn.functional.interpolate(image,
                                               size =
Config.resize_value,
                                               mode = 'bilinear')
        # transfer to device
        image, mask = image.to(device), mask.to(device)

        # set optimizere gradients to zero
        optimizer.zero_grad()

        # perform inference
        pred_mask = model(image)

        # if the image was resized, use a resizing step to make 256
again
        if Config.resize:
            pred_mask = torch.nn.functional.interpolate(pred_mask,
                                                       size = 256,
                                                       mode =
'bilinear')

        # calculate the loss and do a backward pass
        loss = dice_loss_global(pred_mask, mask)
        loss.backward()

        # adjust the weights
        optimizer.step()

        tot_loss_global_test += loss.item()
#         print('loss:', loss.item())
        tot_dice_global_test += 1 - loss.item()
#         print('dice:', 1-loss.item())
        loss_avg = dice_avg(pred_mask, mask).item()
#         print('dice avg:', loss_avg)
        tot_dice_avg_test += loss_avg
        count += 1
        bar_test.set_postfix(TrainDiceLossGlobal =
f'{tot_loss_global_test/count: .4f}',
                             TrainDiceGlobal =
f'{tot_dice_global_test/count: .4f}',
                             TrainDiceAvg =
f'{tot_dice_avg_test/count: .4f}')
```

{"model_id":"2da3e7ade17d4661862825728a72c1ad","version_major":2,"version_minor":0}

```
train_dice_global = []
train_dice_avg = []
eval_dice_global = []
eval_dice_avg = []
bst_dice = 0
bst_epoch = 1

for epoch in range(1, Config.epochs + 1):
    print(f'----------epoch: {epoch}----------')

    # early stopping
    if epoch-bst_epoch >= 10:
        print(f'early stopping in epoch {epoch}')
        break

    model.train()
    bar = tqdm(train_dataloader)
    tot_loss_global = 0
    tot_dice_global = 0
    tot_dice_avg = 0
    count = 0

    for image, mask in bar:

        image = torch.nn.functional.interpolate(image,
                                               size =
Config.resize_value,
                                               mode = 'bilinear')
        # transfer to device
        image, mask = image.to(device), mask.to(device)

        # set optimizere gradients to zero
        optimizer.zero_grad()

        # perform inference
        pred_mask = model(image)
        # if the image was resized, use a resizing step to make 256
again
        if Config.resize:
            pred_mask = torch.nn.functional.interpolate(pred_mask,
                                                        size = 256,
                                                        mode =
'bilinear')

        # calculate the loss and do a backward pass
        loss = dice_loss_global(pred_mask, mask)
        loss.backward()

        # adjust the weights
        optimizer.step()

        tot_loss_global += loss.item()
        tot_dice_global += 1 - loss.item()
        tot_dice_avg += dice_avg(pred_mask, mask).item()

        count += 1
        bar.set_postfix(TrainDiceLossGlobal =
f'{tot_loss_global/count: .4f}',
                       TrainDiceGlobal =
f'{tot_dice_global/count: .4f}',
                       TrainDiceAvg = f'{tot_dice_avg/count: .4f}')

    train_dice_global.append(np.array(tot_dice_global/count))
    train_dice_avg.append(np.array(tot_dice_avg/count))

    model.train(False)
    bar = tqdm(validation_dataloader)
    tot_dice_global = 0
    tot_dice_avg = 0
    count = 0

    for image, mask in bar:
        if Config.resize:
            image = torch.nn.functional.interpolate(image, size =
Config.resize_value, mode = 'bilinear')

        image, mask = image.to(device), mask.to(device)
        pred_mask = model(image)

        if Config.resize:
            pred_mask = torch.nn.functional.interpolate(pred_mask,
size = 256, mode = 'bilinear')

        tot_dice_global += dice_global(pred_mask, mask).item()
        tot_dice_avg += dice_avg(pred_mask, mask).item()
        count += 1
        bar.set_postfix(ValidDiceGlobal =
f'{tot_dice_global/count: .4f}',
                        ValidDiceAcg = f'{tot_dice_avg/count: .4f}')

    eval_dice_global.append(np.array(tot_dice_global/count))
    eval_dice_avg.append(np.array(tot_dice_avg/count))
    scheduler.step(1-(tot_dice_global/count))
    print(f'learning rate: {optimizer.param_groups[0]["lr"]}')

    if tot_dice_global/count > bst_dice:
        bst_dice = tot_dice_global/count
        bst_epoch = epoch
        torch.save(model.state_dict(),
f'model_state_dict_epoch_{epoch}_dice_{bst_dice: .4f}.pth')
        torch.save(model,
f'model_epoch{epoch}_dice_{bst_dice: .4f}.pt')
        print(f'current model saved! Epoch:{epoch} global dice:
{bst_dice} avg dice: {tot_dice_avg/count}')
```

