

## 2.3. Model II

```
In [ ]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf
import cv2
from sklearn.model_selection import train_test_split

from zipfile import ZipFile
import os
import sys

# Load modules from the lib directory
sys.path.insert(0, "../lib")
from resnet import load_resnet, ResNet18
from lossFunctions import get_custom_cross_entropy, l1_loss
from model2 import LabelCleaner, ImageClassifier
```

```
In [ ]: if not os.path.exists("./drive"):
    drive.mount('/content/drive')

if not os.path.exists("../data"):
    os.mkdir("../data")

# Loading the temp.zip and creating a zip object
with ZipFile("./drive/MyDrive/train_data.zip", 'r') as zip_object:

    # Extracting all the members of the zip
    # into a specific location.
    zip_object.extractall(path="../data")
```

```
In [ ]: # [DO NOT MODIFY THIS CELL]

n_images: int = 50_000
n_noisy: int = 40_000
n_clean: int = n_images - n_noisy

images : np.ndarray = np.empty((n_images, 32, 32, 3), dtype=np.float32)

# Load the data
for i in range(n_images):
    image_path = f"../data/images/{i+1:05d}.png"
    images[i,:,:,:] = cv2.cvtColor(cv2.imread(image_path), cv2.COLOR_BGR2RGB)

# Load the Labels
clean_labels = np.genfromtxt('../data/clean_labels.csv', delimiter=',', dtype="int8")
noisy_labels = np.genfromtxt('../data/noisy_labels.csv', delimiter=',', dtype="int8")
```

```
In [ ]: test_ratio: float = 0.2
train_size: float = n_images - (n_clean * test_ratio)
clean_noisy_ratio: float = 1 / 9
train_clean_size: int = int(np.floor(train_size * clean_noisy_ratio))
val_clean_size: int = int(np.floor((n_clean * (1 - test_ratio)) - train_clean_size))
```

```
test_clean_size: int = n_clean - train_clean_size - val_clean_size

IMG_SIZE: int = 32
IMG_SHAPE: tuple = (IMG_SIZE, IMG_SIZE, 3)

BATCH_SIZE: int = 128
```

```
In [ ]: images_normalized = tf.cast(images, dtype = tf.float32) / 255.0
clean_labels_one_hot = tf.one_hot(clean_labels, depth = 10)
noisy_labels_one_hot = tf.one_hot(noisy_labels, depth = 10)
```

```
In [ ]: x_clean = images_normalized[:n_clean]
y_clean = clean_labels_one_hot
x_noisy = images_normalized[n_clean:]
y_noisy = noisy_labels_one_hot
```

```
In [ ]: x_clean_train_full, x_clean_test, y_clean_train_full, y_clean_test = train_test_split(
```

```
In [ ]: x_clean_train, x_clean_val, y_clean_train, y_clean_val = train_test_split(x_clean_train,
```

```
In [ ]: x_clean_train_size = len(x_clean_train)
x_clean_val_size = len(x_clean_val)
x_clean_test_size = len(x_clean_test)
```

```
In [ ]: V = tf.data.Dataset.from_tensor_slices((
    (
        x_clean_train_full,
        noisy_labels_one_hot[:len(y_clean_train_full)])
    ),
    y_clean_train_full
))
V_test = tf.data.Dataset.from_tensor_slices((
    (
        x_clean_test,
        noisy_labels_one_hot[x_clean_train_size + x_clean_val_size:10000])
    ),
    y_clean_test
)).batch(batch_size = BATCH_SIZE)
```

```
In [ ]: V_train = V.take(x_clean_train_size).batch(batch_size = BATCH_SIZE)
V_val = V.skip(x_clean_train_size).take(x_clean_val_size).batch(batch_size = BATCH_SIZE)
del V
```

```
In [ ]: es = tf.keras.callbacks.EarlyStopping(patience = 10, restore_best_weights = True)
```

```
In [ ]: cnn_1 = load_resnet(10, (32, 32, 3))
cnn_1.compile(
    loss = tf.keras.losses.CategoricalCrossentropy(),
    optimizer = tf.keras.optimizers.Adam(learning_rate = 0.001),
    metrics = ["accuracy"])
)
cnn_1.trainable = False
```

```
In [ ]: cnn_1.evaluate(x_clean_test, y_clean_test)
```

