Dongguk University
Division of Electronics and Electrical Engineering
Network Intelligence Lab.

Leveraging Deep Learning for Product Similarit Analysis

Masoumeh Mohammadi Doghozloo

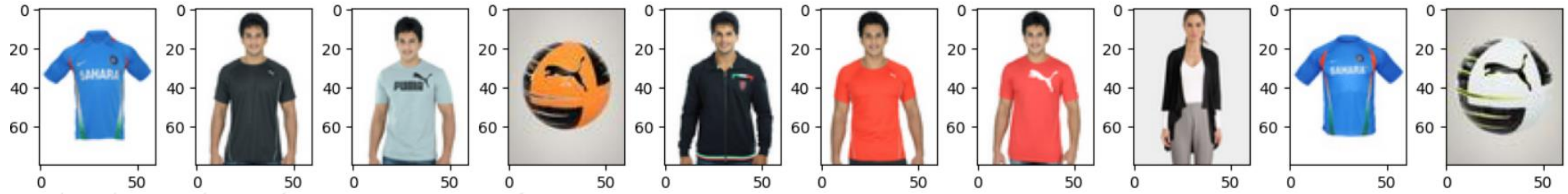
Problem Statement: identifying the most similar products.

1. Exploratory Data Analysis
2. Data Preprocessing
3. Building a Deep Learning Model for Image Processing
4. Computing Embeddings for Images
5. Calculating Similarity between Products
6. Normalization of Similarity Scores
7. Generating Recommendations
8. Visualization of Recommendations

1 Exploratory Data Analysis

- Explore the dataset to understand its structure, characteristics, and distributions.
- Visualize sample images from the dataset

```
[ ]  path=glb('/content/drive/MyDrive/images/*.jpg')
     # Plot samples
     plt.figure(figsize=(20,20))
     for i in range(20,30):
         plt.subplot(6, 10, i-10+1)
         cloth_img =  mpimg.imread(path[i])
         plt.imshow(cloth_img)
     plt.subplots_adjust(wspace=-0.5, hspace=1)
     plt.show()
```



1 Exploratory Data Analysis

id	gender	masterCategory	subCategory	articleType	baseColour	season	year	usage	productDisplayName
1163	Men	Apparel	Topwear	Tshirts	Blue	Summer	2011	Sports	Nike Sahara Team India Fanwear Round Neck Jersey
1164	Men	Apparel	Topwear	Tshirts	Blue	Winter	2015	Sports	Nike Men Blue T20 Indian Cricket Jersey
1165	Men	Apparel	Topwear	Tshirts	Blue	Summer	2013	Sports	Nike Mean Team India Cricket Jersey
1525	Unisex	Accessories	Bags	Backpacks	Navy Blue	Fall	2010	Casual	Puma Deck Navy Blue Backpack