```
----------epoch: 1----------
```

{"model_id":"fea41bef6c724be095121f5d4b1afab6","version_major":2,"vers
ion_minor":0}

{"model_id":"c3a450121af04dc2824aa0d673597127","version_major":2,"vers
ion_minor":0}

```
learning rate: 0.005
current model saved! Epoch:1 global dice: 0.347643858152593 avg dice:
0.23480328082524496
----------epoch: 2----------
```

{"model_id":"e3eec3a0d636452b94c978035fab45ad","version_major":2,"vers
ion_minor":0}

{"model_id":"8e33fbc023ee4ddfb362a45d7d9368c2","version_major":2,"vers
ion_minor":0}

```
learning rate: 0.005
current model saved! Epoch:2 global dice: 0.4307735179463105 avg dice:
0.6415111879850256
----------epoch: 3----------
```

{"model_id":"60f0aeb90c6e446fb3a7d5cb6015a941","version_major":2,"vers
ion_minor":0}

{"model_id":"2b4515d936614b9389c2c16e23d157aa","version_major":2,"vers
ion_minor":0}

```
learning rate: 0.005
----------epoch: 4----------

{"model_id":"900322865ffd4f318ed9986dc1d73522","version_major":2,"vers
ion_minor":0}

{"model_id":"326c596ef9e840578c1faff31f2df89d","version_major":2,"vers
ion_minor":0}

Epoch 00004: reducing learning rate of group 0 to 2.5000e-03.
learning rate: 0.0025
----------epoch: 5----------

{"model_id":"4784f0e22c7944eaaab4b1592384eccc","version_major":2,"vers
ion_minor":0}

{"model_id":"e6f2c9f7ddea46c9b0f44b1a5f4d7ba7","version_major":2,"vers
ion_minor":0}

learning rate: 0.0025
current model saved! Epoch:5 global dice: 0.5056620444367252 avg dice:
0.6910909917847864
----------epoch: 6----------

{"model_id":"6a92c154630c49679ccc79d3b70d43ee","version_major":2,"vers
ion_minor":0}

{"model_id":"de28fee39c5d4d01b35cdbc091f7a3ca","version_major":2,"vers
ion_minor":0}

learning rate: 0.0025
----------epoch: 7----------

{"model_id":"e4e9e2503a4b4a1685af9c4a04179bc0","version_major":2,"vers
ion_minor":0}

{"model_id":"6f657381c6194dd7bfe187759566452e","version_major":2,"vers
ion_minor":0}

Epoch 00007: reducing learning rate of group 0 to 1.2500e-03.
learning rate: 0.00125
----------epoch: 8----------

{"model_id":"7fc78a9d22364007a49fdc4ed17aef36","version_major":2,"vers
ion_minor":0}

{"model_id":"79c6983382354e01aebca88f553c5317","version_major":2,"vers
ion_minor":0}

learning rate: 0.00125
current model saved! Epoch:8 global dice: 0.5211136662376169 avg dice:
0.722567390265136
----------epoch: 9----------
```

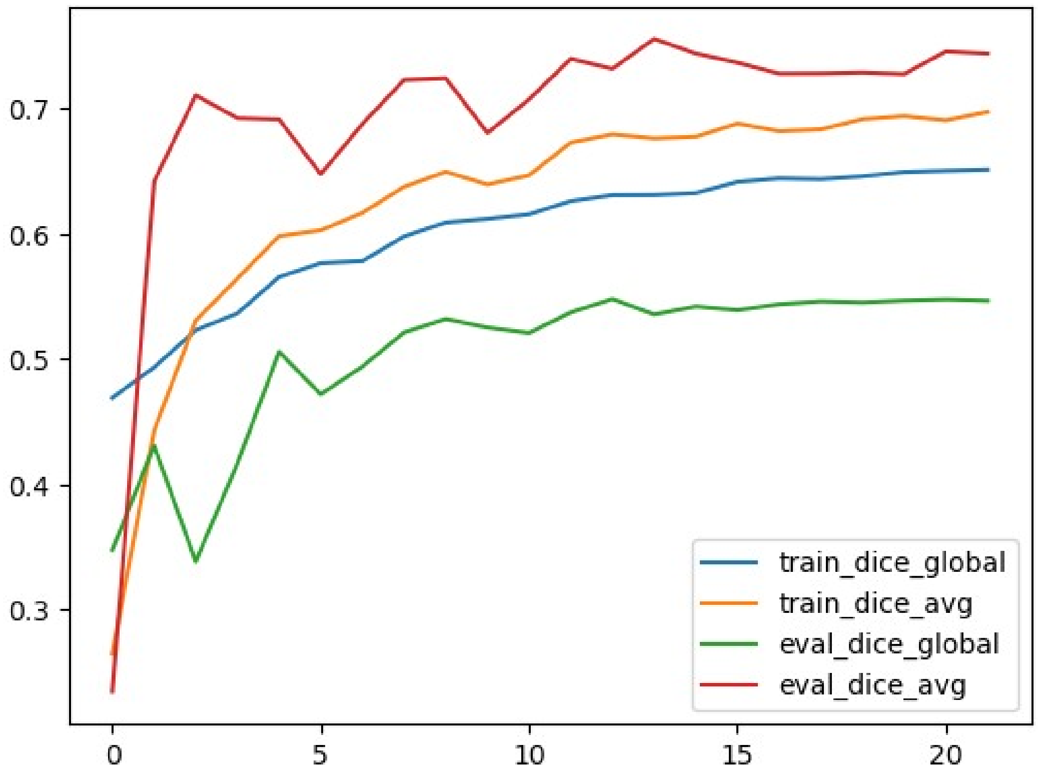
```
{"model_id":"caee93524c854508a59ceeb099aebf9c","version_major":2,"vers
ion_minor":0}

