

Класифікація зображень на GPU за допомогою нейронних мереж

Підготувала
Галушко Марія

Опис задачі

- **Мета:** виявлення ефективних способів та засобів класифікації зображень у прикладних системах, таких як категоризація товарів на сайті.
- **Предмет дослідження:** аналіз способів і засобів використання нейронних мереж у прикладних системах та отримання такої мережі, яка б відповідала поставленим цілям роботи.
- **Об'єкт дослідження:** архітектури нейронних мереж, для навчання на GPU, що дозволяють отримати оптимальну мережу для використання її в реальних задачах з прийнятною точністю.

Наукова новизна

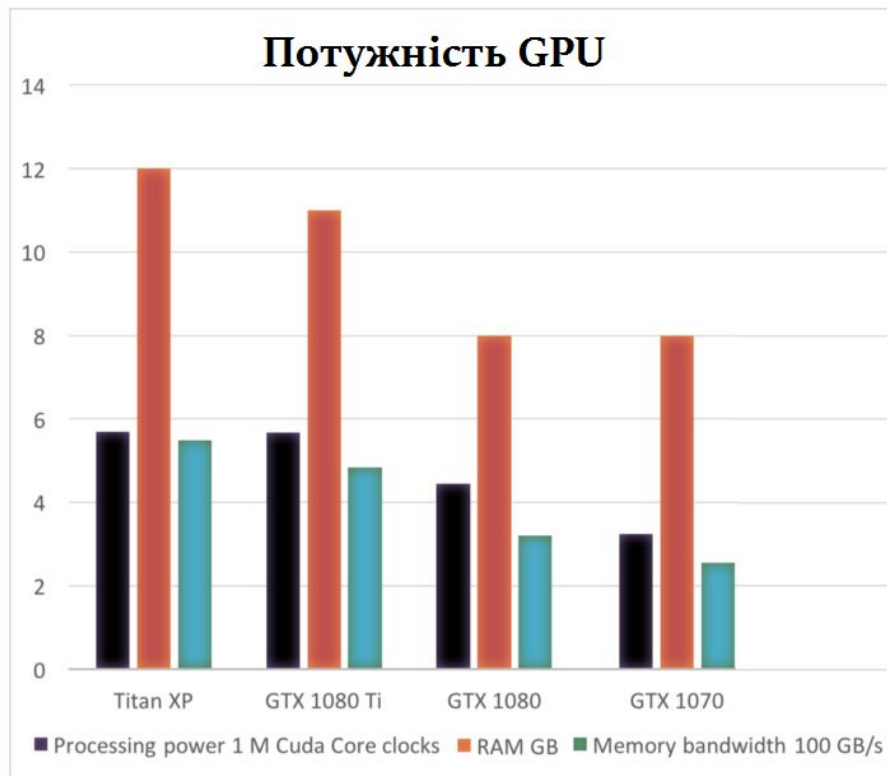
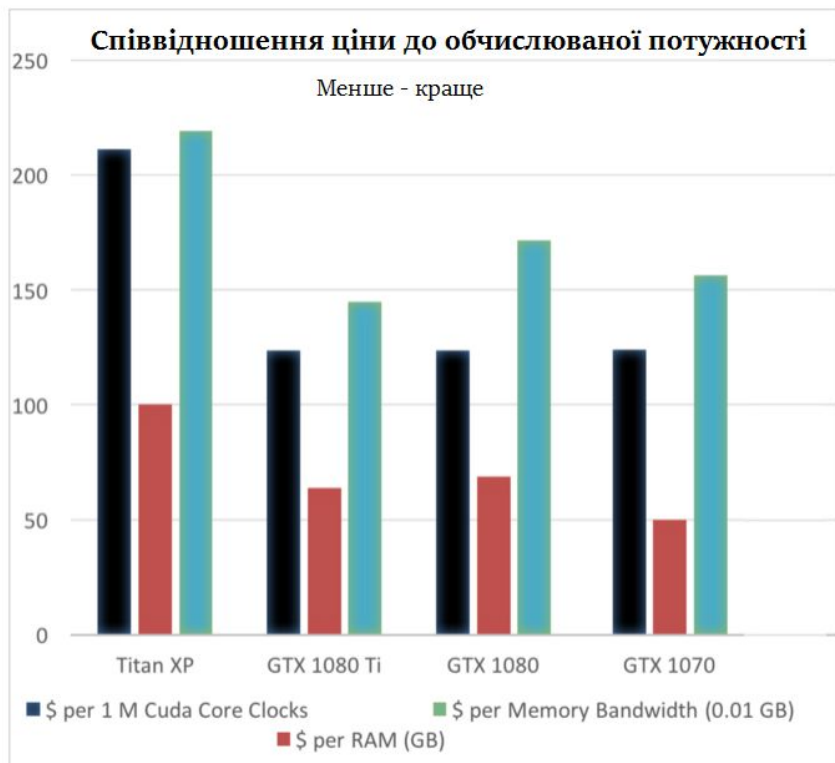
Підібрано архітектуру для класифікації зображень,
реалізовано метод обміну знаннями на GPU,
поєднано декілька методів та підібрано параметри
і застосовано для конкретної задачі.

