```
pip install tensorflow keras matplotlib pandas seaborn scikit-learn
opencv-python

Requirement already satisfied: tensorflow in c:\users\retheck\
anaconda3\lib\site-packages (2.16.1)
Requirement already satisfied: keras in c:\users\retheck\anaconda3\
lib\site-packages (3.3.3)
Requirement already satisfied: matplotlib in c:\users\retheck\
anaconda3\lib\site-packages (3.7.2)
Requirement already satisfied: pandas in c:\users\retheck\anaconda3\
lib\site-packages (2.0.3)
Requirement already satisfied: seaborn in c:\users\retheck\anaconda3\
lib\site-packages (0.12.2)
Requirement already satisfied: scikit-learn in c:\users\retheck\
anaconda3\lib\site-packages (1.4.2)
Requirement already satisfied: opencv-python in c:\users\retheck\
appdata\roaming\python\python311\site-packages (4.10.0.84)
Requirement already satisfied: tensorflow-intel==2.16.1 in c:\users\
retheck\anaconda3\lib\site-packages (from tensorflow) (2.16.1)
Requirement already satisfied: absl-py>=1.0.0 in c:\users\retheck\
anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (2.1.0)
Requirement already satisfied: astunparse>=1.6.0 in c:\users\retheck\
anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=23.5.26 in c:\users\
retheck\anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (24.3.25)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in
c:\users\retheck\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (0.5.4)
Requirement already satisfied: google-pasta>=0.1.1 in c:\users\
retheck\anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (0.2.0)
Requirement already satisfied: h5py>=3.10.0 in c:\users\retheck\
anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (3.11.0)
Requirement already satisfied: libclang>=13.0.0 in c:\users\retheck\
anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (18.1.1)
Requirement already satisfied: ml-dtypes~=0.3.1 in c:\users\retheck\
anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (0.3.2)
Requirement already satisfied: opt-einsum>=2.3.2 in c:\users\retheck\
anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (3.3.0)
Requirement already satisfied: packaging in c:\users\retheck\
anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (23.1)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!
```

```
=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3 in c:\users\retheck\
anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (4.25.3)
Requirement already satisfied: requests<3,>=2.21.0 in c:\users\
retheck\anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (2.32.3)
Requirement already satisfied: setuptools in c:\users\retheck\
anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (75.5.0)
Requirement already satisfied: six>=1.12.0 in c:\users\retheck\
anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\retheck\
anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (2.4.0)
Requirement already satisfied: typing-extensions>=3.6.6 in c:\users\
retheck\anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (4.12.2)
Requirement already satisfied: wrapt>=1.11.0 in c:\users\retheck\
anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (1.14.1)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in c:\users\
retheck\anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (1.63.0)
Requirement already satisfied: tensorboard<2.17,>=2.16 in c:\users\
retheck\anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (2.16.2)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
c:\users\retheck\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (0.31.0)
Requirement already satisfied: numpy<2.0.0,>=1.23.5 in c:\users\
retheck\anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (1.24.3)
Requirement already satisfied: rich in c:\users\retheck\anaconda3\lib\
site-packages (from keras) (13.7.1)
Requirement already satisfied: namex in c:\users\retheck\anaconda3\
lib\site-packages (from keras) (0.0.8)
Requirement already satisfied: optree in c:\users\retheck\anaconda3\
lib\site-packages (from keras) (0.11.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\retheck\
anaconda3\lib\site-packages (from matplotlib) (1.0.5)
Requirement already satisfied: cycler>=0.10 in c:\users\retheck\
anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\retheck\
anaconda3\lib\site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\retheck\
anaconda3\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: pillow>=6.2.0 in c:\users\retheck\
anaconda3\lib\site-packages (from matplotlib) (10.4.0)
```

```
Requirement already satisfied: pyparsing<3.1,>=2.3.1 in c:\users\
retheck\anaconda3\lib\site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\
retheck\anaconda3\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\retheck\
anaconda3\lib\site-packages (from pandas) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\retheck\
anaconda3\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: scipy>=1.6.0 in c:\users\retheck\
anaconda3\lib\site-packages (from scikit-learn) (1.11.1)
Requirement already satisfied: joblib>=1.2.0 in c:\users\retheck\
anaconda3\lib\site-packages (from scikit-learn) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\
retheck\anaconda3\lib\site-packages (from scikit-learn) (2.2.0)
Requirement already satisfied: markdown-it-py>=2.2.0 in c:\users\
retheck\anaconda3\lib\site-packages (from rich->keras) (2.2.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\users\
retheck\anaconda3\lib\site-packages (from rich->keras) (2.15.1)
Requirement already satisfied: wheel<1.0,>=0.23.0 in c:\users\retheck\
anaconda3\lib\site-packages (from astunparse>=1.6.0->tensorflow-
intel==2.16.1->tensorflow) (0.45.0)
Requirement already satisfied: mdurl~=0.1 in c:\users\retheck\
anaconda3\lib\site-packages (from markdown-it-py>=2.2.0->rich->keras)
(0.1.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\
retheck\anaconda3\lib\site-packages (from requests<3,>=2.21.0-
>tensorflow-intel==2.16.1->tensorflow) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\retheck\
anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorflow-
intel==2.16.1->tensorflow) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\retheck\
anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorflow-
intel==2.16.1->tensorflow) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\retheck\
anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorflow-
intel==2.16.1->tensorflow) (2024.2.2)
Requirement already satisfied: markdown>=2.6.8 in c:\users\retheck\
anaconda3\lib\site-packages (from tensorboard<2.17,>=2.16->tensorflow-
intel==2.16.1->tensorflow) (3.4.1)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0
in c:\users\retheck\anaconda3\lib\site-packages (from
tensorboard<2.17,>=2.16->tensorflow-intel==2.16.1->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in c:\users\retheck\
anaconda3\lib\site-packages (from tensorboard<2.17,>=2.16->tensorflow-
intel==2.16.1->tensorflow) (2.2.3)
Requirement already satisfied: MarkupSafe>=2.1.1 in c:\users\retheck\
anaconda3\lib\site-packages (from werkzeug>=1.0.1-
>tensorboard<2.17,>=2.16->tensorflow-intel==2.16.1->tensorflow)
```

```
(2.1.1)
Note: you may need to restart the kernel to use updated packages.
```