WARNING:tensorflow:Detecting that an object or model or tf.train.Checkpoint is being deleted with unrestored values. See the following logs for the specific values in question. To silence these warnings, use `status.expect\_partial()`. See [https://www.tensorflow.org/api\\_docs/python/tf/train/Checkpoint#restorefor](https://www.tensorflow.org/api_docs/python/tf/train/Checkpoint#restorefor) details about the status object returned by the restore function.

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer.iter

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer.beta\_1

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer.beta\_2

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer.decay

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer.learning\_rate

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).conv\_1.kernel

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).conv\_1.bias

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).init\_bn.gamma

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).init\_bn.beta

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).fc.kernel

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).fc.bias

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res\_1\_1.conv\_1.kernel

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res\_1\_1.conv\_1.bias

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res\_1\_1.bn\_1.gamma

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res\_1\_1.bn\_1.beta

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res\_1\_1.conv\_2.kernel

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res\_1\_1.conv\_2.bias

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res\_1\_1.bn\_2.gamma

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res\_1\_1.bn\_2.beta

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res\_1\_2.conv\_1.kernel

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res\_1\_2.conv\_1.bias

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res\_1\_2.bn\_1.gamma

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res\_1\_2.bn\_1.beta

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res\_1\_2.conv\_2.kernel

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res\_1\_2.conv\_2.bias

WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res\_1\_2.bn\_2.gamma











```
ot).optimizer's state 'v' for (root).res_4_2.bn_2.beta
WARNING:tensorflow:Detecting that an object or model or tf.train.Checkpoint is being deleted with unrestored values. See the following logs for the specific values in question. To silence these warnings, use `status.expect_partial()`. See https://www.tensorflow.org/api_docs/python/tf/train/Checkpoint#restorefor details about the status object returned by the restore function.
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer.iter
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer.beta_1
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer.beta_2
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer.decay
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer.learning_rate
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).conv_1.kernel
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).conv_1.bias
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).init_bn.gamma
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).init_bn.beta
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).fc.kernel
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).fc.bias
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res_1_1.conv_1.kernel
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res_1_1.conv_1.bias
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res_1_1.bn_1.gamma
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res_1_1.bn_1.beta
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res_1_1.conv_2.kernel
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res_1_1.conv_2.bias
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res_1_1.bn_2.gamma
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res_1_1.bn_2.beta
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res_1_2.conv_1.kernel
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res_1_2.conv_1.bias
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res_1_2.bn_1.gamma
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res_1_2.bn_1.beta
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res_1_2.conv_2.kernel
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res_1_2.conv_2.bias
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res_1_2.bn_2.gamma
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).res_1_2.bn_2.beta
```











```
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'v' for (root).res_4_2.bn_2.beta  
32/32 [=====] - 2s 10ms/step - loss: 0.3171 - accuracy: 0.91  
10  
Out[ ]: [0.31712672114372253, 0.9110000133514404]
```

```
In [ ]: cnn_1.trainable = False  
cleaner = LabelCleaner(cnn_1)  
  
cleaner.compile(  
    optimizer = tf.keras.optimizers.Adam(0.001),  
    loss = l1_loss,  
    metrics = ['accuracy'])
```

```
In [ ]: cleaner.fit(  
    V_train,  
    epochs = 60,  
    validation_data = V_val,  
    callbacks = [es])
```

