

Здравствуйте! Далее комментарии по коду работы:

В ноутбуке отсутствуют выводы ячеек, так что если в них что-то важное (визуализация, графики), то не смогу посмотреть.

- Как мы видим, данные содержат 10 классов, а распределение классов достаточно равномерное.

Даже если бы оно было неравномерным, целевая метрика у нас accuracy, а не F1, поэтому балансировать классы не обязательно.

```
train_datagen = ImageDataGenerator(  
    rescale=1. / 255,
```

В разных моделях нужна разная нормализация, но почти нигде она не равна делению на 255. Например, для Xception требуется нормализация `tf.keras.applications.xception.preprocess_input` – это не то же самое, что деление на 255. Для EfficientNet из `keras.applications` нормализация не требуется.

```
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.

В новом бейзлайне, ссылка на который есть на странице проекта на lms.skillfactory.ru, показан пример того, как делать валидацию без аугментаций.

- Данная сеть не подходит, так как сеть является 164 слоями глубоко и может классифицировать изображения в 1 000 категорий объектов, таких как клавиатура, мышь, карандаш и многие животные

Здесь непонятно что вы имеете в виду, да, сеть действительно предобучается на задаче классификации по 1000 классам (клавиатура, мышь, карандаш, ...), но затем можно заменить последний слой сети и дообучить ее, чтобы она классифицировала 10 классов автомобилей.

```
base_model = EfficientNetB2(include_top=False, weights='imagenet', input_tensor=None,  
    pooling=None, classes=1000, classifier_activation='softmax')
```

Параметры `classes`, `classifier_activation` не играют роли, если указано `include_top=False`. Ведь задача в том, чтобы получить сеть без последнего слоя.

```
base_model = Sequential()  
base_model.add(Dense(224, input_shape=(input_shape)))  
base_model.add(Activation('relu'))  
base_model.add(Conv2D(224, (3, 3)))  
base_model.add(Activation('relu'))  
base_model.add(Dense(64))  
base_model.add(Activation('relu'))  
base_model.add(Dropout(0.5))  
base_model.add(Dense(10))
```

```
base_model.add(Activation('sigmoid'))
base_model.add(BatchNormalization())
```

Не совсем понимаю что это за сеть, здесь явно ошибки в архитектуре, например, отсутствует слой flatten или global pooling. Поэтому естественно такая сеть не даст высокой точности.

```
x = base_model.output
x = GlobalAveragePooling2D()(x)
# let's add a fully-connected layer
x = Dense(256, activation='relu')(x)
x = Dropout(0.25)(x)
# and a logistic layer -- let's say we have 10 classes
predictions = Dense(CLASS_NUM, activation='softmax')(x)
model = Model(inputs=base_model.input, outputs=predictions)
```

Вот это как раз то, что нужно, здесь в предобученную сеть добавляется новый выходной слой для классификации по 10 классам.

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

Такой код неправильно работает, но почему-то все равно все его используют, и об этом написано практически в каждом отзыве в [репозитории отзывов](#) на проверенные работы. Нужно добавить параметр save_best_only=True.

```
# tta_steps = 10
# predictions = []

# for i in tqdm(range(tta_steps)):
#     preds = model.predict_generator(train_datagen.flow(x_val,
batch_size=bs, shuffle=False), steps = len(x_val)/bs)
#     predictions.append(preds)
```

Судя по тому, что вы закомментировали ТТА, этот способ не дал роста точности. Здесь важная особенность в том, что если в ТТА выполняются сильные аугментации, тогда нужно больше шагов, то есть больше 10.

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.