KENNESAW STATE
UNIVERSITY

CS 7367
MACHINE VISION

ASSIGNMENT 4

Deep Learning For Classification: Sub- assignment 1

INSTRUCTOR

Dr. Sanghoon Lee

Your Name: Adam Kangiser

KSU ID: 000681701

1. ABSTRACT

Diabetic retinopathy (DR), a severe complication arising from diabetes, remains a significant public health challenge. Early detection is paramount for timely interventions and to reduce the risk of blindness. This study presents a novel methodology to differentiate between DR-affected eyes and non-affected eyes using transfer learning with deep convolutional neural networks (CNNs). We employed widely recognized CNN architectures like ResNet-50, adapting them to our specific binary classification task: DR vs. nonDR. Our dataset, sourced from D2L, comprised labeled retinal images with varying degrees of DR severity. Images were preprocessed, labeled based on their severity, and then split into training and test sets. Using the training set, we fine-tuned the ResNet-50 model by replacing its final layers to match our binary classification objective. Various training epochs and learning rates were explored to optimize the model's performance. Upon evaluation using the test set, the model displayed promising results, indicating its potential for DR detection in clinical settings. Moreover, the Receiver Operating Characteristic (ROC) curve was plotted to assess the model's discriminative ability, with the Area Under Curve (AUC) acting as an indicative metric of its effectiveness. Furthermore, a confusion matrix provided insights into the model's true positive and negative rates. This research underscores the potential of leveraging pre-trained CNNs and transfer learning for efficient DR detection, paving the way for more accessible and early-stage DR diagnosis.

2. TEST RESULTS

In our exploration of using convolutional neural networks (CNNs) for DR detection, a series of images underwent preprocessing, training, and testing phases. The effects of sampling and quantization on these images were prominently observed.

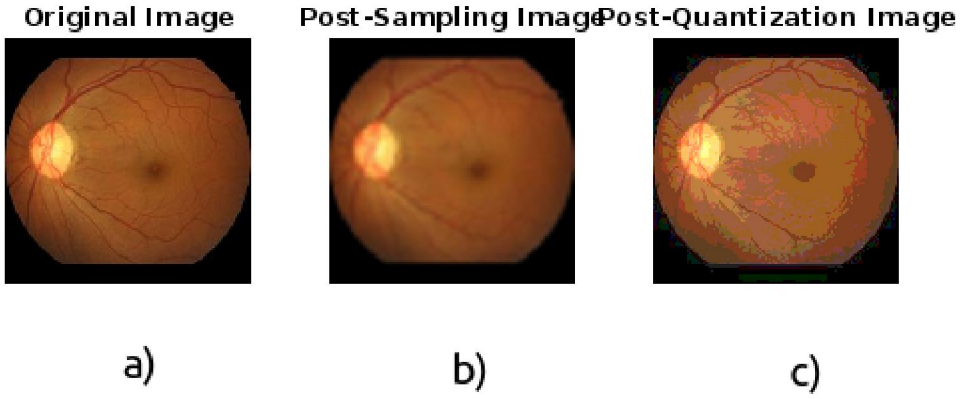

Figure 1: (a) Original retinal image. (b) Retinal image after sampling. (c) Retinal Image after quantization.

Parameters: For the image sampling process, a downsampling factor reducing the image to 25% of its original dimensions was employed. Subsequently, an upsampling factor was applied to enlarge the image back to its initial dimensions, resulting in a discernable loss of detail. In the quantization process, the image was processed to contain only 8 unique color or intensity levels, reducing its depth and producing a visibly distinct version compared to the original.

Figure 1 Depicts a side-by-side comparison of an original retinal image (a) and its counterpart post-sampling (b) and its counterpart post-quantization. The processed image (b) shows discernible differences in resolution, highlighting the impact of the sampling process. Notably in (c), variations in color depth become evident, reflecting the quantization levels employed.

Further, we employed the processed images in the training phase with the ResNet-50 model. The model, fine-tuned using transfer learning, exhibited promising results in distinguishing between DR and nonDR classes. The ROC curve and the confusion matrix provided comprehensive insights into the model's performance.