Epoch 1/60  
57/57 [=====] - 3s 25ms/step - loss: 144.2127 - accuracy: 0.  
6825 - val\_loss: 147.1830 - val\_accuracy: 0.5156  
Epoch 2/60  
57/57 [=====] - 1s 14ms/step - loss: 40.1163 - accuracy: 0.8  
618 - val\_loss: 102.0742 - val\_accuracy: 0.7211  
Epoch 3/60  
57/57 [=====] - 1s 14ms/step - loss: 33.7128 - accuracy: 0.8  
629 - val\_loss: 76.0516 - val\_accuracy: 0.8339  
Epoch 4/60  
57/57 [=====] - 1s 14ms/step - loss: 32.9007 - accuracy: 0.8  
635 - val\_loss: 55.0190 - val\_accuracy: 0.8639  
Epoch 5/60  
57/57 [=====] - 1s 14ms/step - loss: 32.2837 - accuracy: 0.8  
650 - val\_loss: 41.2236 - val\_accuracy: 0.8661  
Epoch 6/60  
57/57 [=====] - 1s 14ms/step - loss: 31.7482 - accuracy: 0.8  
664 - val\_loss: 35.3415 - val\_accuracy: 0.8656  
Epoch 7/60  
57/57 [=====] - 1s 14ms/step - loss: 31.3158 - accuracy: 0.8  
668 - val\_loss: 33.2966 - val\_accuracy: 0.8661  
Epoch 8/60  
57/57 [=====] - 1s 14ms/step - loss: 30.9005 - accuracy: 0.8  
672 - val\_loss: 32.9001 - val\_accuracy: 0.8656  
Epoch 9/60  
57/57 [=====] - 1s 13ms/step - loss: 30.5254 - accuracy: 0.8  
683 - val\_loss: 33.3832 - val\_accuracy: 0.8678  
Epoch 10/60  
57/57 [=====] - 1s 13ms/step - loss: 30.3263 - accuracy: 0.8  
687 - val\_loss: 33.0628 - val\_accuracy: 0.8667  
Epoch 11/60  
57/57 [=====] - 1s 14ms/step - loss: 29.9822 - accuracy: 0.8  
697 - val\_loss: 32.6230 - val\_accuracy: 0.8667  
Epoch 12/60  
57/57 [=====] - 1s 13ms/step - loss: 29.8359 - accuracy: 0.8  
704 - val\_loss: 32.6833 - val\_accuracy: 0.8667  
Epoch 13/60  
57/57 [=====] - 1s 14ms/step - loss: 29.6976 - accuracy: 0.8  
701 - val\_loss: 32.1719 - val\_accuracy: 0.8683  
Epoch 14/60  
57/57 [=====] - 1s 13ms/step - loss: 29.4056 - accuracy: 0.8  
703 - val\_loss: 32.3488 - val\_accuracy: 0.8678  
Epoch 15/60  
57/57 [=====] - 1s 14ms/step - loss: 29.2821 - accuracy: 0.8  
718 - val\_loss: 32.0218 - val\_accuracy: 0.8694  
Epoch 16/60  
57/57 [=====] - 1s 14ms/step - loss: 29.1975 - accuracy: 0.8  
711 - val\_loss: 31.6924 - val\_accuracy: 0.8689  
Epoch 17/60  
57/57 [=====] - 1s 13ms/step - loss: 29.1457 - accuracy: 0.8  
714 - val\_loss: 31.7353 - val\_accuracy: 0.8683  
Epoch 18/60  
57/57 [=====] - 1s 14ms/step - loss: 28.8578 - accuracy: 0.8  
721 - val\_loss: 31.6458 - val\_accuracy: 0.8700  
Epoch 19/60  
57/57 [=====] - 1s 13ms/step - loss: 28.6731 - accuracy: 0.8  
729 - val\_loss: 31.8186 - val\_accuracy: 0.8689

