

Здравствуйте! Далее комментарии по коду работы:

```
!unzip /kaggle/input/train.zip -d /kaggle/temp
```

Вы можете добавить флаг -q после команды unzip, чтобы в консоль не печатались имена файлов (так и работать будет в несколько раз быстрее).

```
!pip show -q
```

Эта команда работает только с именем пакета, например !pip show tensorflow

```
PATH = "../working/car/" # рабочая директория
```

А зачем нужна эта переменная? Она нигде далее не используется, да и папки такой не создается. Фотографии у вас хранятся в /kaggle/temp. Кажется, это лишний код.

```
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(TEMP)

print(os.listdir(TEMP))
```

Но вы уже распаковывали картинки в начале ноутбука командой !unzip, кажется это тоже лишний код.

```
augmentations.OneOf([
    augmentations.CenterCrop(height=220, width=200), # Обрежьте центральную
    часть входа.
    augmentations.CenterCrop(height=200, width=220),
], p=0.5),
```

Это один из вариантов аугментаций, но можно как вариант использовать RandomCrop.

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

Вы используете сеть xception, для нее нужна особая нормализация. Но и деление на 255 тоже подойдет, правда точность на валидации в итоге может оказаться немного ниже, чем могла бы быть.

```
test_generator = train_datagen.flow_from_directory(
    TEMP+'train/',
```

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.

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

Здесь нужно добавить `save_best_only=True`.

```
base_model1 = efn.EfficientNetB5(weights='imagenet', include_top=False,  
input_shape = input_shape)
```

Эту модель, как вариант, можно было взять не из отдельного `pip`-пакета, а из модуля `keras.applications`.

Отзыв подготовил ментор проекта Олег Седухин. 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 **19 февраля**. Анонс вебинара появится позже в канале `#0-project_7-ford_vs_ferrari`.