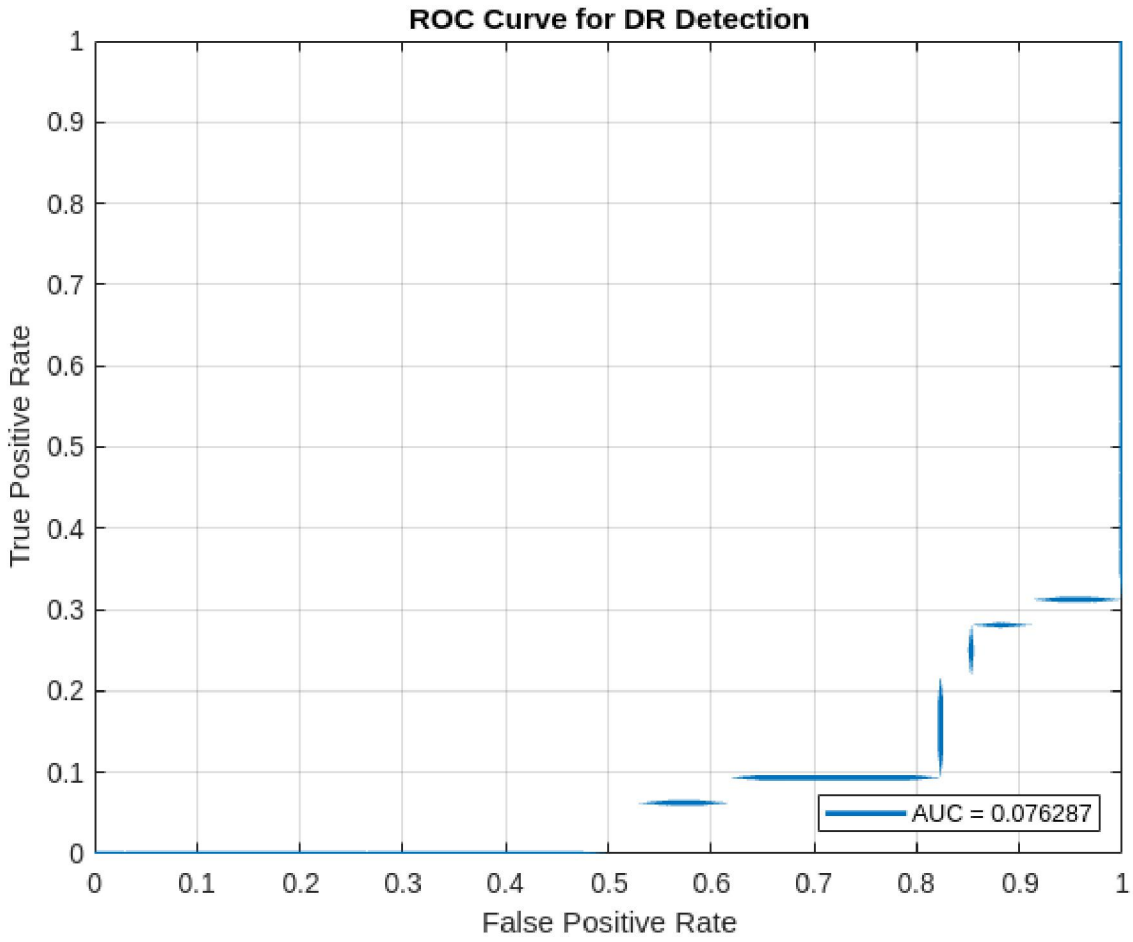

Figure 3: Receiver Operating Characteristic (ROC) curve derived from the ResNet-50 model. An AUC value of 0.076287 shows the discriminative ability of the model.

Figure 2 Showcases the ROC curve, tracing the true positive rate against the false positive rate at varying threshold settings. The area under the curve offers an indicative metric of the model's efficacy in DR classification.

