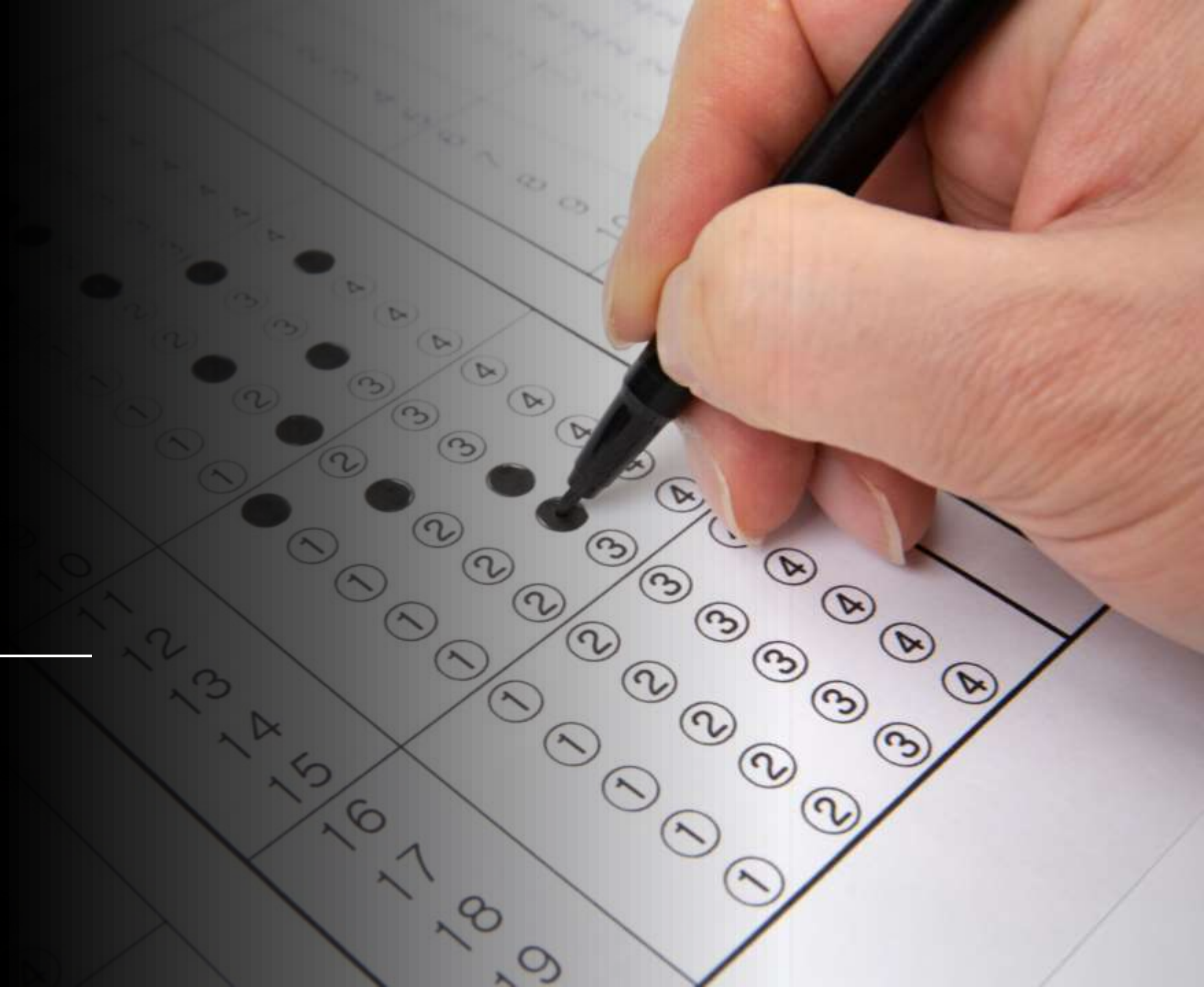
Final Presentation signature detector

D0965676 Kevin Lin

D0970257 Chris Chiu



Motivation

We found that almost 80% children have experience of counterfeit their parents' signature.