{"model_id":"83504e6f210e48b6afd51e97b4e4e3cd","version_major":2,"vers
ion_minor":0}

learning rate: 0.00125
current model saved! Epoch:9 global dice: 0.5318732762661085 avg dice:
0.7238966637644274
----------epoch: 10----------

{"model_id":"01138765d6274e5bbd70d213010f92b8","version_major":2,"vers
ion_minor":0}

{"model_id":"5239dca1d1fc4958a844379803beee8d","version_major":2,"vers
ion_minor":0}

learning rate: 0.00125
----------epoch: 11----------

{"model_id":"144487d97eed4294bb0233920a6defd3","version_major":2,"vers
ion_minor":0}

{"model_id":"d2d14929573840bca1709cc7d7f71540","version_major":2,"vers
ion_minor":0}

Epoch 00011: reducing learning rate of group 0 to 6.2500e-04.
learning rate: 0.000625
----------epoch: 12----------

{"model_id":"5941397f73d2429ab2df15cbe91c5db3","version_major":2,"vers
ion_minor":0}

{"model_id":"6415651f84eb463ea5fc34eac37b8d8c","version_major":2,"vers
ion_minor":0}

learning rate: 0.000625
current model saved! Epoch:12 global dice: 0.5371686486032539 avg
dice: 0.7394045072382894
----------epoch: 13----------

{"model_id":"798c9cf517884442bd6ea221bf356aac","version_major":2,"vers
ion_minor":0}

{"model_id":"4d1e89c2a1244f62972e3a01cd7eeac7","version_major":2,"vers
ion_minor":0}

learning rate: 0.000625
current model saved! Epoch:13 global dice: 0.5477490832623391 avg
dice: 0.731485547690556
----------epoch: 14----------
```

```
{"model_id":"74c1cc66674c44999ede1fad90a404c9","version_major":2,"vers
ion_minor":0}

{"model_id":"0517aa66ede74cc58fd2475c7b4dcb33","version_major":2,"vers
ion_minor":0}

learning rate: 0.000625
----------epoch: 15----------

{"model_id":"601df206519a4a51b3bd013376986586","version_major":2,"vers
ion_minor":0}

{"model_id":"4b399e4b831a42f4afbb21a26efa2803","version_major":2,"vers
ion_minor":0}

Epoch 00015: reducing learning rate of group 0 to 3.1250e-04.
learning rate: 0.0003125
----------epoch: 16----------

{"model_id":"ed88643b823347f08ced240568433b42","version_major":2,"vers
ion_minor":0}

{"model_id":"5319a5653af3425ba5c7be30a67efa6f","version_major":2,"vers
ion_minor":0}

learning rate: 0.0003125
----------epoch: 17----------

{"model_id":"257cc12db5c04ecfaa5fe498d0516718","version_major":2,"vers
ion_minor":0}

{"model_id":"df40e416e2d54711aa6bc8ccfcf1eb71","version_major":2,"vers
ion_minor":0}

Epoch 00017: reducing learning rate of group 0 to 1.5625e-04.
learning rate: 0.00015625
----------epoch: 18----------

{"model_id":"087de1785aab411f97f19fbca649c507","version_major":2,"vers
ion_minor":0}

{"model_id":"08f37ddb25f2483fa9d706352cad203d","version_major":2,"vers
ion_minor":0}

learning rate: 0.00015625
----------epoch: 19----------

{"model_id":"18257981e491486a812999b47d0392f8","version_major":2,"vers
ion_minor":0}

{"model_id":"6d8d934280304c298c243532126c8229","version_major":2,"vers
ion_minor":0}
```

```
Epoch 00019: reducing learning rate of group 0 to 7.8125e-05.
learning rate: 7.8125e-05
----------epoch: 20----------

{"model_id":"c2054cecc7384192a039b38179f7c5b3","version_major":2,"vers
ion_minor":0}

