> Построил сверточную нейросеть CNN на базе SOTA архитектуры сетей - EfficientNetB5, B0, B3.

EfficientNet – сейчас одна из самых популярных сверточных архитектур, но она уже не является SOTA, выпущены более новые и эффективные модели, такие как EfficientNetv2. Ее можно найти в репозитории tensorflow hub.

```
# !unzip {DATA_PATH}train.zip -d{PATH} слишком много логов
```

Вы можете добавить флаг -q к команде unzip, чтобы она работала ничего не печатая.

> Необходима аугментация так как количество фотографий не достаточно для обучения CNN.

Вот здесь не совсем понятно, а сколько данных достаточно для обучения CNN? По идее какого-то конкретного количества данных, которых было бы достаточно, чтобы аугментации не давали прироста точности.

```
datagen = ImageDataGenerator(rescale=1./255, horizontal_flip=True)
```

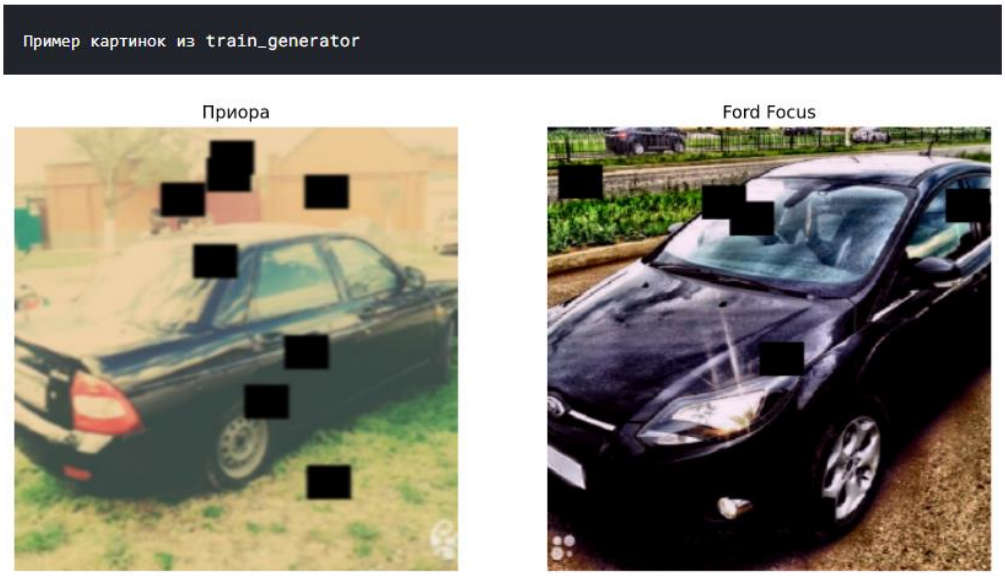
Если EfficientNet берется из keras.applications, то нормализация не нужна, так как встроена в саму модель.

Хорошо, что вы выполняете ручной анализ предсказаний и смотрите, в каких случаях модель ошибается и почему, и есть ли ошибки в разметке датасета. Многие так не делают и даже не пытаются анализировать предсказания модели.

> Ограничения Kaggle на 1000 файлов в датасете подпортили все планы.

Но вы можете поместить эти файлы в архив, а из архива сделать датасет.

При парсинге auto.ru встречаются фотографии изнутри салона, которые в данной задаче не нужны, вы пробовали как-то от них избавляться, вручную или автоматически? Можно использовать предобученную нейросеть для сегментации, чтобы очищать датасет от лишних изображений.



Часто при использовании cutout вырезают большие части изображения. Например, вырезание части фона поможет модели не переобучиться на этом фоне, то есть не пытаться по фону предсказывать класс.

```
# Добавим lr_scheduler (экспоненциально уменьшать скорость через 2 эпохи)
lr_scheduler = ReduceLROnPlateau(monitor='val_loss',
                                 factor=0.2, # уменьшим lr в 5 раз
                                 patience=7, # если нет улучшения через 7 эпохи -
уменьшить lr
                                 min_lr=0.0000001,
                                 verbose=1,
                                 mode='auto')
# Добавим раннюю остановку
earlystop = EarlyStopping(monitor = 'val_accuracy',
                          patience = 7,
                          restore_best_weights = True)
```

У вас параметр patience одинаков в ReduceLROnPlateau и EarlyStopping. Если я правильно помню как работает keras, то смысла в таком нет, поскольку при срабатывании ReduceLROnPlateau сработает также и EarlyStopping. Значит LR никогда не уменьшится.

```
for _ in range(6):
    predictions_tta.append(model.predict(val_sub_generator, verbose=1))
    val_sub_generator.reset()
predictions_tta = np.mean(np.array(predictions_tta), axis=0).argmax(axis=1)
```

Здесь я бы посоветовал сделать TTA на валидации и сравнить точность без TTA и точность с TTA, в зависимости от кол-ва шагов. Если в TTA выполняются сильные аугментации, тогда нужно больше шагов.

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 18 декабря. Анонс вебинара появится позже в канале #0-project_7-ford_vs_ferrari.