Epoch 20/60  
57/57 [=====] - 1s 14ms/step - loss: 28.5593 - accuracy: 0.8  
731 - val\_loss: 31.6456 - val\_accuracy: 0.8700  
Epoch 21/60  
57/57 [=====] - 1s 13ms/step - loss: 28.5263 - accuracy: 0.8  
740 - val\_loss: 31.9543 - val\_accuracy: 0.8683  
Epoch 22/60  
57/57 [=====] - 1s 13ms/step - loss: 28.4680 - accuracy: 0.8  
732 - val\_loss: 31.6660 - val\_accuracy: 0.8678  
Epoch 23/60  
57/57 [=====] - 1s 13ms/step - loss: 28.1531 - accuracy: 0.8  
754 - val\_loss: 31.8029 - val\_accuracy: 0.8706  
Epoch 24/60  
57/57 [=====] - 1s 14ms/step - loss: 28.3030 - accuracy: 0.8  
740 - val\_loss: 31.6323 - val\_accuracy: 0.8689  
Epoch 25/60  
57/57 [=====] - 1s 14ms/step - loss: 28.2648 - accuracy: 0.8  
765 - val\_loss: 31.3427 - val\_accuracy: 0.8694  
Epoch 26/60  
57/57 [=====] - 1s 13ms/step - loss: 28.0794 - accuracy: 0.8  
754 - val\_loss: 31.4929 - val\_accuracy: 0.8711  
Epoch 27/60  
57/57 [=====] - 1s 13ms/step - loss: 28.0483 - accuracy: 0.8  
763 - val\_loss: 31.4669 - val\_accuracy: 0.8717  
Epoch 28/60  
57/57 [=====] - 1s 13ms/step - loss: 27.9692 - accuracy: 0.8  
765 - val\_loss: 31.4872 - val\_accuracy: 0.8711  
Epoch 29/60  
57/57 [=====] - 1s 13ms/step - loss: 27.9077 - accuracy: 0.8  
761 - val\_loss: 31.3515 - val\_accuracy: 0.8728  
Epoch 30/60  
57/57 [=====] - 1s 13ms/step - loss: 27.6583 - accuracy: 0.8  
776 - val\_loss: 31.5206 - val\_accuracy: 0.8706  
Epoch 31/60  
57/57 [=====] - 1s 14ms/step - loss: 26.9778 - accuracy: 0.8  
825 - val\_loss: 29.8108 - val\_accuracy: 0.8839  
Epoch 32/60  
57/57 [=====] - 1s 13ms/step - loss: 26.7351 - accuracy: 0.8  
863 - val\_loss: 29.9785 - val\_accuracy: 0.8822  
Epoch 33/60  
57/57 [=====] - 1s 14ms/step - loss: 26.4671 - accuracy: 0.8  
868 - val\_loss: 29.3014 - val\_accuracy: 0.8961  
Epoch 34/60  
57/57 [=====] - 1s 14ms/step - loss: 18.5209 - accuracy: 0.9  
325 - val\_loss: 18.5981 - val\_accuracy: 0.9278  
Epoch 35/60  
57/57 [=====] - 1s 14ms/step - loss: 16.0708 - accuracy: 0.9  
381 - val\_loss: 18.2304 - val\_accuracy: 0.9294  
Epoch 36/60  
57/57 [=====] - 1s 13ms/step - loss: 15.6734 - accuracy: 0.9  
396 - val\_loss: 18.2561 - val\_accuracy: 0.9272  
Epoch 37/60  
57/57 [=====] - 1s 13ms/step - loss: 15.7077 - accuracy: 0.9  
397 - val\_loss: 18.4675 - val\_accuracy: 0.9283  
Epoch 38/60  
57/57 [=====] - 1s 13ms/step - loss: 15.5144 - accuracy: 0.9  
400 - val\_loss: 18.3966 - val\_accuracy: 0.9267

