МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ ЛЬВІВСЬКА ПОЛІТЕХНІКА ”
ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ

ЗВІТ

про виконання лабораторної роботи №3
з курсу “ Обробка зображень методами штучного інтелекту ”

Виконав:
ст. групи КН-410
Качмарик В. Р.
Перевірив:
Пелешко Д. Д.

Львів – 2022

Тема: Класифікація зображень. Застосування нейромереж для пошуку подібних зображень.

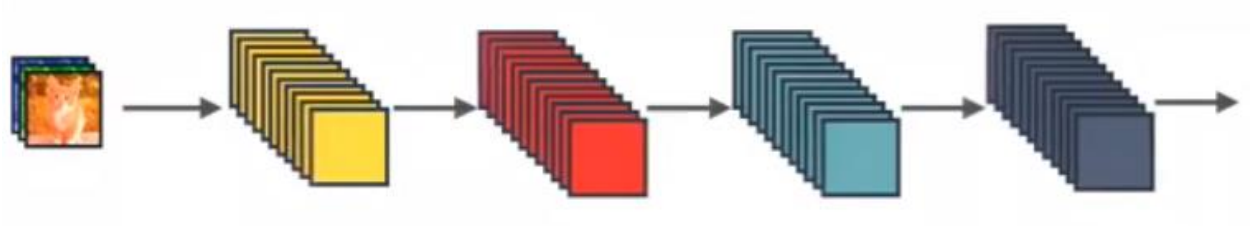
Мета: набути практичних навиків у розв’язанні задачі пошуку подібних зображень на прикладі організації CNN класифікації.

Варіант 11

11. Побудувати CNN на основі DenseNet для класифікації зображень на основі датасету fashion-mnist. Зробити налаштування моделі для досягнення необхідної точності. На базі Siamese networks побудувати систему для пошуку подібних зображень в датасеті fashion-mnist. Візуалізувати отримані результати t-SNE.

