Anthony Samagaio
Aryan Gupta
Sam Xu
ECGR 4090-JR1
05/13/21

Final Project: Object Detection

Introduction:

The objective of this project was to train and deploy an object detection algorithm, using the Arduino Nano 33 BLE Sense with the OV7675 camera. The goal was to choose a specific target other than a person such as cars, dogs, shoes, planes, etc. An example code was provided that was made as a person detection model.

The project was divided into several milestones to check the compatibility of code, train the model with enough data, and accuracy of the camera to ensure correct reads. The target chosen by the team was the fruit “banana”. Several datasets were taken to train the model, including Fruit 360, Coco 2017, and custom datasets such as pictures, and videos of bananas. Afterwards, the Arduino code was compiled into the Arduino board and the OV7675 camera to implement the trained model to identify if the target was a banana or not a banana.

Model Architecture:

The Mobile Net v1 model architecture was chosen for this project because it was the default in the MLCommons that was given to us as a reference for this project. **Full layer structure can be found in Appendix A.**

Data and Training:

At the beginning of the project many datasets were taken from online resources such as Fruit 360 and Coco 2017 where a combination of 1936 images of bananas and 39,347 images of non bananas were used for training the model the first time. These datasets were implemented into the Arduino Nano 33 BLE Sense board with the OV7675 camera. After training the model for the first time, we achieved an accuracy rate of 83% or sometimes the model was overfitting. As the first model had more non banana images we took the next step to re-train the model with more banana images.

A custom image dataset was made by recording a 6 minute video of real bananas in various environments in order to train the model and achieve a higher accuracy. After splitting each individual frame of the video a total of 29,219 images of bananas were trained into the new model. An accuracy rate of 99% was

presented with the new model. With the increase of banana images (almost matching non banana images) the model successfully identified between a banana and a non banana between 3 to 5 seconds.
Training data can be found in Appendix B.

Results:

epoch	loss	accuracy	val_loss	val_accuracy
15/100	**0.0297**	**0.9954**	**0.6286**	**0.7919**
100/100	**0.0040**	**0.9999**	**2.5840**	**0.7280**

More data can be found in Appendix B

When the model was first trained the accuracy rate was about 83% and it would recognize a banana in color and grayscale but at a slow rate. After re-training the model with more datasets, the model as shown above had an accuracy rate of 99% and would detect the banana much faster than the first model. The frame rate was slow for both models but it was nocitible when the model was re-trained that the frame rate was faster than the first model.

From the first model, the camera would take about 8 to 10 seconds to recognize the banana, while in the re-trained model the camera would only take about 3 to 5 seconds to recognize the banana. If the camera was not directed at a banana then the model would exhibit a non banana on the terminal. The main concept that would interest users would be how the trained model could detect a banana if a user was holding it with a decent accuracy rate. Also, for future experiments the model could be trained to detect other fruits or targets within an image.

Discussion:

Throughout the project, there were several issues presented such as, finding a starting point, outdated version of TensorFlowLite, timing the upload with the reset button, and more data needed. After finding solutions to these issues like setting a timer at the beginning, retraining the model with more data (live pictures of bananas), updating TensorFlowLite, the model worked as expected and had better accuracy after being re-trained. With more time and resources an even more accurate model could have been created and a higher accuracy rate could be achieved. These resources could include more banana pictures of different colors and scenarios, more datasets, and more trial runs to test the model.

This object detection model could have much potential in the future such as a grocery notifier where it would alert the store if a certain item (fruits, vegetables, etc.) was running low or even a camera inside a fridge that would tell the user if a certain item was running low when you entered a grocery store. Another implementation of this could be a ripeness detection where it could detect the color difference between a ripe and expired fruits and/or vegetables.

Throughout the project we encountered an outdated TensorFlowLite and compatibility between the Arduino code and OV7675 camera. In the future we will make sure to update the TensorFlow version and TFlite version up to date and learning from these breakthroughs we could go back to previous projects and also for future projects.

Conclusion:

Throughout this project, we realized that the accuracy of the trained model depended on the amount of data fed to the model. While training the model for the first time, no custom dataset (videos, pictures) were used and only online datasets were used (1936 banana images, 39,347 not banana images). This model gave us an 83% accuracy rate or the model was overfitting. When the model was re-trained, custom datasets (27,283 banana images) were used giving us a total of 29,219 banana images and 39,347 not banana images. This new model gave us an accuracy rate of 99% and could identify the target between 3 to 5 seconds.