Epoch 39/60  
57/57 [=====] - 1s 13ms/step - loss: 15.4156 - accuracy: 0.9  
401 - val\_loss: 18.2500 - val\_accuracy: 0.9267  
Epoch 40/60  
57/57 [=====] - 1s 14ms/step - loss: 15.3753 - accuracy: 0.9  
404 - val\_loss: 18.1604 - val\_accuracy: 0.9278  
Epoch 41/60  
57/57 [=====] - 1s 13ms/step - loss: 15.2652 - accuracy: 0.9  
413 - val\_loss: 18.4661 - val\_accuracy: 0.9244  
Epoch 42/60  
57/57 [=====] - 1s 13ms/step - loss: 15.2784 - accuracy: 0.9  
401 - val\_loss: 18.4473 - val\_accuracy: 0.9244  
Epoch 43/60  
57/57 [=====] - 1s 13ms/step - loss: 15.1652 - accuracy: 0.9  
410 - val\_loss: 18.3328 - val\_accuracy: 0.9261  
Epoch 44/60  
57/57 [=====] - 1s 13ms/step - loss: 15.1593 - accuracy: 0.9  
413 - val\_loss: 18.9192 - val\_accuracy: 0.9261  
Epoch 45/60  
57/57 [=====] - 1s 13ms/step - loss: 15.0368 - accuracy: 0.9  
424 - val\_loss: 18.3384 - val\_accuracy: 0.9272  
Epoch 46/60  
57/57 [=====] - 1s 13ms/step - loss: 15.0386 - accuracy: 0.9  
413 - val\_loss: 18.2480 - val\_accuracy: 0.9278  
Epoch 47/60  
57/57 [=====] - 1s 13ms/step - loss: 14.8684 - accuracy: 0.9  
417 - val\_loss: 18.3713 - val\_accuracy: 0.9261  
Epoch 48/60  
57/57 [=====] - 1s 14ms/step - loss: 14.9243 - accuracy: 0.9  
414 - val\_loss: 17.9589 - val\_accuracy: 0.9300  
Epoch 49/60  
57/57 [=====] - 1s 13ms/step - loss: 14.8779 - accuracy: 0.9  
411 - val\_loss: 18.3441 - val\_accuracy: 0.9261  
Epoch 50/60  
57/57 [=====] - 1s 14ms/step - loss: 14.9845 - accuracy: 0.9  
419 - val\_loss: 17.9183 - val\_accuracy: 0.9294  
Epoch 51/60  
57/57 [=====] - 1s 13ms/step - loss: 14.6604 - accuracy: 0.9  
422 - val\_loss: 18.1177 - val\_accuracy: 0.9272  
Epoch 52/60  
57/57 [=====] - 1s 13ms/step - loss: 14.7114 - accuracy: 0.9  
421 - val\_loss: 18.1657 - val\_accuracy: 0.9272  
Epoch 53/60  
57/57 [=====] - 1s 13ms/step - loss: 14.6706 - accuracy: 0.9  
428 - val\_loss: 18.3589 - val\_accuracy: 0.9272  
Epoch 54/60  
57/57 [=====] - 1s 13ms/step - loss: 14.5373 - accuracy: 0.9  
431 - val\_loss: 18.0170 - val\_accuracy: 0.9294  
Epoch 55/60  
