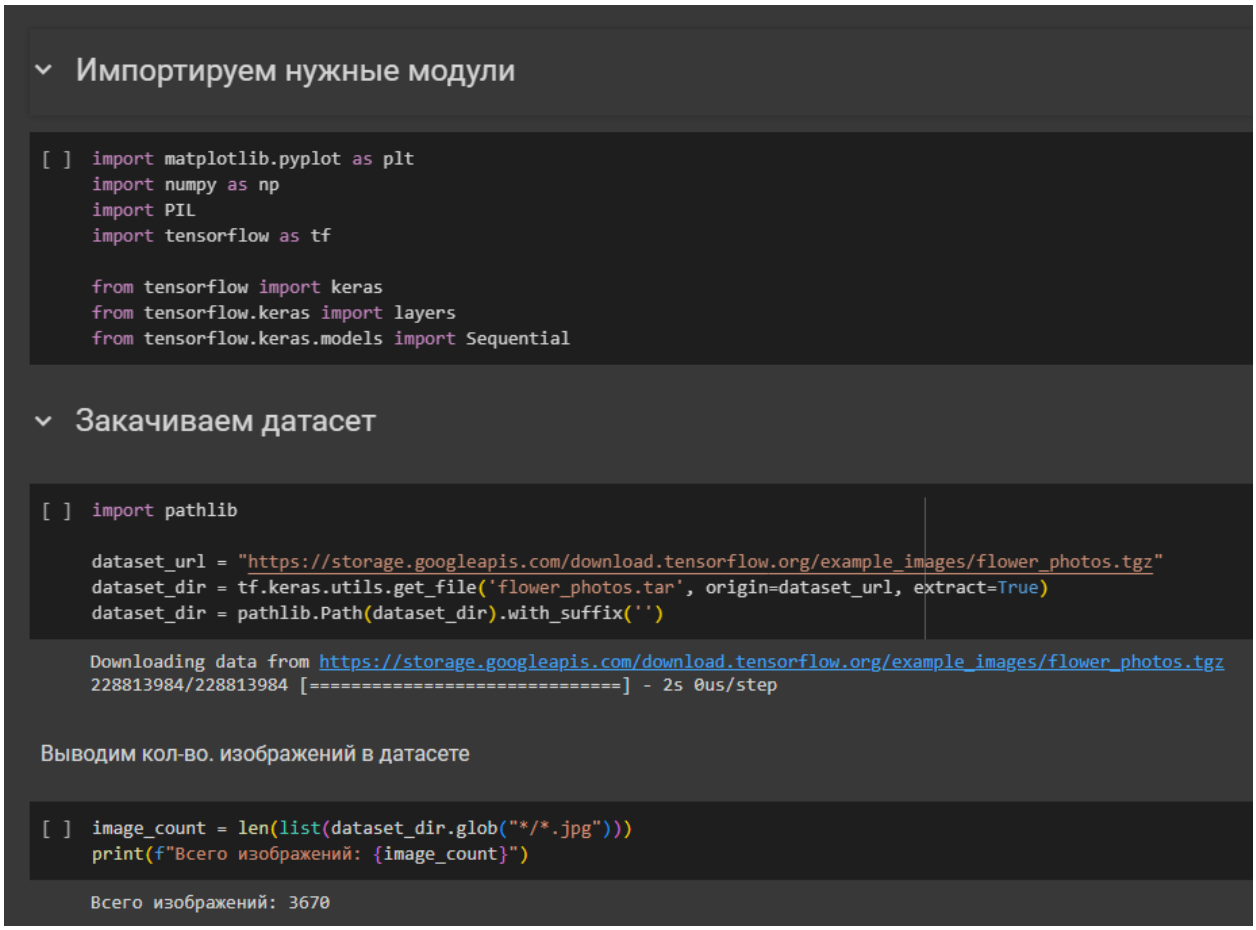
ТЗ

Основная часть работы заключается в рефакторинге уже готового кода под конкретную задачу.

У меня есть уже готовый скрипт написанный в GoogleColab для обучения нейросети распознавать изображения из конкретного датасета: https://colab.research.google.com/drive/120nbtB3ywlPM0b-4QlcVVW_UmJdZacY4?usp=sharing#scrollTo=DT2-NCLnI1wr

Следующие рисунки будут для наглядности того с чем надо работать.



Создаем датасеты и кэшируем их

```
batch_size = 32
img_width = 180
img_height = 180

train_ds = tf.keras.utils.image_dataset_from_directory(
  dataset_dir,
  validation_split=0.2,
  subset="training",
  seed=123,
  image_size=(img_height, img_width),
  batch_size=batch_size)

val_ds = tf.keras.utils.image_dataset_from_directory(
  dataset_dir,
  validation_split=0.2,
  subset="validation",
  seed=123,
  image_size=(img_height, img_width),
  batch_size=batch_size)

class_names = train_ds.class_names
print(f"Class names: {class_names}")

# cache
AUTOTUNE = tf.data.AUTOTUNE
train_ds = train_ds.cache().shuffle(1000).prefetch(buffer_size=AUTOTUNE)
val_ds = val_ds.cache().prefetch(buffer_size=AUTOTUNE)
```