Thus, we are inspired by this phenomenon and decide to make program that can

distinguish the signature is fake or not. Hoping this program can reduce the happening of

this kind of behavior.

Background_Train/Test data

	Train	Test
True	78	52
Fake	78	52

True/130 by handwriting

Fake/130 by handwriting

Accuracy/TrainingTime

- Validation Accuracy 98.08% / spent 61min48sec

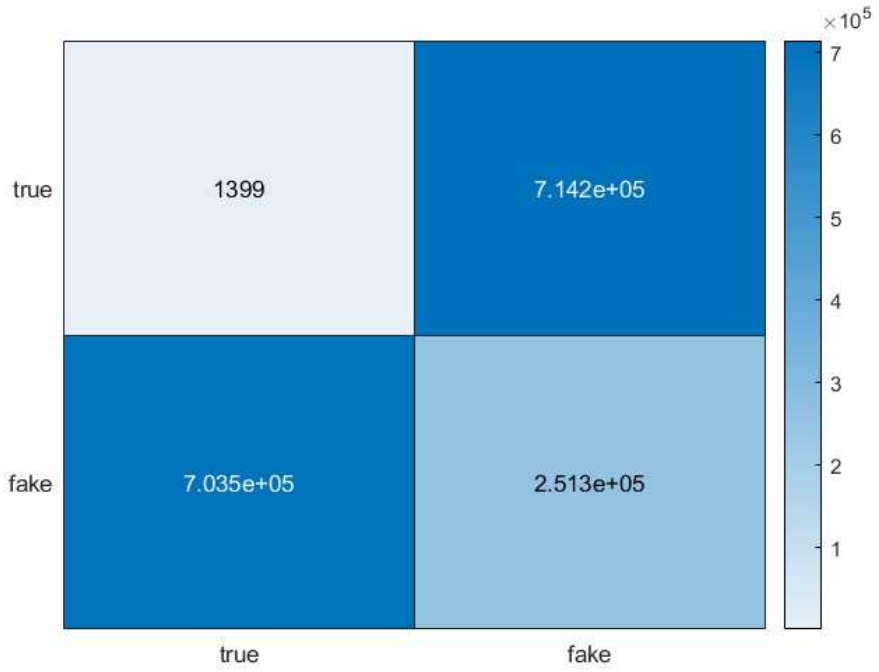
26	fake
27	fake
28	fake
29	fake
30	fake
31	true
32	true
33	fake
34	fake
35	fake

```
confMat =

    50     2
     0    52
```

Validation accuracy:	98.08%
Training finished:	Max epochs completed
Training Time	
Start time:	31-May-2022 05:42:34
Elapsed time:	61 min 48 sec

iou



ans =

2×3 table

	Accuracy	IoU	MeanBFScore
true	0.001955	0.00098587	NaN
fake	0.26323	0.15059	NaN

* Data set metrics:

GlobalAccuracy	MeanAccuracy	MeanIoU	WeightedIoU	MeanBFScore
0.1513	0.13259	0.075788	0.086499	NaN

Purposed method

- Step1: collect data of signature which includes of authentic one and fake one.
- Step2: 1.use adam optimizer and res101 to train NET which distinguish true and fake class.

  2.train a segmentation NET by using the filler to mark ***260*** pictures.
- Step3: 1.upload NET to MATLAB drive and load into program using MATLAB mobile.

  2.put NET to program which connect with webcam.

  3.put NET to program which can load picture directly from computer .
- Step4: Use segmentation NET to ensure that if the word have been marked.
- Step5: After distinguishing the picture, we will get accuracy of true or fake class.

  True : If it is true and over 80% , we can define it is true and then turn on the green light on thingSpeak, display the accuracy and the certification on the Excel.

