

Здравствуйте! Замечательно выполненная работа, далее комментарии по ноутбуку:

- в наивной модели использовал сложную голову на подобии примеров с платформы, как оказалось на следующей же модели этого можно и не делать, результат может быть даже лучше.

Модель со сложной головой может быстрее учиться, но и сильнее переобучается. Я тоже замечал, что простая голова дает результат не хуже.

- все результаты на актуальной (не старой) модели дали более 97% на kaggle, даже без особых заморочек с настройкой головы.

Отчасти потому, что вы используете одну из самых новых сверточных сетей, и отчасти потому, что вы используете большой размер изображения при обучении.

- сейчас понимаю, что некоторые вещи лучше в функции было бы вынести, но ждать такое количество часов на повторный прогон уже не хочу.

Да, если в ноутбуке писать много кода, то в нем очень легко запутаться, лучше по возможности делать функции и далее вызывать их.

- судя по ТТА есть "пограничные" картинки, которые прыгают из класса в класс, в зависимости от примененной аугментации

Скорее всего это в первую очередь классы 7 и 9, там автомобили отличаются только формой задней части кузова, и поэтому при фотографии спереди нельзя понять что это за модель.

```
class_ids = [x for x in range(len(class_names))]
```

Можно было так: `list(range(len(class_names)))`, или `np.arange(len(class_names))`

Хорошо, что вы не делаете «на лету» вычислительно тяжелых аугментаций, таких как ElasticTransform: их часто пытаются применять, они дают мало пользы, но сильно замедляют процесс обучения, так как на выполнение их требуется много времени.

```
reduce_lr = ReduceLROnPlateau(  
    monitor="val_accuracy",  
    factor=0.25,  
    patience=2,  
    min_lr=0.0000001,  
    verbose=1,  
    mode="max",  
)
```

Если есть достаточно времени, то я бы советовал обучать дольше, увеличив patience, и тогда можно получить более высокое качество (в случае сверточных сетей).

```
for _ in range(cycles):  
    predictions.append(model.predict(sub_generator_tta, verbose=1))  
    sub_generator_tta.reset()
```

Как вариант, можно делать больше шагов аугментаций на тех изображениях, на которых предсказания на предыдущих шагах не сходятся. Или можно попробовать сделать ТТА на валидационном датасете и построить график зависимости точности от числа шагов, и так можно будет понять, сколько шагов нужно делать на датасете для сабмита.

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.