{"model_id":"538ccdd5ad324914bb75f0093a1d7b55","version_major":2,"vers
ion_minor":0}

learning rate: 7.8125e-05
----------epoch: 21----------

{"model_id":"1744eeff7fd34e5fa6f1bdc27ac79520","version_major":2,"vers
ion_minor":0}

{"model_id":"c8685343809a46999c5b30cf1f632fca","version_major":2,"vers
ion_minor":0}

Epoch 00021: reducing learning rate of group 0 to 3.9063e-05.
learning rate: 3.90625e-05
----------epoch: 22----------

{"model_id":"dde990bc49344caaa65082660ddefc0c","version_major":2,"vers
ion_minor":0}

{"model_id":"4af20d80a05941aaaf6d588038598110","version_major":2,"vers
ion_minor":0}

learning rate: 3.90625e-05
----------epoch: 23----------
early stopping in epoch 23
```

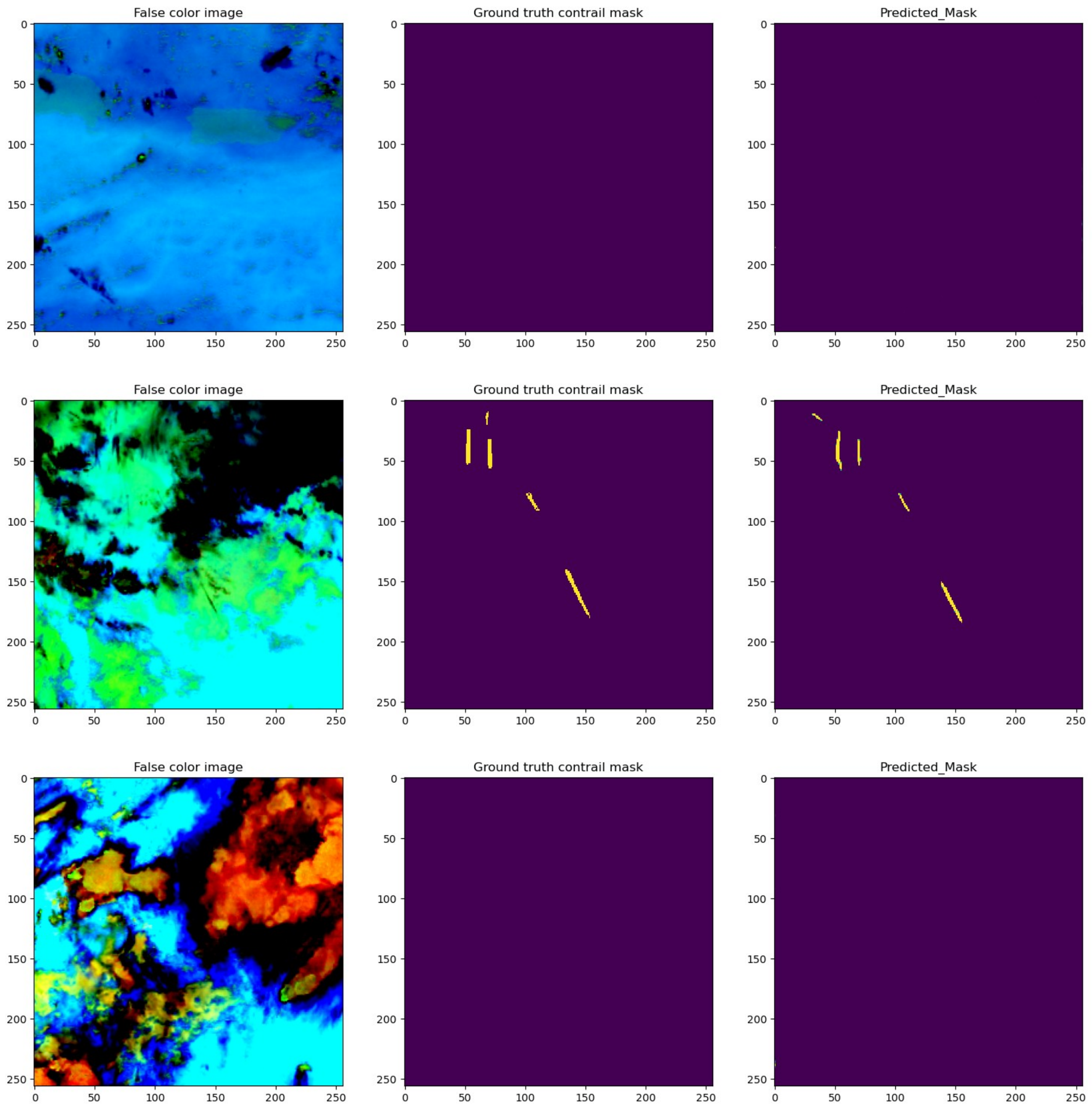
Training and Validation History

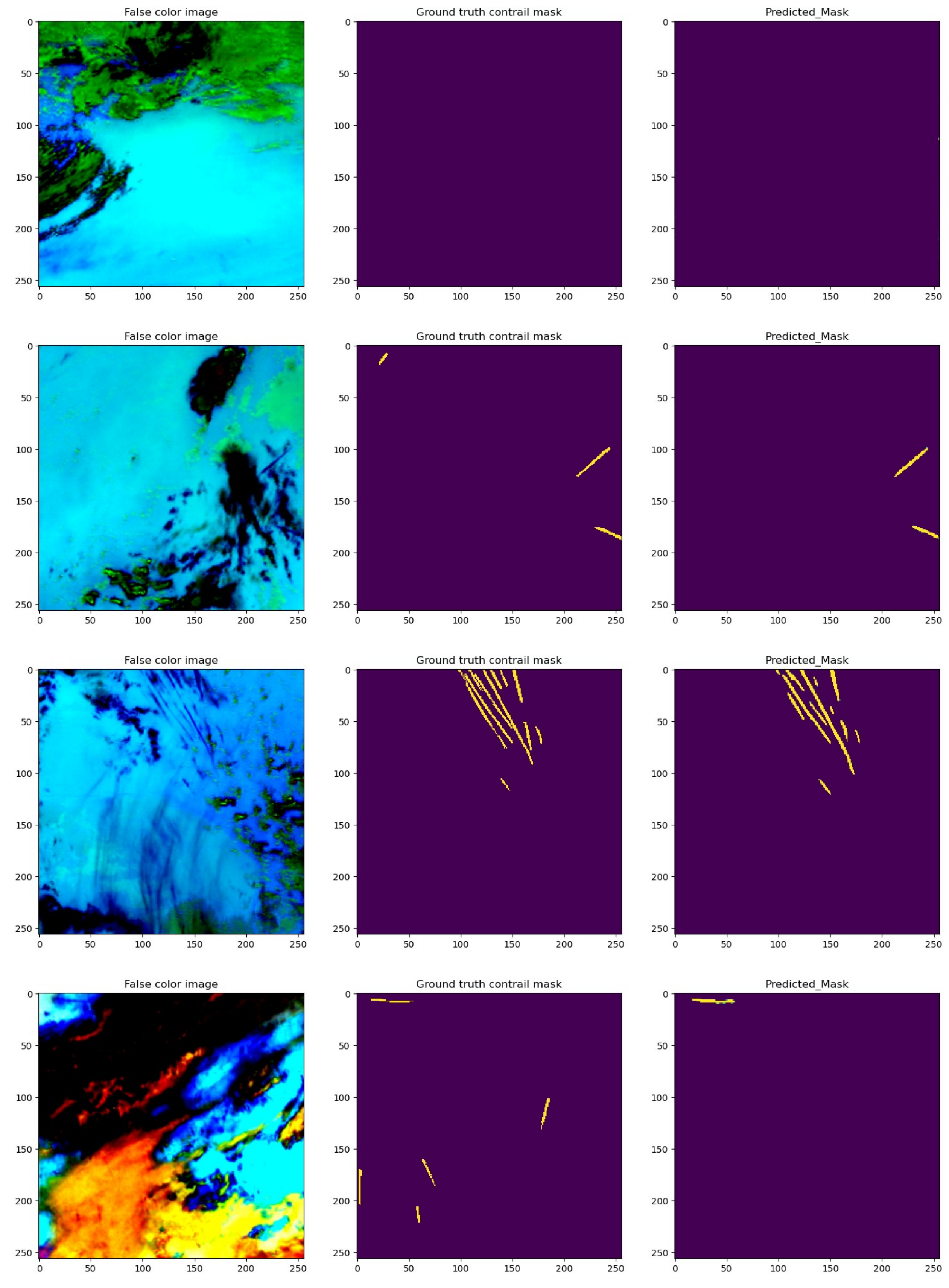
```
plt.plot(train_dice_global, label = 'train_dice_global')
plt.plot(train_dice_avg, label = 'train_dice_avg')
plt.plot(eval_dice_global, label = 'eval_dice_global')
plt.plot(eval_dice_avg, label = 'eval_dice_avg')
plt.legend()
plt.show