Overall this project presented us with the opportunity to keep growing our model and try different targets such as vehicles, animals, more fruits or vegetables, with the end result to create something innovative to present to the world. In conclusion, the project taught us that a simple camera and a small board have the power to tell the difference between specific targets.

Summary table:

Images:
Custom:

Coco:

Bananas
per kg
$11.98 kg
$11.98 per kg
PRODUCT OF AUSTRALIA
Banana Upd
80% of Australia's bananas were wiped o
Cyclone Yasi. Due to the short supply you
retail prices rising. We will continue to su
growers in order to help improve supply.
you for your support during this time.

Fruit 360:

Appendix A

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 96, 96, 3)]       0
_________________________________________________________________
conv2d (Conv2D)              (None, 48, 48, 8)         224
_________________________________________________________________
batch_normalization (BatchNo (None, 48, 48, 8)         32
_________________________________________________________________
activation (Activation)      (None, 48, 48, 8)         0
_________________________________________________________________
depthwise_conv2d (DepthwiseC (None, 48, 48, 8)         80
_________________________________________________________________
batch_normalization_1 (Batch (None, 48, 48, 8)         32
_________________________________________________________________
activation_1 (Activation)    (None, 48, 48, 8)         0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 48, 48, 16)        144
_________________________________________________________________
batch_normalization_2 (Batch (None, 48, 48, 16)        64
_________________________________________________________________
activation_2 (Activation)    (None, 48, 48, 16)        0
_________________________________________________________________
depthwise_conv2d_1 (Depthwis (None, 24, 24, 16)        160
_________________________________________________________________
batch_normalization_3 (Batch (None, 24, 24, 16)        64
_________________________________________________________________
activation_3 (Activation)    (None, 24, 24, 16)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 24, 24, 32)        544
_________________________________________________________________
batch_normalization_4 (Batch (None, 24, 24, 32)        128
_________________________________________________________________
activation_4 (Activation)    (None, 24, 24, 32)        0
_________________________________________________________________
depthwise_conv2d_2 (Depthwis (None, 24, 24, 32)        320
_________________________________________________________________
batch_normalization_5 (Batch (None, 24, 24, 32)        128
_________________________________________________________________
activation_5 (Activation)    (None, 24, 24, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 24, 24, 32)        1056
_________________________________________________________________
batch_normalization_6 (Batch (None, 24, 24, 32)        128
_________________________________________________________________
activation_6 (Activation)    (None, 24, 24, 32)        0
_________________________________________________________________
depthwise_conv2d_3 (Depthwis (None, 12, 12, 32)        320
_________________________________________________________________
batch_normalization_7 (Batch (None, 12, 12, 32)        128
_________________________________________________________________
activation_7 (Activation)    (None, 12, 12, 32)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 12, 12, 64)        2112
_________________________________________________________________
batch_normalization_8 (Batch (None, 12, 12, 64)        256
_________________________________________________________________
activation_8 (Activation)    (None, 12, 12, 64)        0
_________________________________________________________________
depthwise_conv2d_4 (Depthwis (None, 12, 12, 64)        640
_________________________________________________________________
batch_normalization_9 (Batch (None, 12, 12, 64)        256
_________________________________________________________________
activation_9 (Activation)    (None, 12, 12, 64)        0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 12, 12, 64)        4160
_________________________________________________________________
batch_normalization_10 (Batc (None, 12, 12, 64)        256
_________________________________________________________________
activation_10 (Activation)   (None, 12, 12, 64)        0
_________________________________________________________________
depthwise_conv2d_5 (Depthwis (None, 6, 6, 64)          640
_________________________________________________________________
batch_normalization_11 (Batc (None, 6, 6, 64)          256
_________________________________________________________________
activation_11 (Activation)   (None, 6, 6, 64)          0         
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 6, 6, 128)         8320      
_________________________________________________________________
batch_normalization_12 (Batc (None, 6, 6, 128)         512       
_________________________________________________________________
activation_12 (Activation)   (None, 6, 6, 128)         0         
_________________________________________________________________
depthwise_conv2d_6 (Depthwis (None, 6, 6, 128)         1280      
_________________________________________________________________
batch_normalization_13 (Batc (None, 6, 6, 128)         512       
_________________________________________________________________
activation_13 (Activation)   (None, 6, 6, 128)         0         
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 6, 6, 128)         16512     
_________________________________________________________________
batch_normalization_14 (Batc (None, 6, 6, 128)         512       
_________________________________________________________________
activation_14 (Activation)   (None, 6, 6, 128)         0         
_________________________________________________________________
depthwise_conv2d_7 (Depthwis (None, 6, 6, 128)         1280      
_________________________________________________________________
batch_normalization_15 (Batc (None, 6, 6, 128)         512       
_________________________________________________________________
activation_15 (Activation)   (None, 6, 6, 128)         0         
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 6, 6, 128)         16512     
_________________________________________________________________
batch_normalization_16 (Batc (None, 6, 6, 128)         512       
_________________________________________________________________
activation_16 (Activation)   (None, 6, 6, 128)         0         
_________________________________________________________________
depthwise_conv2d_8 (Depthwis (None, 6, 6, 128)         1280      
_________________________________________________________________
batch_normalization_17 (Batc (None, 6, 6, 128)         512       
_________________________________________________________________
activation_17 (Activation)   (None, 6, 6, 128)         0         
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 6, 6, 128)         16512     
_________________________________________________________________
batch_normalization_18 (Batc (None, 6, 6, 128)         512       
_________________________________________________________________
activation_18 (Activation)   (None, 6, 6, 128)         0         
_________________________________________________________________
depthwise_conv2d_9 (Depthwis (None, 6, 6, 128)         1280      
_________________________________________________________________
batch_normalization_19 (Batc (None, 6, 6, 128)         512       
_________________________________________________________________
activation_19 (Activation)   (None, 6, 6, 128)         0         
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 6, 6, 128)         16512     
_________________________________________________________________
batch_normalization_20 (Batc (None, 6, 6, 128)         512       
_________________________________________________________________
activation_20 (Activation)   (None, 6, 6, 128)         0         
_________________________________________________________________
depthwise_conv2d_10 (Depthwi (None, 6, 6, 128)         1280      
_________________________________________________________________
batch_normalization_21 (Batc (None, 6, 6, 128)         512       
_________________________________________________________________
activation_21 (Activation)   (None, 6, 6, 128)         0
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 6, 6, 128)         16512
_________________________________________________________________
batch_normalization_22 (Batc (None, 6, 6, 128)         512
_________________________________________________________________
activation_22 (Activation)   (None, 6, 6, 128)         0
_________________________________________________________________
depthwise_conv2d_11 (Depthwi (None, 3, 3, 128)         1280
_________________________________________________________________
batch_normalization_23 (Batc (None, 3, 3, 128)         512
_________________________________________________________________
activation_23 (Activation)   (None, 3, 3, 128)         0
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 3, 3, 256)         33024
_________________________________________________________________
batch_normalization_24 (Batc (None, 3, 3, 256)         1024
_________________________________________________________________
activation_24 (Activation)   (None, 3, 3, 256)         0
_________________________________________________________________
depthwise_conv2d_12 (Depthwi (None, 3, 3, 256)         2560
_________________________________________________________________
batch_normalization_25 (Batc (None, 3, 3, 256)         1024
_________________________________________________________________
activation_25 (Activation)   (None, 3, 3, 256)         0
_________________________________________________________________
conv2d_13 (Conv2D)           (None, 3, 3, 256)         65792
_________________________________________________________________
batch_normalization_26 (Batc (None, 3, 3, 256)         1024
_________________________________________________________________
activation_26 (Activation)   (None, 3, 3, 256)         0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 1, 1, 256)         0
_________________________________________________________________
flatten (Flatten)            (None, 256)               0
_________________________________________________________________
dense (Dense)                (None, 2)                 514
=================================================================
Total params: 221,794
Trainable params: 216,322
Non-trainable params: 5,472
_________________________________________________________________
```

Appendix B