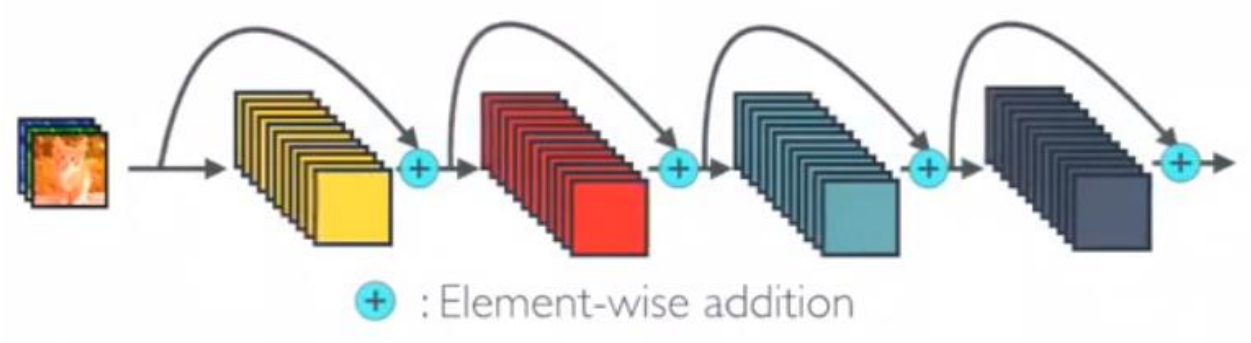
Теоретичні відомості

1. Dense Block



Стандартний концепт ConvNet

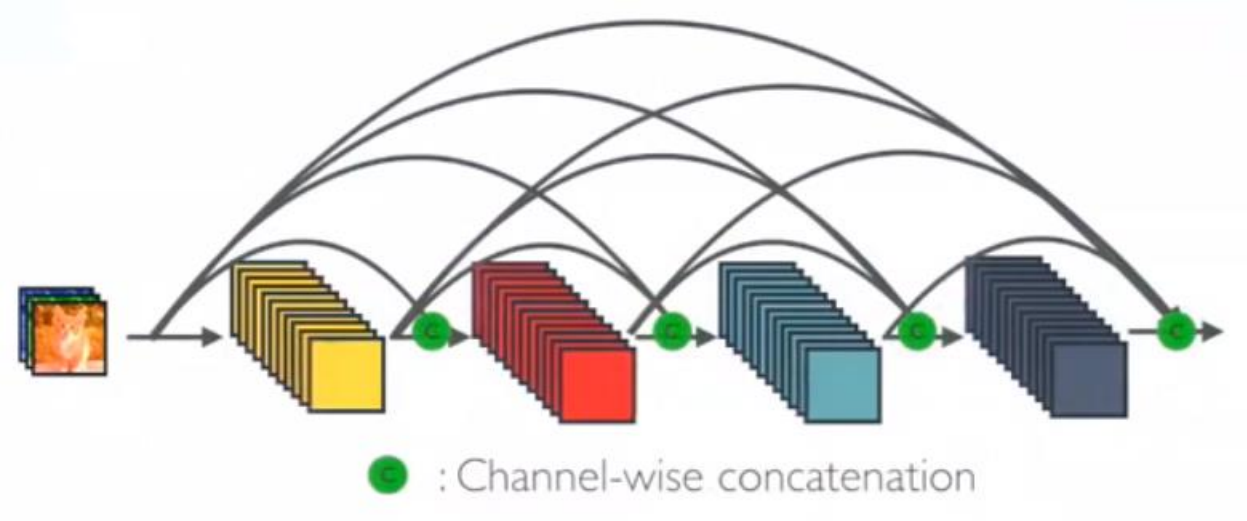
У стандартній ConvNet вхідне зображення проходить кілька згорток і отримує високо-рівневі фічі(features).



ResNet концепт

У ResNet відображення ідентичності пропонується, щоб сприяти поширенню градієнта. Для цього використовується по-елементне додавання.

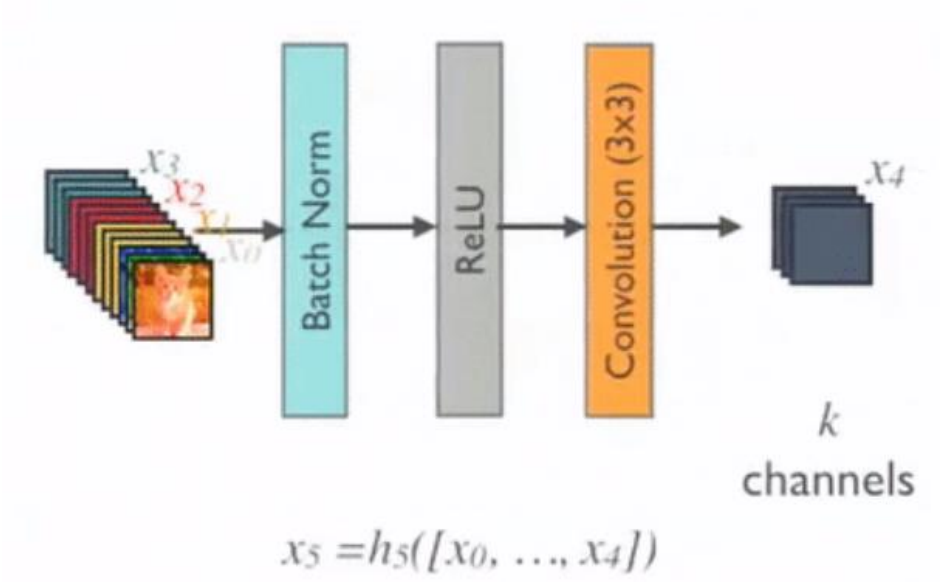
Його можна розглядати як алгоритми, стан яких передається від одного модуля ResNet до іншого.



Один Dense блок в DenseNet

У DenseNet кожен шар отримує додаткові вхідні дані з усіх попередніх шарів і передає свої власні feature-maps всім наступним шарам. Кожен рівень отримує «колективне знання» від усіх попередніх шарів.