<function matplotlib.pyplot.show(close=None, block=None)>
```

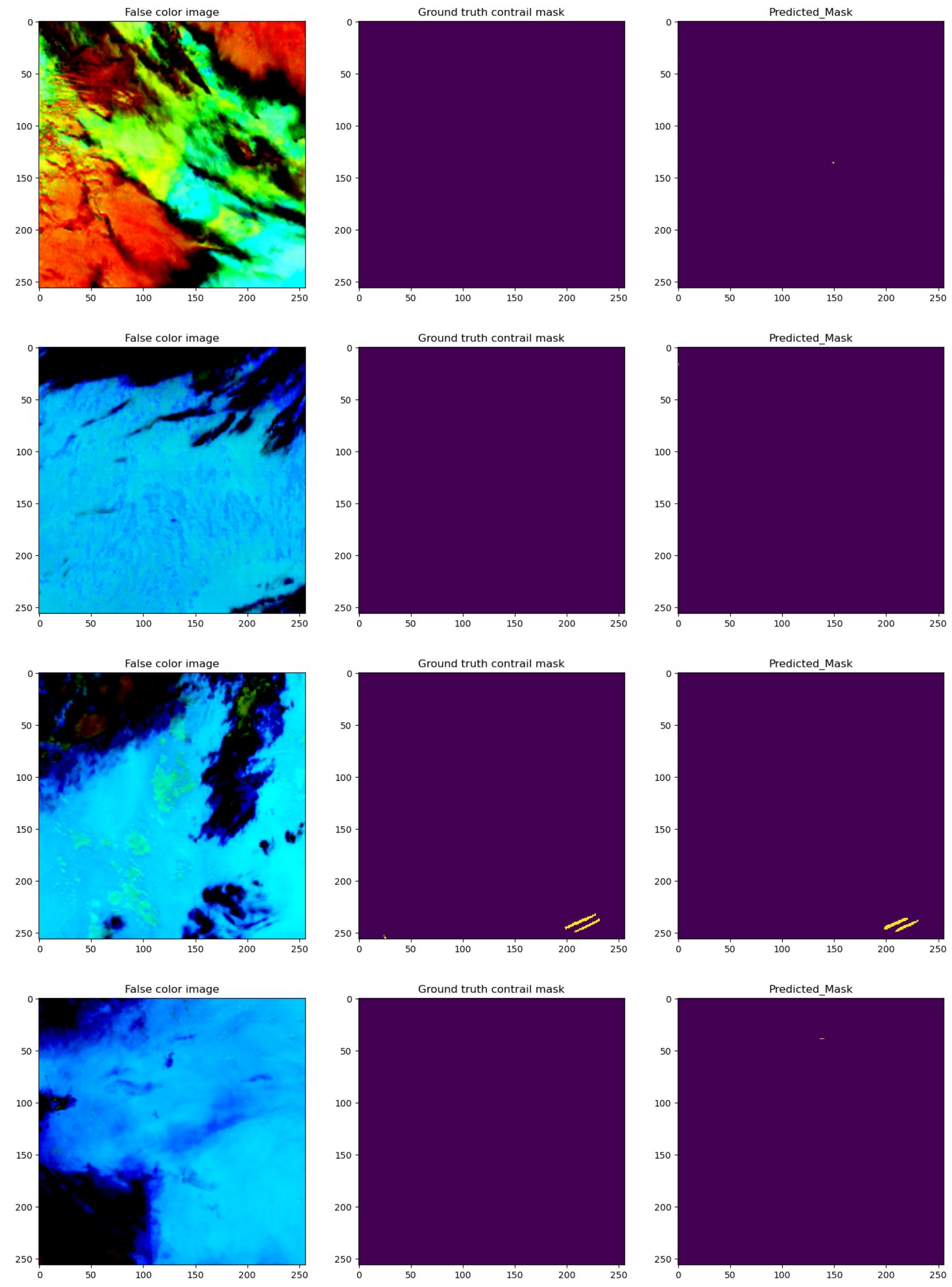


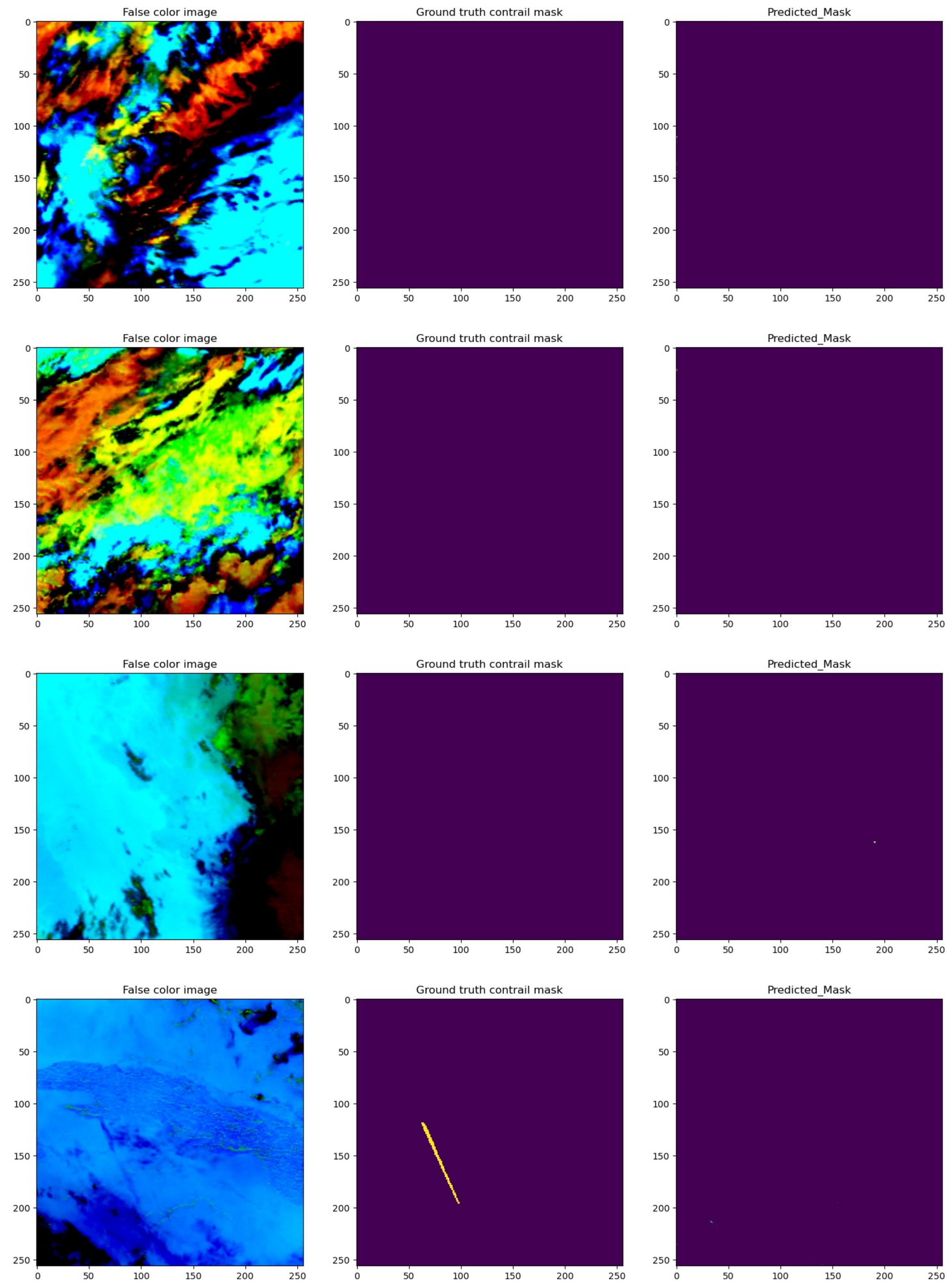