```
Epoch 1/35
419/419 [==============================] - 427s 979ms/step - loss: 0.5172 - accuracy: 0.9277 - val_loss: 0.9622 - val_accuracy: 0.6170
Epoch 2/35
419/419 [==============================] - 404s 965ms/step - loss: 0.1810 - accuracy: 0.9896 - val_loss: 1.3256 - val_accuracy: 0.7206
Epoch 3/35
419/419 [==============================] - 405s 967ms/step - loss: 0.1066 - accuracy: 0.9938 - val_loss: 3.4771 - val_accuracy: 0.7416
Epoch 4/35
419/419 [==============================] - 403s 963ms/step - loss: 0.0943 - accuracy: 0.9930 - val_loss: 1.0410 - val_accuracy: 0.7266
Epoch 5/35
419/419 [==============================] - 403s 962ms/step - loss: 0.0590 - accuracy: 0.9963 - val_loss: 4.6297 - val_accuracy: 0.5652
Epoch 6/35
419/419 [==============================] - 402s 960ms/step - loss: 0.0574 - accuracy: 0.9938 - val_loss: 1.4963 - val_accuracy: 0.7531
Epoch 7/35
419/419 [==============================] - 403s 963ms/step - loss: 0.0565 - accuracy: 0.9942 - val_loss: 5.3315 - val_accuracy: 0.7258
Epoch 8/35
419/419 [==============================] - 405s 968ms/step - loss: 0.0486 - accuracy: 0.9939 - val_loss: 0.6135 - val_accuracy: 0.7365
Epoch 9/35
419/419 [==============================] - 403s 962ms/step - loss: 0.0469 - accuracy: 0.9946 - val_loss: 12.3067 - val_accuracy: 0.5095
```

