Tecnológico de Monterrey

Instituto Tecnológico y de Estudios Superiores de Monterrey

U-net 2 Parte

BI2009B. Procesamiento de imágenes médicas para el diagnóstico
(Gpo 300)

Equipo 6

NOMBRE	MATRÍCULA
Mariely Charles Rodríguez	A00828348
Sebastián A. Mencías	A00828056
Ariana Fragoso Pérez	A00829129
Danya Rivera López	A01568331

Asesor: José Gerardo Tamez Peña

Monterrey, NL. 03 de junio, 2022

Código:

```
x = unetLayers([480 640 3], 5, 'EncoderDepth', 3);
plot(x); title('480X640X3, 5 classes, 3 depth')

y = unetLayers([480 640 3], 5, 'EncoderDepth', 4);
figure
plot(y); title('480X640X3, 10 classes, 4 depth')

z = unetLayers([480 640 3], 5, 'EncoderDepth', 2);
plot(z); title('480X640X3, 5 classes, 2 depth')

% Aquí se puede apreciar que el valor que tiene un mayor impacto
% en la estructura de la red neuronal es el encoder-decoder depth.
% Al reducir su valor, se reduce el número de 'escalones' que
% componen a la red neuronal.

%InitialLearnRate = 1e-3
% MaxEpochs = 20
dataSetDir = fullfile(toolboxdir('vision'),'visiondata','triangleImages');
imageDir = fullfile(dataSetDir,'trainingImages');
labelDir = fullfile(dataSetDir,'trainingLabels');

imds = imageDatastore(imageDir);
pxds = pixelLabelDatastore(labelDir,["triangle","background"],[255 0]);

a = unetLayers([32 32], 2);
ds = combine(imds,pxds);

options = trainingOptions('sgdm', 'InitialLearnRate',1e-3, ...
    'MaxEpochs',20, ...
    'VerboseFrequency',10);
net = trainNetwork(ds, a, options)

% Specify test images and labels
testImagesDir = fullfile(dataSetDir,'testImages');
testimds = imageDatastore(testImagesDir);
testLabelsDir = fullfile(dataSetDir,'testLabels');

pxdsTruth = pixelLabelDatastore(testLabelsDir,["triangle","background"],[255
0]);
%guarda los labels correctos
pxdsResults = semanticseg(testimds,net,"WriteLocation",tempdir);
%guarda los labels identificados por la red

metrics = evaluateSemanticSegmentation(pxdsResults,pxdsTruth);

%DISPLAY RESULTS BY CLASS
% 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, ...
["triangle","background"], Normalization ,'=','row-normalized');
cm.Title = 'Normalized Confusion Matrix (%)';
```

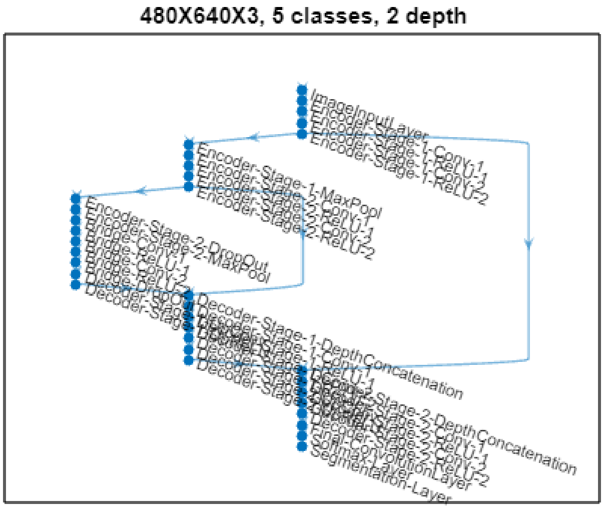
```
%VISUALIZE A HISTOGRAM OF THE IoU PER IMAGE
imageIoU = metrics.ImageMetrics.MeanIoU;
figure (3)
histogram(imageIoU)
title('Image Mean IoU')
```

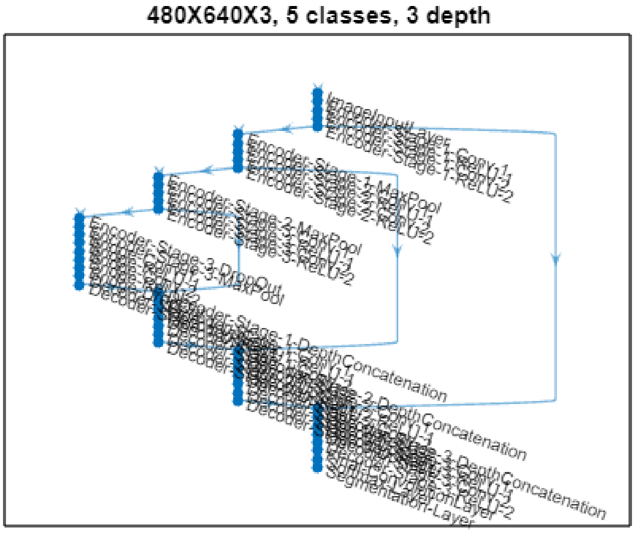
What was the most common mean IoU through the images?
Between 0.8 and 0.85