Show Some Predictions for the Validation Dataset

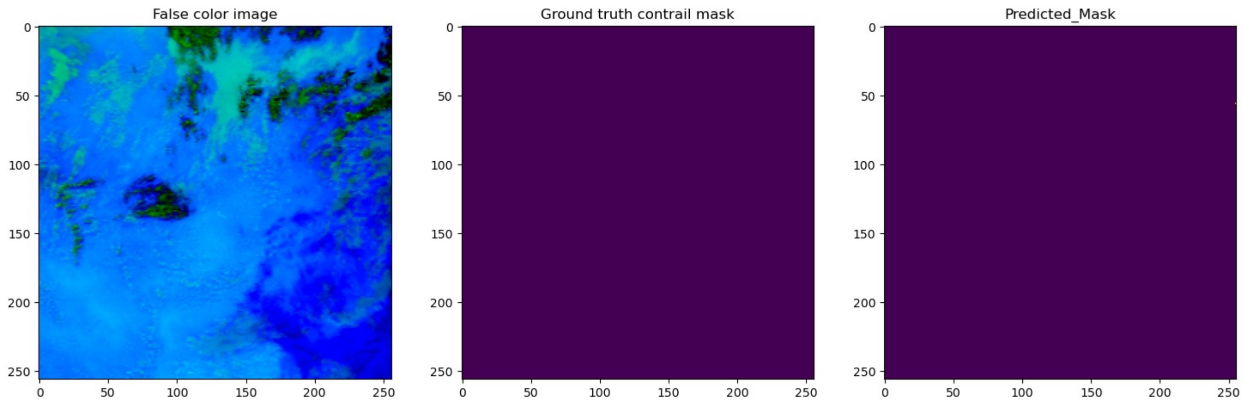
```
image, mask = next(iter(validation_dataloader))

