In [ ]:

```
# Import Statements
import os, sys
import numpy as np
import matplotlib.pyplot as plt
import cv2
from tqdm import tqdm, trange
import unet_dense
```

In [ ]:

```
# Best architecture as reported was using the DenseNet + UNet
!pwd
```

In [ ]:

```
os.chdir('/home/jjonathanmak/cs271proj/unet_dense')
```

In [ ]:

```
#Experiment 2 uses geometric shapes
!python train.py --model densenet --expname geometric_ --bs 8 --useGPU True --epochs 20 --dataset /home/jjonathanmak/cs271proj/Semantic_Segmentation_Dataset/
# !/home/jjonathanmak/cs271proj/unet_dense/train.py --model densenet --expname geometric_ --bs 8 --useGPU True --dataset Semantic_Segmentation_Dataset/
```

In [ ]:

```
!python test.py --model densenet --load best_model.pkl --bs 4 --dataset /home/jjonathanmak/cs271proj/Semantic_Segmentation_Dataset
```

In [ ]:

```
test_dir = '/home/jjonathanmak/cs271proj/Semantic_Segmentation_Dataset/test/images'
for im in os.listdir(test_dir):
    curr = test_dir+'/'+im
    img = cv2.imread(curr)
#     img = cv2.bilateralFilter(img, 15, 75, 75)
#     clahe = cv2.createCLAHE(clipLimit=1.5, tileGridSize=(8,8))
    # CLAHE across all 3 channels
#     img[:, :, 0] = clahe.apply(img[:, :, 0])
#     img[:, :, 1]= clahe.apply(img[:, :, 1])
#     img[:, :, 2] = clahe.apply(img[:, :, 2])
    plt.imshow(img)
    cv2.imwrite("/home/jjonathanmak/cs271proj/figs/original_eds.png", img)
    break
```

In [ ]:

```
keratitis_dir = '/home/jjonathanmak/cs271proj/images_raw/'
keratitis_debug_dir = '/home/jjonathanmak/cs271proj/images_raw/bacterial/04'
```

In [ ]:

```
for im in os.listdir(keratitis_debug_dir):
    curr = keratitis_debug_dir+'/'+im
    img = cv2.imread(curr)
    scale_percent = (400/3264) * 100
    width = int(img.shape[1] * scale_percent / 100)
    height = int(img.shape[0] * scale_percent / 100)
    dim = (width, height)
    img = cv2.resize(img, dim, interpolation = cv2.INTER_AREA)
    delta_h = 640 - img.shape[0]
    top, bottom = delta_h//2, delta_h-(delta_h//2)
    color = [0, 0, 0]
    img = cv2.copyMakeBorder(img, top, bottom, 0, 0, cv2.BORDER_CONSTANT
,
        value=color)
#     img = cv2.bilateralFilter(img, 15, 75, 75)
    clahe = cv2.createCLAHE(clipLimit=1.5, tileGridSize=(8,8))
    # CLAHE across all 3 channels
    img[:, :, 0] = clahe.apply(img[:, :, 0])
    img[:, :, 1]= clahe.apply(img[:, :, 1])
    img[:, :, 2] = clahe.apply(img[:, :, 2])
    plt.imshow(img)
    cv2.imwrite('/home/jjonathanmak/cs271proj/figs/clahe_keratitis.png',
img)
    break
```