Чому GPU?

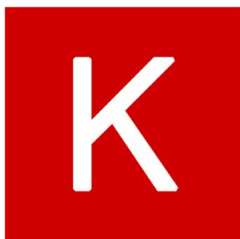


Процесор	Час навчання на різних наборах даних	
	MNIST, с	CIFAR-10, с
Titan XP	0.005	21.10
GTX 1080 Ti	0.005	26.45
GTX 1080	0.0055	27.8
GTX 1070	0.0055	29.6
Intel Core i7	0.3777	1665.7

GPU



Використані інструменти

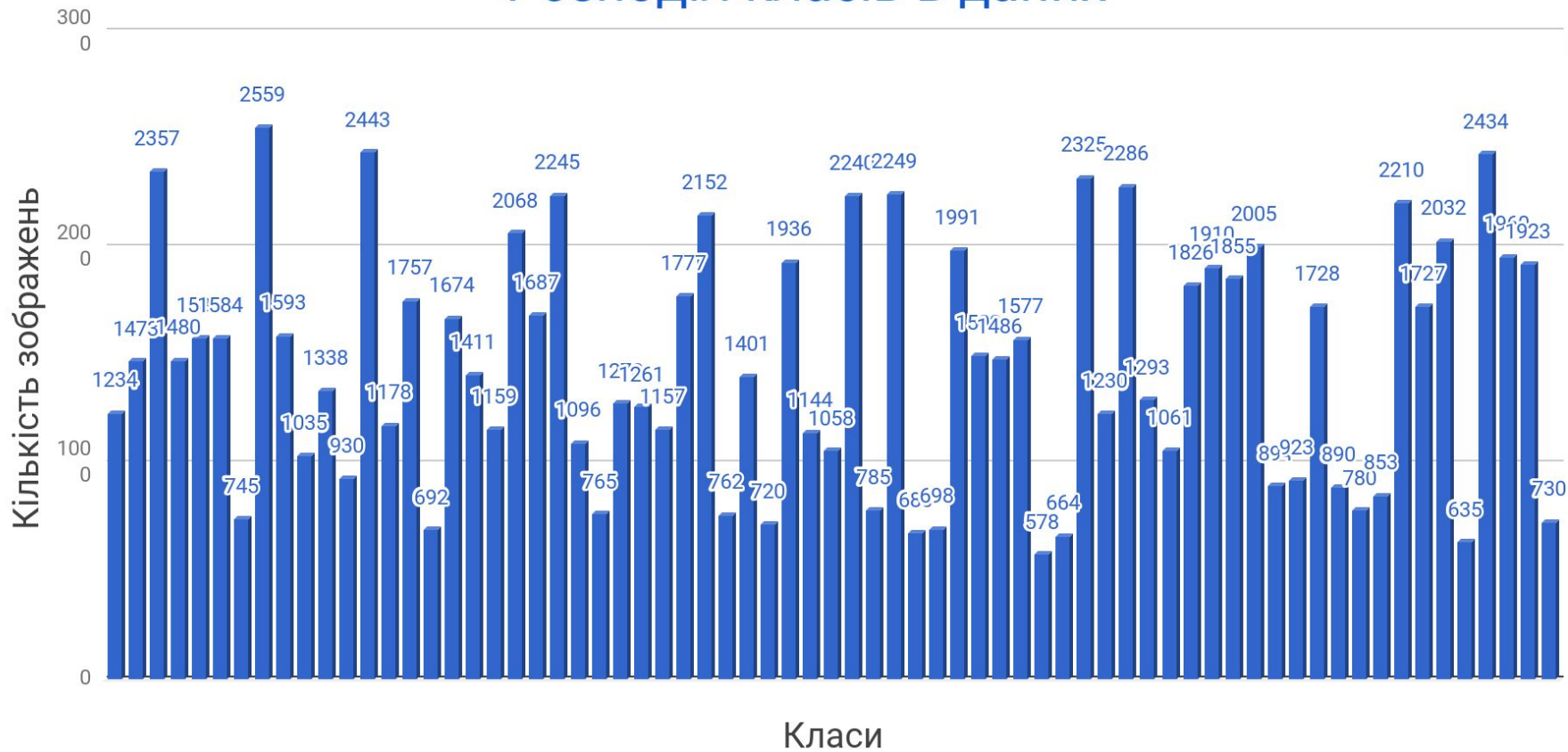


Keras

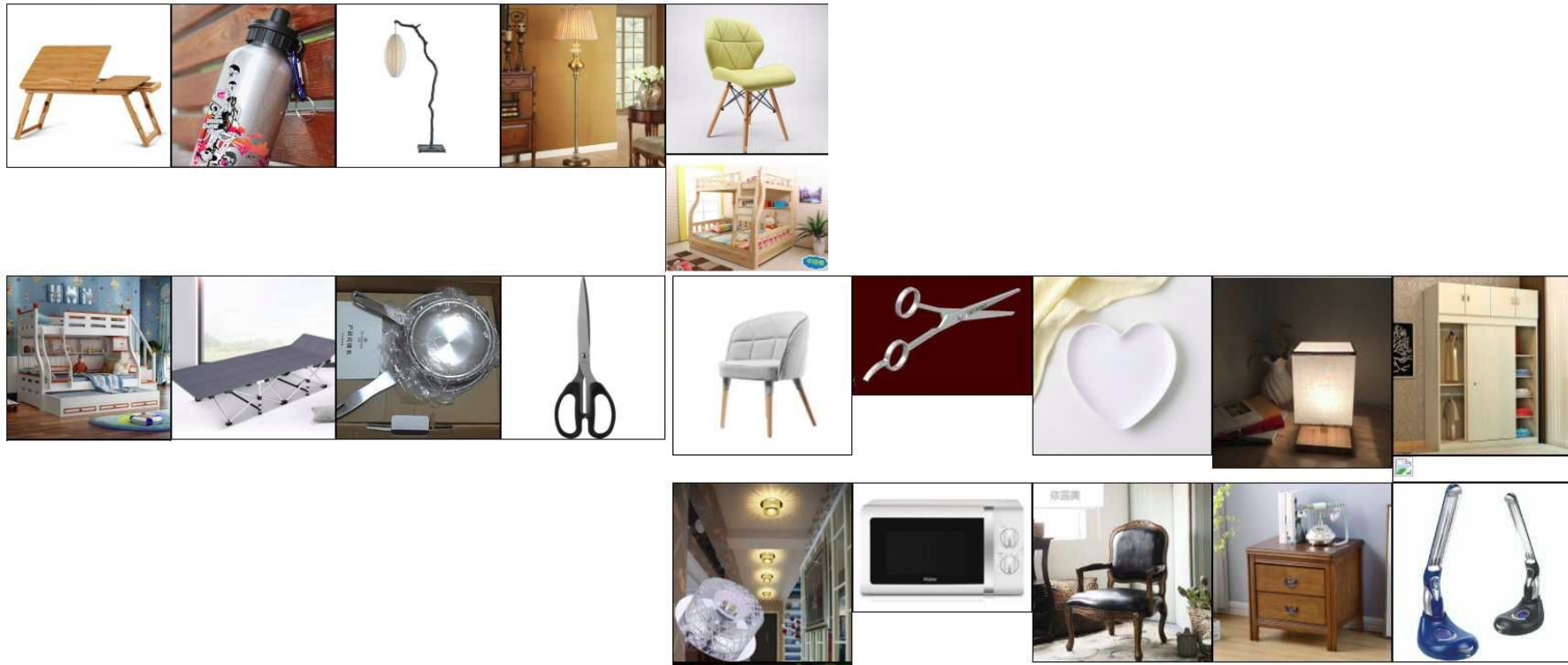
Опис даних

- Дані було взято із змагання на платформі **Kaggle**;
- Всього було **69** класів, і **104461** зображень;
