



Tecnológico de Monterrey

Instituto Tecnológico y de Estudios Superiores de Monterrey

Procesamiento de Imágenes Médicas para el Diagnóstico (Grupo 101)

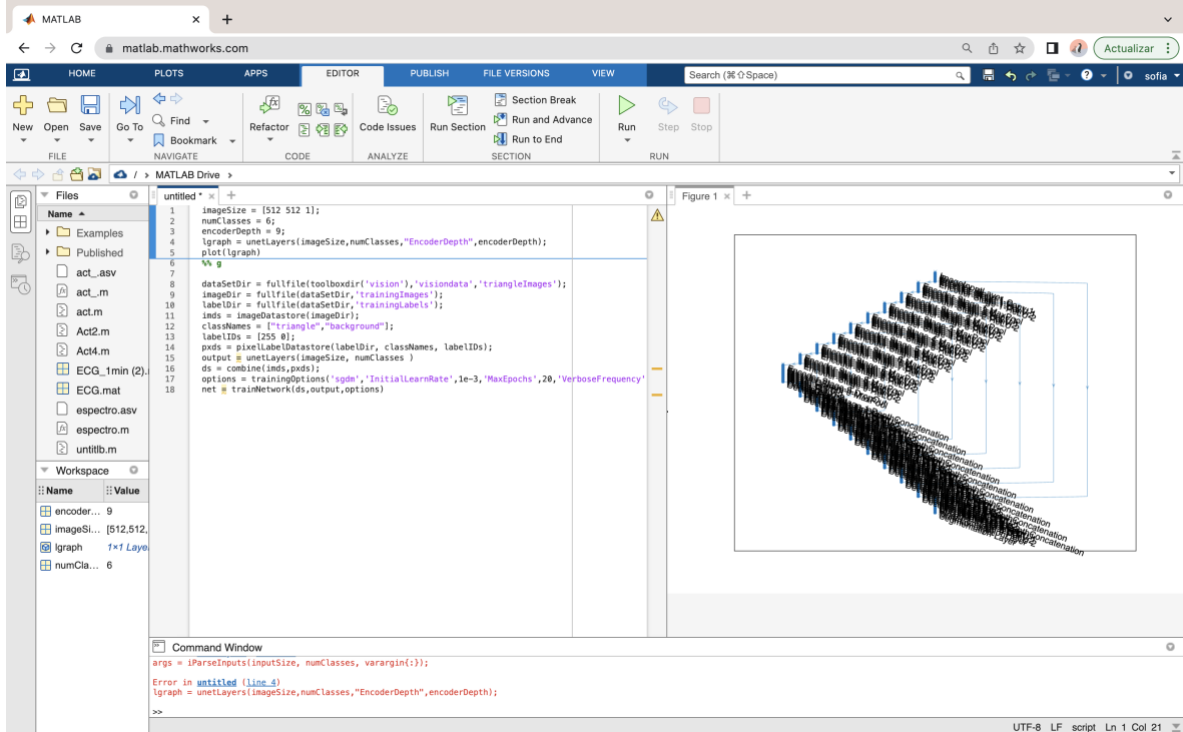
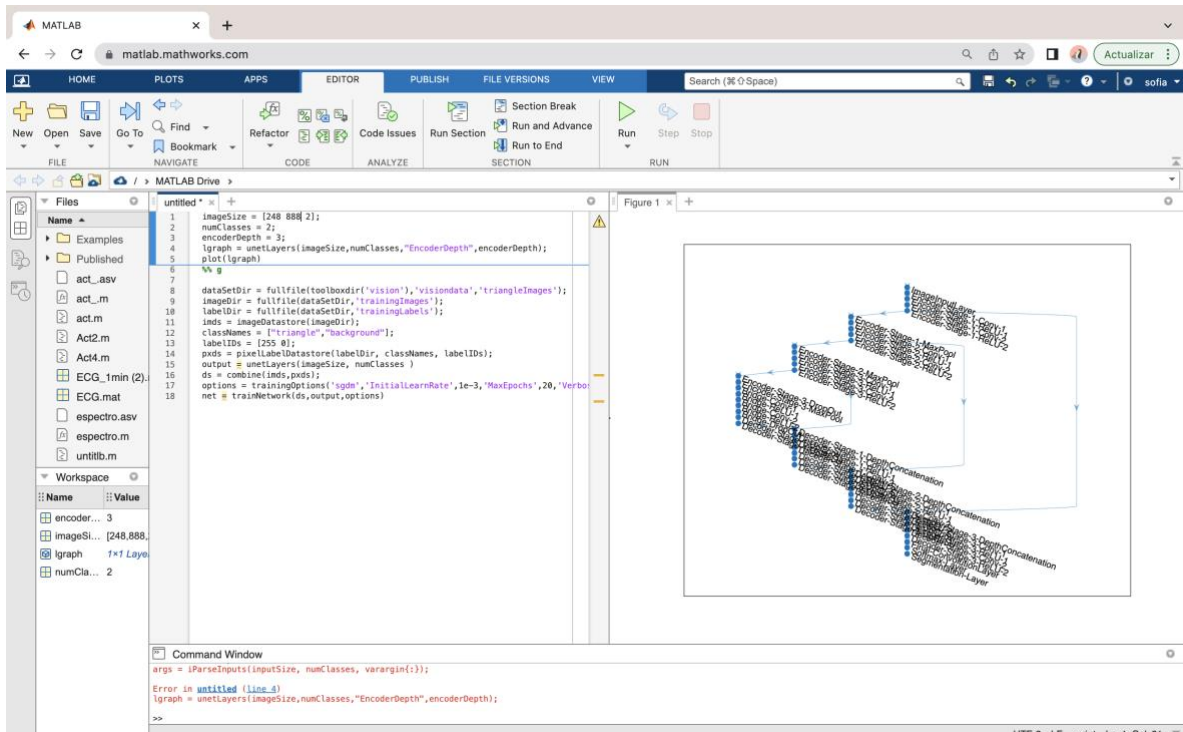
Profesor

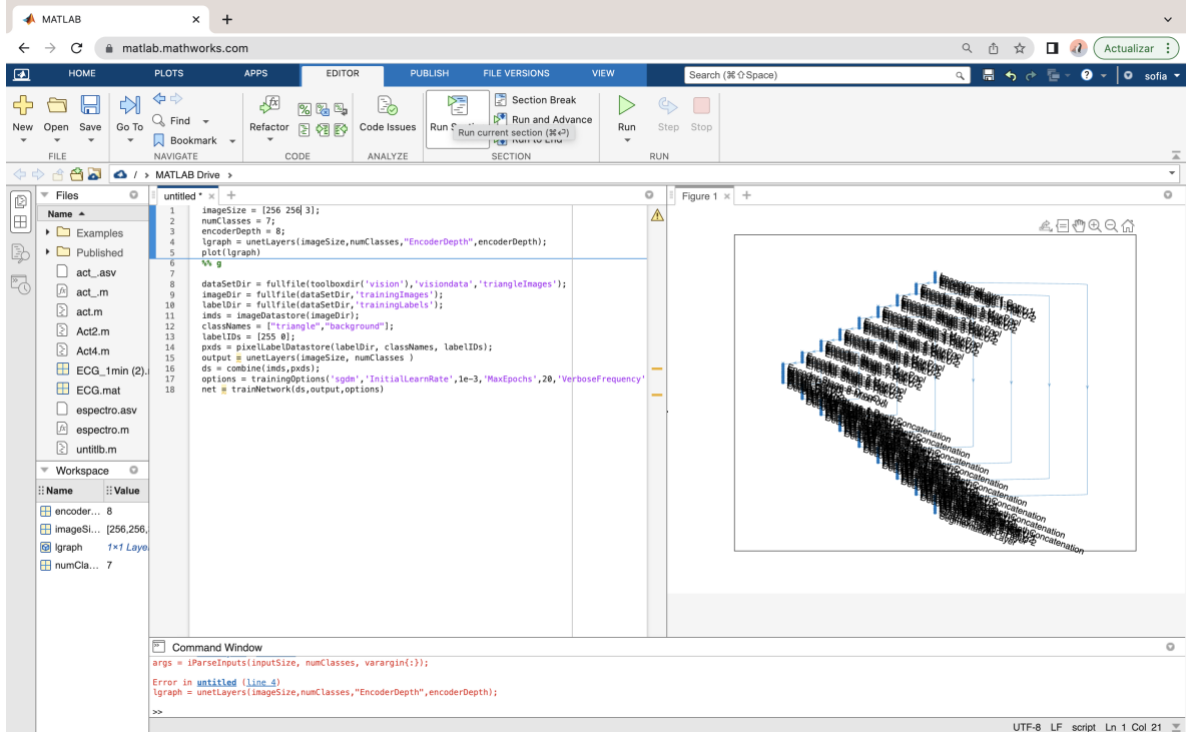
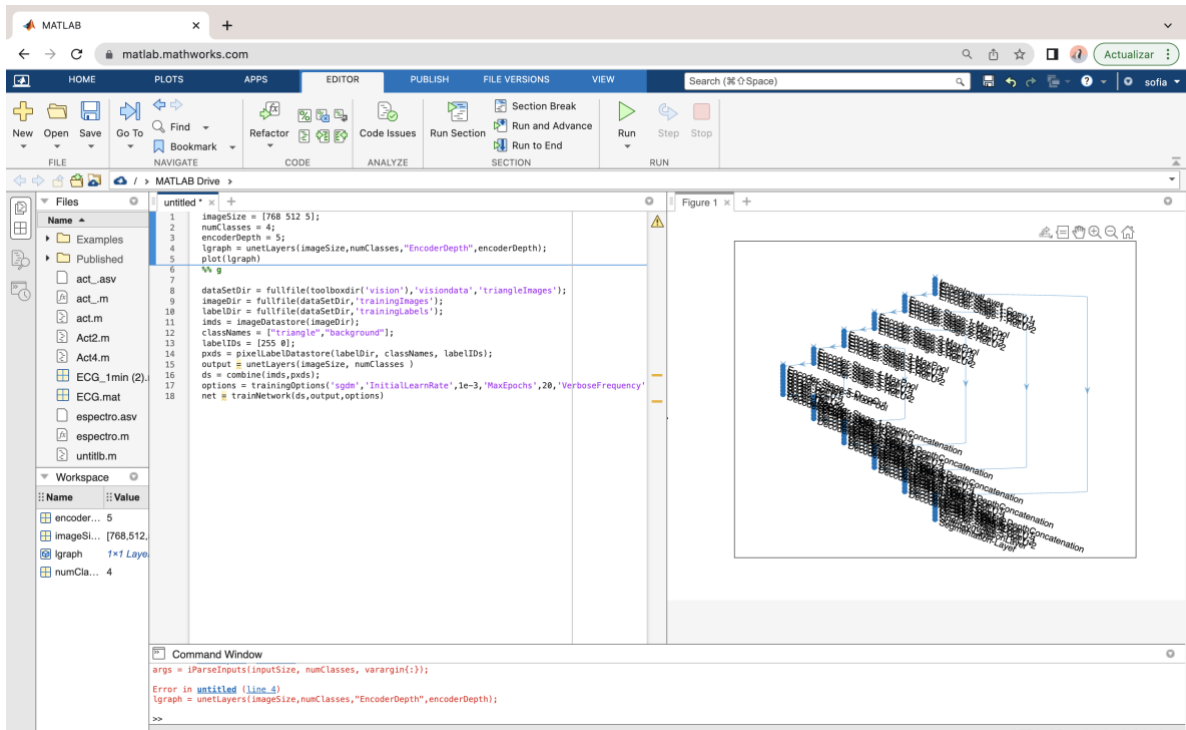
Dr. José Gerardo Tamez Peña

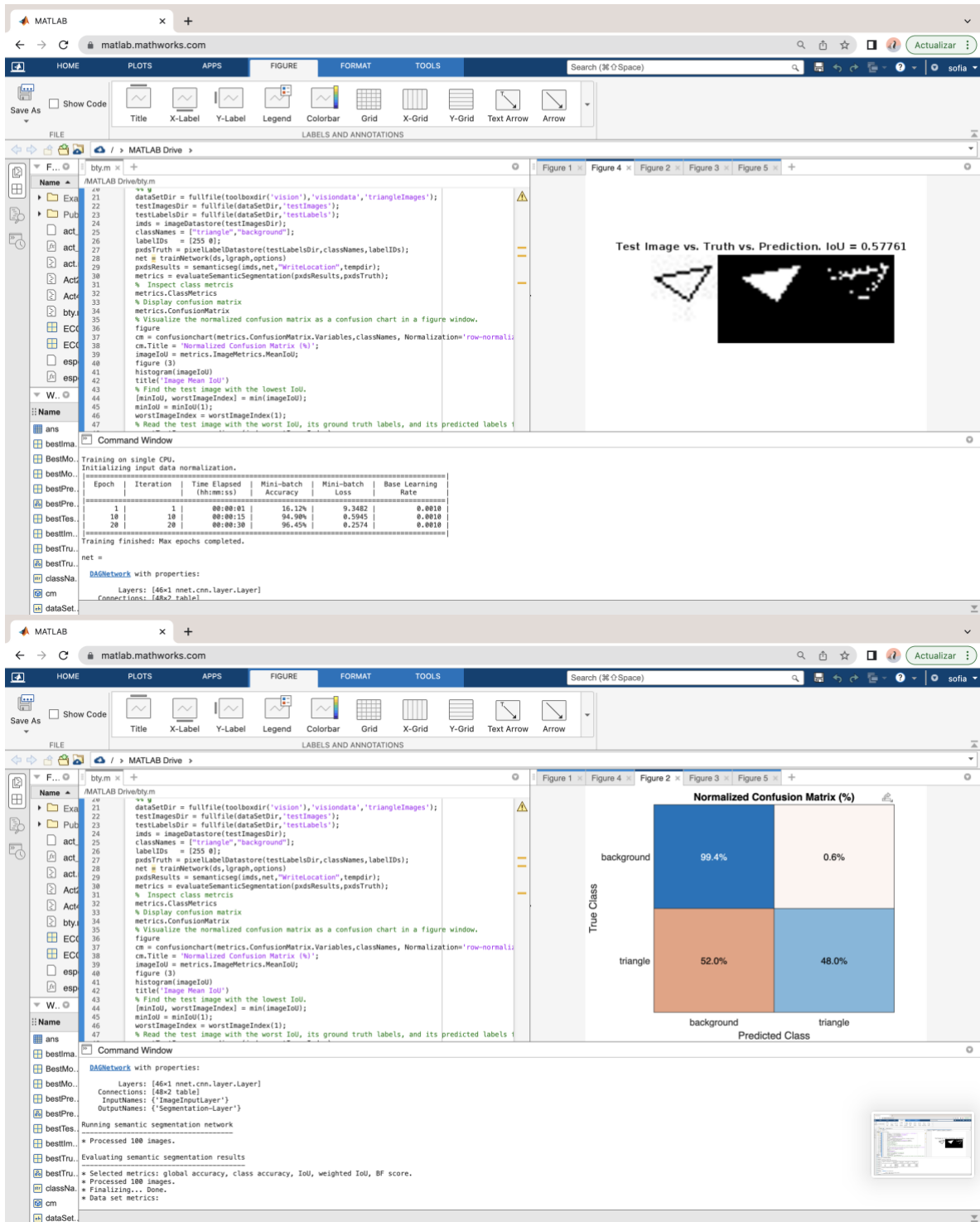
Azul Sofía Moctezuma Enriquez

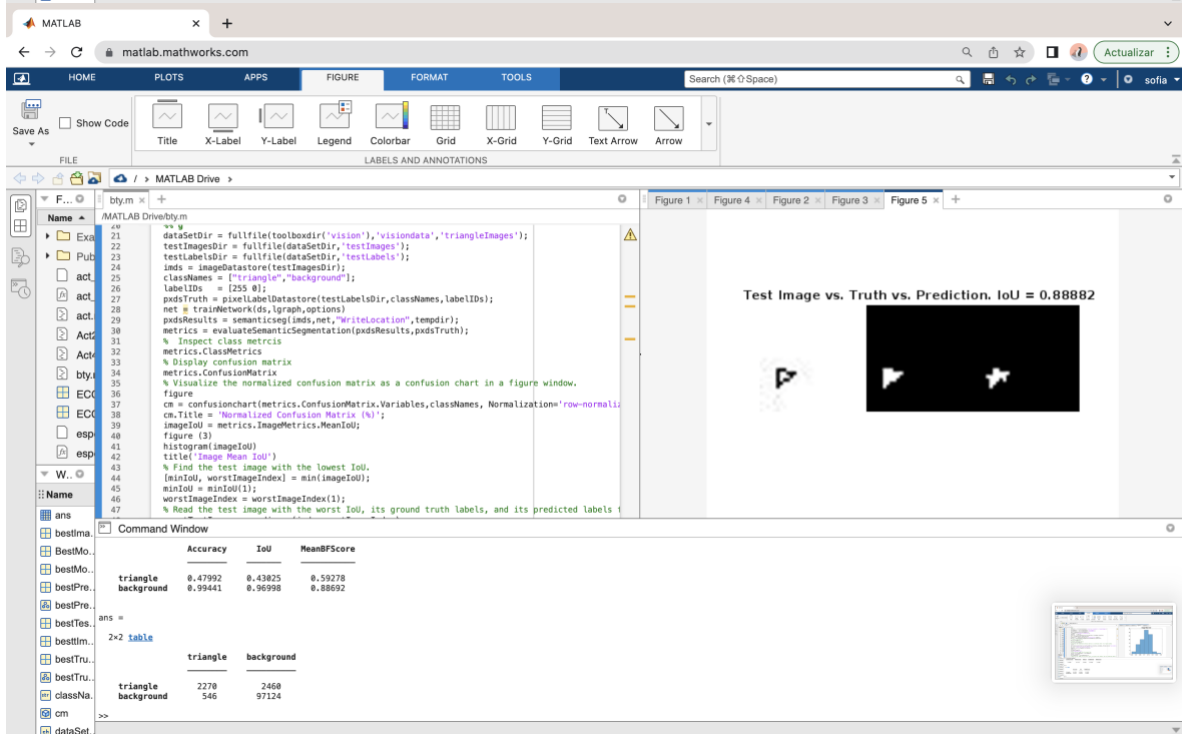
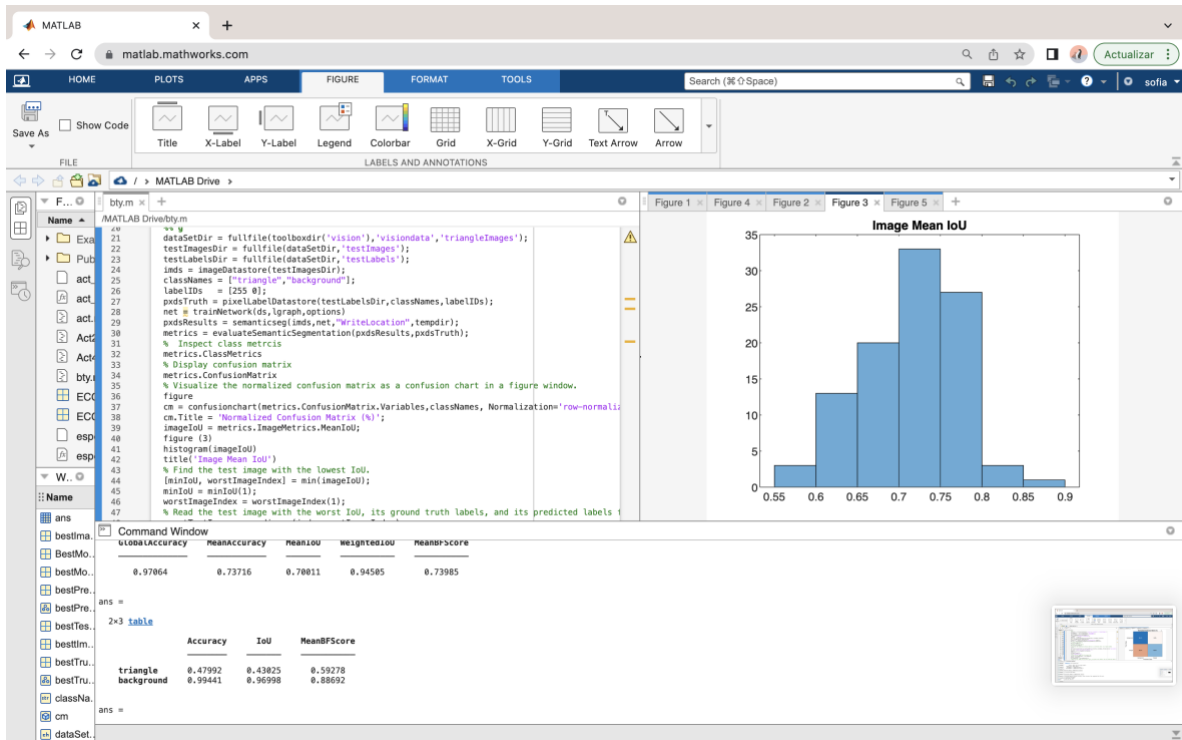
A01562585

febrero de 2023









Anexo de código:

```
imageSize = [32 32 1];
numClasses = 2;
encoderDepth = 3;
lgraph = unetLayers(imageSize,numClasses,"EncoderDepth",encoderDepth);
plot(lgraph)

dataSetDir = fullfile(toolboxdir('vision'),'visiondata','triangleImages');
imageDir = fullfile(dataSetDir,'trainingImages');
labelDir = fullfile(dataSetDir,'trainingLabels');
imds = imageDatastore(imageDir);
classNames = ["triangle","background"];
labelIDs = [255 0];
pxds = pixelLabelDatastore(labelDir, classNames, labelIDs);
output = unetLayers(imageSize, numClasses )
ds = combine(imds,pxds);
options = trainingOptions('sgdm','InitialLearnRate',1e-3,'MaxEpochs',20,'VerboseFrequency',10);
net = trainNetwork(ds,output,options)

%% g
dataSetDir = fullfile(toolboxdir('vision'),'visiondata','triangleImages');
testImagesDir = fullfile(dataSetDir,'testImages');
testLabelsDir = fullfile(dataSetDir,'testLabels');
imds = imageDatastore(testImagesDir);
classNames = ["triangle","background"];
labelIDs = [255 0];
pxdsTruth = pixelLabelDatastore(testLabelsDir,classNames,labelIDs);
net = trainNetwork(ds,lgraph,options)
pxdsResults = semanticseg(imds,net,"WriteLocation",tempdir);