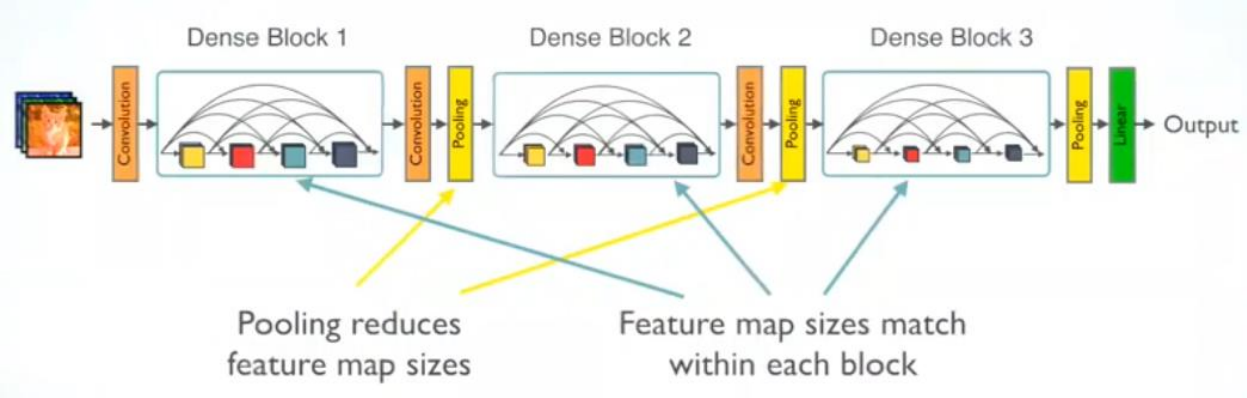
2. DenseNet архітектура



Composition Layer

Для кожного Composition Layer використовується Pre-Activation Batch Norm(BN) and ReLU, а потім 3×3 Conv з вихідними feature-maps з k каналами, для перетворення x0, x1, x2, x3 в x4. Це ідея від ResNet перед активацією.

2.1. Численні Dense Blocks з Transition Layers



Численні Dense Blocks

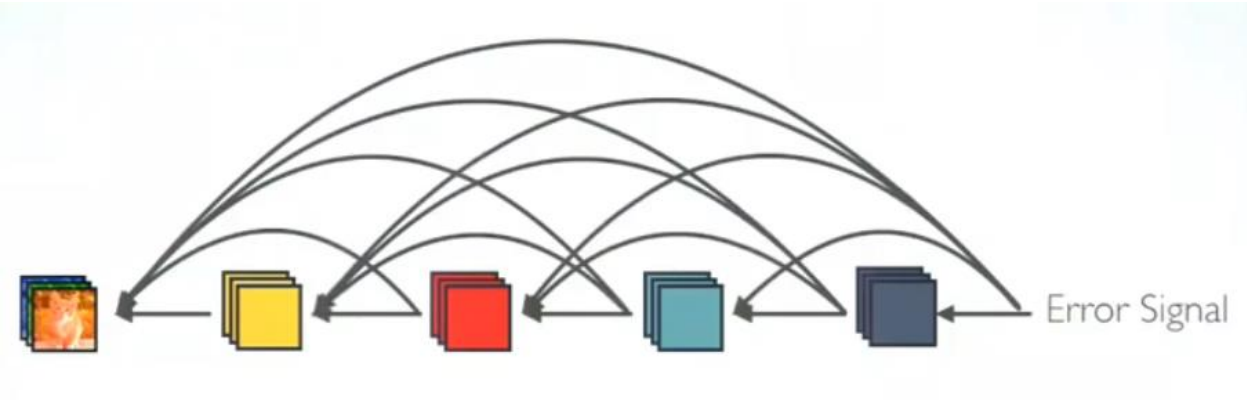
1×1 Conv з наступним 2×2 average pooling використовуються як перехідні шари між двома суміжними Dense Blocks.

Розміри feature map у Dense Blocks однакові, тому їх можна легко об’єднати.

В кінці останнього щільного блоку виконується глобальне average pooling, а потім додається класифікатор softmax.

