Здравствуйте! Далее комментарии по коду работы:

```
!pip install efficientnet
```

Библиотека efficientnet была актуальна года два назад, но с тех пор EfficientNet появился в модуле keras.applications, так что устанавливать доп. библиотеку не обязательно.

Важно, что в библиотеке efficientnet сети нужна нормализация данных, а в keras.applications. EfficientNet не нужна.

Почему-то вы устанавливаетсе отдельную библитеку efficientnet, но никак ее не используете.

```
def draw_graph(history):
      …
      return plt
```

А зачем return plt?

```
augumentation = albumentations.Compose([
    albumentations.ShiftScaleRotate(shift_limit=0.0625,
                        scale_limit=0.01,
                        interpolation=1,
                        border_mode=4,
                        rotate_limit=20,
                        p=.75),
    albumentations.HorizontalFlip(p=0.5),
    albumentations.Rotate(limit=10, interpolation=1, border_mode=4, value=
None, mask_value=None, always_apply=False, p=0.5),
    albumentations.OneOf([
        albumentations.CenterCrop(height=200, width=224),
        albumentations.CenterCrop(height=200, width=224),
    ],p=0.5),
```

У вас два раза повторяется поворот (в ShiftScaleRotate и в Rotate), скорее всего так не было задумано. CenterCrop можно заменить на RandomCrop, если цель – создание случайных аугментаций.

```
checkpoint = ModelCheckpoint('best_model.h5' , monitor = ['val_accuracy']
, verbose = 1  , mode = 'max')
```

Здесь нужно добавить параметр save_best_only=True.

```
Xception(include_top=False, input_shape=input_shape),
```

Для этой сети нужна нормализация, но по-моему вы обучаете на генераторе без нормализации. Или возможно вы раскомментировали rescale при обучении Xception, что более правильно.

> Однако точность на валидационной выборке стала значительно отставать от показателя на обучающей выборке, что хорошо видно на графике.

Точность на валидации почти всегда отстает от точности на обучении, но это не всегда повод прекращать обучение модели. Вы можете нарисовать график точности в логарифмическом масштабе по оси эпохи, и тогда станет видно, растет точность на валидации или нет.

Еще надо помнить о том, что ExponentialDecay(LR, 100, 0.96) быстро сведет практически к нулю learning_rate, и модель перестанет обучаться, даже если у нее был потенциал к дальнейшему обучению.

> Использование размера 512 x 512 и соответсвующих настроек аугментации позолило добиться хорошего результата

Да, размер изображений сильно влияет на точность, но увы также и на время обучения.

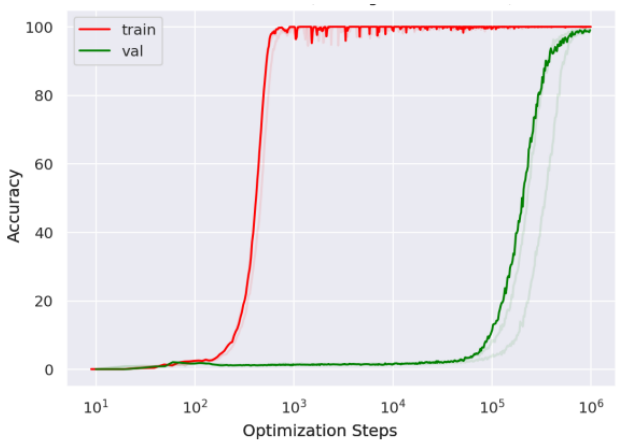
> Этой модели не удалось побить рекорд, но показатели очень неплохие, можно использовать в ансамблировании

Мне кажется, у вас слишком мало эпох, что делать такой вывод. Обычно сети обучаются очень долго.

- **leave it training**. I've often seen people tempted to stop the model training when the validation loss seems to be leveling off. In my experience networks keep training for unintuitively long time. One time I accidentally left a model training during the winter break and when I got back in January it was SOTA ("state of the art").

А еще можете посмотреть эту статью: https://arxiv.org/pdf/2201.02177.pdf

В ней показывается, как точность на валидации начинает расти через много эпох после того, как точность на обучении достигла 100%.




```
for _ in range(4):
  predictions_tta_10.append(model_10.predict(sub_generator, verbose=1))
```

TTA работает тем лучше, чем больше шагов, и при 4 шагах точность наоборот может оказаться ниже, чем без TTA. Поэтому прежде, чем делать сабмит с TTA, нужно понять, дает TTA повышение точности или нет.

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 **29 января**. Анонс вебинара появится позже в канале #0-project_7-ford_vs_ferrari.