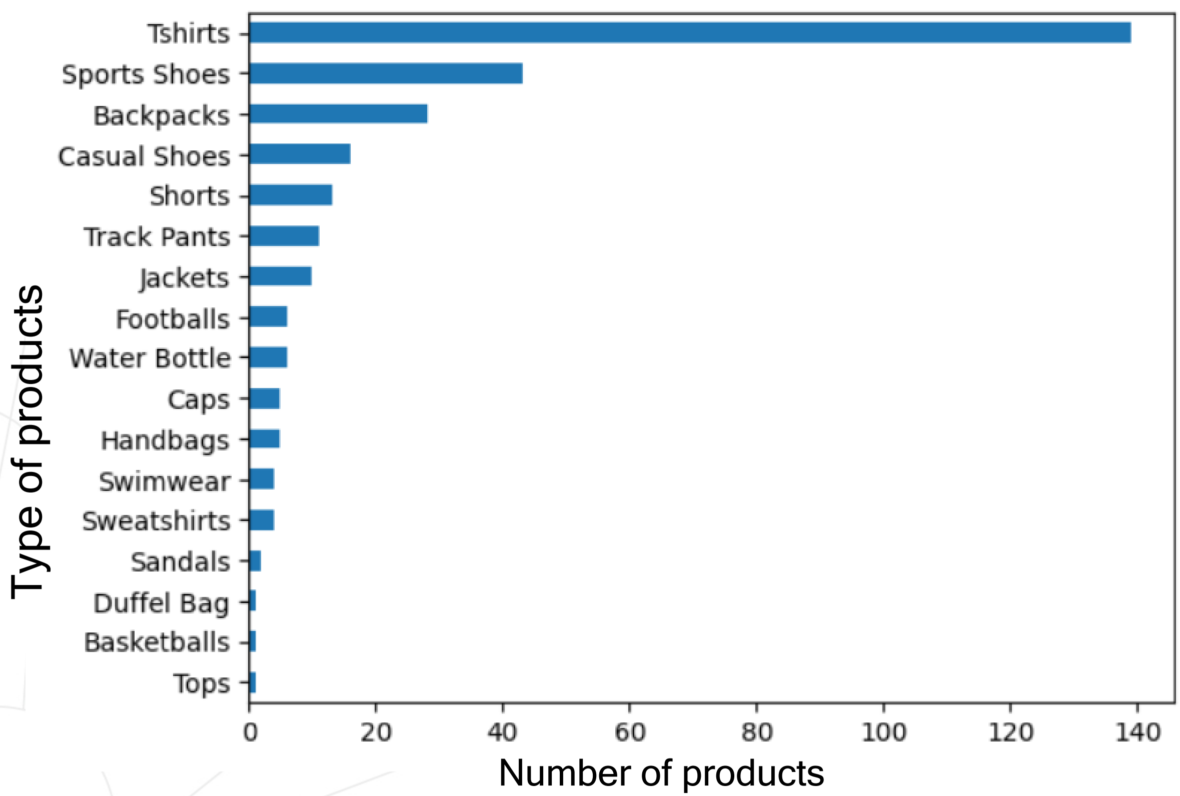
```python
styles_df = pd.read_csv("/content/drive/MyDrive/styles.csv", nrows=296)
styles_df['image'] = styles_df.apply(lambda x: str(x['id']) + ".jpg", axis=1)
print(styles_df.shape)
print(styles_df.head(5))
```

```
(295, 11)
     id  gender masterCategory subCategory articleType baseColour  season  \
0  1163     Men        Apparel     Topwear     Tshirts       Blue  Summer
1  1164     Men        Apparel     Topwear     Tshirts       Blue  Winter
2  1165     Men        Apparel     Topwear     Tshirts       Blue  Summer
3  1525  Unisex    Accessories        Bags   Backpacks  Navy Blue    Fall
4  1526  Unisex    Accessories        Bags   Backpacks      Black    Fall

     year   usage                             productDisplayName     image
0  2011.0  Sports  Nike Sahara Team India Fanwear Round Neck Jersey  1163.jpg
1  2015.0  Sports           Nike Men Blue T20 Indian Cricket Jersey  1164.jpg
2  2013.0  Sports               Nike Mean Team India Cricket Jersey  1165.jpg
3  2010.0  Casual                      Puma Deck Navy Blue Backpack  1525.jpg
4  2010.0  Sports                      Puma Big Cat Backpack Black  1526.jpg
```

1 Exploratory Data Analysis

```
[ ] styles_df.articleType.value_counts().sort_values().plot(kind='barh')
```



2 Data Preprocessing

- Resize images to a standard size and perform any necessary normalization or augmentation.

- Preprocess the dataset to ensure that the images are in a suitable format for input into the deep learning model.

```
img_width, img_height, chnls = 100, 100, 3
```

```
def predict(model, img_name):

    # Reshape
    img = image.load_img(img_path(img_name), target_size=(img_width, img_height))
    # img to Array
    img = image.img_to_array(img)
    # Expand Dim (1, w, h)
    img = np.expand_dims(img, axis=0)
    # Pre process Input
    img = preprocess_input(img)
    return model.predict(img)
```

3 Building a Deep Learning Model for Image Processing

ELSEVIER

Engineering Applications of Artificial Intelligence

Volume 133, Part A, July 2024, 108062

Survey paper

CNNRec: Convolutional Neural Network based recommender systems - A survey

Ronakkumar Patel [a, b], Priyank Thakkar [a], Vijay Ukani [a]

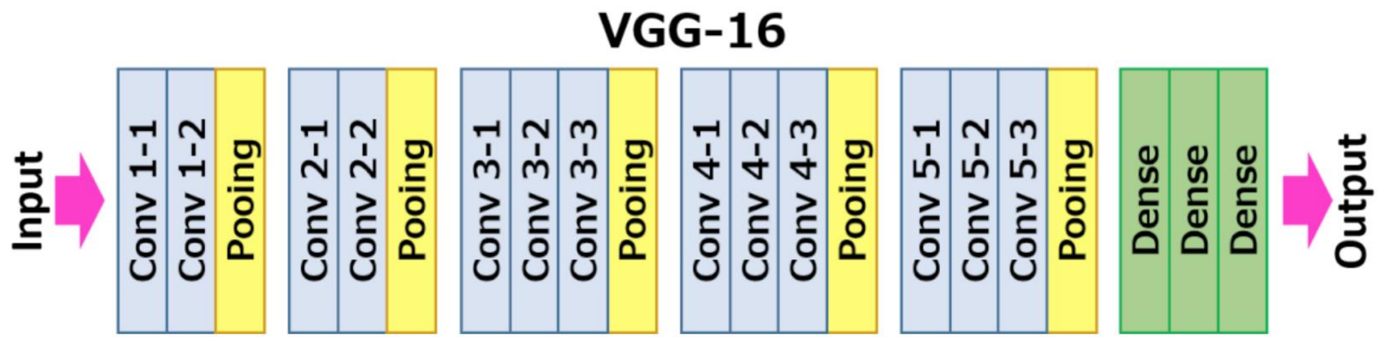
SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 4, October 2023
DOI : https://doi.org/10.33395/sinkron.v8i4.12936

e-ISSN : 2541-2019
p-ISSN : 2541-044X

Electronic Product Recommendation System Using the Cosine Similarity Algorithm and VGG-16

