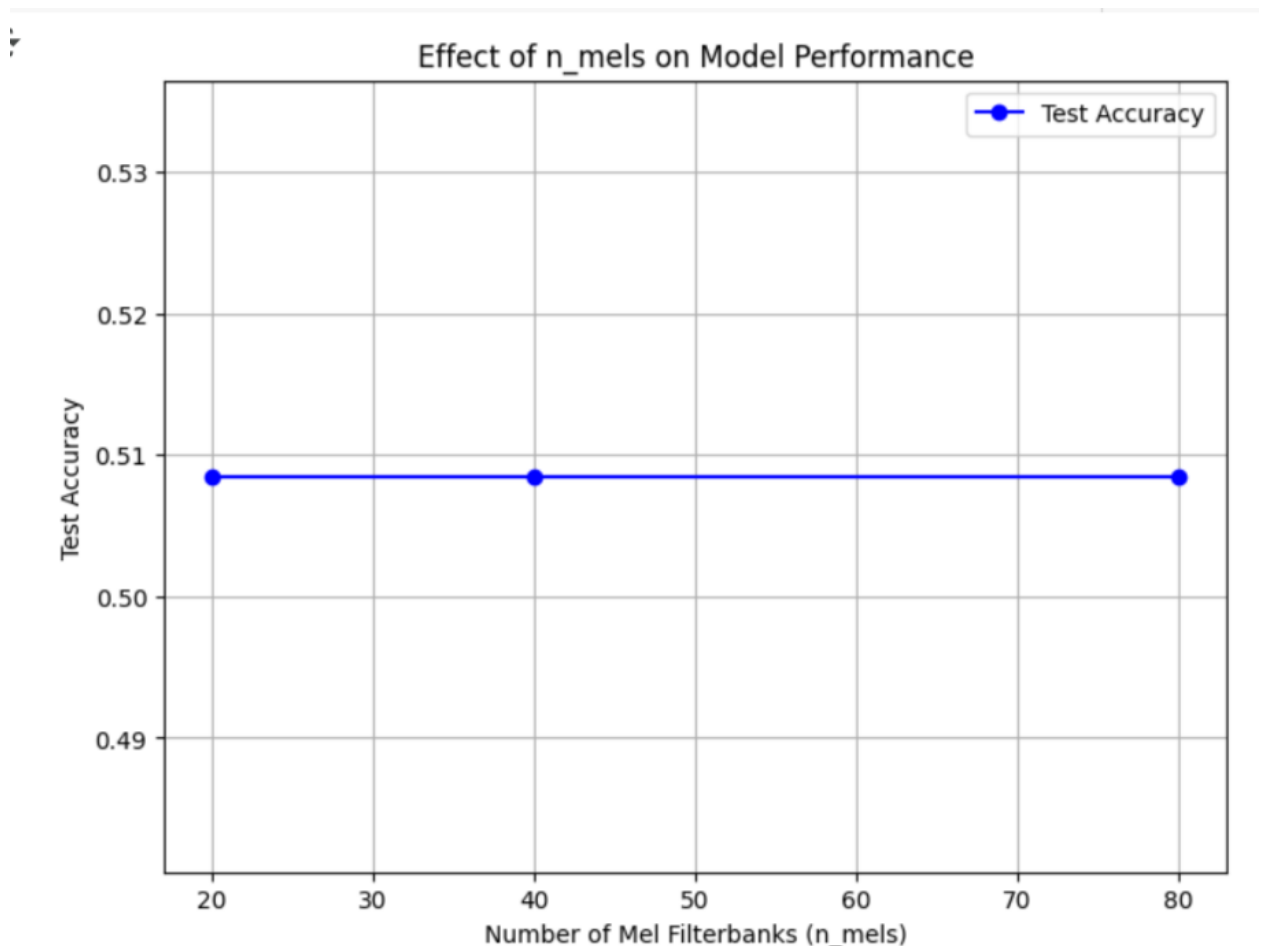


Отчет по дисциплине: Распознавание и генерация речи.

При обучении CNN модели с признаками LogMelFilterBanks для бинарной классификации на данных [Google Speech Commands](#)

график показывает как меняется **точность** модели Test Accuracy при изменении количества фильтров n\_mels:



Результаты обучения модели:

n\_mels=20, Test Accuracy: 0.5085, Test Loss: 45.9757

n\_mels=40, Test Accuracy: 0.5085, Test Loss: 0.7024

n\_mels=80, Test Accuracy: 0.5085, Test Loss: 0.6991

График Test Accuracy остается примерно постоянным при изменении n\_mels, возможно то, что увеличение фильтров не влияет на производительность модели.

