

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА
ПОЛІТЕХНІКА»

ІНСТИТУТ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ



ЛАБОРАТОРНА РОБОТА №3
З ДИСЦИПЛІНИ “ОБРОБКА ЗОБРАЖЕНЬ МЕТОДАМИ
ШТУЧНОГО ІНТЕЛЕКТУ”

Виконав:
СТ. ГР. КН-408
ПЕТРОВСЬКИЙ О.С.

Викладач:
ПЕЛЕСЬКО 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.

Завдання

Побудувати CNN на основі **AlexNet** 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.

1. AlexNet

Нейронна мережа AlexNet складається з 8 шарів: перших 5 - згорткові (використовуються для виділення фіч), останніх 3 - лінійні (використовуються для знаходження закономірностей і визначення результуючого класу).

```
AlexNet = Sequential()
```

```
AlexNet.add(Conv2D(filters=96, input_shape=x_train.shape[1:],  
    kernel_size=(11,11), strides=(4,4), padding='same'))
```

```
AlexNet.add(BatchNormalization())
```

```
AlexNet.add(Activation('relu'))
```

```
AlexNet.add(MaxPooling2D(pool_size=(2,2), strides=(2,2),  
    padding='same'))
```

```
AlexNet.add(Conv2D(filters=256, kernel_size=(5, 5), strides=(1,1),  
    padding='same'))
```

```
AlexNet.add(BatchNormalization())
```

```
AlexNet.add(Activation('relu'))
```

```
AlexNet.add(MaxPooling2D(pool_size=(2,2), strides=(2,2),  
    padding='same'))
```

```
AlexNet.add(Conv2D(filters=384, kernel_size=(3,3), strides=(1,1),  
    padding='same'))
```

```
AlexNet.add(BatchNormalization())
```

```
AlexNet.add(Activation('relu'))
```

```
AlexNet.add(Conv2D(filters=384, kernel_size=(3,3), strides=(1,1),  
    padding='same'))
```

```
AlexNet.add(BatchNormalization())
```

```
AlexNet.add(Activation('relu'))
```

```
AlexNet.add(Conv2D(filters=256, kernel_size=(3,3), strides=(1,1),  
    padding='same'))
```

```
AlexNet.add(BatchNormalization())
```

```

AlexNet.add(Activation('relu'))
AlexNet.add(MaxPooling2D(pool_size=(2,2), strides=(2,2),
padding='same'))

AlexNet.add(Flatten())
AlexNet.add(Dense(4096, input_shape=(32,32,3)))
AlexNet.add(BatchNormalization())
AlexNet.add(Activation('relu'))
AlexNet.add(Dropout(0.4))

AlexNet.add(Dense(4096))
AlexNet.add(BatchNormalization())
AlexNet.add(Activation('relu'))
AlexNet.add(Dropout(0.4))

AlexNet.add(Dense(1000))
AlexNet.add(BatchNormalization())
AlexNet.add(Activation('relu'))
AlexNet.add(Dropout(0.4))

AlexNet.add(Dense(10))
AlexNet.add(BatchNormalization())
AlexNet.add(Activation('softmax'))

```

2. Siamese Networks

Сіамські мережі побудовані наступним чином: використовуються 2 AlexNet з окремими входами, спільним виходом і вагами. На виході кожна AlexNet продукує вектори передбачень, які можна порівнювати між собою з метою визначення схожості вхідних зображень.

```

img_a_in = Input(shape=x_train.shape[1:], name='ImageA_Input')
img_b_in = Input(shape=x_train.shape[1:], name='ImageB_Input')
img_a_feat = AlexNet(img_a_in)
img_b_feat = AlexNet(img_b_in)

```

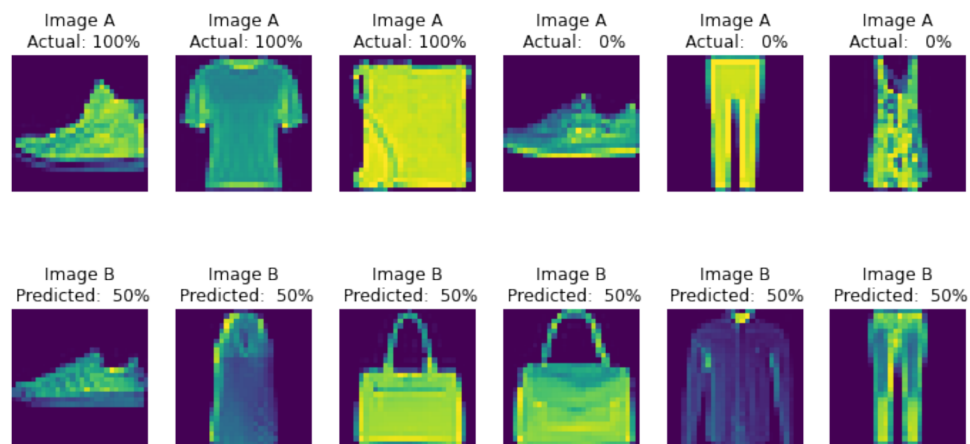
```

combined_features = concatenate(
    [img_a_feat, img_b_feat], name='merge_features')
combined_features = Dense(16, activation='linear')(combined_features)
combined_features = BatchNormalization()(combined_features)
combined_features = Activation('relu')(combined_features)
combined_features = Dense(4, activation='linear')(combined_features)
combined_features = BatchNormalization()(combined_features)
combined_features = Activation('relu')(combined_features)
combined_features = Dense(1, activation='sigmoid')(combined_features)
similarity_model = Model(inputs=[img_a_in, img_b_in], outputs=[
    combined_features], name='Similarity_Model')

```

3. Експериментальна частина

Випадково ініціалізовані сіамські мережі не дали ніякого корисного результату.