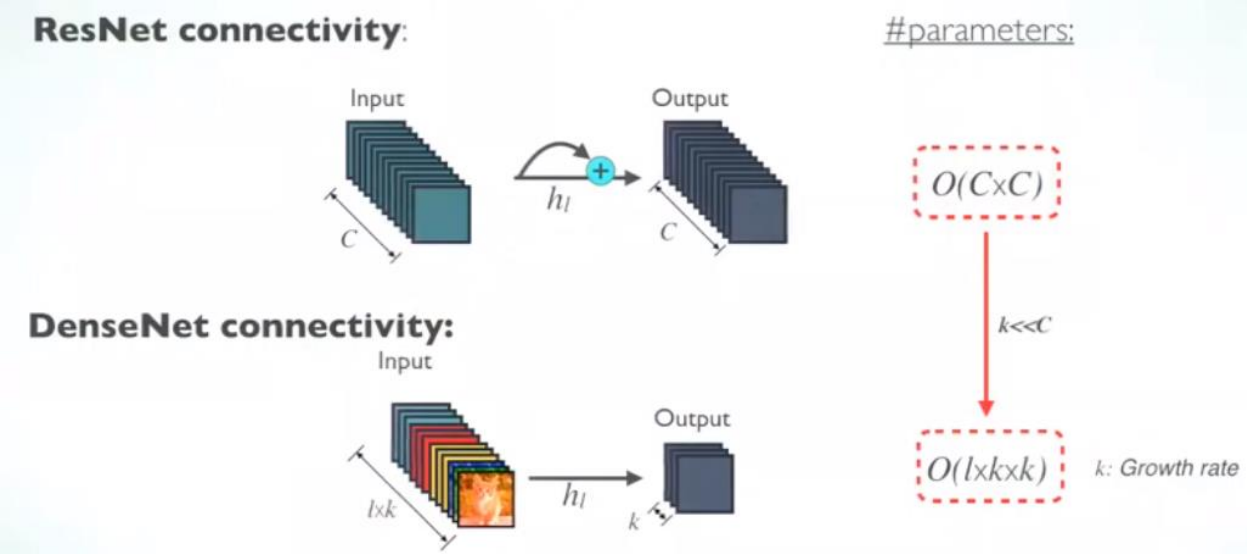
3. Переваги DenseNet

1) Strong Gradient Flow



Сигнал помилки може легко поширюватися на попередні шари більш безпосередньо. Це свого роду неявний deep supervision, оскільки попередні рівні можуть отримати прямий supervision з останнього рівня класифікації.

2) Параметри та обчислювальна ефективність

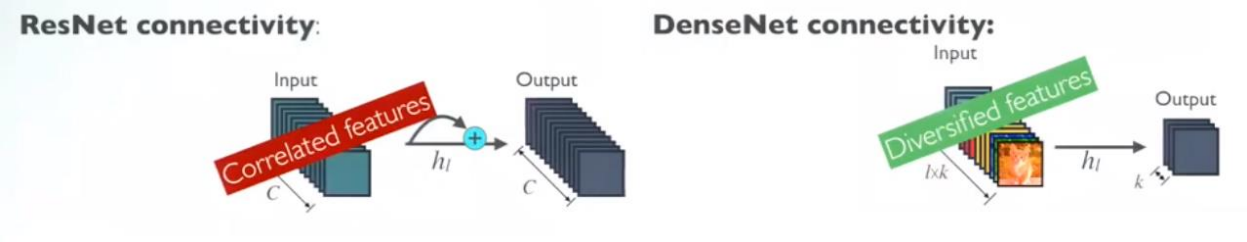


Кількість параметрів для ResNet і DenseNet

Для кожного рівня кількість параметрів у ResNet прямо пропорційна C×C, а кількість параметрів у DenseNet прямо пропорційна l×k×k.

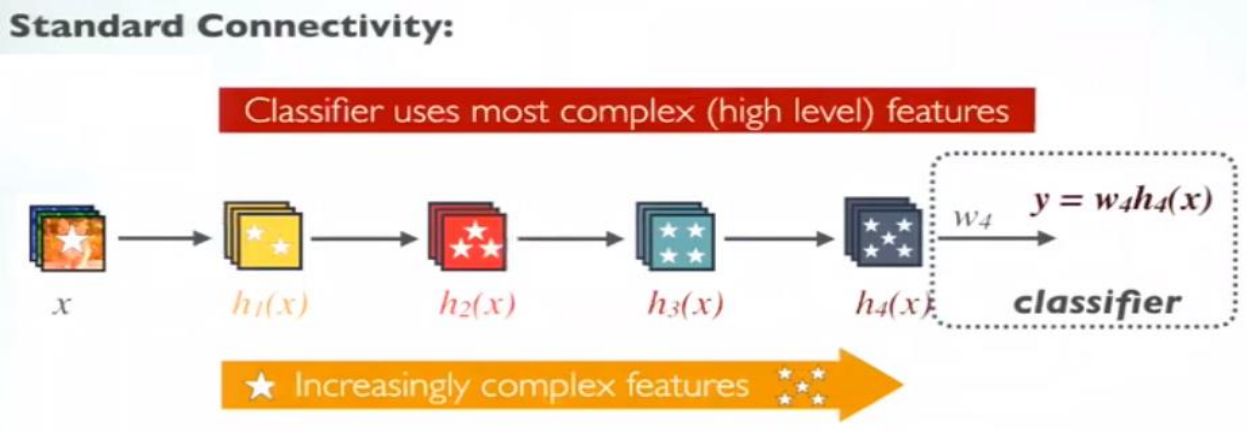
Оскільки k<<C, DenseNet має набагато менший розмір, ніж ResNet.