Loading the Dataset with Deep Lake

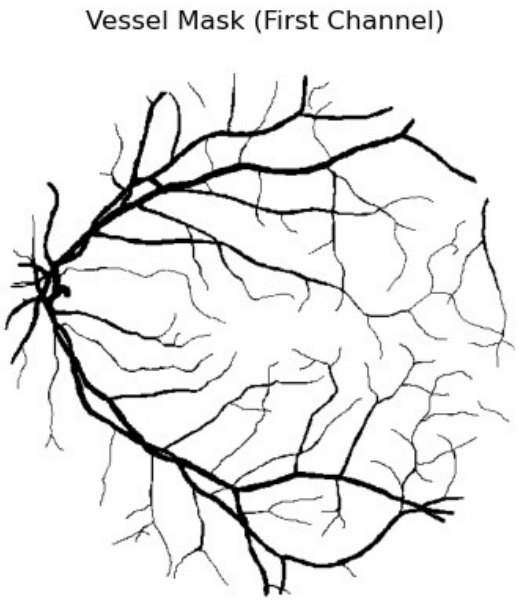
```
import deeplake

# Load the training subset
train_ds = deeplake.load("hub://activeloop/drive-train")

# Load the testing subset
test_ds = deeplake.load("hub://activeloop/drive-test")

# Inspect the dataset
print("Training Dataset Structure:")
print(train_ds.summary())

print("\nTesting Dataset Structure:")
print(test_ds.summary())
```

```
-

Opening dataset in read-only mode as you don't have write permissions.

\

This dataset can be visualized in Jupyter Notebook by ds.visualize()
or at https://app.activeloop.ai/activeloop/drive-train


|

hub://activeloop/drive-train loaded successfully.



Opening dataset in read-only mode as you don't have write permissions.

-

This dataset can be visualized in Jupyter Notebook by ds.visualize()
or at https://app.activeloop.ai/activeloop/drive-test


-

hub://activeloop/drive-test loaded successfully.

Training Dataset Structure:
Dataset(path='hub://activeloop/drive-train', read_only=True,
```

```
tensors=['rgb_images', 'manual_masks/mask', 'masks/mask'])

      tensor           htype            shape          dtype
compression
      -------         -------          -------        -------  -------
    rgb_images          image     (20, 584, 565, 3)   uint8    tiff
 manual_masks/mask  binary_mask  (20, 584, 565, 2)   bool      lz4
    masks/mask      binary_mask  (20, 584, 565, 2)   bool      lz4
None

Testing Dataset Structure:
Dataset(path='hub://activeloop/drive-test', read_only=True,
tensors=['rgb_images', 'masks'])

   tensor       htype            shape          dtype  compression
  -------      -------          -------        -------  -------
 rgb_images     image     (20, 584, 565, 3)   uint8    tiff
   masks     binary_mask  (20, 584, 565, 2)   bool      lz4
None
```