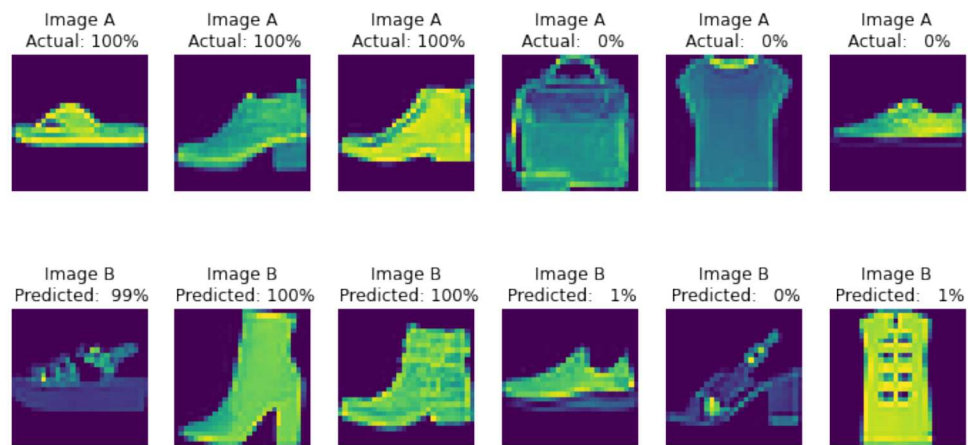


Проте після 25 епох тренування моделлю була досягнута точність близька до 80%.

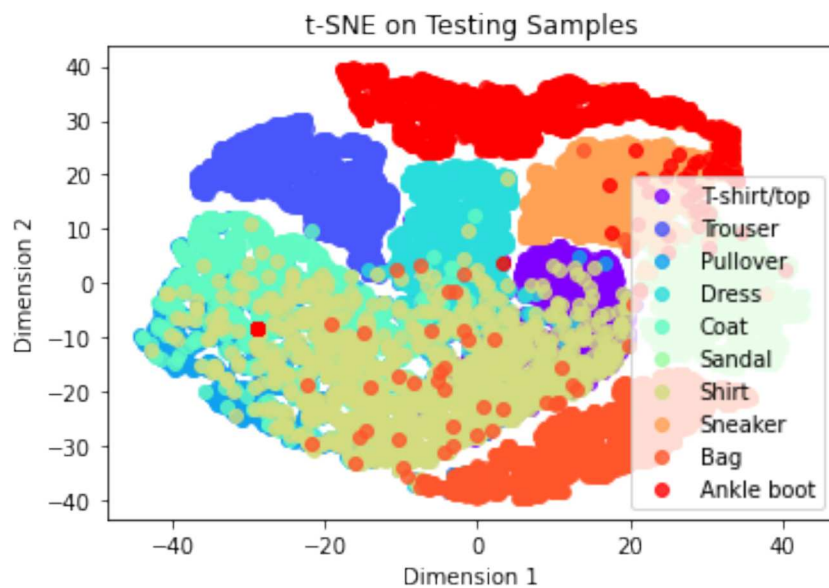
```

Epoch 21/25
500/500 [=====] - 546s 1s/step - loss: 0.2290 - mae: 0.1406 - val_loss: 0.2157 - val_mae: 0.1315
Epoch 22/25
500/500 [=====] - 546s 1s/step - loss: 0.2296 - mae: 0.1407 - val_loss: 0.2304 - val_mae: 0.1323
Epoch 23/25
500/500 [=====] - 547s 1s/step - loss: 0.2244 - mae: 0.1371 - val_loss: 0.2248 - val_mae: 0.1292
Epoch 24/25
500/500 [=====] - 547s 1s/step - loss: 0.2233 - mae: 0.1369 - val_loss: 0.2120 - val_mae: 0.1272
Epoch 25/25
500/500 [=====] - 546s 1s/step - loss: 0.2143 - mae: 0.1314 - val_loss: 0.2239 - val_mae: 0.1234

```



Використання t-SNE продемонструвало, що моделі, що генерували фічі (AlexNet), змогли досить добре виокремити ознаки чобіт, пуловерів, штанів, сумок, сандалей та кросівок (які є візуально відмінними і легшими до розрізнення). Проте, вони з меншою впевненістю розрізняють сукні, плащі, сорочки і футболки (ймовірно через зовнішню схожість). Також деякі плаття плутаються з сумками.



Висновки

У результаті виконання лабораторної роботи я набув практичних навиків у розв'язанні задачі пошуку подібних зображень на прикладі організації CNN класифікації.