- На кожен клас було в середньому по **1400** екземплярів, найменший клас містив **578** зображень і найбільший – **2559**.

Розподіл класів в даних



Приклади тестових та тренувальних даних



Приріст зображень

Вихідне зображення



Створені



Використані підходи. Базова модель нейронної мережі

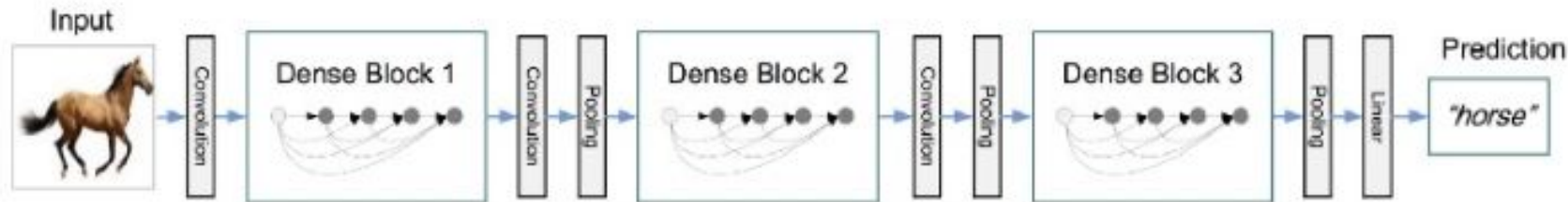


- Модель була взята за основу, з результатами якої порівнювались інші
- **61,2%** – максимальна точність, яку вдалось досягти