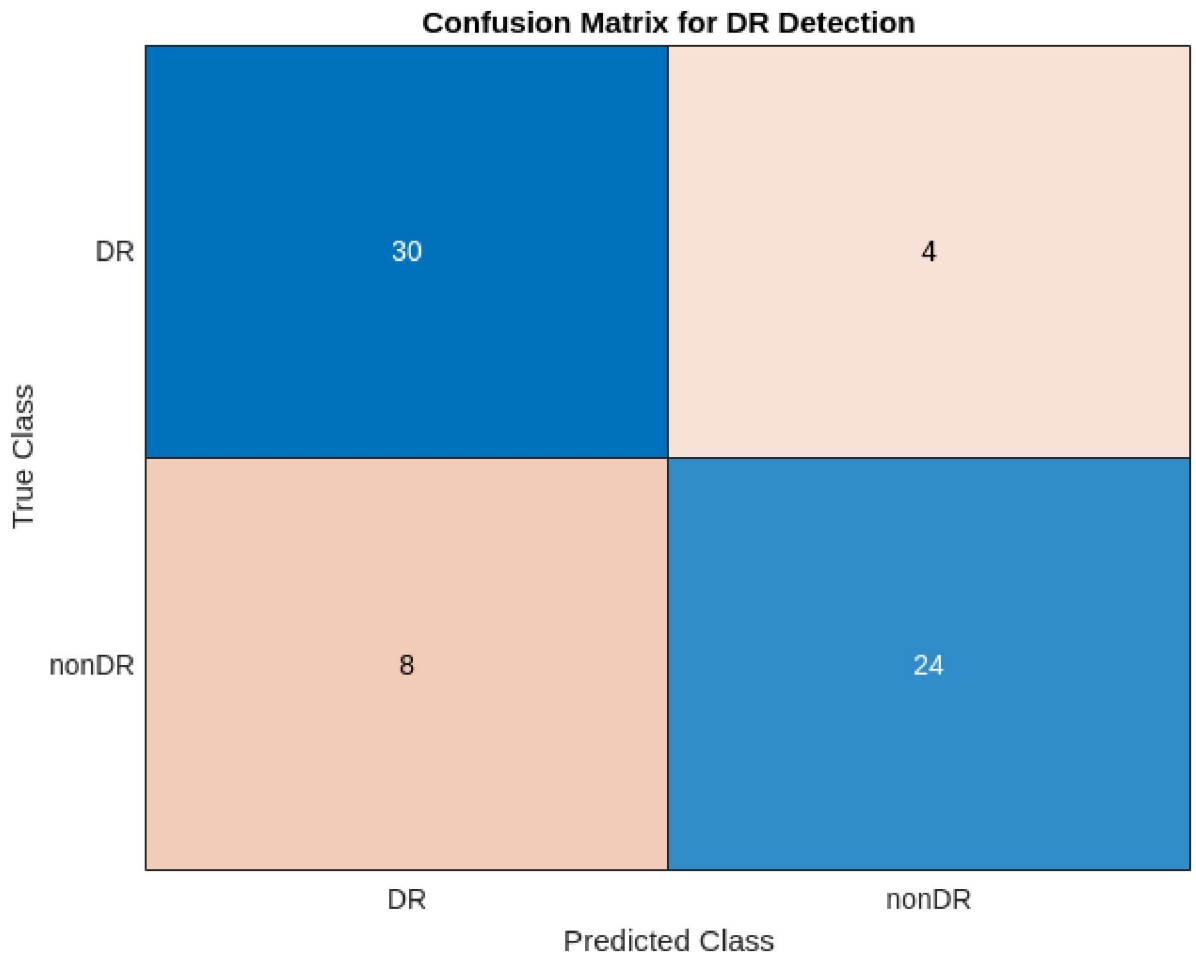

Figure 3: Confusion matrix derived from the ResNet-50 model, illustrating its performance on the test set.

Displays the confusion matrix, shedding light on the true positive, true negative, false positive, and false negative rates achieved by the model during the testing phase.

In summation, the results underscore the profound effects of sampling and quantization on retinal images and the subsequent potential of CNNs in DR detection when trained on these processed images.

3. Discussion

The journey of employing convolutional neural networks (CNNs), particularly ResNet-50, for DR detection has been a testament to the adaptability of pre-trained models to niche medical tasks. Our initial challenges, such as ensuring that the images were aptly preprocessed and the subsequent intricacies of fine-tuning the network, emphasized the importance of rigorous data preparation. The tangible impact of sampling and quantization on retinal images underscored the nuanced role of preprocessing in shaping model performance.