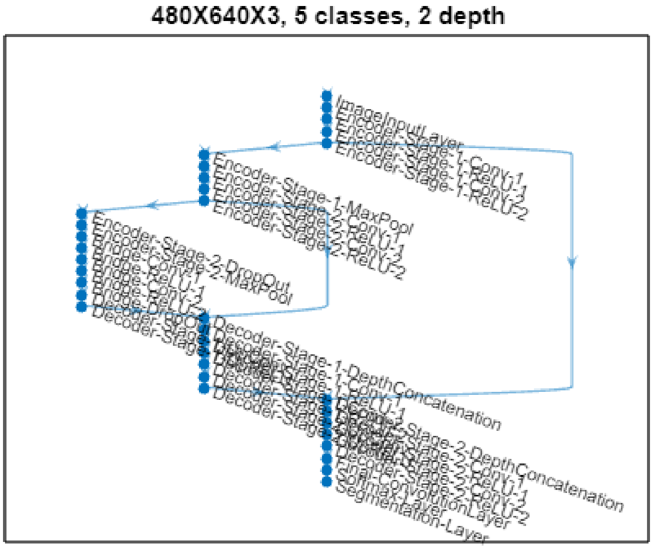
```
%IMAGE WITH LOWEST IoU
classNames = ["triangle","background"];
% Find the test image with the lowest IoU.
[minIoU, worstImageIndex] = min(imageIoU);
minIoU = minIoU(1);
worstImageIndex = worstImageIndex(1);

% 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.
bestMontage =
cat(4,worstTestImage,worstTrueLabelImage,worstPredictedLabelImage);
bestMontage = imresize(bestMontage,4,"nearest");
figure (4)
montage(bestMontage,'Size',[1 3])
title(['Test Image vs. Truth vs. Prediction. IoU = ' num2str(minIoU)])

%IMAGE WITH HIGHEST IoU
% Find the test image with the highest IoU.
[maxIoU, bestImageIndex] = max(imageIoU);
maxIoU = maxIoU(1);
bestImageIndex = bestImageIndex(1);

% Read the test image with the best 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 best test image, the ground truth, and the prediction.
bestMontage =
cat(4,bestTestImage,bestTrueLabelImage,bestPredictedLabelImage);
bestMontage = imresize(bestMontage,4,"nearest");
figure (4)
montage(bestMontage,'Size',[1 3])
```

```
title(['Test Image vs. Truth vs. Prediction. IoU = ' num2str(maxIoU)])

% %InitialLearnRate = 1e-2
% % MaxEpochs = 30
% dataSetDir = fullfile(toolboxdir('vision'),'visiondata','triangleImages');
% imageDir = fullfile(dataSetDir,'trainingImages');
% labelDir = fullfile(dataSetDir,'trainingLabels');
%
% imds = imageDatastore(imageDir);
% pxds = pixelLabelDatastore(labelDir,["triangle","background"],[255 0]);
%
% a = unetLayers([32 32], 2);
% ds = combine(imds,pxds);
%
% options = trainingOptions('sgdm', 'InitialLearnRate',1e-2, ...
%     'MaxEpochs',30, ...
%     'VerboseFrequency',10);
% net = trainNetwork(ds, a, options)

% %InitialLearnRate = 1e-1
% % MaxEpochs = 20
% dataSetDir = fullfile(toolboxdir('vision'),'visiondata','triangleImages');
% imageDir = fullfile(dataSetDir,'trainingImages');
% labelDir = fullfile(dataSetDir,'trainingLabels');
%
% imds = imageDatastore(imageDir);
% pxds = pixelLabelDatastore(labelDir,["triangle","background"],[255 0]);
%
% a = unetLayers([32 32], 2);
% ds = combine(imds,pxds);
%
% options = trainingOptions('sgdm', 'InitialLearnRate',1e-1, ...
%     'MaxEpochs',20, ...
%     'VerboseFrequency',10);
% net = trainNetwork(ds, a, options)
```

Resultados:




480X640X3, 5 classes, 3 depth
ImageInputLayer
Encoder-Stage-1-Conv-1
Encoder-Stage-1-ReLU-1
Encoder-Stage-1-Conv-2
Encoder-Stage-1-ReLU-2
Encoder-Stage-1-MaxPool
Encoder-Stage-2-Conv-1
Encoder-Stage-2-ReLU-1
Encoder-Stage-2-Conv-2
Encoder-Stage-2-ReLU-2
Encoder-Stage-2-MaxPool
Encoder-Stage-3-Conv-1
Encoder-Stage-3-ReLU-1
Encoder-Stage-3-Conv-2
Encoder-Stage-3-ReLU-2
Encoder-Stage-3-DropOut
Encoder-Stage-3-MaxPool
Decoder-Stage-1-DepthConcatenation
Decoder-Stage-2-DepthConcatenation
Decoder-Stage-3-DepthConcatenation
Decoder-Stage-3-Conv-1
Decoder-Stage-3-ReLU-1
Decoder-Stage-3-Conv-2
Decoder-Stage-3-ReLU-2
Final-ConvolutionLayer
Softmax-Layer
Segmentation-Layer