Структура одного блоку

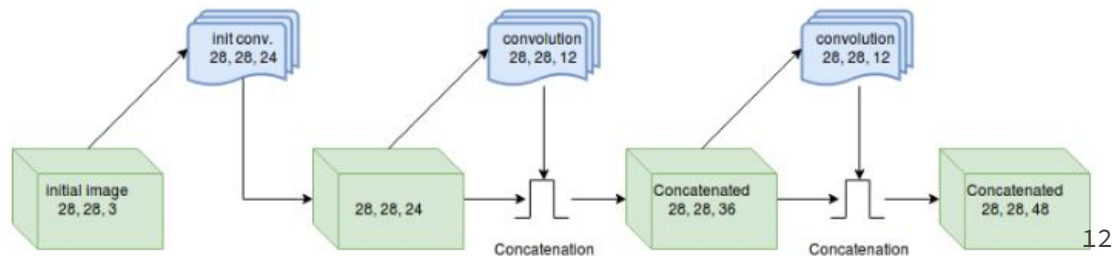


Використані підходи DenseNet

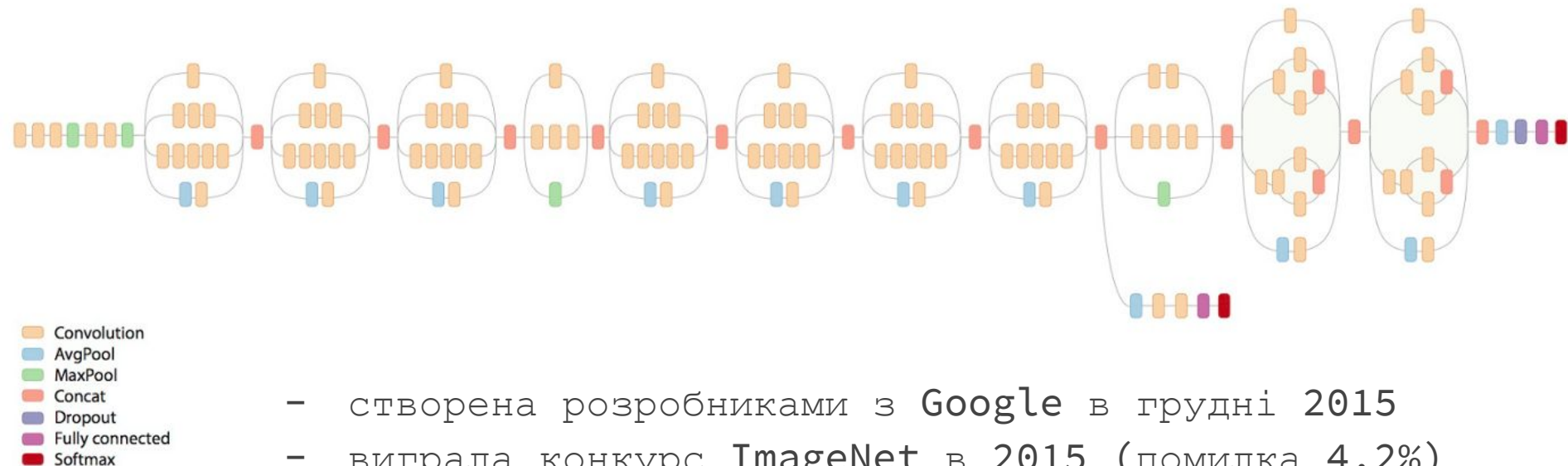


- архітектура була описана в грудні 2017 року
- вона показала найнижчу похибку з усіх існуючих архітектур на CIFAR/SVHN наборах даних

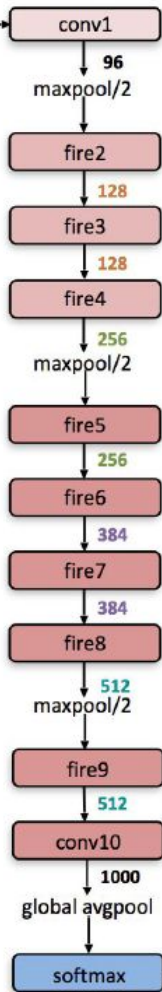
Структура одного блоку



Використані підходи. InceptionV3



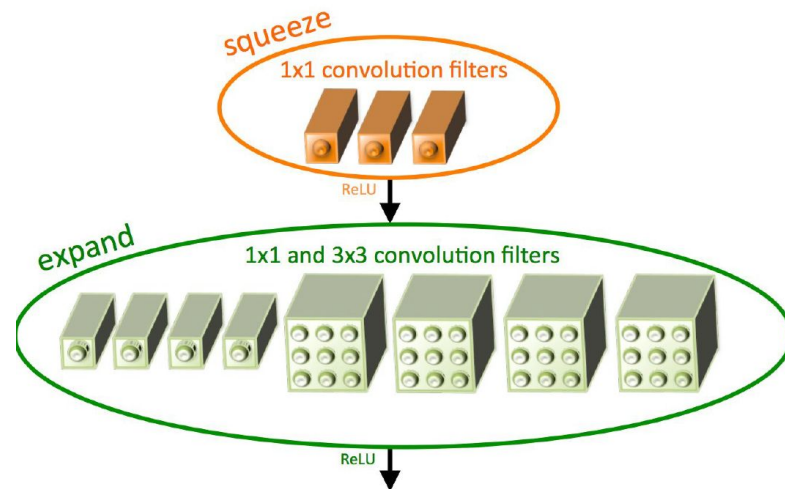
- створена розробниками з Google в грудні 2015
- виграла конкурс ImageNet в 2015 (помилка 4.2%)