3 Building a Deep Learning Model for Image Processing




```python
#VGG16
from tensorflow.keras.applications import VGG16

vgg16 = VGG16(include_top=False, weights='imagenet', input_shape=(img_width, img_height, chnls))
vgg16.trainable=False
vgg16_model = keras.Sequential([vgg16, GlobalMaxPooling2D()])
vgg16_model.summary()
```

feature extraction

4 Computing Embeddings for Images

An embedding for images is a vector representation of an image in a lower-dimensional space where similar images are closer together.

```
def get_embeddings(df, model):

    df_copy = df
    df_embeddings = df_copy['image'].apply(lambda x: predict(vgg16_model, x).reshape(-1))
    df_embeddings = df_embeddings.apply(pd.Series)
    return df_embeddings
```

```
df_embeddings = get_embeddings(styles_df, vgg16_model)
```

•Use the trained deep learning model to generate embeddings for each image in the dataset.

•Define functions to preprocess images, make predictions using the deep learning model, and obtain embeddings.

5 Calculating Similarity between Products

```
url="/content/drive/MyDrive/images/1995.jpg"
a = plt.imread(url)
plt.imshow(a)
```

sample_image:

1995

```
sample_image = predict(vgg16_model, '1995.jpg')
sample_image.shape

df_sample_image = pd.DataFrame(sample_image)
print(df_sample_image)

sample_similarity = linear_kernel(df_sample_image, df_embeddings)
print(sample_similarity)
```

- Compute the similarity between products based on their embeddings.
- Utilize similarity metrics like linear kernel to measure the similarity between embeddings.

linear_kernel:The linear kernel computes the dot product between the vectors, which is a measure of their similarity.

```
[ ] def get_similarity(model):

        sample_image = predict(vgg16_model, '1995.jpg')
        df_sample_image = pd.DataFrame(sample_image)
        sample_similarity = linear_kernel(df_sample_image, df_embeddings)
        return sample_similarity
```

6 Normalization of Similarity Scores

1. Normalize the similarity scores to ensure that they are on a consistent scale, typically between 0 and 1.

2. Normalization helps in comparing similarity scores across different pairs of products.

```
[ ] def normalize_sim(similarity):

        x_min = similarity.min(axis=1)
        x_max = similarity.max(axis=1)
        norm = (similarity-x_min)/(x_max-x_min)[:, np.newaxis]
        return norm
```

```
[ ] sample_similarity_norm = normalize_sim(sample_similarity)
    sample_similarity_norm.shape
    sample_similarity_norm
```

```
0.9207467 , 0.62337416, 0.327665  , 0.6623712 , 0.9332883 ,
0.52291435, 0.6734664 , 0.7804008 , 0.07126626, 0.08768513,
0.0245542 , 0.06237724, 0.09645419, 0.08294881, 0.07084358,
0.09520397, 0.10327026, 0.09640364, 0.12888147, 0.05487006,
0.06148893, 0.1696546 , 0.29190215, 0.22282305, 0.1390158 ,
0.06813246, 0.07620154, 0.32562327, 0.63153857, 0.649822  ,
0.11243404, 0.3766658 , 0.23266809, 0.25795764, 0.45703053,
0.11765788, 0.13071674, 0.42392522, 0.6174044 , 0.52062076,
0.6398321 , 0.72773147, 0.33833644, 0.38354254, 0.37370875,
```

7 Generating Recommendations

•Identify the most similar products based on the normalized similarity scores.
•Define a function to get recommendations for a given input image by selecting the top similar products from the dataset.

```
def get_recommendations(df, similarity):

    sim_scores = list(enumerate(similarity[0]))

    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)

    sim_scores = sim_scores[0:5]
    print(sim_scores)

    cloth_indices = [i[0] for i in sim_scores]

    return df['image'].iloc[cloth_indices]
```

```
[ ] recommendation = get_recommendations(styles_df, sample_similarity_norm)
    recommendation_list = recommendation.to_list()
```

Visualization of Recommendations

Visualize the recommended products using matplotlib to provide users with a visual representation of the recommendations.

```
#recommended images
plt.figure(figsize=(20,20))
j=0
for i in recommendation_list:
    plt.subplot(6, 10, j+1)
    cloth_img =  mpimg.imread('/content/drive/MyDrive/images/'+ i)
    plt.imshow(cloth_img)
    plt.axis("off")
    j+=1
plt.title("Recommended images",loc='left')
plt.subplots_adjust(wspace=-0.5, hspace=1)
plt.show()
```