metrics = evaluateSemanticSegmentation(pxdsResults,pxdsTruth);
metrics.ClassMetrics
metrics.ConfusionMatrix
figure
cm = confusionchart(metrics.ConfusionMatrix.Variables,classNames, Normalization='row-normalized');
cm.Title = 'Normalized Confusion Matrix (%)';
imageIoU = metrics.ImageMetrics.MeanIoU;
figure (3)
histogram(imageIoU)
title('Image Mean IoU')
[minIoU, worstImageIndex] = min(imageIoU);
minIoU = minIoU(1);
worstImageIndex = worstImageIndex(1);
worstTestImage = readimage(imds,worstImageIndex);
```

```

worstTrueLabels = readimage(pxdsTruth,worstImageIndex);
worstPredictedLabels = readimage(pxdsResults,worstImageIndex);
worstTrueLabelImage = im2uint8(worstTrueLabels == classNames(1));
worstPredictedLabelImage = im2uint8(worstPredictedLabels == classNames(1));
worstMontage = cat(4,worstTestImage,worstTrueLabelImage,worstPredictedLabelImage);
WorstMontage = imresize(worstMontage,4,"nearest");
figure (4)
montage(worstMontage,'Size',[1 3])
title(['Test Image vs. Truth vs. Prediction. IoU = ' num2str(minIoU)])
[maxIoU, bestImageIndex] = max(imageIoU);
maxIoU = maxIoU(1);
bestImageIndex = bestImageIndex(1);
bestTestImage = readimage(imds,bestImageIndex);
bestTrueLabels = readimage(pxdsTruth,bestImageIndex);
bestPredictedLabels = readimage(pxdsResults,bestImageIndex);
bestTrueLabelImage = im2uint8(bestTrueLabels == classNames(1));
bestPredictedLabelImage = im2uint8(bestPredictedLabels == classNames(1));
bestMontage = cat(4,bestTestImage,bestTrueLabelImage,bestPredictedLabelImage);
BestMontage = imresize(bestMontage,4,"nearest");
figure (5)
montage(bestMontage,'Size',[1 3])
title(['Test Image vs. Truth vs. Prediction. IoU = ' num2str(maxIoU)])

```