```
Epoch 10/35
419/419 [==============================] - 402s 961ms/step - loss: 0.0436 - accuracy: 0.9943 - val_loss: 0.6249 - val_accuracy: 0.7496
Epoch 11/35
419/419 [==============================] - 401s 957ms/step - loss: 0.0415 - accuracy: 0.9942 - val_loss: 3.4745 - val_accuracy: 0.7034
Epoch 12/35
419/419 [==============================] - 401s 956ms/step - loss: 0.0323 - accuracy: 0.9957 - val_loss: 1.8153 - val_accuracy: 0.7649
Epoch 13/35
419/419 [==============================] - 401s 957ms/step - loss: 0.0301 - accuracy: 0.9953 - val_loss: 1.1057 - val_accuracy: 0.5095
Epoch 14/35
419/419 [==============================] - 401s 957ms/step - loss: 0.0296 - accuracy: 0.9961 - val_loss: 0.7684 - val_accuracy: 0.5763
Epoch 15/35
419/419 [==============================] - 401s 957ms/step - loss: 0.0297 - accuracy: 0.9954 - val_loss: 0.6286 - val_accuracy: 0.7919
Epoch 00015: early stopping
Epoch 1/15
419/419 [==============================] - 411s 977ms/step - loss: 0.0220 - accuracy: 0.9974 - val_loss: 0.7688 - val_accuracy: 0.7295
Epoch 2/15
419/419 [==============================] - 403s 962ms/step - loss: 0.0145 - accuracy: 0.9987 - val_loss: 0.8030 - val_accuracy: 0.7290
Epoch 3/15
419/419 [==============================] - 401s 957ms/step - loss: 0.0120 - accuracy: 0.9992 - val_loss: 1.4409 - val_accuracy: 0.7348
Epoch 4/15
419/419 [==============================] - 402s 961ms/step - loss: 0.0111 - accuracy: 0.9994 - val_loss: 3.7050 - val_accuracy: 0.7117
Epoch 5/15
419/419 [==============================] - 406s 970ms/step - loss: 0.0096 - accuracy: 0.9994 - val_loss: 1.9013 - val_accuracy: 0.7280
Epoch 6/15
419/419 [==============================] - 406s 969ms/step - loss: 0.0106 - accuracy: 0.9988 - val_loss: 0.7006 - val_accuracy: 0.7312
Epoch 7/15
419/419 [==============================] - 407s 972ms/step - loss: 0.0094 - accuracy: 0.9994 - val_loss: 3.5871 - val_accuracy: 0.7471
Epoch 8/15
419/419 [==============================] - 407s 970ms/step - loss: 0.0093 - accuracy: 0.9992 - val_loss: 4.2158 - val_accuracy: 0.7317
Epoch 9/15
419/419 [==============================] - 406s 969ms/step - loss: 0.0084 - accuracy: 0.9994 - val_loss: 4.2546 - val_accuracy: 0.7122
Epoch 10/15
419/419 [==============================] - 407s 973ms/step - loss: 0.0086 - accuracy: 0.9992 - val_loss: 1.8448 - val_accuracy: 0.7320
Epoch 11/15
419/419 [==============================] - 408s 974ms/step - loss: 0.0073 - accuracy: 0.9997 - val_loss: 1.7824 - val_accuracy: 0.7288
Epoch 12/15
419/419 [==============================] - 408s 974ms/step - loss: 0.0074 - accuracy: 0.9995 - val_loss: 2.8048 - val_accuracy: 0.7264
Epoch 13/15
419/419 [==============================] - 408s 973ms/step - loss: 0.0070 - accuracy: 0.9995 - val_loss: 3.7779 - val_accuracy: 0.7338
Epoch 00013: early stopping
Epoch 1/15
419/419 [==============================] - 410s 973ms/step - loss: 0.0072 - accuracy: 0.9995 - val_loss: 1.7944 - val_accuracy: 0.7328
Epoch 2/15
419/419 [==============================] - 404s 965ms/step - loss: 0.0061 - accuracy: 0.9997 - val_loss: 1.8704 - val_accuracy: 0.7318
Epoch 3/15