3) Більш різноманітні фічі(diversified features)

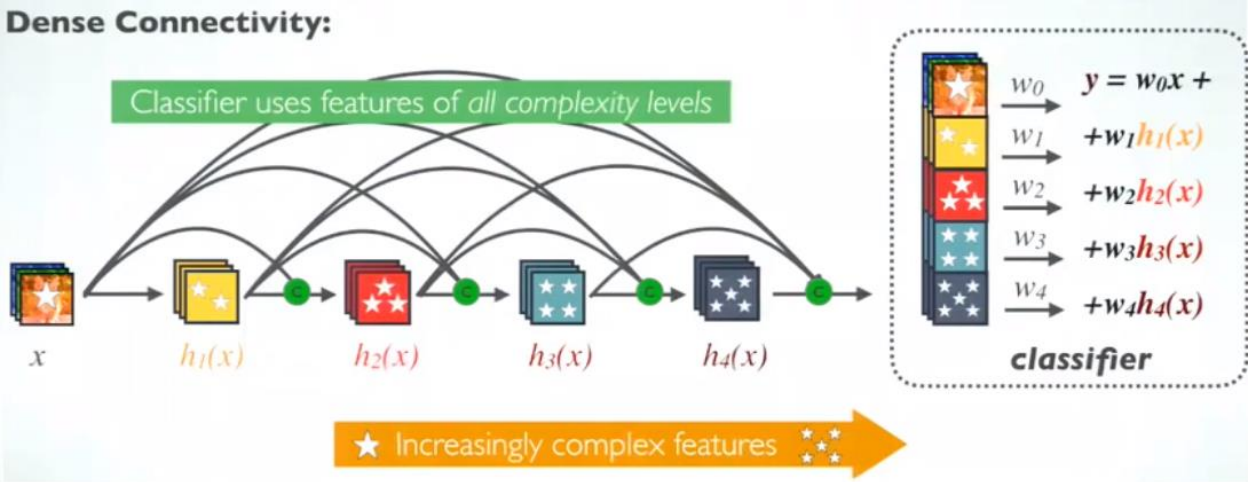


Оскільки кожен шар у DenseNet отримує всі попередні шари як вхідні дані, більш різноманітні фічі(features), як правило, мають багатші шаблони.

4) Зберігає фічі(features) низької складності



У Standard ConvNet класифікатор використовує найскладніші функції.



У DenseNet класифікатор використовує функції всіх рівнів складності. Він має тенденцію надавати більш гладкі межі рішень. Це також пояснює, чому DenseNet працює добре, коли навчальних даних недостатньо.

Виконання

Для виконання цієї лабораторної роботи я використав науково-обчислювальний пакет на основі Python – PyTorch, а саме претреновану модель “densenet121”:

model = torch.hub.load('pytorch/vision:v0.10.0', 'densenet121', pretrained=True).

