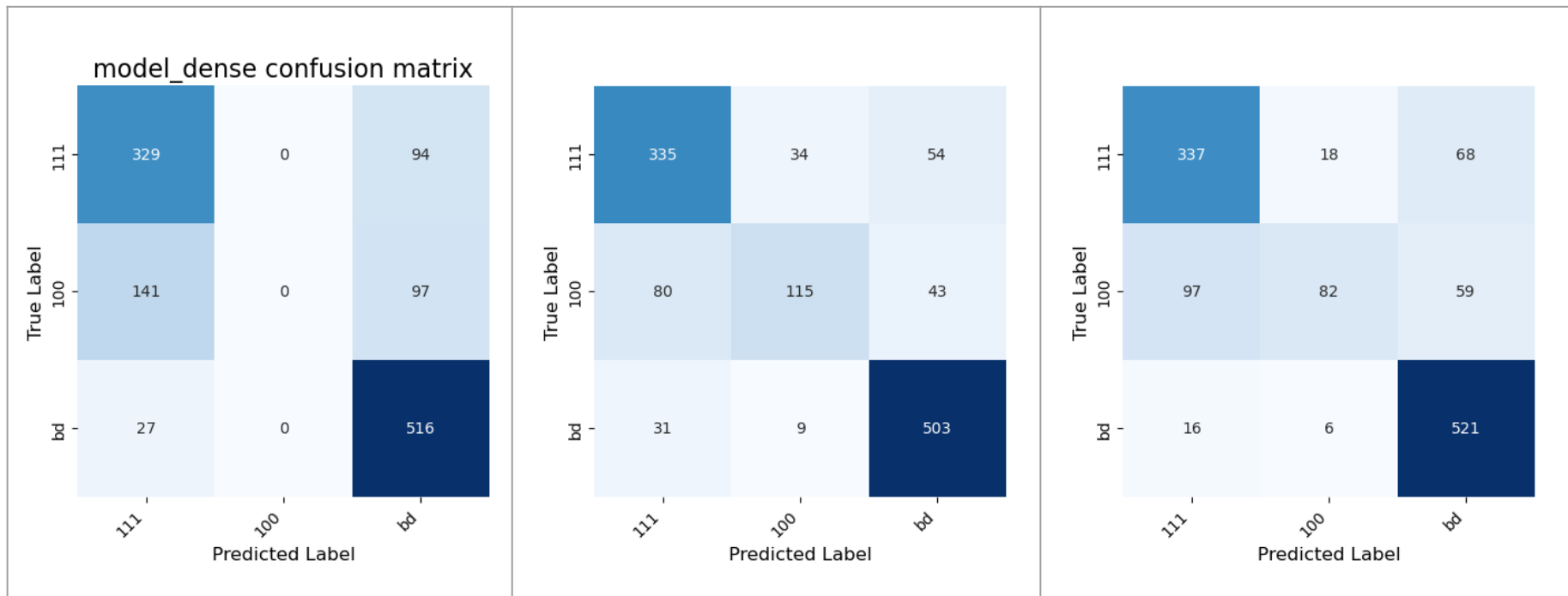


# Assessment Figures

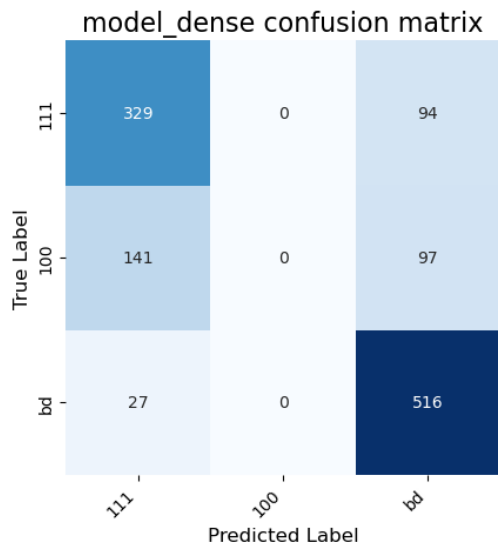


# ML4ER Assignment 7

Jiahui Yang  
Informatics Skunkworks  
MSE 401, 3 Credits  
Aug 07 2024

# Progress

- Your confusion matrix and F1 scores for Section 1 with the default Fully Connected Network (FCN)



The model has a validation accuracy of 70.18%, with high F1 scores for classes "111" (0.715) and "bd" (0.826), but the model fails to correctly predict any instances of class "100" (F1 score of 0). The average F1 score is 0.514.