image, mask = image.to(device), mask.to(device)
pred_mask = model(image)

image = torch.moveaxis(image, 1, -1)
mask = torch.moveaxis(mask, 1, -1)
pred_mask = torch.moveaxis(pred_mask, 1, -1)

image, mask, pred_mask = image.cpu(), mask.cpu(),
pred_mask.detach().cpu()

for i in range(Config.batch_size):
    plt.figure(figsize=(18, 6))

    ax = plt.subplot(1, 3, 1)
    ax.imshow(image[i])
    ax.set_title('False color image')

    ax = plt.subplot(1, 3, 2)
    ax.imshow(mask[i], interpolation = 'none')
    ax.set_title('Ground truth contrail mask')

    ax = plt.subplot(1, 3, 3)
    ax.imshow(pred_mask[i], interpolation = 'none')
    ax.set_title('Predicted_Mask')
```




False color image
Ground truth contrail mask
Predicted_Mask
False color image
Ground truth contrail mask
Predicted_Mask
False color image
Ground truth contrail mask
Predicted_Mask
False color image
Ground truth contrail mask
Predicted_Mask
0
50
100
150
200
250


False color image
Ground truth contrail mask
Predicted_Mask
False color image
Ground truth contrail mask
Predicted_Mask
False color image
Ground truth contrail mask
Predicted_Mask
False color image
Ground truth contrail mask
Predicted_Mask


False color image
Ground truth contrail mask
Predicted_Mask
False color image
Ground truth contrail mask
Predicted_Mask
False color image
Ground truth contrail mask
Predicted_Mask
False color image
Ground truth contrail mask
Predicted_Mask
0
50
100
150
200
250



Run-Length Code

```
def rle_encode(x, fg_val=1):
    """
    Args:
        x:  numpy array of shape (height, width), 1 - mask, 0 -
background
    Returns: run length encoding as list
    """

    dots = np.where(
        x.T.flatten() == fg_val)[0]  # .T sets Fortran order down-
then-right
    run_lengths = []
    prev = -2
    for b in dots:
        if b > prev + 1:
            run_lengths.extend((b + 1, 0))
        run_lengths[-1] += 1
        prev = b
    return run_lengths


def list_to_string(x):
    """
    Converts list to a string representation
    Empty list returns '-'
    """
    if x: # non-empty list
        s = str(x).replace("[", "").replace("]", "").replace(",", "")
    else:
        s = '-'
    return s


def rle_decode(mask_rle, shape=(256, 256)):
    '''
    mask_rle: run-length as string formatted (start length)
              empty predictions need to be encoded with '-'
    shape: (height, width) of array to return
    Returns numpy array, 1 - mask, 0 - background
    '''

    img = np.zeros(shape[0]*shape[1], dtype=np.uint8)
    if mask_rle != '-':
        s = mask_rle.split()
        starts, lengths = [np.asarray(x, dtype=int) for x in (s[0:]
[::2], s[1:][::2])]
        starts -= 1
        ends = starts + lengths
        for lo, hi in zip(starts, ends):
            img[lo:hi] = 1
    return img.reshape(shape, order='F')  # Needed to align to RLE
direction
```

Create a Naive Submission

```
test_recs = os.listdir(Config.test_path)
# print(test_recs)

test_data = ContrailDataset(data_dir = Config.test_path, mode='test')
test_dataloader = DataLoader(test_data,
                             batch_size = Config.batch_size,
                             shuffle = False,
                             num_workers = 2 if
torch.cuda.is_available() else 0,
                             pin_memory = True,
                             drop_last = False)
print('test_dataloader:',test_dataloader)
submission = pd.read_csv('/kaggle/input/google-research-identify-
contrails-reduce-global-warming/sample_submission.csv',
                         index_col='record_id')

model.eval()
fails = []

with torch.no_grad():
    for X, rec in test_dataloader:
        mask = np.zeros((256,256))
        try:
            print('X:',X.shape)
            X = X.to(device)
    #         print(model(X))
    #         pred = (model(X)['out']).cpu().detach().numpy().copy()
```

```
[0,0,:,:]
            pred = model(X).cpu().numpy().copy()[0,0,:,:]
            # if the image was resized, use a resizing step to make
256 again
            if Config.resize:
                pred = torch.nn.functional.interpolate(pred,
                                                       size = 256,
                                                       mode =
'bilinear')
            mask[pred < 0.5] = 0
            mask[pred > 0.5] = 1
        except Exception as e:
            fails.append(e)
            continue

        submission.loc[int(rec[0]), 'encoded_pixels'] =
list_to_string(rle_encode(mask))

submission.head()
```

```
test_dataloader: <torch.utils.data.dataloader.DataLoader object at
0x7e2dfc211a20>
X: torch.Size([2, 3, 256, 256])
```

```
                    encoded_pixels
record_id
1000834164244036115       1 3 10 5
1002653297254493116              -
```

```
submission.to_csv('submission.csv')
```

```
import IPython.display as ipd
```

```
audio_path="/kaggle/input/music-notification-rome-legion/rome-legion-
62972.mp3"
```

```
ipd.Audio(audio_path, autoplay=True)
```

```
<IPython.lib.display.Audio object>
```