

Здравствуйте!

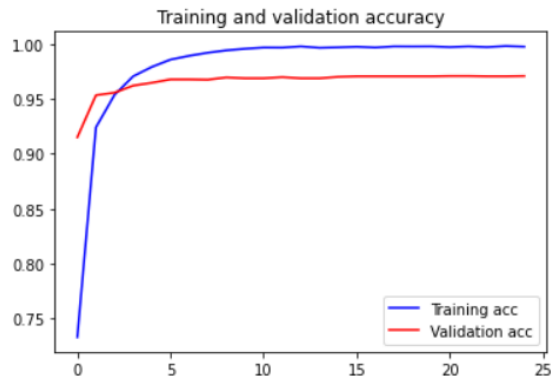
Просьба открывать доступ к Colab, когда даете ссылку, если бы вы не продублировали ссылкой на github, я бы не смог проверить работу.

Далее комментарии по коду работы:

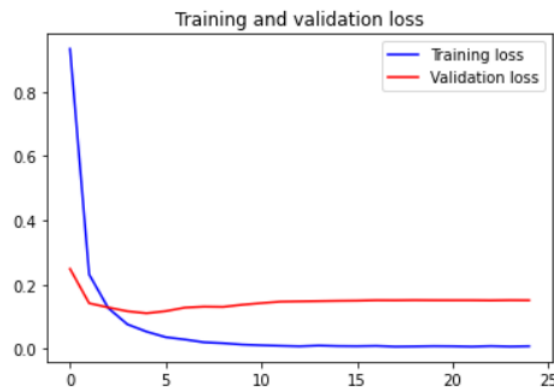
---

➤ `!pip install -U git+https://github.com/albumentations-team/albumentations`

Вы можете добавлять флаг `-q`, чтобы в вывод ячейки не печаталось много текста.



Вы можете выбрать логарифмический масштаб для оси x: `plt.gca().set_xscale('log')`. Тогда даже небольшой рост точности на валидации будет замечен, и поэтому будет понятно, имеет ли смысл продолжать обучение.



Здесь можно сделать логарифмическими обе оси:

```
plt.loglog(epochs, loss, 'b', label='Training loss')
plt.loglog(epochs, val_loss, 'r', label='Validation loss')
```

➤ `# Поменяем размер картинки и уменьшим batch_size. Возьмем другую модель из keras.`

Если при каждом изменении вы дублируете код создания генераторов, то ноутбук получается очень большого размера, и в нем сложно разобраться. Вы можете оборачивать создание генераторов в функцию с параметрами и вызывать ее несколько раз.

```
vertical_flip=True,
```

А вы уверены, что это нужно? Должна ли модель распознавать перевернутые автомобили?

```
optimizer=Adam(ExponentialDecay(1e-3, 100, 0.9)),
```

Такие параметры могут привести к следующей проблеме: у вас эпоха занимает 400 шагов, то есть в течение эпохи LR умножается на 0.66. После всего 10 эпох LR станет настолько низким, что модель перестанет обучаться, даже если у нее был потенциал к дальнейшему обучению.

```
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(len(acc))

plt.plot(epochs, acc, 'b', label='Training acc')
plt.plot(epochs, val_acc, 'r', label='Validation acc')
plt.title('Training and validation accuracy')
plt.legend()

plt.figure()

plt.plot(epochs, loss, 'b', label='Training loss')
plt.plot(epochs, val_loss, 'r', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()

plt.show()
```

Здесь, опять, можно обернуть этот код в функцию, если вам требуется многократная отрисовка графиков.

```
A.Cutout(p=0.5, num_holes=1, max_h_size=50, max_w_size=50)
```

Cutout – эффективная аугментация, хорошо что вы ее применяете.

Было бы хорошо, если в конце было сравнение точности всех опробованных моделей.

```
for _ in range(5):
    predictions.append(model_5.predict(sub_generator, verbose=1))
    sub_generator.reset()
```

Если делать ТТА небольшое число шагов, то точность, наоборот, может стать хуже, поэтому я бы посоветовал большее число шагов, учитывая, что это не слишком затратно по времени.

---

Отзыв подготовил ментор проекта Олег Зяблов. Если есть вопросы, можете задать их в канале #0-project\_7-ford\_vs\_ferrari, постараюсь помочь разобраться. Успехов в дальнейшем обучении! Обязательно подключайтесь на итоговый созвон-вебинар по проекту **5 февраля**. Анонс вебинара появится позже в канале #0-project\_7-ford\_vs\_ferrari.