

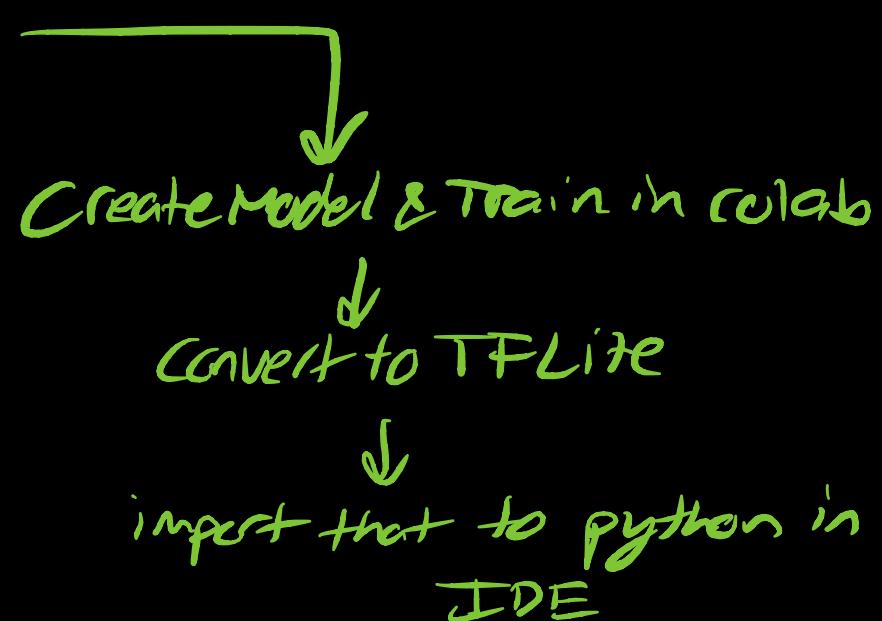
To Toss or Not → Changelog

3/24

- Tried pushing to git, messed up the master repo
- In Colab environment,
 - imported kaggle + kaggle.json ← so annoying
 - Downloaded the waste classification dataset
 - extract
 - Set into variables

3/25

- Importing Kaggle into Colab is sm easier Now
 widget to upload json file
- CNN Model created
- TensorBoard connected
- Predictions created
- Figured out how to save model & save weights
- Figured out how to import model somewhere else



Notes

- Look into shuffling data (flow_from_directory)
- How to improve accuracy / minimize loss
 - prevent overfitting
 - warping images (zoom, rotate, etc.)
- Image Preprocessing
- Using transfer learning
- How to display Recyclable or not based on probability
 - ↳ how to interpret outputted accuracy/loss
 - ↳ how to predict on a single picture (do I need to import saved model then use?)
- ↳ POST TFLITE conversion?
- ↳ is it any different

What's the diff between:

Model.compile (loss = 'binary_crossentropy')

8

↓ This signifies that

Model.compile (loss = tf.keras.losses.BinaryCrossentropy
(from_logits = true))

'binary_crossentropy' = tf.keras.losses.BinaryCrossentropy
(from_logits = false)

[R , O]

3/26

• Updated yesterday's,

→ added tf.Data preprocessing functionality

- IDK how useful but maybe needed

- not integrated w/ the rest

→ Turned models into functions

• Added Data Augmentation + Dropout (to decrease overfitting)

• Used for importing pre-trained CNN Model (Mobilenet)

Notes

- Look into shuffling data (flow-from directory) added tf.data functionality
- How to improve accuracy / minimize loss
 - prevent overfitting (Data Aug + Dropouts added)
 - warping images (zoom, rotate, etc) Image Preprocessing
 - + from_logits := True vs = False (what's the diff / is it useful) Used for datasets / practical CNN
- How to display Recyclable or not based on probability
 - ↳ how to interpret outputted accuracy / loss 0 = organic, 1 = recyclable
 - ↳ how to predict on a single picture (do I need to import saved model then use?)
- +
 - 1 created functionality, but is it accurate??!
 - 2 POST TFLITE conversion?
↳ is it any different
- Make constants for batch-size, width, height, epochs
- What's the diff b/w optimizers (RMSProp vs SGD vs Adam)
 - o What's most efficient
 - o run hello tests & display data
- How to save tensorboard outputs

ACCURACY IS GARBAGE

→ Need more ~~longer~~

→ use pre-existing model
+ transfer learning

DO THIS
PDSJ

~~↓
Fine-tuning~~

Using
MobileNet
now

~~(Convert from logits
to prediction)~~

Need better data

(Try with Recycle/non-recycle
dataset)

Need to change all image sizes

to 128×128

3/27

→ Tried on Recyclable vs Non Recyclable Dataset
& headers w/ type of file inserted
• Use method used in waste v3 to delete
non jpg, png, jpg files
→ Validation accuracy decreased
+ loss inc,
while training acc inc &
loss dec

- Need to start making a bunch of diff models, recording their data & changed variables, & storing their graphs to find the best model

TensorBoard?

- Look up things I can tweak in the model

- Look up what is recyclable & what is NOT

- Do I need to look into Inception or ResNet?

- Diff b/w optimizers

Making a chart to roughly possibilities

& their data

→ Datascraping to make my own dataset?

→ Inception V3 seems to have best accuracy

- Adjust augmentations to fit better

→ Garbage Classification Data

→ cardboard

→ glass

→ metal

→ paper

→ plastic

→ trash

Currently training a model to classify into these categories on the left

Need to export (save model and save weights)

This model will classify

Need to list out details of each model:

Dataset Trained On (Link)

Base Pretrained Model Used

Layers added on top

Optimizer used

Loss function used

Augmented/No Augmentation

Need to export the following

SavedModel.h5

SavedWeights.h5

TrainingScript.py

CallModel.py

USE THE FREAKING
IMAGE
DATAGENERATOR
FROM THIS
INCEPTION 6 CLASS
MODEL ON THE
RECYCLE VS NON
RECYCLE DATASET
AND RUN A TEST

WasteData (3 classes) (Mobilenet)

- LR x 100 ✓
- LR x 50 ✓
- LR x 25 ✓
- LR x 75? ✓
- LR ✓
- LR / 10 ✓
- LR / 25
- LR / 50 } Didn't
- LR / 100 } Do
- LR / 75

choose best

Dropout
Added

1000 epochs + LR

1000 epochs + LR x 25

1000 epochs + LR x 50

1000 epochs + LR x 75

" " + LR x 100

" " + LR / 10

" " + LR / 25

" " + LR / 75

Dropout
is a form
of regularization

Data AUG

Regularization

{ choose best

- optimizer =
- optimizer =
- optimizer =

After running
'1000 epochs + LR',
figure out which

choose
best

→ change freezable
layers

JOIN THE DARKSIDE

(Steps per epoch • batch size = total # of images)

epochs = above process # of times

224x224

pre & post augmentation
to compare

CURRENTLY
RUNNING:

Next :