  Fake: If it is fake or true which is under 80%, we can define it is fake and then turn on the red light on thingSpeak.

code

Matlab mobile program

```
clear all
if exist('mObj')
    clear mbj
end
mObj = mobiledev;
c = camera(mObj,'back');
net = load('/MATLAB Drive/Project/myNet30.mat');

img = snapshot(c,'manual');
[label,score] = classify(net.myNet,imresize(img,[400,400]));

image(img);
title({char(label),num2str(max(score),2)});

if label=="true"&&score>80
    Data = timetable(datetime,score*100);
    web('https://api.thingspeak.com/update?api_key=8FTVCQ65UPHI8IOM&field2=1')
    pause(2)
    thingSpeakWrite(1735704,Data,'WriteKey','8FTVCQ65UPHI8IOM');
else
    web('https://api.thingspeak.com/update?api_key=8FTVCQ65UPHI8IOM&field2=0')
end
```

Program of train net work

```
cd C:\Users\kevin\Desktop\ProjectSignature\test
allData=imageDatastore('test400x400','IncludeSubfolders',true,'LabelSource','foldernames');

% allData = im2bw(I, 0.5)

load('C:\Users\kevin\Desktop\ProjectSignature\layer\matlab1.ma')
[imds60,imds40]=splitEachLabel(allData,0.6,'randomized');


opts=trainingOptions('adam','InitialLearnRate',0.002,'MaxEpochs',50,'MiniBatchSize',64,'Plots','training-progress',...
'ValidationData',imds40,'ExecutionEnvironment','cpu');

myNet=trainNetwork(imds60,lgraph_2,opts);
desiredLabel=imds40.Labels;
predictedLabel=classify(myNet,imds40);
accuracy=mean(desiredLabel==predictedLabel)

confMat = confusionmat(desiredLabel,predictedLabel)
for i = 1:size(confMat,1)
    precision(i)=confMat(i,i)/sum(confMat(i,:));
end
for i = 1:size(confMat,1)
    recall(i)=confMat(i,i)/sum(confMat(:,i));
end
F1 = (2.*precision.*recall)/(precision+recall)
```

1

```
clear all

cam = webcam(1)
preview(cam)
pause(10)
img = snapshot(cam)
img = imresize(img,[400,400]);
imwrite(img,'C:\Users\kevin\Desktop\ProjectSignature\pic\segment.jpg')%store into a folder
net = load('C:\Users\kevin\Desktop\ProjectSignature\Net\myNetres280098.mat');

[label,score] = classify(net.myNet,img);
image(img);
title({char(label),num2str(max(score),2)});
points = max(score)*100;

fname = 'C:\Users\kevin\Desktop\ProjectSignature\excel';
saveas(gcf, fullfile(fname, 'test'), 'jpg');

folder = 'C:\Users\kevin\Desktop\ProjectSignature\excel';
excelFileName = 'Certification.xlsx';
fullFileName = fullfile(folder, excelFileName);

data = xlsread(fullFileName, "工作表1");
cellarray = num2cell(data);
accuracy = strcat(num2str(points), "%");
cellarray{1,1} = accuracy;
xlswrite(fullFileName,cellarray,"工作表1")
```

2

```
myNet1 = load('C:\Users\kevin\Desktop\ProjectSignature\segnet\myNet30965.mat')
testImage = imageDatastore('C:\Users\kevin\Desktop\ProjectSignature\pic\');
prediced = semanticseg(testImage,myNet1.myNet2);
N = 1;
x = labeloverlay(readimage(testImage, N),readimage(prediced,N));
figure,imshowpair(x,img,'montage')
```

