Instituto Tecnológico y de Estudios Superiores de Monterrey

Procesamiento de Imágenes Médicas para el Diagnóstico (Grupo 101)

Actividad 1: Segmentation U-net

Profesor
Dr. José Gerardo Tamez Peña

Diego De La Barreda Martínez
A01197739

03 de Marzo de 2023

Introducción:

El siguiente reporte está dividido en dos partes. La primera parte muestra cómo crear una U-network con una imágen creada y cargada para después poder entrenar este tipo de red para segmentación semántica. En la siguiente parte se evalúa que tan bien entrenada está la red. Se mira la confiabilidad de las predicciones con nuestros argumentos y se muestran los resultados por clases que nos dicen que tanto contraste tiene la imágen comparada con el fondo. La matriz de confusión obtenida nos dice información relacionada con los píxeles y a donde pertenecen de acuerdo a la imágen. El IoU nos va a indicar el grado de superposición entre el “Ground Truth” y la región de predicción.

Resultados:

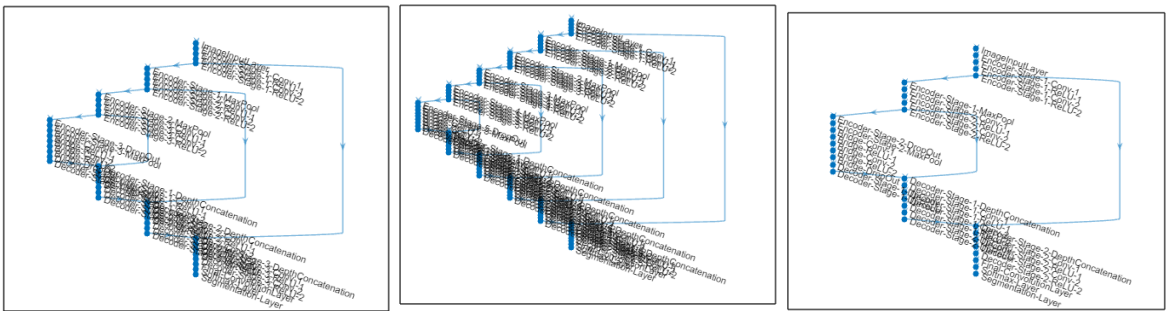

Figura 1. *Gráficas con diferentes especificaciones en tamaño, número de clases y profundidad*

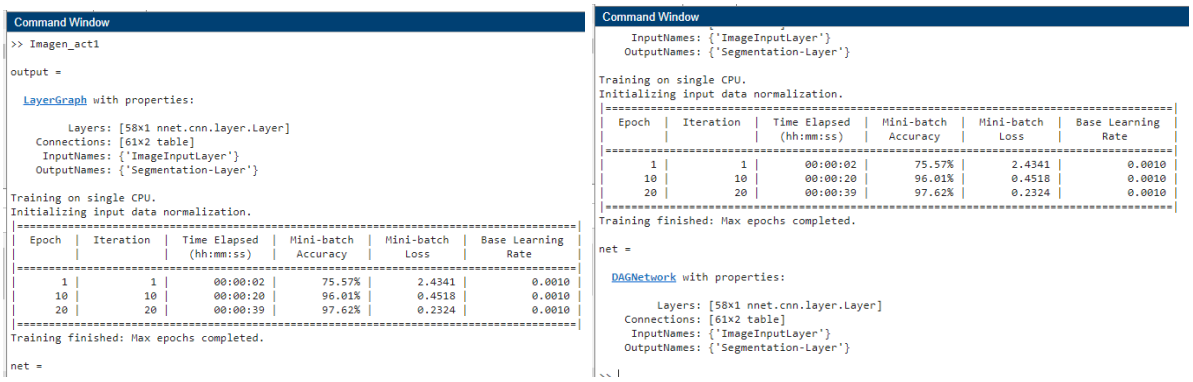

Figura 2. *Entreno de red para segmentación semántica*

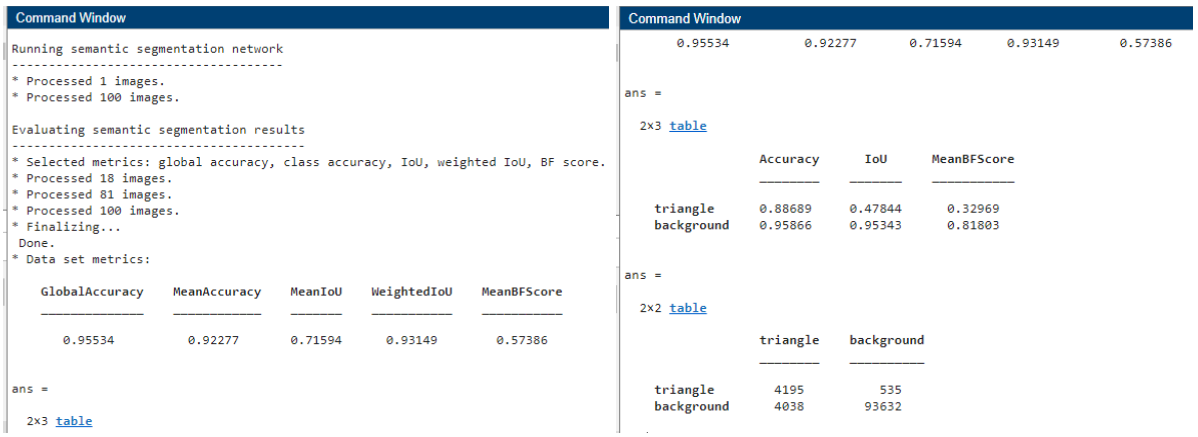

Figura 3. *Evaluación de resultados de segmentación semántica. Calidad de la predicción por píxeles y por clase*

