

```
print('Распаковываем картинки')
# Will unzip the files so that you can see them..
for data_zip in ['train.zip', 'test.zip']:
    with zipfile.ZipFile("../input/"+data_zip,"r") as z:
        z.extractall(PATH)
```

Можно сделать то же самое командой ОС:

```
!unzip -q ../input/train.zip -d {PATH}
```

```
test_generator = train_datagen.flow_from_directory(
```

У вас валидация делается с аугментациями. Так не должно быть. Представьте, что одну модель вы обучили на слабых аугментациях, другую на сильных. Если валидация делается с аугментациями, то модель с сильными аугментациями покажет более низкую точность на валидации. Если же делать валидацию нормально, без аугментаций, то все может быть наоборот. Поскольку валидация влияет на выбор моделей и гиперпараметров, важно делать ее без аугментаций, особенно если вы сравниваете разные модели и значения гиперпараметров.

```
from efficientnet.keras import EfficientNetB7
```

```
train_datagen = ImageDataGenerator(
    rescale=1. / 255,
```

В модуле keras.applications уже есть сеть EfficientNetB7, так что устанавливать отдельную библиотеку не обязательно. Если все же пользоваться pip-пакетом efficientnet, то в нем нормализация делается [таким образом](#): efficientnet.keras.preprocess_input

Это не то же самое, что деление на 255. Правильная нормализация (то есть такая же, какая была при предобучении) может немного улучшить точность, хотя это не принципиальный момент.

```
model.add(Dense(256, activation='relu'))
model.add(BatchNormalization())
```

Непонятно из каких соображений добавляются эти слои. Больше слоев – не всегда значит лучше, иначе можно было бы добавить и 100 слоев. Как правило при файн-тюнинге сверточных сетей не добавляются дополнительные слои, кроме выходного.

```
ModelCheckpoint('best_model.hdf5' , monitor = ['val_accuracy'] , verbose = 1
, mode = 'max')
```

Здесь нужно добавить параметр save_best_only=True

```
model.fit_generator(
```

Это то же самое, что .fit(). Метод fit_generator раньше применялся, но сейчас устарел, и можно использовать .fit(). Аналогично, метод evaluate_generator эквивалентен evaluate.

96% уже хорошая точность, но можно было ее легко увеличить еще сильнее, сделав test time augmentations, это дало бы всего несколькими строчками кода прирост к точности.

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 18 декабря. Анонс вебинара появится позже в канале #0-project_7-ford_vs_ferrari.