

Здравствуйте! Далее комментарии по коду работы:

```
augmentations.Rotate(limit=30, interpolation=1, border_mode=4, value=None,
mask_value=None, always_apply=False, p=0.5),
```

Возможно поворот на 30 градусов слишком большой? Практически все изображения в тестовом датасете не повернуты на такой угол, поэтому нет необходимости распознавать так сильно повернутые изображения. Можно было бы уменьшить угол поворота при аугментации.

```
augmentations.OneOf([
    augmentations.CenterCrop(height=250, width=200),
    augmentations.CenterCrop(height=200, width=250),
], p=0.5),
```

Если мы делаем аугментации, то можно вырезать не центральную, а случайную часть изображения (RandomCrop). Хотя иногда бывает по несколько автомобилей на фото, и нужное авто как правило находится в центре, поэтому может быть так было бы наоборот хуже.

```
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.

```
base_model = efn.EfficientNetB6(
    weights='imagenet', # Подгружаем веса imagenet
    include_top=False, # Выходной слой (голову) будем менять т.к. у нас
    другие классы
    input_shape=input_shape)
```

Модель efficientNet есть в keras.applications, но если вы решили брать эту модель из отдельной библиотеки, то надо посмотреть в документации какая нормализация для нее нужна. Это не просто деление на 255, см. пример

https://github.com/qubvel/efficientnet/blob/master/examples/inference_example.ipynb .

➤ Step 1 - обучение "головы"

По идее, обучение головы – очень простая задача для оптимизатора, так как вся сеть заморожена и обучается только последний слой. Наверное нет особого смысла на этом шаге использовать циклический LR. Но с другой стороны его использование, конечно, не повредит.

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

Здесь нужно добавить параметр save_best_only=True

```
history = model.fit_generator(
predictions = model.predict_generator(
scores = model.evaluate_generator(test_generator, verbose=1)
```

Можно использовать просто model.fit, model.predict и model.evaluate, это то же самое.

Кстати, по моим наблюдениям сеть обучается лучше, если ее всю разморозить с самого начала. Так вдобавок еще и проще, чем с послойной разморозкой.

У вас сразу реализуется сложный алгоритм обучения (с послойной разморозкой и циклическим LR). Но я бы предложил сначала пробовать простой, а затем сравнивать с ним сложный. Может быть окажется, что простой алгоритм (обучение всей сети без заморозки и циклического LR) эффективнее.

Почему-то пропали выводы ячеек кода, поэтому не могу узнать какая у вас точность получилась на промежуточных шагах.

```
tta_steps = 10 # берем среднее из 10 предсказаний
predictions = []

for i in range(tta_steps):
    preds = model.predict_generator(test_sub_generator,
    steps=len(test_sub_generator), verbose=1)
    predictions.append(preds)

pred = np.mean(predictions, axis=0)
```

Можно как вариант делать больше шагов ТТА на сложных изображениях (в которых ответ сети различается на разных шагах ТТА).

У вас очень хорошая, качественная работа, при этом точность в лидерборде 94% выглядит странно. Мне кажется, у вас должно было получиться около 97% или выше. Возможно закралась какая-то ошибка в коде сабмита.

Возможно проблема в том, что при ТТА вы делаете слишком сильные аугментации (поворот на 30 градусов), тогда нужно намного больше шагов.

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 **26 февраля**. Анонс вебинара появится позже в канале #0-project_7-ford_vs_ferrari.