Використані підходи. SqueezeNet

- вийшла 4 листопада 2016
- потребує небагато обчислювальних потужностей, швидко навчаються
- можна використовувати для багатьох задач, таких як самокеровані автомобілі, доповнена реальність, робототехніка;

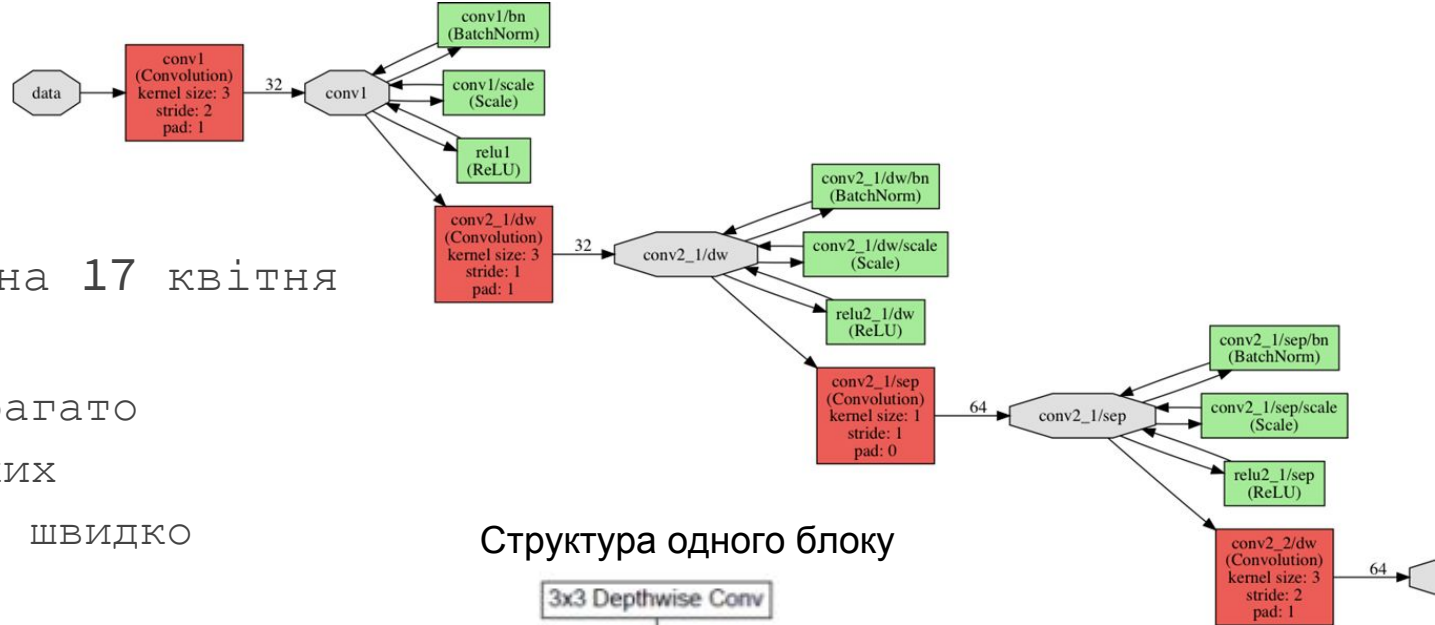
Структура одного блоку



Лаборатор ретривер

Використані підходи MobileNet

- Була створена 17 квітня 2017 року
- потребує небагато обчислювальних потужностей, швидко навчається
- можна використовувати для багатьох задач, таких як самокеровані автомобілі, доповнена реальність, робототехніка

