

Здравствуйте! Далее комментарии по коду работы:

```
def learning_graphic(history):
```

Может быть стоит назвать graph или plot, слово graphic означает «графический». И в ноутбуке по [ссылке](#) нет выводов ячеек кода, поэтому если в них было что-то важное (графики, результаты аугментаций), то не могу посмотреть.

```
print('Распаковываем картинки')
# Will unzip the files so that you can see them..
for data_zip in ['train.zip', 'test.zip']:
    with zipfile.ZipFile(DATA_PATH + data_zip, "r") as z:
        z.extractall(PATH)
```

Можно было использовать команду `!unzip -q { DATA_PATH + data_zip } -d {PATH}`.

```
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=30, interpolation=1, border_mode=4,
value=None, mask_value=None, always_apply=False, p=0.5),
albumentations.OneOf([
    albumentations.CenterCrop(height=224, width=200),
    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.

```
train_datagen = ImageDataAugmentor(
    rescale=1/255,
```

В разных моделях нужна разная нормализация, но почти нигда 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.

```
test_datagen = train_gen.flow_from_directory(PATH+'train',
```

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.

```
model.add(Layer.Dense(256,
                      activation='relu',
                      bias_regularizer=l2(1e-4),
                      activity_regularizer=l2(1e-5)))
model.add(Layer.BatchNormalization())
```

Непонятно из каких соображений добавлены эти слои именно с такими параметрами. Но вы написали, что пробовали разные виды голов, такая оказалась наилучшей? Тогда было бы хорошо сделать таблицу и в ней показать разные архитектуры головы и достигнутые значения точности.

```
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.

EPOCHS = 4 # эксперименты показали, что такого количества эпох
будет достаточно

Скорее всего 4 эпох все-таки недостаточно. Если после 4-й эпохи точность упала – это еще ничего не значит, может быть это случайность, и далее она снова вырастет.

```
for _ in range(EPOCHS):  
    predictions_tta.append(model.predict(sub_generator, verbose=1))  
    sub_generator.reset()
```

Почему ТТА делается столько же шагов, сколько было эпох? Это ведь разные по смыслу значения. Но в принципе количество шагов ТТА можно выбирать произвольно, проблема только в том, что при малом числе шагов ТТА наоборот может ухудшить точность.

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.