57/57 [=====] - 1s 13ms/step - loss: 14.5484 - accuracy: 0.9  
429 - val\_loss: 17.9400 - val\_accuracy: 0.9272  
Epoch 56/60  
57/57 [=====] - 1s 13ms/step - loss: 14.7120 - accuracy: 0.9  
426 - val\_loss: 18.0593 - val\_accuracy: 0.9283  
Epoch 57/60  
57/57 [=====] - 1s 13ms/step - loss: 14.6741 - accuracy: 0.9  
428 - val\_loss: 17.9750 - val\_accuracy: 0.9300

```
Epoch 58/60
57/57 [=====] - 1s 13ms/step - loss: 14.5798 - accuracy: 0.9
425 - val_loss: 18.2836 - val_accuracy: 0.9289
Epoch 59/60
57/57 [=====] - 1s 13ms/step - loss: 14.5435 - accuracy: 0.9
432 - val_loss: 18.0208 - val_accuracy: 0.9278
Epoch 60/60
57/57 [=====] - 1s 14ms/step - loss: 14.4863 - accuracy: 0.9
436 - val_loss: 18.0674 - val_accuracy: 0.9317
Out[ ]: <keras.callbacks.History at 0x7f08fae76ad0>
```

```
In [ ]: cleaner.evaluate(V_test)
cleaner.trainable = False

8/8 [=====] - 0s 18ms/step - loss: 18.5810 - accuracy: 0.923
0

In [ ]: c_train_full = tf.nn.softmax(
    cleaner.predict([
        images_normalized,
        y_noisy
    ])
)

1563/1563 [=====] - 10s 6ms/step
```

```
In [ ]: cnn_1.trainable = False
# image_classifier = tf.keras.Sequential([
#     cnn_1,
#     tf.keras.layers.Dense(units = 512),
#     tf.keras.layers.Dense(units = 10, activation = "sigmoid")
# ])
image_classifier = ImageClassifier(cnn_1)

image_classifier.compile(
    loss = tf.keras.losses.CategoricalCrossentropy(),
    optimizer = tf.keras.optimizers.Adam(0.001),
    metrics = ['accuracy']
)
```

```
In [ ]: def ImageClassifier(cnn: tf.keras.Model) -> tf.keras.Model:
    """
    Build a new model that takes images as input and outputs class probabilities.

    Parameters:
    -----
    `cnn: tf.keras.Model`
        The base CNN model to use.

    `return`
        The new model.
    """
    return tf.keras.Sequential([
        cnn,
        tf.keras.layers.Dense(units = 64),
```

```
    tf.keras.layers.Dense(units = 10, activation = "sigmoid")
])
```

```
In [ ]: image_classifier.fit(
    images_normalized,
    c_train_full,
    epochs = 30,
    batch_size = BATCH_SIZE,
)
```