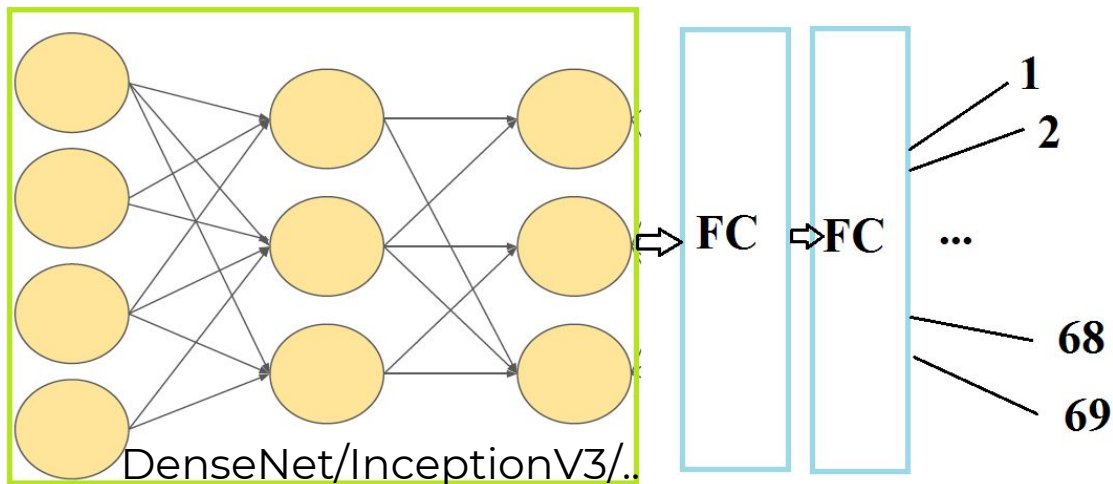


Структура одного блоку

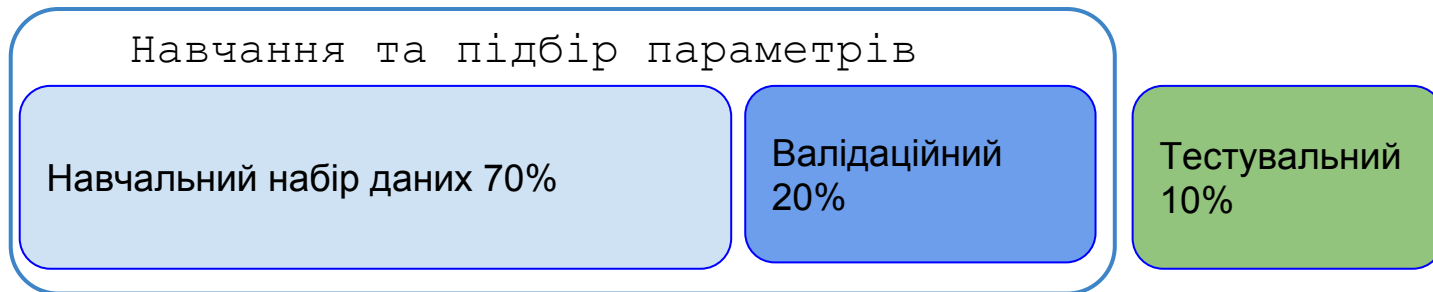


Передача знань

У мережі, яка навчена на **ImageNet** виділена прямокутником “заморозили” ваги і додали декілька повнозв’язних шарів, які до навчили на власному наборі даних (предметів побуту)