График отображает постоянную точность (**Test Accuracy**) на уровне 0.50 независимо от значения n\_mels.

Возможно, CNN адаптируется к входным данным одинаково.

Проведена проверка распределения классов в данных, чтобы определить, что они сбалансированы: Counter({1: 3228, 0: 3130})

Данные сбалансированы и задача с различными классами, возможно, по этой причине модель выдает одинаковые результаты при подаче параметров `n_mels=20, 40, 80`.

Изменение длины на динамическую в методе `def _get_flattened_size(self, n_mels, seq_len=101)` -101 задаем как параметр.

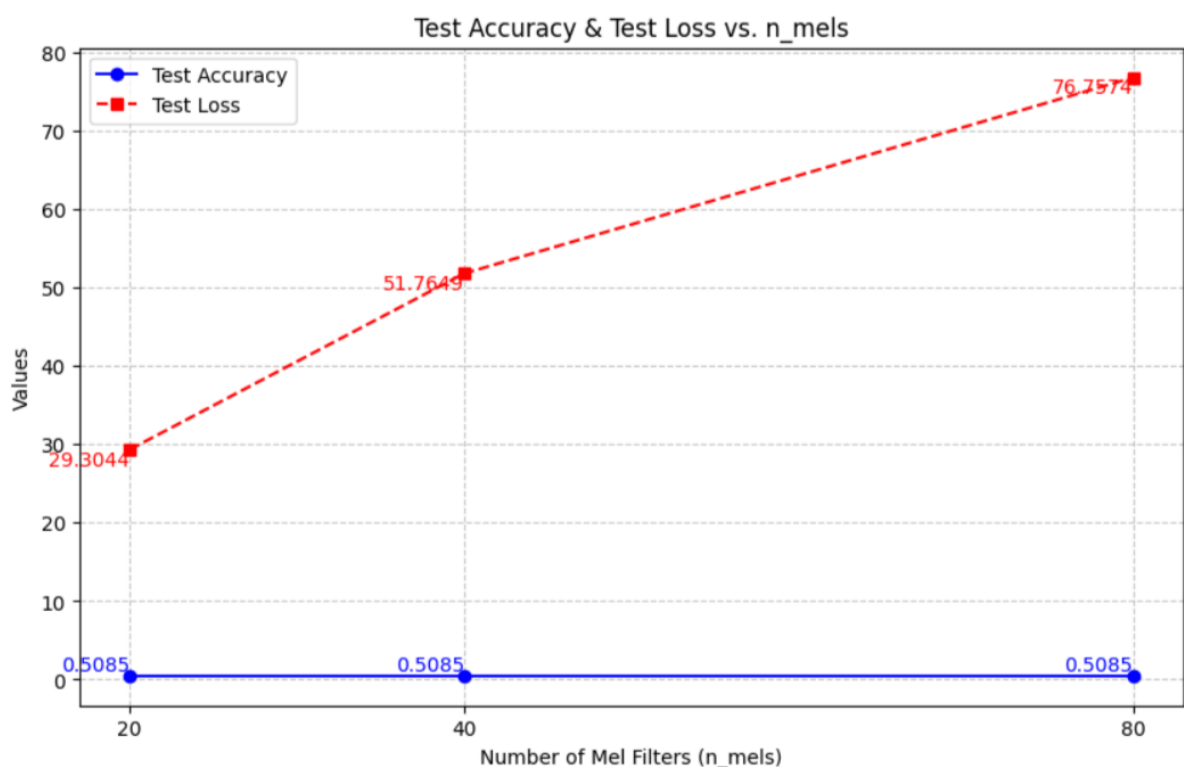
Увеличение количества эпох и добавлена ранняя остановка `-early_stopping-` останавливаем обучение при отсутствии улучшений:

Получились следующие результаты:

`n_mels=20`, Test Accuracy: 0.5085, Test Loss: 29.3044

`n_mels=40`, Test Accuracy: 0.5085, Test Loss: 51.7649

`n_mels=80`, Test Accuracy: 0.5085, Test Loss: 76.7574



Построены два графика -синий показывает `test_acc` точность модели, красный показывает `test_loss` лосс модели, ось Y – Values это значения мел-фильтров [20, 40, 80].