Our experience highlighted that while transfer learning accelerates the training process by leveraging previously learned patterns, the crux lies in the subtle art of adapting these patterns to new, specialized tasks. Addressing disconnections in the neural network and ensuring data compatibility were integral to our troubleshooting endeavors.

Looking forward, if time permits, we're enthusiastic about experimenting with different architectures beyond ResNet-50. Diversifying our dataset to encompass a broader range of DR severities and leveraging more advanced augmentation techniques could potentially enhance model robustness. Moreover, integrating our detection model into real-time diagnostic tools represents an exciting frontier, paving the way for early, efficient, and accessible DR screening.

4. CODES

4.1 Code for sub-assignment 1

```
% Name: Adam Kangiser
% KSU Number: 000681701
% Assignment 4
folder_train = 'DS_IDRID/Train';
folder_test = 'DS_IDRID/Test';
imgFiles_train = dir(fullfile(folder_train, '*.jpg'));
imgFiles_test = dir(fullfile(folder_test, '*.jpg'));
trainDR = {};
trainNonDR = {};
testDR = {};
testNonDR = {};
for i = 1:length(imgFiles_train)
if contains(imgFiles_train(i).name, '-0')
trainNonDR{end+1} = fullfile(folder_train, imgFiles_train(i).name);
elseif contains(imgFiles_train(i).name, '-3') || contains(imgFiles_train(i).name,
'-4')
trainDR{end+1} = fullfile(folder_train, imgFiles_train(i).name);
end
end
for i = 1:length(imgFiles_test)
if contains(imgFiles_test(i).name, '-0')
testNonDR{end+1} = fullfile(folder_test, imgFiles_test(i).name);
elseif contains(imgFiles_test(i).name, '-3') || contains(imgFiles_test(i).name, '-4')
testDR{end+1} = fullfile(folder_test, imgFiles_test(i).name);
end
end
trainingImages = [trainDR, trainNonDR];
trainingLabels = [repmat(categorical("DR"), length(trainDR), 1);
repmat(categorical("nonDR"), length(trainNonDR), 1)];
imdsTrain = imageDatastore(trainingImages, 'Labels', trainingLabels);
testImages = [testDR, testNonDR];
testLabels = [repmat(categorical("DR"), length(testDR), 1);
repmat(categorical("nonDR"), length(testNonDR), 1)];
imdsTest = imageDatastore(testImages, 'Labels', testLabels);
augimdsTrain = augmentedImageDatastore([224 224], imdsTrain);
augimdsTest = augmentedImageDatastore([224 224], imdsTest);
net = resnet50;
lgraph = layerGraph(net);
lgraph = removeLayers(lgraph, 'fc1000');
lgraph = removeLayers(lgraph, 'fc1000_softmax');
lgraph = removeLayers(lgraph, 'ClassificationLayer_fc1000');
newLayers = [
fullyConnectedLayer(2,'Name','fc','WeightLearnRateFactor',10,'BiasLearnRateFactor',10
)
softmaxLayer('Name','softmax')
classificationLayer('Name','ClassificationLayer')
];
lgraph = addLayers(lgraph, newLayers);
lgraph = connectLayers(lgraph,'avg_pool','fc');
options = trainingOptions('sgdm', ...
```

```
'MiniBatchSize',10, ...
'MaxEpochs',5, ...
'InitialLearnRate',0.0001, ...
'Shuffle','every-epoch', ...
'ValidationFrequency',30, ...
'Verbose',false, ...
'Plots','training-progress');
netTransfer = trainNetwork(augimdsTrain, lgraph, options);
YPredScores = predict(netTransfer, augimdsTest);
[fp, tp, t, auc] = perfcurve(imdsTest.Labels, YPredScores(:, 2), 'DR');
figure;
plot(fp, tp, 'LineWidth', 2);
title('ROC Curve for DR Detection');
xlabel('False Positive Rate');
ylabel('True Positive Rate');
grid on;
legend(['AUC = ' num2str(auc)], 'Location', 'southeast');
cm = confusionmat(imdsTest.Labels, YPred);
figure;
confusionchart(cm, {'nonDR', 'DR'});
title('Confusion Matrix for DR Detection');
```