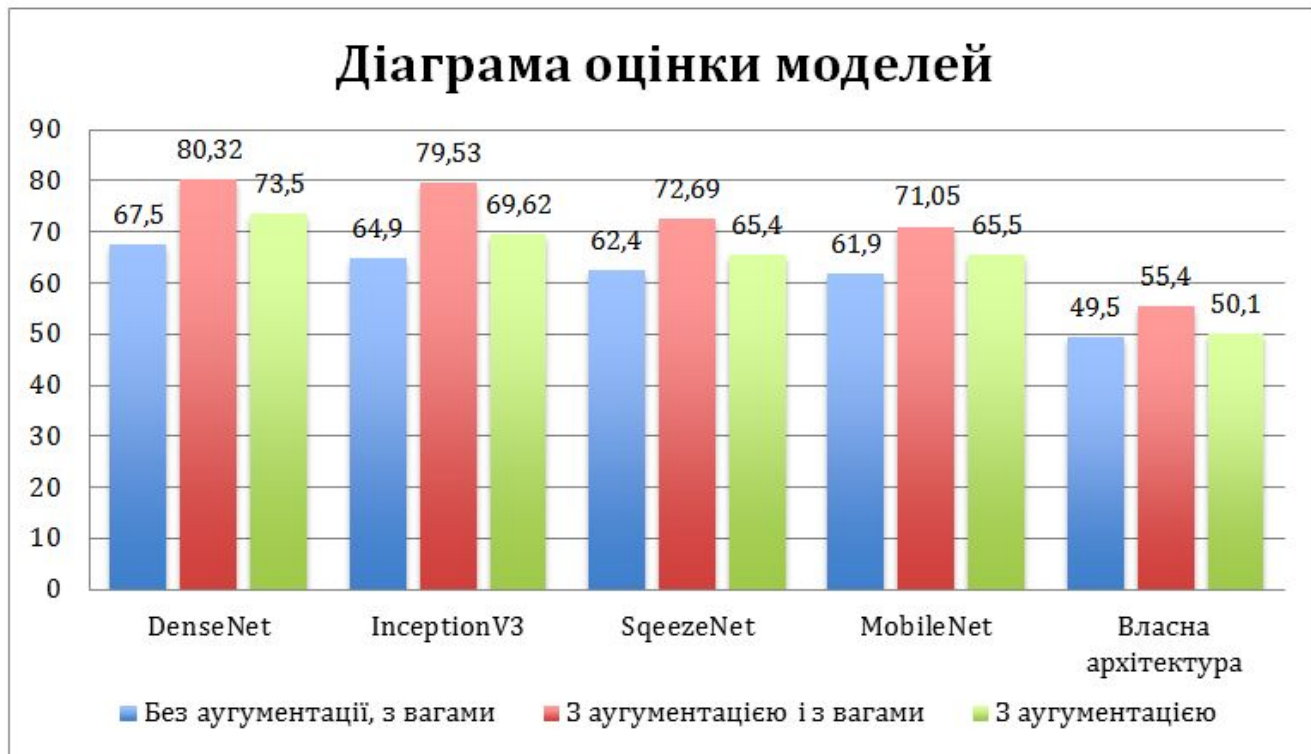


Оцінка якості моделі



- Всі моделі тестувались методом відкладеної вибірки.
- Всі дані були поділені на три частини і зафіксовані і в подальшому на цих частинах навчались всі обрані моделі

Порівняння результатів вибраних моделей

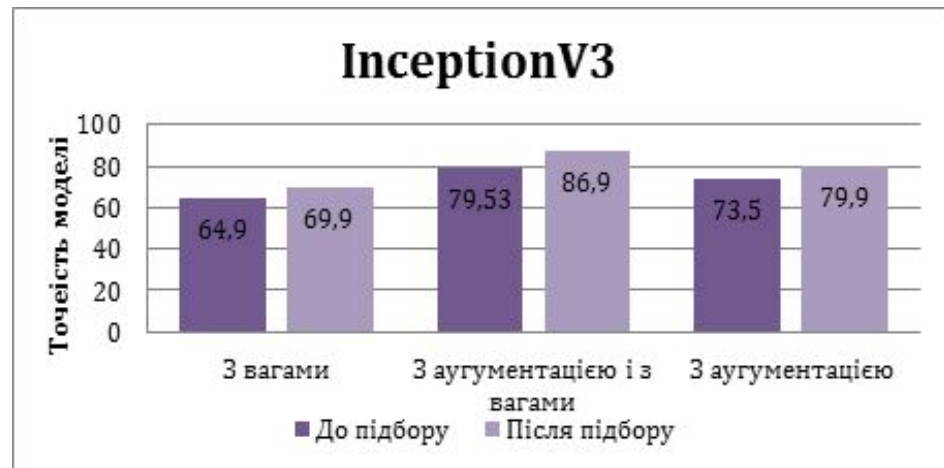
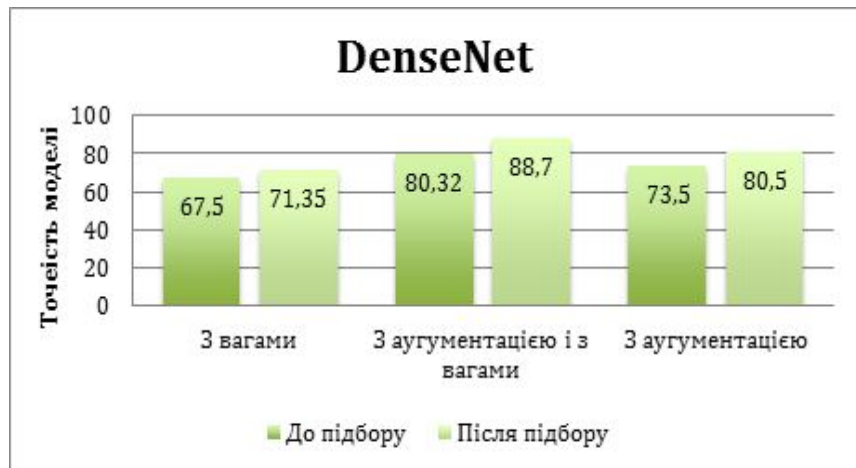


Порівняння моделей



Менше - краще

Діаграми оцінки результатів налаштованих моделей



Діаграми оцінки результатів налаштованих моделей

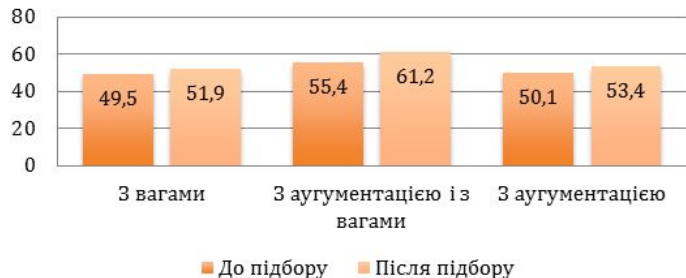
SqueezeNet



MobileNet



Власна архітектура



Обмін параметрами між моделями нейронної мережі

Видозмінена логарифмічна функція втрат

$$-\frac{1}{N} \sum_{i=0}^N \sum_{j=0}^M p_{ij} \log(q_{ij}) - \lambda \frac{1}{N} \sum_{i=0}^N \sum_{j=0}^M y_{ij} \log(q_{ij}|_{T=1})$$