419/419 [==============================] - 405s 967ms/step - loss: 0.0068 - accuracy: 0.9995 - val_loss: 2.2862 - val_accuracy: 0.7330
Epoch 4/15
419/419 [==============================] - 406s 969ms/step - loss: 0.0055 - accuracy: 0.9998 - val_loss: 2.6971 - val_accuracy: 0.7313
Epoch 5/15
419/419 [==============================] - 408s 975ms/step - loss: 0.0055 - accuracy: 0.9998 - val_loss: 2.8443 - val_accuracy: 0.7301
Epoch 6/15
419/419 [==============================] - 410s 978ms/step - loss: 0.0056 - accuracy: 0.9997 - val_loss: 2.6231 - val_accuracy: 0.7322
Epoch 7/15
419/419 [==============================] - 410s 979ms/step - loss: 0.0055 - accuracy: 0.9997 - val_loss: 3.5086 - val_accuracy: 0.7105
Epoch 8/15
419/419 [==============================] - 411s 981ms/step - loss: 0.0057 - accuracy: 0.9997 - val_loss: 2.4662 - val_accuracy: 0.7353
Epoch 00008: early stopping
Epoch 1/35
419/419 [==============================] - 415s 985ms/step - loss: 0.0065 - accuracy: 0.9994 - val_loss: 2.5085 - val_accuracy: 0.7313
Epoch 2/35
419/419 [==============================] - 410s 978ms/step - loss: 0.0051 - accuracy: 0.9998 - val_loss: 2.8460 - val_accuracy: 0.7315
Epoch 3/35
419/419 [==============================] - 410s 978ms/step - loss: 0.0046 - accuracy: 1.0000 - val_loss: 2.3896 - val_accuracy: 0.7303
Epoch 4/35
419/419 [==============================] - 409s 977ms/step - loss: 0.0045 - accuracy: 0.9999 - val_loss: 2.8648 - val_accuracy: 0.7310
Epoch 5/35
419/419 [==============================] - 410s 978ms/step - loss: 0.0046 - accuracy: 0.9999 - val_loss: 2.9836 - val_accuracy: 0.7312
Epoch 6/35
419/419 [==============================] - 409s 977ms/step - loss: 0.0046 - accuracy: 0.9999 - val_loss: 2.3547 - val_accuracy: 0.7308
Epoch 7/35
419/419 [==============================] - 409s 975ms/step - loss: 0.0043 - accuracy: 0.9999 - val_loss: 2.7359 - val_accuracy: 0.7281
Epoch 8/35
419/419 [==============================] - 409s 977ms/step - loss: 0.0042 - accuracy: 0.9999 - val_loss: 3.6926 - val_accuracy: 0.7305
Epoch 9/35
419/419 [==============================] - 409s 976ms/step - loss: 0.0042 - accuracy: 0.9999 - val_loss: 3.2632 - val_accuracy: 0.7285
Epoch 10/35
419/419 [==============================] - 409s 975ms/step - loss: 0.0042 - accuracy: 0.9998 - val_loss: 3.1209 - val_accuracy: 0.7298
Epoch 11/35
419/419 [==============================] - 409s 975ms/step - loss: 0.0041 - accuracy: 0.9999 - val_loss: 3.3544 - val_accuracy: 0.7295
Epoch 12/35
419/419 [==============================] - 408s 975ms/step - loss: 0.0039 - accuracy: 1.0000 - val_loss: 3.5266 - val_accuracy: 0.7335
Epoch 13/35
419/419 [==============================] - 408s 974ms/step - loss: 0.0040 - accuracy: 0.9999 - val_loss: 2.5840 - val_accuracy: 0.7280
Epoch 00013: early stopping
```

References:

Mlcommons. (n.d.). Mlcommons/tiny. Retrieved May 13, 2021, from https://github.com/mlcommons/tiny/tree/master/v0.1/training/visual_wake_words

Oltean, M. (2020, May 18). Fruits 360. Retrieved May 13, 2021, from https://www.kaggle.com/moltean/fruits

Coco : Tensorflow datasets. (n.d.). Retrieved May 13, 2021, from https://www.tensorflow.org/datasets/catalog/coco