In [ ]:

```
def preprocess(filepath, bilateral=False, equalize=False):
    labels = ['bacterial', 'fungal', 'viral']

    with open('/home/jjonathanmak/cs271proj/labels.csv', 'w') as f:
        for l in trange(len(labels)):
            label = labels[l]
            patients = os.listdir(filepath+'/'+label)
            for p in patients:
                img_name = os.listdir(os.path.join(filepath+'/'+label,p))[0]
                image_path = os.path.join(filepath+'/'+label, p, img_name)
                img = cv2.imread(os.path.join(filepath+'/'+label, p, img_name))
                scale_percent = (400/3264) * 100
                width = int(img.shape[1] * scale_percent / 100)
                height = int(img.shape[0] * scale_percent / 100)
                dim = (width, height)
                img = cv2.resize(img, dim, interpolation = cv2.INTER_AREA)
                delta_h = 640 - img.shape[0]
                top, bottom = delta_h//2, delta_h-(delta_h//2)
                color = [0, 0, 0]
                img = cv2.copyMakeBorder(img, top, bottom, 0, 0, cv2.BORDER_CONSTANT,
                    value=color)
                if bilateral:
                    img = cv2.bilateralFilter(img, 15, 75, 75)
                if equalize:
                    clahe = cv2.createCLAHE(clipLimit=1.5, tileGridSize=(8,8))
                    # CLAHE across all 3 channels
                    img[:, :, 0] = clahe.apply(img[:, :, 0])
                    img[:, :, 1]= clahe.apply(img[:, :, 1])
                    img[:, :, 2] = clahe.apply(img[:, :, 2])
                plt.imshow(img)
                break
                cv2.imwrite(os.path.join('/home/jjonathanmak/cs271proj/Semantic_Segmentation_Dataset/test/images', p + '.png'), img)
                f.write('%s.jpg,%s\n' % (p, l))
```

In [ ]:

```
preprocess(keratitis_dir, True, True)
```

In [ ]:

```
# test on EDS and keratitis Baseline
!python test.py --model densenet --load best_model.pkl --bs 4 --save keratitis5 --dataset /home/jjonathanmak/cs271proj/Semantic_Segmentation_Dataset
```

In [ ]:

```
# Train Geometric
!python train.py --model densenet --expname geometric_2 --bs 8 --useGPU True --epochs 20 --dataset /home/jjonathanmak/cs271proj/Semantic_Segmentation_Dataset/ --savemodel
```

In [ ]:

```
# Test Geometric
!python test.py --model densenet --load /home/jjonathanmak/cs271proj/unet_dense/logs/geometric_2/models/dense_net19.pkl --bs 4 --save keratitis6 --dataset /home/jjonathanmak/cs271proj/Semantic_Segmentation_Dataset
```

In [ ]:

```
# Train Bilateral Filtering
!python train.py --model densenet --expname bilateral_1 --bs 8 --useGPU True --epochs 20 --dataset /home/jjonathanmak/cs271proj/Semantic_Segmentation_Dataset/ --savemodel
```

In [ ]:

```
# Test Bilateral Filtering
!python test.py --model densenet --load /home/jjonathanmak/cs271proj/unet_dense/logs/bilateral_1/models/dense_net19.pkl --bs 4 --save keratitis7 --dataset /home/jjonathanmak/cs271proj/Semantic_Segmentation_Dataset
```

In [ ]:

```
# Train CLAHE
!python train.py --model densenet --expname clahe_1 --bs 8 --useGPU True --epochs 20 --dataset /home/jjonathanmak/cs271proj/Semantic_Segmentation_Dataset/ --savemodel
```

In [ ]:

```
# Test CLAHE
!python test.py --model densenet --load /home/jjonathanmak/cs271proj/unet_dense/logs/clahe_1/models/dense_net19.pkl --bs 4 --save keratitis8 --dataset /home/jjonathanmak/cs271proj/Semantic_Segmentation_Dataset
```

In [ ]:

```
# All 3
!python train.py --model densenet --expname combined_1 --bs 8 --useGPU True --epochs 20 --dataset /home/jjonathanmak/cs271proj/Semantic_Segmentation_Dataset/ --savemodel
```

In [ ]:

```
# Test all 3
!python test.py --model densenet --load /home/jjonathanmak/cs271proj/unet_dense/logs/combined_1/models/dense_net19.pkl --bs 4 --save keratitis10 --dataset /home/jjonathanmak/cs271proj/Semantic_Segmentation_Dataset
```

In [ ]:

```
# Random Masks + All 3
!python train.py --model densenet --expname combined_2 --bs 8 --useGPU True --epochs 20 --dataset /home/jjonathanmak/cs271proj/Semantic_Segmentation_Dataset/ --savemodel
```

In [ ]:

```
# Test all 3 + Random Masks
!python test.py --model densenet --load /home/jjonathanmak/cs271proj/unet_dense/logs/combined_1/models/dense_net18.pkl --bs 4 --save keratitis11 --dataset /home/jjonathanmak/cs271proj/Semantic_Segmentation_Dataset
```