**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**



**Лабораторна робота №3**

з дисципліни

«Обробка зображень методами штучного інтелекту»

**Виконав:**

студент групи КН-409

Дрібнюк Ярослав

**Викладач:**

д.т.н., професор

Пелешко Д.Д.

Львів – 2022 р.

**Лабораторна робота № 3**

**КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ. ЗАСТОСУВАННЯ НЕЙРОМЕРЕЖ ДЛЯ ПОШУКУ ПОДІБНИХ ЗОБРАЖЕНЬ.**

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

**Завдання до виконання роботи**

Побудувати CNN на основі Inception-ResNets 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.

**Хід роботи**

1. Завантажуємо дані та проводимо попередню обробку:

x\_train = np.expand\_dims(x\_train, axis=-1)

x\_train = tf.image.resize(x\_train, [84,84]).numpy()

#x\_train = np.repeat(x\_train, 3, axis=3)

x\_test = np.expand\_dims(x\_test, axis=-1)

x\_test = tf.image.resize(x\_test, [84,84]).numpy()

#x\_test = np.repeat(x\_test, 3, axis=3)

y\_train = y\_train.astype('int')

y\_test = y\_test.astype('int')

print('Training', x\_train.shape, x\_train.max())

print('Testing', x\_test.shape, x\_test.max())

1. Проводимо невеликі зміни в моделі з keras:

base\_model = InceptionResNetV2(

    weights=None,

    include\_top=False,

    input\_shape=(84, 84, 1),

    classes=10

)

x = base\_model.output

x = GlobalAveragePooling2D()(x)

x = Dense(1024, activation = 'relu')(x)

x = Dropout(0.5)(x)

predictions = Dense(36, activation='relu')(x)

model = Model(inputs=base\_model.input, outputs=predictions)

model.summary()

1. Siamese networks:

from keras.layers import concatenate

from keras.models import Model

from keras.layers import Input, Conv2D, BatchNormalization, MaxPool2D, Activation, Flatten, Dense, Dropout

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 = model(img\_a\_in)

img\_b\_feat = model(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)

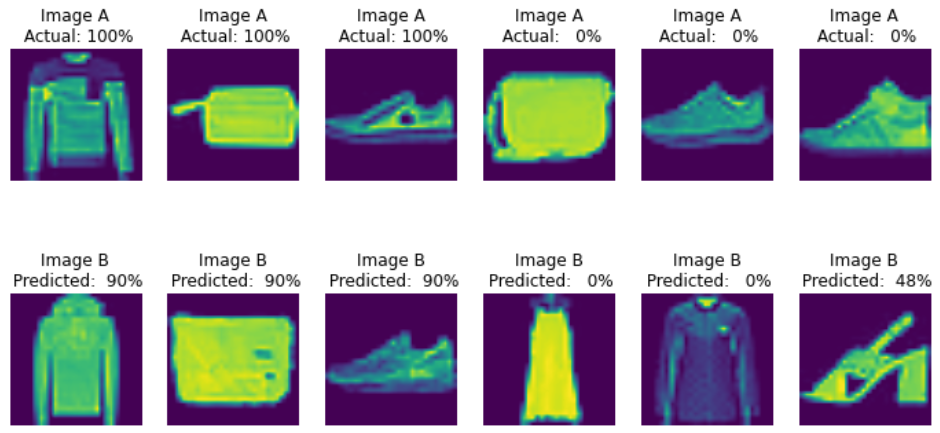
combined\_features = Activation('relu')(combined\_features)

combined\_features = Dense(1, activation = 'sigmoid')(combined\_features)

similarity\_model = Model(inputs = [img\_a\_in, img\_b\_in], outputs=[combined\_features], name = 'Similarity\_Model')

similarity\_model.summary()

1. Результат роботи:



**Висновки:** при виконанні цієї лабораторної роботи я набув практичних навиків у розв’язанні задачі пошуку подібних зображень на прикладі організації CNN класифікації з використанням Siamese networks.