[(290, 1.0), (219, 0.973908), (127, 0.93346256), (14, 0.9332883), (10, 0.9207467)]

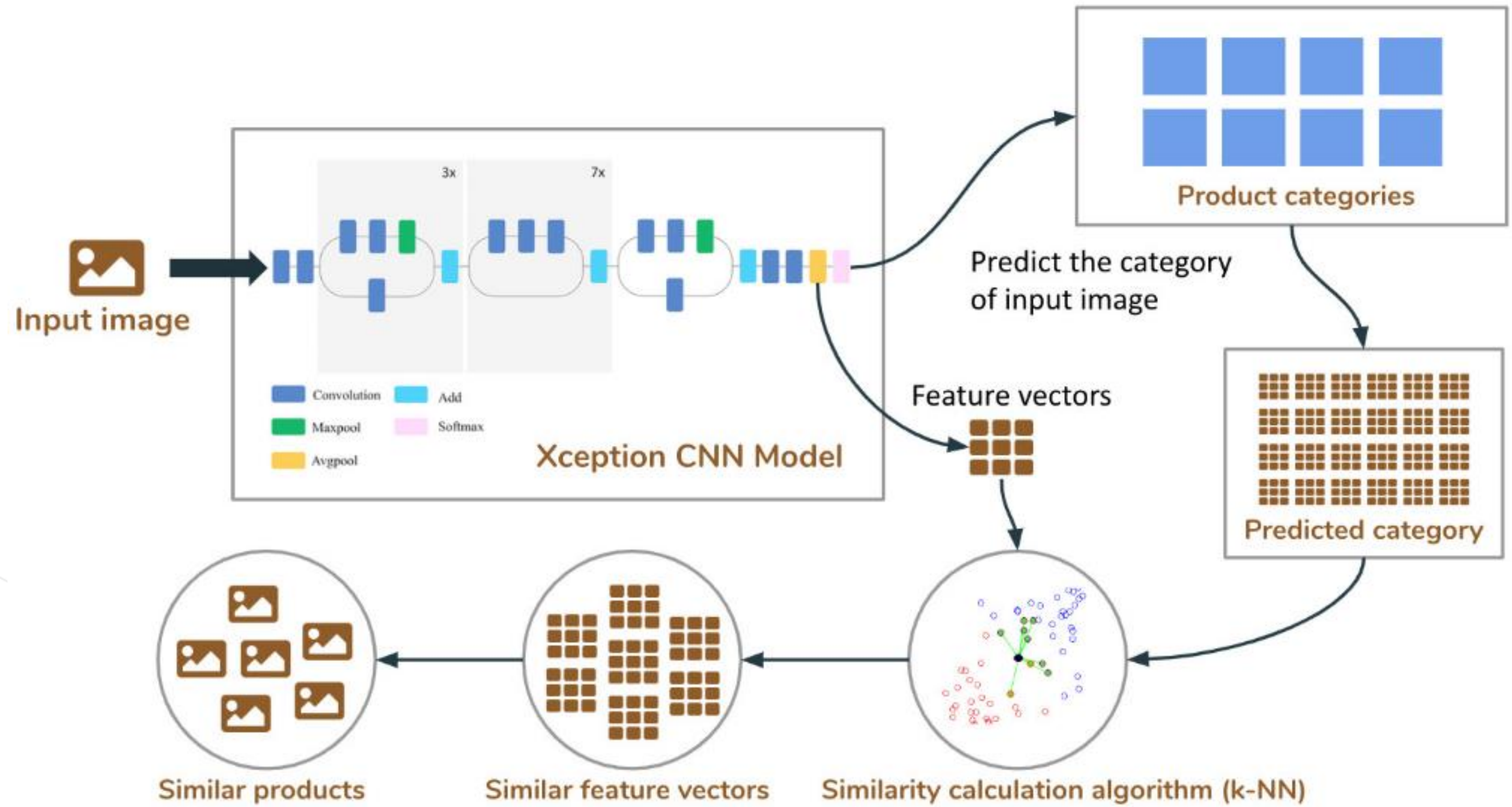
Discussion and Conclusions

- The project leverages deep learning techniques, particularly the VGG16 model, to process images of clothing items. Unlike traditional recommendation systems that rely on user purchase history,

- With image processing techniques, we can extract features from products and then identify similar products by comparing these features. This allows us to provide visually appealing recommendations to users.

- Utilize other feature extraction machine learning models, such as Transformers, ResNet, and MobileNet, then compare the results to evaluate their performance.

Future project:



동국대학교
dongguk university

Thank you!