Validation accuracy= 0.7018272425249169

F1 scores:

111 F1 = 0.7152173913043479

100 F1 = 0.0

bd F1 = 0.8255999999999999

Average F1 score = 0.5136057971014493

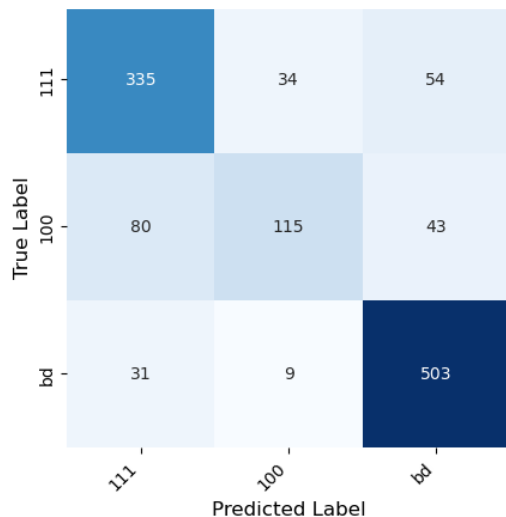
# Progress

The model has a validation accuracy of 79.15%, with high F1 scores for classes "111" (0.771), "bd" (0.880), and "100" (0.581). The average F1 score is 0.744.

- Your confusion matrix and F1 scores for exercise 2 - adding layers to your FCN

Model: "sequential\_7"

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 64, 64, 75)	750
batch_normalization_6 (Batch Normalization)	(None, 64, 64, 75)	300
max_pooling2d_6 (Max Pooling2D)	(None, 32, 32, 75)	0
conv2d_7 (Conv2D)	(None, 32, 32, 50)	33800
dropout_4 (Dropout)	(None, 32, 32, 50)	0
batch_normalization_7 (Batch Normalization)	(None, 32, 32, 50)	200
max_pooling2d_7 (Max Pooling2D)	(None, 16, 16, 50)	0
conv2d_8 (Conv2D)	(None, 16, 16, 25)	11275
batch_normalization_8 (Batch Normalization)	(None, 16, 16, 25)	100
...		
Total params: 58642 (229.07 KB)		
Trainable params: 58292 (227.70 KB)		
Non-trainable params: 350 (1.37 KB)		



Validation accuracy= 0.7915282392026578

F1 scores:

111 F1 = 0.7710011507479861

100 F1 = 0.5808080808080808

bd F1 = 0.8801399825021873

Average F1 score = 0.7439830713527514

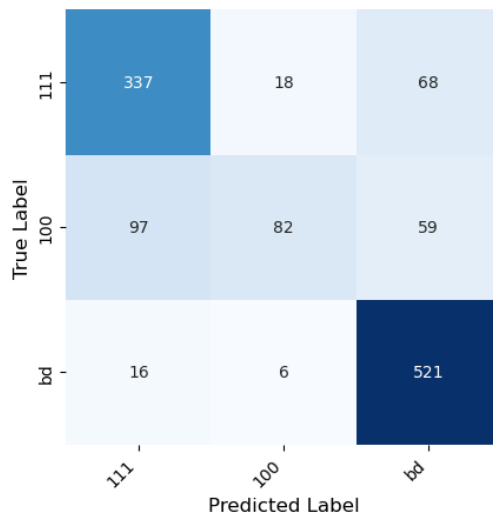
# Progress

The model has a validation accuracy of 78.01%, with high F1 scores for classes "111" (0.772), "bd" (0.875), and "100" (0.477). The average F1 score is 0.744.

