

## Качество и понятность кода

Оценка: 3 балла из 3.

## Метрика качества

0.96794 в лидерборде: +1 доп. балл за точность, согласно критериям оценивания.

## Работа с обучающими данными

Есть однако большая проблема: валидация (`test_generator`) делается также с аугментациями (на основе `train_datagen`). 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 – и выяснится, что модель, обученная на сильных аугментациях, обучилась лучше. Поскольку валидация влияет на выбор моделей и гиперпараметров, важно делать ее без аугментаций.

Оценка: 1 балл из 3.

## Работа с архитектурой модели

Вы сравнивали 3 модели: Xception, ResNet50V2 и EfficientNetB5. Но сравнение, как мне кажется, проведено не совсем корректно. В процессе сравнения у Xception вы указывали параметр `trainable = False`. В коде, где вы работали с другими моделями (ResNet50V2 и EfficientNetB5) я такого параметра не увидел. Конечно нельзя сравнивать 2 модели, одна из которых заморожена, а две другие нет.

Но если предположить, что модели ResNet50V2 и EfficientNetB5 вы также замораживали, то все равно сравнение некорректно. Замороженная модель – это фактически модель, обученная на совсем другой задаче (imagenet). Ее на файн-тюнить под задачу распознавания автомобилей, то есть размораживать и обучать, и только тогда проявится ее потенциал. А вы сравниваете замороженные модели друг с другом. Такое сравнение очень ненадежно.

Может быть конечно я неверно понял, и работая со всеми тремя моделями вы сначала их замораживаете, а потом постепенно размораживаете. Но тогда точность получается нереалистично низкой: всего 82%, 73% и 79% на разрешении 320x320, или у вас использовалось более низкое разрешение, но опять-таки вы об этом не написали. Вам надо было подробнее написать, по какому алгоритму вы сравнивали модели, чтобы можно было понять, корректно такое сравнение или нет. Но я готов поверить, что все-таки сравнение проводилось корректно, а я вас неверно понял.

И наконец, вы добавляете скрытый полносвязный слой, но не обосновываете такой выбор. Почему один слой, почему не 0, не 2 или 3? Если посмотреть современные работы по сверточным сетям, то там практически никогда скрытый полносвязный слой не используется.

И еще: не совсем понятно почему в процессе сравнения с другими моделями сеть Xception показывает точность 82% (хотя указан параметр `trainable = False`), а в следующем разделе в процессе обучения всего 68% после 5 эпох.

Оценка: 2 балла из 3.

## Работа с процессом обучения

Вы делаете файн-тюнинг сети, постепенно размораживая слои. Но опять-таки я должен признать, что такой решение вы никак не обосновываете. Почему бы не разморозить изначально всю сеть? Вдруг точность окажется выше? Согласно [критериям оценивания](#), нужно было выбрать оптимальный алгоритм обучения, основываясь либо на нескольких экспериментах (их можно проводить с низким размером картинки для ускорения), либо на научных статьях или статьях их интернета. Вы же ни на чем не основываетесь в своем выборе.

Оценка: 1 балл из 3.

## Работа с процессом инференса

У вас делается ТТА, хотя я бы рекомендовал делать его немного другим способом, не просто по 5 предсказаний на каждую картинку. Сделав 5 предсказаний сравните их – если они все совпали,

значит данный пример легкий. А если нет – значит надо продолжить ТТА на данном примере. И кроме того, судя по коду, ТТА делается у вас неправильно: `test_sub_generator` возвращает изображения без аугментаций, поэтому все 5 предсказаний будут идентичны.  
Оценка: 0 баллов из 3.

**Создание работающего прототипа**

Этого пункта у вас не нашел.

Сумма:  $3+1+1+2+1 = 8$  баллов