3

```
if label=="true" && points >80

    if ~exist(fullFileName, 'file')
  message = sprintf('Existing Excel workbook not found"\n%s', fullFileName);
  uiwait(errordlg(message));
  return;
end
objExcel = actxserver('Excel.Application');
objExcel.Visible = true;
ExcelWorkbook = objExcel.Workbooks.Open(fullFileName);
oSheet = objExcel.ActiveSheet;
imageFolder = 'C:\Users\kevin\Desktop\ProjectSignature\excel'
imageFullFileName = fullfile(imageFolder, 'test.jpg')
Shapes = oSheet.Shapes;
Shapes.AddPicture(imageFullFileName, 0, 1, 18, 36, 400, 400);
objExcel.DisplayAlerts = false;

    web('https://api.thingspeak.com/update?api_key=8FTVCQ65UPHI8IOM&field2=1')
    pause(20)
    Data = timetable(datetime,points);
    thingSpeakWrite(1735704,Data,'WriteKey','8FTVCQ65UPHI8IOM');


else
    web('https://api.thingspeak.com/update?api_key=8FTVCQ65UPHI8IOM&field2=0')
end
```

Webcam with *Excel* + *Segmentation*

```
clear all

[filename pathname] = uigetfile({'*.jpg';'*.bmp'},'Select MRI');
inputimg = strcat(pathname, filename);
img = imread(inputimg);
img = imresize(img,[400,400]);
imwrite(img,'C:\Users\kevin\Desktop\ProjectSignature\pic\segment.jpg')%store into a folder
net = load('C:\Users\kevin\Desktop\ProjectSignature\Net\myNetres280098.mat');

[label,score] = classify(net.myNet,img);
image(img);
title({char(label),num2str(max(score),2)});
points = max(score)*100;

fname = 'C:\Users\kevin\Desktop\ProjectSignature\excel';
saveas(gcf, fullfile(fname, 'test'), 'jpg');

folder = 'C:\Users\kevin\Desktop\ProjectSignature\excel';
excelFileName = 'Certification.xlsx';
fullFileName = fullfile(folder, excelFileName);

data = xlsread(fullFileName, "工作表1");
cellarray = num2cell(data);
accuracy = strcat(num2str(points), "%");
cellarray{1,1} = accuracy;
xlswrite(fullFileName,cellarray,"工作表1")

myNet1 = load('C:\Users\kevin\Desktop\ProjectSignature\segnet\myNet30965.mat')
testImage = imageDatastore('C:\Users\kevin\Desktop\ProjectSignature\pic\');
prediced = semanticseg(testImage,myNet1.myNet2);
N = 1;
x = labeloverlay(readimage(testImage, N),readimage(prediced,N));
figure,imshowpair(x,img,'montage')
```

```
if label=="true" && points >80

    if ~exist(fullFileName, 'file')
  message = sprintf('Existing Excel workbook not found"\n%s', fullFileName);
  uiwait(errordlg(message));
  return;
end
objExcel = actxserver('Excel.Application');
objExcel.Visible = true;
ExcelWorkbook = objExcel.Workbooks.Open(fullFileName);
oSheet = objExcel.ActiveSheet;
imageFolder = 'C:\Users\kevin\Desktop\ProjectSignature\excel'
imageFullFileName = fullfile(imageFolder, 'test.jpg')
Shapes = oSheet.Shapes;
Shapes.AddPicture(imageFullFileName, 0, 1, 0, 36, 400, 400);
objExcel.DisplayAlerts = false;

    web('https://api.thingspeak.com/update?api_key=8FTVCQ65UPHI8IOM&field2=1')
    pause(20)
    Data = timetable(datetime,points);
    thingSpeakWrite(1735704,Data,'WriteKey','8FTVCQ65UPHI8IOM');


else
    web('https://api.thingspeak.com/update?api_key=8FTVCQ65UPHI8IOM&field2=0')
end
```