де p_{ij} ймовірність того, що екземпляр i належить класу j $y_{ij}=1$, коли екземпляр i належить класу j , в іншому випадку $y_{ij}=0$;

N - кількість екземплярів

M - кількість класів

z_{ij} - виходи останнього лінійного шару;

T - коефіцієнт згладження;

λ - коефіцієнт значимості звичайної функції втрат

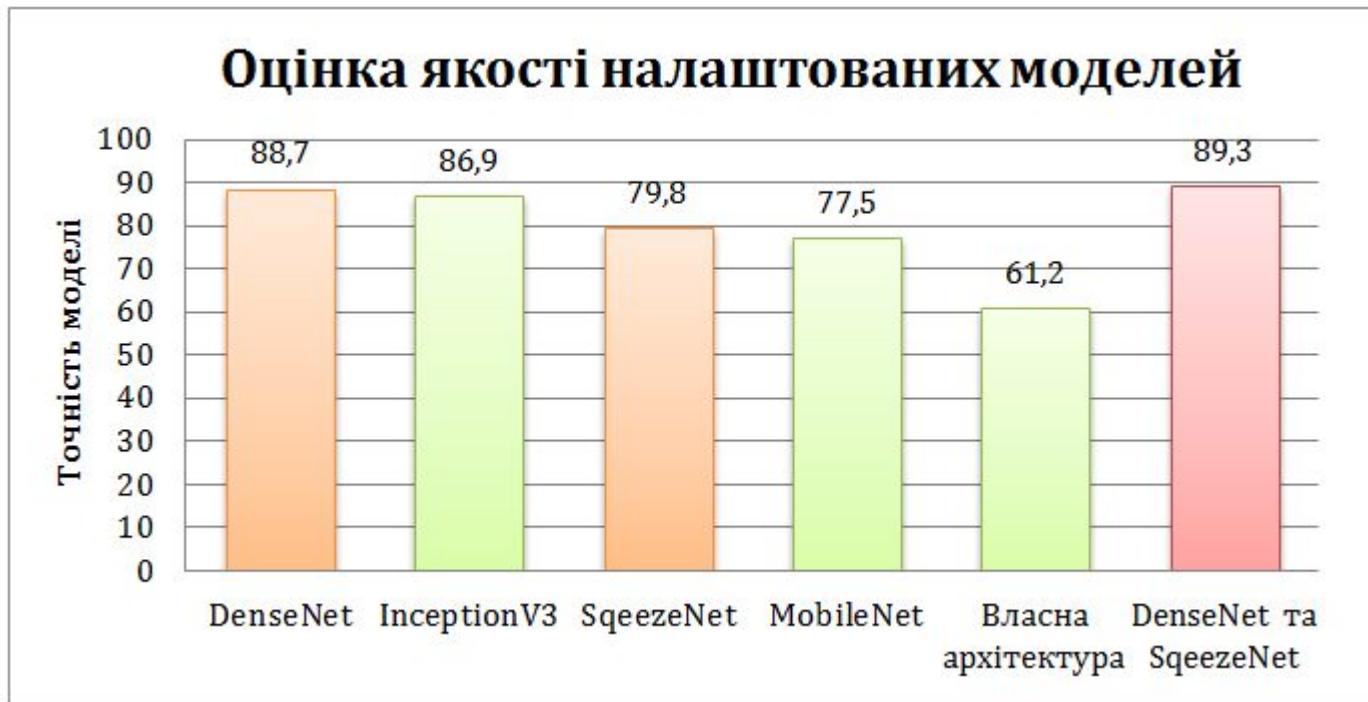
$$q_{ij} = \frac{\exp(z_{ij}/T)}{\sum_k \exp(z_{ik}/T)}$$

Пібрані параметри:

$$T = 2 \quad [2; 5]$$

$$\lambda = 0.3 \quad (0; 1)$$

Обмін досвідом DenseNet і SqueezeNet



Висновки

1. Досліджено існуючі методи та підходи до задач класифікації зображень на GPU
2. Розроблено власну архітектуру для класифікації зображень на GPU
3. Проведено аналіз та порівняння архітектур, оцінки якості різних моделей, функцій активації, оптимізатора, метрик якості на вибраному наборі даних предметів побуту в задачах класифікації зображень на GPU
4. Використано метод передачі знань для вибраних архітектур, підібрано параметри для моделей, для отримання найкращої якості та проведено аналіз отриманих результатів
5. Застосовано метод обміну досвіду DenseNet до SqueezeNet та підібрано параметри при яких було покращено отриману точність
6. Експериментально встановлено діапазон параметрів, для яких метод працював найкраще
7. Отримано модель, яка виконує задачу класифікації на 69 класів з точністю 89,3% та об'ємом 4,4 МБ, в той час як DenseNet - 430 МБ, та точність 88,7%

Дякую за увагу