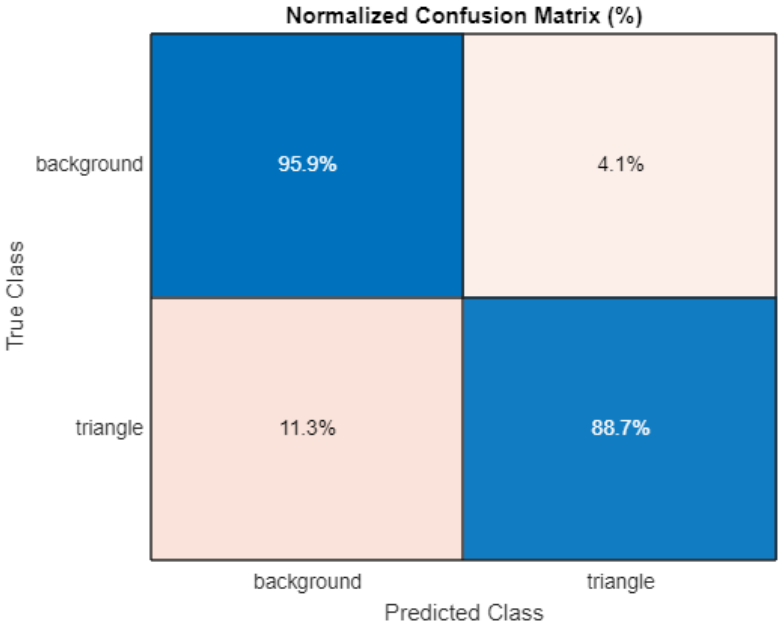

Figura 4. *Matriz de confusión*

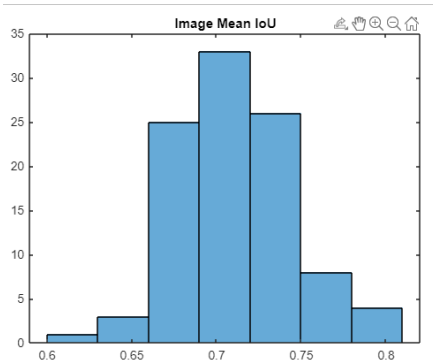

Figura 5. *Histograma de IoU por imágen*

Figura 6. *Imágenes con máximos y mínimos IoU*

Discusión:

La mayor cantidad de épocas que pueden ser usadas son 20 como se observa en la figura 2. Al utilizar 20 épocas vemos como tenemos una precisión de 97%. La tasa de aprendizaje no se ve que afecta la precisión al quedarse constante mientras aumentaban las épocas y las iteraciones. En la figura 5 el valor 0.7 se muestra como el promedio más común de IoU. En la figura 6 se muestran la superposición de las imágenes con el mejor y peor promedio de IoU. El peor promedio con un valor de 0.62 y el mejor con un 0.80. El promedio mayor muestra una imágen con mejor superposición a comparación del peor promedio. Al tener una mayoría de 0.7 como se muestra en la figura 5 se interpreta que se tiene una buena intersección sobre la unión en nuestras imágenes.

Anexos

Código primera parte

```
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,'VerboseFrequen
cy',10);
net = trainNetwork(ds,output,options)
```

Código segunda parte

```
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);
%  Inspect class metrcis
metrics.ClassMetrics
% Display confusion matrix
metrics.ConfusionMatrix
% Visualize the normalized confusion matrix as a confusion chart in a figure
window.
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')
% Find the test image with the lowest IoU.
[minIoU, worstImageIndex] = min(imageIoU);
```

```
minIoU = minIoU(1);
worstImageIndex = worstImageIndex(1);
% Read the test image with the worst IoU, its ground truth labels, and its
predicted labels for comparison.
worstTestImage = readimage(imds,worstImageIndex);
worstTrueLabels = readimage(pxdsTruth,worstImageIndex);
worstPredictedLabels = readimage(pxdsResults,worstImageIndex);
% Convert the label images to images that can be displayed in a figure
window.
worstTrueLabelImage = im2uint8(worstTrueLabels == classNames(1));
worstPredictedLabelImage = im2uint8(worstPredictedLabels == classNames(1));
% Display the worst test image, the ground truth, and the prediction.
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)])
% Find the test image with the highest IoU.
[maxIoU, bestImageIndex] = max(imageIoU);
maxIoU = maxIoU(1);
besttImageIndex = bestImageIndex(1);
% Read the test image with the worst IoU, its ground truth labels, and its
predicted labels for comparison.
bestTestImage = readimage(imds,bestImageIndex);
bestTrueLabels = readimage(pxdsTruth,bestImageIndex);
bestPredictedLabels = readimage(pxdsResults,bestImageIndex);
% Convert the label images to images that can be displayed in a figure
window.
bestTrueLabelImage = im2uint8(bestTrueLabels == classNames(1));
bestPredictedLabelImage = im2uint8(bestPredictedLabels == classNames(1));
% Display the worst test image, the ground truth, and the prediction.
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)])
```