Load from PC with *Excel* + *Segmentation*

Segmentation

```
load('C:\Users\kevin\Desktop\ProjectSignature\gtruth\train\train_gTruth.mat')
option = trainingOptions('adam','MaxEpochs', 50,'MiniBatchSize', 16,'InitialLearnRate',0.00025,'ExecutionEnvironment','gpu');
layers = deeplabv3plusLayers([400 400], 3,"resnet18");
trainData = pixelLabelImageDatastore(gTruth);
myNet2 = trainNetwork(trainData, layers, option);

testImage = imageDatastore('C:\Users\kevin\Desktop\ProjectSignature\test\test400x400_seg_testing\');
test_data = load('C:\Users\kevin\Desktop\ProjectSignature\gtruth\test\test_gTruth.mat');
desired = pixelLabelDatastore(test_data.gTruth);
prediced = semanticseg(testImage,myNet2);

metrics = evaluateSemanticSegmentation(prediced,desired);
metrics.ClassMetrics
metrics.ConfusionMatrix
heatmap(["true" "fake" "back"], ["true" "fake" "back"],metrics.ConfusionMatrix.Variables )

N = 63;
x = labeloverlay(readimage(testImage, N),readimage(prediced,N));
y = readimage(desired,N) == 'bad';
figure,imshowpair(x,y,'montage')
```

Result of testing webcam



Segmentation Detection of signature

	True	Fake
True	3	2
Fake	0	5

Accuracy = 80% (15 sec each time)



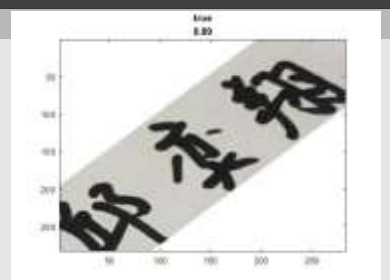




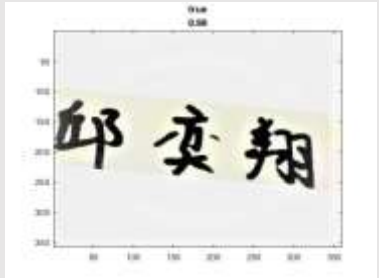


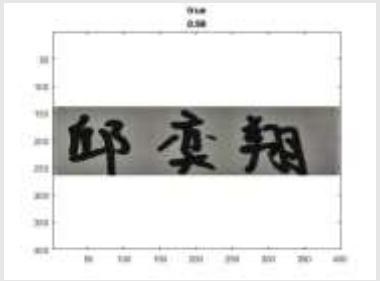
Result of Testing different environment of picture

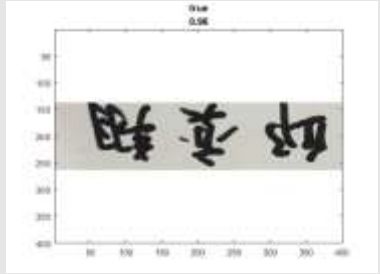
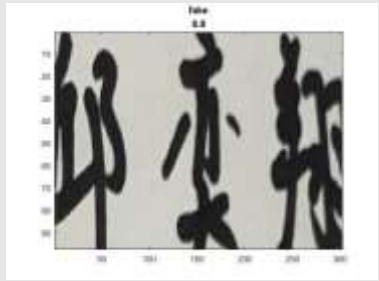






Result(ThingSpeak/*Excel*)

Webcam with fake signature

Conclusion

If this App is totally complete, it can apply to every kind of system which needs hand-writing signature, such as contract, bank procedure or credit card payment. It can be the institution which specialize in certificate signatures, like third party who has authority of identification.

Future Work

- To train a complete hand-wirting signature database needs at least thousands pieces of data ; however, our database is not quite enough to precisely detect real-time image especially the segamtation.
- In this stage, we can only detect one type of signature in the program.
- In different environment, it will reduce the accuracy of the detection,such as different shape of shadow, strong or dim light.

New function improvement

- We want to build an auto-filled system, but in the current stage we can only use Excel to display.
- Our real-time image is uploaded to ThingSpeak currently, we hope to have our own website to synchronize the image in the future.