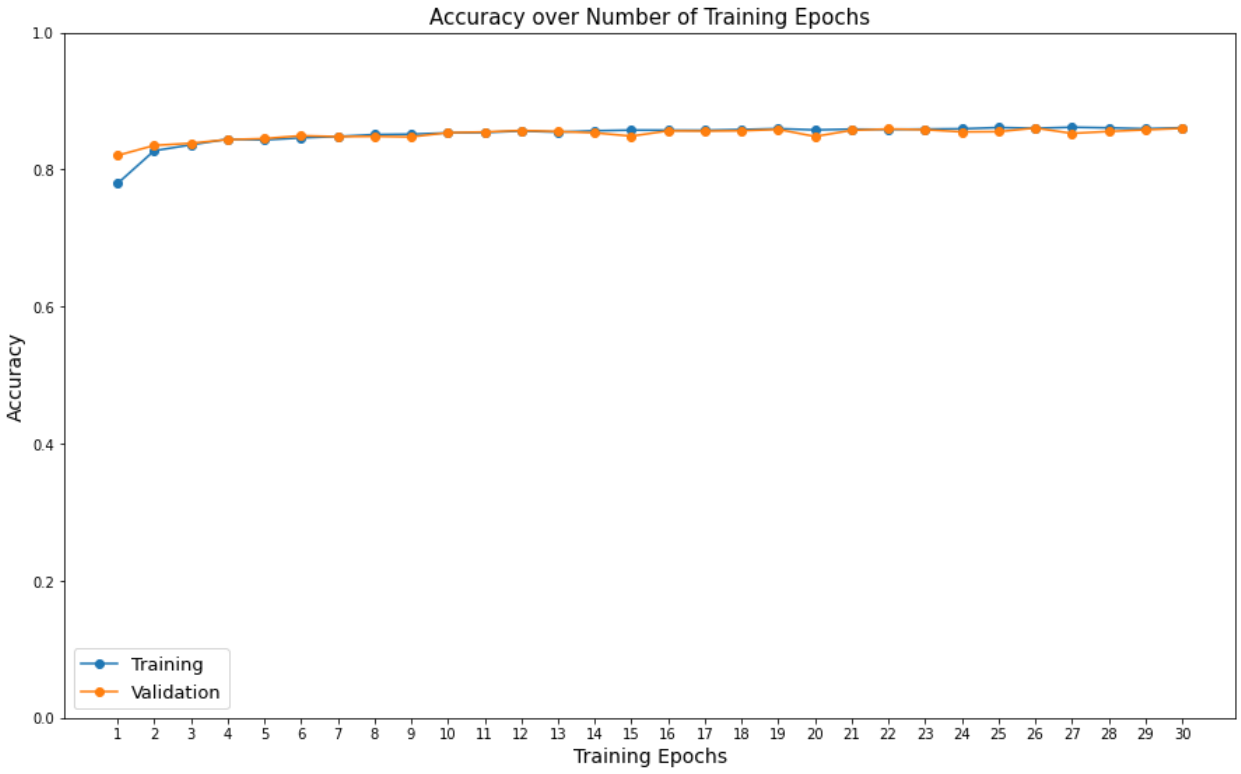
При завантаженні датасету fashion-mnist, додатково провів трансформацію даних, щоб можна було використовувати претреновані параметри мережі:

```
'train': transforms.Compose([transforms.Resize(256),
                             transforms.CenterCrop(input_size),
                             transforms.ToTensor(),
                             transforms.Lambda(lambda x: x.repeat(3, 1, 1)),
                             transforms.Normalize(mean=[0.485, 0.456, 0.406],
                                                  std=[0.229, 0.224, 0.225])
                            ])
```

Спершу змінюємо розмір вхідних зображень до розмірності (256 x 256), після чого обрізаємо до розміру, який використовувався при тренування мережі, після чого копіюємо вихідний шар, щоб утворити тришарове зображення(rgb) і нормалізуємо.

Оскільки завдання полягає в класифікації зображень на основі датасету fashion-mnist, останній шар(класифікаційний) цієї мережі було змінено – до вихідної кількісті фіч(out_features) було присвоєно кількість класів в fashion-mnist датасеті: (classifier): Linear(in_features=1024, out_features=10, bias=True).

Тренували модель протягом 30 епох, і отримали наступні результати:



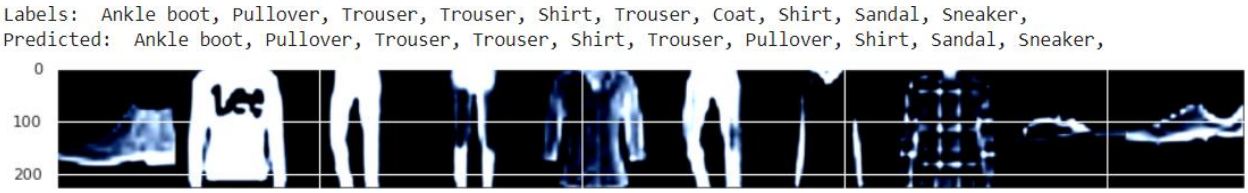
Статистика(precision, recall, f1-score, support):

**** Classification Report ****

	precision	recall	f1-score	support
T-shirt/top	0.81	0.84	0.83	1000
Trouser	0.99	0.96	0.97	1000
Pullover	0.76	0.80	0.78	1000
Dress	0.84	0.88	0.86	1000
Coat	0.74	0.79	0.76	1000
Sandal	0.95	0.95	0.95	1000
Shirt	0.66	0.54	0.59	1000
Sneaker	0.94	0.92	0.93	1000
Bag	0.95	0.97	0.96	1000
Ankle boot	0.95	0.96	0.95	1000
accuracy			0.86	10000
macro avg	0.86	0.86	0.86	10000
weighted avg	0.86	0.86	0.86	10000

Confusion matrix

Приклад передбачення:



Висновок

Під час виконання цієї лабораторної роботи, я набув практичних навиків у розв’язанні задачі пошуку подібних зображень на прикладі організації CNN класифікації. Для цього я побудував CNN на основі DenseNet для класифікації зображень на основі датасету fashion-mnist, та зробив налаштування моделі для досягнення необхідної точності.