Visualizing Sample Images and Masks

```
# Selecting the first channel of the mask
single_channel_mask = mask[:, :, 0]

# Ploting the image and the single-channel mask
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.imshow(image)
plt.title("Retinal Image")
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(single_channel_mask, cmap='gray')
plt.title("Vessel Mask (First Channel)")
plt.axis('off')

plt.show()
```



PyTorch DataLoader for Training

```
from torch.utils.data import DataLoader

# Creating a PyTorch dataloader
train_loader = train_ds.pytorch(num_workers=0, batch_size=4,
shuffle=True)
test_loader = test_ds.pytorch(num_workers=0, batch_size=4,
shuffle=False)

# Example: Iterate over a batch
for batch in train_loader:
    images = batch['rgb_images']
    masks = batch['manual_masks/mask']
    print(f"Batch of images shape: {images.shape}")
    print(f"Batch of masks shape: {masks.shape}")
    break
```

```
Batch of images shape: torch.Size([4, 584, 565, 3])
Batch of masks shape: torch.Size([4, 584, 565, 2])
```

TensorFlow DataLoader for Training

```
# Iterate over a batch from the training DataLoader
for batch in train_tf_loader.take(1):  # Take one batch
    # Access the keys in the batch
    print("Batch keys:", batch.keys())
```

```
    # Extract images and masks
    images = batch['rgb_images']
    masks = batch['manual_masks/mask']

    # Printing shapes of the images and masks
    print(f"Image shape: {images.shape}")
    print(f"Mask shape: {masks.shape}")
```

```
Batch keys: dict_keys(['rgb_images', 'manual_masks/mask',
'masks/mask'])
Image shape: (584, 565, 3)
Mask shape: (584, 565, 2)
```

```
masks = masks[:, 0:1, :, :]  # Select the first channel, keeping the
dimensions


# Converting masks to integer (binary format) for logical OR
masks_binary = masks > 0.5  # If mask values are continuous, threshold
to binary
masks = (masks_binary[:, 0:1, :, :] | masks_binary[:,
1:2, :, :]).float()  # Combine channels
```