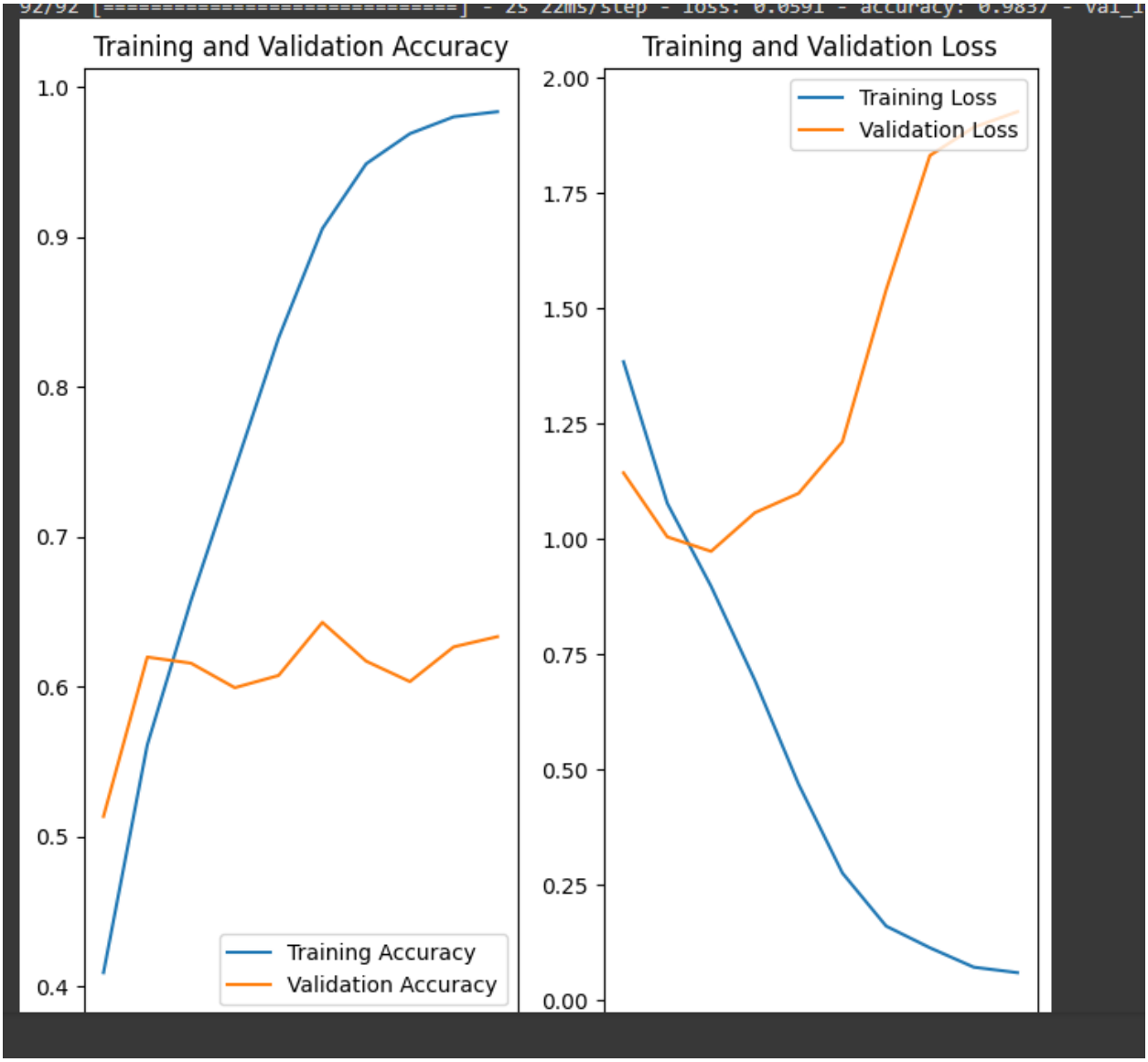
```
Found 3670 files belonging to 5 classes.
Using 2936 files for training.
Found 3670 files belonging to 5 classes.
Using 734 files for validation.
Class names: ['daisy', 'dandelion', 'roses', 'sunflowers', 'tulips']
```

Создаем модель, компилируем её и выводим summary

```
# create model
num_classes = len(class_names)
model = Sequential([
  # т.к. у нас версия TF 2.6 локально
  layers.experimental.preprocessing.Rescaling(1./255, input_shape=(img_height, img_width,

  # дальше везде одинаково
  layers.Conv2D(16, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),

  layers.Conv2D(32, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),

  layers.Conv2D(64, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),

  layers.Flatten(),
  layers.Dense(128, activation='relu'),
  layers.Dense(num_classes)
])

# compile the model
model.compile(
  optimizer='adam',
  loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
  metrics=['accuracy'])

# print model summary
model.summary()
```

Обучаем нейросеть и выводим графики точности

```
[ ] # train the model
epochs = 10 # количество эпох тренировки
history = model.fit(
  train_ds,
  validation_data=val_ds,
  epochs=epochs)

# visualize training and validation results
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

loss = history.history['loss']
val_loss = history.history['val_loss']

epochs_range = range(epochs)

plt.figure(figsize=(8, 8))
plt.subplot(1, 2, 1)
plt.plot(epochs_range, acc, label='Training Accuracy')
plt.plot(epochs_range, val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

plt.subplot(1, 2, 2)
plt.plot(epochs_range, loss, label='Training Loss')
plt.plot(epochs_range, val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()
```



Задание:

1. Переписать данный код, так чтобы вместо изображений, нейросеть смогла распознавать аудиофайлы из уже готового датасета.
2. Нейросеть должна иметь высокую точность более 90%
3. Должны быть представлены графики валидационного датасета и тренировочного как показано в документе GoogleColab

<u>У меня есть уже готовый датасет</u> со звуками аварий маших на дорогах, а так же датасеты, со звуками, которые происходят на дороге, такие как: шум дороги, сирена экстренных служб а так же огромный датасет Urbansound8k где несколько гб различных звуков города.

Так же есть GoogleColab, в котором уже начал работу с обучением данной нейросети но возникли сложности в процессе обработки входных данных.

Ссылка на данный гугл колаб ниже:

https://colab.research.google.com/drive/1yisDAmQqzcKP9XPsxRGPeZrje69Bclih#scrollTo=5NVA_ipYOQPV