480X640X3, 5 classes, 2 depth
ImageInputLayer
Encoder-Stage-1-Conv-1
Encoder-Stage-1-ReLU-1
Encoder-Stage-1-Conv-2
Encoder-Stage-1-ReLU-2
Encoder-Stage-1-MaxPool
Encoder-Stage-2-Conv-1
Encoder-Stage-2-ReLU-1
Encoder-Stage-2-Conv-2
Encoder-Stage-2-ReLU-2
Encoder-Stage-2-DropOut
Encoder-Stage-2-MaxPool
Bridge-Conv-1
Bridge-ReLU-1
Bridge-Conv-2
Bridge-ReLU-2
Bridge-DropOut
Decoder-Stage-1-DepthConcatenation
Decoder-Stage-1-Conv-1
Decoder-Stage-1-ReLU-1
Decoder-Stage-2-DepthConcatenation
Decoder-Stage-2-Conv-1
Decoder-Stage-2-ReLU-1
Decoder-Stage-2-Conv-2
Decoder-Stage-2-ReLU-2
Final-ConvolutionLayer
Softmax-Layer
Segmentation-Layer

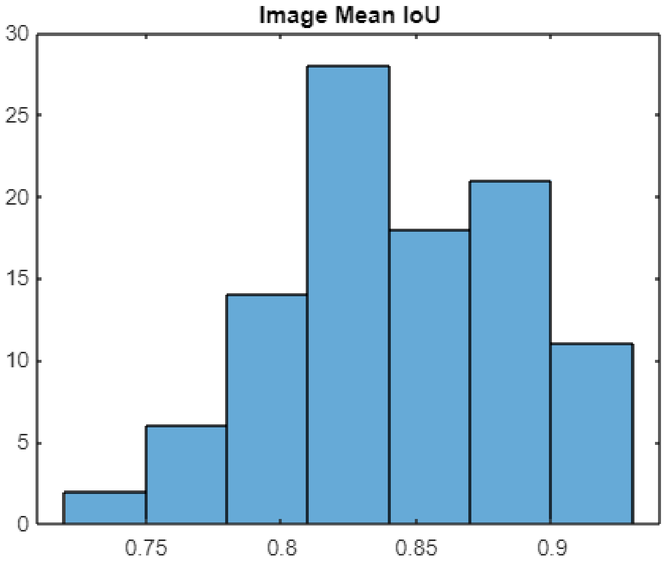
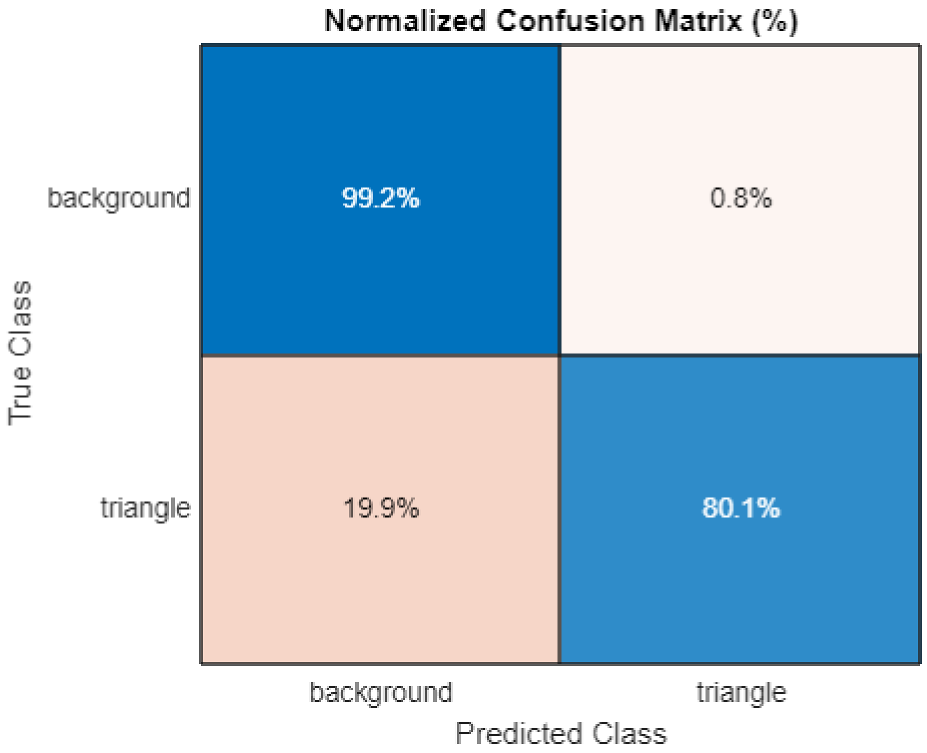


Image Mean IoU
30
25
20
15
10
5
0
0.75
0.8
0.85
0.9


Normalized Confusion Matrix (%)
True Class
background
triangle
99.2%
0.8%
19.9%
80.1%
background
triangle
Predicted Class