Training the model

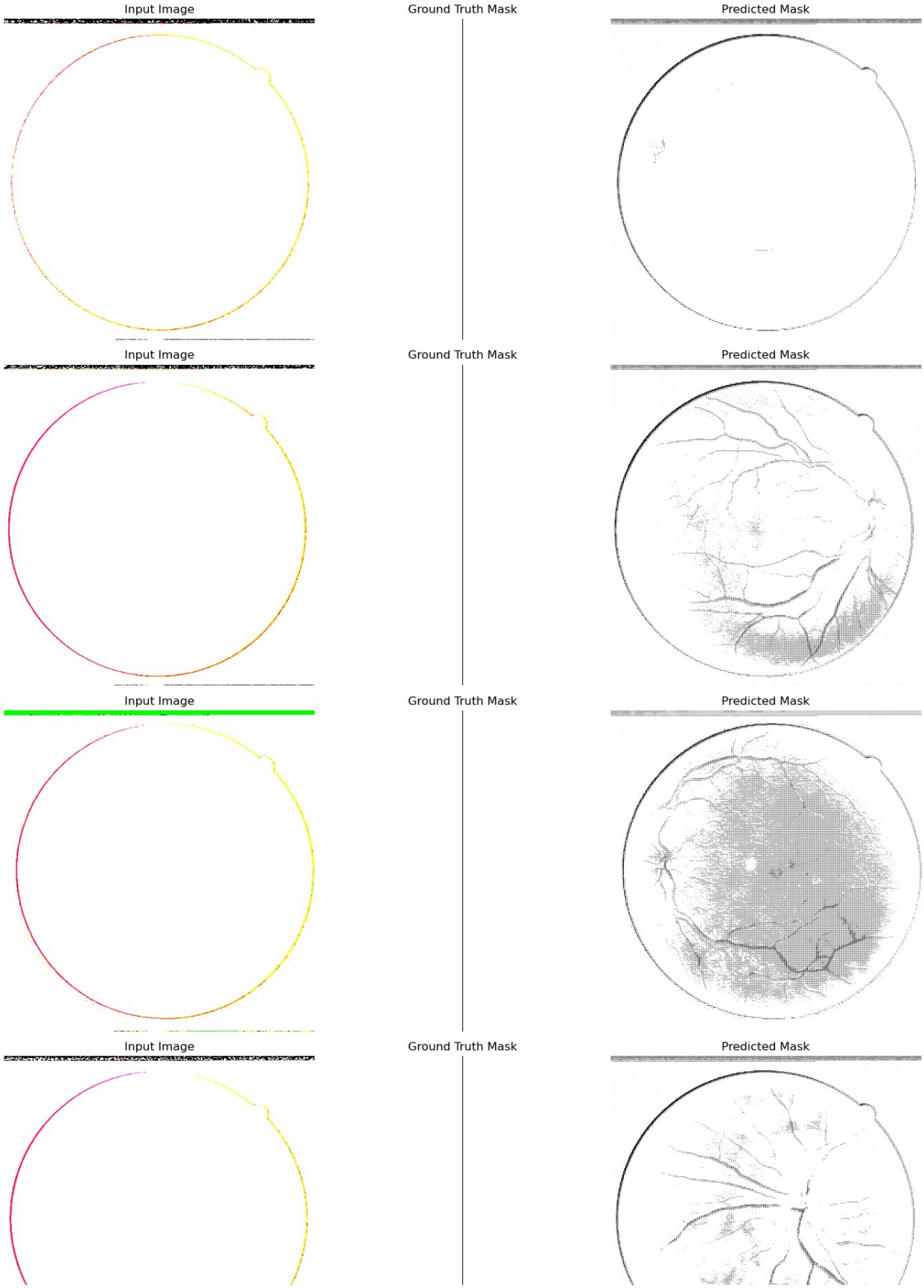
```
for batch in train_loader:
    images = batch['rgb_images'].to(device).float()
    masks = batch['manual_masks/mask'].to(device).float()

    # Reshaping and combine mask channels
    masks_binary = masks > 0.5  # Threshold masks to binary
    masks = (masks_binary[:, 0:1, :, :] | masks_binary[:,
1:2, :, :]).float()

    # Resizing masks to match output dimensions
    masks = F.interpolate(masks, size=(outputs.shape[2],
outputs.shape[3]), mode='nearest')

    # Forward pass
    outputs = model(images.permute(0, 3, 1, 2))  # Permute for PyTorch
format
    loss = criterion(outputs, masks)

    # Backward pass and optimization
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
```

```
    print(f"Loss: {loss.item():.4f}")
```

```
Loss: 31.0895
Loss: 31.0929
Loss: 29.9667
Loss: 29.8394
Loss: 31.7298
```

```
for batch in test_loader:  # Use the test loader
    images = batch['rgb_images'].to(device).float()

    # Correct mask key based on dataset
    ground_truth = batch['masks'].to(device).float()  # Use 'masks'
instead of 'manual_masks/mask'

    # Combine ground truth masks if necessary
    masks_binary = ground_truth > 0.5
    ground_truth = (masks_binary[:, 0:1, :, :] | masks_binary[:,
1:2, :, :]).float()

    # Predict masks
    predictions = model(images.permute(0, 3, 1, 2))  # Permute for
PyTorch format

    # Visualize the first few samples
    visualize_predictions(images.permute(0, 3, 1, 2), ground_truth,
predictions, num_samples=4)
    break  # Visualize one batch
```

```
Clipping input data to the valid range for imshow with RGB data
([0..1] for floats or [0..255] for integers).
Clipping input data to the valid range for imshow with RGB data
([0..1] for floats or [0..255] for integers).
Clipping input data to the valid range for imshow with RGB data
([0..1] for floats or [0..255] for integers).
Clipping input data to the valid range for imshow with RGB data
([0..1] for floats or [0..255] for integers).
```


Input Image
Ground Truth Mask
Predicted Mask
Input Image
Ground Truth Mask
Predicted Mask
Input Image
Ground Truth Mask
Predicted Mask
Input Image
Ground Truth Mask
Predicted Mask