Значение `test_acc` остаётся почти одинаковым для всех значений `n_mels`. Поэтому увеличение количества фильтров `n_mels` не сильно влияет на точность модели.

Значение `test_loss` увеличивается с увеличением `n_mels`.

Останавливаемся на `n_mels=20`, потому что меньше `loss`, точность и так одинаковая и вычислений будет меньше.

Значения для `groups`:

- 1 — обычная свёртка
- 2 — группированная свёртка с двумя группами.

4 — группированная свёртка с четырьмя группами.

Результаты обучения:

groups=1, Test Accuracy: 0.5085, Test Loss: 0.6930, Epoch Time: 22.2896s, Params: 925954, FLOPs: 6587778

groups=2, Test Accuracy: 0.5085, Test Loss: 18.1537, Epoch Time: 21.9896s, Params: 872962, FLOPs: 3742338

groups=4, Test Accuracy: 0.5085, Test Loss: 13.7783, Epoch Time: 21.7195s, Params: 846466, FLOPs: 2319618

Высокие значения test Loss при группах 2 и 4.

Чем больше групп, тем меньше параметров.

FLOPs (Количество операций):

Группированная свёртка снижает количество операций и ускоряет модель.

График зависимости времени от групп:

Время на эпоху уменьшается с увеличением groups.

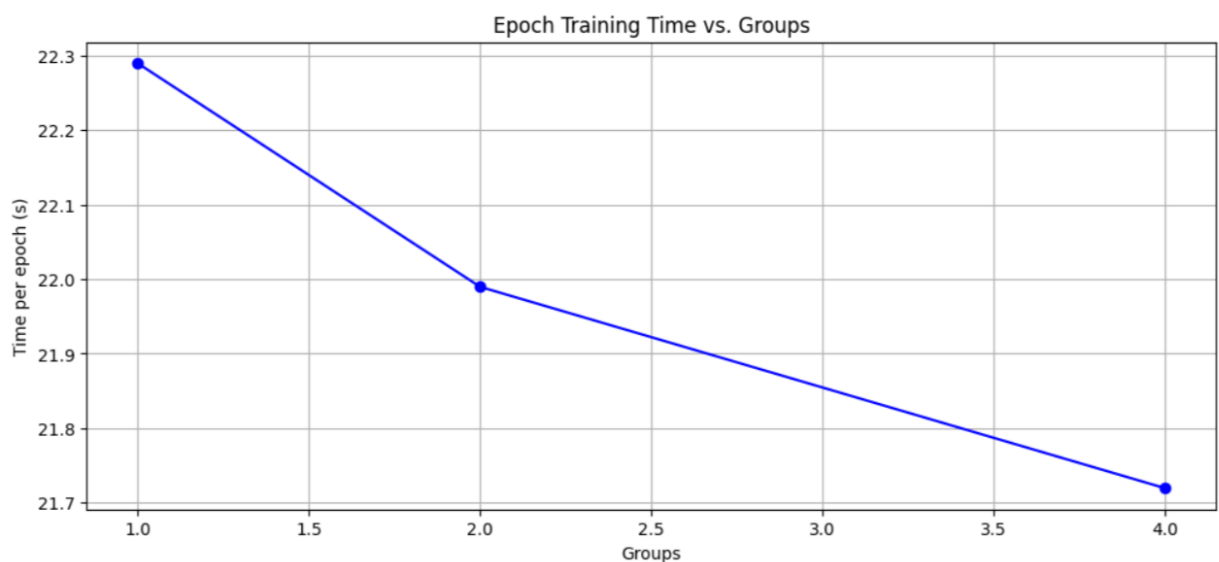


График зависимости параметров модели от групп

Увеличение количества groups сокращает количество параметров и уменьшает сложность модели.

