

```
AUGMENTATIONS = A.Compose([
    ...
])
```

Вижу что разобрались с аугментациями через albumentations, отлично!

```
test_datagen = train_gen.flow_from_directory(
```

Валидация (test_datagen) делается также 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.

```
!pip install -q efficientnet
import efficientnet.keras as efn
```

Сейчас установка этой библиотеки уже не требуется, efficientnet есть в составе keras.applications.

```
model=Model.Sequential()
model.add(base_model)
model.add(Layer.GlobalAveragePooling2D())
model.add(Layer.Dense(256,
                      activation='relu',
                      bias_regularizer=l2(1e-4),
                      activity_regularizer=l2(1e-5)))
model.add(Layer.BatchNormalization())
model.add(Layer.Dropout(0.25))
model.add(Layer.Dense(CLASS_NUM, activation='softmax'))
```

Обычно (даже в серьезных научных исследованиях) голову сети делают очень простой: в ней только global pooling и выходной слой. Однако это не говорит о том, что сложная голова хуже обучится. По сути на текущем уровне развития deep learning'a выбор архитектуры – это алхимия: нужно пробовать и сравнивать.

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

Тут нужно добавить save_best_only=True, иначе модель будет сохраняться после каждой эпохи.

Step 2

defrost a half of the pre-trained model weights

А вы пробовали обучать всю модель сразу, без постепенной разморозки слоев? Возможно результат был бы не хуже, а то и лучше.

```
tta_steps = 10
for i in range(tta_steps): ...
```

Можно как вариант делать ТТА больше раз на сомнительных изображениях. Если на изображении все 10 ответов совпали – дальше не продолжаем, а если не совпали – делаем еще больше итераций ТТА на данном изображении.

100% weights defrosting accuracy 93.99%
image size rising accuracy 96.82%

Да, к сожалению точность современных сверточных сетей сильно зависят от размера изображения. Почему это так – сложно сказать, возможная причина описана в этой статье: <https://habr.com/ru/company/ods/blog/453788/>

SOTA архитектура семей - Xception, InceptionV3

А из чего следует что это SOTA архитектура? То что многие в лидерборде используют эту модель – не означает, что это лучший вариант. Все просто копируют бейзлайн. Но бейзлайн – это именно базовое, а не наилучшее решение. Наилучший результат наверняка достигается на более современных моделях. Можно было бы например взять например модель efficientNetV2 из tensorflow_hub.

В целом хорошая работа! Вижу что разобрались во всех важных моментах.