Epoch 1/30  
391/391 [=====] - 6s 11ms/step - loss: 2.2450 - accuracy: 0.  
9397  
Epoch 2/30  
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.  
9390  
Epoch 3/30  
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.  
9390  
Epoch 4/30  
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.  
9394  
Epoch 5/30  
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.  
9393  
Epoch 6/30  
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.  
9390  
Epoch 7/30  
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.  
9391  
Epoch 8/30  
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.  
9391  
Epoch 9/30  
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.  
9389  
Epoch 10/30  
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.  
9391  
Epoch 11/30  
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.  
9392  
Epoch 12/30  
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.  
9392  
Epoch 13/30  
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.  
9392  
Epoch 14/30  
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.  
9392  
Epoch 15/30  
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.  
9391  
Epoch 16/30  
391/391 [=====] - 4s 11ms/step - loss: 2.2386 - accuracy: 0.  
9390  
Epoch 17/30  
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.  
9389  
Epoch 18/30  
391/391 [=====] - 4s 11ms/step - loss: 2.2386 - accuracy: 0.  
9392  
Epoch 19/30  
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.  
9393

```
Epoch 20/30
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.
9392
Epoch 21/30
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.
9391
Epoch 22/30
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.
9392
Epoch 23/30
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.
9393
Epoch 24/30
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.
9393
Epoch 25/30
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.
9391
Epoch 26/30
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.
9392
Epoch 27/30
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.
9390
Epoch 28/30
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.
9391
Epoch 29/30
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.
9394
Epoch 30/30
391/391 [=====] - 4s 10ms/step - loss: 2.2386 - accuracy: 0.
9394
```

```
Out[ ]: <keras.callbacks.History at 0x7f07efb1efd0>
```

```
In [ ]: image_classifier.fit(
    x_clean_train,
    y_clean_train,
    epochs = 30,
    batch_size = BATCH_SIZE,
    validation_data = (x_clean_val, y_clean_val),
    callbacks = [es]
)
```

```
Epoch 1/30
57/57 [=====] - 1s 25ms/step - loss: 0.5831 - accuracy: 0.92
44 - val_loss: 0.3229 - val_accuracy: 0.9222
Epoch 2/30
57/57 [=====] - 1s 13ms/step - loss: 0.3031 - accuracy: 0.92
76 - val_loss: 0.3202 - val_accuracy: 0.9217
Epoch 3/30
57/57 [=====] - 1s 13ms/step - loss: 0.2979 - accuracy: 0.92
81 - val_loss: 0.3195 - val_accuracy: 0.9228
Epoch 4/30
57/57 [=====] - 1s 13ms/step - loss: 0.2960 - accuracy: 0.92
82 - val_loss: 0.3192 - val_accuracy: 0.9211
Epoch 5/30
57/57 [=====] - 1s 13ms/step - loss: 0.2949 - accuracy: 0.92
90 - val_loss: 0.3185 - val_accuracy: 0.9222
Epoch 6/30
57/57 [=====] - 1s 12ms/step - loss: 0.2939 - accuracy: 0.92
87 - val_loss: 0.3192 - val_accuracy: 0.9222
Epoch 7/30
57/57 [=====] - 1s 13ms/step - loss: 0.2931 - accuracy: 0.92
85 - val_loss: 0.3201 - val_accuracy: 0.9222
Epoch 8/30
57/57 [=====] - 1s 12ms/step - loss: 0.2925 - accuracy: 0.92
93 - val_loss: 0.3197 - val_accuracy: 0.9222
Epoch 9/30
57/57 [=====] - 1s 12ms/step - loss: 0.2921 - accuracy: 0.92
97 - val_loss: 0.3205 - val_accuracy: 0.9222
Epoch 10/30
57/57 [=====] - 1s 13ms/step - loss: 0.2915 - accuracy: 0.92
97 - val_loss: 0.3204 - val_accuracy: 0.9217
Epoch 11/30
57/57 [=====] - 1s 13ms/step - loss: 0.2914 - accuracy: 0.92
90 - val_loss: 0.3205 - val_accuracy: 0.9211
Epoch 12/30
57/57 [=====] - 1s 13ms/step - loss: 0.2911 - accuracy: 0.93
06 - val_loss: 0.3202 - val_accuracy: 0.9217
Epoch 13/30
57/57 [=====] - 1s 13ms/step - loss: 0.2912 - accuracy: 0.92
96 - val_loss: 0.3209 - val_accuracy: 0.9222
Epoch 14/30
57/57 [=====] - 1s 12ms/step - loss: 0.2909 - accuracy: 0.92
90 - val_loss: 0.3204 - val_accuracy: 0.9228
Epoch 15/30
57/57 [=====] - 1s 13ms/step - loss: 0.2907 - accuracy: 0.93
03 - val_loss: 0.3206 - val_accuracy: 0.9228
Out[ ]: <keras.callbacks.History at 0x7f07ef8b9650>
```

```
In [ ]: image_classifier.evaluate(images_normalized, c_train_full)

1563/1563 [=====] - 12s 8ms/step - loss: 4.3865 - accuracy: 0.9573
Out[ ]: [4.386488914489746, 0.9573400020599365]
```

```
In [ ]: image_classifier.save_weights("data/image_classifier/image_classifier")

In [ ]: # [ADD WEAKLY SUPERVISED LEARNING FEATURE TO MODEL I]
```

```
# write your code here...

def model_II(image):
    ...
    This function should takes in the image of dimension 32*32*3 as input and returns
    ...
    # write your code here...
    image_classifier.load_weights("data/image_classifier/image_classifier")

    pred = image_classifier(image.numpy().reshape(1, 32, 32, 3))
    label = int(tf.argmax(pred, 1))

    return label
```