Использовал пайплайн из мдз2:

1) загружаю весь датасет в память, потому что помещается
2) создаю пайплайн аугментаций с помощью torchvision
3) генерирую шафленные батчи
4) тренирую и валидирую, сохраняя метрику точности

Пайплайн на 52.5% для соревнования

- на трейне
    - случайный кроп [8%;100%]
    - ресайз до [224;224] с билинейной интерполяцией
    - конвертация пикселей во флоат на отрезке [0;1]
- на тесте
    - ресайз до [256;256] с бил. инт.
    - центр кроп [224;224]
    - конвертация во флоат [0;1]
- Модель resnet34, последний слой это dense(200)+dropout(20%)+dense(200)
- оптимизатор SGD(lr=0.01, momentum=0.9, weight_decay=0.001)
- шедулер ReduceLROnPlateu, ждет 4 эпохи без улучшения лосса на валидации и уменьшает lr в 5 раз
- батч размера 64, тренировка 60 эпох, выбор лучшей модели по точности на валидации
- лосс CrossEntropy

Пайплайн на 58.3% на окончательную сдачу

- на трейне
    - случайный горизонтальный флип
    - случайный кроп [50%;100%]
    - ресайз до [224;224] с бил. инт.
    - случайный поворот [-10;10] градусов с бил. инт.
    - конвертация во флоат [0;1]
    - нормировка по каналам со статистикой из imagenet
- на тесте
    - ресайз до [256;256] с бил. инт.
    - центр кроп [224;224]
    - конвертация во флоат [0;1]
    - нормировка по каналам как на трейне
- Модель resnet34, последний слой это dense(300)+dropout(50%)+dense(200)
- оптимизатор, шедулер, лосс как в посылке на 52.5%
- батч и тренировка как в посылке на 52.5%

Ниже есть про эксперименты

Эксперименты

- Основной проблемой было переобучение, точность на трейне спокойно доходила до ~100%, но на валидации упиралась в порядка 60%. Из всех придуманных экспериментов есть только один, который я не успел провести, это аугментация картинок с помощью mixup. Возможно, он дал бы прирост.
- Результат на валидации равен порядка 61%, но на тесте порядка 58%, то есть на тесте стабильно на пару процентов меньше чем на валидации. Я считаю, что дело не в переобучении под валидацию, а в другом распределении на тесте и валидации. У меня были идеи как это можно исправить, но они не были состоятельными и не работали на практике. Например я хотел бы посчитать распределение классов на тесте через построенную модель, но распределение сильно зависит от модели, так что этот вариант не подойдет.
- Пытался брать различную статистику для нормировки по каналам, с трейна, с трейна+валидации, отдельно для трейна и отдельно для валидации. Последнее вроде как не работало, потому что я нормирую валидацию и трейн по разным статистикам, что плохо. В целом не увидел прироста, поэтому оставил значения imagenet.
- Пытался добавлять различные встроенные аугментации
    - AutoAugment, TrivialAugmentWide. На первых 10 эпохах валидация опустилась до 0% и не улучшалась
    - Бикубическая интерполяция где можно, сильно замедляла время работы, прирост не дала
    - другие проценты для кропа на трейне вместо [50%;100%]. На практике я считаю, что именно 50% здесь и 50% в дропаут на последнем слое дали прирост с 52% до 58% точности. Изменение этого значения ухудшает результат.
    - Размытие по Гауссу, инвертация цветов, randomPosterize, randomAdjustSharpness, randomAutocontrast. Ничего не улучшило результат. Я предполагаю, что результат этих преобразований схож с autoaugment но в меньших масштабах: тренировочных сет становится слишком непохож на реальные картинки.
    - изменить угол поворота, не повлияло.
- Попробовал добавить регуляризатор SAM https://arxiv.org/abs/2010.01412 Не помогло
- Попробовал регуляризатор SWA, тренировка занимала много времени, поэтому после двух неудачных(не улучшило результат) попыток я закончил
- Попробовал ограничить норму градиента, ухудшило результат
- Попробовал искусственно увеличить размер батча, результат не изменился
- Попробовал использовать другие модели
    - resnet18 => resnet34 улучшило результат
    - resnet50, wideresnet50, resnext50 ухудшили результат, переобучились
- Вначале, когда точность была порядка 40%, пробовал поменять оптимизатор на adam и вариации. Результат получился хуже, скорее всего из-за переобучения. Далее не пытался это делать, ведь adam легко переобучается.
- Пробовал менять параметры SGD: lr, momentum, weight_decay, nesterov. По итогу очень важно иметь weight_decay > 0 но не слишком большой, lr=0.01 оказался оптимальным начальным значением, нестеров ухудшил результат, предполагаю переобучился.

- Косинусный шедулер не пробовал, не думаю что он улучшил бы результат. К моему шедулеру у меня претензий нет.
- Пытался в своем шедулере менять параметры, в основном число эпох до уменьшения и коэффициент уменьшения lr.
- Пытался ставить разную модель вместо последнего слоя. Здесь важным оказалось настэкать несколько полносвязных слоев, причем дропаут 50% между ними оказался решающим фактором, который дал большой прирост. Как уже говорилось, основной задачей было исправить переобучение.