

Здравствуйте! Далее комментарии по коду работы:

---

- в моем решении выбрана сеть EfficientNetB6, так как она относится к SOTA на ImageNet, имеет хорошее качество и относительно не большая

Эта архитектура все еще считается очень эффективной, но уже есть более новые, например EfficientNetV2, доступная в tensorflow hub.

Вы делаете постепенную разморозку слоев, но можно было сначала попробовать обучить всю модель целиком сразу (без заморозки слоев), вероятно так точность получилась бы не хуже.

- Попробовать другие архитектуры сетей из SOTA на ImageNet позднее B6, дающие большую точность, например ImageNetB7

Не факт, что B7 дала бы точность выше. B7 – более глубокая сеть, а значит ей легче переобучиться (даже при фэйн-тюнинге) и поэтому нужно больше данных. Так что на не слишком большом объеме обучающих данных, возможно, эффективнее окажутся модели среднего размера.

- Попробовать больше эпох на 5 этапе обучения (увеличить до 30 эпох с callback ReduceLROnPlateau с параметрами monitor='val\_accuracy', factor=0.2-0.5, patience=3-5).

Patience я бы посоветовал ставить выше. Часто бывает, что точность как будто идет вниз, а затем снова идет вверх, поэтому имеет смысл ждать дольше. Представьте, что вы тренируете спортсмена, и прекращаете тренировки если в течение 2 тренировок подряд его результаты не растут. В итоге спортсмен мог бы тренироваться годами, а вы прекратите тренировать его уже через неделю. Но это ведь неправильно: может быть много случайностей, которые влияют на результат в какой-то из дней. Так же и с нейросетями: график точности подвержен случайным колебаниям, поэтому patience лучше делать существенно выше.

```
# I choose transformations that more approximate to images in task
# that why I exclude transformations and distortions of forms of cars
# but we can try use it in training model
def augment_data():
    AUGMENTATIONS = augmentations.Compose([
        augmentations.CLAHE(...),
        augmentations.ChannelShuffle(...),
        ...
```

Часто оказывается, что простые аугментации дают в итоге точность не хуже, а то и лучше, чем сложные. При этом сложные аугментации сильно замедляют обучение из-за того, что на их выполнение также требуется время.

```
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.

```
ModelCheckpoint(PATH_BEST_MODEL+'best_model.hdf5', monitor='val_accuracy',
                verbose=1, mode='max', save_best_only=True)
```

Хорошо что добавили `save_best_only`, часто этот параметр пропускают, и в итоге сохранение происходит каждую эпоху.

```
base_model = efn.EfficientNetB6(weights='imagenet', include_top=False,
input_shape=input_shape)
```

EfficientNet есть в `keras.applications`, поэтому нет особого смысла в установке `pip`-библиотеки `EfficientNet`.

В целом в ноутбуке очень много копирования кода. Например, если вам нужно пересоздавать генераторы, то напишите функцию для создания генераторов и вызывайте ее несколько раз. Так код получится раз в 5 короче, а следовательно понятнее.

```
label_map = (train_generator.class_indices)
label_map = dict((v,k) for k,v in label_map.items()) #flip k,v
predictions = [label_map[k] for k in predictions]
```

Мне кажется что конкретно на данном датасете смысла в этом действии нет, ведь `label_map` будет содержать записи такие как 1 -> «1». То есть удалив эти две строки, вы получите тот же результат в файле `submission.csv`.

---

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 **5 февраля**. Анонс вебинара появится позже в канале #0-project\_7-ford\_vs\_ferrari.