- Your confusion matrix and F1 scores for exercise 3 - modifying convolution layers

Model: "sequential\_8"

Layer (type)	Output Shape	Param #
conv2d_10 (Conv2D)	(None, 64, 64, 64)	1664
batch_normalization_10 (Batch Normalization)	(None, 64, 64, 64)	256
max_pooling2d_10 (Max Pooling2D)	(None, 32, 32, 64)	0
conv2d_11 (Conv2D)	(None, 32, 32, 128)	73856
dropout_5 (Dropout)	(None, 32, 32, 128)	0
batch_normalization_11 (Batch Normalization)	(None, 32, 32, 128)	512
max_pooling2d_11 (Max Pooling2D)	(None, 16, 16, 128)	0
conv2d_12 (Conv2D)	(None, 16, 16, 256)	295168
batch_normalization_12 (Batch Normalization)	(None, 16, 16, 256)	1024
...		
Total params: 1421315 (5.42 MB)		
Trainable params: 1420419 (5.42 MB)		
Non-trainable params: 896 (3.50 KB)		



F1 scores:

111 F1 = 0.7720504009163802

100 F1 = 0.47674418604651153

bd F1 = 0.874895046179681

Average F1 score = 0.7078965443808576

# Problems

- I was running the Jupyter notebook on my local machine and had to manually change several paths. Below is a screenshot as an example.



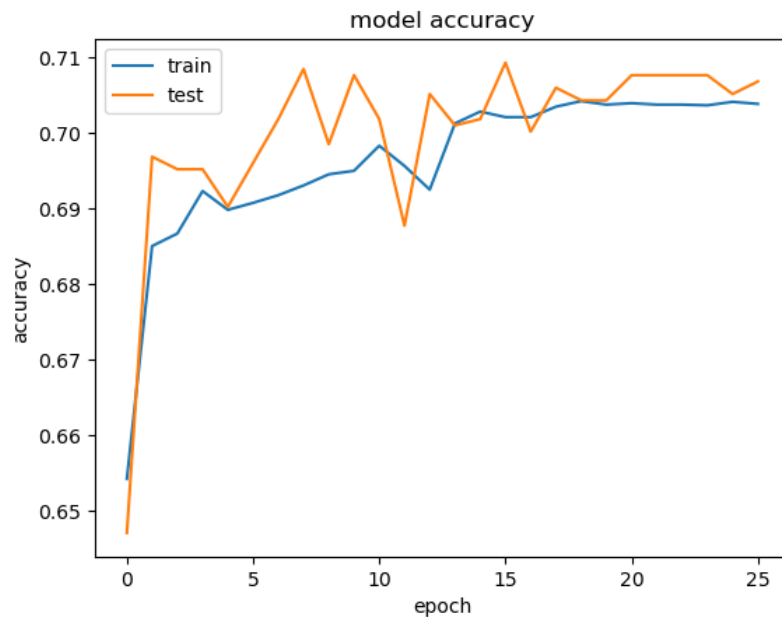
```
#from google.colab.patches import cv2_imshow
import cv2
import pandas as pd
RESIZE_PX = 480

im_path = r'c:/Users/jhyan/Documents/all/UW-Madison/course_enroll/MSE_401/Module_5/yolo_loops_data/train/images'
anno_path = r'c:/Users/jhyan/Documents/all/UW-Madison/course_enroll/MSE_401/Module_5/yolo_loops_data/train/labels'
train_images = os.listdir(im_path)
train_im = np.random.choice(train_images)
im_name = train_im.split('.')[0]
train_anno = os.path.join(anno_path, im_name+'.txt')
#anno = np.loadtxt(train_anno)
anno = pd.read_csv(train_anno, header=None)
im = Image.open(os.path.join(im_path, train_im)).convert('RGB')
im_resize = cv2.resize(np.array(im), (RESIZE_PX, RESIZE_PX))
#cv2_imshow(np.array(im_resize))
```

✓ 0.0s

# Questions

- How to further enhance model performance and improve its accuracy?



# Hours Summary

Date	Hours	Description of Work
08/05/2024	3 hours	Work Through the "ML4M - Image Data Activities.ipynb" notebook and follow along with the recording