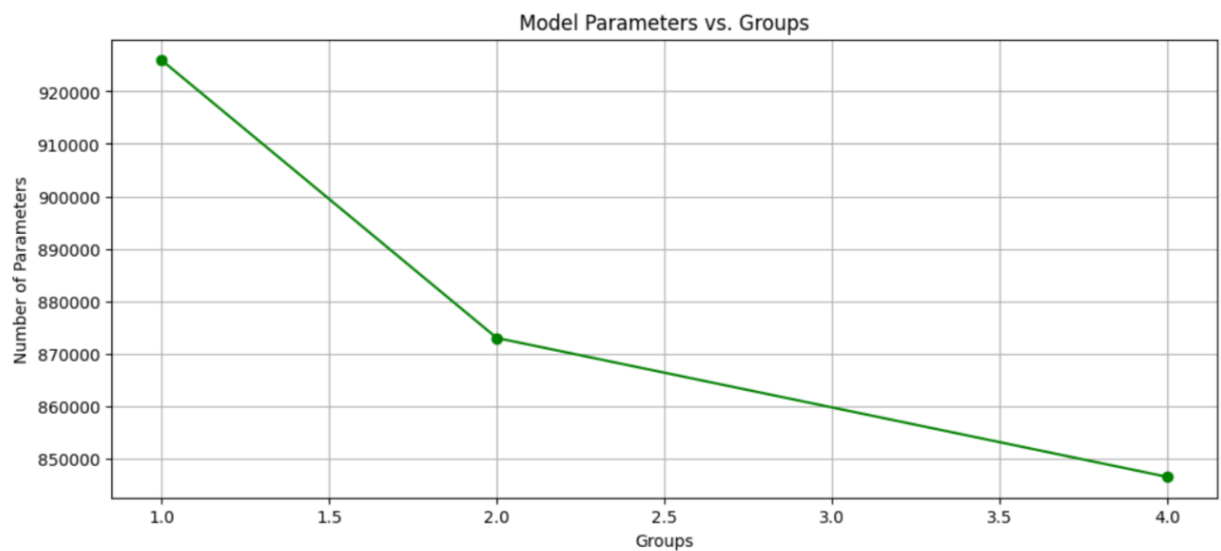
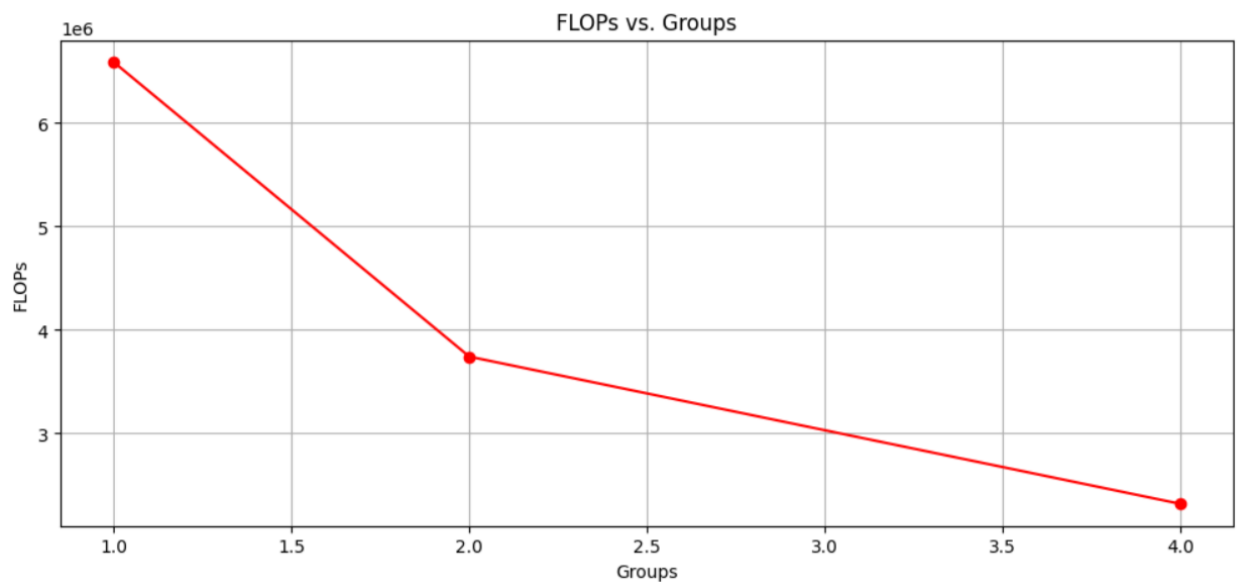


График зависимости количества операций от групп:

Увеличение groups уменьшает объем вычислений и снижает требования к ресурсам процессора или GPU.



Выводы:

Увеличение groups:

- сокращает время обучения и уменьшает количество параметров и FLOPs.
- модель демонстрирует производительность по `test_acc`, оставаясь на значении 0.50.